

**UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**

CARRERA: INGENIERÍA DE SISTEMAS INFORMÁTICOS

**TRABAJO DE GRADUACIÓN: “INVESTIGACIÓN, ANÁLISIS,
DISEÑO Y ELABORACIÓN DE UN LABORATORIO DIDÁCTICO PARA
LA PROGRAMACIÓN DE MICROPROCESADORES EN LA CARRERA
DE INGENIERÍA DE SISTEMAS INFORMÁTICOS DE LA FACULTAD
MULTIDISCIPLINARIA DE OCCIDENTE”**

PRESENTAN:

**DUBÓN CASTRO MARÍA ELISA
GUZMAN TURCIOS ALICIA CAROLINA
LÓPEZ MENÉNDEZ RINA ELIZABETH**

DOCENTE DIRECTOR:

ING. JOSÉ FRANCISCO ANDALUZ

**16 DE FEBRERO
DE 1841**

**HACIA LA
LIBERTAD**

SANTA ANA, 9 DE OCTUBRE DE 2009.

AUTORIDADES DE LA UNIVERSIDAD DE EL SALVADOR

MÁSTER RUFINO ANTONIO QUEZADA SÁNCHEZ
RECTOR

MÁSTER MIGUEL ÁNGEL PÉREZ RAMOS
VICERRECTOR ACADÉMICO

MÁSTER ÓSCAR NOÉ NAVARRETE
VICE RECTOR ADMINISTRATIVO

LIC. DOUGLAS VLADIMIR ALFARO CHÁVEZ
SECRETARIO GENERAL

**AUTORIDADES DE LA
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE**

LIC. JORGE MAURICIO RIVERA
DECANO

LIC. ELADIO ZACARÍAS ORTEZ
VICE DECANO

LIC. VÍCTOR HUGO MERINO QUEZADA
SECRETARIO

DEPARTAMENTO DE INGENIERÍA

ING. RAUL ERNESTO MARTÍNEZ BERMÚDEZ
JEFE DE DEPARTAMENTO

ING. JOSÉ FRANCISCO ANDALUZ
DOCENTE DIRECTOR

AGRADECIMIENTOS

A DIOS SER SUPREMO

Por permitirme la vida y darme las fuerzas necesarias para seguir adelante sin dejar que los obstáculos me detuvieran.

A MI MAMI

Por darme la vida, su paciencia además de ser un apoyo incondicional; el cual me ayudó a seguir adelante para poder culminar mi carrera con satisfacción, este triunfo es dedicado con mucho amor a ella, que es una persona súper especial en mi vida y lo será siempre.

A MIS FAMILIARES

Por darme el apoyo necesario siempre que lo necesite y estar ahí en todos mis éxitos y fracasos.

A MIS COMPAÑERAS/ AMIGOS

Ya que me han brindado el apoyo necesario siempre que lo necesite además de brindarme su cariño y confianza ante todo, y durante todo el desarrollo de mi carrera estando en la buenas y en la malas sin importar lo que pasara, por todo esto gracias por estar ahí siempre.

MARIA ELISA DUBON CASTRO

AGRADECIMIENTOS

A MI PADRE CELESTIAL, DIOS

Por darme la oportunidad de lograr una meta tan importante en mi vida, por darme las fuerzas para seguir adelante y no dejarme vencer nunca a pesar de las diferentes adversidades, por darme tan abundantes bendiciones, además de darme sabiduría para hacer lo que me propongo siempre.

A MI MADRE ENA ROSA TURCIOS

Por ser un ejemplo que inspiro en mí el incentivo para seguir adelante, seguir luchando y creer que siempre es posible superarse cada día más; gracias por su inmenso amor y apoyo desde todos los puntos de mi vida, como madre, amiga, y consejera.

A MIS COMPAÑERAS

Por brindarme su confianza en este proyecto. Por toda la ayuda y apoyo que me han brindado a lo largo de la carrera y ha permitido culminar este trabajo de grado. Siempre deseándoles a mis queridas amigas muchísimas bendiciones de parte de Dios y brindándoles siempre mi más sincera amistad.

A MIS FAMILIARES

Por creer siempre que era posible terminar y lograr este triunfo. Por brindarme su tan preciado soporte cuando lo necesitaba, por darme palabras de aliento cuando lo necesitaba.

A MIS AMIGOS Y AMIGAS

Que siempre tuvieron palabras de aliento y consejos en todos los momentos; a todos ellos y ellas les agradezco inmensamente.

ALICIA CAROLINA GUZMAN TURCIOS

AGRADECIMIENTOS

A DIOS TODOPODEROSO

Por cubrirme con su manto divino y guiarme en el largo camino que emprendí hace muchos años y nunca dejarme caer aun en los momentos mas difíciles de mi vida ; por regalarme la paciencia , sabiduría y perseverancia necesaria para obtener este triunfo.

A MI MADRECITA QUERIDA

Por darme su amor y apoyo incondicional en todo momento a lo largo de este proyecto y por tener siempre las palabras adecuadas en los momentos mas difíciles, por siempre luchar para darme las herramientas necesarias para alcanzar mi superación, sobre todo por ser mi amiga incondicional.

A MI PADRE

Que aunque lejos de mi, me apoyo y me guió a lo largo de mi carrera y por siempre confiar en mi y en mis sueños.

A MIS HERMANOS

Porque siempre han cuidado de mi y me han dado su apoyo incondicional en cada uno de mis proyectos, siendo un ejemplo a seguir en mi vida. Sin sus consejos y cariño no hubiera podido realizar este proyecto.

RINA ELIZABETH LOPEZ MENENDEZ

A MI NOVIO

Por ser mí amigo incondicional en los buenos y malos momentos, por ser el ángel en la tierra que me cuida y protege.

A MIS COMPAÑERAS Y AMIGAS

Por estar siempre ahí cuando mas las necesite, por ayudarme en las dificultades y por trabajar conmigo de forma conjunta no solo a lo largo de este proyecto sino también a lo largo de toda la carrera.

RINA ELIZABETH LOPEZ MENENDEZ

CAPITULO I.....	1
1.1 INTRODUCCIÓN	2
1.2 ANTECEDENTES.....	3
1.2.2 Misión Y Visión Del Departamento De Ingeniería Y Arquitectura	7
1.2.3 Estructura Organizativa Del Departamento De Ingeniería y Arquitectura	9
1.2.4 Servicios Que Presta El Departamento De Ingeniería y Arquitectura	12
1.2.4.1 Docencia.....	12
1.2.4.2 Cursos De Refuerzo A Las Clases Impartidas Por El Departamento De Ingeniería y Arquitectura.....	13
1.3.2 Objetivos Específicos:.....	15
1.4 PLANTEAMIENTO DEL PROBLEMA	16
1.7 RESULTADOS ESPERADOS.....	20
1.8 LIMITANTES.....	22
1.9 METODOLOGÍA DE LA INVESTIGACIÓN Y DESARROLLO	23
CAPITULO II	28
2.1 MICROPROCESADORES	29
2.1.1 ¿Qué es el Microprocesador?	29
2.1.2 Historia De Los Microprocesadores	30
2.1.2.1 Historia De Los Primeros Microprocesadores	30
2.2 TIPOS DE MICROPROCESADORES SEGÚN NÚMERO DE BITS	38
2.2.1 Microprocesadores De 8 Bits.....	38
2.2.1.1 Microprocesador 8085	39
2.2.2 Microprocesadores De 16 Bits.....	53
2.2.2.1 Historia Del 8086/8088.....	54
2.2.2.2 Arquitectura De Los Procesadores 8088 Y 8086.....	56
2.2.3 Microprocesadores De 32 Bits.....	57
2.2.4 Microprocesadores De 64 Bits.....	58
2.2.4.1 La Arquitectura DEC Alpha:	59
2.2.4.2 La Arquitectura IA-64.....	60
2.2.4.3 La Arquitectura AMD64.....	60
2.2.4.4 La Arquitectura SPARC.....	60

2.2.4.5 La Arquitectura POWER	61
2.2.4.6 La Arquitectura PA-RISC	62
2.2.4.7 Chipset Para Microprocesadores De 64 Bits.....	62
2.3 CARACTERÍSTICAS DE LOS MICROPROCESADORES	63
2.4 MICROPROCESADORES MÁS UTILIZADOS	64
2.5 SIMULADORES PARA MICROPROCESADORES	65
2.6 VENTAJAS Y DESVENTAJAS DE LOS SIMULADORES	67
2.7 COMPONENTES BÁSICOS DE LOS MICROPROCESADORES	69
2.7.1 Internos.....	69
2.7.1.1 Coprocesador Matemático	69
2.7.1.2 Memoria Caché	71
2.7.1.3 Encapsulado	73
2.7.1.4 Interrupciones.....	78
2.7.1.5 Instrucciones	78
2.7.1.5.1 ¿Qué Es Una Instrucción?.....	78
2.7.1.5.2 Formato de las Instrucciones.....	79
2.7.1.5.3 Tipos De Instrucciones.....	80
2.8 COMPONENTES ADICIONALES	81
2.8.1 Arquitectura Y Organización De Un Microprocesador	81
2.8.2 Puertos De Entrada/Salida	82
2.8.3 Buses	84
2.8.3.1 Tipos De Buses	84
2.8.4 Memorias	85
2.8.5 Placa Base	86
2.8.5.2 Tipos De Placa Base	87
2.8.5.4 ¿Qué Es Zócalo Del Microprocesador?	90
2.9 APLICACIONES DE LOS MICROPROCESADORES.....	92
2.10 ELECCIÓN DE UN MICROPROCESADOR	97
2.11 EMPRESAS FABRICANTES DE MICROPROCESADORES	99
2.12 PROGRAMACIÓN PARA MICROPROCESADORES	100
2.12.1 Sistemas De Numeración.....	100

2.12.1.1	¿Qué Es Un Sistema De Numeración?	101
2.12.1.2	Sistema Binario	102
2.12.1.3	Sistema Octal	103
2.12.1.4	Sistema Hexadecimal	104
2.12.1.5	Conversiones	106
2.12.1.6	Operaciones Aritméticas	108
2.12.2	Lenguaje Ensamblador	112
2.12.2.1	Importancia Del Lenguaje Ensamblador	112
2.12.2.2	Ventajas Del Lenguaje Ensamblador	113
2.12.2.3	Desventajas Del Lenguaje Ensamblador	113
2.12.2.4	Relación Del Lenguaje Ensamblador Con Los Componentes Internos Del Procesador	114
2.12.2.5	Tipos De Ensambladores	115
CAPITULO III		118
3.1	ANTECEDENTES DE LA SITUACIÓN ACTUAL	119
3.2	METODOLOGÍA DE LA INVESTIGACIÓN	120
3.2.1	Elementos De La Investigación	122
3.2.2	Fuentes De Información	122
3.2.2.2	Fuentes Secundarias	123
3.2.3	Investigación Documental	123
3.2.4	Investigación De Campo	123
3.2.5	Determinación Del Universo Y Tamaño De Muestra	124
3.3	INTERPRETACIÓN DE LOS DATOS DE LA ENCUESTA REALIZADA	126
3.4	VISITAS TÉCNICAS	128
3.5	CONCLUSIONES DE LA INVESTIGACIÓN	129
3.5.1	Laboratorio Práctico PRIMER TRAINER Para Microprocesador 8085	130
CAPITULO IV		132
4.1.1	Análisis Del Temario y Actividades	133
4.1.2	Plan de Guías de Práctica	135
4.2	PROPUESTA DE EQUIPAMIENTO PARA LOS LABORATORIOS	136
4.2.1	Programación en Lenguaje Ensamblador	136

4.2.2 Descripción del Equipo de Hardware Primer TRAINER 8085	137
4.2.3 Diagramas “Primer TRAINER 8085”	141
4.2.4 Simulador de Microprocesador Intel 8085 Versión 1.0	145
4.2.4.1 Requerimientos e Instalación del simulador	146
4.2.5 Programa Ensamblador	148
4.2.5.1 Pasos a seguir para convertir un archivo .ASM a .HEX a través del programa ensamblador.	149
4.2.6 Cargar un Archivo Hexadecimal al HYPERTERMINAL para Windows.....	152
4.3 MEDIDAS DE SEGURIDAD Y RECOMENDACIONES EN EL MANEJO DEL EQUIPO ESPECIALIZADO PARA EL DESARROLLO DE PRÁCTICAS DE LABORATORIO CON MICROPROCESADORES	160
4.3.1 Precauciones Generales.....	160
4.3.2 Condiciones Recomendadas del Lugar de Trabajo	161
4.3.3 Recomendaciones a los Auxiliares de Cátedra e Instructores.....	161
4.3.4 Recomendaciones a los Docentes	162
4.3.5 Recomendaciones a los Estudiantes.....	163
4.3.6 Del Funcionamiento de los Laboratorios	164
4.3.7 Reglas Básicas de Seguridad.....	164
4.3.8 Para Programar los Microprocesadores	165
4.3.9 La Electricidad Estática.....	165
4.3.9.1 Precauciones.....	165
4.3.10 Prevenir los Sigüientes Lugares para la Operación e Instalación de los Dispositivos Electrónicos.....	166
4.5 PRESUPUESTO DEL PROYECTO.	168
4.6 PROYECCIÓN DEL TIEMPO DE UTILIDAD DEL EQUIPO.....	169
4.7 BENEFICIOS.....	171
4.7.1 Beneficios Proporcionados a la Población Estudiantil.....	171
4.7.2 Beneficios Proporcionados al Departamento de Ingeniería y a la UES-FMO.....	172
4.8 MODELO DE PLÁN DE CAPACITACIÓN.....	174
4.8.1 Generalidades.....	174
4.8.2 Módulos que Integran el Plan de Capacitación.....	177

4.8.3. Material y Equipo Utilizado.....	177
4.8.4. Presupuesto de Capacitación.....	178
5.0 CONCLUSIONES	179
6.0 RECOMENDACIONES	181
7.0 BIBLIOGRAFÍA	183
9.0 ANEXOS	191

CAPITULO I

CAPITULO I

Generalidades

1.1 INTRODUCCIÓN

El microprocesador es uno de los logros más sobresaliente del siglo XX. Es por eso que se ha vuelto parte esencial de nuestras vidas, concibiéndose en el núcleo de una máquina tras otra. Su presencia ha comenzado a cambiar la forma en que se percibe el mundo e incluso a la sociedad misma. Cada vez se hace más difícil pasar por alto el microprocesador como otro simple producto en una larga línea de innovaciones tecnológicas.

Ninguna otra invención en la historia se ha esparcido de manera tan rápida por todo el mundo, existen casi 15,000 millones de microchips de alguna clase en uso; lo que equivale a dos computadoras poderosas para cada hombre, mujer y niño del planeta. En la actualidad, nadie puede dudar que el microprocesador no sólo esté transformando los productos que se utilizan, sino también la forma de vivir y la forma en que se percibe la realidad. No obstante hay que reconocer la introducción del microprocesador en la vida diaria, de tal manera que se está evolucionando de manera indiferente a la presencia de miles de máquinas diminutas que se encuentran a diario.

Por tanto, antes de que el uso del microprocesador pase de forma desapercibida se debe realizar un completo estudio y análisis sobre él, su funcionamiento y la revolución que ha originado, para apreciar de manera más clara cada una de las partes que lo componen.

En el presente trabajo de grado se presenta un estudio completo acerca de los microprocesadores específicamente el microprocesador 8085, su funcionamiento y partes que lo conforman, además de presentar el estudio, análisis y conclusiones de una propuesta completa para la implementación de un laboratorio didáctico para la programación con microprocesadores.

1.2 ANTECEDENTES

1.2.1 Marco Teórico.¹

La UNIVERSIDAD DE EL SALVADOR, fue fundada el 16 de febrero de 1841, permitiendo con ello la Enseñanza Superior Universitaria en nuestro país; conforme el paso del tiempo, la demanda de estudiantes fue haciéndose mayor, debido a esto surge la necesidad de ampliar las alternativas para aspirantes de todas las zonas del país; es así como se da origen a los Centros Universitarios Regionales de Occidente, Oriente y paracentral.

El Centro Universitario de Occidente fue fundado el 15 de octubre de 1874, pero por falta de apoyo desaparece en 1880, así transcurre el tiempo y después de incansables esfuerzos de la Sociedad de Abogados de Occidente y por iniciativa de los Doctores Ángel Góchez Marín, Gustavo Adolfo Noyola, Luís Ernesto Arévalo y otros, se llevó a cabo una serie de reuniones con las autoridades superiores de la Universidad de El Salvador, entre ellos los doctores Fabio Castillo Figueroa y Rafael Antonio Vásquez, Rector y Vicerrector, respectivamente, quienes junto a sectores del magisterio, sindicatos, señoras de los mercados, empleados, profesionales y pueblo en general, se llega a lograr que mediante el acuerdo N° 46 del Honorable Consejo Superior Universitario, se funda el 16 de julio de 1965 el Centro Universitario de Occidente.

Para esa época el centro ofrecía los estudios que integran las distintas carreras profesionales de la Universidad de El Salvador; así poseía los mismos departamentos de servicios generales existentes en la Unidad Central de San Salvador; como son: Física y Matemáticas, Ciencias Biológicas y Química; además el Centro contaba con un departamento de Ciencias Sociales, Filosofía y Letras.

¹ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

Este departamento estaba integrado por diversas secciones generales: Filosofía, Letras, Artes Ciencias de la Educación y Ciencias Sociales.²

El Centro Universitario de Occidente nació como una proyección de la Universidad de El Salvador, ante la necesidad de fomentar y difundir la enseñanza superior universitaria en la Zona Occidental del país. El terreno fue adquirido por la Facultad Multidisciplinaria de Occidente en el mes de febrero de 1966, comprando varias parcelas que forman un solo cuerpo de una superficie de 8 hectáreas, 95 áreas y 15.13 centiáreas equivalentes a 12 manzanas, 8,080 Vrs².

En relación a las instalaciones, se puede mencionar que entre los años de 1966-1976, contaba con lo siguiente: 7 salones de clases, 1 auditorio y 8 laboratorios; dichos espacios están distribuidos en un total de tres edificios. Las aulas se construyeron bajo el sistema de elementos prefabricados, con el propósito de resolver momentáneamente el problema de la demanda estudiantil, o sea que eran provisionales. Sin embargo, debido a la falta de recursos presupuestarios, las mismas adquirieron el carácter de permanentes.

En la década de los ochenta, se incrementa la infraestructura, con un edificio de tres plantas que tiene un total de 12 aulas, el cual es conocido como “Bunker”. Además en esta misma década, se construye un edificio de 2 niveles para el área de Ciencias Jurídicas.

En la década de los noventa, específicamente para la fecha del 4 de junio de 1992, en sesión ordinaria del Consejo Superior Universitario, se llegó al acuerdo N° 39-91-95-IX denominado “Proyecto de Acuerdo del Consejo Superior Universitario, sobre la Creación de las Facultades Regionales Multidisciplinarias”, en el que se establece la Facultad Multidisciplinaria Occidental con sede en el Departamento de

² Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

Santa Ana, se organizo y funciona actualmente en el terreno e instalaciones que ocupa el Centro Universitario de Occidente.

En el año 2002 gracias a la celebración de los Juegos Centroamericanos y del Caribe, la Universidad de El Salvador se ve beneficiada, y especialmente la Facultad Multidisciplinaria de Occidente, con la construcción de dos nuevos elementos físicos a su infraestructura, como son el Edificio de Usos Múltiples, y el de Medicina.³

En lo que respecta al Departamento de Ingeniería, inicialmente nace atendiendo las especialidades de Física, Matemática y las asignaturas propias de las diferentes carreras de Ingeniería en sus dos primeros años básicos, teniendo los alumnos en esa época que trasladarse a San Salvador para proseguir sus estudios en la Facultad de Ingeniería y Arquitectura. Lo anterior se daba prácticamente desde la fundación de esa casa de estudios como CUO.

Posteriormente a finales de la época de los 80 el departamento de Ingeniería se separa de las especialidades de Física y Matemática y comienza atender las asignaturas propias de sus carreras. En ese momento se permite que los docentes por afinidad o por inclinación profesional se asocien a alguno de los tres nuevos departamentos que se crearon. Ya en la década de los 90 se empieza a valorar la posibilidad de completar alguna de las carreras de Ingeniería que hasta esa fecha solo atendían los años comunes o sea los dos primeros años de todas las especialidades. De tal manera que se presento el proyecto de apertura del tercer año de ingeniería civil para lo cual se contó con la disposición y el apoyo de todo el cuerpo docente del departamento asumiendo una responsabilidad académica mayor a la que tenían obligación para poder de esa manera impartir las asignaturas que correspondían a ese año adicional, lo anterior en un primer momento funciono en cuanto a la operacionalización de la carrera, no así con el reconocimiento institucional hacia el

³ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

esfuerzo realizado por los académicos , ya que al finalizar ese tercer año y querer poner el cuarto año de ingeniería civil, se tuvo como respuesta oficial de que reasumiera mas compromiso del que ya se tenía hasta el momento, sin reconocimiento alguno y sin el ofrecimiento de nuevas contrataciones para solventar los compromisos que implicaba otro año de la carrera. Paralelamente a lo anterior se comenzó a generar una inquietud del gremio estudiantil y los padres de familia, al grado que para no abortar el entusiasmo y la posibilidad de completar la carrera de ingeniería civil en una asamblea de padres de familia se tomo la decisión de crear la sociedad de padres de familia de ingeniería de la FMO, con el compromiso de aportar una cuota social que serviría para solventar el pago de docentes necesarios para completar la carrera de Ingeniería Civil. Lo anterior permitió contratar personal externo en concepto de hora clase que vinieron a cubrir la necesidad antes mencionada.⁴

Cuando se estaba finalizando la carrera de ingeniería civil y luego de evaluar el éxito en la gestión económica que generaba la sociedad de padres se siguió con la carrera de ingeniería industrial y se hizo la gestión para iniciar la carrera de Ingeniería de Sistemas Informáticos.

En el mes de octubre del año 2004 el Consejo Superior Universitario aprobó la creación de la carrera de Arquitectura en este departamento, además se estuvo trabajando para presentar antes de finalizar el pasado año el proyecto del Diplomado en Informática Educativa.

De esta manera el nombre del departamento de ingeniería fue actualizado por lo que ahora es llamado Departamento de Ingeniería y Arquitectura, teniendo como oferta académica las carreras completas de Ingeniería Civil, Industrial, de Sistemas

⁴ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

Informáticos, la apertura de la carrera de Arquitectura y los dos primeros años de las carreras de Ingeniería Eléctrica, Mecánica y Química.

En el 2005, se continúa con el manejo de software libre y es organizada una feria que ya es de carácter latinoamericano correspondiente a dicho tema, y en la cual la FMO sirve como sede.

Dando continuidad a la política de mejoramiento y ampliación de espacios físicos, la Facultad Multidisciplinaria de Occidente cuenta, a partir de principios de junio de 2008 con un conjunto de aulas completamente nuevo.⁵

El nuevo módulo consta de 5 aulas nuevas completamente equipadas que fueron construidas desde el año 2008 hasta el año 2009, además de contar con un espacio reservado para la librería universitaria para proporcionar un mejor servicio a la comunidad estudiantil.

Hoy por hoy, la Facultad Multidisciplinaria de Occidente exhibe en su campus espacios físicos modernos, que le proporcionan un ambiente adecuado, cómodo y atractivo para la enseñanza, además de permitirle paulatinamente, extender su capacidad de atención a más alumnos y carreras dentro de la Comunidad Universitaria.

1.2.2 Misión Y Visión Del Departamento De Ingeniería Y Arquitectura

VISIÓN

Construir una Facultad líder en educación universitaria cuyas actividades académicas, científicas, tecnológicas y de proyección social tengan un real y significativo impacto en el desarrollo de una alta calidad educativa y,

⁵ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

consecuentemente, en las condiciones de vida de la población valores socialmente positivos como: la democracia, la solidaridad, la justicia, la honestidad, la tolerancia, y la investigación científica al servicio de la población.

MISIÓN

Ubicarnos en la perspectiva de formar profesionales con alta calidad académica, moral, científica y humana; capaz de incidir en la solución de los problemas de la región occidental y de El Salvador en general.

FINES⁶

- Conservar, fomentar y difundir la ciencia, el arte y la cultura.
- Formar profesionales capacitados moral e intelectualmente para desempeñar la función que les corresponde en la sociedad, integrando para ello las funciones de docencia, investigación y proyección social.
- Realizar investigación filosófica, científica, artística y tecnológica de carácter universal, principalmente sobre la realidad salvadoreña y centroamericana;
- Propender, con un sentido social-humanístico, a la formación integral del estudiante.
- Contribuir al fortalecimiento de la identidad nacional y al desarrollo de una cultura propia, al servicio de la paz y de la libertad.
- Promover la sustentabilidad y la protección de los recursos naturales y el medio ambiente.
- Fomentar entre sus educandos el ideal de unidad de los pueblos centroamericanos.
- Para la mejor realización de sus fines, la Universidad podrá establecer relaciones culturales y de cooperación con otras universidades e instituciones, sean éstas

⁶ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

públicas o privadas, nacionales o extranjeras, dentro del marco de la presente Ley y demás leyes de la República.

- Sin menoscabo de su autonomía, la Universidad prestara su colaboración Estado en el estudio de los problemas nacionales.

1.2.3 Estructura Organizativa Del Departamento De Ingeniería y Arquitectura

La estructura organizativa del departamento de ingeniería y arquitectura muestra su posición en el organigrama de la facultad multidisciplinaria de Occidente, como se detalla en la sección siguiente.

1.2.3.1 Posición Organizativa Del Departamento De Ingeniería y Arquitectura Dentro De La UES⁷

Para poder conocer más fácilmente en qué posición se encuentra el Departamento de Ingeniería y Arquitectura dentro de la Universidad se muestra el Organigrama de La Universidad de El Salvador (Ver Figura 1.1), el que muestra que el Departamento de Ingeniería y Arquitectura se encuentra ubicado en el tercer Nivel después del Decanato y Vicedecanato al igual que los demás departamentos y oficinas administrativas.

⁷ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

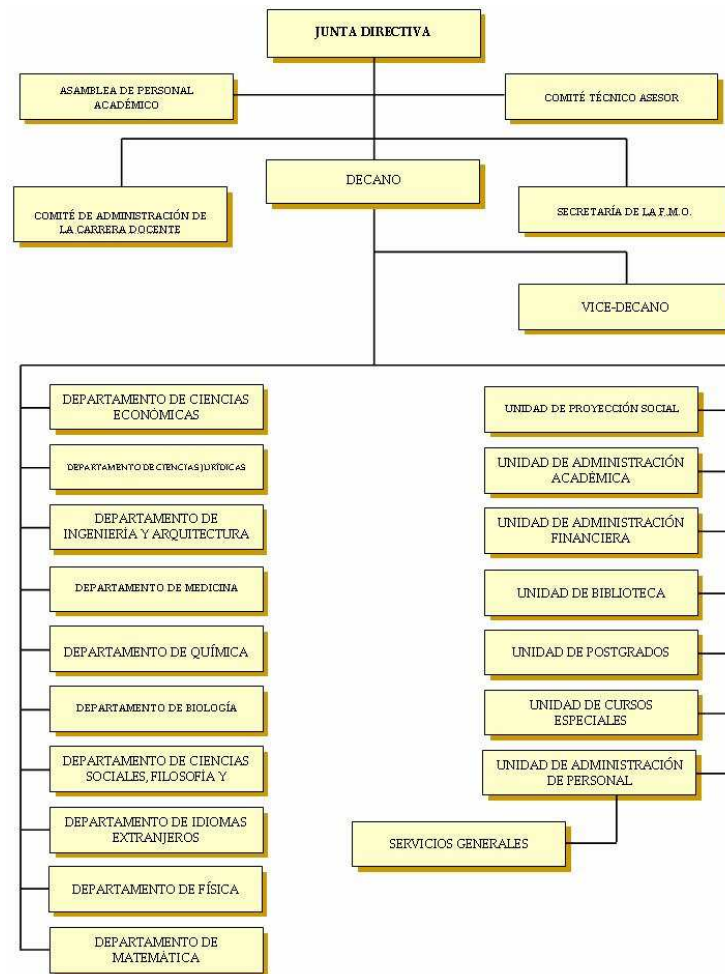


Figura 1.1: Organigrama de la Facultad Multidisciplinaria de Occidente

1.2.3.2 Organigrama Del Departamento De Ingeniería y Arquitectura⁸

Como en toda entidad u organización es necesario tener una estructura jerárquica bien definida para su buen funcionamiento o administración, es por ello que el Departamento de Ingeniería y Arquitectura ha definido su Organigrama tal como se en la siguiente figura:

⁸ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

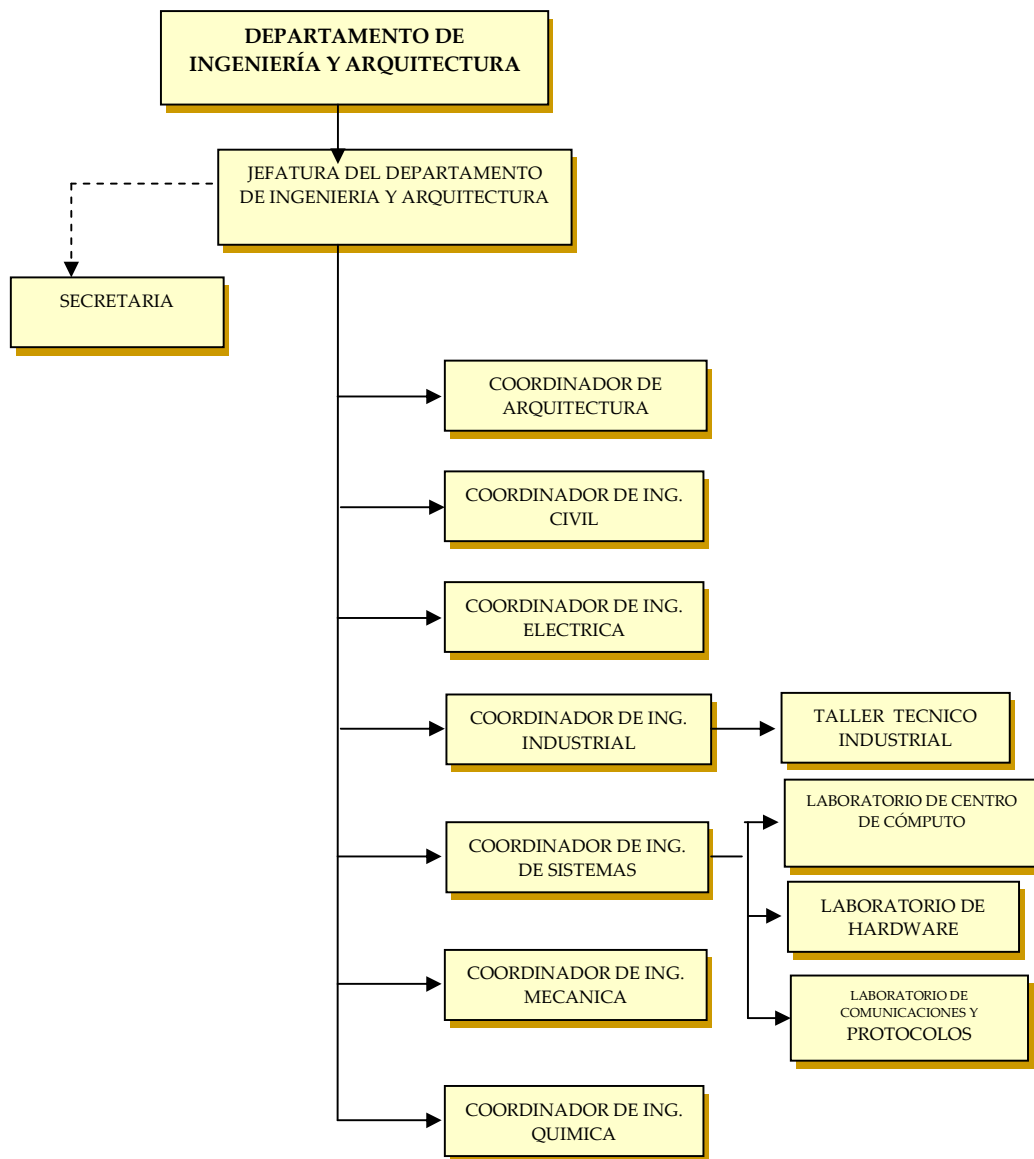


Fig. 1.2: Organigrama del Departamento de Ingeniería y Arquitectura.⁹

⁹ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

1.2.3.3 Ubicación Geográfica Del Departamento De Ingeniería y Arquitectura¹⁰

La Facultad Multidisciplinaria de Occidente, se encuentra ubicada aproximadamente a 4 kilómetros al sur del Centro de la ciudad de Santa Ana, siendo la vía de acceso principal la Avenida Fray Felipe de Jesús Moraga. En lo que respecta al Departamento de Ingeniería y Arquitectura se encuentra ubicado en el segundo nivel del edificio de usos múltiples, al costado norte del departamento de Idiomas.

La infraestructura está formada de varios cubículos en forma rectangular, en cada uno de los cuales se encuentran ubicados los docentes del departamento, cada uno cuenta con su propio cubículo para que pueda guardar el material que necesita para impartir las clases como lo son los libros, folletos, guías y los trabajos de investigación que los alumnos realizan, además cuando un alumno necesita hacerle una consulta es el lugar preciso donde puede buscarlo.

1.2.4 Servicios Que Presta El Departamento De Ingeniería y Arquitectura

El departamento de ingeniería y arquitectura presta una serie de servicios como lo son: docencia, laboratorios y algunos cursos de refuerzo de las materias impartidas, servicio de biblioteca y préstamo de equipo informático a los docentes del Departamento de Ingeniería y Arquitectura.

1.2.4.1 Docencia

Los docentes o catedráticos son una parte muy fundamental en el aprendizaje o desarrollo de todo alumno de nivel superior, por lo que es necesario que ellos estén muy bien preparados para poder lograr que los estudiantes pongan mucha atención y empeño en las clases, para así ampliar sus conocimientos y poner en práctica toda la

¹⁰ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

información que es transmitida por los profesionales los cuales con el pasar de los años se han enfrentado con diferentes problemas o experiencias las cuales ellos comparten con los alumnos para evitarles tropiezos o dificultades en el futuro y facilitarles su desenvolvimiento cuando sean profesionales.¹¹

El Departamento de Ingeniería y Arquitectura cuenta con un grupo de docentes profesionales entre ellos Ingenieros Industriales, Civiles, Ingenieros de Sistemas Informáticos, Arquitectos, Licenciados, Técnicos y 3 Ingenieros Mecánicos. Además cuenta con la ayuda de auxiliares de cátedra y con estudiantes que realizan su servicio social para apoyarlos en los diferentes laboratorios o practicas de cada materia para darles la oportunidad de adquirir experiencia en el área de la docencia. Los estudiantes proporcionan también su servicio social a instituciones que lo solicitan al encargado correspondiente del departamento de Ingeniería y Arquitectura.

1.2.4.2 Cursos De Refuerzo A Las Clases Impartidas Por El Departamento De Ingeniería y Arquitectura

Con respecto a su metodología de crear profesionales existe un centro de cómputo con un aproximado de 20 computadoras conectadas en red y con acceso a Internet, en el cual se reciben las prácticas de determinadas materias que lo requieren para que los alumnos pongan en práctica los conocimientos adquiridos en clases, de esta forma los alumnos comprenderán de manera más fácil y adecuada la teoría vista en clases, formando grupos de práctica, en los cuales cada alumno pueda practicar en una maquina en los horarios asignados por el encargado de dicho centro de computo.

Además se imparten otra clase de cursos para las diversas carreras como refuerzo para el buen desarrollo de los estudiantes del departamento, entre los cuales se

¹¹ Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

pueden mencionar: cursos de programación, redes y montajes de servidores así como también de Linux.

Por otra parte, si el alumno necesita o desea especializarse en determinada área de su carrera, es conveniente que por sus propios métodos o de acuerdo a sus esfuerzo pueda capacitarse ya sea asistiendo a cursos que imparten diferentes centros fuera de la Universidad o que por medio de libros o artículos que se encuentran en Internet obtenga los conocimientos que necesita y que los programas de estudio de su carrera no puedan cubrir, de tal forma no tendrá deficiencias en sus conocimientos a la hora de enfrentarse en la sociedad como profesional. 12

¹² Francia, García, Umaña, "Diseño de guías de trabajo y construcción de equipo didáctico para la implantación de prácticas de laboratorio con Micro Controladores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente" (Tesis de Ingeniería de Sistemas Informáticos, 2007)

1.3 OBJETIVOS

1.3.1 Objetivo General:

Desarrollar e Implementar un laboratorio didáctico para la programación con microprocesadores en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente.

1.3.2 Objetivos Específicos:

1. Realizar una investigación detallada de toda la información acerca de la historia y funcionamiento de los Microprocesadores que pueden ser utilizados en las prácticas de laboratorio.
2. Analizar profundamente toda la información acerca de los microprocesadores recabada durante la investigación de los mismos.
3. Adquirir el equipo didáctico y de Software necesarios para el correcto manejo del material didáctico a utilizar en las prácticas con microprocesadores con el fin de incrementar la asimilación del estudio de los mismos.
4. Proporcionar el equipo didáctico y de software para el exitoso desarrollo de las prácticas de laboratorio.
5. Elaborar Guías de Trabajo Practico para la utilización del equipo didáctico que se proveerá para realizar prácticas relacionadas con microprogramación.
6. Mejorar el aprendizaje de los alumnos de la carrera de ingeniería en sistemas, al proporcionar el equipo adecuado para las prácticas con microprocesadores, brindando una excelente plataforma educativa para los estudiantes.
7. Brindar un apoyo didáctico a los docentes que tengan la necesidad de impartir las clases tanto teóricas como prácticas con microprogramadores.
8. Desarrollar un programa de capacitaciones para facilitar y promover el uso adecuado del equipo y de todos los recursos didácticos obtenidos a largo del desarrollo del proyecto.

1.4 PLANTEAMIENTO DEL PROBLEMA

A medida que el desarrollo de la tecnología se ha incrementado, ha surgido la necesidad de profesionales altamente capacitados para el estudio e implementación de la misma: y así ser más competitivos en el mercado laboral. Por esto nace la necesidad de incrementar los conocimientos prácticos de los futuros profesionales en esta área.

Comprender el término calidad en la educación superior universitaria requiere una explicación de las diferentes dimensiones y los ejes fundamentales, desde donde se puede reconocer la calidad de un sistema educativo, por ello es importante que se brinden todos los medios posibles para el satisfactorio aprendizaje de los educandos, actualmente el Departamento de Ingeniería y Arquitectura de la Facultad Multidisciplinaria de Occidente no cuenta con los medios necesarios para impartir conocimientos prácticos de alta calidad basados en el estudio de los microprocesadores puesto que no se cuenta con las herramientas y equipo necesarios para desarrollar prácticas de laboratorio en la asignatura de microprogramación.

Con lo descrito anteriormente se observan los siguientes problemas:

1. El Departamento de Ingeniería y Arquitectura no cuenta con los medios y equipos para poder realizar los laboratorios prácticos acerca del uso de Microprocesadores y sus aplicaciones que apoyen los conceptos teóricos de la asignatura de Microprogramación de la carrera Ingeniería de Sistemas Informáticos.
2. Actualmente no se cuenta con un laboratorista o especialista que cubra el área del laboratorio de hardware en la materia de microprogramación, con lo cual se limita a la elaboración de guías de prácticas para la asignatura antes mencionada con las que se complementaban algunos de los conocimientos básicos teóricos expuestos en clase, en contenidos referentes al hardware, además de que no existen los recursos didácticos y las guías de trabajo para llevar a cabo los laboratorios

prácticos con Microprocesadores y sus aplicaciones en el programa de estudio de la asignatura.

3. La aportación de la asignatura al perfil de los egresados de la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente, se ve limitada en su participación para el desarrollo de las capacidades y competencias en el perfil de un Ingeniero informático, ocasionando que los profesionales se desempeñen en el ámbito laboral con ciertas deficiencias en el área de microprogramación; ya que no existen, cursos prácticos con Microprocesadores dentro de la facultad multidisciplinaria de occidente, dando como resultado un déficit en esa área.

Debido a los problemas antes mencionados, se presenta una propuesta que permita al Departamento de Ingeniería y Arquitectura implantar un Laboratorio didáctico para el trabajo con microprocesadores que logre cubrir las necesidades de clases prácticas en la asignatura de microprogramación de la especialidad de dicha carrera.

1.5 JUSTIFICACIÓN

En la asignatura de Microprogramación no existen el equipo adecuado para prácticas en el laboratorio de hardware orientadas a la aplicación y manipulación de Microprocesadores, lo que conlleva a una enorme deficiencia en la asimilación de conocimientos de las distintas aplicaciones en el mundo real de este tipo de dispositivo electrónico; ya que dicho tema se aborda únicamente de forma teórica y simulaciones basadas en software, dejando de lado la práctica, la cual es de suma importancia en el mundo real.

En la Facultad Multidisciplinaria de Occidente se puede encontrar un alto déficit en el área de programación con microprocesadores en comparación con otras instituciones educativas, las cuales si poseen el equipo y laboratorio especializados en el área de electrónica y microprocesadores; tal como lo es la Universidad Don Bosco que se encuentra ubicada en la ciudadela Don Bosco, la cual si cuenta con dichos laboratorios especializados, así como también otras Instituciones educativas.

Es importante que el alumno tenga una visión amplia y experiencia en el trabajo practico con Microprocesadores, ya que es común encontrarse con dichos dispositivos en todo sistema automatizado en la industria, que en un momento determinado a lo largo de la carrera pueda necesitar un análisis técnico y resolución de problemáticas.

1.6 ALCANCES

- Se realizará una profunda Investigación sobre los Microprocesadores y los equipos de laboratorio para realizar prácticas con Microprocesadores, además de establecer modelos que puedan facilitar la correcta asimilación de conocimientos de parte de los estudiantes de una manera didáctica.
- Se analizará toda la información recabada que se obtenga a partir de la investigación de los microprocesadores así como también los modelos prácticos a realizar.
- Se desarrollará un plan de estudio sobre los microprocesadores, guías prácticas, manuales y material didáctico para los estudiantes, instructores y personal docente.
- Se elaborarán manuales para el uso correcto del equipo de electrónica proporcionado para la implementación del laboratorio didáctico
- Se desarrollará un programa de capacitaciones para facilitar y promover el uso adecuado del equipo y de todos los recursos didácticos obtenidos a largo del desarrollo del proyecto.
- Se adquirirá equipo didáctico para realizar las prácticas con microprocesadores para el desarrollo de guías de trabajo.
- Se entregará equipo didáctico para el desarrollo de los laboratorios prácticos para la programación con microprocesadores.
- Se mejorará el aprendizaje de los alumnos de la carrera de Ingeniería en Sistemas con la obtención del equipo didáctico y software.

1.7 RESULTADOS ESPERADOS

Los resultados esperados al finalizar el proyecto de “Investigación, Análisis, Diseño y Elaboración de un Laboratorio Didáctico para la Programación de Microprocesadores en la Carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente”, para las diferentes entidades de la Facultad son:

a) Departamento de Ingeniería y Arquitectura:

- Diseño completo de toda la documentación necesaria para la implantación de un laboratorio para el aprendizaje práctico del uso de Microprocesadores y sus aplicaciones, específicamente un conjunto de guías, manuales de uso y demás material bibliográfico para la utilización del equipo didáctico para desarrollar prácticas de los laboratorios acerca de los Microprocesadores, normas utilizadas dentro del laboratorio, sus precauciones y procedimientos para utilizar de una forma adecuada el equipo didáctico.
- Diseño y obtención de equipo didáctico(Dispositivos para programar microprocesadores) para la implementación del laboratorio práctico para microprocesadores; que contribuya al proceso de enseñanza- aprendizaje a los estudiantes de la materia de microprogramación

b) Jefatura del Departamento de Ingeniería y Arquitectura:

- Toda la información necesaria para la implementación del laboratorio didáctico con microprocesadores para el aprendizaje práctico del uso de los Microprocesadores y sus aplicaciones.

c) Estudiantes del Departamento de Ingeniería y Arquitectura.

- La utilización de los recursos (Entornos de programación, Simuladores) como herramientas fuertemente motivadoras para el aprendizaje de la programación con Microprocesadores y que apoye eficientemente el proceso de Enseñanza - Aprendizaje de los estudiantes de la carrera de Ingeniería de Sistemas Informáticos en la Facultad Multidisciplinaria de Occidente para una mayor comprensión de los temas en la asignatura de microprogramación.

d) Autoridades de la Facultad Multidisciplinaria de Occidente.

- Toda la información necesaria para la implementación del laboratorio didáctico con microprocesadores para el aprendizaje práctico del uso de estos y sus aplicaciones. Que sirva de apoyo al Proceso de Enseñanza - Aprendizaje de los estudiantes del Departamento de Ingeniería y Arquitectura, que colabore con la Misión y la Visión de la Universidad de El Salvador y poder así poder preparar mejores profesionales a la sociedad. Así como también para que dichas autoridades brinden un soporte económico para garantizar el funcionamiento, conservación y actualización del laboratorio didáctico.

1.8 LIMITANTES

1. Existe un número reducido de personal capacitado en la Universidad de El Salvador de la Facultad Multidisciplinaria de Occidente para el manejo de equipos especializados para la realización de prácticas con microprocesadores.
2. Dificultad de adquisición del equipo de practicas para microprocesadores debido a la falta de empresas proveedoras dentro del país.
3. Recursos económicos limitados para la adquisición de múltiples equipos.

1.9 METODOLOGÍA DE LA INVESTIGACIÓN Y DESARROLLO

El trabajo investigativo debe hacerse de manera ordenada, clara y comprensiva, con el propósito de tratar con sencillez, y precisión todos los aspectos que afecten en forma directa las ideas y conocimientos adquiridos en el transcurso de la investigación.

Para la recopilación de la información se hará uso dos tipos de fuentes de información muy importantes, tales como:

- **Básicas:**

Consiste en la investigación de campo por medio de encuestas y entrevistas que se les realizarán a un universo de personas que tengan conocimientos en dicha área.

- **Complementarias:**

Es aquella cuya información escrita está relacionada con el Trabajo de Grado, a manera de investigación bibliográfica.

Técnicas investigativas

- Encuestas: Consta de una técnica de recopilación de datos de manera impresa, ya sea por medio de preguntas puntuales o abiertas en un formulario.
- Entrevista: Otro medio para la obtención de información de manera verbal.
- Consultas bibliográficas: Por medio de acceso a Internet, consulta de libros, tesis, y otra información escrita que se pueda llegar a obtener.

Para la recopilación de información se utilizaran las 3 técnicas de investigación mencionadas anteriormente, para comprender y analizar de la mejor forma posible la

situación actual en la que se encuentra la asignatura relacionada al área de programación de microprocesadores, en el departamento de Ingeniería y Arquitectura de la Facultad Multidisciplinaria de Occidente, así también como a un universo de personas egresadas de la carrera de Ingeniería en Sistemas Informáticos que actualmente se encuentran laborando en la industria relacionado con microprocesadores, a otras instituciones en las cuales se realizan prácticas con la programación de microprocesadores.

Metodología para la exposición del trabajo de graduación

La investigación de trabajo de grado se ha dividido en cuatro capítulos cuyo contenido se presentara en tres avances, en los cuales se utilizará la siguiente metodología:

- Solicitud de equipo necesario para la presentación de avances, una semana antes del día programado para la realización de la misma.
- Solicitud de Sala de Defensa de Tesis para la presentación.
- Se hará uso del equipo de proyección audiovisual del departamento de ingeniería y arquitectura de la Facultad Multidisciplinaria de Occidente.

Planificación De Los Recursos A Utilizar

Es de mucha importancia planificar los recursos que serán utilizados y se consideran indispensables para el desarrollo del trabajo de grado, es por esto que se darán a conocer a continuación:

A. RECURSOS HUMANOS.

Se necesitarán durante todo el periodo de ejecución del estudio y los cuales se clasifican de la siguiente forma:

- Equipo de trabajo:

Integrado por tres alumnos egresados de la carrera de Ingeniería de Sistemas Informáticos, quienes estarán a cargo del desarrollo de la investigación, estudio, análisis de guías de práctica y equipo didáctico para la implementación de material didáctico para microprocesadores de la asignatura de Microprogramación.

- Coordinador General de Procesos de Graduación:

Es el responsable la supervisión del proceso de graduación y coordinación permanente de la atención a los estudiantes.

- Docente Director:

Es la persona destinada por la Comisión Evaluadora del Departamento de Ingeniería y arquitectura, quien tendrá la atribución de asesorar, guiar y evaluar el proceso de graduación en todos sus componentes.

- Docente Director Adjunto:

Es el docente elegido por la Comisión Evaluadora quien asistirá a las evaluaciones parciales y final con el propósito de garantizar que el proceso se realice de forma satisfactoria, transparente e imparcial.

B. RECURSOS MATERIALES.

Para el desarrollo del Trabajo de Grado se requerirá la disponibilidad de los equipos clasificados según su uso:

B.1 Documentación de Trabajo de Grado:

- Computadora.

- Impresor
- Ayudas audiovisuales (cañón, pantalla).
- Papelería.
- Un local para las presentaciones parciales y final.

C. RECURSOS FINANCIEROS.

El proceso de elaboración de Trabajo de Grado involucra recursos económicos, los cuales serán proporcionados por las estudiantes del equipo de trabajo.

Entre estos se pueden mencionar:

- Investigación de Campo.
- Elaboración de Documento.
- Consultas en Internet.
- Viáticos.
- Equipo didáctico para prácticas.
- Presentaciones parciales y final.

D. RECURSOS INSTITUCIONALES.

Las instituciones gubernamentales y no gubernamentales facilitarán información bibliográfica, de campo, entrevistas que aportaran en la investigación a realizarse:

- Universidad de El Salvador Facultad Multidisciplinaria de Occidente.
- Instituciones educativas que cuenten con prácticas de laboratorio para Microprocesadores y nos permitan el acceso.

Se obtendrá información a partir de:

- Biblioteca de Facultad Multidisciplinaria de Occidente.
- Biblioteca del Departamento de Ingeniería y Arquitectura.
- Internet.

CAPITULO II

CAPITULO II

Marco Teórico

2.1 MICROPROCESADORES

2.1.1 ¿Qué es el Microprocesador?

El microprocesador es la parte de la computadora diseñada para llevar a cabo o ejecutar los programas. Este viene siendo el cerebro de la computadora, el motor, el corazón de esta máquina. Este ejecuta instrucciones que se le dan a la computadora a muy bajo nivel haciendo operaciones lógicas simples, como sumar, restar, multiplicar y dividir. Es un chip, un tipo de componente electrónico en cuyo interior existen miles (o millones) de elementos llamados transistores, cuya combinación permite realizar el trabajo que tenga encomendado el chip.

Pero un procesador no actúa por propia iniciativa, recibe constantemente órdenes de múltiples procedencias. Cuando se enciende una computadora, lo primero que hace el micro es cumplir con las instrucciones de la BIOS (Basic input/output system), que forma parte de la memoria de la computadora. Una vez funcionando, además de la BIOS, será el sistema operativo y los programas instalados los que seguirán haciéndose obedecer por el microprocesador.

Pese a que los microprocesadores siempre hacen pensar en ordenadores, lo cierto es que están disponibles en multitud de dispositivos electrónicos que rodean a la sociedad habitualmente, como cámaras de fotografía o vídeo, coches, teléfonos móviles. No obstante, es cierto que aquellos que se emplean en las computadoras son los más potentes y complejos.¹³

¹³ Flores Q., Josefina. "Los microprocesadores" 27 de Octubre, 2002. www.monografias.com (13 de Abril, 2009)

2.1.2 Historia De Los Microprocesadores

2.1.2.1 Historia De Los Primeros Microprocesadores¹⁴

Han pasado más de 25 años desde que Intel diseñara el primer microprocesador, siendo la compañía pionera en el campo de la fabricación de estos productos, y que actualmente cuenta con más del 90 por ciento del mercado. Un tiempo en el que todo ha cambiado enormemente, y en el que desde aquel 4004 hasta el actual se ha visto pasar varias generaciones de máquinas que han entretenido y han ayudado en el trabajo diario.

Dicen que es natural en el ser humano querer investigar constantemente hacia el futuro, buscando información de hacia dónde se encamina, en lugar de en dónde se ha estado. Es por ello, el asombro ante las previsiones que los científicos predicen para dentro de unos quince años. Según el Dr. Albert Yu, vicepresidente de Intel y responsable del desarrollo de los procesadores desde el año 1984, para el año 2011 se utilizara procesadores cuyo reloj irá a una velocidad de 10 GHz (10.000 MHz), contendrán mil millones de transistores y será capaz de procesar cerca de 100 mil millones de instrucciones por segundo. Un futuro prometedor, que permitirá realizar tareas nunca antes pensadas.

Los Inicios

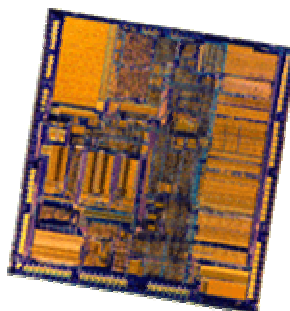


Fig. 2.1 – Microprocesador 8008

¹⁴ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

Sin embargo, para que esto llegue, la historia de los procesadores ha pasado por diferentes situaciones, siguiendo la lógica de la evolución de este mundo. Desde aquel primer procesador 4004 del año 1971 hasta el actual, se ha evolucionado mucho en el campo de los microprocesadores. Tanto, que no se puede estar seguro si las cifras que se predicen en Intel se pueden, incluso, quedar cortas. Aquel primer procesador 4004, presentado en el mercado el día 15 de noviembre de 1971, poseía unas características únicas para su tiempo. Para empezar, la velocidad de reloj sobrepasaba por poco los 100 KHz, disponía de un ancho de bus de 4 bits y podía manejar un máximo de 640 bytes de memoria. Realmente una auténtica joya que para entonces podía realizar gran cantidad de tareas, pero que por desgracia no tiene punto de comparación con los actuales micros. Entre sus aplicaciones, se puede destacar su presencia en la calculadora Busicom, así como dotar de los primeros tintes de inteligencia a objetos inanimados.¹⁵

Poco tiempo después, sin embargo, el 1 de abril de 1972, Intel anunciaba una versión mejorada de su procesador. Se trataba del 8008, que contaba como principal novedad con un bus de 8 bits, y la memoria direccionable se ampliaba a los 16 Kb. Además, llegaba a la cifra de los 3500 transistores, casi el doble que su predecesor, y se le puede considerar como el antecedente del procesador que serviría de corazón al primer ordenador personal. Justo dos años después, Intel anunciaba ese tan esperado primer ordenador personal, de nombre Altair, cuyo nombre proviene de un destino de la nave Enterprise en uno de los capítulos de la popular serie de televisión Star Trek la semana en la que se creó el ordenador. Este ordenador tenía un costo de entorno a los 400 dólares de la época, y el procesador suponía multiplicar por 10 el rendimiento del anterior, gracias a sus 2 MHz de velocidad (por primera vez se utiliza esta medida), con una memoria de 64 Kb. En unos meses, logró vender decenas de miles

¹⁵ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

de unidades, en lo que suponía la aparición del primer ordenador que la gente podía comprar, y no ya simplemente utilizar.¹⁶

La Introducción De IBM

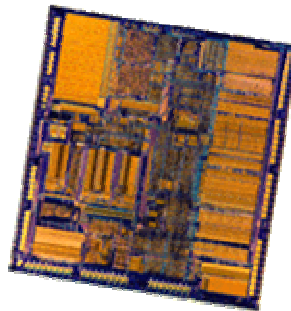


Fig. 2.2 – Microprocesadores IBM

Sin embargo, como se sabe, el ordenador personal no pasó a ser tal hasta la aparición de IBM, el gigante azul, en el mercado. Algo que sucedió en dos ocasiones en los meses de junio de 1978 y de 1979. Fechas en las que respectivamente, hacían su aparición los microprocesadores 8086 y 8088, que pasaron a formar el denominado IBM PC, que vendió millones de unidades de ordenadores de sobremesa a lo largo y ancho del mundo. El éxito fue tal, que Intel fue nombrada por la revista "Fortune" como uno de los mejores negocios de los años setenta. De los dos procesadores, el más potente era el 8086, con un bus de 16 bits (por fin), velocidades de reloj de 5, 8 y 10 MHz, 29000 transistores usando la tecnología de 3 micras y hasta un máximo de 1 Mega de memoria direccionable. El rendimiento se había vuelto a multiplicar por 10 con respecto a su antecesor, lo que suponía un auténtico avance en lo que al mundo de la informática se refiere. En cuanto al procesador 8088, era exactamente igual a éste, salvo la diferencia de que poseía un bus de 8 bits en lugar de uno de 16, siendo más barato y obteniendo mejor respaldo en el mercado.

¹⁶ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

En el año 1982, concretamente el 1 de febrero, Intel daba un nuevo vuelco a la industria con la aparición de los primeros 80286. Como principal novedad, cabe destacar el hecho de que por fin se podía utilizar la denominada memoria virtual, que en el caso del 286 podía llegar hasta 1 Giga. También hay que contar con el hecho de que el tiempo pasado había permitido a los ingenieros de Intel investigar más a fondo en este campo, movidos sin duda por el gran éxito de ventas de los anteriores micros. Ello se tradujo en un bus de 16 bits, 134000 transistores usando una tecnología de 1.5 micras, un máximo de memoria direccionable de 16 Megas y unas velocidades de reloj de 8, 10 y 12 MHz. En términos de rendimiento, se podía decir que se había multiplicado entre tres y seis veces la capacidad del 8086, y suponía el primer ordenador que no fabricaba IBM en exclusiva, sino que otras muchas compañías, alentadas por los éxitos del pasado, se decidieron a crear sus propias máquinas. Como dato curioso, basta mencionar el hecho de que en torno a los seis años que se le concede de vida útil, hay una estimación que apunta a que se colocaron en torno a los 15 millones de ordenadores en todo el mundo.¹⁷

Microsoft También Juega

El año de 1985 es clave en la historia de los procesadores. El 17 de octubre Intel anunciaba la aparición del procesador 80386DX, el primero en poseer una arquitectura de 32 bits, lo que suponía una velocidad a la hora de procesar las instrucciones realmente importante con respecto a su antecesor. Dicho procesador contenía en su interior en torno a los 275000 transistores, más de 100 veces los que tenía el primer 4004 después de tan sólo 14 años. El reloj llegaba ya hasta un máximo de 33 MHz, y era capaz de direccionar 4 Gigas de memoria. En 1988, Intel desarrollaba un poco tarde un sistema sencillo de actualizar los antiguos 286 gracias a la aparición del 80386SX, que sacrificaba el bus de datos para dejarlo en uno de 16 bits, pero a menor costo. Estos procesadores irrumpieron con la explosión del entorno gráfico Windows, desarrollado por Microsoft unos años antes, pero que no había

¹⁷ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

tenido la suficiente aceptación por parte de los usuarios. También existían algunos entornos que funcionaban incorrectamente, como por ejemplo el Gem 3, pero no es hasta este momento cuando este tipo de entornos de trabajo se popularizan, facilitando la tarea de enfrentarse a un ordenador, que por aquel entonces sólo se conocían unos pocos. Windows vino a ser un soplo de aire fresco para la industria, pues permitió que personas de cualquier condición pudiera manejar un ordenador con unos requerimientos mínimos de informática.

Y si esto parecía la revolución, no se tuvo que esperar mucho para que el 10 de abril de 1989 apareciera el Intel 80486DX, de nuevo con tecnología de 32 bits y como novedades principales, la incorporación del caché de nivel 1 (L1) en el propio chip, lo que aceleraba enormemente la transferencia de datos de este caché al procesador, así como la aparición del co-procesador matemático, también integrado en el procesador, dejando por tanto de ser una opción como lo era en los anteriores 80386. Dos cambios que unido al hecho de que por primera vez se sobrepasaban el millón de transistores usando la tecnología de una micra (aunque en la versión de este procesador que iba a 50 MHz usó ya la tecnología 0.8 micras), hacía posible la aparición de programas de calidad sorprendente, entre los que los juegos ocupan un lugar destacado. Se había pasado de unos ordenadores en los que prácticamente cualquier tarea compleja requería del intérprete de comandos de MS-DOS para poder ser realizada, a otros en los que con mover el cursor y pinchar en la opción deseada simplificaba en buena medida las tareas más comunes. Por su parte, Intel volvió a realizar, una versión de este procesador dos años después. Se trataba del 80486SX, idéntico a su hermano mayor salvo que no disponía del famoso co-procesador matemático incorporado, lo que suponía una reducción del costo para aquellas personas que desearan introducirse en el segmento sin necesidad de pagar una suma elevada.¹⁸

¹⁸ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

Pentium

Sin embargo, Intel no se quedó contemplando la gran obra que había creado, y rápidamente anunció que en breve estaría en la calle una nueva gama de procesadores que multiplicaría de forma general por cinco los rendimientos medios de los 80486. Se trataba de los Pentium, conocidos por P5 en el pequeño mundo de la informática mientras se estaban desarrollando, y de los que la prensa de medio mundo auguraba un gran futuro, tal y como así ha sido.

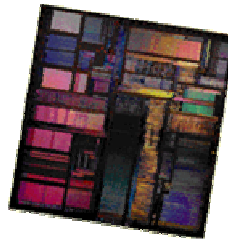


Fig. 2.3 – Microprocesadores Pentium¹⁹

Estos procesadores pasarán a la historia por ser los primeros a los que Intel no los bautizó con un número, y sí con una palabra. Esto era debido a que otras compañías dedicadas a la producción de procesadores estaban utilizando los mismos nombres puesto que no se podía registrar una cadena de ellos como marca, y por lo tanto, eran de dominio público. De modo que a Intel no le quedó más remedio que ponerle una palabra a su familia de procesadores, que además, con el paso del tiempo, se popularizó en los Estados Unidos de tal forma, que era identificada con velocidad y potencia en numerosos cómics y programas de televisión. Estos procesadores que partían de una velocidad inicial de 60 MHz, han llegado hasta los 200 MHz, algo que nadie había sido capaz de augurar unos años antes.

Con una arquitectura real de 32 bits, se usaba de nuevo la tecnología de 0.8 micras, con lo que se lograba realizar más unidades en menos espacio. Los resultados

¹⁹ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

no se hicieron esperar, y las compañías empezaron aunque de forma tímida a lanzar programas y juegos exclusivamente para el Pentium, hasta el punto que en ese momento quien no poseía un procesador de este tipo, estaba seriamente atrasado y no podía trabajar con garantías con los programas que habían en el mercado. Algo que ha venido a demostrar la aparición del nuevo sistema operativo de Microsoft Windows 95, que aunque funciona en equipos dotados de un procesador 486, lo hace sin sacar el máximo partido de sus funciones.

Pentium Pro Y Pentium II²⁰

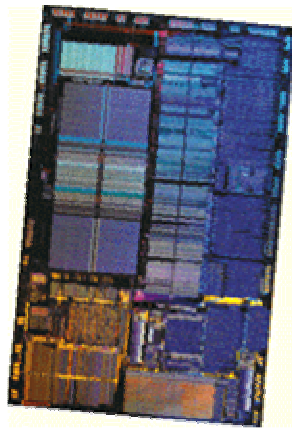


Fig. 2.4 – Microprocesadores Pentium Pro y Pentium II

La aparición, el 27 de marzo de 1995, del procesador Pentium Pro supuso para los servidores de red y las estaciones de trabajo un aire nuevo, tal y como ocurriría con el Pentium en el ámbito doméstico. La potencia de este nuevo procesador no tenía comparación hasta entonces, gracias a la arquitectura de 64 bits y el empleo de una tecnología revolucionaria como es la de 0.32 micras, lo que permitía la inclusión de cinco millones y medio de transistores en su interior. El procesador contaba con un segundo chip en el mismo encapsulado, que se encargaba de mejorar la velocidad de

²⁰ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

la memoria caché, lo que resultaba en un incremento del rendimiento sustancioso. Las frecuencias de reloj se mantenían como límite por arriba en 200 MHz, partiendo de un mínimo de 150 MHz. Un procesador que en principio no tiene muchos aspectos de saltar al mercado doméstico, puesto que los procesadores Pentium MMX parecen cubrir de momento todas las necesidades en este campo. ²¹

Una evolución que demostró Intel con un nuevo procesador, denominado Pentium II, que venía a ser simplemente un nuevo ingenio que suma las tecnologías del Pentium Pro con el MMX. Como resultado, el Pentium II era el procesador más rápido de cuantos había comercializado Intel. Únicamente se disponía de las versiones a 233 y 266 MHz, que supuso un nuevo récord de velocidad de reloj.

Procesador Pentium III²²

El Pentium III es un microprocesador de arquitectura i686, el cual es una modificación del Pentium Pro. Fue lanzado el 26 de febrero de 1999.

Procesador Pentium 4

El Pentium 4 (erróneamente escrito Pentium IV) es un microprocesador de séptima generación basado en la arquitectura x86. Es el primer microprocesador con un diseño completamente nuevo desde el Pentium Pro de 1995. El Pentium 4 original, denominado Willamette, trabajaba a 1,4 y 1,5 GHz; y fue lanzado en noviembre de 2000.

Procesador Pentium M

Introducido en marzo de 2003, el Intel Pentium M es un microprocesador con arquitectura x86 (i686) diseñado y fabricado por Intel. El procesador fue originalmente diseñado para su uso en computadoras portátiles. Su nombre en clave

²¹ Roper, Guillermo. "Microprocesadores" Noviembre, 2006. www.duiops.net (14 de Abril, 2009)

²² "Historia de los Microprocesadores" 19 de Mayo de 2009. <http://blog.puceing.edu.ec>

antes de su introducción era “Banyas”. Todos los nombres clave del Pentium M son lugares de Israel, la ubicación del equipo de diseño del Pentium M.

Procesador Pentium D

Pentium D fueron introducidos por Intel en 2005. Los chips Pentium D consisten básicamente en dos procesadores Pentium 4 (de núcleo Prescott) ubicados en una única pieza de silicio con un proceso de fabricación de 90 nm. El nombre en clave del Pentium D antes de su lanzamiento era “Smithfield”. Incluye una tecnología DRM (Digital rights management) para hacer posible un sistema de protección anticopia de la mano de Microsoft.

Procesador Core 2 Duo y Core 2 Quad

Últimamente se libero la gama Core 2 Duo y Core 2 Quad, los cuales engloban dos procesadores físicos dentro de uno solo, obteniendo resultados impresionantes.

En un futuro se lograra desarrollar procesadores con 12, 32 y 80 núcleos, algo que nunca antes se pudo haber imaginado alguien.²³

2.2 TIPOS DE MICROPROCESADORES SEGÚN NÚMERO DE BITS

2.2.1 Microprocesadores De 8 Bits

Es el micro en el que están basados los ordenadores personales actuales. Se caracteriza por tener un bus de 8 bits y una velocidad de impulso de reloj de 4,77 MHz Otros micros de 8 bits pueden ser: el Z80 de Zilog, el 6800 de Motorola y el 1802 de RCA. Estos micros se incorporan en ordenadores del tipo XT.²⁴

²³ “Historia de los Microprocesadores” 19 de Mayo de 2009. <http://blog.puceing.edu.ec>

²⁴ “Componentes, Estructura Y Funcionamiento De La UCP”. <http://html.rincondelvago.com>

2.2.1.1 Microprocesador 8085²⁵

El primer microprocesador fue introducido por Intel Corporation en 1971. Intel introdujo ese año el 4004 de 4 bits y el 8008 de 8 bits. En 1974, Intel introdujo el microprocesador 8080, que maneja palabras de datos de 8 bits y tiene 16 líneas de dirección y un puntero de pila de 16 bits. El microprocesador Intel 8085 es una versión mejorada del 8080. Integra el reloj, control del sistema y prioridad de interrupciones en el CI microprocesador, reduciendo así el número de CI utilizados en la mayoría de los sistemas. El 8085 también opera con una sencilla fuente de alimentación de +5V. La MPU 8085 utiliza las mismas instrucciones que el 8080, haciendo así los programas compatibles. El microprocesador 8085 tiene dos instrucciones más para aprovechar las características del hardware añadido.

Una tendencia en la evolución de los microprocesadores ha sido integrar más funciones en menos CI. Un sencillo sistema basado en microprocesador.

El 8080/8085 y sus parientes de 16 y 32 bits se han convertido en algunos de los microprocesadores de propósito general más ampliamente utilizados en el mundo. Entre los parientes del microprocesador 8080/8085 están los Intel 8086, 8088, 80186, 80188, 80286 y 80386. Algunos de estos circuitos integrados fueron seleccionados para utilizarlos como CPU de las populares microcomputadoras IBM.

Funciones Y Diagramas De Pines Del 8085

El microprocesador de ocho bits Intel 8085 está ubicado en un encapsulamiento de doble línea (DIP) de 40 pines. El diagrama de pines del MPU 8085 se produce en la figura 2.5.

El microprocesador genérico utilizaba 16 pines para las líneas de dirección y ocho pines para las conexiones del bus de datos. Con las funciones añadidas del 8085,

²⁵ Sclipa, Andrés L. **Microprocesadores de los Sistemas de Control**. 6^{to} año 1^{ra} división Electromecánica, 1998

el de DIP de 40 pines no tiene pines suficientes para todas las entradas y salidas. Por esta razón, el fabricante utiliza las pines 12 a 19 como líneas de bus de propósito doble de direcciones/datos (AD_0 - AD_7). Se dice que esta unidad tiene un bus multiplexado de datos/direcciones de 8 bits. Las 10 con las ocho líneas de dirección menos significativas comparten las pines con las ocho líneas del bus de datos multiplexado significa primero seleccionar una y después otra, etc. Por tanto, multiplexado del bus de direcciones/datos significa primero utilizar el bus para enviar una dirección y a continuación enviar o recibir datos vía el mismo bus. El 8085 tienen la señal especial para informar a los periféricos cuando el bus de direcciones/datos envía una dirección y cuando funcionan como bus de datos. La señal especial se denomina señal de control de habilitación del cerrojo de direcciones (ALE). Observar que las pines del bus de direcciones/datos son bidireccionales o pueden ser de tres estados o alta impedancia (también llamadas triestado). La patilla ALE es una salida de control.²⁶

El 8085, igual que el microprocesador genérico tiene un total de 16 líneas de dirección. Las ocho líneas de dirección más significativas están conectadas a las pines A_8 a A_{15} en la figura. Estas se conectan directamente al bus de direcciones como en el microprocesador genérico estas pines son de salida o pueden ser de tres estados. Otras pines similares a las encontradas en el microprocesador genérico son las de alimentación (V_{cc} y V_{ss}) conectada a una fuente de alimentación de +5V. El microprocesador 8085 tiene circuitería de reloj permanente sin más que conectar las pines de entrada X_1 y X_2 a un cristal. La frecuencia interna de la MPU será la mitad de la frecuencia del cristal.

Algunas de las pines del 8085 listadas en la figura 2.5 realizan funciones de control similares a las observadas en el microprocesador genérico. La salida del control de lectura (\overline{RD}) y escritura (\overline{WR}) son utilizadas para informar a la memoria o dispositivos de E/S cuando hay que enviar o recibir datos vía el bus de datos

²⁶ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

(realmente por el bus de direcciones/datos en esta unidad). La entrada reinicialización (REINICIALIZACIÓN IN) actúan como la reinicialización de la MPU genérico, ya que el contador de programa se pone, al reiniciar, a 0000 H. Los buses de datos y direcciones y líneas de control están en el estado de tres estados durante la reinicialización. El contenido de los registros internos también puede ser alterado durante una reinicialización. La patilla REINICIALIZACION OUT está asociada con la operación del reinicialización. Cuando la MPU se está reiniciando, la patilla REINICIALIZACION OUT envía una señal a los periféricos que les informa que el sistema se va a reiniciar.

La salida de reloj (CLK) del microprocesador 8085 de la figura 2.5 opera igual que la de la MPU genérica. La entrada de petición de interrupción (INTR) al 8085 es una interrupción de propósito general como en el microprocesador genérico. Sin embargo, de forma distinta a la interrupción del microprocesador genérico, la INTR del 8085 habilitada o inhabilitada por instrucciones software. Además de la petición regular de interrupciones (INTR), la MPU 8085 tiene otras cuatro entradas de interrupción. Éstos son las pines de las entradas TRAP, RST 7.5, RST 6.5 y RST 5.5. La TRAP es la interrupción de más alta prioridad, seguida por RST 7.5, RST 6.5 y RST 5.5 y finalmente por la interrupción, INTR. TRAP o cualquiera de los tres rearranques (RST 7.5, RST 6.5 y RST 5.5) hacen que la MPU salte a una subrutina en una dirección específica de llamada. Las interrupciones de rearranque pueden ser habilitadas o inhabilitadas por software, mientras de la interrupción TRAP no puede ser inhabilitada. La interrupción INTR salta a una dirección dictada por una instrucción especial recibida, de un dispositivo periférico, cuando es activada la salida del conocimiento de interrupción ($\overline{\text{INTA}}$) de la MPU.²⁷

El microprocesador 8085 tiene una entrada/salida serie primitivas. Éstas son las pines SID (entrada de datos serie) y SOD (salida de datos serie) mostrado en la figura 2.5.

²⁷ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

Considerar la entrada READY al microprocesador 8085 de la figura 2.5. Esta entrada proveniente de un dispositivo periférico informa al microprocesador que está listo para enviar o recibir datos. Si la entrada READY del dispositivo periférico está en el nivel BAJO, durante un ciclo de lectura o escritura, la MPU lo interpreta como una petición para entrar en el estado de espera, donde permanecerá hasta que las señales periféricas estén listas para transmitir o recibir datos. La MPU entonces procederá con el ciclo de lectura o escritura. La entrada READY es manual cuando se utilizan memorias o periféricos que son demasiado lentos comparados con la velocidad de tratamiento del microprocesador.²⁸

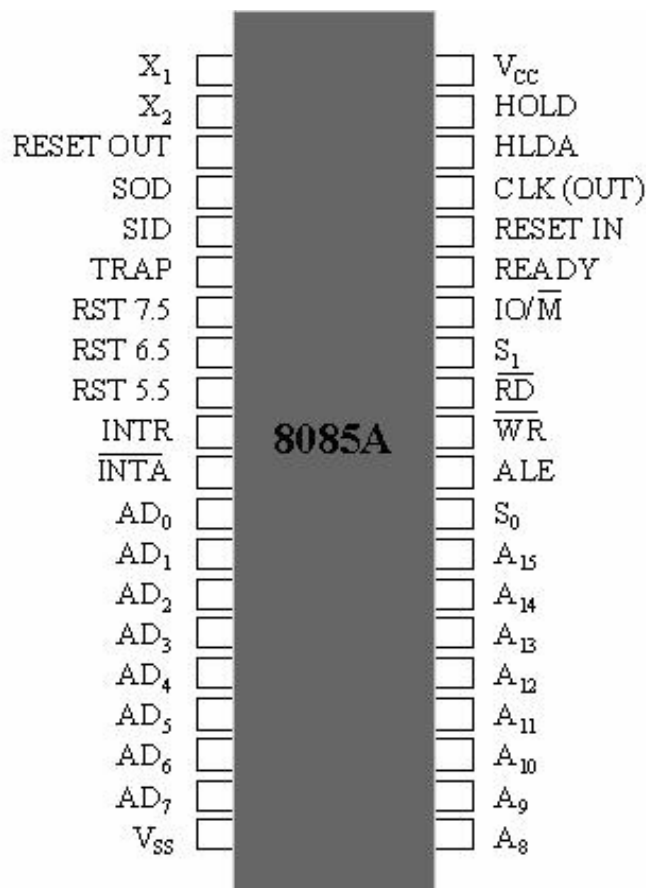


Fig. 2.5 - Configuración de terminales del microprocesador 8085A

²⁸ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990

Considerar las pines de entrada HOLD (mantenimiento) y de salida HLDA (conocimiento de mantenimiento) de la MPU 8085 de la Figura 2.5. La entrada HOLD notifica a la MPU que otro dispositivo quiere utilizar los buses de direcciones y datos. Esto puede ocurrir durante operaciones de acceso directo a memoria. Después de recibir una entrada HOLD, la CPU completara las transferencias actuales de datos en los buses.²⁹

Entonces las pines de direcciones, datos, \overline{RD} , \overline{WR} e IO/\overline{M} del 8085 se ponen en alta impedancia para no interferir con las transferencias de datos en los buses. Una salida HLDA indica a un periférico que se ha recibido una petición de mantenimiento y que el microprocesador abandonara el control de los buses en el siguiente ciclo de reloj.

Las salidas IO/\overline{M} , S_0 y S_1 son señales de control que notifican a los periféricos el tipo de ciclo de maquina que está realizando la MPU.

Arquitectura Del 8085

El microprocesador 8085 está constituido por tres grandes bloques: una unidad de control, un conjunto de registros y una unidad aritmética-lógica. Es difícil separarlos, debido a la interrelación que existe entre todos ellos; no obstante, para facilitar su comprensión, se estudiaran en primer lugar por separado.

²⁹ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990

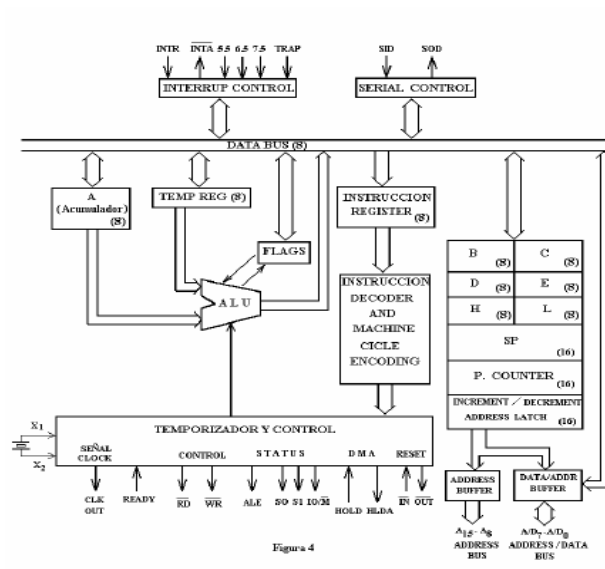


Fig. 2.6 - Unidad aritmético-lógica completa.

La organización interna o arquitectura del microprocesador Intel 8085 se reproduce en la (figura 2.6). El 8085 tiene un contador de programa y un cerrojo de direcciones de 16 bits que alimenta al bus de direcciones dedicado ($A_{15} - A_8$) y al uso de direcciones/datos de doble propósito ($AD_7 - AD_0$). Los datos paralelos entran y dejan la MPU vía el bus de direcciones/datos multiplexado ($AD_7 - AD_0$). El bus de direcciones/datos transmite una dirección cuando la línea de control ALE esta en ALTA y datos cuando la línea ALE esta en BAJA.³⁰

Las entradas o salidas de datos a la unidad se realizan a través del bus de datos interno de 8 bits. Los datos pueden fluir desde el bus de datos interno al acumulador de 8 bits o a los registros temporales, señalizadores, registros de instrucción, unidad de control de interrupción, unidad de control de E/S serie, cualquiera de los registros de propósito general (B, C, D, E, H y L), puntero de pila de 16 bits, contador de programa de 16 bits, o buffer de datos/direcciones de 8 bits. Las entradas y salidas serie (SID y SOD) se muestran en la parte superior derecha de la Figura 2.6. Las entradas de interrupción (INTR, RST 5.5, RST 6.5, RST 7.5, y TRAP) y la salida INTA (conocimiento de interrupción) se muestran en la parte superior izquierda. La

³⁰ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990

unidad aritmético-lógica (ALU) está alimentada por dos registros de 8 bits (registro acumulador y registro temporal) como en el microprocesador genérico. Los flips-flops señalizadores tienen cinco indicadores de status en lugar de dos como en el microprocesador genérico.

El registro de instrucción alimenta al decodificador de instrucciones, el cual interpreta la instrucción actual y determina el microprograma que debe seguir. El decodificado de instrucciones entonces instruye a las secciones de control y temporización sobre la secuencia de eventos que deben realizarse.

Registros

Igual que el microprocesador genérico, la MPU 8085 utiliza registros de 8 y 16 bits. El 8085 tiene 8 registros direccionables de 8 bits. Seis de estos registros pueden utilizarse como registros de 8 bits o como registros pares de 16 bits. Además, el 8085 contiene dos registros más de 16 bits. Los registros del 8085 son los siguientes:

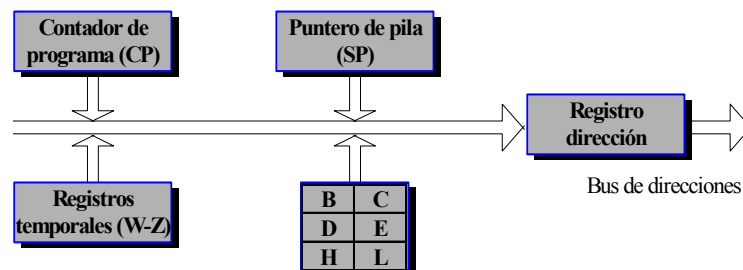


Fig. 2.7 - Registros del microprocesador 8085.³¹

³¹ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

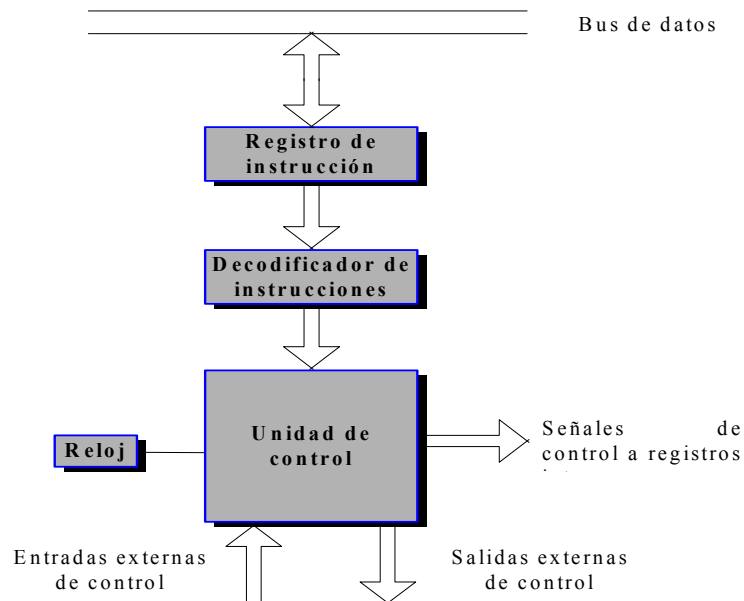


Fig. 2.8 - Unidad de control conectada al registro de instrucción (RI) y al decodificador de instrucciones.³²

1. El acumulador (registro A) es el foco de todas las operaciones del acumulador que incluyen instrucciones aritméticas, lógicas, de carga y almacenamiento y de E/S. Es un registro de 8 bits.
2. Los registros de propósito general BC, DE y HL pueden ser utilizados como seis registros de 8 bits o como tres registros de 16 bits dependiendo de la instrucción que se vaya a ejecutar. Como en el microprocesador genérico, el registro par HL llamado puntero de datos por INTEL, puede ser utilizado para apuntar a direcciones. Pocas instrucciones utilizan los registros BC y DE como punteros de dirección, ya que normalmente se utilizan como registros de datos de propósito general.
3. El contador de programa (PC) siempre señala la posición de memoria de la siguiente instrucción que se va a ejecutar. Siempre contiene una dirección de 16 bits.
4. El puntero de pila (SP) es un puntero de dirección de propósito especial (o puntero de dato) que siempre apunta a la parte superior de la pila de la RAM. Es un registro de 16 bits.

³² Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

5. El registro de señalización contiene cinco señalizadores de 1 bit que contiene información del estado de la CPU. Estos señalizadores son utilizados entonces por instrucciones de bifurcación condicional, llamada y vuelta de subrutina.

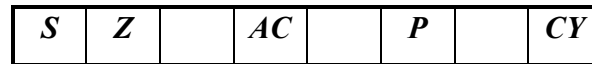


Fig. 2.9 - Registro de estado con la situación de cada flag.

Señalizadores³³

Los cinco señalizadores encontrados en la CPU del 8085 están representados en la figura 2.9. El señalizador de arrastre (CY) es puesto a 1 o 0 por operaciones aritméticas, y su status es examinado por las instrucciones del programa. Como en el microprocesador genérico, un <<overflow>> (rebose) de una suma de 8 bits hará que el bit CY se ponga a 1. En la resta el señalizador CY actúa como un señalizador de <<préstamo>>, indicando que es menor que el sustraendo si el señalizador esta a 1.

El señalizador de cero (Z) se pone a 1 si el resultado de ciertas instrucciones es cero. El señalizador de cero esa a 0 si el resultado no es cero. La acción del señalizador de cero se observo en el microprocesador genérico.

El señalizador de signo (S) indica la condición del bit más significativo del acumulador después de la ejecución de instrucciones aritmético-lógicas. Estas instrucciones utilizan el MSB del dato para representar el signo del número contenido en el acumulador. Un señalizador de signo a 1 representa a un número negativo, mientras que a 0 significa un número positivo.

El señalizador de arrastre auxiliar (AC) indica un <<overflow>> o arrastre del bit 3 del acumulador de la misma forma que el señalizador de arrastre indica un

³³ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

<<overflow>> del bit 7. Este señalizador es utilizado comúnmente en aritmética BCD (decimal codificado binario).

El señalizador de paridad (P) examina el número de bits 1 del acumulador. Si este contiene un número par de 1, se dice que existe paridad par y el señalizador de paridad se pone a 1. Sin embargo si el acumulador contiene un número impar de 1 (paridad impar), el señalizador de paridad del 8085 se pone a 0. Por ejemplo, si una instrucción ADD da como resultado la suma 00110011_2 en el acumulador, el señalizador de paridad estará a 1 porque hay un número par de 1 (4). Sin embargo, si la suma fuese 10101110_2 , el señalizador P estaría a 0 porque hay un número impar (5) de 1 en el acumulador.

Puntero de Pila³⁴

El puntero de pila contiene la dirección del último byte introducido en la pila. El puntero de pila puede ser inicializado para utilizar cualquier parte de la RAM como pila. Como en el microprocesador genérico, el puntero de pila se decrementa cada vez que se introduce un dato en la pila y se incrementa cada vez que se saca el dato de la pila.

Unidad Aritmética-Lógica

La ALU está muy relacionada con el acumulador, registro de señalizadores y algunos registros temporales que son inaccesibles al programador. Las operaciones aritméticas, lógicas y de desplazamiento circular son realizadas en la ALU. Los resultados normalmente se depositan en el acumulador.

Registro de Instrucción y Decodificador

Durante una búsqueda de instrucción, el primer byte de la instrucción, el código de op, es transferido al registro de instrucción de 8 bits. El contenido del registro de instrucción está disponible al decodificador de instrucciones. La salida del

³⁴ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

decodificador, a través de puertas y con señales de temporización, controla los registros, ALU y buffer de datos y direcciones. Las salidas del decodificador de instrucciones y del generador de reloj interno producen las señales de temporización de ciclo y estado de la maquina.

Generador Interno de Reloj

La MPU 8085 incorpora un generador completo de reloj en su patilla. Requiere solamente la adición de un cristal de cuarzo para establecer la temporización de su operación. El 8085 A puede utilizar un cristal de hasta 6.25 MHz La versión 8085A_2 del 8085 opera con un cristal de hasta 10 MHz. La patilla de salida CLK es una salida de reloj con buffer que tiene una frecuencia una vez y media la frecuencia de cristal. Las entradas de reloj del 8985AH, 8085AH_2 y 8085 AH_1 también pueden ser excitadas con cristales de 6, 10 o 12 MHz³⁵

Interrupciones

La interrupción de más alta prioridad es la de la entrada TRAP. Una vez que alcanza el nivel ALTO, la entrada TRAP hará que el 8085 guarde en pila el contenido del contador del programa y salte a la posición de memoria 0024H. La entrada TRAP no puede ser inhabilitada y por tanto se denomina interrupción no enmascarable.

Las tres siguientes interrupciones hardware se denominan reinicializadoras (para reinicializar el programa en una posición de memoria). La interrupción RST 7.5 hace que el 8085 guarde en la pila el contenido del contador de programa y salte a la posición de memoria 003CH. La siguiente interrupción de prioridad más elevada es la RST 6.5, que hace que el 8085 guarde en la pila el contenido del contador de programa y salte a la posición de memoria 0034H. La interrupción de prioridad más baja RST 5.5 hace que el 8085 guarde en la pila el contenido del contador de programa y salte a la posición de memoria 002CH.

³⁵ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

La interrupción de hardware de más baja prioridad es la entrada INTR, que hace que el procesador busque una instrucción desde una fuente externa especial. Las cuatro últimas interrupciones pueden ser habilitadas o inhabilitadas por software.

Entrada y Salida Serie

Las pines de entrada y salida serie del 8085 ayudan a minimizar las cuentas de pines en los pequeños sistemas al proporcionar un puerto de interfaz serie. La instrucción especial RIM del 8085 transfiere datos desde la patilla de entrada serie (SID) al bit 7 del acumulador. Donde se transfiere un nivel ALTO desde la entrada SID a la posición MSB del acumulador.

Modos de Direccionamiento

Como en el microprocesador genérico, la MPU 8085 utiliza cinco modos de direccionamiento. Estos son:

1. Direccionamiento implicado.
2. Direccionamiento de registro.
3. Direccionamiento inmediato.
4. Direccionamiento directo.
5. Direccionamiento indirecto de registro.

Direccionamiento Implicado³⁶

El modo de direccionamiento de ciertas instrucciones de la MPU 8085 es implicado por (o inherente a) la función de la instrucción. Por ejemplo, la instrucción STC (poner a 1 el señalizador de arrastre) está relacionada con el señalizador de arrastre solamente y no con otros registros o posiciones de memoria.

Direccionamiento de Registro.

Muchas instrucciones de la MPU 8085 utilizan direccionamiento de registro. Cuando se utilizan estas instrucciones, se especifican las fuentes del operando y la

³⁶ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

operación. Considerar la instrucción ADD C. El operando del registro fuente (registro C en este caso) se suma al operando localizado en el acumulador. Una vez que se ejecuta la instrucción ADD C, la suma se deposita en el acumulador. Como es habitual, la operación de suma también afecta a los señalizadores del registro de status.

La instrucción ADD C utilizaba el direccionamiento de registros. Ambos operandos estaban localizados en registros internos de la MPU (registro A y C). Las instrucciones que utilizan el direccionamiento de registro son muy eficientes ya que solamente utilizan un espacio de la memoria de programa de 1 byte. También se ejecuta rápidamente porque no tienen que buscar operandos en memoria. Buscar en memoria es una operación relativamente lenta.

Direccionamiento Inmediato.³⁷

Las instrucciones que utilizan direccionamiento inmediato tienen los datos inmediatamente a continuación del código op en la memoria del programa. El microprocesador busca el código de op (C6H como ejemplo) en la memoria del programa. Después de decodificar, la MPU determina que esta es una instrucción de direccionamiento inmediato. Por tanto encuentra el dato inmediato en la siguiente posición consecutiva de la memoria de programa después del código de op. Este dato inmediato se suma al contenido del acumulador. Después de la operación, la suma se coloca en el acumulador.

Las instrucciones inmediatas del microprocesador 8085 utilizan el acumulador como operando implicado, como en la operación ADI (suma inmediata). La instrucción MVI (transferir de inmediato) del 8085 puede transferir datos inmediatos a cualquier registro o posición de memoria. La instrucción LXI (carga de registro par

³⁷ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

inmediato) del 8085 carga un registro par con un valor de 16 bits y por tanto no afecta al acumulador.

Direccionamiento Directo.³⁸

Las operaciones que utilizan direccionamiento directo en la MPU del Intel 8085 se especifican utilizando formatos de instrucción de 3 bytes.

El byte 1 contiene el código de op para la instrucción de direccionamiento directo. El byte 2 contiene el byte de orden inferior de la dirección del operando. El byte 3 de la instrucción contiene el byte de orden superior de la dirección del operando.

El código op de la instrucción LDA es 3AH. Los 2 siguientes bytes de la memoria de programa son ensamblados por el microprocesador en una dirección de 16 bits. Esta dirección de la memoria de datos es accedida entonces por la MPU y su contenido es cargado entonces en el acumulador.

Las instrucciones que utilizan direccionamiento directo a veces son evitadas por diseñadores de microprocesadores y por los programadores. El direccionamiento directo requiere de mucho espacio de la memoria de programa (3 bytes de memoria de programa por una instrucción de la MPU 8085). Las operaciones de direccionamiento directo también requieren un tiempo de ejecución relativamente largo, debido a los numerosos accesos a memoria que se necesitan.

Direccionamiento Indirecto de Registro

Las instrucciones de direccionamiento indirecto referencian la memoria utilizando el contenido de un registro par para señalar la dirección del operando. La instrucción ADD *M* (sumar memoria); suma el contenido del acumulador con el

³⁸ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

contenido de la posición de memoria indicado por la dirección del registro par *HL* de la MPU.

Modos de Direccionamiento Combinado

Algunas instrucciones de la MPU 8085 utilizan una combinación de modos de direccionamiento. Una instrucción CALL (LLAMADA) a subrutina, por ejemplo, combina el direccionamiento directo e indirecto del registro. El direccionamiento directo en una instrucción CALL especifica la dirección de la subrutina deseada. El direccionamiento indirecto de registro es el del puntero de pila. La instrucción CALL introduce primero el contenido actual del contador de programa en la posición de memoria específica por el puntero de pila. A continuación, el procesador carga la dirección directa en el contador de programa. Finalmente, la MPU salta a la subrutina cuya dirección se encuentra ahora en el contador del programa.

Instrucciones Del Microprocesador 808539

Cuando se refiere a los registros "r" (ya sea r1, r2 o simplemente r) se está hablando de cualquier registro propio del 8085, A (Acumulador), o los registros B, C, D, E, H, L de 8 bits respectivamente.

Estos registros se pueden agrupar por parejas BC, DE, HL (rp) y así formar registros de 16 bits para direccionar una posición de memoria directamente(16 BITS), siendo el par HL el más importante por tener instrucciones que trabajan directamente con él. Referirse a Anexos 1 (Tabla 1, 2, 3, 4 y 5).

2.2.2 Microprocesadores De 16 Bits

Otro microprocesador de 16 bits a destacar es el Motorola 68000, su arquitectura es diferente pero el resultado es el mismo. El 8086 es uno de los

³⁹ Roger, Tokheim. **Fundamentos de Microprocesadores**. Editorial McGraw-Hill, 1990.

primeros micros de 16 bits con una potencia bruta cercana a 0,2 MIPS, funciona a 8 Mhz. Su capacidad de direccionamiento es de 1 MB. Al igual que el Intel 8088 forma parte de los ordenadores del tipo XT.

El micro 80286 es prácticamente igual al anterior a excepción de sus prestaciones ya que tiene una potencia 4 o 5 veces mayor, su potencia bruta es de 1 MIPS, por lo que la velocidad ha aumentado sin modificar el software.

Con este micro se pasa de un direccionamiento de 16 MB a otro de 1 GB, a parte permite detectar y eliminar los posibles errores de Software. Otra cosa a tener en cuenta es que los programas escritos en Ensamblador sufren ligeras modificaciones.

2.2.2.1 Historia Del 8086/8088⁴⁰

En junio de 1978 Intel lanzó al mercado el primer microprocesador de 16 bits: el 8086. En junio de 1979 apareció el 8088 (internamente igual que el 8086 pero con bus de datos de 8 bits) y en 1980 los coprocesadores 8087 (matemático) y 8089 (de entrada y salida). El primer fabricante que desarrolló software y hardware para estos chips fue la propia Intel. Reconociendo la necesidad de dar soporte a estos circuitos integrados, la empresa invirtió gran cantidad de dinero en un gran y moderno edificio en Santa Clara, California, dedicado al diseño, fabricación y venta de sus sistemas de desarrollo que, como se explicó anteriormente, son computadoras autosuficientes con el hardware y software necesario para desarrollar software de microprocesadores.

Los sistemas de desarrollo son factores clave para asegurar las ventas de una empresa fabricantes de chips. La inmensa mayoría de ventas son a otras empresas, las cuales usan estos chips en aparatos electrónicos, diseñados, fabricados y comercializados por ellas mismas. A estas empresas se las llama "fabricantes de equipo original", o en inglés, OEM (Original Equipment Manufacturer). El disminuir el tiempo de

⁴⁰ Alpern, Darío. "Los microprocesadores". www.alpertron.com.ar (16 de Abril, 2009)

desarrollo de hardware y software para las OEM es esencial, ya que el mercado de estos productos es muy competitivo. Necesitan soporte pues los meses que les puede llevar el desarrollo de las herramientas apropiadas les pueden significar pérdidas por millones de dólares. Además quieren ser los primeros fabricantes en el mercado, con lo cual pueden asegurarse las ventas en dos áreas importantes: a corto plazo, ya que al principio la demanda es mucho mayor que la oferta, y a largo plazo, ya que el primer producto marca a menudo los estándares.⁴¹

De esta manera la empresa Intel había desarrollado una serie completa de software que se ejecutaba en una microcomputadora basada en el 8085 llamada "Intellec Microcomputer Development System". Los programas incluían ensambladores cruzados (éstos son programas que se ejecutan en un microprocesador y generan código de máquina que se ejecuta en otro), compiladores de PL/M, Fortran y Pascal y varios programas de ayuda. Además había un programa traductor llamado CON V86 que convertía código fuente 8080/8085 a código fuente 8086/8088. Si se observan de cerca ambos conjuntos de instrucciones, queda claro que la transformación es sencilla si los registros se traducen así: A -> AL, B -> CH, C -> CL, D -> DH, E -> DL, H -> BH y L -> BL. Puede parecer complicado traducir LDAX B (por ejemplo) ya que el 8088 no puede utilizar el registro CX para direccionamiento indirecto, sin embargo, se puede hacer con la siguiente secuencia: MOV SI, CX; MOV AL, [SI]. Esto aprovecha el hecho que no se utiliza el registro SI. Por supuesto el programa resultante es más largo (en cantidad de bytes) y a veces más lento de correr que en su antecesor 8085. Este programa de conversión sólo servía para no tener que volver a escribir los programas en una primera etapa. Luego debería reescribirse el código fuente en assembler para poder obtener las ventajas de velocidad ofrecidas por el 8088. Luego debía correr el programa en la iSBC 86/12 Single Board Computer basado en el 8086. Debido al engorro que resultaba tener dos plaquetas diferentes, la empresa Godbout Electronics (también de California) desarrolló una placa donde estaban el 8085 y el 8088, donde se utilizaba un

⁴¹ Alpern, Darío. "Los microprocesadores". www.alpertron.com.ar (16 de Abril, 2009)

ensamblador cruzado provisto por la compañía Microsoft. Bajo control de software, podían conmutarse los microprocesadores. El sistema operativo utilizado era el CP/M (de Digital Research).⁴²

El desarrollo más notable para la familia 8086/8088 fue la elección de la CPU 8088 por parte de IBM (International Business Machines) cuando en 1981 entró en el campo de las computadoras personales. Esta computadora se desarrolló bajo un proyecto con el nombre "Acorn" (Proyecto "Bellota") pero se vendió bajo un nombre menos imaginativo, pero más correcto: "Computadora Personal IBM", con un precio inicial entre 1260 dólares y 3830 dólares según la configuración (con 48KB de memoria RAM y una unidad de discos flexibles con capacidad de 160KB costaba 2235 dólares). Esta computadora entró en competencia directa con las ofrecidas por Apple (basado en el 6502) y por Radio Shack (basado en el Z-80)

2.2.2.2 Arquitectura De Los Procesadores 8088 Y 8086

El 8086 es un microprocesador de 16 bits, tanto en lo que se refiere a su estructura como en sus conexiones externas, mientras que el 8088 es un procesador de 8 bits que internamente es casi idéntico al 8086. La única diferencia entre ambos es el tamaño del bus de datos externo. Intel trata esta igualdad interna y desigualdad externa dividiendo cada procesador 8086 y 8088 en dos sub-procesadores. O sea, cada uno consta de una unidad de ejecución (EU: Execution Unit) y una unidad interfaz del bus (BIU: Bus Interface Unit). La unidad de ejecución es la encargada de realizar todas las operaciones mientras que la unidad de interfaz del bus es la encargada de acceder a datos e instrucciones del mundo exterior. Las unidades de ejecución son idénticas en ambos microprocesadores, pero las unidades de interfaz del bus son diferentes en varias cuestiones, como se desprende del siguiente diagrama en bloques:

⁴² Alpern, Darío. "Los microprocesadores". www.alpertron.com.ar (16 de Abril, 2009).

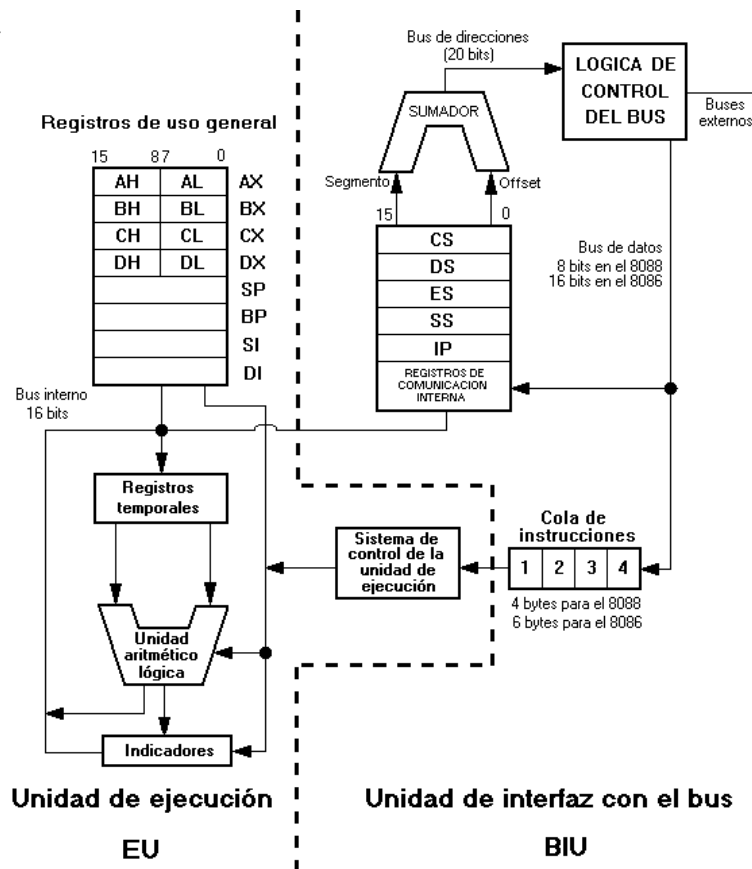


Fig. 2.10 – Diagrama en bloques de 8086

La ventaja de esta división fue el ahorro de esfuerzo necesario para producir el 8088. Sólo una mitad del 8086 (el BIU) tuvo que rediseñarse para producir el 8088.

2.2.3 Microprocesadores De 32 Bits⁴³

Los motivos en la carrera de los micros están en el tamaño de las aplicaciones y en la velocidad. Otra de las razones es la de protección ya que estamos trabajando con sistemas operativos complejos, multitarea y/o multiprogramación. Todo esto se alcanzó con el 80286 aunque en los posteriores micros se ha perfeccionado, aumentado en velocidad y direccionamiento. El 80386 direcciona hasta 4GB reales y tiene una potencia bruta de 3 a 4 MIPS.

⁴³ Díaz, Bethsy. "Microprocesadores" 23 de Marzo, 2000. www.monografias.com (15 de Abril, 2009)

En Arquitectura de computadoras, 32 bits se emplea para describir enteros, direcciones de memoria u otras unidades de datos que tienen un ancho de 32 bits. También se utiliza para hacer referencia a las arquitecturas de CPU basadas en registros, bus de direcciones o bus de datos de un ancho de 32 bits.

Los microprocesadores de 32 bits pueden hacer referencia a la longitud o tamaño de la palabra (unidad de trabajo básica) de su microprocesador o bien, más comúnmente, al número de bits transferidos a través del bus de datos de la computadora (la trayectoria de datos por la que viaja la información hacia y desde el microprocesador) en un determinado momento.

Por consiguiente, un microprocesador de 32 bits tiene un tamaño de palabra de 32 bits, o 4 bytes; un bus de datos de 32 bits tiene 32 líneas de datos, por lo que transporta la información dentro del sistema en conjuntos de 32 bits cada vez.

Los modelos de Apple a partir del Macintosh II y sus versiones posteriores son unas máquinas de 32 bits, tanto en términos de la longitud de palabra de su microprocesador como del tamaño del bus de datos, al igual que el IBM PS/2 modelo 80 y similares basados en el microprocesador Intel 80386.⁴⁴

2.2.4 Microprocesadores De 64 Bits⁴⁵

Los microprocesadores de 64 bits han existido en las supercomputadoras desde 1960 y en servidores y estaciones de trabajo basadas en RISC desde mediados de los años 1990. En 2003 empezaron a ser introducidos masivamente en las computadoras

⁴⁴ "Componentes, Estructura Y Funcionamiento De La UCP". <http://html.rincondelvago.com>

⁴⁵ "64 bits" 4 de marzo, 2008. <http://es.wikipedia.org> (12 de julio, 2009)

personales (previamente de 32 bits) con las arquitecturas x86-64 y los procesadores PowerPC G5.

Aunque una CPU puede ser internamente de 64 bits, su bus de datos o bus de direcciones externos pueden tener un tamaño diferente, más grande o más pequeño y el término se utiliza habitualmente para describir también el tamaño de estos buses. Por ejemplo, muchas máquinas actuales con procesadores de 32 bits usan buses de 64 bits (p.ej. el Pentium original y las CPUs posteriores) y pueden ocasionalmente ser conocidas como "64 bits" por esta razón. El término también se puede referir al tamaño de las instrucciones dentro del conjunto de instrucciones o a cualquier otro elemento de datos (p.ej. las cantidades de 64 bits de coma flotante de doble precisión son comunes). Sin más calificaciones, sin embargo, la arquitectura de las computadoras de 64 bits tiene integrados registros que son de 64 bits, que permite soportar (interna y externamente) datos de 64 bits.

2.2.4.1 La Arquitectura DEC Alpha⁴⁶:

La organización de sus registros es de uso general con una arquitectura que se puede encuadrar como de registro-registro. Esto hace que la mayoría de sus instrucciones operen sobre los registros, haciendo uso de la memoria RAM sólo para instrucciones de carga y almacenamiento. La razón es que se intenta minimizar los accesos a memoria, puesto que suponen el cuello de botella para los procesadores actuales. La longitud de palabra de los registros es de 64 bits, ya sea desde el PC (contador de programa), pasando por los registros de enteros, punto flotante, etc. El primer procesador que hizo gala de la tecnología Alpha fue el 21064.

⁴⁶ Cavazos, Anabel. "Chipset para Microprocesadores de 64 bits" 22 de octubre, 2009. www.scribd.com (15 de Abril, 2009)



Fig. 2.11 – Microprocesador de 21064

2.2.4.2 La Arquitectura IA-64⁴⁷

Línea de procesadores Itanium e Itanium 2. Representan el diseño de producto más complejo del mundo con más de 1.700 millones de transistores. Esto permite obtener sólidas capacidades de virtualización, mejorar la confiabilidad y niveles de rendimiento líderes del mercado.

2.2.4.3 La Arquitectura AMD64

El conjunto de instrucciones del AMD x86-64 (renombrado posteriormente como AMD64) es una extensión directa de la arquitectura del x86 a una arquitectura de 64 bits, motivado por el hecho de que los 4GB de memoria que son direccionables directamente por una CPU de 32 bits ya no es suficiente para todas las aplicaciones. El primer procesador con soporte para este conjunto de instrucciones fue el Opteron, lanzado en abril de 2003. Posteriormente ha sido implementado en múltiples variantes del Athlon 64 y del Pentium 4 de Intel, en éste último caso bajo una versión de Intel llamada EM64T.

2.2.4.4 La Arquitectura SPARC

Es la primera arquitectura RISC abierta y como tal las especificaciones de diseño están publicadas, así otros fabricantes de microprocesadores pueden desarrollar su propio diseño.

⁴⁷ Cavazos, Anabel. "Chipset para Microprocesadores de 64 bits" 22 de octubre, 2009. www.scribd.com (15 de Abril, 2009)

Una de las ideas innovadoras de esta arquitectura es la ventana de registros que permite hacer fácilmente compiladores de alto rendimiento y una significativa reducción de memoria en las instrucciones load/store en relación con otras arquitecturas RISC. Las ventajas se aprecian sobre todo en programas grandes.

La CPU SPARC está compuesta de una unidad entera, UI (*Integer Unit*) que procesa la ejecución básica y una FPU (*Floating-Point Unit*) que ejecuta las operaciones y cálculos de reales. La IU y la FPU pueden o no estar integradas en el mismo chip.

Aunque no es una parte formal de la arquitectura, las computadoras basadas en sistemas SPARC de Sun Microsystems tienen una unidad de manejo de memoria (MMU) y un gran caché de direcciones virtuales (para instrucciones y datos) que están dispuestos periféricamente sobre un bus de datos y direcciones de 32 bits.⁴⁸



Fig. 2.12 – Microprocesador Ultra Sparc II

2.2.4.5 La Arquitectura POWER

Es usada en servidores IBM. Pero sin embargo hay muchos microprocesadores que son derivados o variantes de este que se encuentran en gran variedad de equipos que van desde computadores para automóviles hasta consolas de videojuegos. Su nombre proviene de "Performance Optimization With Enhanced RISC".

⁴⁸ Cavazos, Anabel. "Chipset para Microprocesadores de 64 bits" 22 de octubre, 2009. www.scribd.com (15 de Abril, 2009)

2.2.4.6 La Arquitectura PA-RISC

Una característica interesante de PA-RISC es que la mayoría de sus microprocesadores no tiene caché L2. En su lugar se implementaba una caché L1 mayor, formada por chips separados conectados al microprocesador a través de un *bus* (actualmente está integrada en el propio chip). Sólo el modelo PA-7300LC tiene caché L2. Otra innovación de esta arquitectura fue la adición de un repertorio de instrucción multimedia (SIMD) conocido como MAX e introducido por primera vez en el 7100LC. También se suelen referir a ella como la arquitectura HP/PA, Hewlett Packard Precision Architecture. PA se desarrolla en Palo Alto, donde se encuentra la central de HP.⁴⁹

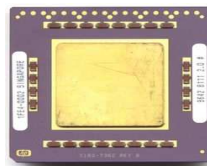


Fig. 2.13 – Microprocesador HP Precisión

2.2.4.7 Chipset Para Microprocesadores De 64 Bits

- Intel 4004: f14cadc
- MOS 6502: 65co2, 65c816
- Motorola 6800: 6805,6807,6808
- Zilog z80: Amstrad
- Intel 8086: mycron 2000
- Intel Pentium: 430hx,430tx
- Intel Pentium II: 450nx,440bx,440ex
- Intel Pentium III: 840,815,815e,810,810ez
- Intel Pentium 4: 423,478,775,850

⁴⁹ Cavazos, Anabel. "Chipset para Microprocesadores de 64 bits" 22 de octubre, 2009. www.scribd.com (15 de Abril, 2009)

- Intel Pentium Xeon: 603,860,840
- Intel Pentium Xeon MP: 603 versión gc-le
- Intel Pentium M: 479,478
- Intel Pentium D: lga 775,845 gv
- Intel Pentium 2Duo :775

2.3 CARACTERÍSTICAS DE LOS MICROPROCESADORES⁵⁰

Los microprocesadores se pueden diferenciar por sus características físicas y lógicas:

Características lógicas:

1. Longitud de la palabra procesada, esto es, número de bits procesados en el mismo ciclo de reloj.
2. Capacidad de acceso a la memoria o la cantidad de memoria que puede manejar.
3. Velocidad de ejecución de instrucciones, su velocidad de proceso.
4. Repertorio de instrucciones a nivel máquina que puede procesar.

Características físicas:

1. Retraso de propagación de la señal eléctrica: representa el tiempo que tarda la señal en tomar uno u otro valor dentro del circuito.
2. Disipación de potencia: Este valor indica el calor que genera el procesador al permanecer operativo.
3. Abanico de salida: es la cantidad de señales eléctricas que el microprocesador es capaz de manejar entre su circuitería interna y el sistema informático exterior al que se conecta.
4. Márgenes de ruido: indican la fiabilidad de que la señal eléctrica que contiene la información generada por el microprocesador al realizar sus operaciones se propague correctamente a través de sus circuitos, o esté corrompida por una señal proveniente del exterior.

⁵⁰ Mujica, Carlos. "Microprocesadores" 18 de Marzo, 2009. microprocesadores-darcy.blogspot.com (17 de Abril, 2009)

Una de las principales limitaciones de los actuales microprocesadores es el abanico de salida de señales debido al limitado número de pines de conexión con la placa principal que aquéllos pueden tener. El número de pines del microprocesador limita o permite el manejo de mayor o menor cantidad de señales desde y hacia el microprocesador, lo que facilita o perjudica su capacidad en el manejo de la información necesaria para realizar los diferentes procesos.⁵¹

2.4 MICROPROCESADORES MÁS UTILIZADOS⁵²

Entre los microprocesadores y arquitecturas más comunes en América y Europa, desde su aparición a la actualidad (2009) según Home Electronics se pueden mencionar las siguientes:

- AMD K5 (1996), K6 (1997), K6-2 (1998), K6-III (1999), Duron(2000 - 2006), Athlon(1999 - 2005), Athlon XP (2001), Athlon XP-M (arquitectura 32-bit de IA-32 Intel x86 - 2001)
- AMD Athlon 64 (2003), Athlon 64 FX (2003 – 2004), Athlon 64 X2 (2005) , Opteron (2003), Sempron (2004), Turion 64 (arquitectura AMD64 - 2005)
- Cyrix M1 (1996), M2 (arquitectura 32-bit de IA-32 Intel x86 - 1997)
- Intel 8080 (1974), 8085 (1977), Zilog Z80 (1976)
- Intel 8086 (1979 – 1982), 8088 (1979 – 1982), 80186 (1982), 80188 (modo verdadero 16-bit- solamente arquitectura de Intel x86 - 1982).
- Intel 80286 (modo verdadero 16-bit y arquitectura protegida de Intel x86 del modo – 1982 - 1986)
- Intel 80386 (1986 – 2007), 80486 (1989 – 2007), Pentium (1993), Celeron (1998), dúo de la base de Intel (2006), Pentium II (1997), Pentium III (1999), Xeon (1998), Pentium 4 (2000 – 2008), Pentium M (2003), Celeron M (2004)
- Pentium D (2005-2007), Celeron D (2005)

⁵¹ Mujica, Carlos. “Microprocesadores” 18 de Marzo, 2009. microprocesadores-darcy.blogspot.com (17 de Abril, 2009)

⁵² “Microprocesador”. www.buy-home-electronics.com (17 de Abril, 2009)

- Intel Itanium (arquitectura IA-64 – 2001 - 2002)
- Motorola 6800 (1975), Tecnología 6502 (1975), Motorola 6809 del MOS (1979)
- IBM (padres de la familia de PowerPC - 1993)
- Arquitectura de OpenCores OpenRISC (2001 – 2009)
- SPARC (1980 – 1982), UltraSPARC (1996), UltraSPARC II-IV (1996 -2006)

2.5 SIMULADORES PARA MICROPROCESADORES

Para Aprender a trabajar en la programación con lenguaje ensamblador existe diversidad de programas simuladores que permiten el correcto aprendizaje del lenguaje de programación; algunos de estos programas pueden ser gratis y otros necesitan licencia ofreciendo más ventajas sobre otros.

Se presenta a continuación una lista de algunos simuladores para lenguaje ensamblador que se pueden encontrar en Internet:

- **SimuProc⁵³**: SimuProc es un simulador de un procesador hipotético con el cual podrá aprender las nociones básicas para empezar a programar en lenguaje ensamblador, observando todo el proceso interno de ejecución del programa a través de cada ciclo del procesador. Dispone de más de 50 instrucciones (el 90% de un procesador Intel y otras abreviaturas de instrucciones), un editor con corrector de sintaxis, conversor de bases, conversión de números de punto flotante de decimal a IEEE 754 y viceversa, etc. Es ideal para aprender a programar en ensamblador, ya que este simulador muestra cómo funciona el procesador internamente y qué está haciendo en cada ciclo de reloj, permitiendo depurar los programas realizados y ver los errores con facilidad.

⁵³ Soriano, Noé. "SimuProc" 1 de abril, 2003. www.idg.es (17 de Abril, 2009)

- **gnusim8085⁵⁴**: GNUSim8085 es un simulador gráfico, ensamblador y depurador para el microprocesador 8085 de Intel especializado para sistemas Linux.
- **8085 Simulator⁵⁵**: tiene una interfaz muy amigable con el usuario y la mejor parte es que puede descargarse GRATIS. Ahora se puede dominar la programación del microprocesador 8085 en cuestión de días, con las completas ilustraciones gráficas del simulador. Despliega constantemente los contenidos de los lugares de memoria, incluye un ensamblador para la construcción que traslada el código escrito a código de máquina. Ofrece un fácil y extensivo depurador que incluye puntos de rotura y ejecución paso a paso. Provisto de rutinas de interrupción de memoria y otras subrutinas.
- **Sim8085⁵⁶**: Programa diseñado y desarrollado de tal forma que pueda ayudar a cualquier persona que quiera tomar algún curso o laboratorio del microprocesador 8085 como tema de estudio. Posee un editor: Se puede escribir código en un ambiente amigable al usuario con todas las herramientas que podrían ayudar. El editor posee todos comandos conocidos, además de poseer la capacidad de “marcador de sintaxis”. Un compilador en el cual se puede revisar el código escrito para el 8085 en sintaxis de lenguaje ensamblador. Mientras revisa sintaxis, se generan automáticamente las listas de código y codifica los archivos objetos si así se desea. Y un depurador el cual es la parte principal del programa; los programas que se escriben son convertido a código de máquina y guardados en una memoria virtual de 64KB. El depurador consiste en 5 partes internas: Código, Datos, una pila, registros y banderas.

⁵⁴ Ratnakumar, Sridhar. “gnusim8085”. gnusim8085.sourceforge.net (17 de Abril, 2009)

⁵⁵ Singla, Vaneet. “8085 Simulator” 10 de abril, 2004. www.softlookup.com (17 de Abril, 2009)

⁵⁶ Ved, Anurag. “Sim8085” 27 de noviembre, 2001. www.simtel.net (17 de Abril, 2009)

2.6 VENTAJAS Y DESVENTAJAS DE LOS SIMULADORES⁵⁷

La utilización de un simulador interactivo en el marco del constructivismo en la educación, para promover el desarrollo de habilidades y actitudes con los alumnos de la clase, independientemente del nivel escolar en el que se encuentra, permite ampliar las técnicas y herramientas tecnológicas que permiten un mayor aprendizaje en nuestros días.

Sin duda alguna que entre los tantos retos a resolver en este siglo en la educación destaca el gran reto de desarrollo e implementación de estrategias de enseñanza, dirigidas a desarrollar habilidades y actitudes en los estudiantes, con el fin de que adquieran distinciones y competencias que les permitan estar funcionalmente activos en lo profesional en las próximas 5 décadas del nuevo siglo y sean capaces de tomar decisiones que les lleven a resolver genuinamente los problemas, lo que demanda del alumno el ejercicio del principio de compartir los recursos y conocimientos de que dispongan, a través de la práctica de aprendizajes colaborativos.

Ventajas de los simuladores.

La Función de los simuladores: Están basados en aprendizaje de tipo experimental, y en conjeturas, para llevar a cabo un aprendizaje por descubrimiento. El alumno obtiene conocimiento por medio de la interacción con un micro-mundo logrando así simular una situación del mundo real, en la que el educando logrará controlar diferentes situaciones, y aprenderá a tomar las decisiones del caso.

La principal característica de los simulacros no es la de simular la realidad, sino la de proveer situaciones llamativas y entretenidas para el usuario, con lo que se logra el aprendizaje de un determinado contexto, dependiendo de la naturaleza del material.

⁵⁷ Loya, Elva. "Los Simuladores" 19 de marzo, 2007. eloyat.blogspot.com (17 de Abril, 2009)

- El estudiante es un agente activo, o sea que es el actor y fuente principal de aprendizaje, por lo que debe resolver la situación del micro-mundo que se le presenta, procesar la información, tomar decisiones, y obtener los resultados.
- El problema llega a ser un reto para el alumno, con lo que el material logra la fase de motivación.
- Se logra cumplir con los requerimientos de los sistemas de ejercitación y práctica en cuanto a practicar y afinar lo aprendido, por medio de vivencias.
- Le permite manipular al alumno un número reducido de variables, dentro del proceso simulado.
- Proporciona explicación sobre los principios involucrados y cómo son afectadas las variables, cuando se manipula cada una de ellas.
- Ofrece restricciones de operación, de forma que hace necesario que el alumno realice un esfuerzo para lograr las metas propuestas. La razón de ésta propuesta, es que de ésta manera el alumno se verá motivado a lograr los objetivos de operación propuestos.

Desventajas de los Simuladores.⁵⁸

- Si el profesor no está de acuerdo con la filosofía de este tipo de material, y cree que sus estudiantes no serán capaces de lograr lo compuesto, no se sacará provecho de este tipo de material.
- Hay una necesidad de conocer los objetivos o funciones para los que ha sido diseñado el software educativo por parte del docente para que pueda aplicarlo de manera adecuada y eficiente.
- En muchos de los casos de simuladores no se pueden medir de alguna manera los efectos producidos por el mensaje, de tal forma que dichos efectos no se puedan atribuir a otros factores diferentes. Es imprescindible que se pueda establecer de manera clara una relación causa/efecto entre lo aplicado en el simulador y el efecto producido en los alumnos.

⁵⁸ Loya, Elva. "Los Simuladores" 19 de marzo, 2007. eloyat.blogspot.com (17 de Abril, 2009)

- En muchas de las instituciones educativas, sobre todo las públicas, no cuentan con estas nuevas tecnologías por falta de recursos económicos y materiales para poder ser aplicados en todas las aulas.

2.7 COMPONENTES BÁSICOS DE LOS MICROPROCESADORES⁵⁹

2.7.1 Internos

2.7.1.1 Coprocesador Matemático

El coprocesador matemático está diseñado para que funcione en paralelo con el microprocesador. El conjunto de instrucciones incluye muchas operaciones extremadamente potentes en coma flotante.

Cuando el microprocesador encuentre una instrucción en coma flotante, envía el código de operación necesario y operaciones de dirección de memoria al coprocesador matemático. Esto libera al microprocesador de ejecutar la siguiente instrucción, mientras el coprocesador matemático realiza simultáneamente el cálculo numérico.

El coprocesador matemático puede hacer peticiones de acceso a memoria a través de un canal de datos dedicado permanente en el microprocesador.

Los coprocesadores fueron vistos por primera vez en los mainframes donde se añadían para funcionalidad opcional como el soporte matemático para punto flotante, otro uso muy común era para el control de los canales de Entrada/Salida, aunque este dispositivo se conocía normalmente como controlador de canal, un ejemplo de estos dispositivos lo tenemos en los controladores DMA.

⁵⁹ Girón, Manuel. "Coprocesador" 17 feb 2009. www. es. Wikipedia.org (15 abril de 2009)

Los coprocesadores también comenzaron a hacerse comunes en los ordenadores de escritorio a través del año de 1980 y a principios del año de 1990 debido a las limitaciones del diseño de la CPU y consideraciones de costo. El coprocesador matemático fue un elemento extra en común para los ordenadores de gama alta como el Macintosh II y muchas estaciones de trabajo que requerían capacidades de aritmética en coma flotante, pero hasta principios de los 90 la demanda de estos dispositivos, fue mínima. Otro coprocesador que empezó a ser común durante esta era fue el coprocesador gráfico, usado por la Familia Atari de 8 bits y el Commodore Amiga. El procesador gráfico en los Commodore era denominado habitualmente como "Copper".⁶⁰

Finalmente, el costo de los coprocesadores matemáticos fue lo suficientemente bajo para incluir en el microprocesador de propósito general (CPU) las funciones del coprocesador matemático eliminando, de este modo, los componentes separados. La demanda de coprocesador gráfico dedicado también ha crecido, sin embargo, debido a la alta demanda de procesamiento gráfico por parte de los videojuegos de ordenador que requieren gráficos 3D por computadora muy reales; este procesador dedicado elimina una considerable carga computacional a la CPU principal e incrementa el rendimiento en las aplicaciones gráficas intensivas. A partir del año 2000, las tarjetas gráficas con una Unidad de Procesado de Gráfico (GPU) son comunes. Las tarjetas de sonido actuales también vienen con un potente procesador incluido con extensiones multimedia para eliminar tiempo de cómputo en el procesador principal. En 2006, AGEIA anunció otra tarjeta de expansión para ordenadores llamada PhysX, este dispositivo tiene un procesador integrado diseñado para ejecutar computación de los aspectos físicos de los escenarios 3D liberando de esta carga a la CPU y GPU. Está diseñado para trabajar con videojuegos, pero teóricamente se podrán ejecutar otras tareas matemáticas en él.

⁶⁰ Girón, Manuel. "Coprocesador" 17 feb 2009. [www. es. Wikipedia.org](http://www.es.wikipedia.org) (15 abril de 2009)

Un coprocesador no es un procesador de propósito general, algunos coprocesadores no pueden ni siquiera leer las instrucciones desde la memoria sino que ejecutan flujo de instrucciones. Estos procesadores requieren de un procesador principal que lea las respuestas del coprocesador y maneje todas las operaciones junto con las funciones del procesador. En algunas arquitecturas el coprocesador es otro procesador de propósito general, pero que solamente ejecutará un rango de funciones limitadas por el procesador principal que le ejercerá el control. Nótese la diferencia de este modelo con los términos de un multiprocesador, que también tiene más de un microprocesador de propósito general.

2.7.1.2 Memoria Caché⁶¹

Una memoria caché es una memoria en la que se almacena una serie de datos para su rápido acceso. Básicamente, la memoria caché de un microprocesador es un tipo de memoria volátil (del tipo RAM), pero de una gran velocidad.

En la actualidad esta memoria está integrada en el procesador, y su cometido es almacenar una serie de instrucciones y datos a los que el procesador accede continuamente, con la finalidad de que estos accesos sean instantáneos. Estas instrucciones y datos son aquellas a las que el procesador necesita estar accediendo de forma continua, por lo que para el rendimiento del procesador es imprescindible que este acceso sea lo más rápido y fluido posible.

Hay tres tipos diferentes de memoria caché para procesadores:

1. Caché de 1er nivel (L1):

Esta caché está integrada en el núcleo del procesador, trabajando a la misma velocidad que este. La cantidad de memoria caché L1 varía de un procesador a otro,

⁶¹ Calderon, José "Qué es la memoria caché" 01 de Agosto 2007 .Configuraequipos.com(15 abril de 2009)

estando normalmente entre los 64KB y los 256KB. Esta memoria suele a su vez estar dividida en dos partes dedicadas, una para instrucciones y otra para datos.

2. Caché de 2do nivel (L2):

Integrada también en el procesador, aunque no directamente en el núcleo de este, tiene las mismas ventajas que la caché L1, aunque es algo más lenta que esta. La caché L2 suele ser mayor que la caché L1, pudiendo llegar a superar los 2MB.

A diferencia de la caché L1, esta no está dividida, y su utilización está más encaminada a programas que al sistema.

3. Caché de 3er nivel (L3):

Es un tipo de memoria caché más lenta que la L2, muy poco utilizada en la actualidad. En un principio esta caché estaba incorporada a la placa base, no al procesador, y su velocidad de acceso era bastante más lenta que una caché de nivel 2 o 1, ya que si bien sigue siendo una memoria de una gran rapidez (muy superior a la RAM, y mucho más en la época en la que se utilizaba), depende de la comunicación entre el procesador y la placa base.⁶²

Para hacerse una idea más precisa de esto, se debe imaginar en un extremo el procesador y en el otro la memoria RAM. Pues bien, entre ambos se encuentra la memoria caché, más rápida en cuanto más cerca se encuentre del núcleo del procesador (L1).

Las memorias caché son extremadamente rápidas (su velocidad es unas 5 veces superior a la de una RAM de las más rápidas), con la ventaja añadida de no tener latencia, por lo que su acceso no tiene ninguna demora pero es un tipo de memoria muy cara.

⁶² Calderon, José "Qué es la memoria caché "01 de Agosto 2007 .Configuraequipos.com(15 abril de 2009)

Esto, unido a su integración en el procesador (ya sea directamente en el núcleo o no) limita bastante el tamaño, por un lado por lo que encarece al procesador y por otro por el espacio disponible.

En cuanto a la utilización de la caché L2 en procesadores multinucleares, existen dos tipos diferentes de tecnologías a aplicar.

Por un lado está la habitualmente utilizada por Intel, que consiste en que el total de la caché L2 está accesible para ambos núcleos y por otro está la utilizada por AMD, en la que cada núcleo tiene su propia caché L2 dedicada solo para ese núcleo. La caché L2 apareció por primera vez en los Intel Pentium Pro, siendo incorporada a continuación por los Intel Pentium II, aunque en ese caso no en el encapsulado del procesador, sino externamente (aunque dentro del procesador).

2.7.1.3 Encapsulado⁶³

La comunicación de un microprocesador con el exterior, esto es, con la memoria principal y con las unidades de control de los periféricos, se realiza mediante señales de información y señales de control que son enviadas a través del patillaje del microprocesador. Posteriormente, estas señales viajarán por el bus del sistema que comunica al procesador con los demás componentes situados en la placa base, pasando a continuación al bus de E/S hasta llegar al periférico correspondiente. El número y tamaño de las pines han ido variando con el tiempo según las necesidades y las tecnologías utilizadas. Para comunicarse con el resto del sistema informático el procesador utiliza las líneas de comunicación a través de sus pines (*pines*). Se define como encapsulado la forma en que se empaqueta la oblea de silicio para efectuar su conexión con el sistema.

⁶³ Diego. "Encapsulado de un microprocesador" 13 dic 2008. es.Wikipedia.org (16 abril de 2,009)

Encapsulados más importantes:

❖ Dual in-line package.

DIP, o Dual in-line package por sus siglas en inglés, es una forma de encapsulamiento común en la construcción de circuitos integrados. La forma consiste en un bloque con dos hileras paralelas de pines, la cantidad de éstos depende de cada circuito. Por la posición y el espacio entre pines, los circuitos DIP son especialmente prácticos para construir prototipos en tablillas de protoboard. Concretamente, la separación estándar entre dos pines o terminales es de 0.1“(2.54 mm).

La nomenclatura normal para designarlos es DIP n , donde n es el número de pines totales del circuito. Por ejemplo, un circuito integrado DIP16 tiene 16 pines, con 8 en cada fila.

Dada la actual tendencia a tener circuitos con un nivel cada vez más alto de integración, los paquetes DIP han sido sustituidos por encapsulados SMD (Superficial mounted device) o de montaje superficial. Estos últimos tienen un diseño mucho más adecuado para circuitos con un alto número de pines, mientras que los DIP raras veces se encuentran en presentaciones de más de 40 pines.⁶⁴

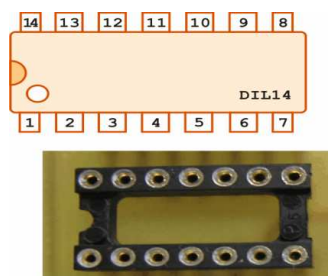


Fig. 2.14 -Dual in-line package

⁶⁴ Diego. "Encapsulado de un microprocesador" 13 dic. 2008. es.Wikipedia.org (16 abril de 2,009)

❖ **Pin grid array**⁶⁵

El Pin grid array o PGA es un tipo de empaquetado usado para los circuitos integrados, particularmente microprocesadores.

Originalmente el PGA, el zócalo clásico para la inserción en una placa base de un microprocesador, fue usado para procesadores como el 80386 y el 80486; consiste en un cuadrado de conectores en forma de agujero donde se insertan las patitas del chip por pura presión. Según el chip, tiene más o menos agujeros (uno por cada patita).

En un PGA, el circuito integrado (CI) se monta en una losa de cerámica de la cual una cara se cubre total o parcialmente de un conjunto ordenado de pines de metal. Luego, los pines se pueden insertar en los agujeros de un circuito impreso y soldados. Casi siempre se espacian 2.54 milímetros entre sí. Para un número dado de pines, este tipo de paquete ocupa menos espacio los tipos más viejos como el Dual in-line package (DIL o DIP).

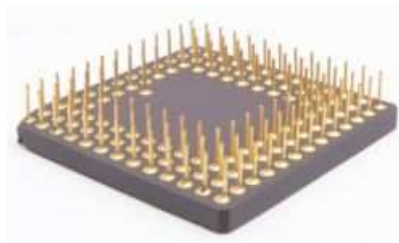


Fig. 2.15- Pin grid array.

❖ **Quad Flat Package**

Un encapsulado **Quad Flat Package** (QFP o encapsulado cuadrado plano) es un encapsulado de circuito integrado para montaje superficial con los conectores de

⁶⁵ Diego. "Encapsulado de un microprocesador" 13 dic 2008. es.Wikipedia.org (16 abril de 2,009)

componentes extendiéndose por los cuatro lados. Los pines se numeran en sentido contrario a las agujas del reloj a partir del punto guía.

QFP utiliza habitualmente de 44 a 200 pines, con una separación entre ellos de 0,4 a 1 mm. Esto es una mejora respecto del encapsulado Small-Outline Integrated Circuit (SOP o SOIC) pues permite una mayor densidad de pines y utiliza las cuatro caras del chip (en lugar de solo dos). Para un número de pines mayor se utiliza la técnica Ball grid array (BGA) que permite usar toda la superficie inferior.

El antecesor directo de QFP es Plastic leaded chip carrier (PLCC), que utiliza una distancia entre pines mayor 1.27 mm (50 milésimas de pulgada, a veces abreviada mil) y una mayor altura del encapsulado.



Fig. 2.16 - Quad Flat Package

❖ **Low-profile Quad Flat Package⁶⁶**

Un encapsulado Low-profile Quad Flat Package (LQFP o encapsulado cuadrado plano de perfil bajo) es un encapsulado de circuito integrado para montaje superficial con los conectores de componentes extendiéndose por los cuatro lados. Los pines se enumeran en sentido contrario a las agujas del reloj a partir del punto guía. El espacio entre pines puede variar; los intervalos más comunes son 0.4, 0.5, 0.65 y 0.80 mm.

⁶⁶ Diego. "Encapsulado de un microprocesador" 13 dic 2008. es.Wikipedia.org (16 abril de 2,009)



Fig. 2.17- Low-profile Quad Flat Package

❖ **Plastic leaded chip carrier⁶⁷**

Un Plastic Leaded Chip Carrier (PLCC), también llamado Quad-Flat-J-Leg Chipcarrier (QFJ) es un encapsulado de circuito integrado con un espaciado de pines de 1,27 mm (0,05 pulgadas). El número de pines oscila entre 20 y 84. Los encapsulados PLCC pueden ser cuadrados o rectangulares. El ancho oscila entre 0,35 y 1,15 pulgadas. PLCC es un estándar JEDEC. Las configuraciones PLCC requieren menos espacio en placa que sus competidores los leadless chip carrier (similares a los encapsulados DIP pero con *bolitas* en lugar de pines en cada conector).

Un dispositivo PLCC puede utilizarse tanto para montaje en superficie como para instalarlo en un zócalo PLCC. A su vez los zócalos PLCC pueden montarse en superficie o mediante through-hole (perforaciones en la placa con borde metalizado). La causa de usar un zócalo montado en superficie puede ser que el chip no soporte el calor generado durante el proceso.



Fig. 2.18- Plastic leaded chip carrier

⁶⁷ Diego. "Encapsulado de un microprocesador" 13 dic 2008. es.Wikipedia.org (16 abril de 2,009)

2.7.1.4 Interrupciones⁶⁸

Una interrupción se puede definir como una señal proveniente de un dispositivo externo, que llega a una entrada del microprocesador dedicada a este propósito y que indica al microprocesador que el dispositivo que la originó está solicitando servicio.

Cuando ocurre una interrupción, el microprocesador suspende temporalmente la ejecución del programa principal y transfiere el control a una subrutina especialmente diseñada para atender al dispositivo que provocó la interrupción, a esta subrutina se le denomina subrutina de servicio de la interrupción (interrupt service routine) o manejador de interrupción (interrupt handler). Al terminar el servicio, el microprocesador regresa al programa principal, continuando con sus actividades normales. Desde este punto de vista una interrupción es esencialmente una llamada a subrutina iniciada por un circuito externo (hardware) y con un retorno controlado por programa (software). Además, las interrupciones tienen la característica de que son eventos asíncronos, es decir, pueden ocurrir en cualquier momento durante la ejecución del programa principal sin posibilidad de hacer una predicción exacta. Por ello hay que hacer consideraciones especiales y tener cuidados adicionales al manejarlas.

2.7.1.5 Instrucciones

2.7.1.5.1 ¿Qué Es Una Instrucción?⁶⁹

- La instrucción es el elemento clave del equipo, ya que permite indicar al procesador la acción que debe realizar. Las instrucciones a realizar se indican en el archivo fuente y el equipo pasa de una instrucción a otra siguiendo las

⁶⁸ Hueso Zavala, Jaime. "Interrupciones" 15 febrero de 2000. Academia de Sistemas Digitales (16 abril de 2,009)

⁶⁹ "Introducción a los Microprocesadores" 19 Febrero, 2002. canalhanoi.iespana.es (16 abril de 2,009)

instrucciones que se indican de arriba hacia abajo (ya que la lectura de un archivo se realiza en una secuencia de arriba hacia abajo).

- Instrucción en informática, se le llama a un conjunto de datos insertados en una secuencia estructurada o específica que el procesador interpreta y ejecuta.

Los tipos de instrucción permitidos están definidos y determinados dentro de cada plataforma en la Arquitectura del Repertorio de Instrucciones (ARI) (**ISA**, *instruction set architecture*), que también determina los registros de origen y destino de la CPU, y en ocasiones un dato inmediato (aquellos que son especificados explícitamente en la instrucción).

Se conoce como set de instrucciones al conjunto de instrucciones que es capaz de entender y ejecutar un microprocesador.

En función del tipo de microprocesador, concretamente si es más avanzado o no, podrá entender y ejecutar más o menos instrucciones.

2.7.1.5.2 Formato de las Instrucciones⁷⁰

Las líneas de instrucciones se integran de 4 campos.

- ❖ Identificador: Se utiliza para etiquetar a las instrucciones donde se realizara el salto, nombre de un elemento o directiva definida.
- ❖ Operación: Instrucción específica o definición de área de datos.
- ❖ Operandos: Objeto u objetos sobre los cuales se ejecutará el nemónico, si son más de uno se separan con comas.
- ❖ Comentario: Contiene documentación sobre la instrucción y se inician con punto y coma (;)

Ejemplo:

ORD: SUB AX, CX

⁷⁰ "Introducción a los Microprocesadores" 19 Febrero, 2002. canalhanoi.iespana.es (16 abril de 2,009)

JZ GCD; Instrucción de salto utilizando bandera cero

JNS NXT; Instrucción de salto utilizando bandera de signo

2.7.1.5.3 Tipos De Instrucciones⁷¹

Las instrucciones se clasifican según su función en:

- **Instrucciones de transferencia de datos.** Estas instrucciones mueven datos (que se consideran elementos de entrada/salida) desde la memoria hacia los registros internos del microprocesador, y viceversa. También se usan para pasar datos de un registro a otro del microprocesador. Existen algunas instrucciones que permiten mover no sólo un dato, sino un conjunto de hasta 64 Kbyte con una sola instrucción.
- **Instrucciones de cálculo.** Son instrucciones destinadas a ejecutar ciertas operaciones aritméticas, como por ejemplo sumar, restar, multiplicar o dividir, o ciertas operaciones lógicas, como por ejemplo AND, OR, así como desplazamiento y rotación de bits.
- **Instrucciones de transferencia del control del programa.** Permiten romper la secuencia lineal del programa y saltar a otro punto del mismo. Pueden equivaler a la instrucción **GOTO** que traen muchos lenguajes de programación.
- **Instrucciones de control.** Son instrucciones especiales o de control que actúan sobre el propio microprocesador. Permiten acceder a diversas funciones, como por ejemplo activar o desactivar las interrupciones, pasar órdenes al coprocesador matemático, detener la actividad del microprocesador hasta que se produzca una interrupción, etc.

Prácticamente todas las instrucciones están formadas por dos elementos:

- **Código de operación:** que indica el tipo de operación se va a realizar.
- **Operandos:** que son los datos sobre los que actúa.

⁷¹ "Introducción a los Microprocesadores" 19 Febrero, 2002. canalhanoi.iespana.es (16 abril de 2,009)

Por ejemplo, una instrucción que sume dos números está formado por:

- código de operación que indique "sumar"
- primer número a sumar
- segundo número a sumar

Existen instrucciones que sólo tienen un operando o incluso que no tienen ninguno, estando formadas solamente por el código de operación.

2.8 COMPONENTES ADICIONALES⁷²

2.8.1 Arquitectura Y Organización De Un Microprocesador

Arquitectura: La arquitectura se refiere a los atributos de un sistema que son visibles para un programador o bien, a aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa.

Organización: La organización se refiere a las unidades funcionales y sus interconexiones, que dan lugar a especificaciones arquitectónicas.

Lo interesante de estos conceptos es que para el diseño de un microprocesador es necesario involucrarse en su arquitectura (conjunto de instrucciones, número de bits para representar números, instrucciones y caracteres, etc.) y en su organización (señales de control, memoria, registros, etc.). De hecho, en la actualidad muchos microprocesadores han conservado su arquitectura de una manera u otra pero su organización ha cambiado para mejorar su rendimiento. En las mismas compañías que ofrecen microprocesadores la arquitectura es la misma o muy similar con diferencia únicamente en la organización lo que hace variar precios y rendimiento.

A continuación se detalla la organización del microprocesador:

⁷² Jiménez Alaniz, Juan Ramón." Aplicaciones de Microprocesadores e Interfaces". Universidad Autónoma Metropolitana (16 abril de 2,009)

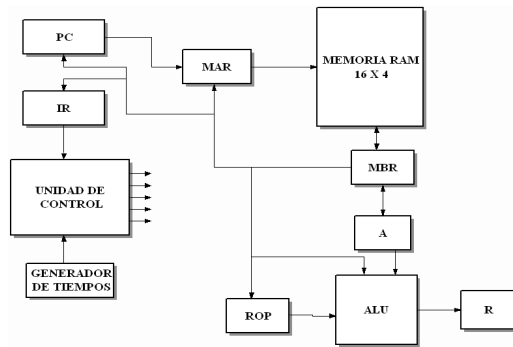


Fig. 2.19- Organización de un microprocesador sencillo.⁷³

2.8.2 Puertos De Entrada/Salida⁷⁴

Cualquier dispositivo que intercambia datos con el sistema lo hace a través de un "puerto". Un puerto es una interfaz con ciertas características. Como los dispositivos E/S tienen algún tipo de conexión con el bus externo, deben tener una dirección (o conjunto de ellas) que los identifique.

Estos son una clase de puertos por donde el CPU puede enviar y recibir información. Son muy importantes, ya que entre estos se encuentran las memorias del CPU como son la RAM, ROM, los floppys y discos duros. Estos puertos pueden ser usados para controlar dispositivos, tales como las impresoras y los quemadores externos. Existen infinidad de dispositivos que sirven de extensión a la computadora, muchos son para fines muy específicos y no se pueden abarcar, entre los dispositivos que son de uso común se encuentra la impresora, el teclado, el Mouse y el monitor.

Puerto Serial

El puerto serial es aquel que envía y recibe los datos BIT por BIT, entre los puertos seriales se puede mencionar el puerto de teclado, o el puerto del MODEM.

⁷³ Jiménez Alaniz, Juan Ramón." Aplicaciones de Microprocesadores e Interfaces".Universidad Autónoma Metropolitana (16 abril de 2,009)

⁷⁴ "Arquitectura de Computadoras " 09 de enero de 2009 .Instituto Tecnológico del Istmo (16 abril de 2,009)

Puerto Paralelo

Este tipo de puerto transmite la información byte por byte, o sea que transmite ocho bits al mismo tiempo, de forma paralela. Un puerto paralelo por excelencia pues es el puerto para impresora.

Se puede observar que un puerto de entrada puede ser paralelo o serial, lo mismo que un puerto de entrada o de entrada/salida.

A cada puerto la BIOS le asigna una dirección de memoria para que pueda trabajar, dependiendo de qué clase de puerto se le asigna un determinado espacio exclusivo para él.

Por medio de estas localidades de memoria el sistema puede enviar o recibir información, es una especie de memoria de intercambio para la transmisión de datos de un lugar a otro. A esto se le llama espacio de localidades de memoria y se realiza dentro del primer kilobyte de la memoria principal. Existen otras asignaciones de memoria en otras capas superiores de memoria pero estas son hechas por el sistema operativo y por los demás programas, pero estas asignaciones son para fines específicos de los demás programas.

Los puertos no solo se limitan a recibir la información, o enviarla, según sea el caso. El puerto provee la corriente eléctrica necesaria para el funcionamiento del dispositivo y revisa el estado en el que se encuentra este.⁷⁵

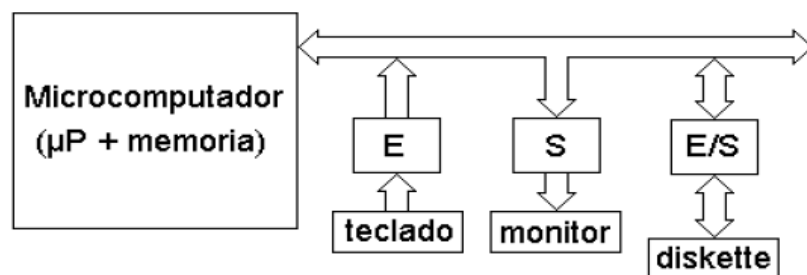


Fig. 2.20- Puertos de Entrada/Salida

⁷⁵ "Arquitectura de Computadoras" 09 de enero de 2009. Instituto Tecnológico del Istmo (16 abril de 2,009)

2.8.3 Buses

Los buses son los encargados de llevar los datos de un lugar a otro en el computador.

2.8.3.1 Tipos De Buses

En forma muy general existen tres tipos de buses, de acuerdo a la función que realizan.

Bus De Datos

Sirve para transmitir información entre el microprocesador, la memoria y los periféricos. Por él circulan los datos y las instrucciones. Tiene tantas líneas como bits se transmiten en paralelo (una para cada bit). El flujo es de doble sentido y a mayor número de bits paralelos mayor podrá ser la velocidad de transmisión que se obtiene.

El ancho de este bus (Nº de bits que puede transmitir simultáneamente en paralelo) es una medida de la potencia del microprocesador. Este bus es como una autopista de datos electrónicos y cuanto más ancho sea, más datos podrán moverse al mismo tiempo.

El ancho del bus de datos es una de las características más importantes del microprocesador. Cuando se dice que un microprocesadores, por ej., de 16 bits, se refiere al ancho de su bus de datos.⁷⁶

Bus De Direcciones

Es unidireccional, por el solamente circula información proveniente del microprocesador. Comprende a las líneas que transmiten una dirección generada por el CPU, la cual selecciona a un puerto o a una localidad de memoria.

⁷⁶ "Arquitectura de Computadoras" 09 de enero de 2009. Instituto Tecnológico del Istmo (16 abril de 2,009)

Bus De Control

Lo conforman la sincronización y el sentido de transferencia de información en el bus de datos, y el tipo de transferencia indicada por medio de señales de control originadas en el CPU. Cada una de las señales en el bus de control es unidireccional.

Buses Internos

Son buses que existen dentro del microprocesador que sirven para comunicar entre sí a la ALU, los registros internos y la unidad de control.

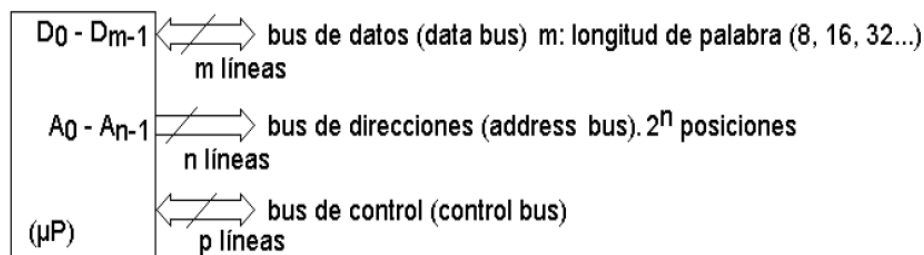


Fig. 2.21- Buses Internos del microprocesador.

2.8.4 Memorias⁷⁷

Memoria RAM

(RAM: Random Access Memory). Contiene los programas y los datos con los que el ordenador está trabajando en un momento determinado. Se denomina memoria de acceso aleatoria, pero en realidad es ésta una denominación un tanto confusa. La RAM es una memoria de acceso directo y de carácter efímero, puesto que su contenido se borra cuando se apaga el ordenador.

⁷⁷ "Memoria RAM" 26 de Agosto de 2008 .www.Redcoon.com(16 de abril de 2,009)

Memoria ROM

No volátil. Almacena programas de forma permanente. Sólo lectura.

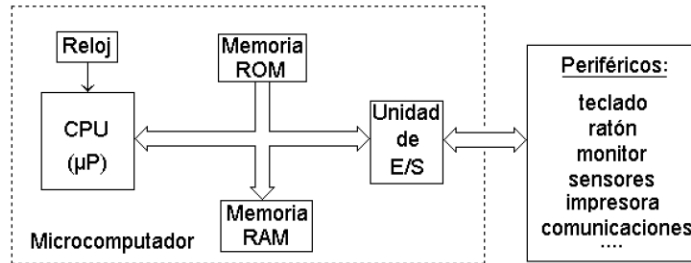


Fig. 2.22- Diagrama de Memoria ROM

2.8.5 Placa Base⁷⁸

La placa base, también conocida como placa principal, placa madre, mainboard, motherboard, etc. Es uno de los componentes principales y esenciales de toda computadora, en el que se encuentran alojados, montados o conectados todos los demás componentes y dispositivos de la PC.

Físicamente, se trata de una “oblea” de material sintético, sobre la cual existe un circuito electrónico que conecta diversos componentes que se encuentran insertados o montados sobre la misma, los principales son:

▶	El microprocesador: (CPU) el cerebro de la PC, montado sobre un elemento llamado zócalo.
▶	La memoria principal: montados sobre las ranuras de memoria llamados generalmente bancos de memoria.
▶	Las ranuras de expansión: o slots donde se conectan las placas de video, sonido, módem, etc.
▶	Diversos chips de control: como ser: la BIOS, el Chipset, controladoras, etc.

Tabla 2.1. Elementos de la placa base

⁷⁸ Morales, Marco. “Placa Base”. 2009. www.Cespro.com (17 abril de 2009)

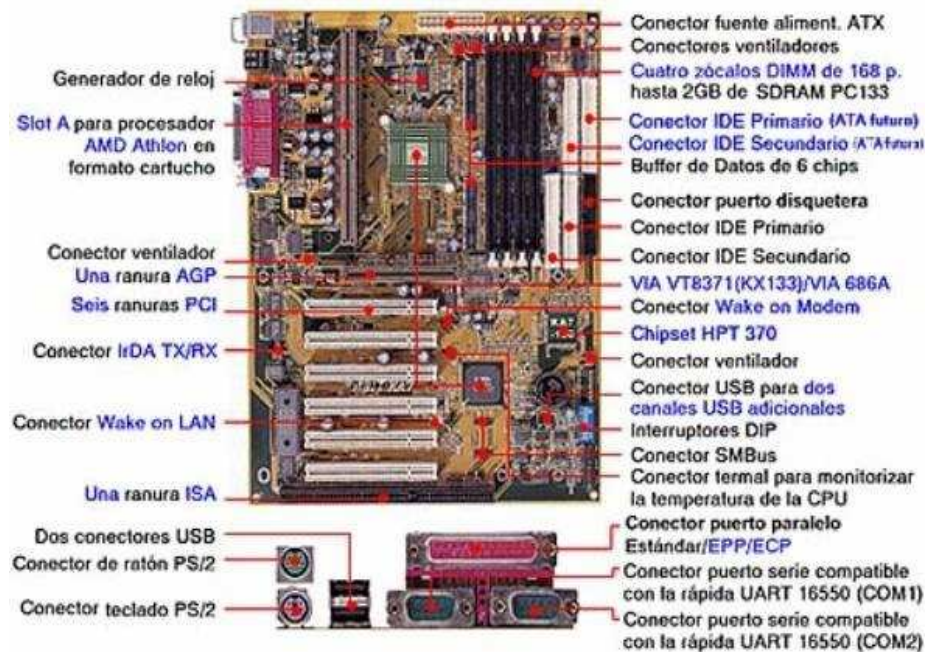


Fig. 2.23- Partes de Placa Base⁷⁹

2.8.5.2 Tipos De Placa Base

Existen diferentes formas de placa base y con diversos conectores para dispositivos, periféricos, etc. Los fabricantes han ido definiendo varios estándares que congregan recomendaciones sobre su tamaño y la disposición de los componentes sobre ellas.

De cualquier modo y teóricamente la forma y categoría de placa no tiene nada que ver con sus prestaciones ni mucho menos con su calidad.

Seleccionando una Motherboard

Una vez usted ha escogido una CPU, es tiempo para seleccionar una motherboard. La motherboard es uno de los componentes más importantes en cualquier PC, y hay varias consideraciones para tener en cuenta al seleccionar una.

⁷⁹ Morales, Marco. "Placa Base". 2009. www.Cespro.com (17 abril de 2009)

La motherboard y la CPU son los componentes más importantes de su computadora. Su motherboard proporciona las conexiones a todos los otros componentes que deben unirse a esta. La motherboard y la CPU sirven como las principales fuentes computacionales y de procesamiento de su PC.⁸⁰

ATX

Cada vez fueron más comunes y difundidas en el mercado. Sus principales diferencias con las AT son las de más fácil ventilación y menos maraña de cables, debido a la colocación de los conectores ya que el microprocesador suele colocarse cerca del ventilador de la fuente de alimentación y los conectores para discos cerca de los extremos de la placa, los conectores suelen ser más (por ejemplo, con USB o con FireWire), están agrupados y tienen el teclado y ratón en clavijas mini-DIN. Además, reciben la electricidad mediante un conector formado por una sola pieza.

AT ó Baby-AT

Fue el estándar durante años. Define una placa de unos 22 x33 cm., con unas posiciones determinadas para el conector del teclado, los slots de expansión y los orificios de anclaje al gabinete, así como un conector eléctrico dividido en dos piezas a diferencia de las ATX que está formado por una sola pieza mencionado anteriormente.

Estas placas eran y suelen ser las típicas de las computadoras "clónicas" desde el 286 hasta los primeros Pentium. Con el apogeo de los periféricos como las tarjeta sonido, CD-ROM, discos extraíbles, etc. salieron a la luz sus principales defectos: mala circulación del aire en los gabinetes (que fue más tarde uno de los motivos de la aparición de disipadores y ventiladores de chip) y, sobre todo, una maraña enorme de cables que impide acceder a la placa sin desmontar al menos alguno.

⁸⁰ Morales, Marco. "Placa Base" 2009. www.Cespro.com (17 abril de 2009)

LPX o Desktop ⁸¹

Estas placas son de tamaño similar a las AT, aunque con la peculiaridad de que las ranuras para las placas o tarjetas de expansión no se encuentran sobre la placa base, sino en un conector especial en el que están montadas, la Riser Card.

De esta forma, una vez montadas, las tarjetas quedan paralelas a la placa base, en vez de perpendiculares como en las AT, es un diseño típico de las computadoras de escritorio con un gabinete pequeño y horizontal con menos de 15 cm. de alto y más de 30 cm. de ancho, y el monitor se encuentra sobre el mismo y no generalmente a un costado como las AT y su único inconveniente es que la Riser Card no suele tener más de dos ó tres ranuras de expansión, contra cuatro ó cinco en una AT típica.

En Resumen los Componentes De La Placa Base son:

- Zócalo del microprocesador.
- Ranuras de memoria (SIMM, DIMM, RIMM, etc.)
- Chipset de control.
- BIOS.
- Ranuras de expansión (ISA, PCI, AGP, etc.)
- Memoria caché.
- Conectores internos.
- Conectores externos.
- Conector eléctrico.
- Pila.
- Componentes integrados.

⁸¹ Morales, Marco. "Placa Base" 2009. www.Cespro.com (17 abril de 2009)

2.8.5.4 ¿Qué Es Zócalo Del Microprocesador?

Es el lugar donde se inserta el microprocesador de la computadora. Siempre ha consistido en un cuadrado donde el microprocesador se introduce con mayor o menor facilidad, recientemente, la aparición de los Pentium II ha cambiado un poco esta perspectiva.

Tipos De Zócalo

- **PGA:** fueron usados en el 386 y el 486; consisten en un cuadrado de conectores en forma de orificios muy pequeños donde se insertan los pines o patas del chip a presión. Según el tipo de chip, tiene más o menos orificios.⁸²
- **ZIF:** (Zero Insertion Force – Cero fuerza de inserción) Eléctricamente es como un PGA, con la diferencia de que posee un sistema mecánico que permite introducir el micro sin necesidad de presión alguna eliminando el peligro de dañar el chip tanto al introducirlo como extraerlo del zócalo.

Surgió en la época del 486 y sus distintas versiones (Socket's 3, 5 y 7) principalmente se han utilizado hasta que apareció el Pentium II. Se fabricaron algunos tipos de zócalos ZIF:

- ***Súper Socket 7:*** variante del Socket 7 que se caracteriza por poder usar velocidades de bus de hasta 100 MHz, que es el que utilizan los micros AMD K6-2.
- ***Socket 370 ó PGA 370:*** físicamente similar al anterior, pero incompatible con él por utilizar un bus distinto. Dos versiones: PPGA (la más antigua, sólo para micros Intel Celeron) y FC-PGA (para Celeron y los más recientes Pentium III).
- ***Socket A:*** utilizado únicamente por algunos AMD K7 Athlon y por los AMD Duron.

⁸² Morales, Marco. "Placa Base" 2009. www.Cespro.com (17 abril de 2009)

- **El Socket 775** de Intel es otro de los zócalos para dar soporte a los microprocesadores Pentium 4; debido precisamente a la cantidad de zócalos disponibles. Este viene a sustituir el socket 478. Actualmente se considera el Zócalo 775 para pentium 4 como el del presente y se pueden encontrar placas madres (motherboards) con este zócalo, con soporte para memoria RAM del tipo DDR2 y las nuevas ranuras de expansión PCI Express. Este tipo de zocalo es el "estandar", para casi todos los procesadores de consumo de "INTEL" para equipos sobremesa, y algunos portátiles. En la actualidad, desde los "Celeron D", hasta los "Core 2 Duo", pasando por los "Pentium D", su principal atractivo, es que los procesadores para socket 775 carecen de pines, es decir que la motherboard es la que contiene los contactos para comunicarse con el procesador, con esto se consigue que los procesadores sean menos fragiles a nivel físico. Los procesadores se "anclan" a la placa base con una pletina metálica, que los fuerza sobre los pines. Las velocidades de bus disponibles para esta arquitectura andan desde los 533Mhz hasta los 1600MHz.
- **Slot 1:** es un invento de Intel para montar los Pentium II, III. Físicamente muy distinto al anterior. En vez de ser un cuadrado con orificios para las patitas del chip, es una ranura muy similar a un conector ISA o PCI que no tiene muchas ventajas frente a los ZIF o PGA e incluso puede que al estar los conectores en forma de "peine" provoquen más interferencias.
- **Slot A:** la respuesta de AMD al Slot 1; físicamente ambos "slots" son idénticos, pero lógica y eléctricamente son totalmente incompatibles ya que Intel no tuvo ninguna intención de vender la idea y es utilizado únicamente por el AMD K7 Athlon.⁸³
- **Otros:** en ocasiones, no existe zócalo en absoluto, sino que el chip está soldado a la placa, en cuyo caso a veces resulta hasta difícil de reconocer. Es el caso de muchos 8086, 286 y 386SX ó bien se trata de chips antiguos como los 8086 ó 286, que tienen forma rectangular alargada parecida al del chip de la BIOS y

⁸³ Morales, Marco. "Placa Base" 2009. www.Cespro.com (17 abril de 2009)

pinos ó patitas planas en vez de redondas, en este caso, el zócalo es rectangular, del mismo modelo que se usa para multitud de chips electrónicos de todo tipo. ⁸⁴

2.9 APLICACIONES DE LOS MICROPROCESADORES⁸⁵

A continuación se enumeran algunos ejemplos de aplicaciones que incluyen microprocesadores en el campo de la medicina.

Microprocesador de genes:

De aplicación en un futuro cercano, sirve para predecir como actuarán los fármacos en cada paciente, para ello incluye su perfil genético. Consiste en una placa de vidrio de aproximadamente un centímetro cuadrado que contiene secuencias de ADN que se comparan con miles de fragmentos individuales de ADN de unos genes concretos. Sin duda este avance optimizará la eficiencia terapéutica de los medicamentos.

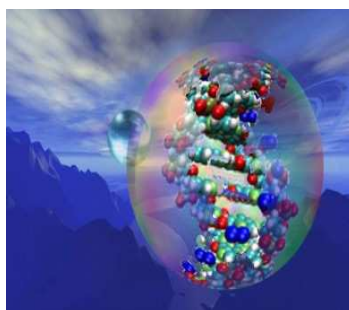


Fig. 2.24- Microprocesador de genes

⁸⁴ Morales, Marco. "Placa Base" 2009. www.Cespro.com (17 abril de 2009)

⁸⁵ Álvarez Suárez, Pablo. "Avances de la Electrónica en la Medicina" 16 enero de 2009 .(17 abril de 2009

Artificial Silicón Retina⁸⁶:

Se trata de un microprocesador que permite a los ciegos volver a percibir de nuevo la luz y distinguir formas. Del tamaño la cabeza de un alfiler, este dispositivo está compuesto de 3500 fotopilas que convierten la luz en señales eléctricas que son enviadas al cerebro por el nervio óptico. Pero aun se desconoce su fiabilidad a largo plazo.



Fig. 2.25- Artificial Silicón Retina

Robótica:

Ha permitido que la medicina se desarrolle con mayor precisión, convirtiéndose en un elemento indispensable en los quirófanos, ya que las manos de los cirujanos pueden ser sujetadas por brazos robóticos y así realizar incisiones con la mayor precisión posible.



Fig. 2.26- Robótica

⁸⁶ Álvarez Suárez, Pablo. "Avances de la Electrónica en la Medicina" 16 enero de 2009 .(17 abril de 2009

Nanotecnología:

Uno de los avances que más agradecen médicos y pacientes ya que en muchos casos sustituyan a la cirugía o por lo menos tratamientos mucho más invasivos. Un ejemplo es la píldora endoscópica, que es del tamaño de una pastilla y posee una micro cámara. Ésta es tragada por el paciente para someterse a una endoscopia, así se evitan dolorosas penetraciones por sondas⁸⁷.



Fig. 2.27- Nanotecnología

Diseño electrónico:

Podría considerarse como el conjunto de todas las demás. Su finalidad es desarrollar, innovar e investigar. Gracias a I+D se han obtenido biochips capaces de regular fluidos vitales.



Fig. 2.28- Diseño electrónico

⁸⁷ Álvarez Suárez, Pablo. "Avances de la Electrónica en la Medicina" 16 enero de 2009 .(17 abril de 2009

Máquinas industriales⁸⁸:

Este campo ha desarrollado exitosamente aparatos de monitoreo y herramientas auxiliares de quirófano, como aparatos de diálisis o bombeo renal.

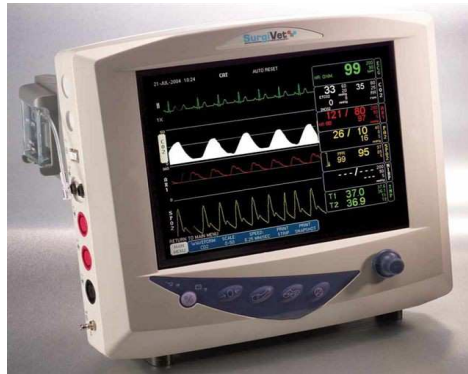


Fig. 2.29- Máquinas industriales

Aparatos de medición y análisis de pruebas:

Con el fin de que los pacientes puedan seguir su propio monitoreo estos aparatos portátiles tienen un programa informático de manejo sencillo para cualquier paciente.

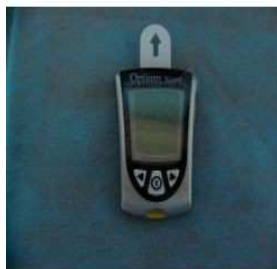


Fig. 2.30- Aparato de medición y análisis de pruebas

⁸⁸ Álvarez Suárez, Pablo. "Avances de la Electrónica en la Medicina" 16 enero de 2009 .(17 abril de 2009

Comunicaciones:

La telemedicina es uno de los aspectos más revolucionarios de la medicina, pues no imaginamos un médico sin ordenador en el que consultar la historia clínica de un modo automático. Esta técnica consiste en la transferencia de información a través de redes.

Software y programación:

En lo últimos tiempos se han desarrollado programas informáticos que mediante computadoras interpretan exámenes médicos en 3 dimensiones, como resonancias o encefalogramas.

Marcapasos⁸⁹:

Es un dispositivo electrónico que hace latir el corazón aplicando impulsos eléctricos que controlan el ritmo de los latidos, tiene una batería de litio que dura más de 10 años, su reducido tamaño y poco peso permite implantarlo bajo la piel.

Endoscopio

Consiste en una minúscula lamparita situada en el extremo de un delgado alambre de fibra óptica. Esta técnica permite realizar cualquier intervención quirúrgica demasiado pequeña para la vista humana minimizando la agresividad hacia el paciente, ya que no son necesarias incisiones tan grandes.



Fig. 2.31- Endoscopio

⁸⁹ Álvarez Suárez, Pablo. "Avances de la Electrónica en la Medicina" 16 enero de 2009 .(17 abril de 2009

2.10 ELECCIÓN DE UN MICROPROCESADOR

Criterios de selección

Cuando se desea preparar un sistema de computo la primera decisión es cual Microprocesador usar, para ello se utilizara la clasificación hecha por los fabricantes de Microprocesadores más reconocidos Intel y AMD, ellos han dividido sus procesadores en dos tipos.

- Procesadores de Alto Rendimiento
- Procesadores de valor

Porque se debe comprar una computadora de alto rendimiento con respecto a una de valor.

PC de valor	PC de alto rendimiento
<ul style="list-style-type: none">• Costo de compra inicial bajo• Desempeño adecuado• Ningún cambio principal planeado en las aplicaciones ni en el sistema operativo.	<ul style="list-style-type: none">• Desempeño excelente• Espacio para el futuro• Valor a largo plazo• Actualizaciones planeadas de aplicaciones y de sistemas operativo

Tabla 2.2. Comparación de Valor-Rendimiento de una PC.⁹⁰

⁹⁰ Gutiérrez, Henry. "El microprocesador" 31 de mayo de 2009. www.reparesupc.com (17 de abril de 2,009)

Uso típico actual y futuro

PC de valor	PC de alto rendimiento
<p>Utilice un procesador de valor cuando:</p> <ul style="list-style-type: none"> • Use Correo electrónico • Utilice programas de finanzas en el Hogar • Use editores de texto u hojas de cálculo. • Juegos básicos puede jugar algunos juegos complejos, pero el sistema se vuelve lento o se bloquea de vez en cuando. • Aprendizaje • Navegación sencilla en la Web • Entrada de datos de PC 	<p>Utilice un procesador de alto rendimiento cuando:</p> <ul style="list-style-type: none"> • Desea jugar juegos de última generación o Juegos online 3D • Desea editar videos o imágenes digitales • Desea crear sus librerías de música • Ejecute Aplicaciones múltiples • Publicación de documentos • Uso de tarjetas de gráficos y video de alta resolución • Uso de Web, multimedia y gráficos • Encriptación / compresión en segundo plano

Tabla 2.3. Situaciones decisivas al obtener una PC.⁹¹

En resumen una PC de valor ha de funcionar bien si se utilizan programas como Word, StarOffice o Excel, envía o recibe correo electrónico, deja de ser funcional cuando se utilizan varios programas a la vez, como por ejemplo: editar audio o video, programas como Autocad, o le gusta jugar juegos de última generación, si estas son

⁹¹ Gutiérrez, Henry. "El microprocesador" 31 de mayo de 2009. www.reparesupc.com (17 de abril de 2,009)

las actividades en las que se ocupara la PC regularmente se recomienda comprar una de alto rendimiento.

2.11 EMPRESAS FABRICANTES DE MICROPROCESADORES⁹²

En cuanto a las empresas fabricantes de procesadores los más conocidos son: Apple, Motorola, Cyrix, Sun Microsystems, Digital Equipment Corporation, Compaq, IBM y AMD son las principales empresas que se dedican también a la fabricación de microprocesadores. En un principio la mayoría de ellas lo que hacía era copiar los procesadores de Intel pero esto suponía un doble problema. Siempre estaban por detrás de Intel Inside lo que se traducía en pérdidas ya que el que llega antes se lleva más cuotas de ventas. Por otro lado, los fabricantes de ordenadores exigen actualizar sus equipos con cierta rapidez para poder ser los primeros en sacar el mejor modelo del momento.

El caso de Apple es un poco diferente debido a que esta empresa es la única que fabrica el modelo Macintosh. De todas maneras es frecuente que Apple utilice microprocesadores Intel aunque nunca ha llegado a necesitar un Pentium.

De los antes mencionados el principal competidor de Intel Inside es AMD. Aunque es la segunda empresa en fabricación de microprocesadores más importante la distancia entre AMD e Intel es abismal. Mientras Intel controla cerca del 80% del mercado (habiendo llegado incluso al 90%) AMD se conforma con un 15%.

AMD se dedicó largo tiempo a clonar los microprocesadores de Intel pero llegó un punto de inflexión en el que diseñaba sus propios productos. En la actualidad AMD tiene defensores acérrimos que consideran sus microprocesadores de mayor calidad que los de Intel.

Para más información ver tabla 6 -8 de anexo 1

⁹² Rodríguez, Elizabeth. "Historia de los microprocesadores" www.maestrosdelweb.com (4 de Febrero de 2008)

2.12 PROGRAMACIÓN PARA MICROPROCESADORES

2.12.1 Sistemas De Numeración⁹³

En programación es frecuente acudir a diferentes sistemas de numeración según las circunstancias.

Hay que tener en cuenta que el hombre usa el sistema decimal, debido a una circunstancia más o menos afortunada: por la simple razón de que tiene diez dedos entre las dos manos. La palabra dígito y dedo tienen la misma raíz latina, por eso se usa una numeración con 10 dígitos o dedos.

Cuando se interpretan números de varias cifras, se debe de comenzar por la derecha, el primer dígito son unidades, el siguiente decenas, es decir cuántos grupos de 10 elementos estamos contando. Las siguientes centenas, es decir el número de grupos de 10 elementos, o sea el número de grupos de 100 elementos. Y así sucesivamente.

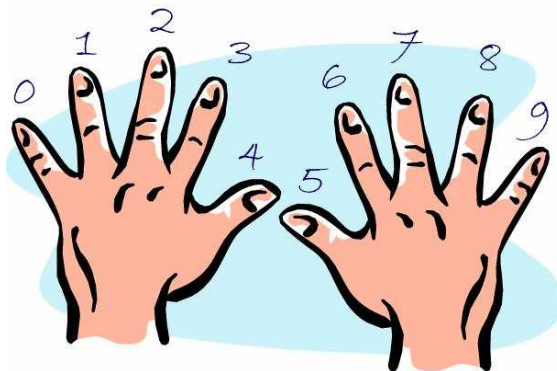


Fig. 2.32- Sistema de Conteo Decimal

⁹³ Pozo Coronado, Salvador. "Sistemas de numeración" Diciembre de 2000. articulos.conclase.net (14 de Abril 2009)

El método numérico que habitualmente es utilizado por los sistemas electrónicos digitales es el Binario, que utiliza únicamente dos cifras para representar la información: el 0 y el 1. Otros sistemas como el Octal (base 8) y el Hexadecimal (base 16) son utilizados en las computadoras.⁹⁴

2.12.1.1 ¿Qué Es Un Sistema De Numeración?⁹⁵

Un sistema de numeración se puede considerar como un conjunto de símbolos y reglas que se usan para representar los números. Dicho sistema sólo puede usar los símbolos de que dispone para representar los números. Esto quiere decir que, se pueden usar desde dos hasta infinitos símbolos para representar las cantidades. A lo largo de la historia el sistema más usado ha sido el decimal, aunque muchos de las culturas y pueblos antiguos desarrollaron otro tipo de sistemas.

Estos sistemas podían básicamente ser de dos tipos, posicionales o no posicionales. Un sistema posicional es un sistema en el cuál un símbolo o cifra no tendrá un único valor numérico, sino que según su posición dentro del número final éste valor cambiará. Por ejemplo, nuestro sistema decimal o el sistema binario son tipos de sistemas posicionales. Los sistemas no posicionales son aquellos en los cuáles los símbolos nunca cambian su valor independientemente de su posición. Por ejemplo, en los números romanos se colocan los símbolos en posiciones diferentes para indicar sumas o restas, pero el valor o cantidad que representa cada símbolo no cambia.

En los sistemas posicionales, se utiliza el concepto de base; se llama base a aquel número que indica el máximo de símbolos diferentes que podemos usar en ese

⁹⁴ López Sánchez, Juan. "Sistemas de numeración". 21 de marzo de 2000.

www.gratisweb.com/lula_f/sistemas_de_numeracion.htm (14 de Abril 2009)

⁹⁵ Carreras González, José María. "Sistemas de numeración" 12 de Agosto de 2008.

www.monografias.com (16 abril 2009)

sistema para representar los números. Así por ejemplo, en nuestro sistema decimal sólo disponemos de 10 dígitos, que van del 0 al 9.

Para indicar en qué base está un número, esto es, para saber qué sistema se utiliza; se coloca un subíndice al número indicando la base y por tanto el sistema del mismo. Evidentemente el único sistema en el que no se utiliza subíndice es el decimal.

2.12.1.2 Sistema Binario⁹⁶

El sistema de numeración más simple que usa la notación posicional es el sistema de numeración binario. Este sistema, como su nombre lo indica, usa solamente dos dígitos (0,1).

Por su simplicidad y por poseer únicamente dos dígitos diferentes, el sistema de numeración binario se usa en computación para el manejo de datos e información. Normalmente al dígito cero se le asocia con cero voltios, apagado, desenergizado, inhibido (de la computadora) y el dígito 1 se asocia con +5, +12 volts, encendido, energizado (de la computadora) con el cual se forma la lógica positiva. Si la asociación es inversa, o sea el número cero se asocia con +5 volts o encendido y al número 1 se asocia con cero volts o apagado, entonces se genera la lógica negativa.

A la representación de un dígito binario se le llama bit (de la contracción binary digit) y al conjunto de 8 bits se le llama byte, así por ejemplo: 110 contiene 3 bits, 1001 contiene 4 y 1 contiene 1 bit. Como el sistema binario usa la notación posicional entonces el valor de cada dígito depende de la posición que tiene en el número, así por ejemplo el número 110101_b es:

⁹⁶ Elizalde Vieyra, Guadalupe "Sistemas Numéricos" 2000-08-31.
www.fismat.umich.mx/~elizalde/curso/node118.html (28 Mayo 2009)

$$1*(2^0) + 0*(2^1) + 1*(2^2) + 0*(2^3) + 1*(2^4) + 1*(2^5) = 1 + 4 + 16 + 32 = 53_d$$

La computadora está diseñada sobre la base de numeración binaria (base 2). Por eso este caso particular merece mención aparte. Siguiendo las reglas generales para cualquier base expuestas antes, tendremos que:

Existen dos dígitos (0 o 1) en cada posición del número.

Numerando de derecha a izquierda los dígitos de un número, empezando por cero, el valor decimal de la posición es 2^n .

Por ejemplo, 11012 (en base 2) quiere decir:

$$1*(2^3) + 1*(2^2) + 0*(2^1) + 1*(2^0) = 8 + 4 + 0 + 1 = 13_{10}$$

2.12.1.3 Sistema Octal⁹⁷

El sistema de numeración octal es también muy usado en la computación por tener una base que es potencia exacta de 2 o de la numeración binaria. Esta característica hace que la conversión a binario o viceversa sea bastante simple. El sistema octal usa 8 dígitos (0, 1, 2, 3, 4, 5, 6,7) y tienen el mismo valor que en el sistema de numeración decimal. Como el sistema de numeración octal usa la notación posicional entonces para el número 3452.32_q tenemos:

$$2*(8^0) + 5*(8^1) + 4*(8^2) + 3*(8^3) + 3*(8^{-1}) + 2*(8^{-2}) = 2 + 40 + 4*64 + 64 + 3*512 + 3*0.125 + 2*0.015625 = 2 + 40 + 256 + 1536 + 0.375 + 0.03125 = 1834 + 40625_d \text{ entonces, } 3452.32_q = 1834.40625_d$$

El subíndice q indica número octal, se usa la letra q para evitar confusión entre la letra o y el número 0.

⁹⁷ Elizalde Vieyra, Guadalupe "Sistemas Numéricos" 2000-08-31.
www.fismat.umich.mx/~elizalde/curso/node118.html (28 Mayo 2009)

2.12.1.4 Sistema Hexadecimal

Un gran problema con el sistema binario es la verbosidad. Para representar el valor 20210 se requieren ocho dígitos binarios, la versión decimal sólo requiere de tres dígitos y por lo tanto los números se representan en forma mucho más compacta con respecto al sistema numérico binario. Desafortunadamente las computadoras trabajan en sistema binario y aunque es posible hacer la conversión entre decimal y binario, ya vimos que no es precisamente una tarea cómoda. El sistema de numeración hexadecimal, o sea de base 16, resuelve este problema (es común abreviar hexadecimal como hex. aunque hex. significa base seis y no base dieciséis). El sistema hexadecimal es compacto y nos proporciona un mecanismo sencillo de conversión hacia el formato binario, debido a esto, la mayoría del equipo de cómputo actual utiliza el sistema numérico hexadecimal. Como la base del sistema hexadecimal es 16, cada dígito a la izquierda del punto hexadecimal representa tantas veces un valor sucesivo potencia de 16, por ejemplo, el número 123416 es igual a:

$$1*16^3 + 2*16^2 + 3*16^1 + 4*16^0$$

Lo que da como resultado:

$$4096 + 512 + 48 + 4 = 466010$$

Cada dígito hexadecimal puede representar uno de dieciséis valores entre 0 y 1510. Como sólo tenemos diez dígitos decimales, necesitamos inventar seis dígitos adicionales para representar los valores entre 1010 y 1510. En lugar de crear nuevos símbolos para estos dígitos, utilizamos las letras A a la F. La conversión entre hexadecimal y binario es sencilla, considere la siguiente tabla: ⁹⁸

⁹⁸ Elizal de Vieyra, Guadalupe "Sistemas Numéricos" 2000-08-31.
www.fismat.umich.mx/~elizalde/curso/node118.html (28 Mayo 2009)

Binario	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Tabla 2.4. Tabla de Conversión de Binario a Decimal⁹⁹

Esta tabla contiene toda la información necesaria para convertir de binario a hexadecimal y viceversa. Para convertir un número hexadecimal en binario, simplemente sustituya los correspondientes cuatro bits para cada dígito hexadecimal, por ejemplo, para convertir 0ABCDh en un valor binario:

0 A B C D (Hexadecimal)

0000 1010 1011 1100 1101 (Binario)

Por comodidad, todos los valores numéricos los empezaremos con un dígito decimal; los valores hexadecimales terminan con la letra h y los valores binarios

⁹⁹ Elizal de Vieyra, Guadalupe "Sistemas Numéricos" 2000-08-31.
www.fismat.umich.mx/~elizalde/curso/node118.html (28 Mayo 2009)

terminan con la letra b. La conversión de formato binario a hexadecimal es casi igual de fácil, en primer lugar necesitamos asegurar que la cantidad de dígitos en el valor binario es múltiplo de 4, en caso contrario agregaremos ceros a la izquierda del valor, por ejemplo el número binario 1011001010, la primera etapa es agregarle dos ceros a la izquierda para que contenga doce ceros: 001011001010. La siguiente etapa es separar el valor binario en grupos de cuatro bits, así: 0010 1100 1010. Finalmente buscamos en la tabla de arriba los correspondientes valores hexadecimales dando como resultado, 2CA, y siguiendo la convención establecida: $02CA_h$.¹⁰⁰

2.12.1.5 Conversiones¹⁰¹

Conversiones Entre Sistemas De Numeración De Base Decimal A Cualquier Base

Se divide la cantidad entre la base del sistema al que se desea convertir, de esta manera:

Deseamos convertir 1453 a base 8:

$$1453/8=181 \text{ Residuo: } 5$$

$$181/8=22 \text{ Residuo: } 5$$

$$22/8=2 \text{ Residuo: } 6$$

El número sería $2655_{(8)}$

Este método de divisiones sucesivas se usa dividiendo el número decimal inicial entre la base, para posteriormente dividir los cocientes entre la base hasta que ya no se pueda dividir de manera natural. Luego se coge el último cociente y se cuenta junto con todos los residuos empezando por el último. Ese número final será el número decimal inicial convertido al sistema de la base.

¹⁰⁰ Elizalde Vieyra, Guadalupe "Sistemas Numéricos" 2000-08-31.

www.fismat.umich.mx/~elizalde/curso/node118.html (28 Mayo 2009)

¹⁰¹ Carreras González, José María. "Sistemas de numeración" 12 de agosto de 2008
www.monografias.com (1 de Mayo de 2009)

Otro ejemplo:

Deseamos convertir 3829 a base 2:

$3829/2=1914$	Residuo: 1
$1914/2=957$	Residuo: 0
$957/2=478$	Residuo: 1
$478/2=239$	Residuo: 0
$239/2=119$	Residuo: 1
$119/2=59$	Residuo: 1
$59/2=29$	Residuo: 1
$29/2=14$	Residuo: 1
$14/2=7$	Residuo: 0
$7/2=3$	Residuo: 1
$3/2=1$	Residuo: 1

El número final será: $1110\ 1111\ 0101_{(2)}$

De Cualquier Base A Decimal¹⁰²

Si se tiene un número en base n, por ejemplo 163, el proceso es el siguiente:

$$\begin{aligned} &163_{(7)} \\ &1 \times 7 = 7 \quad 7 + 6 = 13 \\ &13 \times 7 = 91 \quad 91 + 3 = 94 \end{aligned}$$

Entonces $163_{(7)}$ equivale a 94 en base decimal.

Este método es útil para números grandes, ya que para números pequeños se puede utilizar el cálculo mental según las potencias de la base, como se explicó anteriormente. De ser los números muy grandes, se tendría que conocer las tablas de multiplicar de las diferentes bases, lo que resultaría tedioso. De ahí la utilización de éste sencillo método de multiplicaciones y sumas.

Es altamente recomendable conocer las tablas de multiplicar del 2, del 8, y del 16; debido al amplio uso que se le da a éstos sistemas en informática y sistemas de computación.

¹⁰² Carreras González, José María. "Sistemas de numeración" 12 de agosto de 2008
www.monografias.com (1 de Mayo de 2009)

2.12.1.6 Operaciones Aritméticas¹⁰³

Operaciones Aritméticas De Los Distintos Sistemas.

Al igual que en el sistema decimal, también en otros sistemas de numeración, se pueden realizar operaciones aritméticas, tales como: suma, resta, multiplicación y división tomando como referencia la base del sistema dado.

Suma Binaria, Octal Y Hexadecimal.

En general, para realizar la suma se procede de la misma forma como se hace en el sistema decimal. Por ejemplo, si $a_{n-1}a_{n-2}\dots a_2a_1a_0, a_{-1}a_{-2}\dots a_{-k}$ es un número dado en una base b y $h_{n-1}h_{n-2}\dots h_2h_1h_0, h_{-1}h_{-2}\dots h_{-k}$ es otro dado en la misma base entonces la suma se debe realizar de la siguiente forma:

$$\begin{array}{cccccccc} a_{n-1} & a_{n-2} & \dots & a_1 & a_0, & a_{-1} & \dots & a_{-k} & + \\ h_{n-1} & h_{n-2} & \dots & h_1 & h_0, & h_{-1} & \dots & h_{-k} & \\ \hline (a_{n-1} + h_{n-1} + c_{n-2}) & (a_{n-2} + h_{n-2} + c_{n-3}) & \dots & (a_1 + h_1 + c_0) & (a_0 + h_0 + c_{-1}), & (a_{-1} + h_{-1} + c_{-2}) & \dots & (a_{-k} + h_{-k}) \end{array}$$

Fig. 2.33- Formula para realizar sumas

Los dígitos $m_j = (a_j + h_j + c_{j-1})$ pertenecientes al resultado se forman sumando los dígitos de cada columna de los sumandos, más el acarreo c_{j-1} que viene de la columna anterior. Cada unidad de acarreo tiene el mismo valor de la base del sistema, por ejemplo, en la suma binaria es dos, en octal ocho y en hexadecimal dieciséis. Por ejemplo, llevar 2 en hexadecimal significa que el acarreo es el doble de la base y vale exactamente 32; de este mismo modo, en binario equivale a 4 veces y 16 en octal. Los acarreos aparecen cuando las semisumas de las columnas superan la base del sistema numérico.

¹⁰³ Baena, Juan Ignacio. "Operaciones Aritméticas De Los Distintos Sistemas" 15 febrero de 2009 mii3314jibp.wordpress.com (1 mayo de 2009)

SUMA BINARIA¹⁰⁴: Las operaciones de suma binaria se realizan de la siguiente forma:

0	+	0	=	0	
0	+	1	=	1	
1	+	0	=	1	
1	+	1	=	0	Llevo 1

Tabla 2.5. Tabla de Reglas para Operaciones Binarias

Ejemplo: Dado los números binarios: $W=111110001_2$; $T=1101110101_2$; Obtener $W+T$

$$\begin{array}{r}
 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1 \\
 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\
 \hline
 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0
 \end{array}$$

Suma Octal: Se debe restar o dividir la semisuma de cada columna, cuando la misma exceda la base del sistema, y colocar en la columna inmediata del lado izquierdo, el valor del acarreo tantas veces se haya superado la base del sistema. De esta misma forma cada unidad que se acarree equivale a ocho unidades de la columna anterior.

Ejemplo: Dado los números octales: A. 40740647 y B. 25675300, Obtener $A+B$

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 6\ 6\ 6\ 3\ 6\ 1\ 4\ 7
 \end{array}$$

Fig. 2.34- Suma de Números Octales

¹⁰⁴ Baena, Juan Ignacio. "Operaciones Aritméticas De Los Distintos Sistemas" 15 febrero de 2009 mii3314jibp.wordpress.com (1 mayo de 2009)

Suma Hexadecimal: Se debe restar o dividir la semisuma de cada columna, cuando la misma exceda la base del sistema, y colocar en la columna inmediata del lado izquierdo, el valor del acarreo tantas veces se haya superado la base del sistema. Cada unidad que se acarree equivale a dieciséis unidades de la columna anterior.

Ejemplo: Dado los números binarios¹⁰⁵:

$$\begin{array}{r}
 \text{2 1 1} \leftarrow \text{ACARREO} \\
 \text{F 3 B C} \\
 \text{9 D D 0} \\
 + \text{3 A 0 6 0} \\
 \hline
 \text{5 3 1 E C}
 \end{array}$$

Fig. 2.35- Suma de Números Hexadecimales

Multiplicación Binaria, Octal Y Hexadecimal.

La operación aritmética de multiplicar se realiza del mismo modo que en el sistema numérico decimal.

Multiplicación Binaria:

Ej.: Multiplicar A. 111011_2 y B. 111_2

$$\begin{array}{r}
 111011 \\
 \times 111 \\
 \hline
 111011 \\
 111011 \\
 111011 \\
 \hline
 110011101
 \end{array}$$

¹⁰⁵ Baena, Juan Ignacio. "Operaciones Aritméticas De Los Distintos Sistemas" 15 febrero de 2009 mii3314jibp.wordpress.com (1 mayo de 2009)

Multiplicación Octal¹⁰⁶:

Ej.: Multiplicar A. 67234_8 y B. 16_8

$$\begin{array}{r} 67234 \\ \quad \times 16 \\ \hline 513650 \\ + 67234 \\ \hline 1406210 \end{array}$$

Multiplicación Hexadecimal:

Ej.: Multiplicar A. $67D34_{16}$ y B. 12_{16}

$$\begin{array}{r} 67D34 \\ \quad \times 12 \\ \hline CFA68 \\ + 67D34 \\ \hline 74CDA8 \end{array}$$

División Binaria, Octal Y Hexadecimal.

La operación aritmética de dividir se realiza del mismo modo que en el sistema numérico decimal.

¹⁰⁶ Baena, Juan Ignacio. "Operaciones Aritméticas De Los Distintos Sistemas" 15 febrero de 2009 mii3314jibp.wordpress.com (1 mayo de 2009)

División Binaria:

$$\begin{array}{r} 11011101 \mid 101 \\ -101 \\ \hline 111 \\ 101 \\ \hline 0101 \\ 101 \\ \hline 00001 \end{array}$$

RESIDUO

0 1

Fig. 2.36- División de Números Binarios

División Octal Y Hexadecimal: La división se efectúa del mismo modo que en el sistema decimal y se realiza directamente en la misma base del sistema octal o hexadecimal. Sin embargo, también se puede obtener previamente la conversión en binario y proceder, como en el caso anterior, a realizarla en binario; y después el resultado transformarlo de nuevo al sistema numérico original.

2.12.2 Lenguaje Ensamblador¹⁰⁷

El lenguaje ensamblador es un lenguaje de programación que es una traducción directa del código de máquina (Éste código es interpretado por el microprocesador), para que pueda ser entendible por los seres humanos, por lo tanto es un lenguaje de bajo nivel. El lenguaje ensamblador permite una optimización que no se consigue con lenguajes de medio y alto nivel. Permite la creación de programas muy rápidos y muy pequeños, además se tiene un control que no se consigue con un lenguaje de alto nivel.

2.12.2.1 Importancia Del Lenguaje Ensamblador¹⁰⁸

La importancia del lenguaje ensamblador radica principalmente en que se trabaja directamente con el microprocesador; por lo cual se debe de conocer el

¹⁰⁷ Wikilibros. "Programación en lenguaje ensamblador" 8 may 2009
<http://es.wikibooks.org/wiki> (30 abril de 2009)

¹⁰⁸ Mendoza, Pech y Contreras. "Lenguaje ensamblador" 25 de septiembre de 2003
www.monografias.com (30 abril de 2009)

funcionamiento interno de este, tiene la ventaja de que en él se puede realizar cualquier tipo de programas que en los lenguajes de alto nivel no lo pueden realizar. Otro punto sería que los programas en ensamblador ocupan menos espacio en memoria.

2.12.2.2 Ventajas Del Lenguaje Ensamblador¹⁰⁹

1. Velocidad.- Como trabaja directamente con el microprocesador al ejecutar un programa, pues como este lenguaje es el más cercano a la máquina la computadora lo procesa más rápido.
2. Eficiencia de tamaño.- Un programa en ensamblador no ocupa mucho espacio en memoria porque no tiene que cargar librerías y demás como son los lenguajes de alto nivel.
3. Flexibilidad.- Es flexible porque todo lo que puede hacerse con una máquina, puede hacerse en el lenguaje ensamblador de esta máquina; los lenguajes de alto nivel tienen en una u otra forma limitante para explotar al máximo los recursos de la máquina. O sea que en lenguaje ensamblador se pueden hacer tareas específicas que en un lenguaje de alto nivel no se pueden llevar a cabo porque tienen ciertas limitantes que no se lo permiten.

2.12.2.3 Desventajas Del Lenguaje Ensamblador

1. Tiempo de programación.- Como es un lenguaje de bajo nivel requiere más instrucciones para realizar el mismo proceso, en comparación con un lenguaje de alto nivel. Por otro lado, requiere de más cuidado por parte del programador, pues es propenso a que los errores de lógica se reflejen más fuertemente en la ejecución.
2. Programas fuente grandes.- Por las mismas razones que aumenta el tiempo, crecen los programas fuentes; simplemente requerimos más instrucciones primitivas para describir procesos equivalentes. Esto es una desventaja porque

¹⁰⁹ Mendoza, Pech y Contreras. "Lenguaje ensamblador" 25 de septiembre de 2003
www.monografias.com (30 abril de 2009)

dificulta el mantenimiento de los programas, y nuevamente reduce la productividad de los programadores.

3. Peligro de afectar recursos inesperadamente.- Que todo error que se pueda cometer, o todo riesgo que se pueda tener, se podrá afectar los recursos de la maquina, programar en este lenguaje lo más común que pueda pasar es que la máquina se bloquee o se reinicie. Porque con este lenguaje es perfectamente posible (y sencillo) realizar secuencias de instrucciones inválidas, que normalmente no aparecen al usar un lenguaje de alto nivel.
4. Falta de portabilidad.- Para cada máquina existe un lenguaje ensamblador; por ello, evidentemente no es una selección apropiada de lenguaje cuando deseamos codificar en una máquina y luego llevar los programas a otros sistemas operativos o modelos de computadoras.

2.12.2.4 Relación Del Lenguaje Ensamblador Con Los Componentes Internos Del Procesador¹¹⁰

1. En la memoria se almacena la información en celdas especiales llamados registros los cuales tienen un nivel alto y un nivel bajo.
2. Unidad aritmética y lógica es la responsable de realizar como su nombre lo indica operaciones aritméticas y lógicas.
3. Unidad de control se encarga de coordinar de que los otros componentes ejecuten las operaciones correctamente.
4. Bus interno son los canales por donde pasa la información que la máquina va a procesar (bus de entrada) o procesada (bus de salida).

¹¹⁰ Mendoza, Pech y Contreras. "Lenguaje ensamblador" 25 de septiembre de 2003
www.monografias.com (30 abril de 2009)

2.12.2.5 Tipos De Ensambladores ¹¹¹

Aunque todos los ensambladores realizan básicamente las mismas tareas, se pueden clasificar de acuerdo a características, de la siguiente manera:

Ensambladores Cruzados (Cross-Assembler).

Se denominan así los ensambladores que se utilizan en una computadora que posee un procesador diferente al que tendrán las computadoras donde va a ejecutarse el programa.

El empleo de este tipo de traductores permite aprovechar el soporte de medios físicos (discos, impresoras, pantallas, etc.), y de programación que ofrecen las máquinas potentes para desarrollar programas que luego los van a ejecutar sistemas muy especializados en determinados tipos de tareas.

Ensambladores Residentes.

Son aquellos que permanecen en la memoria principal de la computadora y cargan, para su ejecución. Este tipo de ensamblador tiene la ventaja de que se puede comprobar inmediatamente el programa sin necesidad de transportarlo de un lugar a otro, como se hacía en cross-assembler, y sin necesidad de programas simuladores.

Sin embargo, puede presentar problemas de espacio de memoria, ya que el traductor ocupa espacio que no puede ser utilizado por el programador. Asimismo, también ocupará memoria el programa fuente y el programa objeto. Esto obliga a tener un espacio de memoria relativamente amplio. Es el indicado para desarrollos de pequeños sistemas de control y sencillos.

La ventaja de estos ensambladores es que permiten ejecutar inmediatamente el programa; la desventaja es que deben mantenerse en la memoria principal tanto el ensamblador como el programa fuente y el programa objeto.

¹¹¹ Prof. Lauro Soto. "Lenguaje Ensamblador" www.mitecnologico.com (2 mayo de 2009)

Macroensambladores.¹¹²

Son ensambladores que permiten el uso de macroinstrucciones (macros). Debido a su potencia, normalmente son programas robustos que no permanecen en memoria; una vez generado el programa objeto. Puede variar la complejidad de los mismos, dependiendo de las posibilidades de definición y manipulación de las macroinstrucciones, pero normalmente son programas bastantes complejos, por lo que suelen ser ensambladores residentes.

Microensambladores.¹¹³

Generalmente, los procesadores utilizados en las computadoras tienen un repertorio fijo de instrucciones, es decir, que interpretaba de igual forma un determinado código de operación.

El programa que indica al intérprete de instrucciones de la UCP cómo debe actuar se denomina microprograma. El programa que ayuda a realizar este microprograma se llama microensamblador. Existen procesadores que permiten la modificación de sus microprogramas, para lo cual se utilizan microensambladores.

Ensambladores de una Fase.

Estos ensambladores leen una línea del programa fuente y la traducen directamente para producir una instrucción en lenguaje máquina o la ejecuta si se trata de una pseudoinstrucción. También va construyendo la tabla de símbolos a medida que van apareciendo las definiciones de variables, etiquetas, etc.

Debido a su forma de traducción, estos ensambladores obligan a definir los símbolos antes de ser empleados para que, cuando aparezca una referencia a un determinado símbolo en una instrucción, se conozca la dirección de dicho símbolo y

¹¹² Prof. Lauro Soto. "Lenguaje Ensamblador" www.mitecnologico.com (2 mayo de 2009)

¹¹³ "Introducción a los Microprocesadores" 19 Febrero, 2002. canalhanoi.iespana.es (16 abril de 2,009)

se pueda traducir de forma correcta. Estos ensambladores son sencillos, baratos y ocupan poco espacio.

Ensambladores de Dos Fases. ¹¹⁴

Los ensambladores de dos fases se denominan así debido a que realizan la traducción en dos etapas. En la primera fase, leen el programa fuente y construyen una tabla de símbolos; de esta manera, en la segunda fase, vuelven a leer el programa fuente y pueden ir traduciendo totalmente, puesto que conocen la totalidad de los símbolos utilizados y las posiciones que se les ha asignado. Estos ensambladores son los más utilizados en la actualidad.

¹¹⁴ "Introducción a los Microprocesadores" 19 Febrero, 2002. canalhanoi.iespana.es (16 abril de 2,009)

CAPITULO III

CAPITULO III

Antecedentes De La Situación Actual

3.1 ANTECEDENTES DE LA SITUACIÓN ACTUAL

El proyecto impulsado por el departamento de Ingeniería y Arquitectura en la FMO orientado a desarrollar profesionales en el área de Ingeniería de Sistemas Informáticos cuenta con el objetivo principal de volverse una alternativa viable hacia los estudiantes en dicha área, en la región Occidente del país y que estos tengan una opción adicional de estudio en la Facultad Multidisciplinaria de Occidente.

La situación en la que se ha ido desarrollando la carrera de Ingeniería de Sistemas Informáticos a nivel de prácticas de laboratorio de hardware ha carecido de desarrollo y evolución, dándole más énfasis a prácticas realizadas a nivel de software (programación y simulación).

Se han tomado iniciativas anteriormente al impartir cursos adicionales dirigidos a los estudiantes que cumplieran con cierto nivel de estudio de la carrera; dichos cursos fueron orientados a las áreas de mantenimiento de computadoras y redes informáticas.

Desde el año 2004 se cuenta con un auxiliar de cátedra quien es encargado del desarrollo de las prácticas de laboratorio en las asignaturas de Sistemas Digitales y Arquitectura de Computadoras. En la actualidad, se cuenta con un encargado de los laboratorios prácticos en la materia de Microprogramación de Computadoras.

El Departamento de Ingeniería y Arquitectura de la FMO, posee un laboratorio de hardware ubicado en el edificio de usos múltiples de la FMO, en dicho lugar se realizan las prácticas de laboratorio para las asignaturas relacionadas a la electrónica y redes. Este no cuenta con una infraestructura y equipos adecuados para la realización de algunas prácticas especializadas en el área de microprocesadores.

3.2 METODOLOGÍA DE LA INVESTIGACIÓN

Existen diferentes tipos de investigación entre las cuales tenemos:

Por la clase de medios utilizados para obtener los datos: documental, de campo o experimental.

- Investigación documental: Este tipo de investigación es la que se realiza, como su nombre lo indica, apoyándose en fuentes de carácter documental, esto es, en documentos de cualquier especie. Como subtipos de esta investigación encontramos la investigación bibliográfica, la hemerográfica y la archivística; la primera se basa en la consulta de libros, la segunda en artículos o ensayos de revistas y periódicos, y la tercera en documentos que se encuentran en los archivos, como cartas, oficios, circulares, expedientes, etc.
- Investigación de campo: Este tipo de investigación se apoya en informaciones que provienen entre otras, de entrevistas, cuestionarios, encuestas y observaciones. Como es compatible desarrollar este tipo de investigación junto a la investigación de carácter documental, se recomienda que primero se consulten las fuentes de la de carácter documental, a fin de evitar una duplicidad de trabajos.
- Investigación experimental: Recibe este nombre la investigación que obtiene su información de la actividad intencional realizada por el investigador y que se encuentra dirigida a modificar la realidad con el propósito de crear el fenómeno mismo que se indaga, y así poder observarlo. Por el nivel de conocimientos que se adquieren: exploratoria, descriptiva o explicativa.
- Investigación exploratoria: Recibe este nombre la investigación que se realiza con el propósito de destacar los aspectos fundamentales de una problemática determinada y encontrar los procedimientos adecuados para elaborar una investigación posterior. Es útil desarrollar este tipo de investigación porque, al contar con sus resultados, se simplifica abrir líneas de investigación y proceder a su consecuente comprobación.
- Investigación descriptiva: Mediante este tipo de investigación, que utiliza el método de análisis, se logra caracterizar un objeto de estudio o una situación

concreta, señalar sus características y propiedades. Combinada con ciertos criterios de clasificación sirve para ordenar, agrupar o sistematizar los objetos involucrados en el trabajo indagatorio. Al igual que la investigación que hemos descrito anteriormente, puede servir de base para investigaciones que requieran un mayor nivel de profundidad.

- Investigación explicativa: Mediante este tipo de investigación, que requiere la combinación de los métodos analítico y sintético, en conjugación con el deductivo y el inductivo, se trata de responder o dar cuenta del porqué del objeto que se investiga.

A partir del hecho que una investigación puede incluir elementos de los seis tipos de estudios mencionados, esto quiere decir abarcar los fines de dichos métodos de investigación; y teniendo en consideración que su clasificación se encuentra sostenida a partir del conocimiento actual del tema de investigación, y del enfoque que el investigador pretenda darle, se determina que el presente estudio puede definirse como una investigación social “Exploratoria – Descriptiva”, la cual tiene las siguientes características:

1. No se pretende demostrar ninguna teoría o hipótesis.
2. La problemática de la investigación, esta basada en las condiciones actuales, las cuales servirán para realizar un diagnóstico y así determinar los requerimientos necesarios.
3. A pesar de existir investigaciones similares, esta no pierde su carácter exploratorio puesto que se dirigen explícitamente a la aplicación de laboratorios especializados.

3.2.1 Elementos De La Investigación

Los elementos a investigar en el capítulo siguiente son mencionados a continuación:

La comunidad estudiantil de la Facultad Multidisciplinaria de Occidente que cursaron la materia de microprogramación en la carrera de Ingeniería de Sistemas Informáticos, a los cuales se les aplicara una encuesta como involucrados directos en la implementación de los laboratorios prácticos, profesionales que laboran actualmente en el área de la microprogramación y docentes de la carrera de sistemas informáticos, Instituciones educativas que cuentan con laboratorios de especialización en microprocesadores.

3.2.2 Fuentes De Información

En toda investigación es de importancia fundamental los hechos y relaciones que se establecen; que los resultados obtenidos o nuevos conocimientos posean el grado máximo de exactitud y confiabilidad. Para ello se planea una metodología o procedimiento ordenado para establecer lo más significativo de los hechos y fenómenos estudiados hacia los cuales está encaminado el significado de la investigación. Se utilizaran dos tipos de fuentes: información primaria e información secundaria.

3.2.2.1 Fuentes Primarias

Dentro de estas fuentes se encuentran tanto entidades como personas las cuales facilitaran información teórica a través de documentación técnica que son de utilidad para el progreso del trabajo. Así mismo se tomara en cuenta toda la información directa que puedan brindarnos por medio de los conocimientos adquiridos en la ejecución de su trabajo. Entre estas se pueden mencionar:

- a) La comunidad estudiantil y docente del Departamento de Ingeniería y Arquitectura de la Facultad Multidisciplinaria de Occidente; como involucrados directos en la propuesta.
- b) Las Instituciones Educativas que cuenten con Laboratorios de Microprogramación
- c) Estudiantes Egresados de la Carrera de Ingeniería de Sistemas Informáticos que brinden una perspectiva diferente de la problemática.

3.2.2.2 Fuentes Secundarias

Este tipo de medios se toma como referencia para establecer el sitio de las fuentes de investigación primarias, puesto que muestran los listados donde se sintetiza la información más importante acerca del tema en estudio. Entre este tipo de medios se pueden mencionar: tesis, revistas, bibliotecas e internet.

3.2.3 Investigación Documental

A continuación se presenta la información documental que se ha logrado recopilar para el correcto desarrollo de la investigación:

- Hojas de Datos Técnicos del microprocesador 8085
- Manual de uso para el Software propuesto
- Programa y Plan de estudio actual de la asignatura de Microprogramación
- Fuentes bibliográficas (Libros, Internet, Trabajos de Grado, etc.)

3.2.4 Investigación De Campo

La investigación de campo con la cual fue desarrollado el estudio de la problemática consistió en visitas técnicas a diversas instituciones educativas públicas y privadas que trabajan con microprocesadores ya sea de una forma didáctica u operacional.

Las herramientas utilizadas para lograr la recopilación de información en la investigación de campo fueron:

- **Entrevistas:** Las cuales han sido dirigidas a personal profesional que se encuentra directamente implicado en el área de electrónica específicamente en el area de programación de microprocesadores, ya sea en las instituciones visitadas así como a una parte de la comunidad estudiantil de La Universidad de El Salvador en la Facultad Multidisciplinaria de Occidente.
- **Encuestas:** Así como se indicó con anterioridad se han dirigido a la comunidad universitaria (estudiantes) y docentes de la carrera de Ingeniería de Sistemas

Informáticos de la Facultad Multidisciplinaria de Occidente. Con este método, se desea lograr recopilar datos que permitan valorar el desempeño de los laboratorios prácticos de la facultad determinando si son los adecuados para la correcta asimilación de los conocimientos y logrando el cumplimiento de objetivos de la asignatura orientados a los estudiantes para implementar una mejor investigación y experimentación. La información que se logre recabar será de suma importancia para poder fundamentar la implementación de un laboratorio práctico con microprocesadores. El diseño de las encuestas se ha hecho con preguntas cerradas, con el objetivo de no involucrarse en opiniones divididas afectadas por agentes externos.

3.2.5 Determinación Del Universo Y Tamaño De Muestra

Tomando como base las fuentes primarias de investigación (encuestas dirigidas a Estudiantes, Docentes, Auxiliares de Cátedra e Instructores así como también a Ex-alumnos de la Universidad de El Salvador FMO de la carrera de ingeniería de sistemas informáticos) para el desarrollo de este trabajo investigativo se ha logrado establecer el tamaño de los universos y las muestras de la siguiente manera:

TIPO	UNIVERSO	MUESTRA
Estudiantes	63	-
Docentes, Auxiliares de Cátedra e Instructores	6	-
Ex-alumnos	133	13

Tabla 1 - Población encuestada

Para esta investigación se obvio la determinación de muestra por medio de formula debido a que el universo era demasiado pequeño, por esto se tomo todo el universo para el sector estudiantil y sector docentes, auxiliares de cátedra e instructores.

Después de haber aplicado el método de recopilación de datos, se procede a la interpretación de resultados, interpretando cada pregunta mostrándolas en el orden establecido.

Dicha interpretación consiste en la presentación de cada pregunta con su respectivo objetivo y con su respectiva tabla de datos. Además se presenta un gráfico como ayuda visual para observar y comprender con mayor facilidad los resultados obtenidos. Además se presenta un comentario por cada pregunta, con el propósito de describir en forma general los resultados obtenidos, así como también se realizan las observaciones adicionales que se consideren pertinentes.

Determinación de Tamaño de la Muestra de los Ex-Alumnos

Para poder realizar el cálculo de la muestra que se va a tomar una muestra forzada del universo total de ex-alumnos graduados de la universidad, siendo esto un 10% del universo total.

$$133 \times 0.10 = 13 \text{ ex-alumnos}$$

Muestra = 13 ex-alumnos graduados de la carrera de ingeniería de sistemas informáticos.

3.3 INTERPRETACIÓN DE LOS DATOS DE LA ENCUESTA REALIZADA

A continuación se presenta el análisis e interpretación de los datos obtenidos a través de la recopilación de los mismos por medio de las diferentes encuestas que fueron realizadas y que permitieron dar a conocer las opiniones y necesidades que se presentan en la Carrera de Ingeniería en Sistemas Informáticos de la Universidad de El Salvador-FMO. De esta forma permite la identificación de problemas claves en cuanto a las metodologías e implementación de equipos especializados en las prácticas de laboratorio con microprocesadores. De esta manera permitir que los estudiantes puedan asimilar los conocimientos de forma efectiva, permitiendo que los mismos sean protagonistas en las prácticas de laboratorio con microprocesadores y que puedan así resolver problemas prácticos que puedan presentarse en el campo laboral en su futuro como profesionales. Esto conllevará a que la calidad de enseñanza académica en dicha área mejore y que el perfil profesional de los estudiantes sea más competitivo.

Algunos de los problemas identificados por la población total encuestada, los cuales se presentan a la hora de recibir los conocimientos sobre microprocesadores y el área de prácticas son los siguientes:

- No se posee equipo especializado de hardware para realizar prácticas de laboratorio.
- Los conocimientos de los ex-alumnos no se encuentran en un nivel aceptable para trabajar profesionalmente en el área de microprocesadores.
- Falta de material didáctico para impartir prácticas de laboratorio.
- Poca asimilación de los conocimientos teóricos por falta de laboratorios prácticos con equipos reales.

- Espacio físicamente no adecuados para la realización de las prácticas de laboratorio.
- Falta de conocimiento previo, por parte de los estudiantes, sobre áreas afines a los microprocesadores.
- Falta de apoyo por parte de las autoridades de la FMO para obtener más recursos tecnológicos.
- Ausencia de capacitaciones al personal docente, auxiliares de cátedra e instructores sobre el manejo de microprocesadores.
- El tiempo para el desarrollo de laboratorios prácticos con microprocesadores es de muy baja relevancia dando como resultado que los conocimientos no sean asimilados correctamente.

Para más información ver Anexo 2, 3 y 4.

3.4 VISITAS TÉCNICAS

En la actualidad, el ámbito laboral exige a los profesionales en sistemas informáticos desarrollarse en diversas áreas de la carrera. Para poder obtener los conocimientos de una manera eficiente y correcta se debe contar con el equipo especializado y herramientas necesarias para la asimilación de los conocimientos teóricos recibidos en las aulas y llevarlos al área práctica.

En El Salvador existen algunas instituciones educativas que se encuentran bien equipadas con herramientas que permiten trabajar en el área de microprocesadores. Con el fin de recopilar nuevas ideas para poder tener una visión más amplia y clara del equipamiento en dichas instituciones se han realizado visitas y entrevistas técnicas a los encargados de los laboratorios y especialistas de dichas instituciones. Las instituciones en las cuales se realizaron las visitas técnicas fueron:

- Instituto Tecnológico Centroamericano (ITCA/FEPADE)
- Instituto Técnico Ricaldone
- Universidad Don Bosco
- Instituto Nacional Santa Lucía

Estas instituciones educativas cuentan con laboratorios especializados y/o entrenadores para el manejo de microprocesadores. Los cuales están relacionados directamente con la carrera de Ingeniería en Sistemas Informáticos que es impartida en la Facultad Multidisciplinaria de Occidente.

Los datos resultantes de las visitas técnicas se encuentran organizados en las tablas 1 y 2 en la sección de Anexos 6. Dentro de los cuales se pueden encontrar las descripciones de las instituciones, información de contacto, condiciones físicas, equipamiento, entre otros; incluyendo además fotografías tomadas al equipo especializado y a los laboratorios donde se desarrollan las prácticas.

Para más información ver Anexo 5, 6 y 7.

3.5 CONCLUSIONES DE LA INVESTIGACIÓN

Después de haber realizado la investigación técnica pertinente a través de encuestas, entrevistas y visitas técnicas a otras instituciones educativas, y luego de analizar la información recabada se concluye lo siguiente:

Primeramente se debe destacar que si los docentes y personal que imparten clases teóricas y laboratorios prácticos tienen el equipo y herramientas necesarias facilitaría la enseñanza de microprocesadores puesto que el nivel de interés de los estudiantes aumentaría, lo cual permitiría que estos asimilaran los conocimientos sobre dicha área de una mejor manera y la Facultad Multidisciplinaria de Occidente produciría mejores profesionales para trabajar en una área que aun no ha sido explotada o no muy bien capacitada para trabajar en ella. Esto aumentaría el nivel de competitividad de los profesionales que egresan de la facultad pudiendo así obtener el nivel de conocimiento necesario para tener más oportunidades en cuanto al manejo de equipos que utilizan microprocesadores desde equipos móviles, computadoras, hasta maquinaria industrial.

En las instituciones educativas en las cuales se realizaron las visitas técnicas se logró observar que se utiliza equipo especializado de hardware que posee microprocesadores (de las cuales usan el 8085, 8088, Z80 o 8086), debido a que su estructura y su set de instrucciones son más sencillos; facilitando así la enseñanza y comprensión del estudio de los microprocesadores.

Luego de haber realizado la investigación correspondiente tanto en la UES FMO como en las diferentes instituciones educativas que se encuentran equipadas con laboratorios especializados en el área de microprocesadores se lograron desglosar algunas ideas claves al momento de implementar un laboratorio de este tipo en la UES FMO, puesto que gracias a esta investigación se han logrado comprender diversas perspectivas de la situación en la que se encuentran implementados; visualizando así algunos aspectos como lo son: los programas de estudio que son

utilizados, herramientas de software y hardware, tipo de microprocesadores utilizados, metodología de enseñanza, etc., para el desarrollo de las materias relacionadas con microprocesadores

Esto permite la determinación de una solución que pueda ser implementada en los laboratorios prácticos relacionados al manejo y programación de microprocesadores en la UES FMO permitiendo así mejorar la calidad de enseñanza impartida a los estudiantes.

En el mercado existe una amplia gama de microprocesadores, pero debido a su nivel de complejidad en cuanto a los modelos actuales, se ha determinado que el modelo más factible para el estudio del mismo es el 8085 gracias a la simplicidad con la que cuenta en su diseño, arquitectura y juego de instrucciones. El 8085 es el microprocesador que permitirá al estudiante asimilar de una mejor manera los conocimientos teóricos de los microprocesadores, además de facilitar su enseñanza a los docentes, lo cual conlleva a la adquisición de conocimientos más sólidos sobre el funcionamiento de los mismos.

Al implementar los recursos y herramientas tanto de hardware como de software mejora en gran medida la enseñanza en las instituciones educativas, las cuales buscan mejorar la calidad de profesionales en el área laboral, esto es resultado de desarrollar programas educativos que permitan la clara aplicación de los conocimientos teóricos a la práctica, utilizando así todos los sentidos. Es por lo anteriormente mencionado que se propone **La Implementación de un Laboratorio Didáctico (Primer TRAINER para Microprocesador 8085 y Herramientas de Desarrollo)**

3.5.1 Laboratorio Práctico PRIMER TRAINER Para Microprocesador 8085

El Laboratorio Práctico Primer TRAINER para el Microprocesador 8085 se encuentra compuesto de los siguientes elementos:

Kit de entrenamiento (Primer TRAINER), modelo basado en el microprocesador 8085 el cual contiene diferentes elementos como los siguientes:

- Puerto serial RS232 y cable (para conectividad con ordenador personal)
- Cable de Interfaz de Entrada-salida
- Sistema operativo de Monitor Realizado (EMOS)
- EPROM w/software
- Suministro de energía de transformador de la pared, 9V 500mA
- Estuche plástico
- Teclado Hex
- Display

Las herramientas de software a utilizar constan de tres tipos de aplicaciones como lo son: programa ensamblador, programa simulador o emulador de microprocesador 8085 y la terminal con la cual se podrá administrar el equipo. Algunas de las funciones que estas aplicaciones permitirán son:

- Ensamblador: compilar el código de .asm a .hex (entendible al equipo)
- Simulador: Permite la creación y edición de los programas en ensamblador, simulación del comportamiento del programa, compilación
- Terminal: Transferencia de programas de la memoria del kit entrenador (Primer TRAINER 8085) a la terminal

CAPITULO IV

CAPITULO IV

Plan De Elaboración De Material Didáctico Y Manuales
Para El Uso De Microprocesadores En El Laboratorio
De Hardware, Propuesta De Equipamiento, Metodología
Y Presupuesto Para La Implementación Del Laboratorio
Didáctico

4.1 PROPUESTA DE LABORATORIO PRÁCTICO CON MICROPROCESADORES

4.1.1 Análisis Del Temario y Actividades.

Uno de los objetivos más importantes que posee la asignatura de microprogramación es el de proporcionar a los estudiantes conocimientos técnicos que sean sólidos relacionados al manejo de microprocesadores. Entre estos se encuentran la de conocer y aplicar técnicas tanto básicas como avanzadas del lenguaje ensamblador, así como también el manejo de las diferentes tecnologías acerca de los microprocesadores tanto de hardware como de software. Todo esto con la finalidad de que el estudiante sea capaz de resolver los diferentes problemas que se puedan presentar en el desarrollo del entorno profesional en aspectos relacionados con microprocesadores.

Temario de la Asignatura.

UNIDAD	CONTENIDOS
UNIDAD I. INTRODUCCIÓN AL LENGUAJE ENSAMBLADOR	Conceptos Generales Lenguajes de programación Representación de datos Binario Octal Hexadecimal
UNIDAD II. ARQUITECTURA DEL PC	Bases de microprocesadores Componentes básicos del procesador Ciclo de fetch-execute Interrupciones Características de la familia 80X86 Organización de la memoria
UNIDAD III. LENGUAJE ENSAMBLADOR	Instrucciones básicas Acceso a registros y memoria Aritméticas y lógicas Manipulación de bits Control de secuencia Modos de direccionamiento Entrada y salida básicas
UNIDAD IV. TÉCNICAS LÓGICAS	Estructura de programación

DE PROGRAMACIÓN	Subrutinas Arreglos Ciclos
UNIDAD V. INTERRUPCIONES	Manejo de directorios Crear archivos Abrir archivos Leer archivos Escribir en el archivo Posicionarse en el archivo Cerrar el archivo Borrar el archivo

Tabla 4.1. Temario de la Asignatura

La necesidad de una nueva metodología en la realización de prácticas de laboratorio con microprocesadores para la asignatura de microprogramación, conlleva a la implementación de un plan de trabajo práctico que favorezca a los estudiantes.

Se han examinado y establecido diferentes herramientas que ayuden al desarrollo de prácticas de laboratorio (Software y Hardware), teniendo como punto importante que las herramientas de software que se seleccionaron fueran de carácter gratuito.

Se tiene a disposición herramientas de hardware que permitan el trabajo en tiempo real, sin dejar a un lado el trabajo de simulación. Con esto se pretende que el alumno tenga una mejor asimilación de los conocimientos y pueda enfrentarse a la realidad sin ningún problema.

A lo largo de la investigación se ha determinado y comprobado que lo importante es que el estudiante conozca y experimente con dispositivos reales que le aporten más confianza y conocimientos acerca de microprocesadores.

El incentivar a los estudiantes a estudiar y trabajar con microprocesadores se logra al poner en práctica los conocimientos teóricos impartidos en clases despertando así la curiosidad y permitiendo formar profesionales capacitados en dicha área.

4.1.2 Plan de Guías de Práctica

Dentro del desarrollo de la materia de Microprogramación el estudiante conocerá sobre el lenguaje ensamblador, además aprenderá a utilizar las herramientas de software y hardware, desarrollar y resolver diferentes programas escritos en lenguaje ensamblador aplicando las técnicas y lógicas de programación y luego poniéndolos en práctica en el equipo especializado.

Relación con el Plan de Estudio	Numero de Guía	Nombre de Guía
Introducción al Lenguaje Ensamblador	1 (ver anexo 8.1)	Electrónica digital para computadoras
	2 (ver anexo 8.2)	Introducción al simulador e instrucciones básicas del 8085.
Arquitectura del PC	3 (ver anexo 8.3)	Introducción al Equipo Primer TRAINER
	4 (ver anexo 8.4)	Utilización de los programas de software del Primer TRAINER
Lenguaje Ensamblador	5 (ver anexo 8.5)	Suma / Resta de 8 y 16 bits
	6 (ver anexo 8.6)	BCD, Multiplicación y División
	7 (ver anexo 8.7)	Uso de Instrucciones Lógicas y Condicionales
	8 (ver anexo 8.8)	Uso de Direccionamiento Indirecto de Registros
	9 (ver anexo 8.9)	Usando Variables
Técnicas Lógicas de Programación	10 (ver anexo 8.10)	El Registro de Pila e Instrucciones Relacionadas.
	11 (ver anexo 8.11)	Subrutinas
	12 – opcional (ver anexo 8.12)	Uso de Subrutinas del Sistema Operativo MOS

	13 (ver anexo 8.13)	Uso de Instrucciones de Comparación
Interrupciones	14 (ver anexo 8.14)	Uso de Interrupciones
	15 – opcional (ver anexo 8.15)	Escribiendo Programas Propios.

Tabla 4.2. Plan de Guías Prácticas

4.2 PROPUESTA DE EQUIPAMIENTO PARA LOS LABORATORIOS

Para que los estudiantes puedan tener una mejor comprensión de los conocimientos teóricos es necesario y se considera una importante herramienta la realización de prácticas de laboratorio para mejorar sus habilidades técnicas en el área.

4.2.1 Programación en Lenguaje Ensamblador

El lenguaje ensamblador es, en síntesis, una secuencia de instrucciones que se convierten a un código hexadecimal ejecutable por la máquina a través de un programa llamado Ensamblador.

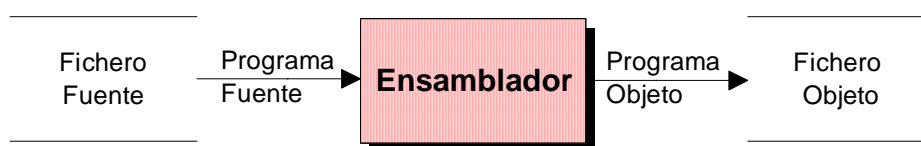


Figura 4.1 Pasos de Creación del Lenguaje Ensamblador

El Ensamblador convierte el programa fuente, escrito en lenguaje ensamblador, en su equivalente en hexadecimal, denominado programa objeto. El programa objeto es muy similar a la representación en memoria que tendrá el programa.

4.2.2 Descripción del Equipo de Hardware Primer TRAINER 8085

El Primer TRAINER es un equipo compacto, basado en el sistema del microprocesador 8085, diseñado principalmente para fines educativos. A pesar de su reducido tamaño, contiene muchos elementos importantes y educativos. Primer TRAINER tiene E/S digitales, E/S analógicas, además de una pantalla y teclado para interfaz humana. También hay varias opciones que pueden añadirse a la unidad básica para mejorar la flexibilidad y funcionalidad.

Características Estándar de todos los Kits Primer TRAINER:

- Microprocesador 8085 que opera a 3.072 MHZ.
- 256 bytes de memoria RAM
- Entradas digitales de 8 bits a través de un conmutador de 8 estaciones.
- 8 salidas digitales con LED para indicar los estados.
- 6 dígitos numéricos mostrados por display.
- 20 teclas alfanuméricas (pulsadores discretos) para la entrada de datos.
- Convertidor digital a analógico, 6 bits de resolución.
- Convertidor analógico digital, 6 bits de resolución.
- Un puerto de sonido, que consiste en un zumbador piezoeléctrico accionado por un contador programable, se pueden variar las frecuencias.
- Monitor de sistema operativo (MOS) EPROM.
- Operación de 7 a 10 Vcc de corriente.

Además, se le pueden agregar más opciones al Primer TRAINER puesto que la placa se presta para añadir más componentes:

- Memoria RAM Extra: Un 32K bytes de RAM estática o se puede instalar un disco RAM.
- Memoria RAM Extra / Reloj en tiempo real: El reloj se mantiene en tiempo de centésimas de segundos, segundos, minutos, horas, día de la semana, día del mes, mes y año, incluso cuando el Primer TRAINER está apagado. Ello compensa

automáticamente los años bisiestos, y mantiene el modo en tiempo militar o AM / PM.

- EPROM extra: El estudiante puede crear sus propios programas, o pueden añadirse otros sistemas operativos, incluyendo lenguajes de alto nivel como: E-FORTH, MTBASIC, etc.
- Puerto RS-232: Provee del Primer la posibilidad de conectar a una computadora personal.

Las siguientes placas periféricas se pueden conectar al bus de expansión llamado conector CN1:

- Línea Paralelo de 32 Placa E / S: Esto le da al usuario 32 líneas de E / S digital en dos Opto-22 E / S en conexiones rack compatibles. De las 32 líneas, sólo 8 son de salida y las otras 24 son programables como entradas o salidas. Las 24 líneas son ejecutadas utilizando el chip popular PPI 8255.
- Placa EPROM Programable: Permite la programación de dispositivos EPROM populares como el 27c512, 27c256, 27c128 y 27c64 con 12.5V y 21v con capacidad de programación de voltaje. Esta opción es requerida junto con la memoria RAM y puerto serie de actualización a fin de obtener el máximo provecho del menú de programación de interfaz EPROM. La documentación sobre esta interfaz se incluye al comprar la Placa EPROM programable.

Descripción de la unidad

El Primer TRAINER estándar es un kit que contiene piezas, un manual de montaje y manual de instrucciones. Los estudiantes y usuarios pueden montar la unidad utilizando el manual de ensamblaje del Primer TRAINER sin ayuda adicional. Una unidad ensamblada con éxito permitirá a los estudiantes/usuarios entrar en los programas de las unidades de memoria, permitiendo al estudiante/usuario ejecutar, y ver los resultados.

El Monitor del sistema operativo (el software estándar monitor de EPROM) incluido hará que el chip controlador del teclado permita explorar los botones del teclado, interpretar las entradas, y permitirá el almacenamiento de datos en los 256 bytes de RAM incluidos en el Primer TRAINER. El sistema operativo también mostrara datos relacionados a los seis (6) dígitos de display mostrados. El sistema operativo permitirá al estudiante examinar los datos almacenados en diversas ubicaciones de la memoria, examinar el contenido de los registros del microprocesador y otras funciones, a través de la pantalla y el teclado. El software del sistema operativo permitirá que el estudiante ejecute el programa introducido, y ejecute el programa paso a paso, instrucción por instrucción, para permitir la depuración de programas.

El manual de instrucciones contiene varias lecciones en las que el estudiante puede escribir programas para acceder y utilizar la E / S digitales, E / S analógicas y experimentar con el sonido del puerto, interrupciones y temporizadores. Todo lo que se necesita para realizar esto se incluye en el kit básico Primer TRAINER.

Hardware y Software Adicional

Fuente de Alimentación

Cualquier fuente de alimentación DC filtrada de siete (7) a diez (10) Voltios en DC puede ser utilizado para alimentar el Primer TRAINER. La fuente se alimenta a través de la unidad de potencia jack J1. Se debe asegurar de que la fuente de alimentación del conector de salida tiene un extremo positivo y uno negativo. El consumo de corriente será inferior a 500 mA. (350mA a 420mA típicamente).

Se debe tener cuidado de observar el tipo de tensión y la polaridad, o bien el Primer TRAINER puede ser gravemente dañado.

Memoria

El Primer TRAINER incorpora una memoria RAM de 256 bytes. El Monitor de sistema operativo utiliza unos pocos bytes en la parte superior de esta memoria. Esto deja más de doscientos bytes para el código que se puede introducir, que es bastante teniendo en cuenta que se introduce a mano. Sin embargo, si se desea una memoria RAM de mayor capacidad, como es el caso de los sistemas operativos de nivel superior, tales como E-EN-MT BASE, otros 32 Kbytes de RAM estática que pueden ser agregados. También una extensión del monitor del sistema operativo (EMOS) se puede instalar en el lugar de la estándar del monitor Sistema Operativo EPROM.

Teclas de Función de Datos

El Primer TRAINER estándar posee 20 teclas para seleccionar las funciones, e introducir los datos.

Puerto de Comunicación

Sin todas las opciones, el Primer TRAINER es una potente unidad autónoma que tiene el potencial de hacer algo más que ser un dispositivo de entrenamiento. Posee una interfaz fácil de usar que tiene el potencial, con un poco más de hardware añadido y convertirse en un ordenador/controlador. Uno de las adiciones al Primer TRAINER que ayuda a lograr esto es el puerto serie RS-232 (puerto de comunicación). Con esta opción de hardware instalado, el Primer TRAINER puede comunicarse con un terminal o un PC, que permite mayor flexibilidad. Esta opción es esencial cuando se usa un lenguaje de alto nivel EPROM en el Primer TRAINER.

Puerto de Expansión.

El puerto de expansión es un conector de 50 pines que proporciona acceso a los microprocesadores 8085 multiplexados en el bus de Dirección/Datos, el control de señales, y fuente de DC de Primer TRAINER. La conexión de otras tarjetas de circuito permite a este bus realizar un número casi ilimitado de funciones en el Primer TRAINER. El conector permite al Primer TRAINER poseer una interfaz con la mayoría de los periféricos, entre los que se destacan: Programador de memorias

EPROM y tarjetas digitales en paralelo de E / S en la placa. El conector tiene pines adicionales no utilizados por los periféricos de EMAC, pero que son útiles a los estudiantes que pueden diseñar periféricos propios

Puertos Externos de E/S

El conector para el puerto externo de E / S digital se puede añadir en cualquier momento para proporcionar acceso directo al nivel TTL de salida digital y líneas de entrada, a través de un cable. Estos puertos son de 8 bits cada uno, y están en paralelo con el interruptor DIP y salida LED. El conector del puerto E / S es de fácil acceso a las líneas de E / S por lo que el estudiante puede conectar interruptores y contactos de relé para las líneas digitales de entrada y los conductores de relé, relés de estado sólido (SSR) a la salida de líneas. Estas entradas y salidas se utilizan para permitir que el Primer TRAINER pueda controlar diversos dispositivos de muy interesantes proyectos.

4.2.3 Diagramas “Primer TRAINER 8085”

A continuación se muestran algunos diagramas del entrenador Primer 8085 y la identificación de algunos componentes que posee el entrenador. El entrenador está diseñado con una gran variedad de componentes electrónicos.

Los capacitores que han sido utilizados son: disco, monolítico y radial de plomo electrolítico.

Identificación de los capacitores:

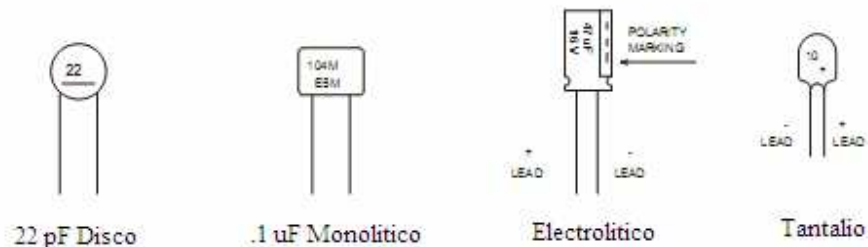
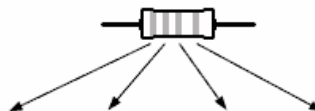


Fig. 4.2. Capacitores del Entrenador

Códigos y series de las Resistencias

► Código de colores ► Resistencias SMD ► Series normalizadas ► Simbología

Código de colores



Colores	1ª Cifra	2ª Cifra	Multiplicador	Tolerancia
Negro		0	0	
Marrón	1	1	$\times 10$	$\pm 1\%$
Rojo	2	2	$\times 10^2$	$\pm 2\%$
Naranja	3	3	$\times 10^3$	
Amarillo	4	4	$\times 10^4$	
Verde	5	5	$\times 10^5$	$\pm 0.5\%$
Azul	6	6	$\times 10^6$	
Violeta	7	7	$\times 10^7$	
Gris	8	8	$\times 10^8$	
Blanco	9	9	$\times 10^9$	
Oro			$\times 10^{-1}$	$\pm 5\%$
Plata			$\times 10^{-2}$	$\pm 10\%$
Sin color				$\pm 20\%$

■ Ejemplo:



Si los colores son: (**Marrón** - **Negro** - **Rojo** - **Oro**) su valor en ohmios es:
 $1 \quad 0 \quad \times 100 \quad 5\% = 1000 \Omega = 1K\Omega$
 Tolerancia de $\pm 5\%$

5 bandas de colores

■ También hay resistencias con 5 bandas de colores, la única diferencia respecto a la tabla anterior, es que la tercera banda es la 3ª Cifra, el resto sigue igual.

► **Descargue (CodRes.exe)** Programa freeware para el cálculo de las resistencias, cortesía de Cesar Pérez.

Fig. 4.3. Códigos de Resistencias.

Circuitos Misceláneos

Algunos circuitos misceláneos para agregar al Primer TRAINER son:

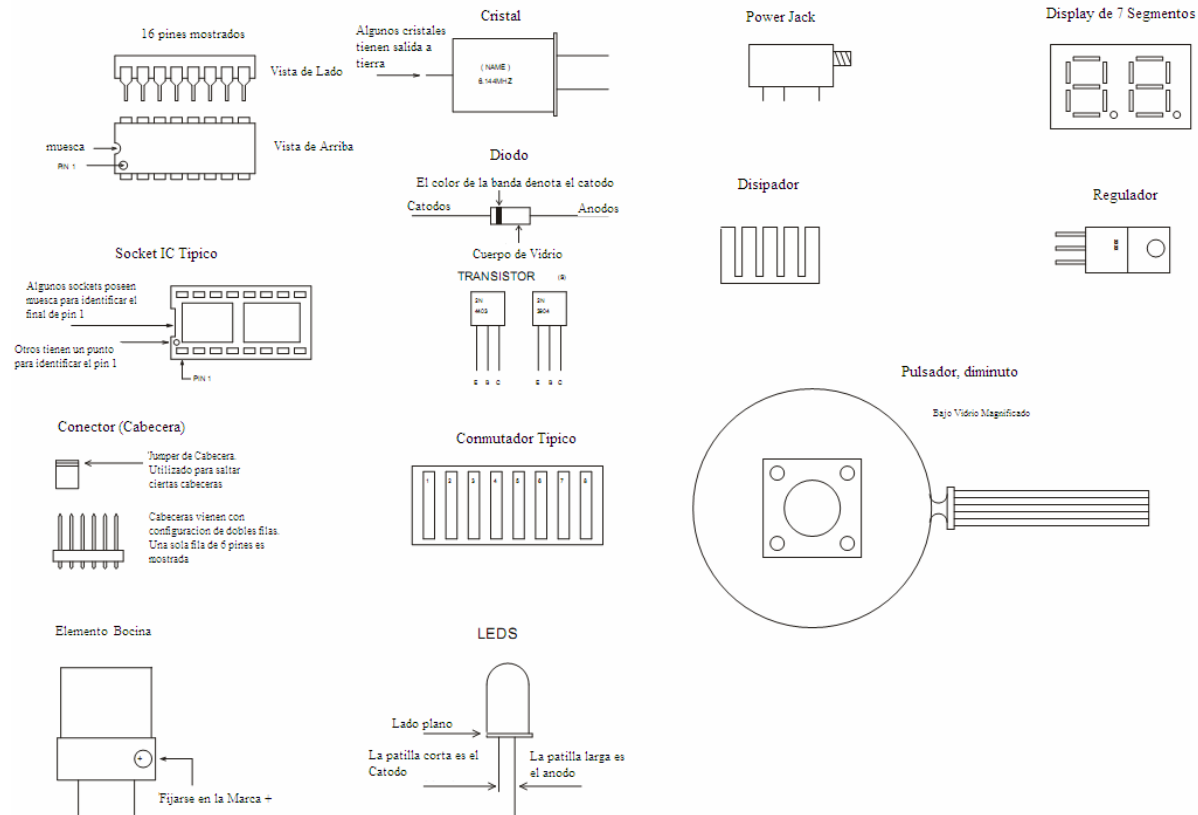


Fig. 4.4. Circuitería del Primer TRAINER

Dibujo de Ensamblaje del Primer TRAINER 8085

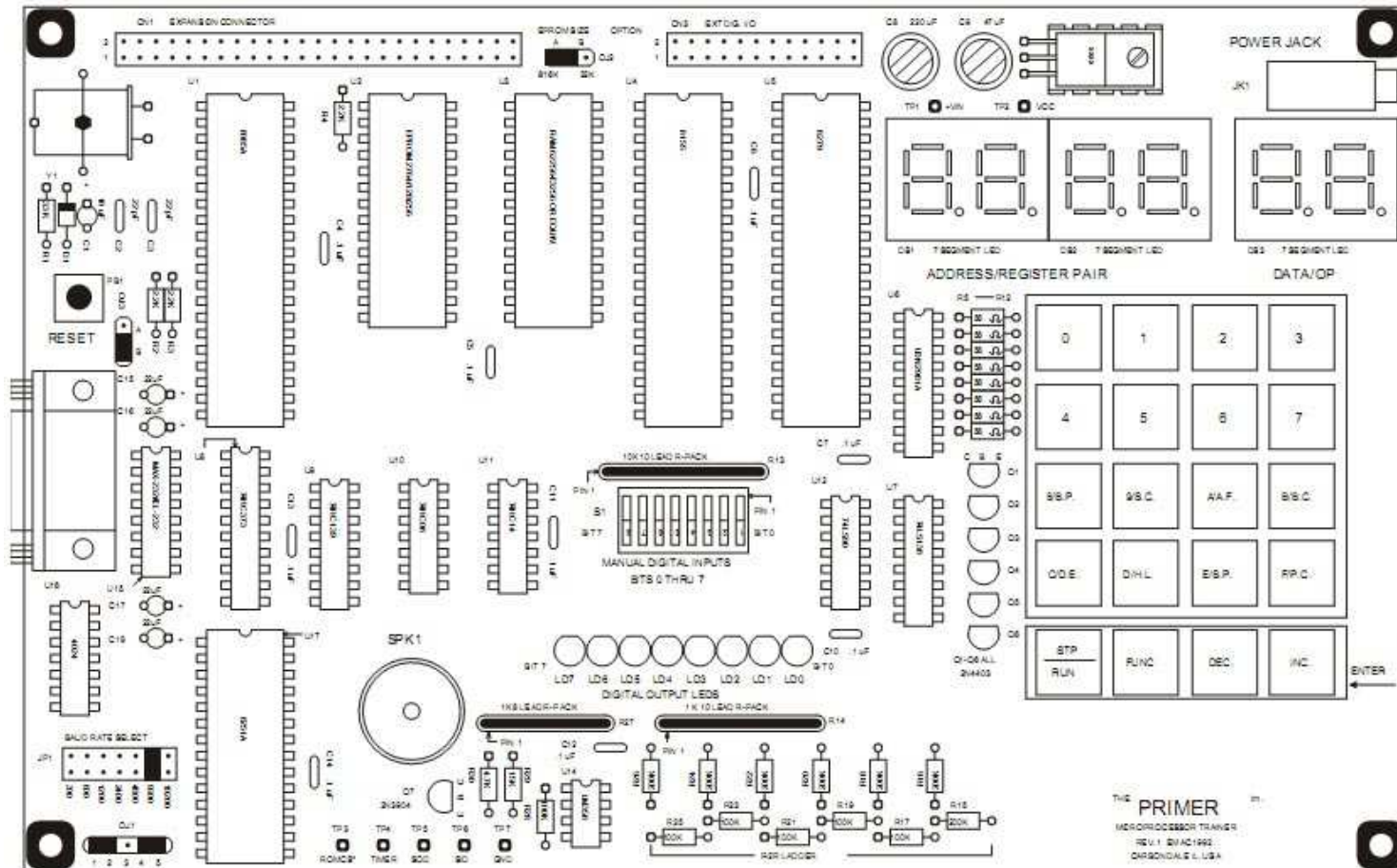


Fig. 4.5 Ensamblaje del Primer TRAINER 8085

Luego de instalados los elementos en la placa, el paso final para realizar el ensamblaje es el instalar las puntas de goma en el Primer TRAINER. Este tiene que ser el último paso a realizar para el ensamblaje y la placa del circuito debe estar limpia y seca. El flujo removedor (si es utilizado) disolverá el adhesivo de las puntas de goma.

Representan el lado de la soldadura de la placa del Primer TRAINER, con las puntas de goma en su lugar. Las posiciones son aproximadas, puesto que no hay puntos de montaje para ellas. Un posicionamiento no es tan importante, puesto que estas puntas no son conductivas y solamente sirven para agregar espacio en la placa del Primer TRAINER. Luego de ensamblada completamente la placa deberá ser probada.

4.2.4 Simulador de Microprocesador Intel 8085 Versión 1.0



Fig. 4.6. Ventana Acerca del Simulador

Un simulador del microprocesador 8085 es un programa que se comporta de igual forma que el procesador simulado.

Un simulador del microprocesador 8085 al nivel de programación muestra el comportamiento de las instrucciones. Permitiendo conocer los resultados de las

instrucciones sobre los diferentes elementos sobre los que actúan (memoria, registros, puertos, etc.) en todo momento.

Como elemento didáctico es sorprendente, ya que también dispone de una multitud de dispositivos que pueden emplearse en prácticas. Estos dispositivos son similares a los de PC, de esta forma, se hace mucho más atractiva la programación sobre este tipo de procesadores antiguos, que suelen ser menospreciados por su “poca” potencia. La posibilidad de construir dispositivos con un microprocesador y un programa embebidos está al alcance de la mano. El estudiante no solo puede comprobar cómo funciona su programa a través de los dispositivos, sino que además puede interactuar con él.

Si lo que se desea es programar aplicaciones para 8085 porque este procesador se vaya a utilizar en un dispositivo real, el simulador dispone de una gran cantidad de herramientas orientadas. Es posible introducir puntos de ruptura, saltar llamadas a subrutina, simplificar expresiones, etc.

4.2.4.1 Requerimientos e Instalación del simulador

Configuración mínima recomendada:

- Windows 95/98.
- PC con procesador a 166 MHz
- 32 MB de RAM.
- Lector de CD-ROM de 8x o superior.
- 10 MB de espacio libre en su disco duro.

Conociendo ya las características técnicas que debe tener la máquina, se va a comenzar la instalación de Simulador 8085 en el sistema. Para ello bastará con ejecutar el instalador, lo que causará la auto ejecución del programa de instalación.

Realizado el paso anterior, aparecerá en pantalla una ventana. En la (figura 4.7) puede ver el aspecto que muestra la ventana de instalación del simulador.



Figura 4.7. Ventana Inicial del Programa de Instalación.

Pulsando en botón Examinar que aparece en la figura 4.7 puede escoger la ubicación de los ficheros que forman el programa. Por defecto se utiliza el directorio Archivos de Programa \ simulador 8085 de la unidad C del computador. Debajo de la ubicación se indica el espacio requerido por el programa y el espacio disponible en su disco duro.

Para salir de la instalación de Simulador 8085 pulse el botón Salir que aparece en la parte inferior derecha de la ventana.

Pulse el botón Comenzar para iniciar el proceso de instalación. Se inicia entonces la copia de ficheros en el equipo. Tal y como muestra la (figura 4.8), el programa de instalación irá indicando en todo momento el curso del proceso.

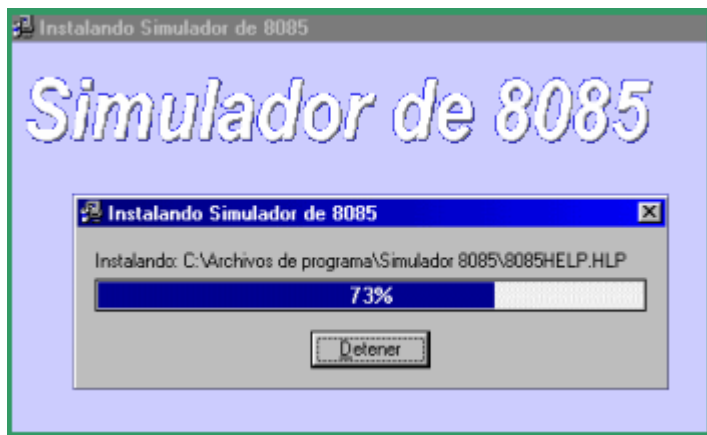


Figura 4.8. Proceso de Copia de Archivos Al Camino de Destino.

Cuando la instalación haya llegado a su fin verá aparecer en pantalla una nueva ventana con información. Aparece también una casilla marcada indicando que, una vez terminada la instalación, el Simulador 8085 se ejecutará. Si desea que no se inicie el simulador, desactive la casilla.

Dando clic en el botón Aceptar le llevará al término de la instalación, quedando la carpeta de Simulador 8085 tal y como se muestra en la (figura 4.9). Ya puede ejecutar Simulador 8085, simplemente haciendo doble click sobre el icono correspondiente.



Figura 4.9. Carpeta de Archivos de Simulador 8085.

En la sección de anexos y en el CD de Manuales podrá encontrar un manual extenso y detallado del Simulador 8085, ofreciendo de esta manera una herramienta más para facilitar al estudiante el aprendizaje y permitiéndole avanzar en el estudio del Microprocesador 8085.

4.2.5 Programa Ensamblador

El término ensamblador (del inglés assembler) se refiere a un tipo de programa informático que se encarga de traducir un fichero fuente escrito en un lenguaje ensamblador, a un fichero objeto que contiene código máquina, ejecutable directamente por la máquina para la que se ha generado. El propósito para el que se crearon este tipo de aplicaciones es la de facilitar la escritura de

programas, ya que escribir directamente en código binario, que es el único código entendible por la computadora, es en la práctica imposible. La evolución de los lenguajes de programación a partir del lenguaje ensamblador originó también la evolución de este programa ensamblador hacia lo que se conoce como programa compilador.

Algunos tipos de archivos que contienen código ensamblador se listan a continuación:

*.ASM

Archivo fuente. Este es el archivo de texto que contiene las instrucciones, variables, constantes, etc. que describen el programa que deseamos. Este archivo debe ser creado con un editor de textos (no procesador de palabras), tal como EDIT.EXE de MSDOS, o NOTEPAD.EXE de Windows.

*.DBG

Archivo Depurador (DEBUG). Este archivo es creado por el ensamblador y es utilizado con un simulador específico el cual no tenemos, por lo que este archivo no sirve para ningún fin práctico.

*.LST

Archivo Listado. Este archivo es creado por el ensamblador y nos indica en qué dirección se ensambla cada instrucción; cuales son los octetos (bytes) que representan una instrucción; los lugares donde hay un error; etc.

*.HEX

Archivo Hexadecimal. Este archivo es creado por el ensamblador, está escrito en formato Intel Hex y contiene la información que será almacenada en la memoria del microprocesador.

Este es el archivo que se programa en el microprocesador o la memoria EPROM.

4.2.5.1 Pasos a seguir para convertir un archivo .ASM a .HEX a través del programa ensamblador.

Para poder realizar la conversión a archivo objeto (lenguaje de maquina) se deberán realizar los siguientes pasos en un sistema Windows:

Inicio > Ejecutar



Figura 4.10. Menú Inicio

Luego se debe digitar cmd. A continuación aparecerá la línea de comandos.

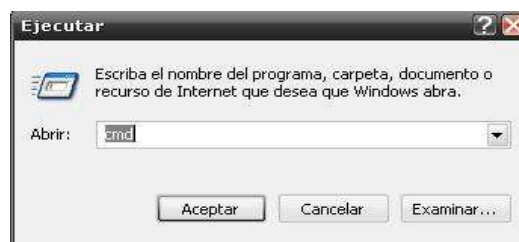
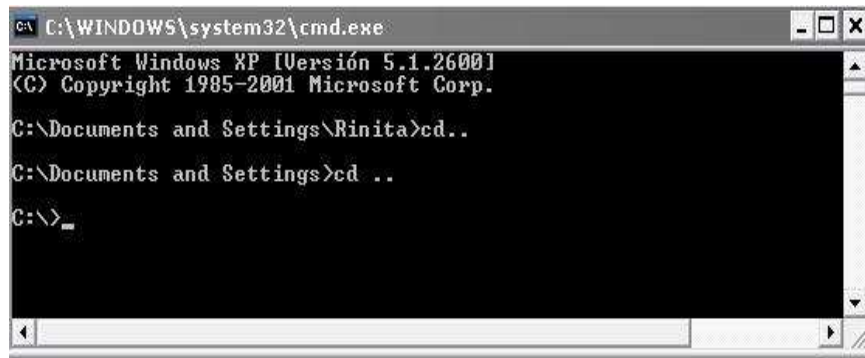


Figura 4.11. Inicio de Consola de Comandos

Para poder ejecutar el programa ensamblador (asm85.exe) se debe acceder a la ruta donde se encuentra el ejecutable. Para poder moverse entre las carpetas en la línea de comando se debe utilizar el comando CD. Para bajar un nivel cd o acceder a una carpeta cd /8085 (esto es puesto que la carpeta se encuentra en c:/8085)



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Rinita>cd..
C:\Documents and Settings>cd ..
C:\>_
```

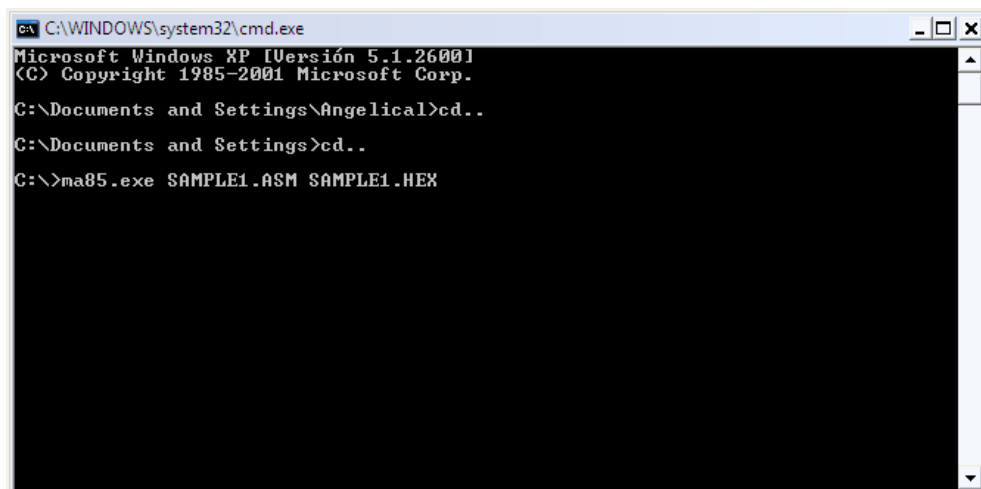
Figura 4.12. Consola de Comandos de Windows

Ya estando dentro de la carpeta se deberá seguir la siguiente sintaxis:

```
C:\>ma85.exe [archivo-a-convertir.asm] [archivo-convertido.hex]
```

Ejemplo:

```
C:\> ma85.exe [EJEMPLO1.ASM] [EJEMPLO1.HEX]
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Angelical>cd..
C:\Documents and Settings>cd..
C:\>ma85.exe SAMPLE1.ASM SAMPLE1.HEX
```

Figura 4.13. Ejemplo de Creación de Un Archivo Tipo Hex

El o los archivos convertidos se pueden encontrar en la misma carpeta en la cual se encuentra la línea de comandos.

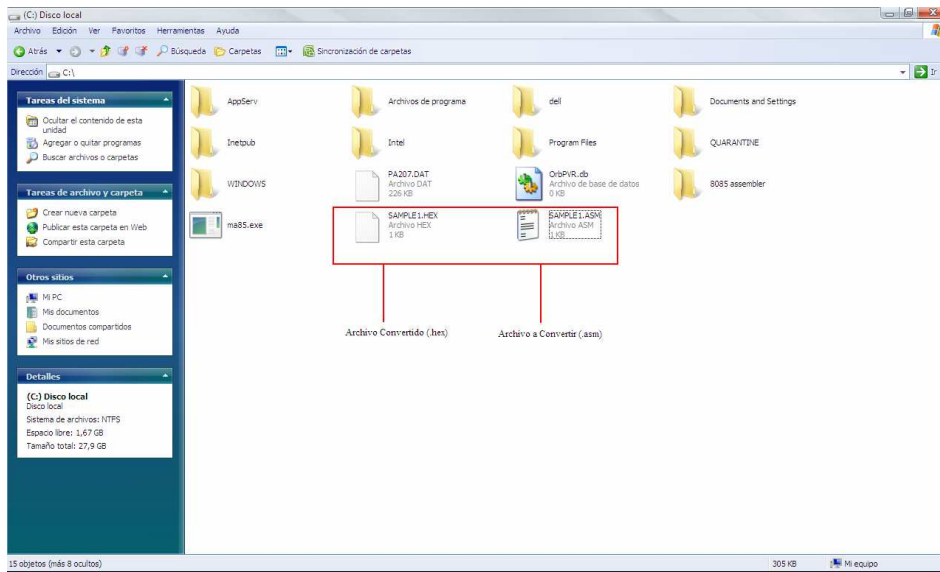


Figura 4.14. Visualización de Archivos Creados

4.2.6 Cargar un Archivo Hexadecimal al HYPERTERMINAL para Windows.

EL HYPERTERMINAL es una aplicación dentro de Windows que le permite al equipo entrenador comunicarse con dispositivos en serie. Cuando un carácter es recibido por el puerto serial del PC que se está siendo usado por el HYPERTERMINAL, el carácter será mostrado en la ventana de la aplicación y cuando una tecla sea presionada, el carácter correspondiente será enviado vía el puerto serie.

Copiar el archivo de configuración del HYPERTERMINAL del disco de distribución ("PRIMER_95-98.HT" para Win95/98 o " PRIMER_XP-2K-NT.HT " para WIN XP, 2000 o NT) al escritorio de Windows luego hacer doble clic sobre en dicho icono. En el HYPTERTERMINAL deberá determinar si se quiere instalar un modem, se debe hacer clic en la opción de "ninguno".

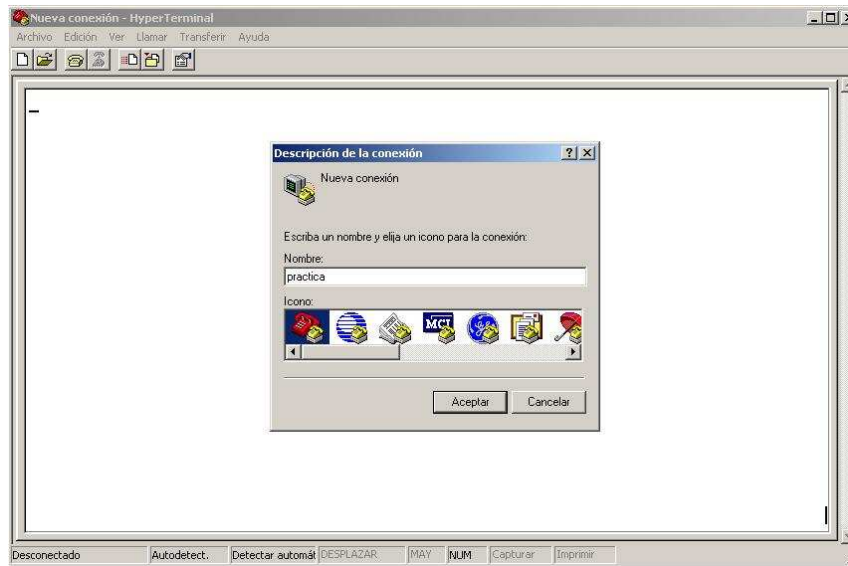


Figura 4.15. Inicio de Sesión Del Hyperterminal

Se debe preparar el Primer TRAINER para 9600 Baudios con las colocaciones siguientes de los jumpers:

JP1 = 9600

OJ1 = jumper 1-2 y 3-4

OJ2 = jumper b

OJ3 = jumper b

Los archivos de configuración usaran COM1 como el puerto de serie. Si COM1 se encuentra disponible en el PC y no esté siendo utilizado para otro dispositivo tal como el mouse.

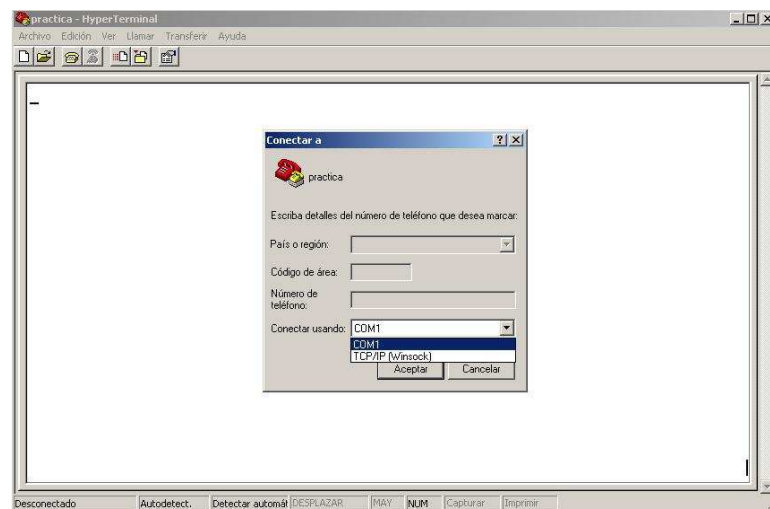


Figura 4.16. Selección del Puerto a Utilizar

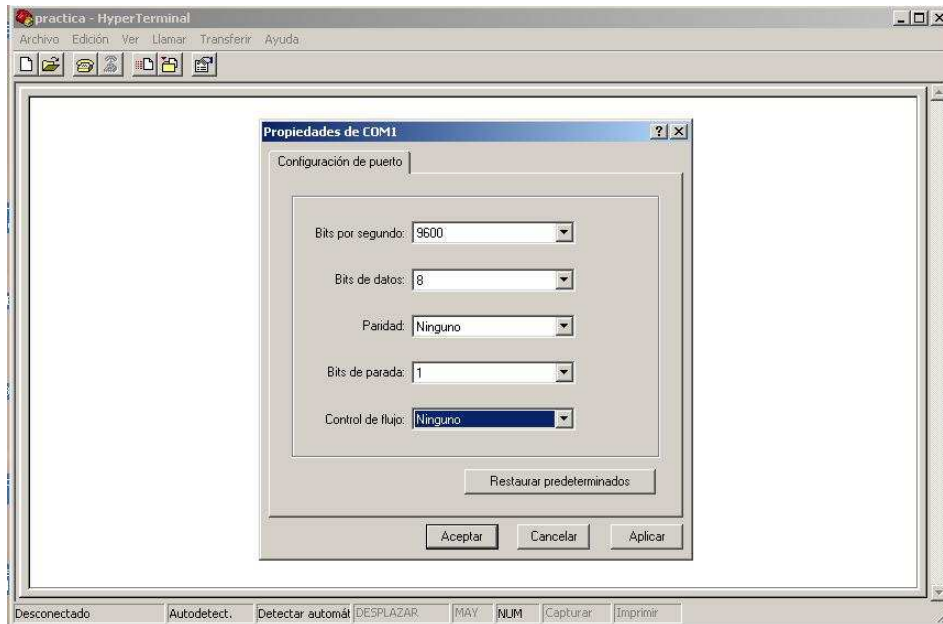


Figura 4.17. Configuración de Datos de Inicio de Sesión

Si el puerto COM1 está siendo ya usando en el PC, se necesitará apretar el botón de desconectar (En la parte superior izquierda de la ventana terminal, mostrado como un teléfono con el auricular levantado sobre el teléfono).

Escoja las propiedades de archivo, luego en la ventana que aparece, se puede encontrar la pestaña que se llama Conectar a. Si la pestaña se encuentra en primer plano, se observará una ventana de selección llamada Conectar Usando. Usar esta opción para escoger un puerto de COM que no esté siendo usado en el PC y apretar el botón OK. Luego apretar el botón de llamada (junto al botón de desconectar) para establecer la conexión. Escoger Guardar Archivo para salvar la configuración actual.

Utilizando el cable serial (que es incluido en el equipo), conectar el cable serial del entrenador Primer al puerto COM para el cual ha sido configurado el HYPERTERMINAL.

Restablecer el Primer y apretar las teclas "Func." y "0" para invocar EMOS. Se debería de mostrar el siguiente menú en el HYPERTERMINAL:

E-FORTH V1.10 SBC+
Copyright 1987-91 EMAC inc.
Licencia no. 00.

EMOS V2.00 HELP MENU

B --> Bring Block from RAMDISK to memory
C --> Change register contents
D --> Dump memory contents
E --> Edit memory contents
F --> Fill memory with byte
G --> Go execute program { full speed }
H --> Hex/dec math { 1st + 2nd, 1st - 2nd }
I --> Input from I/O port
L --> List memory contents using mnemonics
M --> Move section of memory
O --> Output to I/O port
R --> display Register contents
S --> MOS Service call
T --> Trace program execution
W --> Write memory to RAMDISK block
Z --> Zap application EPROM
< --> hex download from TRAINER to host
> --> hex upload to TRAINER from host
? --> display this help menu

En español las opciones son las siguientes:

E-FORTH V1.10 SBC+
Copyright 1987-91 EMAC inc.
Licencia no. 00.

EMOS V2.00 HELP MENU

B --> Trae el bloque de RAMDISK a la memoria
C --> Cambiar contenidos de registro
D --> Eliminar contenidos de memoria
E --> Editar los contenidos de memoria
F --> Llenar la memoria con el Byte
G --> Correr Programa Ejecutable {Velocidad Máxima}
H --> Matemáticas en Hex/dec {1ro + 2do, 1ro - 2do}
I --> Ingresar desde el Puerto de Entrada/Salida
L --> Listar los contenidos de memoria usando mnemotecnia
M --> Mover Sección de Memoria

- O --> Salida al Puerto de Entrada/Salida
- R --> Mostrar contenidos de Memoria
- S --> Servicio de llamada MOS
- T --> Rastreo de programa ejecutable
- W --> Escribir memoria al bloque RAMDISK
- Z --> Aplicación Zap EPROM
- < --> Descarga de hex desde entrenador al host
- > --> Cargar hex al entrenador desde el host
- ? --> Desplegar el menú de ayuda

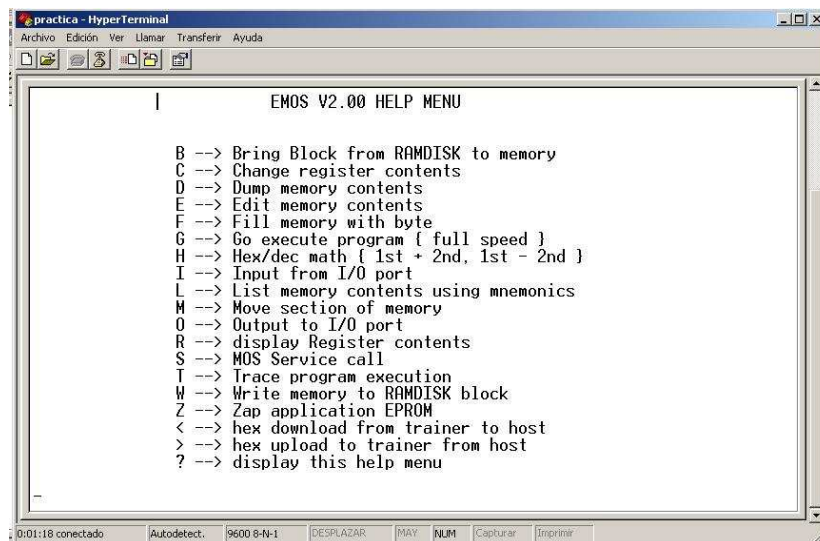


Figura 4.18. Menú del Hyperterminal.

Ahora deberá digitar ">" para cargar en el entrenador y se desplegará el siguiente mensaje:

 - > STARTING ADDRESS..

Digitar " FF01 " para este ejemplo y se desplegará lo siguiente:

 - > STARTING ADDRESS.. FF01

READY TO RECEIVE, <ESC> TO ABORT

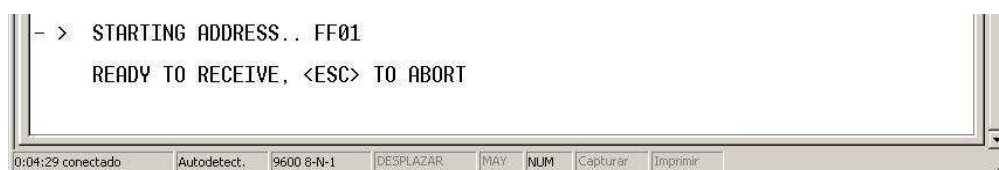


Figura 4.19. Mensaje de Inicio de Sesión

Ahora se podrá empezar a descargar el archivo hex INTEL; seleccionando desde la terminal en el HYPERTERMINAL la opción de "Transferir" y seleccionar "Enviar un Archivo de Texto. Esto mostrará una ventana para la selección del archivo, se debe cambiar el tipo de archivo a la opción para mostrar todos los archivos (*.*) y escoger el archivo TRAJNTST.HEX que está en el mismo directorio que el archivo de configuración del HYPERTERMINAL.

La transferencia sucederá sin ninguna realimentación hasta que termine. Este archivo tomará cerca de 2 segundos para cargar entonces el "-" prompt reaparecerá.



Figura 4.20. Selección de Archivo a Cargar en el Hyperterminal.

Digitar "G" de "GO" para ejecutar el programa y aparecerá lo siguiente:

```
-----  
- G STARTING ADDRESS..  
-----
```



Figura 4.21. Comando de Carga de Archivo.

Escribir "FF01" para la dirección de inicio y cuando el cursor vuelva a cargar para un punto de ruptura (Breakpoint), sólo se debe presionar Enter (para ningún punto de ruptura) y se mostrará lo siguiente antes de que el programa inicie:



Figura 4.22. Inicio de Carga de Archivo.

```
-----
- G STARTING ADDRESS.. FF01  BREAKPOINT ADDRESS..

RUNNING.. HIT ANY KEY TO STOP
-----
```

Algo similar a lo siguiente será mostrado en la terminal:

```
-----
THE DIPSWITCH NOW READS

0087
-----
```

El número mostrado será distinto basado en las posiciones en las que se encuentran los interruptores DIP.

El programa existe cuando los interruptores DIP han sido movidos de tal manera que se leen "0000". Luego de los datos de salida, el prompt retornará y la terminal mostrará algo similar a lo siguiente.

```
-----
THE DIPSWITCH NOW READS

0000
SZxAxPx C A B C D E H L TS SP PC OP CODE MNEMONIC
01000100 00 01 03 00 0D 00 00 0000 FFD4 FF42 FF RST 7
-----
```

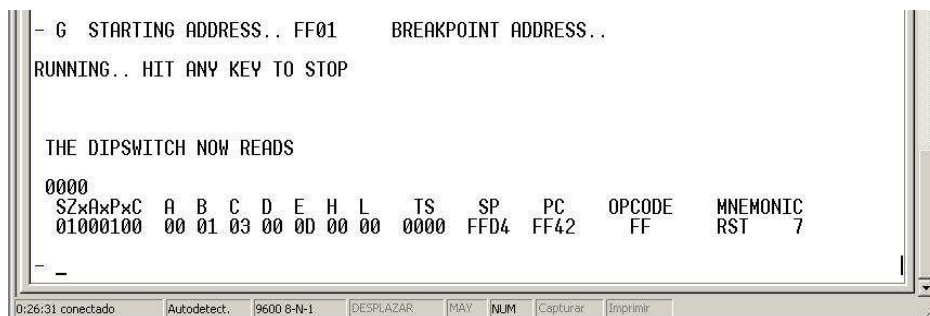


Figura 4.23. Salida de Pantalla de Carga de Archivo.

Para salir del HYPERTERMINAL solo se debe elegir la opción Salir del Menú Archivo

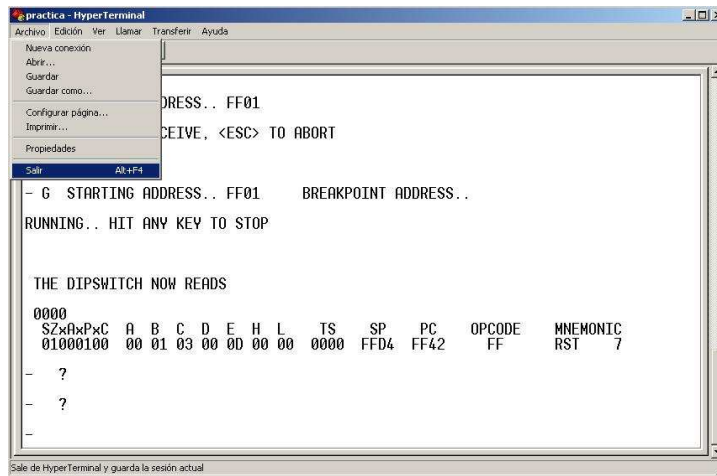


Figura 4.24. Salida del Hyperterminal.

4.3 MEDIDAS DE SEGURIDAD Y RECOMENDACIONES EN EL MANEJO DEL EQUIPO ESPECIALIZADO PARA EL DESARROLLO DE PRÁCTICAS DE LABORATORIO CON MICROPROCESADORES

4.3.1 Precauciones Generales

Al manejar dispositivos electrónicos se deben de tomar en cuenta ciertas consideraciones especiales como las siguientes:

- Antes de manipular los dispositivos electrónicos que contengan electricidad verificar que estos se encuentren debidamente desconectados.
- Cuidar los materiales y equipo que se le provean en el laboratorio, así como también respetar las normas establecidas dentro de este.
- Verificar las instrucciones que se encuentran plasmadas en los manuales de los dispositivos antes de manipularlos.
- Los dispositivos electrónicos son muy sensibles, por ello se deben tratar con delicadeza y sin brusquedad.
- Existen diferentes implementos de limpieza como el aire comprimido que ayuda a remover el polvo de las superficies de forma más efectiva que cuando se hace con una franela.
- Debido a la gran cantidad de estática que se encuentra en el ambiente se deben de tomar ciertas medidas para evitar el daño de los componentes electrónicos como lo es el uso de una pulsera anti-estática conectada a un chasis metálico antes de cualquier manipulación de estos.
- Desconectar y apagar siempre en forma ordenada todos los equipos cuando estos ya no estén siendo utilizados para evitar daños.
- Mantener en forma segura todas las instalaciones eléctricas, conector y equipo electrónico.
- No utilice el equipo cerca de aparatos de televisión, radios, equipos estereofónicos, teléfonos móviles ni dispositivos eléctricos de cualquier otro tipo ya que estos aparatos provocan ruidos que podrían dañar los equipos.
- Colocar siempre los equipos en una superficie estable para evitar caídas accidentales y daños.

4.3.2 Condiciones Recomendadas del Lugar de Trabajo

Se deben de tomar en cuenta ciertas condiciones a la hora de establecer el lugar donde se desarrollarán las prácticas de laboratorio con microprocesadores.

- Con suficiente ventilación, de preferencia con aire acondicionado para evitar el exceso de calor (especialmente el provocado de forma natural por los rayos del sol).
- Aislado de fuentes de vibración y ruido que puedan causar molestias al equipo y desarrollo del trabajo.
- Que sea un lugar limpio y ordenado en el que no exista alguna fuente de polvo o algún otro material que pueda provocar daños al equipo.
- Que contenga las conexiones eléctricas apropiadas (tomacorrientes con conexiones a tierra y nivel de voltaje adecuado).
- Aislado de campos electromagnéticos que puedan causar daños al equipo.
- Que cuente con una temperatura optima para el manejo de dispositivos electrónicos (ni muy caliente ni muy frío).
- Que cuente con suficiente espacio para maniobrar con los dispositivos de forma segura y cómoda.

4.3.3 Recomendaciones a los Auxiliares de Cátedra e Instructores

- El encargado de impartir las prácticas de laboratorio debe de orientar a los estudiantes acerca de las recomendaciones del uso del equipo al comenzar la práctica, así como también acerca de los lineamientos y normas de seguridad establecidas dentro del laboratorio.
- Preparar el equipo necesario para el desarrollo exitoso de la práctica de laboratorio.
- Asistir y orientar a los estudiantes acerca del manejo de los dispositivos electrónicos con el fin de evitar daños o pérdidas del equipo.
- Salvaguardar el orden dentro del laboratorio con el fin de que la práctica se pueda realizar de forma ordenada.
- Será la autoridad dentro del laboratorio, así que podrá llamar la atención de los estudiantes que no estén haciendo un uso adecuado del equipo o lo estén dañando.

- Apoyarse del material didáctico en caso de cualquier duda que se presente en la realización de la práctica de laboratorio.
- Preparar con anticipación todo el material y equipo a utilizar en la práctica de laboratorio con el objetivo de optimizar tiempos.
- Deberá verificar el estado del equipo antes, durante y después de la utilización del mismo.
- Este debe estar presente durante la realización de la práctica de laboratorio y si surgiere algún imprevisto por el cual deba ausentarse debe de solicitar permiso al docente encargado.
- Este será el encargado de darle mantenimiento al equipo siempre que lo necesite, así como también de reportarlo ya sea de forma escrita por medio de bitácoras o simplemente informando al docente encargado.

4.3.4 Recomendaciones a los Docentes

- Proporcionar y facilitar a los auxiliares o instructores de la materia toda la información necesaria para operar de forma eficiente en los laboratorios, tal información incluye: horarios de laboratorio, cantidad adecuada y distribución de los estudiantes dentro del laboratorio, contenido de prácticas, lista de equipo y materiales existentes, así como toda información relacionada con el uso de los equipos y el laboratorio.
- Velar porque las prácticas de laboratorio estén siendo impartidas conforme al contenido establecido para la materia por parte de la UES-FMO.
- Tener un control de asistencia de los auxiliares o instructores encargados de las prácticas de laboratorio y estar pendientes de que las prácticas de laboratorio se desarrollen de forma continua y sin imprevistos.
- Conocer y manejar el equipo especializado con el que se cuente en el laboratorio de prácticas con el fin de velar que este siendo utilizado correctamente para evitar su deterioro, así como también para brindar consultas o ayuda a los auxiliares o instructores de la materia.

- Proporcionar tanto a los estudiantes como a los auxiliares e instructores las medidas de seguridad y recomendaciones a utilizar en el laboratorio con el fin de evitar accidentes o daños al equipo.

4.3.5 Recomendaciones a los Estudiantes

- Antes de comenzar las prácticas de laboratorio el estudiante debe de conocer todas las medidas de seguridad y recomendaciones establecidas para el uso del equipo y trabajo en el laboratorio.
- Debe asistir a las prácticas de laboratorio cumpliendo con el horario previamente establecido para la realización de las prácticas y deberá acatar todas las recomendaciones e instrucciones que ahí le sean impartidas por parte del auxiliar o instructor de la materia.
- Debe de llevar todos los materiales o elementos que le sean pedidos con anterioridad para la realización de prácticas especiales.
- Para hacer uso del laboratorio y del material didáctico del mismo, este deberá presentar al encargado del laboratorio su carné de estudiante de la UES-FMO de forma vigente.
- Es responsable de cuidar y hacer un buen uso de los materiales y equipos proporcionados para la realización de prácticas, ya que cualquier daño o extravío de estos tendrán que ser repuestos por parte del estudiante.
- Debe de realizar las prácticas en el tiempo y la forma establecida por el encargado del laboratorio.
- Al inicio y al final de la práctica este debe de revisar que todos los materiales y equipos que le hayan sido proporcionados se encuentren en buen estado. Y si se presenta alguna anomalía deberá informar de forma inmediata al encargado del laboratorio
- No debe de ingerir ninguna clase de alimentos ni bebidas en el laboratorio ya que podrían causar deterioro o daño de algún equipo.
- Se prohíbe la utilización de dispositivos que distraigan la atención de estos como teléfonos celulares, ipods, cámaras digitales, reproductores de sonido y video. (Al menos que la práctica lo requiera).

- Al terminar la utilización de los equipos y el laboratorio debe de entregar todos los materiales y equipos utilizados al encargado de las prácticas, así como también dejar el área de trabajo de forma ordenada y limpia.
- Podrá hacer uso de permanencia en el laboratorio siempre y cuando cuente con la autorización del docente, auxiliar o instructor encargado de la materia.

4.3.6 Del Funcionamiento de los Laboratorios

Se debe de llenar una requisición especificando los materiales necesarios para la realización de las prácticas de laboratorios, con el fin de prepararlos para cualquier actividad que se realice dentro de los laboratorios.

- Clasificar el equipo de prácticas de laboratorio de hardware que se encuentra dentro este, el cual deberá contener su bitácora respectiva, en las cuales se realizaran las anotaciones necesarias cuando el equipo sea utilizado.
- Especificar una parte de averías estándar del equipo de prácticas de laboratorio de hardware para el mantenimiento justificando la elección de cada uno de sus apartados, explicando la utilidad que tienen las hojas de servicio de averías interno.
- Realizar la respectiva clasificación de los distintos procedimientos a seguir para el diagnóstico y localización de averías.
- Exponer por lo menos un procedimiento diseñado para optimizar el proceso de mantenimiento de equipos de microprogramación.

4.3.7 Reglas Básicas de Seguridad

- Efectuar un mantenimiento periódico a las herramientas para que se encuentren en buen estado al momento de ser utilizadas para el desarrollo de los laboratorios.
- Examinar las herramientas antes de utilizarlas para ver si tienen alguna avería o se encuentran dañadas y así poder utilizarlas de acuerdo con las instrucciones de fábrica.
- Elegir y utilizar el equipo de protección apropiada.

4.3.8 Para Programar los Microprocesadores

Se debe de ser cauteloso a la hora de operar los microprocesadores, ya que se considera casi inevitable, que luego de trabajar un par de semanas con ellos llegar a obtener demasiada confianza, por tal razón es necesario realizar las precauciones mínimas necesarias, entre estas se encuentran:

- Es necesario desenergizar el dispositivo.
- Verificar que el dispositivo este correctamente colocado y conectado.
- No se debe exponer el dispositivo a altas temperaturas.

Antes de manipular el dispositivo, se debe de tocar una superficie que sea derivada a tierra o colocarse una muñequera anti-estática.

Se debe de ubicar el equipo en su base con gran cuidado al momento de insertarse, puesto que se debe de comprobar que todas las pines estén alineadas con su respectivo agujero y así poder evitar daños al equipo.

4.3.9 La Electricidad Estática

La electricidad estática es un fenómeno que se debe a una acumulación de cargas eléctricas en un objeto. Esta acumulación puede dar lugar a una descarga eléctrica cuando dicho objeto se pone en contacto con otro. Para la mayoría de las personas, la electricidad estática no es más que algo molesto.

4.3.9.1 Precauciones

Para evitar las posibles averías que pueden producirse en el equipo al intentar manipularlo es conveniente tomar las siguientes medidas:

- En primer lugar, tocar durante unos segundos con la mano una superficie de metal que disponga de conexión a tierra. Con esto conseguiremos igualar el potencial de nuestro cuerpo con el equipo que vamos a manipular.
- Aumente la humedad relativa del entorno y ubique los componentes en una superficie conectada a tierra.

- Cuando se inserte el equipo en la base, es aconsejable estar apoyando al menos uno de los antebrazos en una superficie metálica. De esta forma, difícilmente se producirán problemas de estática.
- Procure no tocar las pines, los contactos ni los circuitos del equipo que está manipulando.
- Utilice siempre un método de conexión a tierra adecuado cuando toque un componente o una unidad sensible a la electricidad estática.
- Instale elementos de prevención específicos para electricidad estática, como alfombrillas de tierra.

Tomando estas precauciones correctamente se disminuyen los riesgos de una forma importante tanto para la persona como el equipo mismo.

Además existen condiciones que aumentan la generación de la electricidad, lo que puede contribuir a la acumulación de electricidad estática en el entorno. Entre estas condiciones se encuentran:

- La rapidez con la que toca, conecta o desconecta dispositivos electrónicos.
- Baja humedad relativa.
- Si bien se debe de tomar siempre las precauciones necesarias para descargar electricidad estática, si se encuentra en un ambiente en el que se percibe la existencia de electricidad estática, deberá tomar precauciones adicionales para proteger el equipo electrónico.

4.3.10 Prevenir los Sigüientes Lugares para la Operación e Instalación de los Dispositivos Electrónicos

- Lugares en los que los cambios en la temperatura ambiente puedan producirse tan rápidamente como para provocar condensación.
- Donde la temperatura ambiente pueda superar el rango comprendido entre 0 y 50 °c y en los que la humedad ambiente pueda superar el rango comprendido entre 45 y 85% de humedad relativa mientras el controlador este funcionando.
- Lugares en los que el calor pueda acumularse debido a la radiación del instrumento.

- Sitios en los que el equipo esté sometido directamente a vibraciones y golpes lo cual llegaría a provocar problemas en el funcionamiento.
- Lugares en donde el equipo este expuesto a polvo, aire salado o aire que contenga partículas de hierro, además de interferencias con la electricidad estática, ruidos y magnetismo.
- Lugares en los este expuesto a interferencias con la electricidad estática, magnetismo y ruidos.

4.5 PRESUPUESTO DEL PROYECTO.

La siguiente tabla muestra el detalle de los gastos que conlleva el desarrollo e implementación del proyecto.

CANTIDAD	DESCRIPCIÓN	PRECIO UNITARIO	PRECIO TOTAL
Adquisición de Equipo			
1	K Primer Pak TRAINER 8085	\$ 414.00	\$ 414.00
1	Tabla Base para Equipo	\$ 5.00	\$ 5.00
Costo de Envío de Equipo desde el Exterior del País			
	Transferencia Monetaria para Adquisición de equipo	\$ 20.00	\$ 20.00
	Costo de Envío de Equipo desde el Interior de E.E.U.U.	\$ 15.00	\$ 15.00
Gastos de Papelería			
8	Resmas de Papel Bond Tamaño Carta Base 20	\$ 5.00	\$40.00
	Fotocopias	\$ 25.00	\$ 25.00
	Impresiones	\$ 5.00	\$ 5.00
	Lápices y Lapiceros	\$ 1.50	\$ 1.50
	Folders	\$ 1.00	\$ 1.00
	Anillado	\$ 15.00	\$ 15.00
4	Empastado	\$ 15.00	\$ 60.00
Gastos Consumibles			
	Cartuchos de tinta	\$ 60.00	\$ 60.00

CD's	\$ 1.50	\$ 1.50
Otros Gastos		
Servicio de Internet	\$ 38.50	\$ 38.50
Teléfono	\$ 15.00	\$ 15.00
Llamadas a Larga Distancia	\$ 2.50	\$ 2.50
Servicio de Electricidad	\$ 28.00	\$ 28.00
Viáticos	\$ 205.75	\$ 205.75
TOTAL		\$ 952.75

Tabla 4.3 – Presupuesto del Trabajo de Grado

4.6 PROYECCIÓN DEL TIEMPO DE UTILIDAD DEL EQUIPO

Se calcula que el número de estudiantes beneficiados es de aproximadamente 56, en la materia de microprogramación de la carrera de Ingeniería de Sistemas Informáticos.

Beneficios respecto al tiempo de vida útil del equipo			
No. de Entrenadores	1	Kit de entrenamiento	
No. de Software	3	Programas de Software	
		Simulador 8085	
		Hyperterminal	
		ASM85	

Beneficio Cuantificado en el Tiempo de Vida del equipo antes de ser reemplazado	
No. de Prácticas de Laboratorio por Semestre	15 practicas por semestre
No. de Grupo de Estudiantes para cada practica de laboratorio	5 grupos de 3 personas
No. Total de Guías de laboratorio a realizar por semestre	75 guías en total
Promedio de Veces que se Limpia Memoria por ejecución de programa	4 Veces por resolución de guía
No. Total de Veces que se limpia memoria por semestre	300 Veces en total
No. de semestres que durara el equipo antes de su reemplazo	10 semestres
No. de años que durara el equipo antes de su reemplazo	5 años
Es decir que la vida útil aproximada del equipo será de 5 años para cada equipo	

Tabla 4.4 – Vida Útil del Equipo

4.7 BENEFICIOS

4.7.1 Beneficios Proporcionados a la Población Estudiantil

Se otorgará beneficio directo a una población estudiantil aproximada de 56 estudiantes que cursaran la materia de Microprogramación facilitándoles acceso a material didáctico como: guías prácticas para realizar laboratorios con microprocesadores, equipo de software (simuladores, entornos de trabajo y ensambladores), equipo hardware (Primer TRAINER para el microprocesador 8085) y manuales acerca de los diferentes elementos didácticos, los cuales son de suma importancia para la realización de prácticas de laboratorio con microprocesadores, así como también para poder crear diferentes aplicaciones prácticas basadas en el microprocesador 8085.

Es una enorme oportunidad otorgada a los estudiantes ya que estos podrán obtener y reforzar sus conocimientos teóricos vistos en las clases, en base a prácticas de laboratorios que serán ampliamente relacionadas con la temática y además les ayudara en el manejo de diversas aplicaciones tanto de software como de hardware. De tal forma incrementara y mejorará los conocimientos que estos obtengan a lo largo de la carrera y poder desarrollar un buen papel en su futuro como profesionales.

Además se beneficiarán de forma indirecta a otra parte de la población estudiantil que conforma la carrera de ingeniería de Sistemas Informáticos, ya que al saber que en la materia de microprogramación cuenta con equipo especializado para la realización de prácticas de laboratorio estos se motivarán y desarrollaran expectativas acerca de la materia que estudiaran posteriormente.

Los beneficios que se obtendrán con la implementación del laboratorio didáctico son indudables, ya que al contar con este los estudiantes obtendrán y desarrollarán mayores conocimientos acerca de los microprocesadores, que estos conocimientos no estén basados solamente en forma teórica sino que puedan reforzarlos y comprobarlos en base a la práctica.

En conclusión se beneficiará a una población de estudiantes que conforman la carrera de Ingeniería de Sistemas Informáticos ayudando no solo a la formación de estos sino también aportando mejoras al perfil de egresados, ya que estos podrán desenvolverse con mayor seguridad ante diversas problemáticas con microprocesadores que ocurran en el ámbito laboral.

4.7.2 Beneficios Proporcionados al Departamento de Ingeniería y a la UES-FMO

La Universidad de El Salvador cuenta con un amplio reconocimiento a nivel tanto Nacional como Internacional, por la excelente formación académica otorgada a sus estudiantes y la forma en que esta brinda diferentes alternativas a la sociedad.

La UES-FMO incrementa su prestigio al poseer un laboratorio con equipo especializado para trabajar con microprocesadores ya que se vuelve una opción viable de estudio ante la sociedad que exige profesionales de altos niveles académicos y capaces de resolver grandes desafíos.

La Universidad educa a través de poseer un laboratorio didáctico con equipo especializado para el trabajo con microprocesadores que incremente la asimilación de los conocimientos de los estudiantes de una forma práctica haciendo uso de diversas aplicaciones tecnológicas.

Proporcionará además la capacidad de formar estudiantes de alto nivel tecnológico capaces de afrontar los diferentes desafíos que se presentan en el ambiente laboral, ya que como todos sabemos la informática cambia continuamente y se debe de estar en un constante estudio acerca de las nuevas tecnologías que surgen en el mercado. Es por esto que la universidad no solo contará con los medios para enseñar nuevas tecnologías sino que ayudará a la adopción de un esquema de trabajo que le permita al estudiante adaptarse fácilmente al surgimiento de nuevas tecnologías en el futuro.

La adquisición de equipo hardware y software para la realización de prácticas de laboratorio en la UES-FMO aportará mejoras en la infraestructura educativa de los laboratorios de la Universidad, ayudando a la formación de excelentes profesionales en el área de tecnología con bases prácticas, ya que contará con un laboratorio didáctico compuesto por: equipo especializado de hardware para la programación de microprocesadores, herramientas de simulación y material didáctico de apoyo.

Para finalizar impulsar, motivar y crear planes de capacitaciones con el fin de dar a conocer las nuevas tecnologías adquiridas a los docentes interesados.

4.8 MODELO DE PLÁN DE CAPACITACIÓN

4.8.1 Generalidades

Un proceso de reforma en las instituciones supone la transformación de las diversas áreas que constituye un significativo en su misión y visión, que da respuesta a las actuales necesidades.

Uno de los pilares en los que se respalda esta transformación es, sin duda alguna, la labor docente, ámbito en donde frecuentemente se gestan propuestas innovadoras que posibilitan la implantación de reformas.

El reconocimiento de su impacto en la vida de las instituciones supone también generar acciones específicas que refrenden el compromiso institucional con este sector y con la sociedad a la que responde.

La capacitación, formación y actualización del cuerpo docente es una de las acciones que, en forma cotidiana, deben realizarse, al igual que la investigación en este ámbito. Por ello, es fundamental para el desarrollo de las instituciones ofrecer una propuesta de formación y actualización; la cual debe estar centrado en el estudiante, lo cual exige una renovación de los contenidos, métodos, prácticas y medios de transmisión del saber, que han de basarse en nuevos tipos de vínculos y de colaboración con la comunidad estudiantil.

Así, la formación y actualización docente deberá ajustarse a este mismo marco y ofrecer, desde una perspectiva innovadora, una propuesta significativa para el profesorado que le procure nuevas herramientas y nuevas formas de realización para su práctica docente, ordenadas desde un enfoque educativo basado en competencias cuyo centro de acción sea el aprendizaje concibiendo así como un ejercicio docente que considere enseñar para aprender.

En este marco, se propone la realización de un plan de capacitación que ayude a la formación docente para un Nuevo Modelo Educativo en el área de los microprocesadores y sus aplicaciones.

Que se Ofrece con el Plan de Capacitación

- Estrategias para una práctica docente innovadora.
- Incorporar las tecnologías de información y comunicación para innovar la práctica docente.
- Vincular el proceso de formación con un proyecto de investigación-acción.

Plan de capacitación dirigido a:

Al personal docente y auxiliares de cátedra de la Universidad de El Salvador, Facultad Multidisciplinaria de Occidente, del Departamento de Ingeniería y Arquitectura, responsables de incrementar la asimilación y el aprendizaje de los conocimientos en los estudiantes.

Introducción del plan de capacitación

Con el plan de capacitación se aumentaran las técnicas y habilidades para la confección del proceso de asimilación y transmisión de conocimientos tanto para el personal docente como para los estudiantes.

Objetivo General

Innovar la práctica docente a partir de elementos teórico-metodológicos sobre aprendizaje, enseñanza, evaluación y uso de material didáctico, desarrollando el análisis de la propia práctica y el diseño de estrategias didácticas congruentes con modelos educativos centrados en el estudiante y en la construcción de aprendizajes.

Propuesta Metodológica

- La práctica docente como eje para la formación y actualización.
- Estudio independiente.
- Sesiones de trabajo en aula.
- Foros de discusión de las lecturas y materiales del curso.
- Prácticas y aplicación de instrumentos frente a grupo y elaboración de reportes.



Personal Académico que Impartirá los Cursos

El plan de capacitación cuenta con un grupo de trabajo que será el encargado de impartir la capacitación, los cuales se detallan a continuación:

- Dubón Castro, María Elisa
- Guzman Turcios, Alicia Carolina
- López Menéndez, Rina Elizabeth

Ubicación

La capacitación se llevará a cabo en el edificio de usos múltiples de la UES-FMO, específicamente en el Centro de Cómputo del Departamento de Ingeniería y Arquitectura.

Cupo de Personas

El número de personas para asistir a la capacitación será de 8.

4.8.2 Módulos que Integran el Plan de Capacitación

El tiempo de duración por curso será de 4 horas en jornada única.

A continuación se desglosan los diferentes cursos que formarán parte de la capacitación de plan de microprocesadores.

Nombre del Curso	Nombre de Guía	Objetivo	Duración
Arquitectura del PC	Introducción al Equipo Primer TRAINER	Conocer la arquitectura del equipo Primer TRAINER de manera básica.	30 minutos
	Utilización de los programas de software del Primer TRAINER	Aprender a utilizar el software relacionado al trabajo con el equipo Primer TRAINER.	60 minutos
Lenguaje Ensamblador	Realización de ejemplos prácticos	Desarrollar y aplicar los conocimientos prácticos adquiridos sobre las instrucción del microprocesador 8085	150 minutos

Tabla 4.5 – Contenido de Capacitación

4.8.3. Material y Equipo Utilizado

CANTIDAD	MATERIAL Y EQUIPO
EQUIPO	
1	PRIMER TRAINER
2	COMPUTADORAS
2	PROYECTORES

1	CÁMARA DE VIDEO
1	BASE
1	SUITE DE PROGRAMAS INFORMÁTICOS
MATERIAL	
	GUÍAS PRACTICAS EN PDF
1	BORRADOR
3	PLUMONES PARA PIZARRA ACRÍLICA
	LISTADO DE ASISTENCIA

Tabla 4.6 – Material y Equipo para Capacitación

4.8.4 Presupuesto de Capacitación

CANTIDAD	DESCRIPCIÓN	PRECIO UNITARIO	PRECIO TOTAL
MATERIAL			
	GUÍAS PRACTICAS EN PDF	-	-
1	BORRADOR	\$ 1.00	\$ 1.00
3	PLUMONES PARA PIZARRA ACRÍLICA	\$ 0.75	\$ 2.25
	LISTADO DE ASISTENCIA	\$ 0.03	\$ 0.06
8	CD	\$ 0.25	\$ 2.00
MISCELÁNEOS			
	REFRIGERIO POR DÍA	\$ 10.00	\$ 10.00
TOTAL			\$ 15.31

Tabla 4.6 – Presupuesto para Capacitación

5.0 CONCLUSIONES

Teniendo en cuenta que el objetivo primordial del trabajo de grado es el desarrollo e implementación del laboratorio didáctico para el trabajo con el microprocesador 8085 haciendo uso de los diversos medios como lo son: hardware, software, elaboración y desarrollo de guías de laboratorio práctico con el propósito de dar apoyo a la adquisición de los conocimientos teóricos-prácticos y de esta forma llegar a la transformación de los métodos de enseñanza-aprendizaje.

Contando con las mejores expectativas por parte del sector estudiantil de la FMO con respecto a la utilización de este tipo de equipo de laboratorio para mejorar de esta forma la asimilación de conocimientos teóricos. Además se incrementará a la motivación de los estudiantes hacia el estudio de microprocesadores y temas a fines.

Al capturar el interés del sector estudiantil sobre esta temática permitirá que los futuros profesionales cuenten con los conocimientos necesarios para el desarrollo y solución de aplicaciones entre las cuales se pueden mencionar: soporte técnico, desarrollo de nuevas aplicaciones de PC y móviles, trabajos de investigación, y enseñanza.

De esta forma se mejorará el perfil profesional de los egresados de la carrera de ingeniería de sistemas informáticos de la UES-FMO; estableciendo así una nueva rama de especialización a los estudiantes de la carrera.

Al seleccionar el estudio del microprocesador 8085 permitirá el mejoramiento de la enseñanza, a pesar que dicho modelo de microprocesador es de un costo accesible cubre las exigencias de la temática de la materia, tanto así que cuenta con la flexibilidad necesaria para que los estudiantes puedan entender de una manera más fácil la forma en que los microprocesadores trabajan.

También al poseer equipo de hardware para realización de prácticas los estudiantes contarán con herramientas de software que facilitará la sustentabilidad de los resultados obtenidos a lo largo del desarrollo de las guías prácticas. Cuenta con similitudes de operación, utilización, diseño de programas y ejecución de una manera fácil y didáctica.

El presente trabajo de investigación es dirigido a los estudiantes de la carrera de ingeniería de sistemas informáticos.

6.0 RECOMENDACIONES

Durante el desarrollo y la ejecución del proyecto se recopilaron una serie de recomendaciones, entre las cuales están:

- Que se motiven a los estudiantes a realizar las prácticas de laboratorio no solo como una simple formalidad, sino que como algo que en realidad sea importante para su aprendizaje.
- Que a partir de las prácticas de laboratorios se tomen ejemplos para realizar diferentes proyectos prácticos en la materia de Microprogramación, siempre bajo el control y supervisión del docente encargado de la materia.
- Que se establezca un tiempo para dar solución a las guías prácticas, para que los conocimientos queden completamente claros. (Tiempo sugerido de realización de prácticas 100 min.).
- Los grupos de prácticas deben estar formados por un máximo de 3 personas, para que estos puedan realizar las prácticas de laboratorio sin ningún contratiempo.
- Los horarios de prácticas deben de ser previamente establecidos en coordinación del docente encargado de la materia así como de los instructores o auxiliares de cátedra.
- Cada grupo de trabajo debe contar con lo siguiente para la realización de las prácticas de laboratorio: Una computadora con las herramientas de software recomendadas previamente instaladas, un Primer TRAINER y la guía de trabajo ya sea de forma impresa o digital.
- Los estudiantes deben de haber hecho un estudio previo acerca de la temática de la guía de laboratorio para la mejor comprensión de la misma.
- Es imprescindible que los estudiantes posean conocimientos acerca de lenguaje ensamblador, sistema hexadecimal y electrónica digital en general, así como también que estos tengan conocimientos sobre el manejo de diversas herramientas de software.
- Es importante que los estudiantes posean conocimientos sólidos de electrónica digital como: conversión de bases, manejo de operaciones básicas, etc. Para la mejor asimilación de los conocimientos que se pretenden impartir con el desarrollo de la guía práctica de laboratorio.

- Que las Autoridades pertinentes o los mismos estudiantes promuevan y desarrollen planes de ayuda e inversión extranjera para que la Universidad pueda obtener más y mejores equipos tecnológicos en sus laboratorios que ayuden al aprendizaje de los estudiantes y a elevar el reconocimiento de la misma Universidad.
- Promover e incentivar el desarrollo de proyectos prácticos relacionados con la materia haciendo uso del equipo especializado.
- Buscar ayuda extranjera para la especialización de los docentes mediante cursos y capacitaciones en el área de Microprocesadores de manera que ayuden a incrementar sus conocimientos y que estos puedan aplicarlos en las materias que imparten. Contribuyendo así a elevar la excelencia de los estudiantes y el prestigio de la Universidad.

7.0 BIBLIOGRAFÍA

Trabajos de Graduación

Argueta, Chinchilla, Reyes (2006). Propuesta para la implementación del laboratorio de hardware y elaboración de las guías de práctica de las asignaturas de especialidad de la carrera de Ingeniería de Sistemas Informáticos. Trabajo de graduación para optar al título de Ing. de Sistemas Informáticos, Departamento de Ingeniería y Arquitectura, Facultad Multidisciplinaria de Occidente, Santa Ana, El Salvador.

Francia Escobar, Roberto (2007) Diseño de guías de trabajo y construcción de equipo didáctico para la implementación de prácticas de laboratorio con Microcontroladores para la asignatura de Microprogramación, que se imparte en la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente

Bolaños, Coto (2008). “Diseño y propuesta de un laboratorio pedagógico – didáctico para la elaboración de circuitos impresos con software especializado en el laboratorio de hardware de la carrera de ingeniería en sistemas informáticos de la facultad multidisciplinaria de occidente.”

Libros

ENSAMBLADOR BASICO, Alberto Rojas Ponce. México: Computer Editores. S.A de C.V, 1995

LENGUAJE ENSAMBLADOR, Kip R. Irving, 5ª Edición. Prentice Hall, 2008

PRINCIPIOS DE ELECTRÓNICA. Malvino, A.P. Ed. 6ª Edición. McGraw-Hill, 1993

MICROPROCESADORES DE LOS SISTEMAS DE CONTROL. Sclipa, Andrés L. 6to año 1ra división Electromecánica, 1998

FUNDAMENTOS DE MICROPROCESADORES. Roger, Tokheim. Editorial McGraw-Hill, 1990

WEB

Anabel Cavazos
www.scribd.com

Anurag Ved
www.simtel.net

Alejandro Cornejo
<http://www.Hardware12v.com>
acornejo@hardware12v.com

Bethsy Díaz
www.monografias.com

Carlos Mujica
microprocesadores-darcy.blogspot.com

Darío Alpern
www.alpertron.com.ar

Elizabeth, Rodríguez,
www.maestrosdelweb.com

Elva Loya
eloyat.blogspot.com

Guadalupe Elizalde Vieyra
<http://www.fismat.umich.mx/~elizalde/curso/node118.html>

Guillermo Roper
www.duiops.net

Henry, Gutiérrez
www.reparesupc.com

Iván Peñarrocha
<http://personal.redestb.es/juanhr/topsec.htm>
ivan.p@hardware12v.com

Jonathan Mizrachi
<http://www.rincondelvago.com/>

José Calderón
Configuraequipos.com

José María Carreras González
<http://www.monografias.com>

Josefina Flores Q.
<http://www.monografias.com/>
aromalabanda@hotmail.com

Juan Herrerías Rey
<http://es.wikipedia.org/wiki/Wikipedia:portada>

Juan Ignacio Baena
<http://www.mii3314jibp.wordpress.com>

Juan López Sánchez
http://www.gratisweb.com/lula_f/sistemas_de_numeracion.htm

Juan Ramón, Jiménez Alaniz
Universidad Autónoma Metropolitana

Marco, Morales
www.Cespro.com
info@cespro.com

Manuel Girón
[www. es. Wikipedia.org](http://www.es.Wikipedia.org)

Mendoza, Pech y Contreras
<http://www.monografias.com>

Noé Soriano
www.idg.es

Prof. Lauro Soto
<http://www.mitecnologico.com>

Sridhar Ratnakumar
gnusim8085.sourceforge.net

Vaneet Singla
www.softlookup.com

Wikilibros
<http://es.wikibooks.org/wiki>

8.0 GLOSARIO

ALGORITMO: Palabra que viene del nombre del matemático árabe Al-Khwarizmi (780 - 850 aprox.). Define el conjunto de instrucciones que sirven para ejecutar una tarea o resolver un problema. Los motores de búsqueda usan algoritmos para mostrar los resultados de búsquedas.

AMD: El segundo más grande fabricante de microprocesadores, después de Intel. También fabrica memorias flash y circuitos integrados para aparatos para redes, entre otros. Fue fundada en el año 1969 y ha contribuido a que los precios de las PC sean menores por su alta calidad y buenos precios.

ASCII: uno de los primeros y más usados códigos de caracteres. Existe en versiones de 7 u 8 bits.

BCD: Binario Codificado en Decimal.

BinHex: Binario Hexadecimal. Método utilizado para convertir archivos no ASCII o binarios al formato de 7 bits ASCII. Este método es utilizado principalmente por computadoras Macintosh y es necesario porque el correo en Internet solo puede utilizarse archivos ASCII.

BIT: unidad mínima de información de la memoria, equivalente a un "sí" (0) o un "no" (1) binarios. La unión de 8 bits da lugar a un byte.

BUS: canal por el que circula información electrónica en forma de bits. El ancho de bus es el número de bits transmitidos simultáneamente por el bus.

BYTE: Unidad de información compuesta de 8 bits ("1" o "0"). Esta unidad de información es suficiente para representar un número o una letra entre otros.

CA (Corriente Alterna): Corriente eléctrica que cambia su amplitud en forma periódica con el tiempo.

CAPACITOR (CONDENSADOR) DE PASO: Es un capacitor que tiene por finalidad mantener la alta ganancia en c.a. y la ganancia en c.c. es reducida con ayuda de una resistencia de realimentación (Re).

CMOS: Complementary Metal Oxide Semiconductor, un tipo de memoria que se caracteriza por consumir muy poca energía eléctrica, lo que la hace idónea para almacenar datos de la BIOS.

COM: acrónimo con el que se designa a cada uno de los puertos series o de Comunicaciones

CÓDIGO FUENTE: Conjunto de instrucciones que componen un programa, escrito en cualquier lenguaje. En inglés se dice "source code". Hay programas de código abierto y "de código cerrado" como por ejemplo Windows, Photoshop, y la mayoría de los programas comerciales, en donde el código es inaccesible y por lo tanto no se puede alterar la estructura del programa.

CONTROLADOR: forma española de denominar los drivers.

CPU: Central Processing Unit o Unidad Central de Proceso. El "cerebro" de un ordenador; en general, sinónimo de microprocesador. En ocasiones se usa para referirse al toda la caja que contiene la placa base, el micro y las tarjetas de expansión.

DAC (D / A): Convertidos digital – analógico.

DECIMAL: Sistema de numeración en base 10. Ver: Sistema decimal

DIGITAL: Un sistema en que los caracteres o códigos son utilizados para representar números o cantidades físicas en forma discretos.

DISIPADOR: aparato que ayuda a eliminar el calor generado por un cuerpo, en general el microprocesador del equipo, en ocasiones con la colaboración de un ventilador. Para ello, busca tener buena conducción del calor (suelen ser de cobre) y gran superficie.

DRIVER: pequeño programa cuya función es controlar el funcionamiento de un dispositivo del ordenador bajo un determinado sistema operativo.

ELECTRICIDAD: Propiedad de los electrones y protones, expresada numéricamente como carga en coulombios.

ELECTRÓNICA: campo de la tecnología que trata sobre aparatos electrónicos

EPROM (Erasable programmable Read Only Memory): Dispositivo de memoria en que los datos pueden grabarse con un equipo especial. Se borra usualmente con rayos ultravioleta.

ESTÁTICA (ELECTRICIDAD): Carga eléctrica que no fluye (como la corriente). Ver: Electricidad estática

FILTRO: Circuito selectivo que sólo deja pasar frecuencias determinadas, bloqueando todas las otras.

HARDWARE: Maquinaria. Componentes físicos de una computadora o de una red (a diferencia de los programas o elementos lógicos que los hacen funcionar).

HEXADECIMAL: Sistema de numeración de base 16.

IC: Circuitos Integrados

I/O: INPUT/OUTPUT, ENTRADA/SALIDA: Generalmente hace referencia a dispositivos o puertos de comunicación (serie, paralelo, etc.) o a la tarjeta que los controla (si no están integrados en la placa base).

INTEL: El fabricante líder de microprocesadores para PC. Los procesadores Intel fueron usados en las primeras computadoras que incorporaban el sistema operativo DOS de Microsoft. Su línea de procesadores Pentium incremento los niveles de desempeño de las computadoras a niveles superiores. Intel también fabrica tarjetas madre (motherboards), procesadores de red y un sin fin de circuitos procesadores que están pavimentando el futuro de la computación personal.

JUMPER: tipo de interruptor de muy pequeño tamaño que se usa en numerosas piezas hardware, especialmente la placa base. Consiste en dos pines metálicas que deben unirse mediante una pieza metálica, generalmente recubierta a su vez de plástico.

K5: microprocesador de AMD similar al Pentium clásico.

K6: microprocesador de AMD que incluye MMX, de rendimiento superior al Pentium MMX aunque inferior al Pentium II.

K6-2: también llamado "K6-3D"; microprocesador de AMD que incluye MMX y la tecnología "3DNow!" para el manejo de aplicaciones 3D, de rendimiento igual o superior al Pentium II.

KB: kilobyte, múltiplo del byte equivalente a 1024 bytes. Más correcta, aunque menos utilizada, es la forma "kb"; también se emplea "Kb".

KBPS: Kilobits por segundo. Unidad de medida que comúnmente se usa para medir la velocidad de transmisión por una línea de telecomunicación, como la velocidad de un cable MODEM por ejemplo.

KEY: Serie de signos previamente convenidos que sirven como clave o fórmula para transmitir mensaje secretos o privados.

LED: Light Emitting Diode, diodo emisor de luz. Un dispositivo luminoso de pequeño tamaño utilizado en electrónica.

MHz: Unidad de frecuencia que equivale a un millón de ciclos por segundo.

MICROPROCESADOR: Microchip. Circuito integrado en un soporte de silicón el cual está formado por transistores y otros elementos electrónicos miniaturizados. Es uno de los elementos esenciales de una computadora.

MICROSOFT: Fundada en 1975 por Bill Gates, entre otros. Responsable de los sistemas operativos Windows XP y Vista, del grupo de programas Office (Word, Excel, PowerPoint), de los controles Active X, del navegador Internet Explorer entre muchos otros programas.

MIPS: acrónimo de "millones de instrucciones por segundo"

MS-DOS: Sistema operativo DOS, de Microsoft. Su entorno es de texto, tipo consola, y no gráfico. Sigue siendo parte importante de los sistemas operativos gráficos de Windows.

NEGACIÓN EN PINES ($\overline{\text{INTA}}$): Significa que las E/S reciben un voltaje inverso al que se le envía.

OVERCLOCKING: técnica por la cual se fuerza un microprocesador a trabajar por encima de su velocidad nominal.

OVERDRIVE: familia de microprocesadores de Intel dedicada a la actualización de equipos. Existen con núcleos de 486 y de Pentium con o sin MMX.

PENTIUM: microprocesador de Intel de 32 bits con arquitectura superescalar, capaz de hacer el procesamiento paralelo de dos instrucciones por ciclo de reloj y con una unidad matemática muy mejorada respecto de la del 486.

PIN: cada uno de los conectores eléctricos de muchos elementos hardware, como las "patitas" de muchos microprocesadores.

PUERTO SERIAL: conexión por medio de la cual se envían datos a través de un solo conducto. Por ejemplo, el Mouse o ratón se conecta a un puerto serial. Las computadoras tienen dos puertos seriales: COM1 y COM2.

RAM: Random Access Memory (memoria de acceso aleatorio), es la memoria temporal, que se borra apenas se apaga la computadora.

ROM: Read Only Memory (memoria de solo lectura); de esta memoria, solo se puede leer pero no modificar.

SUBÍNDICE EN NÚMEROS (EJ.: 1110111_2): Determina la base numérica en la cual se encuentra dicho número.

TRANSFERENCIA DE ARCHIVOS: Copia de un archivo desde un ordenador a otro a través de una red de computadoras.

UNIDAD ARITMÉTICO / LÓGICA: Esta unidad realiza cálculos (suma, resta, multiplicación y división) y operaciones lógicas (comparaciones). Transfiere los datos entre las posiciones de almacenamiento. Tiene un registro muy importante conocido como: Acumulador ACC Al realizar operaciones aritméticas y lógicas, la UAL mueve datos entre ella y el almacenamiento. Los datos usados en el procesamiento se transfieren de su posición en el almacenamiento a la UAL. Los datos se manipulan de acuerdo con las instrucciones del programa y regresan al almacenamiento. Debido a que el procesamiento no puede efectuarse en el área de almacenamiento, los datos deben transferirse a la UAL. Para terminar una operación puede suceder que los datos pasen de la UAL al área de almacenamiento varias veces.

UNIDAD DE CONTROL: Es en esencia la que gobierna todas las actividades de la computadora, así como el CPU es el cerebro de la computadora, se puede decir que la UC es el núcleo del CPU. Supervisa la ejecución de los programas Coordina y controla al sistema de cómputo, es decir, coordina actividades de E/S. Determina que instrucción se debe ejecutar y pone a disposición los datos pedidos por la instrucción. Determina donde se almacenan los datos y los transfiere desde las posiciones donde están almacenados. Una vez ejecutada la instrucción la Unidad de Control debe determinar donde pondrá el resultado para salida ó para su uso posterior.

VCC, VSS: Conexiones a 5 Voltios, generalmente utilizados en pines de dispositivos electrónicos.

XEON: Es un microprocesador Pentium de 400 MHz hecho por Intel para servidores y estaciones. En una tarjeta madre de Intel para servidores, hasta 8 o más procesadores Xeon pueden hacer multiprocesos usando el mismo bus de 100 Mhz. Xeon está reemplazando al Pentium Pro, y está diseñado para Internet, grandes bases de datos, así como para ingenieros, graficas y aplicaciones multimedia que requieren de mucho movimiento de data de forma rápida. En la línea de Pentium, Xeon es actualmente el más poderoso (siendo Celeron el menos poderoso).

ANEXOS

ANEXOS

Anexo 1

Tablas e información referente a Marco Teórico.

Anexo 2

Formato e interpretación de encuesta dirigida hacia los alumnos de ingeniería de sistemas informáticos.

Anexo 3

Formato e interpretación de encuesta dirigida hacia los docentes, auxiliares de cátedra e instructores encargados de impartir las materias relacionadas en la área de microprogramación y electrónica

Anexo 4

Formato e interpretación de encuesta dirigida hacia alumnos graduados de ingeniería de sistemas informáticos de la UES-FMO y que laboran en el área de microprocesadores y afines.

Anexo 5

Cuestionario utilizado en las visitas técnicas realizadas a las instituciones educativas.

Anexo 6

Visitas técnicas realizadas a las instituciones educativas.

Anexo 7

Imágenes de visitas técnicas realizadas a las instituciones educativas.

Anexo 8

Guías prácticas de laboratorio para la programación de microprocesadores.

ANEXO 1: TABLAS E INFORMACIÓN REFERENTE A MARCO TEÓRICO

Instrucción	Código	Explicación
MOV r1,r2	01DDDRRR	Carga el registro r1 (DDD) con el contenido del r2 (RRR).
MOV r, M	01DDD110	Carga el registro r (DDD) con el contenido de la dirección apuntada en el par HL.
MOV M, r	01110RRR	Carga la dirección apuntada por HL con el contenido del registro r (RRR).
MVI r	00DDD110	Carga el registro r de modo inmediato con el 2° Byte de la instrucción
MVI M	00110110	Carga la dirección apuntada por HL con el 2° Byte de la instrucción
LXI rp	00DD0001	Carga el par de registros rp (serán BC, DE, HL) con el 2° y 3° Byte de la instrucción.
LDA	00111010	Carga el acumulador con el contenido de la dirección apuntada en el 2° y 3° Byte de la instrucción.
STA	00110010	Carga la posición apuntada por el 2° y 3° Byte de la instrucción con el contenido del acumulador.
LHLD	00101010	Carga los registros HL con el contenido de la posición apuntada por 2° y 3° Byte (carga L) y por la apuntada por 4° y 5° Byte (carga H) de la instrucción.
SHLD	00100010	Almacena el contenido de HL en la posición que indica el 2° y 3° Byte y la siguiente respectivamente.
LDAX rp	00RR1010	Carga el acumulador con el contenido de la dirección que apunta el par de registros rp (RR) solo para los registros pares BC y DE
XCHG	11101011	Intercambia el contenido del par HL por el contenido del par DE.

Tabla 1 - Instrucciones de transferencia de datos:

Todas las anteriores:

Instrucción (Hex)	código (Hex)	Instrucción	código
MOV A, A	7F	MOV B, A	47
MOV A, B	78	MOV B, B	40

MOV A, C	79	MOV B, C	41
MOV A, D	7A	MOV B, D	42
MOV A, E	7B	MOV B, E	43
MOV A, H	7C	MOV B, H	44
MOV A, L	7D	MOV B, L	45
MOV A, M	7E	MOV B, M	46
MOV C, A	4F	MOV D, A	57
MOV C, B	48	MOV D, B	50
MOV C, C	49	MOV D, C	51
MOV C, D	4A	MOV D, D	52
MOV C, E	4B	MOV D, E	53
MOV C, H	4C	MOV D, H	54
MOV C, L	4D	MOV D, L	55
MOV C, M	4E	MOV D, M	56
MOV E, A	5F	MOV H, A	67
MOV E, B	58	MOV H, B	60
MOV E, C	59	MOV H, C	61
MOV E, D	5A	MOV H, D	62
MOV E, E	5B	MOV H, E	63
MOV E, H	5C	MOV H, H	64
MOV E, L	5D	MOV H, L	65
MOV E, M	5E	MOV H, M	66
MOV L, A	6F	MOV M, A	77
MOV L, B	68	MOV M, B	70
MOV L, C	69	MOV M, C	71
MOV L, D	6A	MOV M, D	72
MOV L, E	6B	MOV M, E	73
MOV L, H	6C	MOV M, H	74
MOV L, L	6D	MOV M, L	75
MOV L, M	6E	MOV M, M	76
MVI A, BYTE	3E	LXI B, DOBLE BYTE	01
MOV B, BYTE	06	LXI D, DOBLE BYTE	11
MOV C, BYTE	0E	LXI H, DOBLE BYTE	21
MOV D, BYTE	16	LXI SP, DOBLE BYTE	31(para definir el STACK P.)
MOV E, BYTE	1E	LDAX B	0A
MOV H, BYTE	26	LDAX D	1A
MOV L, BYTE	2E	LHLD DIRECCION	2A
MOV M, BYTE	36		

		LDA DIRECCION	3A
STAX B	02		
STAX D	12	XCHG	EB
SHLD DIRECCION	22		
STA DIRECCION	32		

BYTE= VALOR ENTRE 0 Y FF (0 Y 255 EN DECIMAL)

DOBLE BYTE= DOS BYTES PARA FORMAR 16 BITS

DIRECCION = 16 BITS o 2 BYTES

Instrucción	Código	Explicación
ADD r	1000RRR	Suma al acumulador el registro r y el resultado queda en el acumulador.
ADD M	1000110	Suma al acumulador el contenido de la posición apuntada por HL y resultado en acumulador.
ADI	11000110	Suma al acumulador el 2º byte de la instrucción y resultado en acumulador.
ADC r	10001RRR	Suma al acumulador el registro r y el acarreo, resultado en acumulador.
ADC M	10001110	Suma a acumulador el contenido de la posición apuntada por HL y el acarreo, resulta en acumulador.
ACI	11001110	Suma al acumulador el 2º byte de la instrucción y el acarreo y resultado en acumulador.
SUB r	10010RRR	Resta al acumulador el contenido del registro r y lo deja en el acumulador.
SUB M	10010110	Resta al acumulador el contenido de la posición que apunta HL y resultado al acumulador.
SUI	11010110	Resta al acumulador el 2º byte de la instrucción
SBB r	10011RRR	Resta al acumulador el registro r + el acarreo.
SBB M	10011110	Resta al acumulador el contenido de la posición que apunta HL.
SBI	11011110	Resta al acumulador el 2º byte + el acarreo.
INR r	00RRR100	Incrementa en 1 el registro r.(Z;S;P;AC)
INR M	00110100	Incrementa en 1 el contenido de la posición que apunta HL (Z;S;P;AC)
DCR r	00RRR101	Decrementa en 1 el registro r (Z;S;P;AC)
DCR M	00110101	Decrementa en 1 el contenido de la

		posición que apunta HL (Z;S;P;AC)
INX rp	00RR0011	Incrementa en 1 el par rp de registros BC,DE,HL
DCX rp	00RR1011	Decrementa en 1 el par rp de registros BC,DE,HL
DAD rp	00RR1001	Suma a HL el par de registros rp (CY a veces)
DAA	00100111	Los 8 bits del acumulador se ajusta a BCD=decimal (flags)

Tabla 2 - Instrucciones Aritmético Lógicas

ENTRE PARÉNTESIS () FLAGS AFECTADOS POR LA INSTRUCCIÓN.

Todas las anteriores:

Instrucción	código (Hex)	Instrucción	código (Hex)
ADD A	87	ADC A	8F
ADD B	80	ADC B	88
ADD C	81	ADC C	89
ADD D	82	ADC D	8A
ADD E	83	ADC E	8B
ADD H	84	ADC H	8C
ADD L	85	ADC L	8D
ADD M	86	ADC M	8E
SUB A	97	SBB A	9F
SUB B	90	SBB B	98
SUB C	91	SBB C	99
SUB D	92	SBB D	9A
SUB E	93	SBB E	9B
SUB H	94	SBB H	9C
SUB L	95	SBB L	9D
SUB M	96	SBB M	9E
INR A	3C	DCR A	3D
INR B	04	DCR B	05
INR C	0C	DCR C	0D
INR D	14	DCR C	15
INR E	1C	DCR D	1D
INR H	24	DCR H	25
INR L	2C	DCR L	2D
INR M	34	DCR M	35
INX B	03	DCX B	0B

INX D	13	DCX D	1B	
INX H	23	DCX H	2B	
INX SP POINTER)	33	DCX SP	3B	(SP=STACK
ADI BYTE	C6	DAD B	D9	
ACI BYTE	CE	DAD D	19	
SUI BYTE	D6	DAD H	29	
SBI BYTE	DE	DAD SP	39	
DAA	27			

Instrucción	Código	Explicación
JMP	11000011	Salto a la posición indicada por el 2º y 3º byte de la instrucción, 2º byte=parte baja, 3º=alta

Instrucción	Flags	Código	Explicación
JZ	Z	11FF1010	Salto a la posición indicada en el 2º y 3º byte de la instrucción si el flag (FF) tiene valor 1
JC	CY		
JPE	P		
JM	S		
JNZ	Z	11FF0010	ídem si el valor del Flag es cero
JNC	CY		
JPO	P		
JP	S		
PCHL		11101001	Salto a la posición direccionada por HL.

Tabla 3 - Instrucciones de bifurcación

Todas ellas:

Instrucción	código	Instrucción	código
JMP dir.	C3	JPO dir.	E2
JNZ dir.	C2	JPE dir.	EA
jz dir.	CA	JP dir.	F2

JNC dir.	D2	JM dir.	FA
JC dir.	DA	PCHL	E9

Instrucción	Flags	Código	Explicación
CALL		11FF1010	Salto a la posición direccionada por el 2º y 3º byte de la instrucción. Guarda en el STACK el contenido de PC (Contador de programa)
CZ	Z	11FF1100	Salto a la posición direccionada por el 2º y 3º byte de la instrucción si el flag indicado es uno. Guarda el PC en el stack.
CC	CY		
CPE	P		
CM	S		
CNZ	Z	11FF1100	Ídem anterior pero si el Flag es cero. Guarda el P.C. en el STACK.
CNC	CY		
CPO	P		
CP	S		
RET		11001001	Retorno a la última dirección almacenada en el STACK.
RZ	Z	11FF1000	Retorno a la última dirección almacenada en el STACK solo si el Flag indicado vale 1.
RC	CY		
RPE	P		
RM	S		
RNZ	Z	11FF0000	Ídem caso anterior pero si el Flag vale 0.
RNC	CY		
RPO	P		
RP	S		

Tabla4 - Instrucciones de salto y retorno a subrutinas

Todas ellas

Instrucción	código	Instrucción	código
CALL dir.	CD	RET	C9
CNZ dir.	C4	RNZ	C0
CZ dir.	CC	RZ	C8
CNC dir.	D4	RNC	D0
CC dir.	DC	RC	D8
CPO dir.	E4	RPO	E0
CPE dir.	EC	RPE	E8
CP dir.	F4	RP	F0
CM dir.	FC	RM	F8

Instrucción	Código	Explicación
PUSH rp	11RR0101	Guarda el par de registros rp en el Stack.
PUSH PSW	11110101	Guarda el contenido del acumulador y el registro de Flags en el Stack.
POP rp	11RR0001	Carga el par de registros rp el contenido de las dos últimas posiciones del Stack
POP PSW	11110001	Carga el registro de Flags y el acumulador con las dos últimas posiciones del Stack.
XTHL	11100011	Intercambio del contenido de las posiciones direccionadas por el Stack por el contenido de los registros H y L.
SPHL	11111001	Traslada el contenido del par HL al puntero del Stack. Debe ejecutarse al iniciar el sistema para ubicar el Stack en la zona elegida de la memoria.
IN port	11011011	Carga el acumulador con el contenido del canal I/O direccionado por el 2º byte de la instrucción.
OUT port	11010011	Carga el canal I/O direccionado por el 2º byte de la instrucción con el acumulador.
EI	11111011	Habilita las interrupciones (no afecta a la interrupción TRAP)
DI	11110011	Deshabilita las interrupciones (no afecta a TRAP)
HLT	01110110	Detención y entrada de la CPU hasta que se realice un Reset, una interrupción valida o un HOLD.
NOP	00000000	No opera, solo sirve para perder tiempo.
SIM	00110000	Fija el registro de interrupciones y la línea SOD con el valor del acumulador. Bit 7 del acumulador= sale por línea

		<p>SOD.</p> <p>Bit 6 del acumulador = a 1 habilita la línea SOD.</p> <p>Bit 5 no se usa.</p> <p>Bit 4 del acumulador = a 1 borra la bascula RST 7,5</p> <p>Bit 3 del acumulador = a 1 permite modificar este registro.</p> <p>Bit 2 del acumulador = a 1 prohíbe la interrupción 7,5</p> <p>Bit 1 del acumulador = a 1 prohíbe la interrupción 6,5</p> <p>Bit 0 del acumulador = a 1 prohíbe la interrupción 5,5</p>									
RIM	00100000	<p>Lee el estado de la línea SID y el estado de las interrupciones cargándolo en el acumulador:</p> <p>Bit 7 del acumulador con: Lo que tenga la línea SID,</p> <p>Bit 6 del acumulador con: a 1 indica que la interrupción 7,5 esta solicitada por ejecutar.</p> <p>Bit 5 del acumulador con: ídem anterior pero con la 6,5</p> <p>Bit 4 del acumulador con: ídem anterior pero con la 5,5</p> <p>Bit 3 del acumulador con: a 1 indica que las interrupciones INTR han sido habilitadas.</p> <p>Bit 2 del acumulador con: Estado de la interrupción 7,5</p> <p>Bit 1 del acumulador con: Estado de la interrupción 6,5</p> <p>Bit 0 del acumulador con: Estado de la interrupción 5,5</p>									
RST	11AAA111	<p>Guarda el P.C. en el Stack y lo carga con la dir: 00000000AAA000. Cuando se produce la interrupción INTR, después de aceptarla con INTA hay que introducir por el bus de datos el código AAA que será así:</p> <table border="0"> <thead> <tr> <th>AAA</th> <th>RST</th> <th>Dir. de salto (en HEX.)</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>0</td> <td>0000</td> </tr> <tr> <td>001</td> <td>1</td> <td>0008</td> </tr> </tbody> </table>	AAA	RST	Dir. de salto (en HEX.)	000	0	0000	001	1	0008
AAA	RST	Dir. de salto (en HEX.)									
000	0	0000									
001	1	0008									

		010	2	0010
		011	3	0018
		100	4	0020
		101	5	0028
		110	6	0030
		111	7	0038

Tabla 5 – Instrucciones Restantes

Todas las que quedaban:

Instrucción	código	Instrucción	código
PUSH B	C5	POP B	C1
PUSH D	D5	POP D	D1
PUSH H	E5	POP H	E1
PUSH PSW	F5	POP PSW	F1
XTHL	E3	SPHL	F9
OUT byte	D3	IN byte	DB
DI	F3	EI	FB
NOP	D0	HLT	78
RIM	20	SIM	30
RST 0	C7	RST 4	E7
RST 1	CF	RST 5	EF
RST 2	D7	RST 6	F7
RST 3	DF	RST 7	FF

Los flag (bandera en español) es un registro interno de la CPU que indica el estado de una operación después de haberla realizado (más o menos), estos son controlados por la ALU (una cosa interna que hace todas las operaciones aritméticas y lógicas). A continuación la explicación de cada uno:

Flags	Explicación
S	Cuando un resultado ha sido negativo se pone a 1
Z	Cuando una operación ha sido cero este se pone a 1
P	Indica la paridad de una operación, si es par P=1

CY Se pone a 1 cuando se produce un acarreo en una operación de 8 bits

Siempre el primer byte que se carga (por ejemplo en la instrucción JMP dir.) es el más bajo y el segundo el más alto, por lo tanto al lado de las instrucciones que llevan 2º y 3º byte estos indicaran la dirección pero al revés.

Principales Características De Los Procesadores

Procesadores Intel

Modelo	Velocidad (MHz)	Conexión	Micras	Caché L1/L2 (Kbyte)	MHz/ Multiplicador	Voltaje (voltios)	Millones de transistores
Pentium	75	Socket 7	0,6	16/0	50/1,5	3,3	3,1
	100	Socket 7	0,35	16/0	66/1,5	3,3	3,1
	120	Socket 7	0,35	16/0	60/2	3,3	3,1
	133	Socket 7	0,35	16/0	66/2	3,3	3,1
	150	Socket 7	0,35	16/0	60/2,5	3,3	3,1
	166	Socket 7	0,35	16/0	66/2,5	3,3	3,1
	200	Socket 7	0,35	16/0	66/3	3,45	3,1
Pentium MMX	166	Socket 7	0,35	32/0	66/2,5	2,8	4,5
	200	Socket 7	0,35	32/0	66/3	2,8	4,5
	233	Socket 7	0,35	32/0	66/3,5	2,8	4,5
Pentium Pro	150	Socket 8	0,6	16/256-512	60/2,5	3,1	5,5
	180	Socket 8	0,35	16/256-512	60/3	3,3	5,5
	200	Socket 8	0,35	16/256-	66/3,5	3,3	5,5

				512			
Pentium II	233	Slot 1	0,35	32/512	66/3,5	2,8	7,5
	266	Slot 1	0,35	32/512	66/4	2,8	7,5
	300	Slot 1	0,3	32/512	66/4,5	2,8	7,5
	333	Slot 1	0,25	32/512	66/5	2,8	7,5
	350	Slot 1	0,25	32/512	100/3,5	2,2	7,5
	400	Slot 1	0,25	32/512	100/4	2	7,5
	450	Slot 1	0,25	32/512	10/4,5	2	7,5
	500	Slot 1	0,25	32/512	10/5	2	7,5
Pentium II Xeon	400	Slot 2	0,25	32/512	100/4	2	7,5
	450	Slot 2	0,25	32/512	100/4,5	2	7,5
Pentium III	450	Slot 1	0,25	32/512	100/4,5	2	9,5
	500	Slot 1	0,25	32/512	100/5	2	9,5
	533	Slot 1	0,18	32/512	133/4	1,6	9,5
	550	Slot 1	0,18	32/512	100/5,5	1,6	9,5
	550E	Socket 370	0,18	32/256	100/5,5	1,6	28
	600	Slot 1	0,25	32/512	10/6	2	9,5
	600B	Slot 1	0,18	32/512	133/4,5	1,6	9,5
	600E	Slot 1	0,18	32/256	100/6	1,6	28
	600EB	Slot 1	0,18	32/256	133/4,5	1,6	28
	650	Slot 1	0,18	32/256	100/6,5	1,6	28
	667	Slot 1	0,18	32/256	133/5	1,6	28

	700	Slot 1	0,18	32/256	100/7	1,6	28
	733	Slot 1	0,18	32/256	133/5,5	1,6	28
	750	Slot 1	0,18	32/256	100/7,5	1,6	28
	800	Slot 1	0,18	32/256	133/6	1,6	28
	850	Slot 1	0,18	32/256	100/8,5	1,6	28
	866	Slot 1	0,18	32/256	133/5	1,6	28
	1000	Slot 1	0,18	32/256	133/7,5	1,6	28
Pentium III Xeon	500	Slot 2	0,25	32/512	100/5	1,6	28
	500	Slot 2	0,25	32/1024	100/5	1,6	28
	500	Slot 2	0,25	32/2048	100/5	1,6	28
	550	Slot 2	0,25	32/512	100/5,5	1,6	28
	550	Slot 2	0,25	32/1024	100/5,5	1,6	28
	550	Slot 2	0,25	32/2048	100/5,5	1,6	28
	600	Slot 2	0,18	32/256	100/6	1,6	28
	667	Slot 2	0,18	32/256	133/5	1,6	28
	733	Slot 2	0,18	32/256	133/5,5	1,6	28
	800	Slot 2	0,18	32/256	133/6	1,6	28
Celeron	266	Slot 1	0,25	32/0	66/4	2	19
	300	Slot 1	0,25	32/0	66/4,5	2	19
	300	Slot 1	0,25	32/128	66/4,5	2	19
	333	Slot 1	0,25	32/128	66/5	2	19
	300	Socket 370	0,25	32/128	66/4,5	2	19
	333	Socket	0,25	32/128	66/5	2	19

		370					
	366	Socket 370	0,25	32/128	66/5,5	2	19
	400	Socket 370	0,25	32/128	66/6	2	19
	433	Socket 370	0,25	32/128	66/6,5	2	19
	466	Socket 370	0,25	32/128	66/7	2	19
	500	Socket 370	0,25	32/128	66/7,5	2	19
	533	Socket 370	0,25	32/128	66/8	2	19
	566	Socket 370	0,18	32/128	66/8,5	2	19
	600	Socket 370	0,18	32/128	66/9	2	19

Tabla 6 – Información de Procesadores Intel

Procesadores Cyrix

Modelo	Velocidad (MHz)	Conexión	Micras	Caché L1/L2 (Kbytes)	MHz/Multiplicador	Voltaje (voltios)	Millones de transistores
6x86 PR120	100	Socket 7	0,5	16	66/1,5	3,3	3
6x86 PR133	110	Socket 7	0,5	16	73/1,5	3,3	3
6x86 PR150	120	Socket 7	0,5	16	60/2	3,3	3
6x86 PR166	133	Socket 7	0,5	16	66/2	3,3	3

6X86 PR200	150	Socket 7	0,44	16	60/2,5	2,8	3
6x86MX PR166	150	Socket 7	0,35	64	60/2,5	2,9	6
6x86MX PR166	166	Socket 7	0,35	64	66/2	2,9	6
6x86MX PR166	188	Socket 7	0,35	64	73/2,5	2,9	6
6x86MX PR166	208	Socket 7	0,35	64	83/2,5	2,9	6
6x86MX PR166	233	Socket 7	0,3	64	66/3,5	2,9	6
6x86MX PR166	250	Socket 7	0,3	64	83/3	2,9	6

Tabla 7 – Información de Procesadores Cyrix

Procesadores AMD

Modelo	Velocidad (MHz)	Conexión	Micras	Caché L1/L2 (Kbytes)	MHz/ Multiplicador	Voltaje (voltios)	Millones de transistores
K5 PR120	90	Socket 7	0,35	24/0	60/1,5	3,5	4,3
K5 PR133	105	Socket 7	0,35	24/0	66/1,5	3,5	4,3
K5 PR166	120	Socket 7	0,35	24/0	60/2	3,5	4,3
K6	166	Socket 7	0,35	64/0	66/2,5	2,9	8,8
	200	Socket 7	0,35	64/0	66/3	2,9	8,8
	233	Socket 7	0,35	64/0	66/3,5	3,2	8,8

	266	Socket 7	0,35	64/0	66/4	3,2	8,8
	266	Socket 7	0,25	64/0	66/4	2,9	8,8
	300	Socket 7	0,25	64/0	66/4,5	2,9	8,8
K6-2	266	Socket 7	0,25	64/0	66/4	2,2	9,3
	300	Socket 7	0,25	64/0	66/4,5	2,2	9,3
	333	Socket 7	0,25	64/0	66/5	2,2	9,3
	350	Socket 7	0,25	64/0	100/3,5	2,2	9,3
	400	Socket 7	0,25	64/0	100/4	2,2	9,3
	450	Socket 7	0,25	64/0	100/4,5	2,2	9,3
	475	Socket 7	0,25	64/0	133/3,5	2,2	9,3
	500	Socket 7	0,25	64/0	100/5	2,2	9,3
	533	Socket 7	0,25	64/0	133/4	2,2	9,3
	550	Socket 7	0,25	64/0	100/5,5	2,2	9,3
K6-III	400	Socket 7	0,25	64/256	100/4	2,2	21,3
	500	Socket 7	0,25	64/256	100/4,5	2,2	21,3
Athlon	500	Slot A	0,25	128/512	100/5	1,6	22
	550	Slot A	0,25	128/512	100/5,5	1,6	22
	600	Slot A	0,25	128/512	100/6	1,6	22
	650	Slot A	0,25	128/512	100/6,5	1,6	22
	700	Slot A	0,25	128/512	100/7	1,6	22
	750	Slot A	0,18	128/512	100/7,5	1,7	22
	800	Slot A	0,18	128/512	100/8	1,7	22
	850	Slot A	0,18	128/512	100/8,5	1,7	22
	900	Slot A	0,18	128/512	100/9	1,8	22
	950	Slot A	0,18	128/512	100/9,5	1,8	22
	1000	Slot A	0,18	128/512	100/10	1,8	22

Tabla 8 – Información de Procesadores AMD

ANEXO 2: FORMATO E INTERPRETACIÓN DE ENCUESTA DIRIGIDA HACIA LOS ALUMNOS DE INGENIERÍA DE SISTEMAS INFORMÁTICOS.

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



Título: Encuesta orientada a los estudiantes de la carrera de ingeniería de Sistemas Informáticos de la asignatura de microprogramación y materias afines al área de electrónica en la Facultad Multidisciplinaria de occidente.

De antemano se les agradece a los estudiantes que han colaborado y han permitido la obtención de datos para el estudio y análisis de la situación actual en el laboratorio de hardware.

Objetivo: Conocer la situación actual en la que se encuentra la materia de microprogramación y las demás materias relacionadas con el área de la electrónica.

Indicaciones: Conteste las siguientes preguntas de la manera más clara y precisa. Seleccionando la opción más conveniente.

1. ¿Posee algún conocimiento teórico-práctico acerca de electrónica?

Si No

2. ¿En qué lugar ha obtenido usted los conocimientos acerca de electrónica?

Bachillerato Cursos Adicionales Otros Medios

Universidad

3. ¿Ha recibido prácticas de laboratorio acerca de microprocesadores?

Si No

4. Si su respuesta es positiva ¿Contó con el equipo especializado para realizar dichas prácticas de laboratorio?

Si

No

5. ¿Cree usted que existe suficiente equipo en el laboratorio de hardware de la FMO para realizar las prácticas de laboratorio de la mejor manera, para la asimilación de los conocimientos?

Si

No

6. ¿Cómo le gustaría que fueran sus prácticas de laboratorio en la FMO?

7. ¿Cree usted que es importante la implementación de un laboratorio que cuente con equipo adecuado para el trabajo con microprocesadores?

Si

No

8. ¿Considera importante la aplicación práctica en el ámbito laboral de los conocimientos adquiridos de microprocesadores?

Si

No

9. ¿Piensa que los conocimientos teóricos adquiridos en las clases deberían reforzarse con prácticas?

Si

No

Interpretación de Encuesta orientada a los estudiantes de la carrera de ingeniería de Sistemas Informáticos de la asignatura de microprogramación y materias afines al área de electrónica en la Facultad Multidisciplinaria de occidente.

Objetivo: Conocer el porcentaje de estudiantes que poseen conocimientos teóricos- prácticos acerca de la electrónica y programación con microprocesadores.

1. ¿Posee algún conocimiento teórico-práctico acerca de electrónica y programación con microprocesadores?

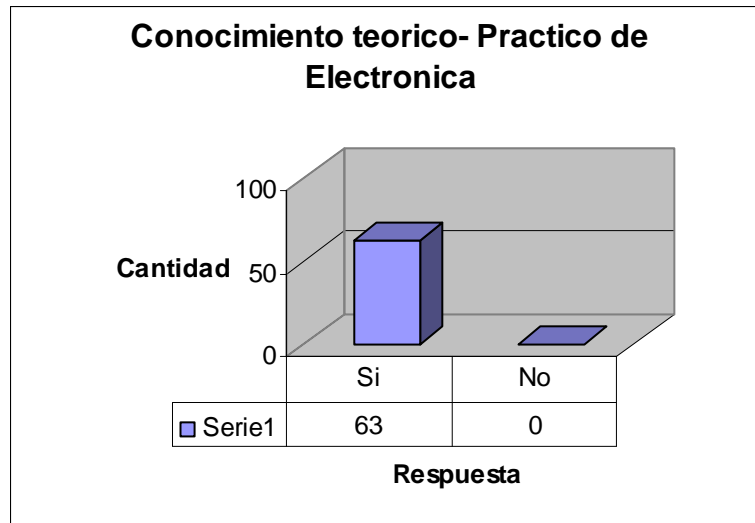


Fig. A2.1 – Grafico de conocimiento teórico-práctico

Se ha determinado a través de la muestra todos los estudiantes de la carrera si poseen conocimientos teóricos – prácticos acerca de electrónica y programación con microprocesadores.

Objetivo: Identificar el lugar donde los estudiantes obtuvieron los conocimientos acerca de la electrónica y así conocer el porcentaje de alumnos que obtuvieron los conocimientos dentro de la UESFMO

2. ¿En qué lugar ha obtenido usted los conocimientos acerca de electrónica y programación con microprocesadores?

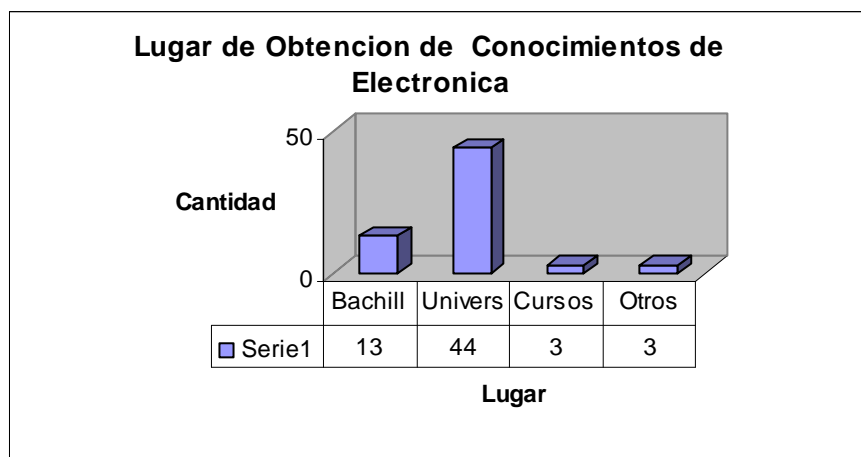


Fig. A2.2 – Grafico de obtención de conocimientos

Por medio de la muestra utilizada se logro conocer el lugar en el cual los estudiantes obtuvieron sus conocimientos sobre electrónica: 13 de los estudiantes obtuvieron sus conocimientos en bachillerato, 44 estudiantes en la universidad, 3 de los estudiantes en cursos adicionales y 3 de los estudiantes lo obtuvo por otros medios.

Objetivo: Verificar si los estudiantes obtuvieron información acerca de la programación de microprocesadores, a la hora de haber adquirido los conocimientos acerca de electrónica.

Objetivo: Investigar si los estudiantes de Ingeniería de Sistemas Informáticos, han recibido prácticas de laboratorio acerca de los microprocesadores

3. ¿Ha recibido prácticas de laboratorio acerca de microprocesadores?

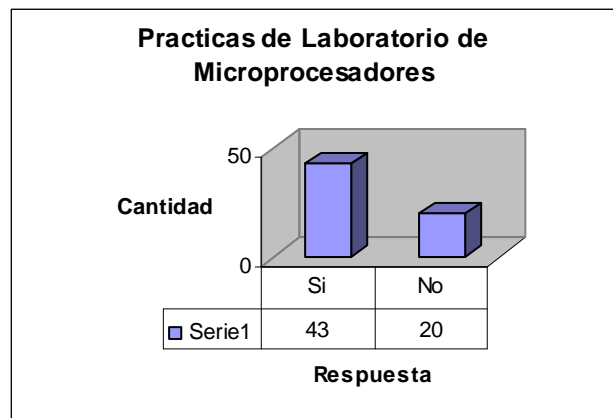


Fig. A2.3 – Grafico de prácticas de laboratorio

De la muestra tomada de estudiantes encuestados 43 de ellos han recibido prácticas de laboratorio relacionadas a la programación y manejo con microprocesadores; a pesar de esto 20 estudiantes no han recibido prácticas de este tipo.

Objetivo: Dar a conocer si los estudiantes de Ingeniería de Sistemas Informáticos que recibieron prácticas de laboratorio acerca de microprocesadores obtuvieron equipos especializados para poder desarrollar las prácticas de la mejor forma posible.

4. Si su respuesta es positiva ¿Contó con el equipo especializado para realizar dichas prácticas de laboratorio?

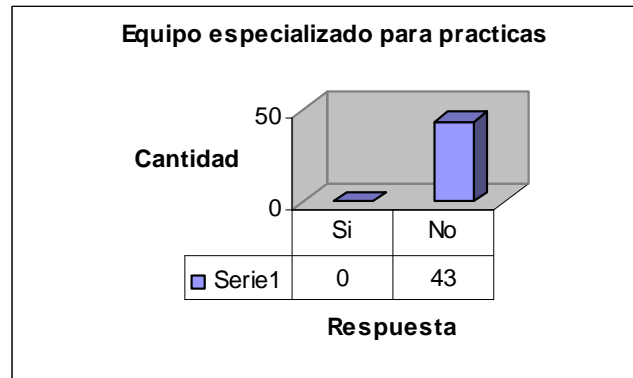


Fig. A2.4 – Grafico de equipo especializado para prácticas

La grafica demuestra que no se cuenta con el equipo especializado para el trabajo con microprocesadores puesto que ninguno de los estudiantes que ha recibido practicas con microprocesadores ha trabajado con equipo especializado para ello.

Objetivo: Identificar si en el laboratorio de Hardware de la FMO se posee el equipo suficiente y adecuado para realizar laboratorios prácticos acerca de microprocesadores.

5. ¿Cree usted que existe suficiente equipo en el laboratorio de hardware de la FMO para realizar las prácticas de laboratorio de la mejor manera, para la asimilación de los conocimientos?

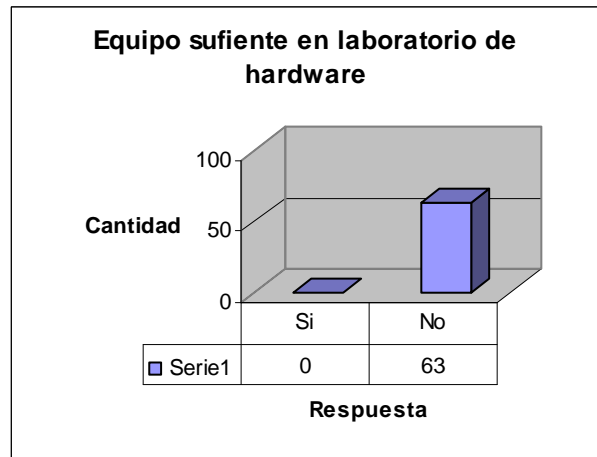


Fig. A2.5 – Grafico de Equipo suficiente en laboratorio

Como se logra observar que en la FMO los estudiantes de la carrera de ingeniería en Sistemas informáticos consideran que no se posee el equipo suficiente y el adecuado para el desarrollo de prácticas de laboratorio de hardware para el trabajo con microprocesadores.

Objetivo: Conocer desde el punto de vista de los estudiantes de la FMO, lo que ellos consideran más importante a la hora de realizar sus prácticas de laboratorio con microprocesadores

6. ¿Cómo le gustaría que fueran sus prácticas de laboratorio en la FMO?

Según la opinión de los estudiantes lo que ellos consideran más importante es la implementación de los conocimientos teóricos vistos en clases de forma práctica así como también incrementando el tiempo de cada una de los laboratorios prácticos y orientándolo a casos de estudio aplicados al ámbito laboral. Las prácticas deberían de ser más participativas y que la persona encargada de impartir las practicas posea los conocimientos necesarios para poder desarrollar la temática adecuada para la correcta asimilación de los conocimientos de los estudiantes.

Objetivo: Conocer si es necesaria e importante la implementación de un laboratorio que cuente con equipo adecuado para el trabajo con microprocesadores

7. ¿Cree usted que es importante la implementación de un laboratorio que cuente con equipo adecuado para el trabajo con microprocesadores?

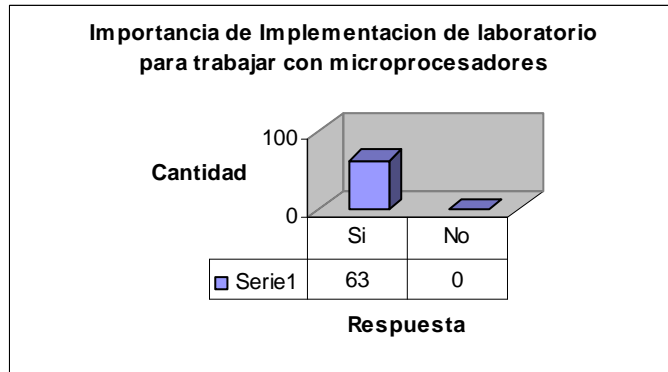


Fig. A2.6 – Grafico de implementación de laboratorios para microprocesadores

De acuerdo con los resultados mostrados anteriormente se puede determinar que todos los estudiantes encuestados consideran necesario la implementación de equipo especializados para el trabajo con microprocesadores a utilizar en el laboratorio de hardware.

Objetivo: Identificar la importancia de la aplicación de la práctica acerca de los conocimientos adquiridos de microprocesadores en la vida laboral

8. ¿Considera importante la aplicación práctica en el ámbito laboral de los conocimientos adquiridos de microprocesadores?

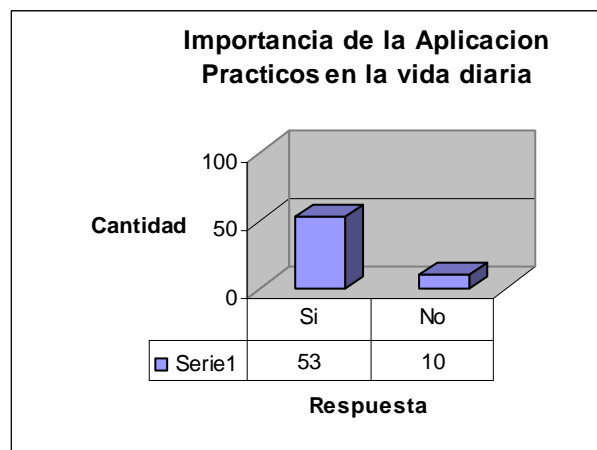


Fig. A2.7 – Grafico de aplicación práctica

La tendencia ilustrada en la grafica anterior refleja que es muy importante la aplicación de los conocimientos prácticos aplicados a la vida laboral y a problemas que se puedan dar en ese ámbito.

Objetivo: Investigar si los conocimientos teóricos adquiridos en la clase deben tener refuerzo con el desarrollo de prácticas en los laboratorios.

9. ¿Piensa que los conocimientos teóricos adquiridos en las clases deberían reforzarse con prácticas?

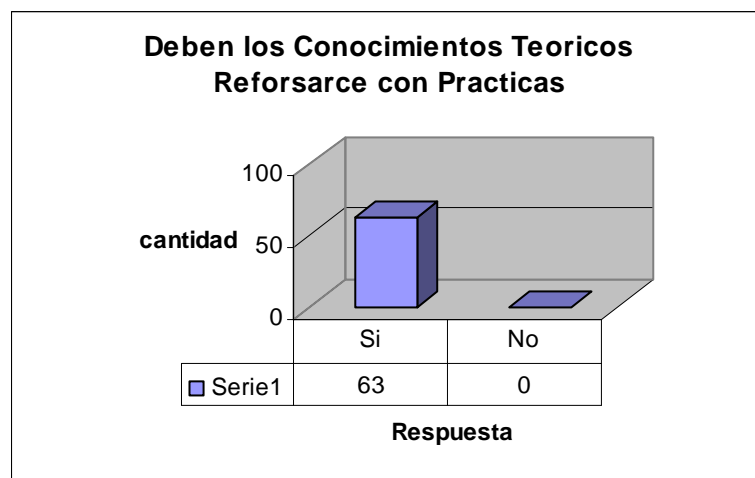


Fig. A2.8 – Grafico de refuerzo de prácticas

En la totalidad, la muestra de los estudiantes considera necesario el reforzar los conocimientos teóricos con prácticas de laboratorio para incrementar la asimilación de los conocimientos de los alumnos.

ANEXO 3: FORMATO E INTERPRETACIÓN DE ENCUESTA DIRIGIDA HACIA LOS DOCENTES, AUXILIARES DE CÁTEDRA E INSTRUCTORES ENCARGADOS DE IMPARTIR LAS MATERIAS RELACIONADAS EN LA ÁREA DE MICROPROGRAMACIÓN Y ELECTRÓNICA

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



Título: Encuesta dirigida a los docentes, auxiliares de cátedra e instructores que ejercen en la carrera de ingeniería de Sistemas Informáticos relacionados a la asignatura de microprogramación y materias afines al área de electrónica en la Facultad Multidisciplinaria de occidente.

Se le agradece su colaboración por facilitar los datos que se le solicitan en esta encuesta con el objetivo de agilizar la recopilación de datos a partir de la opinión especializada de los docentes.

Objetivo: Conocer la situación actual en la que se encuentra la materia de microprogramación y las demás materias relacionadas con el área de la electrónica.

Indicaciones: Conteste las siguientes preguntas de la manera más clara y precisa. Seleccionando la opción más conveniente.

1. ¿Ha impartido alguna vez clases relacionadas al área de microprogramación?

Si

No

2. En caso positivo, ¿con que obstáculos se encontró en el desarrollo de la materia?

3. ¿Considera que es necesario apoyarse en los conocimientos prácticos para aumentar la asimilación de los conocimientos teóricos que son desarrollados en la materia impartida?

Si No

¿Por qué?

4. ¿Ha impartido alguna vez laboratorios prácticos para la cátedra de microprogramación?

Si No

¿Qué tipo de herramientas?

Hardware _____ Software (Simuladores) _____

5. ¿Contó con las herramientas o equipo especializado para realizar las prácticas de laboratorio exitosamente?

Si No

6. ¿Estaría usted dispuesto a participar en capacitaciones relacionadas con el área de microprogramación que permita enriquecer sus conocimientos en dicha área?

Si No

7. ¿Considera importante para la realización de las prácticas de laboratorio, la utilización de equipo especializado de hardware para el trabajo con microprocesadores?

Si No

¿Por qué?

8. ¿Cree usted que son importantes los conocimientos de microprogramación para el correcto desempeño en el ámbito laboral?

Si No

9. ¿Qué recomendaciones podría dar para la correcta asimilación de los conocimientos impartidos en la cátedra de microprogramación?

Interpretación de Encuesta dirigida a los docentes, auxiliares de cátedra e instructores que ejercen en la carrera de ingeniería de Sistemas Informáticos relacionados a la asignatura de microprogramación y materias afines al área de electrónica en la Facultad Multidisciplinaria de occidente.

Objetivo: Obtener la cantidad de personas que cuentan con los conocimientos y experiencias en el desarrollo de materias relacionadas con microprogramación.

1. ¿Ha impartido alguna vez clases relacionadas al área de microprogramación?

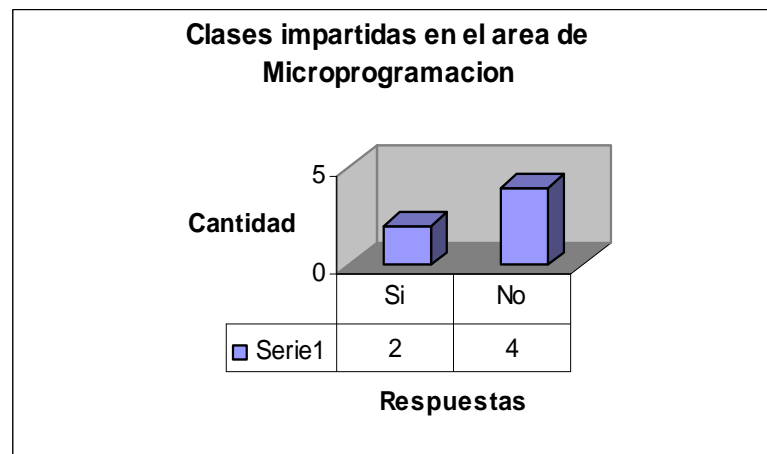


Fig. A3.1 – Grafico de clases impartidas de microprogramación

Por medio de la muestra analizada se logró conocer que la 4 de los docentes no han impartido clases de microprogramación.

Objetivo: Conocer las dificultades con las que los docentes, auxiliares de cátedra o instructores se encontraron al momento de impartir la materia

2. En caso positivo, ¿con que obstáculos se encontró en el desarrollo de la materia?

Según la respuesta de docentes encuestados estos se encontraron con la falta de información necesaria para impartir correctamente toda la temática de la asignatura, así como también con la inexistencia de equipo especializado para la realización de prácticas de laboratorio (Simuladores y Entrenadores).

Objetivo: Obtener parámetros que ayuden a determinar la importancia que tienen los conocimientos prácticos en el área de microprogramación.

3. ¿Considera que es necesario apoyarse en los conocimientos prácticos para aumentar la asimilación de los conocimientos teóricos que son desarrollados en la materia impartida?

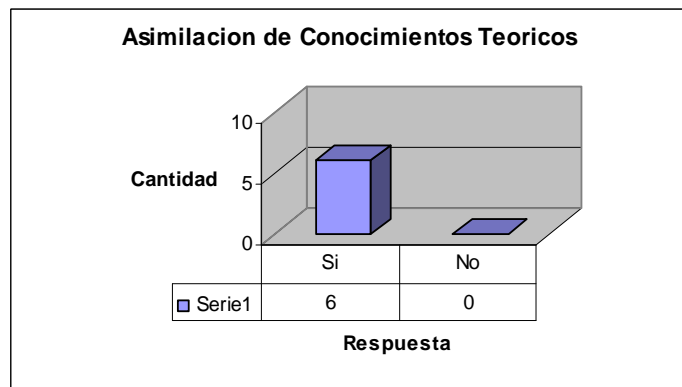


Fig. A3.2 – Grafico de asimilación de conocimientos teóricos

Toda la muestra encuestada respondió que es de suma importancia apoyar los conocimientos teóricos que se imparten en las clases con prácticas de laboratorio, que ambos conocimientos deben de ser complementados entre sí para obtener una mejor asimilación de los conocimientos.

Objetivo: Determinar la cantidad de personas que han impartido laboratorios prácticos acerca de microprogramación.

4. ¿Ha impartido alguna vez laboratorios prácticos para la cátedra de microprogramación?

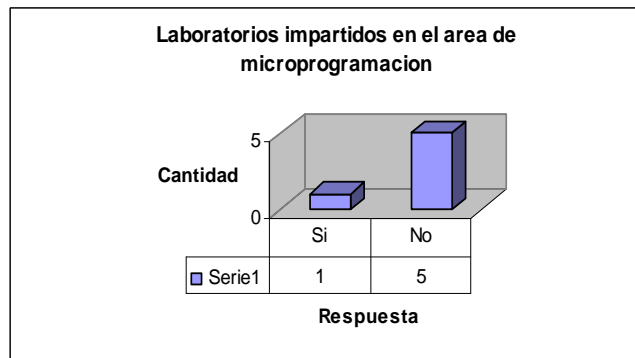


Fig. A3.3 – Grafico de laboratorios impartidos en el área de microprogramación

La mayoría de la muestra encuestada respondió que no ha impartido ninguna vez laboratorios prácticos para la cátedra de microprogramación.

Objetivo: Conocer el estado en el que son impartidos los laboratorios actualmente (existencias o no de equipos especializados).

- ¿Contó con las herramientas o equipo especializado para realizar las prácticas de laboratorio exitosamente?

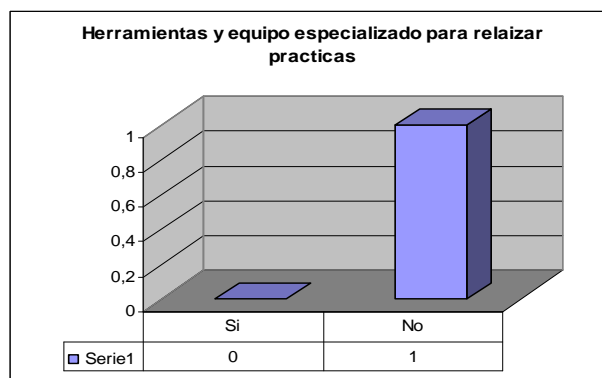


Fig. A3.4 – Grafico de herramientas y equipo especializado

Según los datos recopilados a través de las encuestas se logro determinar que no se contó con ningún tipo de equipo especializado (hardware) para trabajar con microprocesadores; sin embargo si se contó con el apoyo de software (Simuladores).

Objetivo: Conocer el nivel de disposición con el que cuentan las personas encargadas de impartir la materia de microprogramación y afines.

6. ¿Estaría usted dispuesto a participar en capacitaciones relacionadas con el área de microprogramación que permita enriquecer sus conocimientos en dicha área?

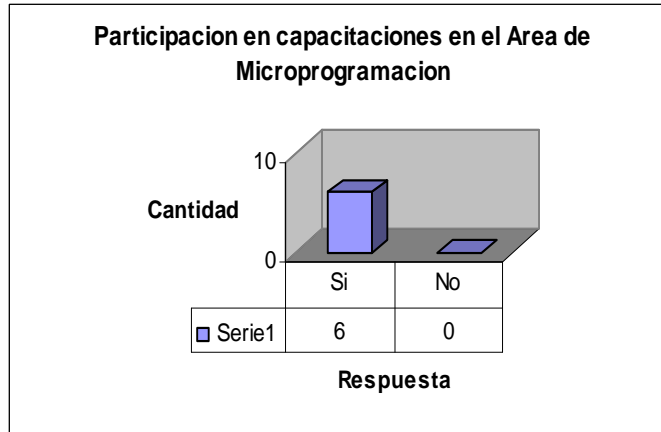


Fig. A3.5 – Grafico de participación en capacitaciones

Toda la muestra encuestada afirmo que si estaría dispuesto a participar en capacitaciones en el área de microprogramación con el objetivo de enriquecer los conocimientos en dicha área.

Objetivo: Obtener las impresiones de los encargados de las materias relacionadas con microprogramación acerca de la importancia de contar con equipo especializado para la realización de prácticas de laboratorio.

7. ¿Considera importante para la realización de las prácticas de laboratorio, la utilización de equipo especializado de hardware para el trabajo con microprocesadores?

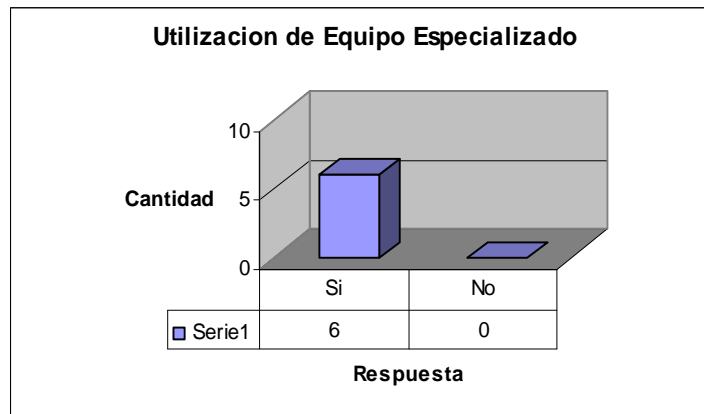


Fig. A3.6 – Grafico de utilización de equipo

Como se logra observar en la grafica anterior todos los encuestados opinan que si es importante la utilización de equipo especializado para la realización de prácticas de laboratorio, es importante para la correcta asimilación de los conocimientos; así como también para poder enseñar de mejor forma a los estudiantes.

Objetivo: Obtener una idea sobre la importancia de los conocimientos acerca de microprogramación aplicados en el área laboral.

8. ¿Cree usted que son importantes los conocimientos de microprogramación para el correcto desempeño en el ámbito laboral?

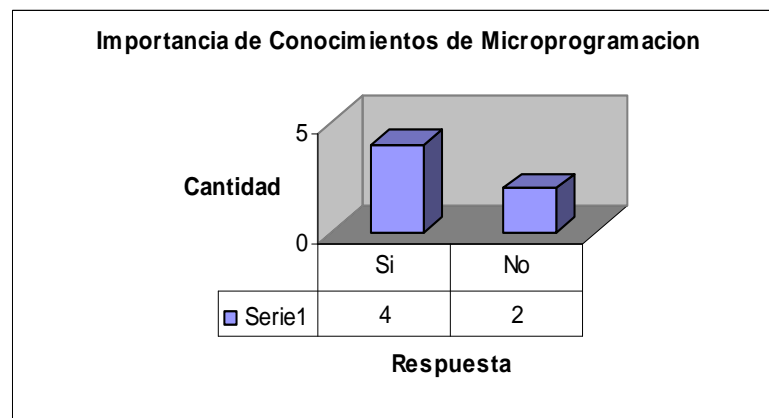


Fig. A3.7 – Grafico de importancia de conocimientos de microprogramación

De la muestra tomada para la realización de este estudio, 4 opinaron que los conocimientos acerca de microprogramación si son importantes para el correcto desempeño en el área laboral, y 2 encuestados opinaron que no es importante.

Objetivo: Tener diferentes puntos de vista e ideas de cómo se puede mejorar la asimilación de los conocimientos impartidos en la materia de microprogramación.

9. ¿Qué recomendaciones podría dar para la correcta asimilación de los conocimientos impartidos en la cátedra de microprogramación?

Las recomendaciones de los encuestados fueron que se debería de procurarse obtener más información acerca de microprogramación así como también

proveer a la FMO de más equipo especializado para prácticas de laboratorio (Entrenadores y Simuladores), así como también sería de gran ayuda hacer convenios con otras universidades y/o tecnológicos para compartir recursos tecnológicos. Además para incrementar los recursos (equipos software y hardware, capacitaciones) se debería de buscar ayuda internacional para implementar un buen laboratorio de microprocesadores.

ANEXO 4: FORMATO E INTERPRETACIÓN DE ENCUESTA DIRIGIDA HACIA ALUMNOS GRADUADOS DE INGENIERÍA DE SISTEMAS INFORMÁTICOS DE LA UES-FMO Y QUE LABORAN EN EL ÁREA DE MICROPROCESADORES Y AFINES.

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



Título: Encuesta enfocada a los ex-alumnos graduados de la carrera de ingeniería de Sistemas Informáticos en la Facultad Multidisciplinaria de occidente.

Se le gratifica con anticipación su valiosa colaboración al brindar su tiempo al desarrollo y ejecución de la investigación.

Objetivo: Conocer la situación actual de las aplicación de los microprocesadores en la industria y ambiente laboral.

Indicaciones: Conteste las siguientes preguntas de la manera más clara y precisa. Seleccionando la opción más conveniente.

1. ¿Se encuentra laborando actualmente?

Si No

2. ¿En qué área profesional se encuentra laborando actualmente?

3. ¿Qué conocimientos posee usted acerca de microprocesadores?

4. ¿En qué lugar obtuvo los conocimientos acerca de microprocesadores?

Bachillerato Cursos Adicionales

Universidad Ámbito Laboral

5. ¿En la empresa en la cual se encuentra laborando existe equipos o mecanismos que trabajen con microprocesadores (aparte de ordenadores)?

Si No

6. ¿Qué tipo de operaciones realizan dicho equipo dentro de la empresa? _____

7. ¿Ha utilizado alguna vez dicho equipo?

Si No

8. Si su respuesta anterior fue sí; ¿en qué forma a utilizado dicho equipo?

9. ¿Cree usted que el desarrollo de aplicaciones con microprocesadores tiene un amplio rango de operatividad en la industria?

Si No

10. ¿Considera usted que es necesario que en las instituciones educativas se implementen más programas de estudio para la programación de microprocesadores?

Si No

¿Porque? _____

11. ¿Considera usted que los laboratorios prácticos con microprocesadores es esencial para la correcta asimilación de los conocimientos teóricos?

Si No

¿Porque? _____

12. ¿Qué recomendaciones daría para mejorar el estudio de microprocesadores en las instituciones educativas? _____

Interpretación de: Encuesta enfocada a los ex-alumnos graduados de la carrera de ingeniería de Sistemas Informáticos en la Facultad Multidisciplinaria de occidente.

Objetivo: Averiguar en la condición laboral en la cual se encuentra el encuestado actualmente.

1. ¿Se encuentra laborando actualmente?

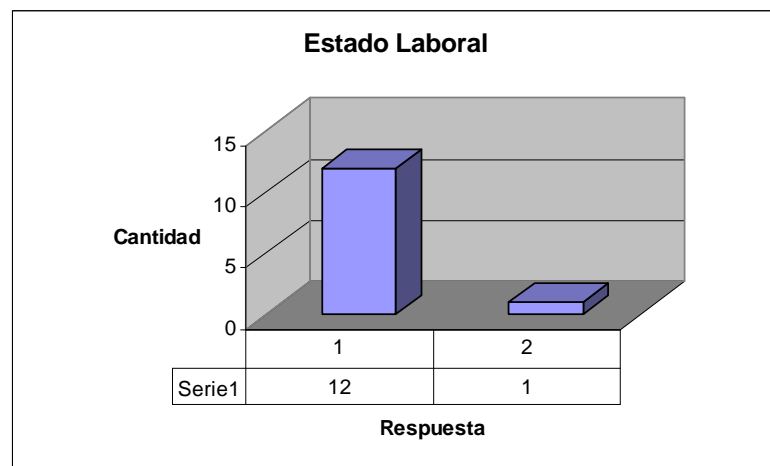


Fig. A4.1 – Grafico de Estado laboral

Como se logra apreciar en el grafico anterior 12 de los ex alumnos encuestados se encuentran desarrollando sus labores en el ámbito profesional y solamente 1 aún no.

Objetivo: Determinar las principales áreas en las cuales los profesionales están siendo solicitados en el área laboral en el país y en las cuales los ex alumnos se encuentran laborando.

2. ¿En qué área profesional se encuentra laborando actualmente?

La mayoría de los entrevistados se encuentran laborando en el área de análisis y desarrollo de sistema informáticos, y diseño de bases de datos. También una parte considerable de la población entrevistada trabaja proveyendo soporte técnico en el área de hardware (computadoras, equipo especializado de red, etc.).

Objetivo: Conocer el nivel de conocimientos asimilados en el área microprocesadores por los ex alumnos de la carrera de ingeniería de sistemas informáticos de la UES Facultad Multidisciplinaria de Occidente.

3. ¿Qué conocimientos posee usted acerca de microprocesadores?

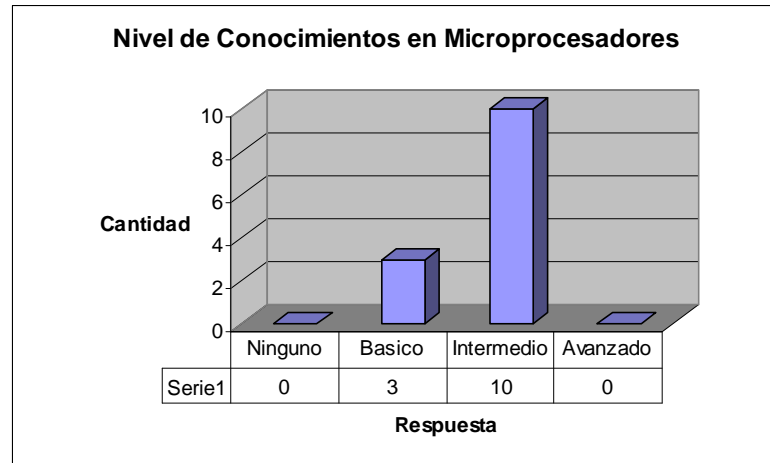


Fig. A4.1 – Grafico de nivel de conocimientos

En la grafica se muestra que el nivel de conocimiento en el área de trabajo con microprocesadores en su gran mayoría es intermedio, y varia a un conocimiento básico. No hay un nivel avanzado o especializado en el área de microprocesadores.

Objetivo: Conocer el lugar de estudio que permitió que los ex alumnos adquirieran sus conocimientos en el área de microprocesadores. Así como también conocer que tantos conocimientos sobre el área tenían al entrar a la universidad.

4. ¿En qué lugar obtuvo los conocimientos acerca de microprocesadores?

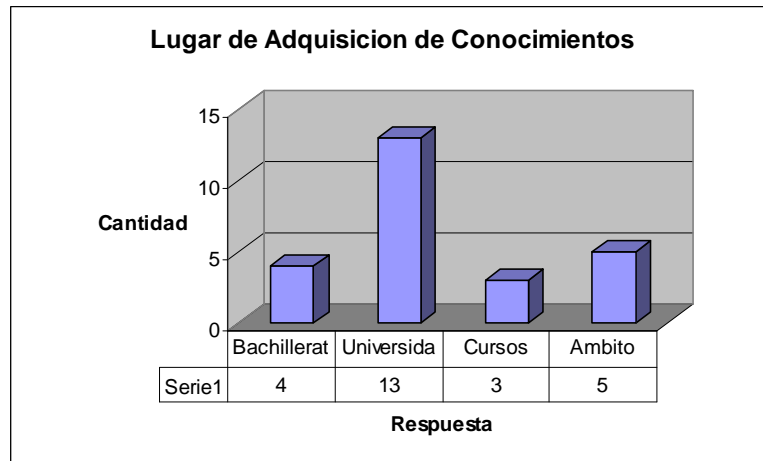


Fig. A4.3 – Grafico de adquisición de conocimientos

El lugar en el cual la mayor parte de la población ha adquirido los conocimientos de microprocesadores fue la Universidad pero igualmente algunos ex alumnos además han obtenido dichos conocimientos tanto en bachillerato, cursos adicionales como también directamente en el área laboral. Algunos encuestados obtuvieron sus conocimientos de microprocesadores en dos o más lugares.

Objetivo: Medir la necesidad en la cual se encuentran las empresas en la actualidad y así así determinar cuán necesaria es la capacitación y conocimientos en el área de microprocesadores.

- ¿En la empresa en la cual se encuentra laborando existe equipos o mecanismos que trabajen con microprocesadores (aparte de ordenadores)?

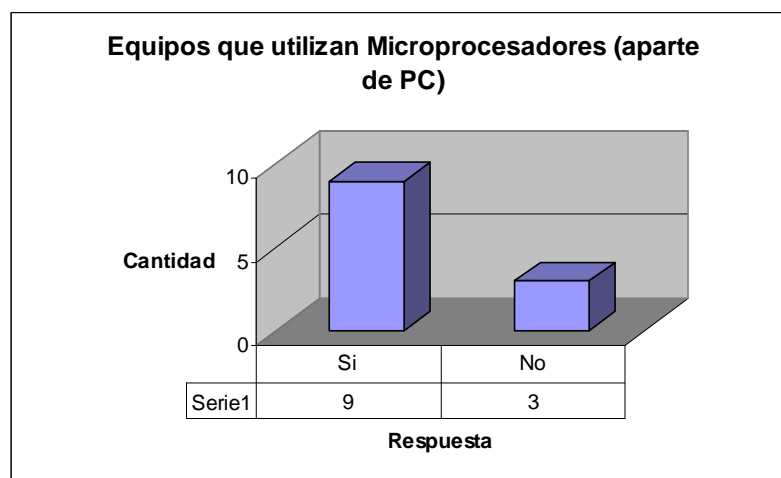


Fig. A4.4 – Grafico de equipos que utilizan microprocesadores

En las empresas en las cuales se desarrollan los profesionales encuestados se cuenta con computadores; pero con un mínimo margen de diferencia se como se muestra en la grafica, las empresas poseen o no otro tipo de equipo especializado que trabaje con microprocesadores.

Objetivos: Averiguar que tipos de equipos con los cuales se esta trabajando en las empresas, poseen como componente interno microprocesador(es)

6. ¿Qué tipo de operaciones realizan dicho equipo dentro de la empresa?

Entre los equipos especializados (que utilizan microprocesadores) que poseen las empresas en las cuales trabajan los encuestados además de la computadoras son equipos de redes informáticas (router, switch, etc.), telefonía IP, equipo para control de asistencia, y equipo para el manejo de maquinas industriales.

Objetivo: Medir el nivel de uso y manejo que los encuestados tienen sobre los equipos que poseen microprocesadores y determinar que operaciones han sido realizadas o qué tipo de trabajo se ha hecho sobre dicho equipo.

7. ¿Ha utilizado alguna vez dicho equipo? ¿En qué forma?

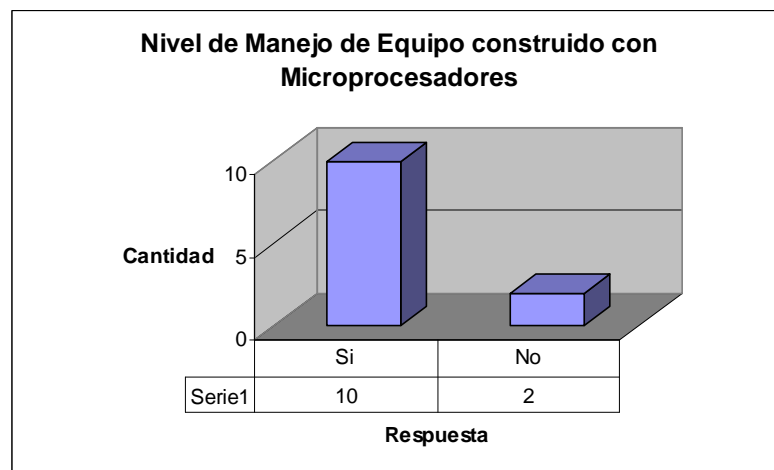


Fig. A4.5 – Grafico de nivel de manejo del equipo

En los lugares de trabajo, a los profesionales informáticos se les solicita en su mayoría el poder manejar y darle mantenimiento a los equipos que utilizan microprocesadores. Por medio del intercambio de equipos informáticos,

configuraciones administración y descarga de datos en redes informáticas así como también dando soporte a los compañeros de trabajo o clientes.

Objetivo: Medir el nivel de operatividad que se tiene en el desarrollo de aplicaciones con microprocesadores en la industria actual y así determinar que tan factible es el estudio de los microprocesadores en la UES FMO.

8. ¿Cree usted que el desarrollo de aplicaciones con microprocesadores tiene un amplio rango de operatividad en la industria?

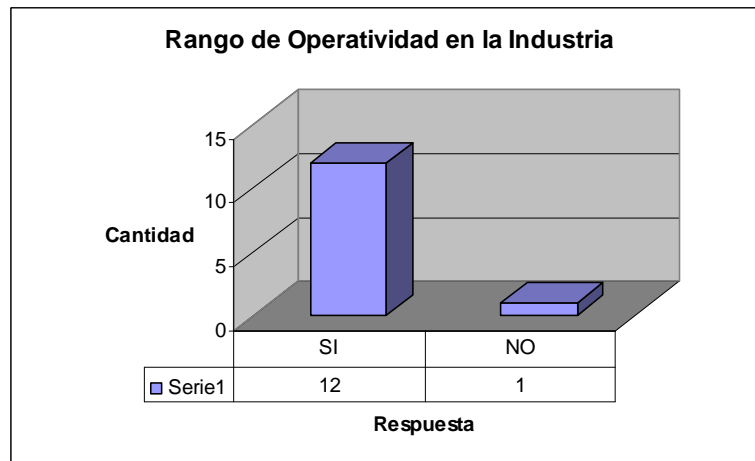


Fig. A4.6 – Grafico de rango de operatividad

La mayoría de los encuestados considera que el desarrollo de aplicaciones con microprocesadores posee un rango alto de operatividad en la industria actual en sus distintas areas.

Objetivo: Averiguar el grado de necesidad que se tiene en la implementacion de nuevos y mejores programas de estudio que promuevan el estudio con microprocesadores como un área en la cual se puedan desarrollar profesionales capaces.

9. ¿Considera usted que es necesario que en las instituciones educativas se implementen más programas de estudio para la programación de microprocesadores? ¿Porque?

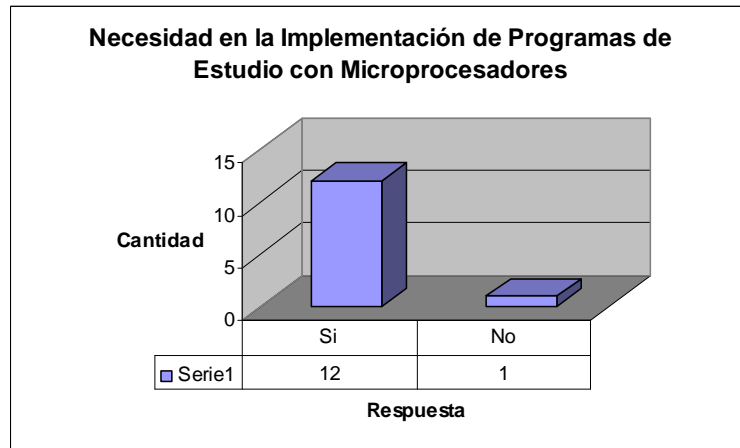


Fig. A4.7 – Grafico de implementación de programas de estudio

Es necesaria la implementación de nuevos y mejores programas de estudio que incluyan dentro de sí, la programación y manejo de microprocesadores. Esto puesto que en El Salvador no se cuenta con suficiente personal especializado en dicha área y esto podría mejorar el nivel de competitividad en la industria, además para saber manejar de una manera más óptima un equipo informático se debe conocer cuál es el funcionamiento interno, para saber localizar problemas que se puedan llegar a presentar y así poder darles una solución. Mejoraría además el nivel de investigación en el país lo cual es muy importante para su correcto desarrollo tanto económico como también industrial.

Objetivo: Calcular hasta qué punto y por qué son importantes los laboratorios prácticos para la comprensión adecuada de los conocimientos, sobre microprocesadores, que han sido impartidos en las clases teóricas.

10. ¿Considera usted que los laboratorios prácticos con microprocesadores son esenciales para la correcta asimilación de los conocimientos teóricos?

¿Porque?

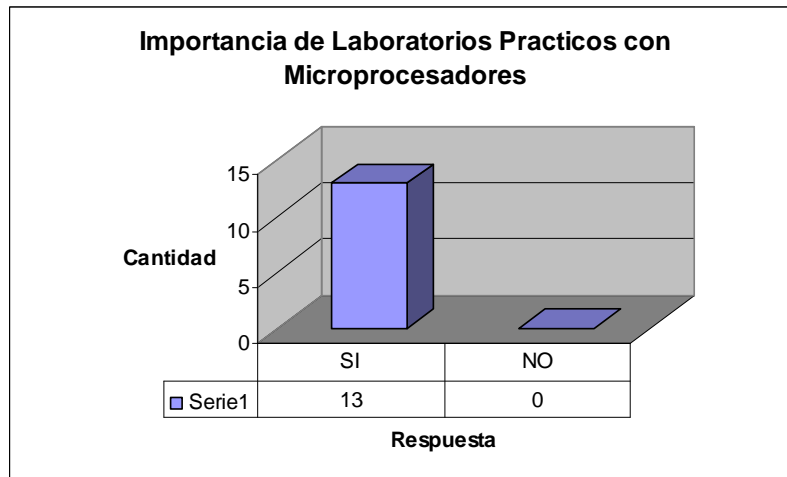


Fig. A4.4 – Grafico de importancia de laboratorios prácticos

Se considera de suma importancia el poder realizar laboratorios practicos para poder entender de una mayor manera los conocimientos teoricos impartidos en las clases; puesto que al poner en practicas dichos conocimientos permite despertar el interes de los estudiantes en el area no solo de microprogramadores sino tambien en todo el area tecnica. Enriqueciendo asi los conocimientos de los futuros profesionales y preparandolos de una mejor manera para la industria y ambito laboral; ofreciendo mejores servicios a las empresas.

Objetivo: Conocer algunas recomendaciones que los encuestados pudiesen tener para el estudio con microprocesadores; para poderlas tener en cuenta en el futuro.

11. ¿Qué recomendaciones daría para mejorar el estudio de microprocesadores en las instituciones educativas?

Toda las personas encuestadas coinciden en la necesidad de poder implementar equipos especializados para el trabajo con microprocesadores, además de la adquisición de guías practicas que permitan a los estudiantes aprender de forma didáctica estudiar posibles casos que puedan llegar a presentárseles en el futuro. Además de hacer hincapié en la necesidad de contar con personal especializado en dicha área, que pueda impartir sus conocimientos de una manera adecuada, proponiendo casos de la vida real y de producción, así los estudiantes a su debido tiempo podrán multiplicar también su conocimiento a otros.

ANEXO 5. CUESTIONARIO UTILIZADO EN LAS VISITAS TÉCNICAS REALIZADAS A LAS INSTITUCIONES EDUCATIVAS

Patrón de entrevistas

- ¿Cuenta la institución con laboratorio de microprocesadores (cuantos)?
- ¿Hace cuanto tiempo que fue implantado el laboratorio?
- ¿Cuánta es la población beneficiada con dicho laboratorio?
- ¿Qué tipo de equipo posee el laboratorio (modelos, marcas, cantidad, etc.)?
- ¿Cómo obtuvieron los medios para la implementación del laboratorio?
- ¿Qué laboratorios prácticos se suelen implementar para el estudio con microprocesadores?
- ¿Cuentan con un especialista en el área de microprogramación, que sea el encargado de dicho laboratorio?
- ¿Cuál es programa de estudio que implementan en esta institución para trabajar con microprocesadores?
- ¿Cuánto es el tiempo de duración de cada práctica de laboratorio?
- ¿Considera usted que los laboratorios prácticos con microprocesadores es esencial para la correcta asimilación de los conocimientos teóricos?
- ¿Cree que al poseer un laboratorio para trabajar con microprocesadores aporta a la institución mayor prestigio y competitividad en el ámbito educativo?
- ¿Considera que en el país se cuenta con una amplia demanda laboral en esta área?
- ¿Considera importante que el gobierno aporte más recursos financieros al desarrollo de profesionales especializados en esta área? ¿Porque?
- ¿Qué microprocesador recomienda para el estudio del mismo en el área educativa?

ANEXO 6. VISITAS TÉCNICAS REALIZADAS A LAS INSTITUCIONES EDUCATIVAS.

PREGUNTAS	INSTITUTO TÉCNICO RICALDONE	ITCA	INSTITUTO SANTA LUCIA	UNIVERSIDAD DON BOSCO
¿Cuenta la institución con laboratorio de microprocesadores (cuantos)?	3 laboratorios, de los cuales uno es de lógica secuencial, lógica digital y telecomunicaciones	1 laboratorio de microprocesadores	Cuenta con un laboratorio de microprocesadores	Cuenta con un laboratorio de microprocesadores
¿Hace cuanto tiempo que fue implantado el laboratorio?	Fundados en el año de 1986 y su última actualización fue a finales del año 2000	La implantación del primer laboratorio fue en el año de 1994 y su última actualización en el 2008 por los equipos de KOREA	Fundado en el año de 1996 y su última actualización fue en el año de 2002	Se carece de la fecha exacta de su fundación pero tuvo su última actualización hace dos años en el local que se encuentran ubicados actualmente (2007)
¿Cuánta es la población beneficiada con dicho laboratorio?	La población beneficiada directamente son 120 alumnos, y tomando en cuenta la población indirectamente beneficiada es de 450 alumnos	La población beneficiada directamente es de 230 alumnos de segundo año de la Carrera de Técnico de Computadoras, Electrónica Industrial y Telecomunicaciones.	150 alumnos por cada año.	150 alumnos beneficiados entre todas las materias que utilizan el laboratorio cada ciclo.
¿Qué tipo de equipo posee el laboratorio (modelos, marcas, cantidad, etc.)?	Equipo israelita de la marca DEGEN, 8085 Equipo canadiense de la marca LABVOLT 8080	Son 3 equipos Diferentes entre ellos: 1 Equipo Elaborado Artesanalmente, 8085 8 Equipos Donados por la Comunidad Jesuita, 8085	Poseen 10 entrenadores marca DeLorenzo i386 marca DL3155E24, 5 CPU z80 de electrónica VENETA (Italianos), 20 bases universales y 25	20 computadoras IBM entre Altiva y PC 300 PL, 5 equipos para regular el voltaje (mini laboratorios – IDL-800 Digital Lab) y 4 tarjetas

		8 Equipos Donados por KOREA. Marca MIDA Multimicom UP-2	tarjetas o placas modulares	modulares (Hecha por el personal de planta), además de utilizar simuladores (Emu8086)
¿Cómo obtuvieron los medios para la implementación del laboratorio?	El equipo fue obtenido por fondos propios de la institución	El equipo fue obtenido por una donación de la república de KOREA y por la comunidad de los Jesuitas	El BCIE (Banco Centroamericano de integración económica) dono el z80, y Fondos del gobierno (MINED) y Unión Europea dono el i386	Por donaciones de diversas empresas como: CESSA, Corporación Amadeus y parte una inversión inicial así como también creaciones propias del personal de planta.
¿Qué laboratorios prácticos se suelen implementar para el estudio con microprocesadores?	20 prácticas de laboratorio entre estas: instrucciones básicas, especiales y lenguaje ensamblador.	Conocer arquitectura interna, externa y programación (Lenguaje Ensamblador)	Guías planificadas a través de software que han sido desarrolladas por los docentes como arquitectura de los microprocesadores, instrucciones básicas, simulaciones, etc., Aun no se han implementado prototipos en hardware.	Registros internos, operaciones lógicas, aritméticas y con caracteres, operaciones con periféricos (puerto serial y paralelo), manejo de tabla de pantalla, etc.
¿Cuentan con un especialista en el área de microprogramación, que sea el encargado de dicho laboratorio?	2 ingenieros electrónicos 1 ingeniero electricista y 3 técnicos electricistas	5 personas especialistas, capacitadas en la República de KOREA	2 Especialistas y desde el año 2003 existen 4 docentes de planta, los cuales se capacitaron 2 docentes en Colombia, Costa Rica , España y México	5 personas de práctica que son docentes de planta, además de 2 personas encargadas de impartir clases teóricas.
¿Cuál es programa de estudio que implementan en esta institución para trabajar con microprocesadores?	Utilizan el plan de estudio del año de 1996 del Ministerio de Educación, y lo adecuan cada año de acuerdo a las nuevas necesidades.	Utilizan el plan de estudio del año 2007	Plan modular, Curricula Renovada o Educación por competencia.	Plan de estudio del año 2007 para las jornadas matutina y vespertina cuya duración es de 2 años (27 materias y 1 trabajo de graduación)

¿Cuánto es el tiempo de duración de cada práctica de laboratorio?	30 horas semanales de practica distribuidas en 3 áreas lo que corresponde a 10 horas practicas semanales por área.	3 horas de laboratorios prácticos semanales, en total 48 horas totales	El tiempo de duración es relativo, 6 horas de prácticas por día de las cuales se realizan de 2 a 3 guías por jornada matutina	Dos horas de práctica por semana más tres horas de clases teóricas
¿Considera usted que los laboratorios prácticos con microprocesadores son esenciales para la correcta asimilación de los conocimientos teóricos?	Sí, todo el enfoque en educación es constructivista y la práctica se vuelve esencial en la asimilación de conocimientos.	Definitivamente la práctica es esencial. De hecho todo el plan educativo utilizado dentro de la institución es 60% practico y 40% teórico	Definitivamente si son esenciales para mayor asimilación de los estudiantes	Todo conocimiento involucra hacer algo práctico para lograr asimilar de una mejor manera los conocimientos teóricos lo que implicar utilizar todos los sentidos.
¿Cree que al poseer un laboratorio para trabajar con microprocesadores aporta a la institución mayor prestigio y competitividad en el ámbito educativo?	Según el punto de vista del encargado de los laboratorios, el poseer laboratorios no da prestigio para esa institución, lo considera algo trascendental.	Si, ayuda tanto a la industria como al prestigio de la institución, en el sentido de que tanto docentes como instructores puedan transmitir los conocimientos hacia los alumnos como a personas ajenas a la institución en base a experiencias.	Si aporta prestigio y competitividad, pero depende de otros factores externos, como ubicación geográfica, nivel económico, nivel delincuencia, etc.	Sí, porque es un área técnica de capacitación. El prestigio está ligado con la especialización en las carreras
¿Considera que en el país se cuenta con una amplia demanda laboral en esta área?	En la industria existe una amplia diversidad de conocimientos en diversas áreas no solo en microprocesadores.	Si existe una amplia demanda laboral en diferentes sectores uno de los principales es la industria textil.	La demanda disminuye debido a la falta de programadores en microprocesadores	Si, ya te todo aparato electrónico especializado contiene microprocesadores, lo cual permite diagnosticar y reparar los aparatos, esto es aplicable a diferentes áreas entre estas están consultoría y lo que es el área de auditoria
¿Considera importante que el gobierno aporte más recursos?	Si es necesario que el Ministerio de Educación	Si, es importante ya que el gobierno actualmente está	Es importante que aporte más recursos financieros al	Si, por los desarrollos de automatización e

recursos financieros al desarrollo de profesionales especializados en esta área? ¿Porque?	actualice los equipos, modelos, así como también que aporte capacitaciones al personal docente y desarrolle nuevas curriculas.	colaborando con la construcción de los nuevos Megatec tanto en la Unión como en Zacatecoluca, pero también podría ayudar a formar profesionales en el área de electrónica desde el bachillerato.	desarrollo de nuevas tecnologías especialmente en microprocesadores porque si hubiese más personas capaces de trabajar con esta tecnología permitiría al país salir del subdesarrollo.	investigación, se debería desarrollar una política de educación tecnológica que ayude a la tecnificación del país, esto mejoraría el nivel el país.
¿Qué microprocesador recomienda para el estudio del mismo en el área educativa?	EL microprocesador 8085, por su sencillez, direccionamiento de memoria básico.	EL microprocesador 8085 y el microprocesador 8088	El i386 porque es con el que desarrollan sus prácticas, pero cualquier microprocesador permite al estudiante aprender sobre la arquitectura de los mismos, todo depende del maestro si trata o no de cumplir con los objetivos de la institución.	El microprocesador del 8086, puesto que la arquitectura no cambia. Los microprocesadores modernos no son viables por ser muy complejos.

Tabla 1 - Síntesis De Visitas Técnicas

INSTITUCIÓN	NOMBRE DEL ENCARGADO	CARGO	CONTACTO	DIRECCIÓN
INSTITUTO TÉCNICO RICALDONE	Ing. Gilberto Moto	Coordinador del área de Electrónica	Tel: 2234-6005 Fax: 2234-6085	Av. Aguilares # 218 Centro Urbano Libertad San Salvador
ITCA	Alexander Alas Castillo	Coordinador de la Carrera de Mantenimiento de Computadoras. Departamento de Ingeniería Eléctrica	Tel: 2514-7511 Email: eaalas@itca.edu.sv	Km 11 carretera a Santa Tecla, La libertad
INSTITUTO SANTA LUCIA	Tec. Juan Carlos Mártir Hernández Ing. Sabas	Encargado de los laboratorios de electrónica y eléctrica	Tel: 2294-0841	Col. Santa Lucia Calle 5 Ilopango
UNIVERSIDAD DON BOSCO	Ing. Carias Ing. Oscar Vizcarra Ing. Néstor Lozano	Jefe del Departamento de Electrónica Jefe del Departamento de Ingeniería Docente encargado de la materia de microprocesadores y relacionadas	Tel: 2251-5000 ext. 1825 Tel: 7850-5767 duran_vizcarra@gmail.com	Calle Plan del Pino, Cantón Venecia, Soyapango, San Salvador

Tabla 2 - Información De Contactos De Visitas Técnicas

ANEXO 7. IMÁGENES DE VISITAS TÉCNICAS REALIZADAS A LAS INSTITUCIONES EDUCATIVAS

Anexo 7.1 Instituto Técnico Ricaldone



Fig. A7.1
Instituto Técnico Ricaldone,
Laboratorio de Microprocesadores,
Motor paso a paso



Fig. A7.2
Instituto Técnico Ricaldone,
Laboratorio de Microprocesadores,
Desarrollo de Guías prácticas



Fig. A7.3
Instituto Técnico Ricaldone,
Laboratorio de Microprocesadores,
Entrenador de Microprocesador 8086



Fig. A7.4
Instituto Técnico Ricaldone,
Laboratorio de Microprocesadores,
Trabajo en el laboratorio

Anexo 7.2 ITCA



Fig. A7.5
ITCA-FEPADE,
Laboratorio de Microprocesadores,
MIDA Multimicom UP-2



Fig. A7.6
ITCA-FEPADE,
Aula de Laboratorios con
Microprocesadores



Fig. A7.7
ITCA-FEPADE,
Laboratorio de Microprocesadores,
Jooyotech



Fig. A7.8
ITCA-FEPADE,
Laboratorio de Microprocesadores,
Equipo de Monitoreo



Fig. A7.9
ITCA-FEPADE,
Laboratorio de Microprocesadores,
Alexander Alas – Coordinador de la Carrera
de Mantenimiento de Computadoras



Fig. A7.10
ITCA-FEPADE,
Laboratorio de Microprocesadores,
Colaboración de la República de Corea



Fig. A7.11
ITCA-FEPADE,
Laboratorio de Electrónica Digital y
Microprocesadores, DELL



Fig. A7.12
ITCA-FEPADE,
Laboratorio de Electrónica Digital y
Microprocesadores, DEL 8085

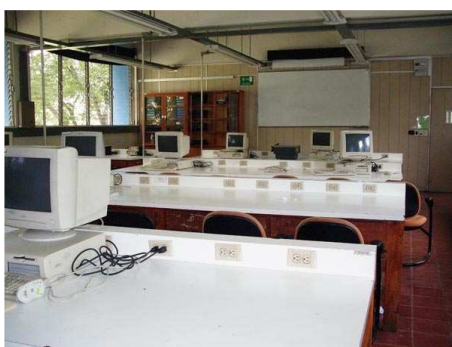


Fig. A7.13
ITCA-FEPADE,
Laboratorio de Electrónica Digital y
Microprocesadores

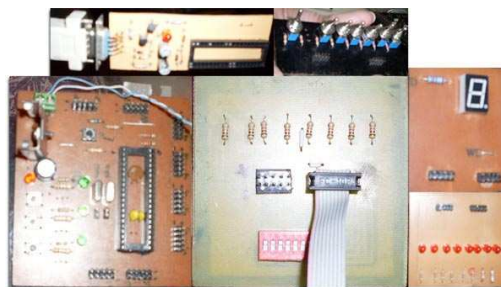


Fig. A7.14
ITCA-FEPADE,
Laboratorio de Electrónica Digital y
Microprocesadores, Equipo elaborado en
ITCA

Anexo 7.3 Instituto Nacional Santa Lucia



Fig. A7.15
Instituto Nacional Santa Lucia, Laboratorio de Microprocesadores, Modulo de Microprocesador De Lorenzo I386



Fig. A7.16
Instituto Nacional Santa Lucia, Laboratorio de Microprocesadores, Modulo de Microprocesador De Lorenzo I386 en su Base

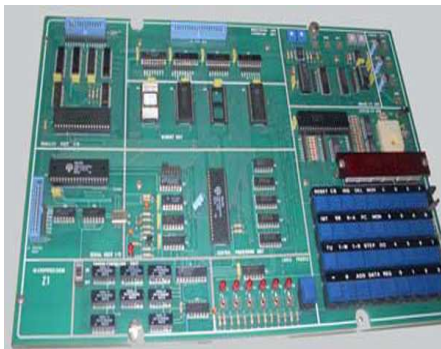


Fig. A7.17
Instituto Nacional Santa Lucia, Laboratorio de Microprocesadores, Modulo de Microprocesador Z1



Fig. A7.18
Instituto Nacional Santa Lucia, Laboratorio de Microprocesadores, Modulo de Microprocesador Z1 en su Base



Fig. A7.19
Instituto Nacional Santa Lucia, Laboratorio de
Microprocesadores, Módulos de
Microprocesadores Z1 E I386 en Su Base



Fig. A7.20
Instituto Nacional Santa Lucia,
Laboratorio de Microprocesadores,
Caja de Herramientas para el Trabajo
con Microprocesadores

Anexo 7.4 Universidad Don Bosco



Fig. A7.21
Universidad Don Bosco,
Laboratorio de Microprocesadores.



Fig. A7.22
Universidad Don Bosco,
Equipo Utilizado para prácticas de
Microprocesadores.



Fig. A7.23
Universidad Don Bosco,
Laboratorio de Microprocesadores, Entrenador
de microprocesador.



Fig. A7.24
Universidad Don Bosco,
Laboratorio de Microprocesadores,
Base reguladora de Voltaje

ANEXO 8. GUÍAS PRÁCTICAS DE LABORATORIO

Anexo 8.1 Guía de Practica N° 1

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 1 Electrónica digital para computadoras

OBJETIVO.

Conocer el sistema binario y hexadecimal, así como también realizar conversiones entre estos sistemas. Aprender el significado de más y menos significativo. Trabajar con operaciones And, Or, Not, y Exor.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO

PRIMERA PARTE: SISTEMAS NUMÉRICOS

Un microprocesador ejecuta toda la aritmética en binario, aunque puede ser traducido a formas diferentes (ej. decimal, hexadecimal). El sistema numérico binario consiste en los números 0 y 1 que sea llamado dígitos binarios o bits para abreviar. Existen varias palabras que se acostumbran usar para representar los números binarios 0 y 1 y son a menudo usados de modo intercambiable, dependiendo del contexto en que son usados. Como sigue:

BINARIO 1 = Verdadero, encendido, alto, 5 voltios

BINARIO 0 = Falso, apagado, bajo, 0 voltios

En el microprocesador 8085, los números binarios son organizados en grupos de 8

bits que es a lo que equivale un byte y grupos de 16 bits que son llamadas palabras. Cuando se refiere a un byte, a menudo es necesario describir los bits particulares, así la numeración de cada uno de los 8 bits es como sigue:

7 6 5 4 3 2 1 0

Así en el número binario, 10001000, el bit 7 y el bit 3 son 1 y el resto es 0. En el número 00010001 binario, el bit 4 y el bit 0 es un valor de 1 y el resto es 0. Cuando se refiere a una palabra, los bits son numerados de la siguiente manera:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Como usted sabe, en los números decimales, cada posición de un dígito tiene un peso diferente. Por ejemplo en el número 1732 decimal: (Los números de dígito son leídos de derecho a izquierda, 0 a 3)

# Dígito	Peso	*	Valor	=	Resultado Decimal
3	10^3	*	1	=	1000
2	10^2	*	7	=	700
1	10^1	*	3	=	30
0	10^0	*	2	=	2
Suma de los números por su peso					1732

Del mismo modo el sistema binario hay pesos para cada posición. El peso es 2 elevado a la potencia del número de bits ($2^{(\text{número de bits})}$). El número 11111111 binario se puede convertir al decimal para saber el peso de cada uno, como en el ejemplo siguiente:

# Bit	Peso	*	Valor	=	Resultado Decimal
7	10^7	*	1	=	128
6	10^6	*	1	=	64

$$5 \quad 10^5 \quad * \quad 1 \quad = \quad 32$$

$$4 \quad 10^4 \quad * \quad 1 \quad = \quad 16$$

$$3 \quad 10^3 \quad * \quad 1 \quad = \quad 8$$

$$2 \quad 10^2 \quad * \quad 1 \quad = \quad 4$$

$$1 \quad 10^1 \quad * \quad 1 \quad = \quad 2$$

$$0 \quad 10^0 \quad * \quad 1 \quad = \quad 1$$

Suma de los bits por su peso 255

Si una de las posiciones de los bits hubiera sido un cero, ese peso del bit de esa posición no podría haber sido añadido a la suma.

Para convertir un número decimal para otra base divide repetidamente el número por la base hasta el cociente igual a cero.

Por ejemplo para encontrar el binario (basar 2) del equivalente del número 58 decimal procedemos como sigue:

$$58/2 = 29 \text{ con un residuo de } 0 \text{ (LSB)}$$

$$29/2 = 14 \text{ con un residuo de } 1$$

$$14/2 = 7 \text{ con un residuo de } 0$$

$$07/2 = 3 \text{ con un residuo de } 1$$

$$03/2 = 1 \text{ con un residuo de } 1$$

$$01/2 = 0 \text{ con un residuo de } 1 \text{ (MSB)}$$

El equivalente binario del número 58 en decimal es por lo tanto: 111010. El número 111010 binario puede ser verificado convirtiendo de nuevo a decimal. La Conversión de decimal a hexadecimal se puede realizar dividiendo el número decimal entre 16 en lugar de 2. El residuo al dividirse entre 16 recorrerá de 0 a 15, o 0 a F en hex. Las tres formas que se usaran en este manual es: binario, hexadecimal (hex., para abreviar) y decimal.

A continuación se muestra un cuadro que representa los números decimales del 0 al 20 y su equivalente en binario y hexadecimal

Decimal	Binario	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

El Hexadecimal se utiliza para representar valores binarios porque es más eficiente y es muy fácil convertir los números de binario a hexadecimal. Por ejemplo, para convertir el siguiente número de binario a hexadecimal:

101101101101011

Se Inicia haciendo grupos de cuatro dígitos binarios (bits) de derecha a izquierda. Si no hay suficientes bits para agrupar un número de cuatro bits, se añaden ceros a la izquierda del número.

0101 1011 0110 1011

Ahora reemplace los grupos binarios con sus equivalentes hexadecimales usando la tabla de arriba y el resultado es el siguiente:

5 B 6 B

Fácil de convertir de hex a binario. Reemplace cada dígito de hex con los 4

dígitos binarios correspondientes a la tabla anterior, por ejemplo:

HEX

F C 1 8

BINARY

1111 1100 0001 1000

En este manual verá las palabras "menos significativo " o "menor orden " y " más significativo " o "mayor orden". Éstos se refieren al peso matemático de la parte de un número que se está describiendo. En todos los sistemas numéricos el dígito de la izquierda es el más significativo o el de mayor orden y el dígito a la derecha es el menos significativo o de menor orden. Por ejemplo en el número 00010010 binario, el bit de la izquierda es el bit más significativo (MSB) y el bit de la derecha es el bit menos significativo (LSB). En la palabra 01FF en hex los dos dígitos izquierdos son los byte más significativos y los dos dígitos derechos son los byte menos significativos menor.

PREGUNTAS

1. ¿Qué números representan a un dígito binario?
2. Dé un ejemplo de los nombres no numéricos para los dos números de la pregunta 1.
3. ¿Cuántos bits hay en un byte?
4. ¿Cuántos bits hay en una palabra?
5. Dé el equivalente decimal del número 10010110 binario.
6. Dé el equivalente en hexa del número 10010110 binario.
7. Dé el equivalente en binario del número 115 decimal.
8. Dé el equivalente en hexa del número 307 decimal.
9. Dé el equivalente en binario del número D6 hexa.
10. Explique qué se entiende cuando se dice que el hexadecimal es más eficiente para el usuario que el binario.

SEGUNDA PARTE: OPERACIONES LOGICAS

- Los microprocesadores 8085 soportan 4 operaciones lógicas:
- La operación AND toma dos bits y retorna 1 bit si ambos bits de entrada tienen 1 y 0 si el bit es 0.
- La operación OR toma dos bits y retorna 1 bit si uno o ambos bits son 1 y retorna 0 si ambos bits de entrada son 0.
- La operación de XOR toma dos bits y retorna unos 0 si los bits de entrada son iguales y unos 1 si son diferentes.
- La operación NOT retorna 1 si el bit de entrada es 0 y retorna 0 si el bit de entrada es 1. Esto se llama complementar, invertir, o alternar los bits.

El 8085 realiza estas operaciones de 8 bits a la vez, mediante la realización de la operación lógica en la posición de cada bit, Por ejemplo:

```
      01110010 <-X
AND   10010011 <-Y
      00010010 <-Z
```

Cuando el Bit 0 de X es operado con el bit 0 de Y da como resultado el bit 0 de Z. Cuando el Bit 1 de X es operado con el bit 1 de Y, y da como resultado el bit 1 de Z y el patrón continúa en un máximo de 7 bits. A continuación se muestra la forma lógica de trabajo con las operaciones de bytes:

```
AND   00010010      OR   01100111      XOR   11101101      NOT   11001010
      01000010      10101110      01111001      00110101
      00000010      11101111      10010100
```

Si vemos cada posición es como estar conectado a una luz diferente en una casa, utilizando la lógica de control de operaciones que ponemos a cada luz individualmente o en grupos. El truco es controlar una luz particular sin conocer el estado actual de cada una y no trastornar otras luces en la casa. Por ejemplo, para encender la luz # 2, uno debe estar en una posición dos. Para apagar la luz # 2, o sea un cero debe estar en posición de dos bits. Nos gustaría convertir la luz # 2 y desactivar sin perturbar cualquier de las otras siete luces en la casa. Para el siguiente ejemplo, el byte elegido arbitrariamente a continuación representa el

actual estado de ocho diferentes luces en una casa.

01001010

Esto byte nos dice que las luces # 6, # 3, # 1 están encendidas, y todas las otras luces están apagadas. Para activar la luz # 2 hay que utilizar un valor de máscara. Una máscara es un valor usado para manipular diferentes posiciones. Este valor cuando se utiliza con la adecuada operación lógica realiza la función que deseamos. La máscara de valor elegido para el ejemplo es:

00000100

Este valor de la máscara es operado con la instrucción OR con el estado actual de las luces de la fuerza de la luz # 2

```
      01001010
OR   00000100
      01001110
```

Observe que todas las luces que se siguen en él, y todas las luces que todavía están fuera, con la excepción de la luz # 2.

Si posteriormente se decide a su vez fuera de la luz # 2, usamos un valor de máscara y de una lógica de funcionamiento diferente. Encontrar el valor correcto de máscara para hacer frente a la luz no es tan sencillo como lo fue para convertir la luz encendida. En primer lugar hay que tener el complemento NOT para utilizar la operación de la máscara para encender la luz, entonces con este valor y estado actual de las luces con el fin de apagar la luz. Esto se traduce simplemente en colocar un cero en cualquier posición para que se pague y 1 en todas las demás posiciones.

```
NOT  00000100
      11111011
```

El complementando de la máscara vuelve a encender la luz, y da el nuevo valor de máscara. Este nuevo valor de máscara tiene un cero en una posición para apagarse, y un 1 en todas otras posiciones de pedacito.

PREGUNTAS

1. Complete las operaciones lógicas siguientes:

AND 11100110 10111011 00110110
 00110101 OR 01101001 XOR 10011010 NOT 11101011

2. Defina el termino de mascara.
3. Dar el valor de máscara de 8 bits y la operación lógica para encender a su vez los bits # 3 y # 5 sin perturbar cualquiera de los demás bits..
4. Dar el valor de máscara de 8 bits y la operación lógica para apagar los bits # 1 y # 4 sin molestar a cualquiera de los demás bits.
5. Dar el valor de 8 bits utilizando el funcionamiento de la máscara XOR para complementar los 8 bits del byte siguiente:

 10011100
XOR
 01100011

6. Dar el valor de máscara de 8 bits y la operación lógica para alternar los bits # 7 y # 6 bits sin molestar a cualquiera de los demás bits.
7. ¿Si se opera un valor en sí mismo con la lógica AND cual sería el resultado?
8. ¿Si se opera un valor en sí mismo con la lógica OR cuál sería el resultado?
9. ¿Si se opera un valor en sí mismo con la lógica OR Exclusiva cual sería el resultado?
10. ¿Si se opera un valor en sí mismo con la lógica EX – NOR cuál sería el resultado?
11. Defina el termino alternar.

Anexo 8.2 Guía de Practica N° 2

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 2

Introducción al simulador e instrucciones básicas del 8085.

OBJETIVO.

Aprender a entrar, salir y usar el simulador de 8085, así como a identificar las partes del mismo que muestra en la pantalla.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

a) Iniciar Simulador 8085 para Windows

Inicie el simulador desde el menú Inicio de Windows. Siga estos pasos:

1. Abra el menú Inicio pulsando el botón Inicio.
2. En el menú Inicio, pulse Programas.
3. En el siguiente menú, pulse Simulador 8085.
4. Por último, pulse Simulador 8085.

Si intenta iniciar el simulador de 8085 habiendo una copia del programa ejecutándose en ese momento, se mostrará en pantalla un pequeño mensaje informativo, como el de la siguiente figura 1:

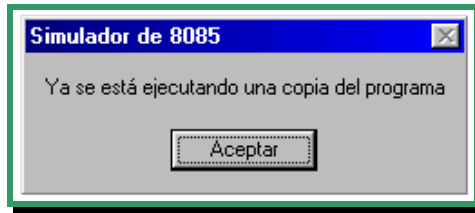


Figura1. Mensaje de advertencia.

b) Conocer y Entender la pantalla de Simulador 8085

Cuando inicie el simulador, verá una pantalla con varios componentes que se pueden modificar. Sin embargo, antes de comenzar necesita conocer las diferentes partes de la pantalla (ver figura 2).

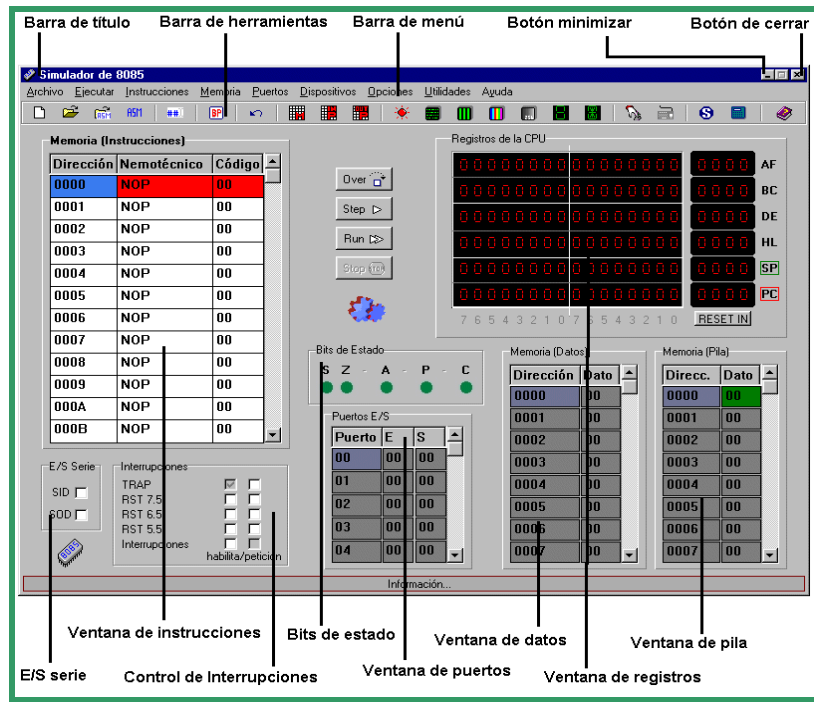


Figura 2. Aspecto inicial del simulador.

c) Usar menús y barras de herramientas

Mientras esté trabajando con Simulador 8085, le dará órdenes para comunicarle las acciones que quiere llevar a cabo. Puede realizar la mayoría de las órdenes de Simulador 8085 usando los menús o la barra de herramientas. El método que escoja dependerá únicamente de sus preferencias personales.

Para seleccionar una orden de menú:

1. Abra un menú pulsando el título de menú sobre la barra del menú. También puede abrir un menú pulsando la tecla Alt al mismo tiempo que mantiene pulsada la letra subrayada del título de menú. Por ejemplo, pulse Alt + A (pulse Alt y mantenga presionada A) para abrir el menú Archivo.
2. Sobre el menú abierto, pulse la orden deseada o pulse la letra subrayada del nombre de la orden.

d) Conociendo las partes del simulador 8085

Registros.

Los registros AF, BC, DE y HL, se presentan por parejas, ya que como el lector sabe, muchas de las instrucciones del 8085 usan estas agrupaciones de registros.

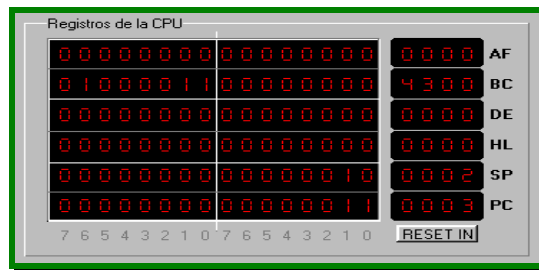


Figura 3. Registros de la CPU.

La mayoría de las instrucciones alteran alguno/s de los registros en su ejecución. Estos cambios que se producen se reflejan en la ventana de la figura 3. Sin embargo, usted también puede cambiar el contenido de los registros en cualquier momento siguiendo estos pasos:

1. Sitúe el ratón sobre el dígito binario o hexadecimal que desee cambiar. Al realizar esta acción, verá que el cursor del ratón cambia para convertirse en una pequeña mano apuntando con el dedo índice.
2. Pulse el botón izquierdo del ratón.

Si lo que cambia es un dígito binario, la pulsación del ratón hará que el bit cambie a cero o a uno, dependiendo de su valor inicial.

Si lo que cambia es un dígito hexadecimal, el hecho de pulsar el ratón hace que el dígito parpadee, tal y como muestra la figura 4. El parpadeo le indica que el dígito está seleccionado y que puede ser cambiado a su nuevo valor. Ahora puede teclear el valor para el dígito.

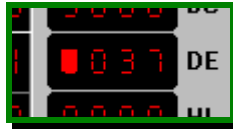

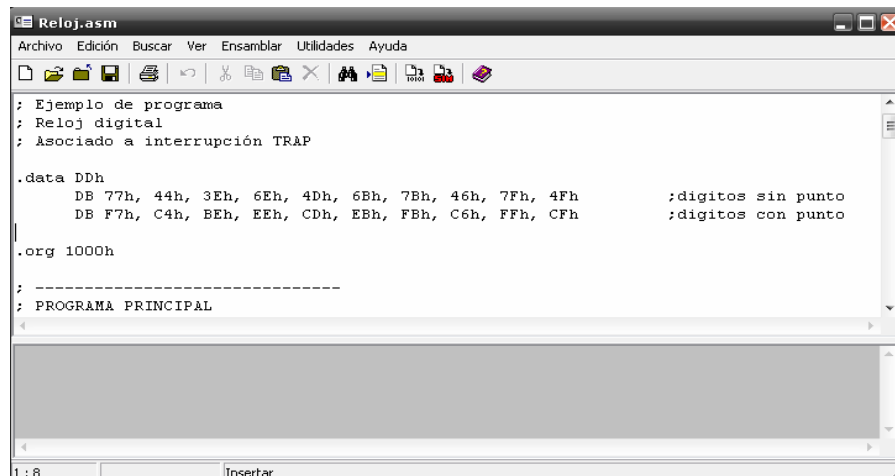


Figura 4. Cambiando un registro

e) Introducir y cargar un programa de ejemplo en el simulador 8085.

1. Para iniciar abra el código fuente desde la barra de menú en la opción Archivo_ Cargar *.asm o simplemente de clic en el icono 
2. En el cuadro de dialogo abrir escoja el archivo de ejemplo llamado Reloj y de clic en abrir.
3. Se abrirá el editor de archivos en lenguaje ensamblador y ahí podrá visualizar todas las partes del programa ejemplo.
4. Ahí encontrara en la parte superior una barra de menús y una barra de herramientas similar a la de la pantalla inicial del simulador.




```
; Ejemplo de programa
; Reloj digital
; Asociado a interrupción TRAP

.data DDh
    DB 77h, 44h, 3Eh, 6Eh, 4Dh, 6Bh, 7Bh, 46h, 7Fh, 4Fh      ;digitos sin punto
    DB F7h, C4h, BEh, EEh, CDh, EBh, FBh, C6h, FFh, CFh      ;digitos con punto

.org 1000h

; -----
; PROGRAMA PRINCIPAL
```

Figura 5. Ejemplo de programa en lenguaje ensamblador

5. Para pasar el código fuente a la pantalla inicial del simulador de clic al botón  , le preguntara si desea limpiar la memoria y de clic en si para hacerlo, se mostrara la pantalla inicial con los registros y las instrucciones cargadas listas para ejecutar.

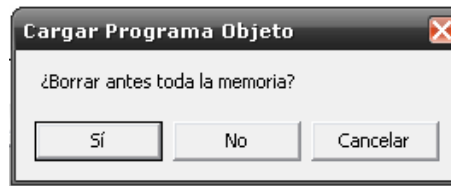


Figura 6. Mensaje de borrado de memoria

6. Ahora en la pantalla principal puede verificar los datos del programa haciendo uso del panel de instrucciones y de los registros que posee el simulador 8085. Para verificar si las instrucciones son las correctas de clic derecho en la parte superior del panel de instrucciones y de clic en la opción ir a dirección 1000h que es la dirección en la que inicia el programa, luego de un enter o clic en aceptar.



Figura 7. Ir a Dirección de memoria

7. Después de verificar todas las instrucciones del programa de clic en el icono del visualizador de display de 7 segmentos para ver la salida de pantalla, y de clic en el botón run para comenzar a correr el programa. Para ver los cambios de los segundos tiene que estar jugando con la interrupción Trap ya que el ejemplo así lo requiere y al ejecutar esa interrupción el reloj avanzará cada segundo.

La salida que obtendrá al ejecutar el programa será la siguiente:

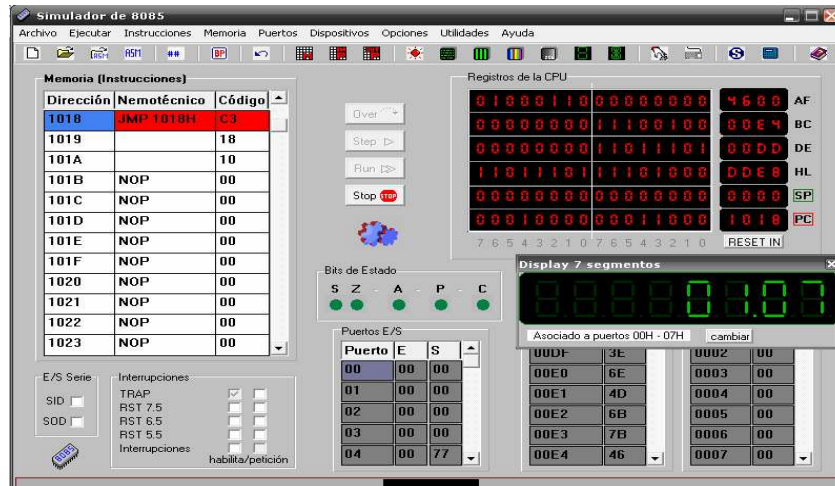


Figura 8. Ejecución del programa de ejemplo.

8. Parar la simulación dando clic en el botón Stop.

e) Salir del simulador

Cuando termine de trabajar con el simulador, hay varias opciones para salir del programa. Todos estos métodos tienen el mismo resultado:

1. Pulse Archivo, Salir.
2. Pulse Alt + F4.
3. Pulse el botón de Cerrar en la barra de título

Anexo 8.3 Guía de Practica N° 3

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 3

Introducción al Equipo PRIMER TRAINER

OBJETIVO.

Conocer e identificar las partes del equipo Primer TRAINER, así como el funcionamiento de cada una de las partes que forman el equipo.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO

PRIMERA PARTE: EL SISTEMA OPERATIVO DE MONITOR (MOS)

EL sistema operativo monitor permite al usuario lo siguiente:

- Ver y cambiar los contenidos de memoria
- Ver y cambiar los contenidos de los registros
- Ver y Cambiar el contenido de pila
- Ejecutar una instrucción a la vez
- escoger un punto de ruptura

El sistema operativo de monitor (MOS) es un poderoso programa que proporciona al usuario las herramientas para entrar y editar programas así como ejecutar, probar, y depurar los programas en lenguaje de máquina. MOS utiliza 6 LEDs para la visualización numérica salida. Este tipo de exhibición es similar a las exhibiciones usadas en las calculadoras, pero permite letras además de los

números.

La pantalla en la izquierda en el encendido de restablecimiento, por defecto contiene la dirección de PC (FF01H) de cuatro dígitos de la pantalla. Esto es lo que se denomina el campo de dirección (ADDRESS FIELD). En los dos dígitos derechos el contenido mostrado es de la dirección. Este campo es conocido como el DATA/OP FIELD. El MOS usa veinte llaves para entrada, con ciertas llaves que tienen una segunda función que puede ser invocada tecleando el botón de "FUNC.". Cada llave produce un tono audible cuando se presione, proporcionando la realimentación para el usuario.

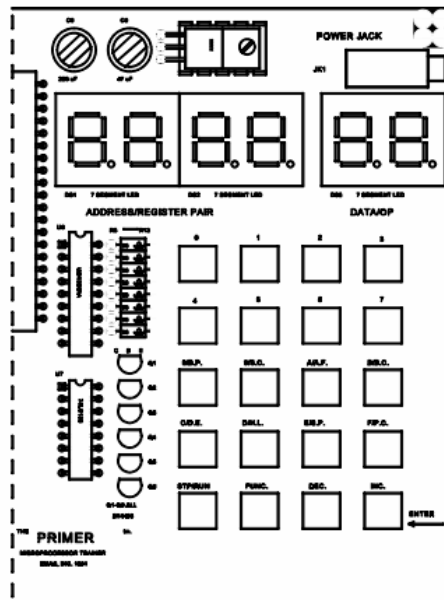


Fig 1. Esquema del PRIMER TRAINER

DESCRIPCIÓN DEL TECLAS

0-F, TECLAS NUMÉRICAS. Cuando una tecla numérica (HEX de 0 f) presionada, el valor numérico aparece en el lado derecho de un campo de datos y mostrar el número que había antes se cambiará la izquierda. Para corregir una entrada bastará con repetir los errores, utilizando el número de clave y el error se sobrescribirá.

Step: Esta clave hace que el microprocesador ejecute una instrucción (paso simple) en el programa actual de la dirección contraria (el valor de la P.C. registro). La ejecución paso a paso es una valiosa herramienta de depuración que

permite al usuario examinar los registros y la memoria después de cada instrucción se ejecuta. La tecla Step ejecuta el programa de instrucción en la dirección contraria y, a continuación, vuelve a la del monitor de operación del Sistema y espera a otro comando. Cuando se ejecuta paso a paso llama a la instrucción CALL está llamando una subrutina en ROM (como una llamada de servicio), el procesador ejecutará la subrutina a toda velocidad y parará en la instrucción inmediatamente después de la instrucción CALL.

Dec.: Esta tecla decrementa el programa y muestra el contador en la izquierda y cuatro displays y los datos en el programa en dos displays de la derecha.

Enter / inc : cuando el monitor está en el modo de entrada de datos, apretando esta tecla almacena los datos que están en los dos displays de la derecha en la dirección de contador de instrucción entonces incrementa el contador de instrucción y muestra la nueva dirección de contador de instrucción en la izquierda en cuatro displays y los datos a esa dirección en los dos displays de la derecha. Esta tecla es también utilizada para almacenar nuevos valores para los registros, el punto de ruptura (breakpoint) y el contenido de la pila (ver sección abajo sobre las segundas teclas de función). Si el registro, o el contenido de la pila de interrupción de la información que se presenta se han cambiado, pero no se desea entrar, simplemente pulse la tecla "Func" dos veces y la información original se mantiene. Esta tecla también se puede utilizar sólo para ver datos en la memoria.

Func.: Esta escoge la segunda función de las teclas que tienen dos funciones. Cuando esta tecla es presionada aparece en pantalla, y si una tecla es presionada y tiene segunda función, dicha función se ejecutará. Esta tecla puede ser presionada dos veces en fila para salir el modo actual y retornar al modo de entrada de datos.

B.P.: Esta llave muestra la dirección actual del punto de ruptura de software. Si 0000 es mostrado, entonces ningún punto de ruptura ha sido establecido El valor mostrado puede ser modificado mediante la introducción de la interrupción deseada en hexadecimal utilizando las teclas numéricas y, a continuación, pulsar la tecla "Intro + Inc.". Cuando el programa alcanza el punto de ruptura la

dirección de punto de ruptura de software es de forma automática restablecida a 0000. NOTA: Obviamente, si la ejecución del programa nunca alcanza el punto de ruptura, en el programa no se alojará la dirección de punto de ruptura. Si la dirección de punto de ruptura apunta a un byte aparte del código de operación en unas dos o tres instrucciones de byte, el programa no se alojará en la dirección de punto de ruptura. También cualquier dirección que hace referencia a las direcciones especificadas en la EPROM (8000 hexadecimal direcciones) no se detendrá la ejecución, incluso si la opción de código en la dirección se está ejecutando.

S.C. Esto muestra los 16 bits que se muestran o en otras palabras, los datos que se va a quitar de la pila con la siguiente instrucción POP, RET o XTHL. Las dos pantallas de la izquierda representan los datos de SP + 1 (puntero de pila dirección + 1) y el próximo par de la derecha representan los datos en SP. El valor mostrado puede ser modificado introduciendo un número hexadecimal deseado utilizando las teclas numéricas y, a continuación, pulsar la tecla "Intro + Inc.".

A.F.: Esta tecla muestra los contenidos de un registro (acumulador) y la condición de banderas. Un registro y las banderas son mostrados como cuatro dígitos hexadecimales en los cuatro displays iniciando en la izquierda. El par de dígitos de la izquierda de las cuatro pantallas muestran el valor de un registro de la pareja y la de la derecha muestra el valor de la condición de Banderas. El valor mostrado puede ser modificado para introducir un número hexadecimal utilizando las teclas numéricas y, a continuación, pulsar la tecla "Intro + Inc.". La condición de Banderas se define poco a poco como sigue:

BIT	CARRY FLAG	(si este bit es 1 indica un salto)
0		
BIT	NOT USED	
1		
BIT	PARITY FLAG	(si este bit es 1 indica un número par de bits, paridad impar)
2		
BIT	NOT USED	
3		

BIT 4	AUX. FLAG	CARRY	(si este bit es 1 indica un salto de 3 bits a 4 bits en un registro)
BIT 5	NOT USED		
BIT 6	ZERO FLAG		(si este bit es 1 indica un resultado cero)
BIT 7	SIGN FLAG		(si este bit es 1 indica que el bit 7 de registro A es 1)

Por ejemplo, si la pantalla muestra: 0044 A.F. Esto indica que el contenido de un registro es el 0 y el cero y PARIDAD Banderas son fijados.

B.C.: Esta tecla muestra los contenidos del par de registro BC. El par de registro BC es mostrado como cuatro dígitos hexadecimales. A partir de la pantalla de la izquierda, el primer par de muestra el contenido del registro B y el segundo par representan el contenido del registro C. El valor mostrado puede ser cambiado introduciendo el número hexadecimal deseado utilizando las teclas numéricas y, a continuación, pulsar la tecla "Intro + Inc."

D.E.: Esta tecla muestra los contenidos del par de registro de DE. El par de registro de DE es mostrado como cuatro dígitos hexadecimales. A partir de la pantalla de la izquierda, el primer par muestra el contenido del registro de la categoría D y el segundo par representan el contenido del registro E. El valor mostrado puede ser cambiado introduciendo el número hexadecimal deseado utilizando las teclas numéricas y, a continuación, pulsar la tecla "Intro + Inc."

H.L.: Esta tecla muestra los contenidos del par de registro de HL. El par de registro de HL es mostrado como cuatro dígitos hexadecimales. A partir de la pantalla de la izquierda, el primer par de muestra el contenido del registro H y el segundo par representa el contenido del registro L. El valor mostrado puede ser cambiado introduciendo el número hexadecimal deseado utilizando las teclas numéricas y, a continuación, pulsar la tecla "Intro + Inc."

S.P.: Esta tecla muestra los contenidos del puntero de pila. El puntero de pila es un registro de 16 bit, mostrados en cuatro dígitos hexadecimales. El valor mostrado puede ser modificado introduciendo el número hexadecimal utilizando

las teclas numéricas y, a continuación, pulsar la tecla "Intro + Inc."

P.C.: Esta tecla muestra los contenidos del contador del programa. El contador del programa es un registro de 16 bits, mostrado como cuatro dígitos hexadecimales. El valor mostrado se puede cambiar introduciendo un número hexadecimal deseado usando las teclas numéricas y luego pulsar la tecla "Intro + Inc."

Run: Esta tecla hace que el microprocesador ejecute un programa a toda velocidad desde la dirección del contador del programa actual (PC). El programa continuará la ejecución de un software opcional hasta que se encuentre una interrupción o hasta que se ejecute la instrucción RST 7 (FF hex).

SEGUNDA PARTE: VER Y CAMBIAR LOS CONTENIDOS DE MEMORIA

Cuando primero se enciende el PRIMER TRAINER, la pantalla marcada "ADDRESS/REGISTER PAIR" mostrará "FF01" y la pantalla marcada "DATA/OP" mostrará un byte aleatorio en hex. El número en la pantalla "ADDRESS/REGISTER PAIR" es el valor del registro del contador del programa (PC) y el número en la pantalla "DATA/OP" son los datos en la dirección de memoria apuntando al contador de programa. Puesto que PC = FF01 "apunta" a los datos a esa dirección de memoria, así si los datos de esa dirección de memoria pasaran a ser el número hex DB entonces "DB" sería mostrado en la pantalla "DATA/OP". Abajo se muestra una ilustración del contador de programa apuntando a los datos en la dirección de memoria FF01. Los valores mostrados en las direcciones de memoria en el diagrama probablemente no son los valores reales que están en la memoria del PRIMER TRAINER. Esto es porque la memoria contiene principalmente valores aleatorios cuando el PRIMER TRAINER está encendido; Los valores fueron escogidos como ejemplo.

DIRECCION	DATOS
FF01	DB <-PC
FF02	12
FF03	D3
:	

: (Direcciones de memoria FF04-FFFF)

Cuando una dirección y los datos a esa dirección son mostrados, el PRIMER TRAINER está en el "modo de entrada de datos". Se debe presionar la tecla "Enter" y el registro de PC será incrementado a FF02 y mostrado en la pantalla "ADDRESS/REGISTER PAIR " y los datos que están apuntando al nuevo valor de PC serán mostrados en la pantalla "DATA/OP". Si los datos en dirección de memoria FF02 se encuentre que sea 12 hex, entonces "12" será mostrado en la pantalla "DATA/OP". El contador de programa apuntando a los datos en la dirección de memoria FF02 es ilustrado abajo.

DIRECCIÓN DATOS

FF01	DB
FF02	12 < -PC
FF03	D3

:

: (Direcciones de memoria FF04-FFFF)

Presionar "Enter" tantas veces como se desee. Apretar la tecla "Dec." y el valor mostrado para PC se decrementará y el número en la pantalla "DATA/OP" es el byte de datos apuntado a ser la nueva dirección de memoria de PC. Apretar la tecla " Dec." hasta que PC sea FF01 de nuevo o presionar el botón de reset.

Digite "F" y "C", y se verá numero hex " FC " en la pantalla "DATA/OP". Si se presiona mal un dígito, o se quiere cambiar el valor que se ha digitado, solamente se debe presionar los dos dígitos hex y estos serán escritos sobre los otros. Ahora digitar "0" y "7" y luego presionar "Enter" y PC será incrementado y el valor de PC, FF02 será mostrado conjuntamente con los datos apuntados como el nuevo valor de PC. Digitar el número 55 y apretar la tecla "Dec.". Se podrá observar que los datos en la dirección FF01 ahora es 07. Apretar la tecla "Enter" y se verá que los datos en la dirección FF02 no es 55 tal cual digitado antes. Esto es porque los números hex que son digitados no son almacenados hasta que la tecla "Enter" es presionada. Esta es una característica útil cuando se ha digitado un número y se decide que no se quiere guardar en memoria o se desea no cambiar el valor original.

TERCERA PARTE: CARGAR UN PROGRAMA EN LA MEMORIA

En las lecciones que siguen, es necesario cargar programas en la memoria. El lenguaje de máquina para los programas es listado en el formato siguiente:

DIRECCIÓN	DATOS	INSTRUCCIÓN
FF01	DB	IN 12
FF02	12	
FF03	D3	OUT 11
FF04	11	
FF05	C3	JMP FF01
FF06	01	
FF07	FF	

Antes de entrar un programa en memoria, presionar el botón de reset. Esto restablece los registros de propósito general y el registro de bandera a cero y pone el indicador de pila a FFD4 y el contador de programa a FF01. Ver la tabla de datos del programa en la parte superior. Para que un programa de lenguaje de máquina pueda ser cargado en memoria correctamente, las direcciones bajo la columna marcadas como "DIRECCIÓN" deben contener los datos a la derecha de ellos en la columna marcada como "DATOS " después de haber completado la carga de datos. La columna llamada "INSTRUCCIÓN" sólo dice lo que la instrucción de los datos representa, así esto se puede ignorar.

Puesto que el valor de PC es FF01, digitar DB, que son los datos de la tabla que pertenecen en esa dirección, y presionar "Enter". PC es ahora FF02 así digitar 12 y presionar Enter. Continuar introduciendo los datos que pertenecen a la dirección de PC y presionar Enter hasta que todas las direcciones listadas en la tabla hayan sido cargadas. Ahora apretar la tecla "DEC" y verificar que los datos a la dirección actual de PC sean iguales que los datos en la tabla de datos de programa.

Si los datos no son los mismos, sólo se debe digitar el número correcto y presionar Enter y luego continuar presionando la tecla "DEC" y verificar los datos hasta el contador de programa sea FF01 de nuevo. Una vez que se sienta cómodo introduciendo los programas en memoria se puede saltar el proceso de

verificación de datos.

No se deberá alarmarse si se digita un programa incorrectamente y no trabaja, porque es imposible para un programa el dañar al PRIMER TRAINER. Lo peor que puede suceder es que se tenga que apretar el botón de reset, o si el programa está corrompido se tiene que entrar el programa en memoria de nuevo.

CUARTA PARTE: VER Y CAMBIAR LOS CONTENIDOS DE REGISTRO

Es una buena ayuda de aprendizaje ser capaz de examinar los valores de los registros del 8085. Para ver los contenidos de un registro se debe apretar primero la tecla "FUNC" (que hará que la pantalla cambie para mostrar "Func") y luego la tecla correspondiente para el registro que se quiere examinar. Cuando la tecla "FUNC" es presionada invocara la función alterna de la tecla,. Las teclas que tienen funciones alternas tienen la función estándar seguida por una pleca "/" y la función alterna. Por ejemplo la función estándar de la tecla " 8/B.P." es el dígito 8 y después haber apretado la tecla "FUNC" la función alterna "B.P" será invocada. Cada línea en la tabla de abajo muestran la tecla que debe ser apretada después de haber apretado la tecla "FUNC" y los datos se mostraran en "PANTALLA ADDRESS/REGISTER PAIR" como consecuencia. A continuación se muestra la pantalla address/register pair:

TECLA	DATOS MOSTRADOS EN DS1 (PANTALLA IZQUIERDA)	DATOS MOSTRADOS EN DS1 (PANTALLA IZQUIERDA)
A.F.	Registro A	Registro de Bandera
B.C.	Registro B	Registro C
D.E.	Registro D	Registro E
H.L.	Registro H	Registro L
S.P.	Byte de orden superior de SP	Byte de orden inferior de SP
P.C.	Byte de orden superior de PC	Byte de orden inferior de PC

Al apretar una de las teclas anteriores después que la tecla "FUNC" es presionada causa que el valor en los registros se muestre en la pantalla marcada como

"ADDRESS/REGISTER PAIR" y el nombre de registro será mostrado en la pantalla marcada como "DATA/OP". Después de esto, apretando cualquier tecla distinta a las teclas numéricas (0 - F) causará que el PRIMER TRAINER retorne al modo de entrada de datos.

Si se quiere cambiar el valor de un registro, apretar la tecla "FUNC" y luego la tecla " A.F." y como consecuencia la pantalla mostrara "0000 A.F.". Para cambiar el valor del registro A a 1F hex sin cambiar el registro de bandera, apretar las teclas siguientes en orden: "1", "F", "0", "0", "Enter". Después de haber apretado la tecla "Enter" la pantalla no mostrará el registro A y el registro de bandera, en vez retornará al modo de entrada de datos. Ver PSW (el registro A y el registro de bandera) de nuevo apretando la tecla "FUNC", luego la tecla "A.F.". Se verá que los valores que han sido entrados antes aun están en los registros. Observar que a fin de no cambiar el valor del registro de bandera, es necesario introducir su valor de nuevo (00). Si el registro de bandera hubiera sido 40 y se quiere cambiar el registro A a 2B sin cambiar el registro de bandera deberá apretar las teclas siguientes (no hacer esto, solamente es un ejemplo): "2 ", "B", " 4 ", " 0 ", " enter".

Ahora cambiar el valor del registro de bandera y se asumirá que no importa cuál es el valor de registro A. Para hacer al registro de bandera 41 hex, teclear "4", "1". Notar que después de apretar estas dos teclas, los dos ceros han sido cambiados en la pantalla del registro A. Ahora apretar "enter", ahora ver el registro A y el registro de bandera de nuevo y verá el nuevo valor (0041 hex) que es sido introducido.

Como fue descrito anteriormente, el registro de bandera es un registro 8 bits que tienen bits individuales llamados banderas, que indican el resultado de las operaciones aritméticas o lógicas. Para ver lo que son los valores de los bits individuales, se debe convertir el valor hexadecimal del registro de bandera a binario. Si no se recuerda cómo hacer esto, refiérase de nuevo a la sección de Matemáticas de computadora en introducción a las computadoras. Por ejemplo, para descubrir cuáles son los valores de la bandera actual, convertir 41 hex a binario (ver el diagrama de abajo).

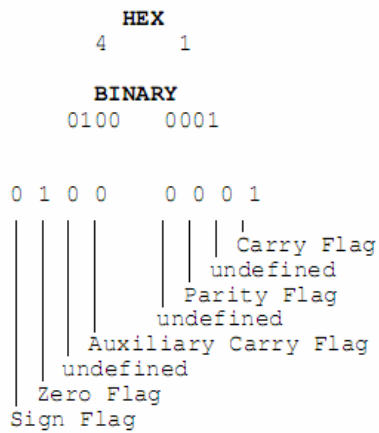


Fig 2. Estado de las banderas

Según el diagrama anterior, el acarreo y las banderas cero tienen un valor de 1 y la señal, el acarreo auxiliar, y las banderas de paridad tienen un valor de 0. Si el registro de bandera es D5, todas las banderas definidas son establecidas. Puede necesitar referirse de nuevo a este diagrama en las lecciones posteriores cuando se le pida para examinar el registro de bandera. Un acceso directo para determinar el estatus de la bandera de acarreo es examinar el dígito de más a la derecha del registro de bandera. Si el dígito es cualquiera (0, 2, 4, 6, 8, A, C, E) entonces la bandera de acarreo es 0, de otra manera, si el dígito es impar (1, 3, 5, 7, 9, B, D, F) la bandera de acarreo es 1.

CIERTAS SUGERENCIAS ACERCA DE CAMBIAR REGISTROS

Se puede apuntar al registro PC (contador de programa) a cualquier ubicación cambiando los contenidos del registro PC. Esto elimina la necesidad de apretar las teclas "enter" o "DEC" muchas veces a fin de ver o cambiar datos de las direcciones de memoria que están distantes de la dirección actual.

Si está digitando un nuevo valor para un registro y no ha apretado "enter" puesto que ha comenzado a hacer esto, se puede apretar las teclas "Step", "func." O "dec." para retornar al modo de entrada de datos y el registro no se cambiara.

Anexo 8.4 Guía de Practica N° 4

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 4

Utilización de los programas de software del PRIMER TRAINER

OBJETIVO.

Aprender a utilizar los programas de software para poder interactuar con el Primer TRAINER, así como también instrucciones para correr programas en el equipo y cambiar registros.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO

PRIMERA PARTE: CORRER EL PRIMER PROGRAMA EN EL PRIMER TRAINER

INSTRUCCIONES

IN <byte> op code = DB

Carga el registro A con los datos que están en el puerto de entrada especificado por el byte siguiendo la instrucción. Ningunas banderas son afectadas.

OUT <byte> op code = D3

Envía los datos del registro A al puerto de salida especificado por el byte siguiendo la instrucción. Ningunas banderas son afectadas.

JMP <addr> op code = C3

Carga el contador de programa (PC) con la dirección contenida en los dos bytes siguiendo la instrucción. El primer byte siguiente al código de operación es el byte menos significativo de la dirección, el segundo byte es el byte más significativo de la dirección. Ningunas banderas son afectadas.

Abajo hay un ejemplo de programa 8085 escrito en lenguaje ensamblador. Este programa lee el valor binario de los interruptores DIP y envía el valor de los indicadores LED de salida digital y luego repite estas instrucciones indefinidamente. El programa se puede parar al apretar el botón de reset.

Programa Ensamblador

```
    leds    equ    11h
    dips    equ    12h

    .org    0ff01h
Loop: in     dips      ; Carga el registro A con
                   ; los valores de los interruptores DIP
    out     leds      ; Salida del registro A a los LEDs
    jmp     loop      ; salta a loop
end
```

Programa de lenguaje de maquina

Dirección	Datos	Instrucción
FF01	DB	IN 12
FF02	12	
FF03	D3	OUT 11
FF04	11	
FF05	C3	JMP FF01
FF06	01	
FF07	FF	

Para correr el programa de lenguaje de máquina, encender el PRIMER TRAINER y la dirección del contador de programa FF01 se mostrará en los 4 displays de la izquierda.

Cargar los datos de lenguaje de máquina de la tabla anterior a la memoria. Una vez todo el programa ha sido introducido correctamente se debe regresar a la dirección del contador de programa FF01 apretando el botón reset o cargando el registro PC con FF01, luego correr el programa apretando la tecla " Func." y la tecla "Step/Run". Se deberá de ver que al mover los interruptores DIP las salidas

digitales LED se apagaran o encenderán en consecuencia. Ahora apretar el botón de reset y el programa hará un alto, las salidas digitales LED se apagaran y la pantalla mostrara " FF01 DB".

El PRIMER permite ejecutar una única instrucción de lenguaje de maquina comenzando en la memoria en el registro PC. Este procedimiento es llamado saltos individuales. Luego que la instrucción ha sido completada se, el registro PC apunta a la dirección de la próxima instrucción y el PRIMER será retornado al modo de entrada de datos. Para un solo paso en la primera instrucción, apretar la tecla "stp/run" y la pantalla mostrara " FF03 d3 " que es el nuevo valor para PC y los datos apuntados a PC. La instrucción IN fue ejecutada y el registro PC fue retornado con la dirección de la próxima instrucción, la instrucción OUT. Paso a paso el resto del programa, hace lo siguiente:

1) Paso a paso, de nuevo y la pantalla mostrara "FF05 C3". La instrucción OUT 11 ha sido ejecutada y el registro PC ha sido retornado con la dirección de la instrucción JMP FF01.

2) Paso a paso, de nuevo y la pantalla mostrara "FF01 db". Esta pantalla mostrara FF01 porque ese es el valor que fue cargado en el registro PC por la instrucción JMP FF01. Se puede ver que el mnemónico vino de la capacidad de las instrucciones para causar que el contador de programa a "salte" a otra ubicación. PC ahora apunta a la instrucción IN 12 de nuevo.

3) Paso a paso, de nuevo y la pantalla mostrara "FF03 d3". La instrucción IN ha sido ejecutada y el registro PC fue retornado con la dirección de la instrucción OUT.

Justo después del paso tres, el valor del interruptor DIP ha sido cargado en el registro A y puede ser vista apretando "Func." luego la tecla "A/A.F.". El valor del registro A se muestra en el par de dígitos a la izquierda. Cambiar el valor del registro A (en este programa los valores de bandera no importan). Ahora al hacer el paso uno se verá la salida digital LED que ahora muestra el nuevo valor del registro A. Si se aprieta el la tecla "step" tres veces más, la instrucción IN cambiara al registro A al valor del interruptor DIP de nuevo y la instrucción OUT

mostrará el nuevo valor del registro A en la salida digital LED.

* **NOTA:** El PRIMER fue diseñado de tal manera que las salidas digitales LED enciendan cuando un 0 lógico es la salida a un bit en particular del puerto A. Así, por ejemplo, si 01h es la salida al puerto, todos los LEDs serán encendidos exceptuando el LED 0, y si F0h es la salida al puerto, todos los LEDs serán apagados excepto del LED 3 al 0.

SEGUNDA PARTE: CARGANDO LOS REGISTROS Y TRANSFIRIENDO DATOS ENTRE REGISTROS.

INSTRUCCIONES

MVI A, <byte> op code = 3E

MVI B, <byte> op code = 06

MVI C, <byte> op code = 0E

MVI D, <byte> op code = 16

MVI E, <byte> op code = 1E

MVI H, <byte> op code = 26

MVI L, <byte> op code = 2E

Cargar el registro que sigue al mnemónico MVI con el valor del byte siguiendo el código de operación (op code). Ningunas banderas son afectadas.

XCHG op code = EB

Cambia el valor del registro D con el del H y el valor del registro E con el del L. Ninguna bandera se afecta.

MOV A, B op code = 78

Copie el registro B al registro A. Ninguna bandera se afecta.

MOV B, C op code = 41

Copia el registro C al registro B. Ninguna bandera se afecta.

RST 7 op code = FF

La dirección que sigue a la instrucción RST 7 es empujada a la pila y luego las instrucciones que empiezan en la dirección 0038 son ejecutadas. Ninguna bandera se afecta.

El programa de la parte posterior demuestra las instrucciones usadas para cargar registros y las instrucciones usadas para mover valores entre los registros.

PROGRAMA DE ENSAMBLADOR

```
.org 0ff01h
mvi a, 1 ; carga a A con 1
mvi b, 2 ; carga a B con 2
mvi c, 3 ; carga a C con 3
mvi d, 4 ; carga a D con 4
mvi e, 5 ; carga a E con 5
mvi h, 6 ; carga a H con 6
mvi L, 7 ; carga a L con 7
xchg    ; cambia el valor de DE con HL
xchg    ; lo hace de nuevo
mov a, b ; copia b al registro A
mov b, c ; copia c al registro B
rst 7   ; retorno a MOS
```

PROGRAMA DE LENGUAJE DE MAQUINA

DIRECCION	DATOS	INSTRUCCIÓN
FF01	3E	MVI A,1
FF02	01	
FF03	06	MVI B, 2
FF04	02	
FF05	0E	MVI C, 3
FF06	03	
FF07	16	MVI D, 4
FF08	04	
FF09	1E	MVI E, 5
FF0A	05	
FF0B	26	MVI H, 6
FF0C	06	
FF0D	2E	MVI L, 7
FF0E	07	
FF0F	EB	XCHG

FF10	EB	XCHG
FF11	78	MOV A, B
FF12	41	MOV B, C
FF13	FF	RST 7

Agregar el programa en memoria y luego apretar el botón de reset que cambiará el contador de programa a FF01. Si se examinan los registros A, B, C, D, E, H y L se verá que apretando el botón de reset los limpia a 0.

Para examinar el programa se usara un punto de ruptura (breakpoint). Un punto de ruptura es una dirección en memoria donde se desea que el programa se detenga (o pare) y retorna al modo de entrada de datos. Esto permite correr o ejecutar una parte del programa a máxima velocidad y luego parar antes de ejecutar la instrucción a la cual el punto de ruptura apunta y así se puede examinar los registros o correr paso a paso u otras cosas. La interrupción de software es ejecutada por el programa MOS (el sistema operativo monitor) usando la interrupción de software RST 7. Cuando el usuario especifica una dirección de interrupción el MOS reemplaza el código de operación de esta dirección con la instrucción RST 7 (FF hex). El programa del usuario puede ejecutar entonces a toda velocidad hasta encontrar la instrucción RST 7. Cuando el programa de usuario alcance la interrupción, la instrucción de RST 7 retorna el control al programa MOS. El programa MOS entonces restaura el código de operación que fue remplazado por la instrucción RST 7. El programa de MOS sólo permite el uso de un punto de ruptura a la vez, y esto es automáticamente restablecido a 0000 después que el punto de ruptura ha sido encontrado. El usuario puede sin embargo insertar puntos de ruptura agregando instrucciones RST 7 en ubicaciones estratégicas a lo largo del programa de usuario.

Recordar que cualquier punto de ruptura insertado a mano deberá también ser quitado a mano cuando sea necesario. Al insertar un punto de ruptura de software es importante recordar que si la ejecución de programa nunca alcanza la dirección de punto de ruptura no se detendrá. Todos los puntos de ruptura deben apuntar a los códigos de operación antes que estos trabajen. Si la dirección de punto de ruptura está apuntando a un byte distinto al de código de operación en una, dos o tres instrucciones de byte, el programa no se detendrá en la dirección de punto de

ruptura. No solo no se detendrá el programa, si no que el programa no trabajara correctamente porque uno de los bytes en los dos o tres bytes de instrucción de byte habrán sido cambiados a de FF hex. Todas las direcciones de punto de ruptura deberán ser mayores a 8000 hex. Esos 8000 hex no detendrán la ejecución, aún si se ejecutara el código de operación en esa dirección.

Para escoger un punto de ruptura, apretar la tecla "FUNC" seguido por la tecla "8/B.P." y la pantalla mostrara la dirección de punto de ruptura en la pantalla " ADDRESS/REGISTER PAIR " y " B.P." en la pantalla "DATA/OP". Un punto de ruptura es escogido de la misma manera que se cambia un registro y todos los reglamentos con respecto a cambiar los registros aplican para establecer un punto de ruptura; sólo digitar la dirección y apretar "enter". Recordar que B.P. (punto de ruptura) no es un registro dentro del microprocesador 8085, es sólo una función soportada por el MOS. Seguir estos pasos:

PC ACTUAL

FF01 establecer un punto de ruptura en la dirección FF0F y ejecutar el programa. Después de que ocurra el punto de ruptura el PRIMER se establecerá en modo de entrada de datos mostrando la dirección FF0F. Examinar los registros y se verá que A, B, C, D, E, H y L han sido cargados con 1-7 respectivamente. Recordar que cuando se examina un registro, el registro de bandera es mostrado también, así la pantalla mostrara " 0100 A.F.".

FF0F Ejecutar paso a paso, y la primera instrucción XCHG se ejecutará. Examinar el par de registros DE y HL y se verá que el valor previo de HL está ahora en DE y el valor previo de DE está ahora en HL.

FF10 Ejecutar paso a paso, y la segunda instrucción XCHG será ejecutada la cual cambiara DE con HL de nuevo. Examinar los pares de registros DE y HL y se verá que sus valores son iguales que antes de la primera instrucción XCHG.

FF11 Ejecutar paso a paso, y la instrucción MOV A, B, será ejecutada que copia el registro de B al registro A. Examinar estos registros para verificar esto.

FF12 Finalmente, ejecutar paso a paso la instrucción MOV B, C y el registro C

será copiado al registro B. Examinar el par de registros BC.

FF13 Fin del programa.

La última instrucción en el programa de lenguaje de máquina es FF que es la instrucción RST 7. Leer la definición de la instrucción al comienzo de esta lección, de nuevo (la pila será explicada más tarde). El ROM del PRIMER tiene instrucciones en la dirección 0038 que retornan el control del microprocesador al MOS. Tratar de ejecutar paso a paso la instrucción RST 7; como se puede ver nada sucede. Ahora apretar el botón de reset, luego ejecutar el programa apretando la tecla "STP/RUN" (sin apretar "FUNC") y se verá que el programa se detiene en la dirección de la instrucción RST 7 y retorna al modo de entrada de datos mostrando " FF13 FF " en la pantalla.

TERCERA PARTE: CONVERTIR UN ARCHIVO .ASM A .HEX A TRAVÉS DEL PROGRAMA ENSAMBLADOR MA85.

Para poder realizar la conversión a archivo objeto (lenguaje de máquina) se deberán realizar los siguientes pasos en un sistema Windows:

Inicio > Ejecutar



Figura 1. Menú Inicio

Luego se debe digitar cmd. A continuación aparecerá la línea de comandos.

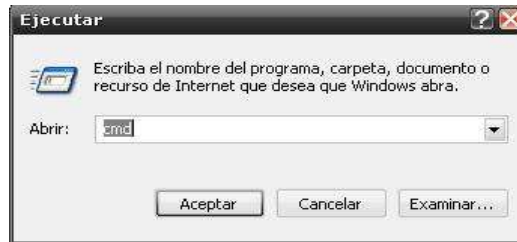


Figura 2. Inicio De Consola De Comandos

Para poder ejecutar el programa ensamblador (asm85.exe) se debe acceder a la ruta donde se encuentra el ejecutable. Para poder moverse entre las carpetas en la línea de comando se debe utilizar el comando CD. Para bajar un nivel.

Ya estando dentro de la carpeta se deberá seguir la siguiente sintaxis:

```
C:\>ma85.exe [archivo-a-convertir.asm] [archivo-convertido.hex]
```

Ejemplo:

```
C:\>ma85.exe [EJEMPLO1.ASM] [EJEMPLO1.HEX]
```

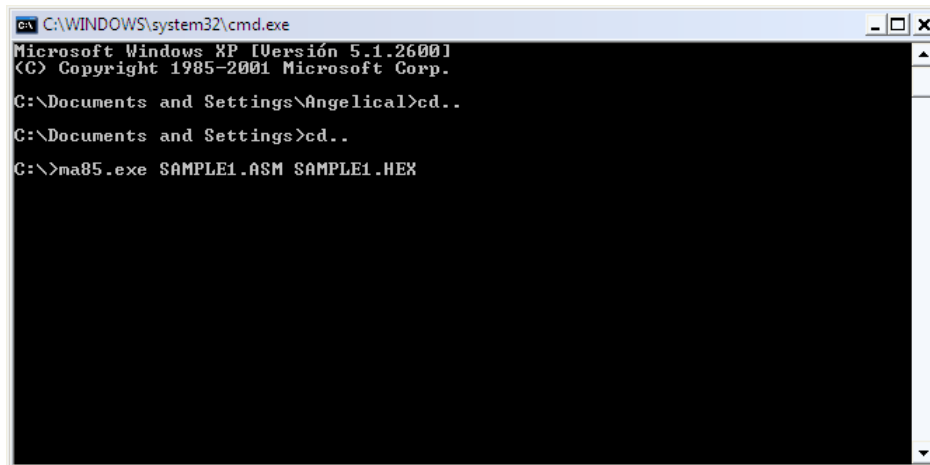


Figura 3. Ejemplo De Creación De Un Archivo Tipo Hex

El o los archivos convertidos se pueden encontrar en la misma carpeta en la cual se encuentra en la línea de comandos

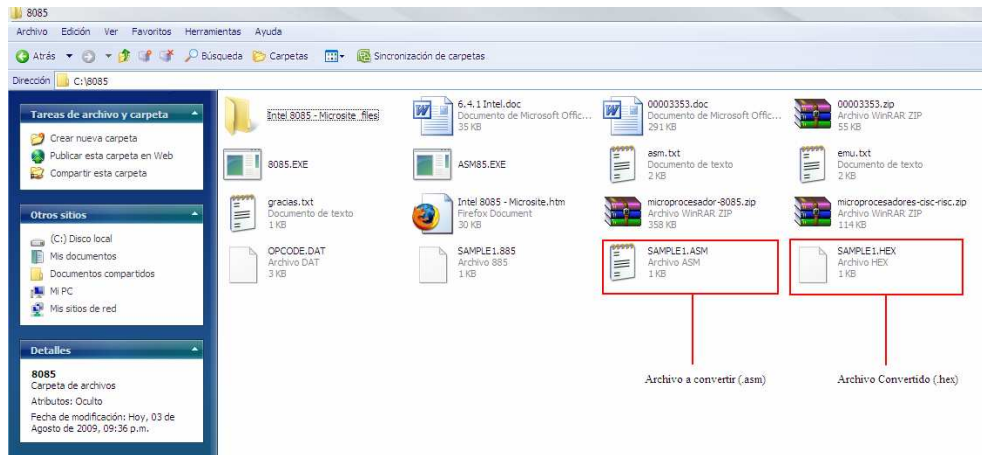


Figura 4. Visualización De Archivos Creados

CUARTA PARTE: USANDO EL HYPERTERMINAL PARA WINDOWS.

EL HYPERTERMINAL es una aplicación dentro de Windows que le permite al equipo entrenador comunicarse con dispositivos en serie. Cuando un carácter es recibido por el puerto serial del PC que se está siendo usado por el HYPERTERMINAL, el carácter será mostrado en la ventana de la aplicación y cuando una tecla sea presionada, el carácter correspondiente será enviado vía el puerto serie.

Copiar el archivo de configuración del HYPERTERMINAL del disco de distribución ("PRIMER_95-98.HT" para Win95/98 o "PRIMER_XP-2K-NT.HT" para WIN XP, 2000 o NT) al escritorio de Windows luego hacer doble clic sobre en dicho icono. En el HYPTERTERMINAL deberá determinar si se quiere instalar un modem, se debe hacer clic en la opción de "ninguno".

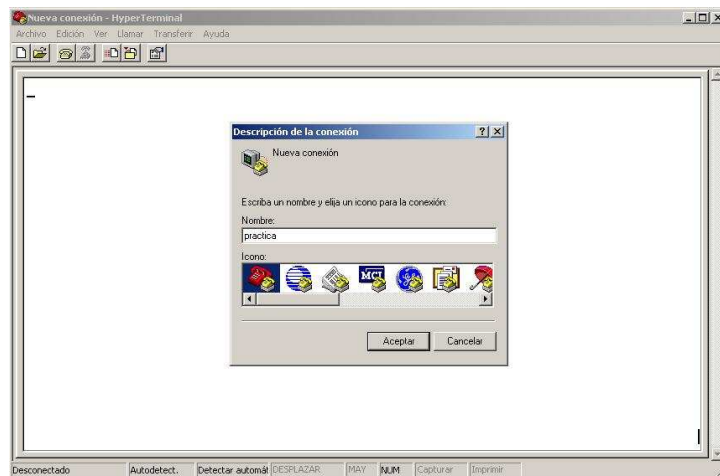


Figura 5. Inicio De Sesión Del Hyperterminal

Se debe preparar el PRIMER TRAINER para 9600 Baudios con las colocaciones siguientes de los jumpers:

JP1 = 9600

OJ1 = jumper 1-2 y 3-4

OJ2 = jumper b

OJ3 = jumper b

Los archivos de configuración usaran COM1 como el puerto de serie. Si COM1 se encuentra disponible en el PC y no esté siendo utilizado para otro dispositivo tal como un ratón, salte el párrafo próximo.

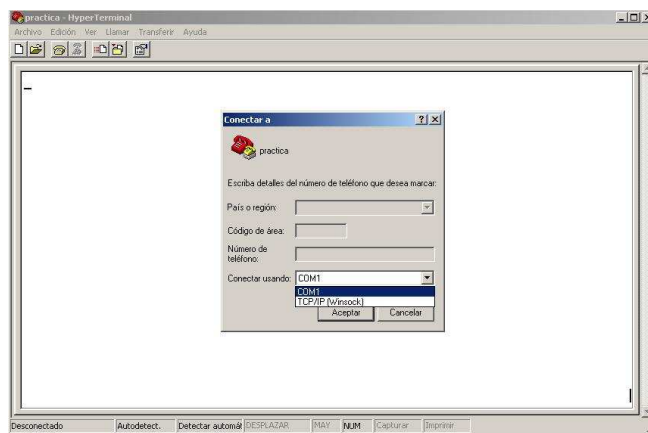


Figura 6. Selección Del Puesto A Utilizar

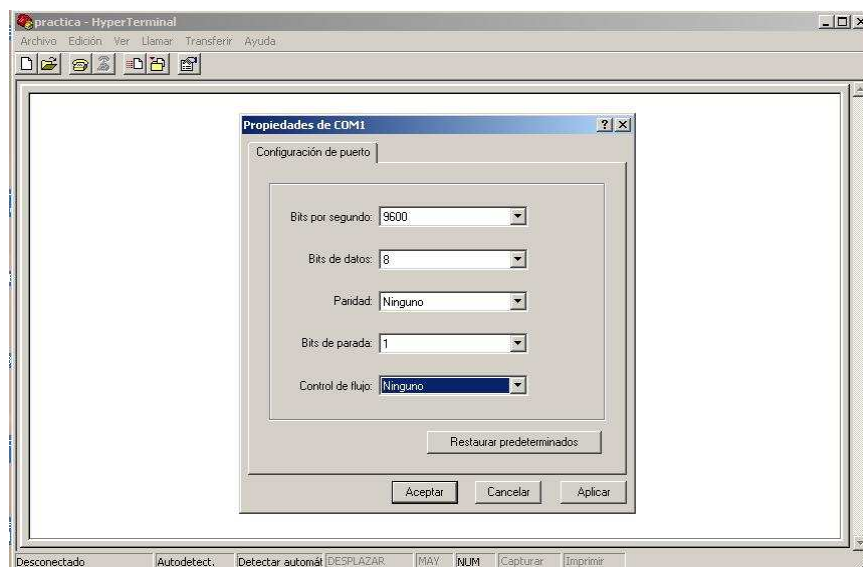


Figura 7. Configuración De Datos De Inicio De Sesión

Si el puerto COM1 está siendo ya usando en el PC, se necesitara apretar el botón de desconectar (En la parte superior izquierda de la ventana terminal, mostrado como un teléfono con el auricular levantado sobre el teléfono).

Escoja las propiedades de archivo, luego en la ventana que aparece, se puede encontrar la pestaña que se llama Conectar a. Si la pestaña se encuentra en primer plano, se observara una ventana de selección llamada Conectar Usando. Usar esta opción para escoger un puerto de COM que es no esté siendo usado en el PC y apretar el botón OK. Luego apretar el botón de llamada (junto al botón de desconectar) para establecer la conexión. Escoger Guardar Archivo para salvar la configuración actual.

Utilizando el cable serial (que es incluido en el equipo tanto en su presentación deluxe como en su presentación estándar), conectar el cable serial del entrenador Primer al puerto de COM para el cual ha sido configurado el HYPERTERMINAL.

Restablecer el PRIMER y apretar las teclas "Func." y "0" para invocar EMOS. Se debería de mostrar el siguiente menú en el HYPERTERMINAL:

E-FORTH V1.10 SBC+
Copyright 1987-91 EMAC inc.
License no. 00.

EMOS V2.00 HELP MENU

B --> Bring Block from RAMDISK to memory
C --> Change register contents
D --> Dump memory contents
E --> Edit memory contents
F --> Fill memory with byte
G --> Go execute program { full speed }
H --> Hex/dec math { 1st + 2nd, 1st - 2nd }
I --> Input from I/O port
L --> List memory contents using mnemonics
M --> Move section of memory
O --> Output to I/O port
R --> display Register contents
S --> MOS Service call
T --> Trace program execution
W --> Write memory to RAMDISK block
Z --> Zap application EPROM
< --> hex download from TRAINER to host
> --> hex upload to TRAINER from host
? --> display this help menú

En español las opciones son las siguientes:

E-FORTH V1.10 SBC+
Copyright 1987-91 EMAC inc.
License no. 00.

EMOS V2.00 HELP MENU

B --> Trae el bloque de RAMDISK a la memoria
C --> Cambiar contenidos de registro
D --> Eliminar contenidos de memoria
E --> Editar los contenidos de memoria
F --> Llenar la memoria con el Byte
G --> Correr Programa Ejecutable {Velocidad Máxima}
H --> Matemáticas en Hex/dec {1ro + 2do, 1ro - 2do}
I --> Ingresar desde el Puerto de Entrada/Salida
L --> Listar los contenidos de memoria usando mnemotecnia
M --> Mover Sección de Memoria
O --> Salida al Puerto de Entrada/Salida
R --> Mostrar contenidos de Memoria
S --> Servicio de llamada MOS
T --> Rastreo de programa ejecutable
W --> Escribir memoria al bloque RAMDISK
Z --> Aplicación Zap EPROM
< --> Descarga de hex desde entrenador al host
> --> Cargar hex al entrenador desde el host
? --> Desplegar el menú de ayuda

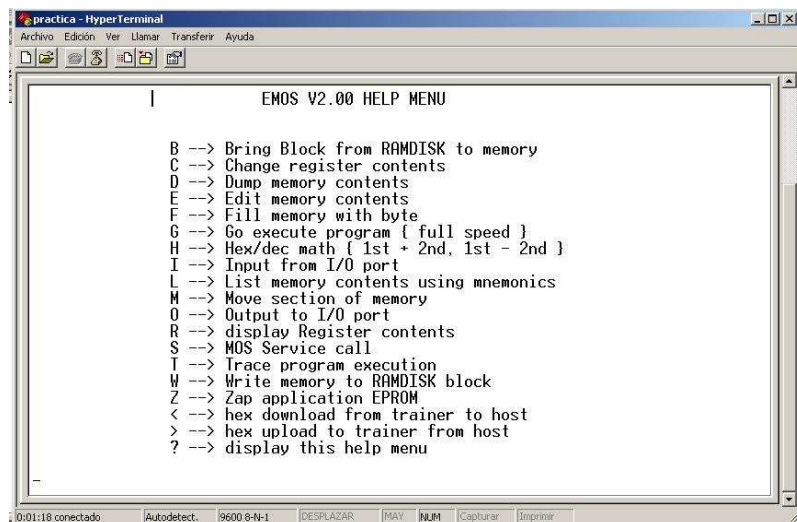


Figura 8. Menú Del Hyperterminal.

Ahora deberá digitar ">" para cargar en el entrenador y usted desplegara el siguiente mensaje:

- > STARTING ADDRESS..

Digitar " FF01 " para este ejemplo y se desplegara lo siguiente:

- > STARTING ADDRESS.. FF01

READY TO RECEIVE, <ESC> TO ABORT

Ahora se podrá empezar a descargar el archivo hex INTEL; seleccionando desde la terminal en el HYPERTERMINAL la opción de "Transferir" y seleccionar "Enviar un Archivo de Texto. Esto mostrara una ventana para la selección del archivo, se debe cambiar el Tipo de archivo a la opción para mostrar todos los archivos (*.*) y escoger el archivo TRAJNTST.HEX que está en el mismo directorio que el archivo de configuración del HYPERTERMINAL.

La transferencia sucederá sin ninguna realimentación hasta que termine. Este archivo tomará cerca de 2 segundos para cargar entonces el "-" prompt reaparecerá.

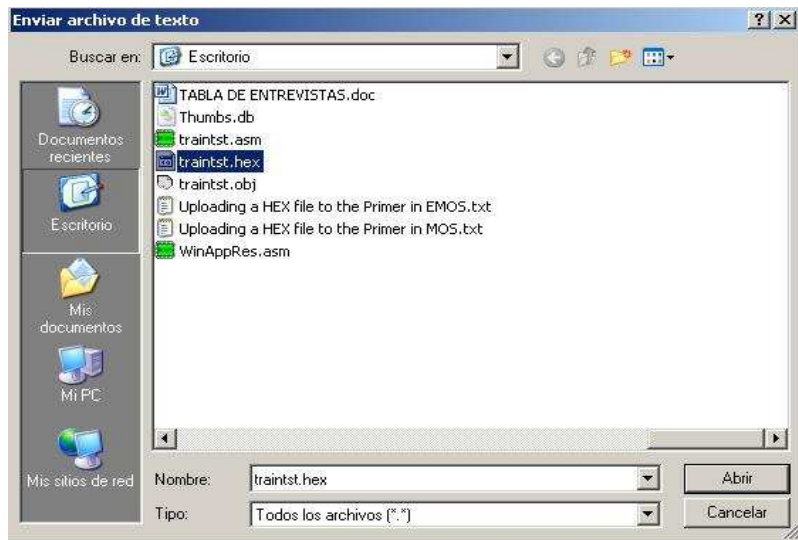


Figura 9. Selección De Archivo A Cargar En El Hyperterminal.

Digitar "G" de "GO" para ejecutar el programa y aparecerá lo siguiente:

- G STARTING ADDRESS..

Escribir "FF01" para la dirección de inicio y cuando el cursor vuelva a cargar para un punto de ruptura (Breakpoint), sólo se debe presionar Enter (para ningún punto de ruptura) y se mostrara lo siguiente antes de que el programa inicie:

```
-----  
- G STARTING ADDRESS.. FF01  BREAKPOINT ADDRESS..  
  
RUNNING.. HIT ANY KEY TO STOP  
-----
```

Algo similar a lo siguiente será mostrado en la terminal:

```
-----  
THE DIPSWITCH NOW READS  
  
0087  
-----
```

El número mostrado será distinto basado en las posiciones en las que se encuentran los interruptores de DIP.

El programa existe cuando los interruptores DIP han sido movidos de tal manera que se leen "0000". Luego de los datos de salida, el prompt retornara y la terminal mostrará algo similar a lo siguiente.

```
-----  
THE DIPSWITCH NOW READS  
  
0000  
SZxAxPxC A B C D E H L TS SP PC OP CODE MNEMONIC  
01000100 00 01 03 00 0D 00 00 0000 FFD4 FF42 FF RST 7  
-----
```

Para salir del HYPERTERMINAL solo se debe elegir la opción Salir del Menú Archivo

Anexo 8.5 Guía de Practica N° 5

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 5

Suma / Resta de 8 y 16 bits

OBJETIVO.

Conocer las instrucciones utilizadas para la suma y la resta tanto de 8 como de 16 bits, y aplicarlas a programas sencillos para verificar su funcionalidad.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO

PRIMERA PARTE: SUMA DE 8 BITS

INSTRUCCIONES

ADD C op code = 81

El registro C es adicionado al registro A con el resultado siendo almacenado en el registro A. Las banderas Z, S, P, CY y AC son afectadas.

MOV A, B op code = 78

Los contenidos del registro de B son copiados al registro A. Ninguna bandera se afecta. Este programa adiciona los registros B y C juntos y almacena el resultado en el registro A. Esto se hace copiando el valor B a A y luego adicionando el valor de C a A.

```
org 00f01h  
loop:  mov a, b    ; A es una copia de B
```

	add c	; adicionar C a A
	rst 7	; retorna el control a MOS
Dirección	Datos	Instrucción
FF01	78	MOV A, B
FF02	81	ADD C
FF03	FF	RST 7

Para probar la instrucción ADD C, cargar el programa en memoria, apretar el botón de reset y hacer lo siguiente:

1) Los registros B y C serán cargados con números cuya suma no será mayor que 9 de modo que el resultado no tendrá que ser traducido a decimal para ver que la instrucción trabajada correctamente. Cargar el par de registros BC con 0403, que es lo mismo que cargar B con 4 y C con 3, luego ejecutar el programa. El programa se alojará en FF03 debido a la instrucción RST 7. (Porque el microprocesador es tan rápido que el programa se detendrá casi inmediatamente, así se puede pensar que el programa no se ejecuta en realidad, pero al examinar los registros se verá que hizo.) Si se examina el registro A verá que tiene el valor 7, y si se convierte el registro de bandera a binario se verá que el bit 0 (la bandera de acarreo) es 0 lo que significa que el resultado de la adición era bastante pequeño como para encajar en un registro.

2) Apretar el botón de reset y cargar la pareja de registros BC con FFFF, luego ejecutar el programa. Cuando el programa se detenga en FF03 se deberá examinar el registro A y el registro de bandera y se verá que el registro A es ahora FE y el bit 0 del registro de bandera (la bandera de acarreo) tiene ahora 1 que indica que el resultado de la adición era demasiado grande como para encajar en un registro.

SEGUNDA PARTE: RESTA DE 8 BITS

INSTRUCCIÓN

SUB C op code = 91

El registro C es sustraído del registro A con el resultado siendo almacenando en el registro A. Las banderas Z, S, P, CY y AC son afectadas.

Si el programa previo está todavía en memoria y se cambia la instrucción FF02 a 91 (SUB C) el programa siguiente es el resultado. Este programa substraerá el

registro C del registro B y almacenara el resultado en el acumulador. Esto se hace copiando el valor de B en A y luego substrayendo C de A. el resultado estar en el registro A.

```

org 00f01h
loop:  mov a, b    ; A es una copia de B
       sub c      ; subtrae C de A
       rst 7      ; retorna el control a MOS

```

Dirección	Datos	Instrucción
FF01	78	MOV A, B
FF02	91	SUB C
FF03	FF	RST 7

1) Para probar la instrucción SUB C, cargar el programa en memoria y mover el contador de programa a FF01. Cargar el par de registros BC 0906, que es igual que B sea cargado con 9 y C con 6, luego ejecutar el programa. Cuando el programa se detenga en FF03, Se notara que el registro A ahora es 3 y si se convierte el registro de bandera a binario se verá que la bandera de acarreo es 0.

2) Apretar el botón de reset, cargar B con 1F y C con 3D luego ejecutar el programa. Se verá que el registro A es E2 y la bandera de acarreo es 1.

En cada substracción, la bandera de acarreo es establecida si el numero en el registro A es más pequeño que el valor que se substraerá de él (en este caso el registro A era 1F y 3D ha sido substraído de él). La bandera de acarreo puede ser más apropiadamente ser referida como una "bandera de préstamo" cuando el valor de bandera es el resultado de una substracción.

TERCERA PARTE: RESTA DE 16 BITS

INSTRUCCIONES

SUB E op code = 93

El registro E es substraído del registro A con el resultado siendo almacenando en el registro A. Las banderas Z, S, P, CY y AC son afectadas.

SBB D op code = 9A

Substracción del registro D y la bandera de acarreo del registro A con el resultado siendo almacenando en el registro A. Las banderas Z, S, P, CY y AC son afectadas.

MOV A, C op code = 79

Los contenidos del registro C son copiados al registro A. Ninguna bandera se afecta.

MOV C, A op code = 4F

Los contenidos del registro A son copiados al registro C. Ninguna bandera se afecta.

MOV B, A op code = 47

Los contenidos del registro A son copiados al registro B. Ninguna bandera se afecta.

Al usar la bandera de acarreo con las instrucciones de substracción permite substraer 16 bits, 24 bits o números aun más grandes. El programa siguiente subtrae el valor del par de registros DE del valor del par de registros BC, dejando el resultado para después.

```
org 00f01h
                ; substraer los bytes de menor orden primero
mov a, c       ; A es una copia de c
sub e          ; A = A - E. Establecer la bandera de acarreo si A < E
mov c, a       ; C = A - E
                ; substraer los bytes de mayor orden y la bandera de acarreo
mov a, b       ; A es una copia de B
sbb d         ; A = A - D - bandera de acarreo
mov b, a       ; B = A
rst 7         ; detener el programa
```

Dirección	Datos	Instrucción
FF01	79	MOV A, C
FF02	93	SUB E

FF03	4F	MOV C, A
FF04	78	MOV A, B
FF05	9A	SBB D
FF06	47	MOV B, A
FF07	FF	RST 7

Cargar el par de registros BC con 2100 hex y cargar el par de registros DE con 10FF hex. Lo siguiente es una ecuación que el programa ejecutara.

$$\begin{array}{l}
 2100 \text{ hex} \\
 - \underline{10FF \text{ hex}} \\
 = 1001 \text{ hex}
 \end{array}$$

Si ejecuta paso a paso el programa empezando desde la dirección FF01 verá lo siguiente después de cada paso:

PC ACTUAL

FF01 A igual a C que es 00.

FF02 $A = A - E$. Puesto que A (00) es menos que E (FF), la bandera de acarreo es establecida (examinar el bit 0 del registro de bandera). La substracción se ejecutará como si hubiera un dígito 1 a la izquierda del registro A. En este caso la substracción se ejecutará como si $A = 100 \text{ hex}$. $(100 \text{ hex} - FF \text{ hex}) = 01$ así A es igual a 01.

FF03 C igual a A que es 01.

FF04 A igual a B que es 21 hex.

FF05 el registro D y la bandera de acarreo son substraídos del registro A ($A = A - D - \text{bandera de acarreo} = 21 \text{ hex} - 10 \text{ hex} - 1 = 10 \text{ hex}$).

FF06 B es igual a A que es 10 hex

FF07 fin del programa.

Examinar el par de registros BC y se verá que ahora es 1001 hex justo como se predijo en la ecuación que fue mostrada. Probar apretando el botón de reset y cargar BC y DE con otros valores tal que el registro D sea mayor que el registro E y ejecutar paso a paso el programa de nuevo.

CUARTA PARTE: SUMA DE 16 BITS

INSTRUCCIONES

ADD E op code = 83

Adiciona el registro E al registro A, almacenando el resultado en el registro A.

Las banderas Z, S, P, CY y AC son afectadas.

ADC D op code = 8A

Adiciona el registro D y la bandera de acarreo al registro A, almacenando el resultado en el registro A. Las banderas Z, S, P, CY y AC son afectadas.

El mismo concepto usado en la lección previa puede ser aplicado para hacer un programa de adición de 16 bits.

```
org 00f01h
; adicionar los bytes de menor orden primero
mov a, c ; A es una copia de c
add e ; A = A + E. Establecer la bandera de acarreo si el resultado > 0FFh
mov c, a ; C = A + E
; adicionar los bytes de mayor orden y la bandera de acarreo
mov a, b ; A es una copia de B
sbb d ; A = A + D + bandera de acarreo
mov b, a ; B = A
rst 7 ; detener el programa
```

Dirección	Datos	Instrucción
FF01	79	MOV A, C
FF02	83	ADD E
FF03	4F	MOV C, A
FF04	78	MOV A, B
FF05	8A	ADC D
FF06	47	MOV B, A
FF07	FF	RST 7

Si el programa previo está todavía en memoria los únicos cambios que necesitan ser hechos son la instrucción SUB se debe reemplazar con un ADD, y el SBC con un ADC. Esto se hace cambiando el byte de la dirección FF02 a 83 y el byte en la dirección FF05 a 8A. Si el programa no está en memoria, cargar el que esta

listado arriba.

Cargar el par de registros BC con 2302 hex y cargar el par de registros DE con 10FF hex. La siguiente es una ecuación que el programa ejecutará.

$$\begin{array}{r} 2302 \text{ hex} \\ + 10FF \text{ hex} \\ \hline = 3401 \text{ hex} \end{array}$$

Si se ejecuta el programa paso a paso empezando con la dirección FF01 se verá lo siguiente después de cada paso:

PC ACTUAL

FF01 A igual a C que es 02.

FF02 $A = A + E$. Puesto que $A + E = 101$ hex y que es demasiado grande para encajar en el registro A, la bandera de acarreo es establecida (examinar el bit 0 del registro de bandera) y el registro A es 01.

FF03 C igual a A que es 01.

FF04 A igual a B que es 23 hex.

FF05 el registro D y la bandera de acarreo es adicionada al registro A ($A = A + D + \text{bandera de acarreo} = 23 \text{ hex} + 10 \text{ hex} + 1 = 34 \text{ hex}$).

FF06 B es igual a A que es 34 hex

FF07 fin del programa.

Examinar el par de registros BC y se verá que ahora es 3401 hex justo como se predijo en la ecuación que se había mostrado. Se puede probar apretar el botón de reset; cargando BC y DE con otros valores y corriendo el programa de nuevo. Se puede probar cargando los registros C y E con valores que no se establecerán en la bandera de acarreo cuando la instrucción ADD E es ejecutada.

Anexo 8.6 Guía de Practica N° 6

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 6

BCD, Multiplicación y División

OBJETIVO.

Conocer las instrucciones utilizadas para la resta utilizando el complemento a dos, así como también las instrucciones utilizadas para la suma de 16 bits en bcd, en la multiplicación y la división siendo aplicadas en programas sencillos para verificar su funcionalidad.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO

PRIMERA PARTE: RESTA 16 BITS USANDO LA SUMA DE COMPLEMENTO DE DOS

INSTRUCCIONES

DAD B op code = 09

Adiciona el número de 16 bits en BC al número de 16 bits en HL y almacena el resultado en HL. La bandera de acarreo es 1 si el resultado es demasiado grande para encajar en HL.

CMA op code = 2F

Complementa los bits en el registro A (ejemplo. Cambia los unos a ceros y los ceros a unos). Ninguna bandera se afecta.

INX B op code = 13

Incrementa el valor en el par de registro BC. Ninguna bandera se afecta

La instrucción DAD del microprocesador 8085 proporciona un método más rápido adición de 16 bits. Lo siguiente es un simple programa que adiciona el par de registros BC al par de registros HL. Probar ejecutando el programa con HL = 01FF y BC = 7001, el resultado en HL será 7200 hex.

Dirección	Datos	Instrucción
FF01	09	DAD B
FF02	FF	RST 7

No hay ninguna instrucción para la substracción de 16 bits, pero esto se puede simular cambiando el registro que se quiere substraer a su forma de complemento dos y luego ejecutando la instrucción DAD. Un número es cambiado a la forma de complemento dos complementando cada bit y luego incrementando el número.

Abajo hay un programa que subtrae BC de HL cambiando BC a la forma de su complemento dos, luego el programa adiciona BC a HL repetidamente. Notar que el resultado en la bandera de acarreo no indicara un préstamo como se hizo en las instrucciones SUB y SBC.

```
org 00f01
mov a, b      ; Poner B en A
cma          ; así puede ser complemento
mov b, a     ; poner el nuevo valor en B
mov a, c     ; hacer lo mismo para C
cma
mov c, a     ; BC ha sido complementado (primer complemento)
inx b       ; ahora incrementa el par de registros BC que será el segundo
           ; complemento
loop: dad b  ; adicionar BC a HL
      jmp loop ; adicionar de nuevo
end
```

Dirección	Datos	Instrucción
FF01	F3	MOV A, B
FF02	2F	CMA
FF03	47	MOV B, A
FF04	79	MOV A, C
FF05	2F	CMA
FF06	4F	MOV C, A

FF07	03	INX B
FF08	09	DAD B
FF09	C3	JMP FF08
FF0A	08	
FF0B	FF	

Agregar el programa anterior en memoria, luego cargar HL con FFFF y BC con 00FF. Poner un punto de ruptura en FF08 y luego ejecutar el programa. Cuando el programa se detenga, se verá que BC es ahora el complemento dos de 00FF que es FF01. Ejecutar paso a paso la instrucción DAD y HL será FF00 (FFFF - FF), ejecutar paso a paso JMP y DAD de nuevo y HL será FE01 (FF00 - FF) y así sucesivamente. Con cada instrucción DAD 00FF es substraído de HL.

SEGUNDA PARTE: SUMA DE 16 BITS EN BCD

INSTRUCCIONES

DAA op code = 27

Cambia el valor en el registro A a dos dígitos decimal codificados en binario. Esto se hace por: (a) adicionar 6 al registro A si el valor de sus 4 bits menores es mayor que 9, o si la bandera auxiliar de acarreo es establecida. (b) adicionando 6 a los 4 bits superiores del registro A si son mayores que 9 o si la bandera de acarreo ha sido establecida. Las banderas Z, S, P, CY, y AC son afectadas.

NOP op code = 00

Esta instrucción no hace nada pero toma un byte en un programa. Ninguna bandera se afecta.

Los números decimales pueden ser guardados en su forma binaria como numero decimal codificado en binario (BCD). Cada dígito decimal toma 4 bits justo como los números hexadecimales lo hacen, pero en BCD, los dígitos A-F no son números validos.

Por ejemplo, el número hexadecimal 9999 es 9999 en BCD. La instrucción DAA fue provista así sería posible ejecutar las matemáticas de decimal codificado en binario usando las instrucciones ordinarias de adición hex.

Según su definición, DAA dará los siguientes resultados siguiendo una instrucción de adición hex:

19	67	43
<u>+ 29</u>	<u>+ 75</u>	<u>+ 72</u> (adición hex)
(adición hex)	(adición hex)	
42	DC	B5
<u>+ 06</u>	<u>+ 66</u> (DAA)	<u>+ 60</u> (DAA)
48	142	115

Notar que DAA pueda solamente dar 2 dígitos decimal y la bandera de acarreo como resultado. En estos ejemplos la bandera de acarreo representa los dígito de los cientos.

El siguiente programa adicionara el numero BCD en DE al número BCD en BC y almacenara el resultado de BCD en BC.

```

org 00f01
                ; agregar los bytes de orden inferior primero
mov a, c       ; A es una copia de C
add e          ; A = A + E. Establecer la bandera de acarreo si A<E
daa           ; acumulador de ajuste decimal
mov c, a       ; C = A + E
                ; agregar los bytes de orden superior y acarreo desde DAA
mov a, b       ; A es una copia de B
adc d          ; A = A + D + bandera de acarreo
daa           ; acumulador de ajuste decimal
mov b, a       ; B = A
rst 7         ; detiene el programa
end

```

Dirección	Datos	Instrucción
FF01	79	MOV A, C
FF02	83	ADD E
FF03	27	DAA
FF04	4F	MOV C, A
FF05	78	MOV A, B
FF06	8A	ADC D
FF07	27	DAA
FF08	47	MOV B, A
FF09	FF	RST 7

Introducir el programa en memoria luego presionar reset y cargar BC con 1934 y cargar DE con 1879.

PC ACTUAL

FF01 Ejecutar un solo paso a FF03 y examinar el registro A (será AD hex). Puesto que ambos 4 bits superiores e inferiores 4 del registro A son mayores que 9 entonces 66 será añadido a él en la instrucción de DAA que sigue.

FF03 Ejecutar un solo paso la instrucción DAA y examinar el registro A y verá que 66 fue añadido a el que establece la bandera de acarreo e hizo al registro A 13.

FF04 Ejecutar un solo paso a FF07 y examinar el registro A (será 32) y el registro de bandera. La bandera de acarreo auxiliar será 1 y la bandera de acarreo será 0 así la instrucción de DAA que sigue añadirá 6 al registro A.

FF07 Ejecutar un solo paso, la instrucción DAA a esta dirección y examinando al registro A verá que 6 ha sido añadido a él.

FF08 Ejecutar un solo paso, y el registro A será copiado al registro B.

FF09 fin del programa.

Verificar que el par de registros BC es 3813, que es la suma de los números decimales: 1934 y 1879. Ahora reemplazar las instrucciones DAA a FF03 y FF07 con 00, que es una instrucción que no hace nada. Cargar DE y BC con los mismos valores como antes luego ejecutar el programa de nuevo. Esta vez los registros BC y DE se sumarán como si fueran números hexadecimales y el resultado será 31AD hex.

TERCERA PARTE: MULTIPLICACIÓN

INSTRUCCIONES

RAL op code = 17

Rota todos los bits en el acumulador una posición a la izquierda y pone la bandera de acarreo en bit 0 y pone el valor que fue rotado fuera del bit 7 a la bandera de acarreo. Sólo la bandera de acarreo es afectada.

DAD H op code = 29

Añade HL a HL y almacena el resultado en HL. Si el resultado es mayor que 16 bits, la bandera de acarreo será 1, de otra forma será 0. Sólo la bandera de acarreo es afectada.

DAD D op code = 19

Añade DE a HL y almacena el resultado en HL. Si el resultado es mayor que 16 bits, la bandera de acarreo será 1, de otra forma es 0. Sólo la bandera de acarreo es afectada.

ORA A op code = B7

Esta instrucción aplica lógicamente el operador OR al registro A consigo mismo. Las banderas CY y AC serán 0, y las banderas Z, S y P serán afectadas. De acuerdo con el valor del registro A. El registro A no es cambiado.

El siguiente programa ilustra el uso de una instrucción de rotación como un método de multiplicación. Cambiar los dígitos de un número base 10 (decimal) a la izquierda es igual que multiplicar el número por su base, que es 10. Del mismo modo, en el sistema numérico base 2 (binario), cambiando un número binario a la izquierda es igual que multiplicarlo por su base, que es 2. Por ejemplo:

Decimal:

1423 1 dígito cambiado a la izquierda = 14230

1423 * 10 = 14230

Binario:

0111 1 dígito cambiado a la izquierda = 01110

0111 * 0010 = 01110

Abajo se muestra el ejemplo convertido a decimal

(7 * 2 = 14)

Usando más de un desplazamiento hacia la izquierda en un número que puede multiplicar eficazmente el número por 4, 8, 16, 32, 64, 128 y así sucesivamente. El número de veces que se cambia un número a la izquierda, determina la cantidad que el número original sea sido multiplicado. Por ejemplo si cambia un número a la derecha 3 veces, es igual que multiplicarlo por $2*2*2$ u 8.

A menudo al hacer la multiplicación usando desplazamientos hacia la izquierda, el resultado no encajará en los 8 bits de registro. La instrucción RAL, debido a la manera en que usa la bandera de acarreo permitirá escribir programas que cambia los bits de registro a registro que puede permitir los cambios de números muy grandes. En las instrucciones previas la bandera de acarreo indica un acarreo o préstamo, pero instrucción RAL las usa para otros dos propósitos: RAL cambia la bandera de acarreo a bit 0 del registro A y luego carga la bandera de acarreo con el valor cambiado del bit 7 del registro A. El programa de abajo muestra cómo la bandera de acarreo puede ser usada para pasar un bit de un registro a otro. El programa cambia una posición de bit a la izquierda los bits en el par de registros DE.

```

.org
0f001h
loop: mov a, e    ; carga A con byte inferior primero
      ora a      ; limpia la bandera de acarreo (cy=0) así RAL cambiara 0 a
                  bit 0
      ral        ; rotación a la izquierda por medio de acarreo
      mov e,a    ; guarda en E
      mov a,d    ; carga A con byte superior
                  ; lo que fue cambiado del bit 7 en el ultimo RAL está en la
                  bandera CY
      ral        ; lo que es cambiado en el bit 0 en este RAL
      mov d,a    ; guardar en D
      rst 7      ; regresa a MOS

```

Dirección	Datos	Instrucción
FF01	7B	MOV A, E
FF02	B7	ORA A
FF03	17	RAL
FF04	5F	MOV E, A
FF05	7A	MOV A, D

FF06	17	RAL
FF07	57	MOV D, A
FF08	FF	RST 7

Cargar el programa anterior en memoria, luego cargar el par de registros DE con 0080 hex (128 decimal) y correr el programa. Este valor es escogido porque mostrará que después que se corre el programa, el bit 7 de E es cambiado al bit 0 de D, que da un resultado de 0100 (128 * 2 = 256 decimal). Examinar el par de registros DE para verificar esto. Correr el programa usando otros valores para DE y verificar que el programa realmente multiplica DE por 2, pero hay que estar seguro que los valores sea menores a 8000 hex o el resultado no encajará en el par de registro DE.

Si se quiere multiplicar el par de registros HL por 2, la instrucción DAD H puede ser usada. Añade HL a HL que es igual que multiplicar HL por 2. Abajo esta un simple programa que ilustra la instrucción de DAD H.

```

                .org 0f001h
loop:  dad h      ; añade hl a hl
        jmp loop  ; lo hace de nuevo

```

Dirección	Datos	Instrucción
FF01	29	DAD H
FF02	C3	JMP FF01
FF03	01	
FF04	FF	

Cargar el programa en memoria y presionar el botón de reset luego ejecute mismas pruebas que fueron hechas para el ejemplo previo, cargando el par de registros HL con los mismos valores con los que DE fue cargado y correrlo paso a paso. Los resultados fueron los mismos.

Un rápido algoritmo " 10 veces " puede ser hecho usando una combinación de 2 instrucciones de multiplicación y una instrucción de adición. Abajo esta la

ecuación " $x = y * 10$ " que está rota en una forma que el 8085 pueden manejar fácilmente:

$$X = Y * 10$$

$$X = (Y * 5) * 2$$

$$X = ((Y * 4) + Y) * 2$$

$$X = ((Y * 2 * 2) + Y) * 2$$

La última ecuación está traducida al siguiente programa.

```
.org 0ff01h
mov d,h      ; pone el valor original
mov e,l      ; de HL a DE
dad h        ; HL = HL * 2
dad h        ; HL = HL * 2
dad d        ; HL = HL + DE
dad h        ; HL = HL * 2
rst 7        ; retornar a MOS
```

Cargar el programa en memoria, luego cargar a HL con 0003. Después de correr el programa, HL será 001E que es 30 en decimal. Cambiar el contador de programa de nuevo a FF01 (no apretar reset, o los registros serán limpiados) luego correr el programa de nuevo. Esta vez HL será 012C que es 300 en decimal.

En la lección "Usando las Subrutinas de Sistema Operativo Monitor " hay una descripción de un servicio que multiplica dos números de 16 bits.

CUARTA PARTE: DIVISIÓN

INSTRUCCIONES

RAR op code= 1F

Cambia todos los bits en el acumulador una posición a la derecha y pone la bandera de acarreo en el bit 7 y pone el valor que fue cambiado del bit 0 en la bandera de acarreo. Sólo la bandera de acarreo es afectada.

MOV A,D op code = 7A

Los contenidos del registro D son copiados al registro A. Ninguna Bandera se afecta.

MOV D, A op code = 57

Los contenidos del registro A son copiados al registro D. Ninguna Bandera se afecta.

MOV A, E op code = 7B

Los contenidos del registro E son copiados al registro A. Ninguna Bandera se afecta.

MOV E, A op code = 5F

Los contenidos del registro A son copiados al registro E. Ninguna Bandera se afecta.

Así como cambiando un número hacia la izquierda es igual que multiplicarlo por su base, cambiar un número hacia la derecha es igual que dividirlo por su base. Esto permite que el 8085 ejecute la división por 2 cambiando a la derecha un número binario. Cuando la división es hecha de este modo, el residuo será el bit movido del bit de la posición más baja. El siguiente programa divide el numero en DE en dos, dejando el cociente en DE y el residuo en la bandera de acarreo.

```
.org
0f001h
loop: mov a, d    ; carga A con byte superior primero
      ora a      ; limpia la bandera de acarreo (cy=0)
      rar        ; rotación a la derecha por medio de acarreo
      mov d,a    ; guarda en D
      mov a,e    ; carga A con byte inferior
                  ; lo que fue cambiado del bit 0 en el ultimo RAR será puesto
                  ; en la ;bandera
      rar        ; CY lo que es cambiado en el bit 7 en este RAR
      mov e,a    ; guardar en E
      rst 7      ; regresa a MOS
```

Dirección	Datos	Instrucción
FF01	7A	MOV A,D
FF02	B7	ORA A
FF03	1F	RAR
FF04	57	MOV D,A
FF05	7B	MOV A,E

FF06	1F	RAR
FF07	5F	MOV E,A
FF08	FF	RST 7

Cargar el programa en memoria, luego cargar el par de registros DE con 0007 hex y luego correr el programa. Se verá que DE tiene ahora 3 y si se examina el registro A.F. se verá que la bandera de acarreo es 1 que indica un residuo de 1. Probar otros valores de DE y verificar que el programa en realidad divide el registro de DE en 2.

En la lección " Usando las Subrutinas de Sistema Operativo Monitor " hay una descripción de un servicio de división que divide un valor en el par de registros HL por el valor en el par de registro de DE.

Anexo 8.7 Guía de Practica N° 7

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 7 Uso de Instrucciones Lógicas y Condicionales

OBJETIVO: Conocer y utilizar las instrucciones lógicas y condicionales por medio de ejemplos prácticos.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

PRIMERA PARTE: USANDO INSTRUCCIONES LÓGICAS

Instrucciones Utilizadas.

ANI <byte> op code = E6

Aplica una operación lógica "AND" del registro A con el <byte> seguido por el código de operación, y almacena el resultado en el registro A. En esta operación las banderas Z, S, P, CY y AC son afectadas.

XRI <byte> op code = EE

Aplica una operación lógica "XOR" del registro A con el <byte> seguido por el código de operación, y almacena el resultado en el registro A. En esta operación las banderas Z, S, P, CY y AC son afectadas.

ORI <byte> op code = FE

Aplica una operación lógica "OR" del registro A con el <byte> seguido por el código de operación, y almacena el resultado en el registro A. En esta operación las banderas Z, S, P, CY y AC son afectadas.

En ocasiones, se desea cambiar los "bits" de un registro a valores distintos.

Observe el resultado del siguiente ejemplo utilizando la instrucción ANI

10010010 (Registro A)

ANI 11000011 (<byte>)

10000010 (resultado)

Si existe un valor de 0 en cualquiera de los bits que componen el <byte> el resultado de cualquier operación con este será 0. Si existe un valor de 1 en cualquiera de los bits que componen el <byte> entonces, el bit correspondiente en el registro A será directamente copiado al resultado. Al conocer esto, la operación AND debería de ser de gran ayuda dentro de un programa en el cual podría ser necesario revisar el estado del bit de puerto de entrada. A continuación se muestra la forma en que se opera. El registro A contiene el valor del puerto de entrada, y <byte> contiene un bit establecido, el cual corresponde al "bit" que se desea examinar. Luego, todos los demás bits están a cero.

00101111 (valor del Puerto de entrada)

ANI 00100000 (examine bit 5)

00100000 (resultado)

Todos los bits correspondientes al puerto de entrada han dado como resultado cero, con excepción del bit que se desea examinar es 1. De acuerdo a la lógica del operador AND, todos los bits deberían tener un valor de cero, si el bit del puerto de entrada numero 5 estuviese apagado. Este método de convertir ciertos "bits" a

cero (y pasando otros bits) es llamado "enmascaramiento", y al método de operar los datos mediante AND y agrégalos al registro A se le llama "mascara"

En caso de que sea necesario cambiar el valor de los bits a uno, deberá utilizarse el operador OR. Como se muestra a continuación, el valor de un bit del registro A es copiado al resultado si el bit correspondiente en <byte> es cero. Y en el resultado, un bit tendrá un valor de 1 si el valor del bit correspondiente en <byte> es 1.

```
01000011 (registro A)
ORI  10101110 (<byte>)
      11101111 (resultado)
```

Para complementar o alterar algún bit en particular (cambie los 1's a 0's y los 0's a 1's). Acá puede ser utilizado el operador XOR

```
10101011 (registro A)
XRI  11110000 (<byte>)
      01011011 (resultado)
```

Si un bit dentro de <byte> tiene un valor de 1, el bit correspondiente en A será alterado en el resultado, de lo contrario si su valor es cero, el bit correspondiente en A será colocado directamente en el resultado sin ser cambiado ni alterado.

El siguiente programa combina las instrucciones ANI, ORI y XRI para cambiar los datos provenientes del puerto de entrada B (conectado a los interruptores DIP) a través de la asignación, reasignación y modificación del valor de los bits. Este proceso de transformación del valor de los bits es llamado "bit de manipulación". Las instrucciones de manipulación de bits son usados por el MOS para permitir que el 8085 controle otros chips en el PRIMER y para leer datos a partir de ellos. Digitar el siguiente programa en ensamblador en el simulador para poder

comprobar los resultados obtenidos. (Desde la instrucción .org 0ff01h)

```
leds    equ 11h
dips    equ 12h
        .org 0ff01h
loop:   in dips          ; toma los valores de entrada de los interruptores DIP
        ani 00111111b   ; hace los bit 6 y 7, cero
        ori 00110000b   ; hace los bit 5 y 4, uno
        xri 00000110b   ; altera los bits 2 y 1
        out leds        ; muestra A en los LEDs
        jmp loop        ; salta a loop
        end
```

Cargar el siguiente programa en el TRAINER siguiendo las instrucciones.

Dirección	Datos	Instrucción
FF01	DB	IN 12
FF02	12	
FF03	E6	ANI 3F
FF04	3F	
FF05	F6	ORI 30
FF06	30	
FF07	EE	XRI 06
FF08	06	
FF09	D3	OUT 11
FF0A	11	
FF0B	C3	JMP FF01
FF0C	01	
FF0D	FF	

Cargar el programa anterior a la memoria, luego presionar "reset" y ejecutarlo. Se podrá observar que los LED's se encenderán de acuerdo a la siguiente tabla:

NUMERO ESTADO
DE LED

7	Encendido
6	Encendido
5	Apagado
4	Apagado
3	Apagado si el interruptor DIP 3 está apagado o encendido si el interruptor esta encendido
2	Encendido si el interruptor DIP 2 está apagado o apagado si el interruptor esta encendido
1	Encendido si el interruptor DIP 1 está apagado o apagado si el interruptor esta encendido
0	Apagado si el interruptor DIP 0 está apagado o encendido si el interruptor esta encendido

NOTA: Si se está frente al PRIMER, los interruptores DIP están enumerados del 7-0 numerándolos de izquierda a derecha. Además, un interruptor DIP en la posición de encendido devolverá un cero lógico, y en la posición de apagado devolverá un uno lógico.

SEGUNDA PARTE: USO DE CONDICIONALES

JNZ <addr> op code = C2

Si la bandera cero es 0, carga el registro PC con los dos bytes siguiendo el código de operación, de lo contrario empezara a ejecutar la instrucción después de este. El primer byte siguiendo al código de operación es el byte bajo de la dirección, el segundo byte es el byte alto de la dirección.

Ninguna bandera es afectada.

Digitar el siguiente programa en el simulador para ver y comprobar los resultados que se obtengan en el Primer.


```

leds    equ 11h
dips    equ 12h
        .org 0ff01h

loop:   in dips        ; carga al registro a con los valores de los interruptores DIP
        out leds       ; envía a la salida el valor de los LEDs
        ora a          ; establece la bandera z si a = 0
        jnz loop       ; salta el loop si a no = 0
        rst 7          ; regresa a MOS

```

La instrucción JNZ es similar que la instrucción JMP solamente que tiene una palabra (NZ) anexa a ella. Los condicionales, por lo general, controlan si se ejecuta o no una instrucción o si se ignorará. Si se ignora la instrucción, la instrucción siguiente se ejecutará. No solo puede que la instrucción JMP sea controlada por condicionales sino también las instrucciones CALL y RET que son tratadas.

Abajo hay una lista de las diferentes instrucciones de salto que usan condicionales y los valores de las banderas que son necesarios para el salto por ocurrir:

JNZ = saltar si el resultado de la operación no fue 0 (si z = 0)
JZ = saltar si el resultado de la operación fue 0 (si z = 1)
JNC = saltar si no hay acarreo desde la operación (si CY = 0)
JC = saltar si la operación causó un acarreo (si CY = 1)
JPO = saltar si la paridad del resultado es impar (si p = 0)
JPE = saltar si la paridad del resultado es par (si p = 1)
JP = saltar si un numero positivo resulta (si s = 0)
JM = saltar si un numero positivo resulta (si s = 1)

Cargar e introducir el siguiente programa al Primer TRAINER, dato por dato.

Dirección	Datos	Instrucción
FF01	DB	IN 12
FF02	12	
FF03	D3	OUT 11
FF04	11	
FF05	B7	ORA A
FF06	C2	JNZ FF01
FF07	01	
FF08	FF	
FF09	FF	RST 7

Apretar el botón reset y entrar al programa en la memoria. Ahora poner los interruptores DIP a la posición de apagado así el valor de los interruptores de DIP no será cero. Correr el programa y ello debe trabajar igual que en la lección 2, hasta que el valor de los interruptores DIP cambie a cero. Esto sucede cuando los interruptores de DIP son movidos a la posición de encendido y cuando esto se hace, el programa ejecutará la instrucción RST 7, retornar al MOS y mostrar "FF09 FF" que es la dirección y el código de operación de la instrucción RST 7.

Para analizar el programa apretar el botón de reset y ejecutar los pasos siguientes:

PC ACTUAL

FF01 Mover los interruptores DIP a la posición de apagado y ejecutar paso a paso a la dirección FF06

FF06 Examinar el registro de bandera y verá que el bit 6 (la bandera cero) es 0 indicando que la OR A no dio un cero como consecuencia. Ejecutando una operación OR A con A solamente dará 0 como resultado si A es igual a cero y así como se puede ver, A no es 0. Ejecutando paso a paso y la instrucción JNZ FF01 será ejecutado porque el condicional es verdadero (entonces la bandera cero es 0).

FF01 mover el interruptor DIP a la posición de encendido, de modo que el valor del interruptor DIP será 0 y ejecutar paso a paso a la dirección FF06 de nuevo.

FF06 examinar el registro de bandera y se verá que el bit 6 (la bandera cero) es 1 indicando que la instrucción ORA A cero como resultado. Como usted puede ver, el registro A es 0. Ejecutar paso a paso y la instrucción JNZ FF01 no serán ejecutados porque el condicional es falso (puesto que la bandera cero es 1).

FF09 fin del programa.

Anexo 8.8 Guía de Practica N° 8

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 8

Uso de Direccionamiento Indirecto de Registros

OBJETIVO: Conocer el uso y comportamiento de los registros, así como las operaciones básicas con ellos.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

PRIMERA PARTE: DIRECCIONAMIENTO INDIRECTO DE REGISTRO SIMPLE.

Instrucciones Utilizadas.

LXI H, <word> op code = 21

Cargar el par de registros HL con la palabra (1 palabra = 2 bytes) que antecede a op code. El byte que antecede a op code va hacia el registro L y el byte que le sigue al primer byte, va hacia el registro H. Ninguna bandera se ve afectada.

MVI M, <byte> op code = 36

Copiar el byte que antecede a op code en el byte de memoria señalado por el registro HL. Ningún registro se ve afectado.

MOV E, M op code = 7E

Copiar el byte en memoria señalada por el registro HL al registro E. Ninguna bandera se ve afectada.

INX H op code = 23

Incrementar el valor del registro HL. Ninguna bandera se ve afectada.

MOV M,E

Copiar el registro E al byte en memoria señalada por el registro HL. Ningún registro se ve afectado.

De acuerdo a la definición de MVI M, <byte>, MOV M, E y MOV E, M, la letra M se refiere a que esa instrucción que trabaja de la misma forma que trabaja un registro, en que los datos pueden ser cargados o almacenados en forma de bytes. Pero aun así se considera diferente a un registro, ya que "M" hace referencia a un espacio de memoria en específico. El registro HL apunta hacia este espacio en memoria. Puesto que el registro HL es formado por 16 bits, puede apuntar hacia cualquier dirección de memoria desde 0000 a FFFF así como lo hace el registro PC.

Por ejemplo, si HL tuvo un valor de 0131(hex) y la instrucción MOV E, M fue ejecutada, los datos en la dirección de memoria 0131 serán copiados al registro E. Esto se muestra a continuación:

DIRECCIÓN	DATOS
: (direcciones desde la 0000 – 012F)	
:	
0130	3C
0131	01 ← HL
0132	CD
0133	D8

:
: (direcciones desde la 0134 – FFFF)

Si el valor de HL fuese cambiado a FFC1 y la instrucción MVI M, 03 fuese ejecutada entonces, el dato en la dirección FFC1 seria cambiado a 03.

Este método de acceso a memoria descrito en esta lección es llamado "Direccionamiento indirecto de Registro". El siguiente programa copiara un byte a partir de un espacio en memoria a otro, y además, almacenara un valor en memoria ilustrando así el uso del registro HL para acceder indirectamente a la memoria.

Digitar el siguiente código en lenguaje ensamblador en el simulador para ver y comprobar los resultados que se obtengan en el Primer TRAINER.

```
.org 00f01h
lxi h, addr ;carga HL con la dirección del byte reservado
mov e,m ; carga E con el dato que está apuntando a HL
inx h ; HL = HL + 1 (apunta a la siguiente dirección)
inx h ; HL = HL + 1 (apunta a la siguiente dirección)
inx h ; HL = HL + 1 (apunta a la siguiente dirección)
inx h ; HL = HL + 1 (apunta a la siguiente dirección)
mov m,e ; guarda E en la nueva dirección de HL
addr: inx h ; HL = HL + 1 (apunta a la siguiente dirección)
mvi m,77h ; guarda a 77h en la nueva dirección de HL
rst 7 ; regresa a MOS
ds 2 ; esta es una dirección reservada de 2 bytes
```

Ahora introducir los datos siguientes en el Primer TRAINER y seguir las instrucciones:

Dirección	Datos	Instrucción
FF01	21	LXI H, FF0A
FF02	0A	
FF03	FF	
FF04	5E	MOV E,M
FF05	23	INX H
FF06	23	INX H
FF07	23	INX H
FF08	23	INX H
FF09	73	MOV M,E
FF0A	23	INX H
FF0B	36	MVI M, 77H
FF0C	77	
FF0D	FF	RST 7
FF0E	00	(NO UNA INSTRUCCIÓN PERO SI DATOS)
FF0F	00	(DATOS)

Ingresar el programa a la memoria, luego presionar el botón reset el cual limpiara los registros de usos múltiples (registros de propósito general) y cambie el contador del programa a FF01. Ejecute los siguientes pasos:

PC ACTUAL

FF01 ejecutar paso a paso y examinar el registro HL el cual debería tener un valor de FF0A.

FF04 ejecutar paso a paso y el dato señalado por HL (el dato en la dirección de memoria FF0A) será colocado en el registro E.

FF05 ejecutar paso a paso cuatro veces, con lo cual se ejecutaran 4 instrucciones INX H, de esta manera señalara a HL al lugar donde almacenara el dato

FF09 ejecutar paso a paso y el valor en el registro E será almacenado en una nueva dirección señalada por HL (en FF0E)

FF0A ejecutar paso a paso y una instrucción INX H será ejecutada.

FF0B ejecutar paso a paso y 77 será almacenado en una nueva dirección señalada por HL (en FF0F)

FF0D Aquí finaliza el programa. Presionar el botón "enter" y el data en FF0E será mostrado. Se mostrara que la instrucción en FF09 (MOV M, E) almaceno el valor del registro E aquí. Presionar "enter" nuevamente y se observara que la instrucción en FF0B (MVI M, 77) actualmente almacena un valor de 77 en FF0F.

Se puede observar que este programa carga el registro E con un byte proveniente de un espacio en memoria dentro de su propio programa (FF0A) y que actualmente dicho byte representa una instrucción. Aunque esto es permitido dentro del lenguaje de máquina, no es del todo útil. El programa fue escrito de esta forma, así que el valor cargado en el registro E debería de ser conocido de antemano. Debe recordarse que el valor de HL puede ser cualquiera dentro del rango de 0000 a FFFF, de esta forma el dato puede ser almacenado o leído desde cualquier espacio en memoria.

SEGUNDA PARTE: DIRECCIONAMIENTO INDIRECTO DE REGISTRO PARA EJECUTAR 24 BITS DE ADICIÓN.

Instrucciones Utilizadas

ADD M op code= 86

Añade el byte en memoria apuntada por el par de registros HL al registro A, almacenando el resultado en HL. Las banderas Z, S, P, CY y AC son afectadas.

ADC M op code = 8E

Añade la bandera de acarreo y el byte en la memoria apuntada por el par de registros HL al registro A, almacenando el resultado en el registro A. Las banderas Z, S, P, CY y AC son afectadas.

INX H op code = 23

Incrementa el par de registros HL. Ninguna bandera es afectada.

A veces la necesidad llega de añadir dos números de 24 bits. Esto se puede hacer usando direccionamiento indirecto de registro, como el programa de ejemplo de abajo. El programa añade el número de 3 bytes a las direcciones FF10, FF11 y FF12 al número de 3 bytes en los registros C, D y E y almacena el resultado en C, D y E. El byte en E y el byte en FF01 son los bytes menos significativos; el byte C y el byte en FF12 son los bytes más significativos.

Digitar el siguiente código en lenguaje ensamblador en el simulador para ver y comprobar los resultados que se obtengan en el Primer TRAINER.

```
.org 00f01h
lxi h, num
mov a,e      ; A = byte menos significativo
add m       ; agrega el dato en la dirección hl a A
mov e,a     ; guarda el resultado en E
inx h      ; apunta a la siguiente dirección de memoria
mov a,d     ; A = segundo byte más significativo
adc m      ; agrega el acarreo y el dato en la dirección hl
           ; a A
mov d,a     ; guarda el resultado en D
inx h      ; apunta a la siguiente dirección de memoria
mov a,c     ; A = byte más significativo
adc m      ; agrega el acarreo y el dato en la dirección hl
           ; a A
mov c,a     ; guarda el resultado en C
rst 7      ; regresa a MOS
num: db 5ch, 0a3h, ; 3 bytes de almacenaje
     0eh
```

Ahora introducir los datos siguientes en el Primer TRAINER y seguir las instrucciones:

Dirección	Datos	Instrucción
FF01	21	LXI H, FF10
FF02	10	
FF03	FF	
FF04	7B	MOV A,E
FF05	86	ADD M
FF06	5F	MOV E,A
FF07	23	INX H
FF08	7A	MOV A,D
FF09	8E	ADC M
FF0A	57	MOV D,A
FF0B	23	INX H
FF0C	79	MOV A,C
FF0D	8E	ADC M
FF0E	4F	MOV C,A
FF0F	FF	RST 7
FF10	5C	(byte menos significativo)
FF11	A3	(segundo byte más significativo)
FF12	0E	(byte más significativo)

Cargar el programa anterior y los datos en la memoria. Después de cargar el programa, el número de 3 bytes almacenado en las direcciones de memoria FF12, FF11 y FF10 serán 0EA35C. Apretar el botón de reset, luego cargar C con 01 y DE con F7BF de modo que C y DE contendrán el número de 3 bytes 01F7BF, y hace lo siguiente:

PC ACTUAL

FF01 ejecutar paso a paso y HL será cargado con FF10 la cual es la dirección del byte menos significativo del numero de 3 bytes que está almacenado en memoria.

FF04 ejecutar paso a paso y el registro A será cargado con el valor del registro E. Examinar el registro A (debe ser BF)

FF05 ejecutar paso a paso y la instrucción ADD M será ejecutada, que agregara al registro A los datos a la dirección de memoria apuntada por HL. La dirección de memoria es FF10 y los datos en esa dirección son 5C, así el resultado de esta instrucción será $BF + 5C = 11B$. Debido a que 11B no encaja en el registro A, la bandera de acarreo es establecida y el registro A es 1B. Examinar el registro A y el registro de bandera para verificar que esto.

FF06 ejecutar paso a paso y el registro A será copiado al registro E.

FF07 ejecutar paso a paso y HL será FF11 el cual apunta al segundo byte más significativo del numero de 3 bytes almacenado en memoria.

FF08 ejecutar paso a paso y el registro D será copiado al registro A. Examinar el registro A; debe ser F7.

FF09 ejecutar paso a paso y la instrucción ADC M será ejecutada que agregara al registro A, la bandera de acarreo de la última instrucción ADD M y los datos en la dirección de memoria apuntada por HL. La dirección de memoria es FF11 y los datos en esa dirección son A3, así el resultado de esta instrucción será $F7 + A3 + \text{bandera de acarreo} = 19B$. Ya que 19B no encaja en el registro A, la bandera de acarreo es establecida y el registro A es 9B. Examinar el registro A y el registro de bandera para verificar esto.

FF0A ejecutar paso a paso y el registro A será copiado al registro D.

FF0B ejecutar paso a paso y HL será FF12 que apunta al byte más significativo del numero de tres bytes almacenado en la memoria.

FF0C ejecutar paso a paso y el registro A será cargado con el valor del registro C. Examinar el registro A (debe ser 01).

FF0D ejecutar paso a paso y la instrucción ADC M será ejecutada que aumentara al registro A, la bandera de acarreo de la última instrucción ADC M y los datos en la dirección de memoria apuntada por HL. La dirección de memoria es FF12 y los datos en esa dirección son 0E, así el resultado de esta instrucción será $01 + 0E + \text{bandera de acarreo} = 10$. Puesto que 10 encaja en el registro A, la bandera de

acarreo es cero y el registro A es 10. Examinar el registro A y el registro de bandera para verificar esto.

FF0E ejecutar paso a paso y el registro A será guardado en el registro C.

FF0F fin del programa.

El resultado almacenado en los registros C, D y E son 109B1B que son la suma de 01F7BF y 0EA35C. Probar correr el programa con valores diferentes en los registros C, D y E con otros valores almacenados a direcciones de memoria FF10, FF11 y FF12.

Ejercicio 3.

Direccionamiento indirecto de registro para Mover Datos

Instrucciones Utilizadas

INX B op code = 03

Incrementa el par de registros BC. Ninguna bandera es afectada.

INX D op code = 13

Incrementa el par de registros DE. Ninguna bandera es afectada.

DCR L op code = 2D

Decrementa el registro L. Las banderas Z, S, P y AC son afectadas.

LDAX B op code = 0A

Copia el registro A a la dirección de memoria apuntada por el par de registros BC al registro A. Ninguna bandera es afectada.

STAX D op code = 12

Copia el registro A al byte en memoria apuntada al par de registros DE. Ninguna bandera es afectada.

El programa de abajo usa direccionamiento indirecto, para copiar una serie de bytes de una dirección de memoria a otra. Copiando bloques de datos de una dirección de memoria a otra, esto es hecho bastante a menudo en programas de procesamiento de palabras y en programas que usan gráficos, tales como juegos de video. Para este programa el registro de BC debe ser cargado con la dirección del comienzo del bloque de memoria a mover, DE debe ser cargado con la dirección (una dirección RAM) que se quiere copiar el bloque de memoria y L deba ser cargado con el número de bytes en el bloque que se quiere copiar.

Digitar el siguiente código en lenguaje ensamblador en el simulador para ver y comprobar los resultados que se obtengan en el Primer TRAINER.

```

.org
00f01h
loop: ldax b      ; carga A con el byte apuntado por BC
      stax d      ; almacena a A en la dirección en DE
      inx b       ; incrementa BC. Esto apunta a BC a la siguiente
                  ; dirección
                  ; desde la cual se tomara el byte
      inx d       ; incrementa DE. Esto apunta a DE a la siguiente
                  ; dirección
                  ; para almacenar el byte
      dcr l       ; decrementa a L
      jnz loop    ; si L no = 0, entonces repetir de nuevo
      rst 7       ; regresa a MOS
      db 0,0      ; hace a estos bytes 0 así se podrá ver que han
                  ; cambiado

```

Ahora introducir los datos siguientes en el Primer TRAINER y seguir las instrucciones:

Dirección	Datos	Instrucción
FF01	0A	LDAX B
FF02	12	STAX D
FF03	03	INX B

FF04	13	INX D
FF05	2D	DCR L
FF06	C2	JNZ FF01
FF07	01	
FF08	FF	
FF09	FF	RST 7
FF0A	00	(datos, no es una instrucción)
FF0B	00	(también datos)

Después de entrar el programa en memoria, presionar el botón de reset y cargar BC con FF03, cargar DE con FF0A, cargar HL con 0002 y hacer lo siguiente:

PC ACTUAL

FF01 ejecutar paso a paso y el registro A será cargado con el valor del byte apuntado por el par de registros BC (el byte en FF03 es 03).

FF02 ejecutar paso a paso y el registro A será guardado en la dirección de memoria apuntada por el par de registro DE (FF0A).

FF03 ejecutar paso a paso y el par de registros BC será incrementado de modo que apuntara a la siguiente dirección de memoria (FF04) desde la cual se tomaran los datos.

FF04 ejecutar paso a paso y el par de registros DE será incrementado de modo que apuntara a la siguiente dirección de memoria (FF0B) para almacenar datos.

FF05 ejecutar paso a paso y el registro L se decrementara. Puesto que el resultado no es 0, la bandera cero es 0.

FF06 ejecutar paso a paso y JNZ FF01 cargará a PC con FF01, ya que la bandera cero es 0.

FF01 ejecutar paso a paso y el registro A será cargado con el valor del byte apuntado por el par de registros BC (el byte en FF04 es 13).

FF02 ejecutar paso a paso y el registro A será guardado en la dirección de memoria dirigida apuntada por el par de registros DE (FF0B).

FF03 ejecutar paso a paso y el par de registros BC será incrementado de modo que apuntara a la siguiente dirección de memoria (FF05) desde donde conseguirá los datos.

FF04 ejecutar paso a paso y el par de registros DE será incrementado de modo que apuntara a la siguiente dirección de memoria (FF0C) para almacenar los datos.

FF05 ejecutar paso a paso y el registro de L será decrementado y puesto que el registro L ha sido decrementado a 0 la bandera cero ha sido establecida.

FF06 ejecutar paso a paso y debido a que la bandera cero es 1 el registro PC será cargado con FF09 que es la dirección de la instrucción siguiendo a la instrucción JNZ FF01.

FF09 fin del programa. Apretar "enter" y se verá que el byte en FF03 ha sido copiado a la dirección de memoria FF0A; al presionar de nuevo "enter" y se verá que el byte en FF04 ha sido copiado a la dirección de memoria FF0B.

Apretar "reset", cargar BC con FF01 y cargar DE con FF0A como antes, pero esta vez cargar HL con 0009 y correr el programa. Al ejecutar el programa hará otra copia del programa comenzando desde la dirección FF0A. Examinar la memoria desde FF0A a FF12 y se verá que concuerda con la memoria desde FF01 a FF09. Se puede experimentar también con valores diferentes para BC, DE y L, pero se debe tener cuidado en la elección de los valores para DE y L porque ciertos valores causarán que el programa se destruya el mismo escribiendo datos en las direcciones FF01 a FF09.

Este programa usa lo que es llamado "Repetición Condicional". La repetición condicional ocasiona que una instrucción o grupo de instrucciones sean repetidas mientras cierta condición sea verdadera. Las instrucciones que son repetidas en este programa son LDAX B, STAX D, INX B, INX D y DCR L y la condición

que debe ser verdadera es que el registro L no debe ser el cero. Como se pudo notar que cuando el programa fue ejecutado, el registro L era 2 y las instrucciones fueron ejecutadas 2 veces. Así que se ve que el valor de L determina el número de veces que las instrucciones se ejecutaron. La manera en que esto trabaja es cuando la instrucción DCR L decrementa el registro L y establece la bandera cero si el registro L es cero y JNZ FF01 saltara a la primera instrucción mientras el registro L no es 0. Si el registro L es 0 antes de que la instrucción DCR L sea ejecutada, esto último causará que el registro L sea cambiado a FF hex y las instrucciones serán repetidas 256 veces.

Anexo 8.9 Guía de Practica N° 9

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 9

Usando variables

OBJETIVO: Asimilar los conocimientos acerca de las variables y su uso.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO

Nota:(Probar este ejercicio solo en el Primer TRAINER no en el simulador).

Instrucciones Utilizadas.

SHLD <addr> op code = 22

Almacena el registro L en la dirección de memoria <addr> y almacena el registro H en la dirección de memoria <addr + 1>. El primer byte después del código de operación es el byte de menor orden de <addr> y el segundo byte después del código de operación es el byte de orden superior de la dirección. Ninguna bandera es afectada.

LHLD <addr> op code = 2A

Carga el registro L con los datos en la dirección de memoria <addr> y carga el registro H con los datos a la dirección de memoria < addr + 1 >.

El primer byte después del código de operación es el byte de orden inferior de

<addr> y el segundo byte después del código de operación es el byte de orden superior de la dirección. Ninguna bandera es afectada.

STA <addr> op code = 32

Almacena el registro A en la dirección de memoria <addr>. El primer byte después del código de operación es el byte de orden inferior y el segundo byte después del código de operación es el byte de orden superior de la dirección. Ninguna bandera es afectada.

LDA <addr> op code = 3A

Copia el valor de la dirección de memoria <addr> a un registro. El primer byte después del código de operación es el byte de orden inferior de <addr> y el segundo byte después del código de operación es el byte de orden superior de la dirección. Ninguna bandera es afectada.

RLC op code = 07

Cambia cada bit a la izquierda y la copia el bit cambiado del bit 7 en el bit 0 y la bandera de acarreo. Sólo la bandera de acarreo es afectada.

Las variables son las áreas de la RAM que el programador intenta cambiar durante la ejecución de un programa. En cambio, los constantes son los valores que un programador no desea cambiar. Por ejemplo, si se quiere cargar un registro con una constante de 8 bits, usar la instrucción MVI A, <byte>, pero si se quiere cargar con una variable de 8 bits, usar la instrucción LDA <addr>. Similarmente, cuando quiera cargar el par de registros HL con una constante de 16 bits constante, usar la instrucción LXI H, <addr>, o si se quiere cargar con una variable de 16 bits, usar la instrucción LHLD <addr>.

Para almacenar un valor en una variable de 8 bits, se puede usar la instrucción STA <addr> y para almacenar una variable de 16 bits, se puede usar SHLD.

Cargar el siguiente programa en el TRAINER siguiendo las instrucciones.

Dirección	Datos	Instrucción
FF01	21	LXI H,8000
FF02	00	
FF03	80	
FF04	11	LXI D, FF00
FF05	00	
FF06	FF	
FF07	3E	MVI A,1
FF08	01	
FF09	32	STA FF24
FF0A	24	
FF0B	FF	
FF0C	19	DAD D
FF0D	22	SHLD FF22
FF0E	22	
FF0F	FF	
FF10	2B	DCX H
FF11	7C	MOV A,H
FF12	B5	ORA L
FF13	C2	JNZ FF10
FF14	10	
FF15	FF	
FF16	3A	LDA FF24
FF17	24	
FF18	FF	
FF19	D3	OUT 11
FF1A	11	
FF1B	07	RLC
FF1C	2A	LHLD FF22
FF1D	22	
FF1E	FF	
FF1F	C3	JMP FF09

FF20	09	
FF21	FF	
FF22	FF	(byte menos significativo dlaytm)
FF23	FF	(byte más significativo dlaytem)
FF24	FF	(leddta)

Entrar el programa en memoria y hacer lo siguiente:

PC ACTUAL

FF01 ejecutar paso a paso una vez y HL será cargado con la constante 8000h.

FF04 ejecutar paso a paso de nuevo y DE será cargado con -100h (FF00h) así la substracción se puede ejecutar al usar DAD D.

FF07 ejecutar paso a paso y el registro A será 1.

FF09 ejecutar paso a paso y el registro A será copiado a un espacio variable en FF24.

FF0C cargar el registro PC con FF24 y se verá que los datos que estaban allí antes (FF) han sido cambiados a 01. Restaurar el valor original del registro PC cargándolo con FF0C. Ejecutar paso a paso y DE será añadido a HL el cual cambiara a HL a 7F00h que es el mismo resultado que ocurre si 100h es sido substraído de él.

FF0D ejecutar paso a paso y HL será guardado en el espacio variable de 16 bits. Cargar el registro PC con FF22 y se verá que el registro L ha sido almacenado en FF22. Presionar enter y se verá que el registro H ha sido almacenado en FF23. Cargar el registro PC con FF10.

FF10 poner un punto de ruptura a FF16 y correr el programa desde la dirección (FF10) actual. Esto es hecho porque la instrucción de esta dirección y las 3 que siguen serán repetidas hasta que el registro HL sea igual a 0, que en este caso será 32, 512 veces.

FF16 examinar el registro A luego ejecutar paso a paso y examinarlo de nuevo. Se verá que ha sido cargado con el valor que fue guardado en FF24 antes.

FF19 ejecutar paso a paso dos veces y el registro A será enviado a las salidas de la salida digital de LEDs y luego su valor será rotado a la izquierda.

FF1C examina el par de registros HL y luego ejecutar paso a paso y lo examinar de nuevo. HL se ha cargado ahora con la variable que fue almacenada antes usando SHLD FF22.

FF1F correr el programa desde aquí para ver lo que el programa hace.

Este programa ha demostrado la utilidad de las variables para quitar las limitaciones causadas por tener sólo unos cuantos registros. En la manera en que se escribe programas más largos se puede encontrar que el uso de variables es una necesidad absoluta. La última página del listado de lenguaje ensamblador de MOS muestra las variables de sistema usadas de MOS. (nota: Este programa solo puede ser utilizado a través del TRAINER)

Anexo 8.10 Guía de Practica N° 10

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 10

El Registro de Pila e Instrucciones Relacionadas.

OBJETIVO: Conocer la utilización del registro de pila, así como las instrucciones que están estrechamente relacionadas a este.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

PRIMERA PARTE: EL REGISTRO DE PILA E INSTRUCCIONES RELACIONADAS

Instrucciones Utilizadas.

PUSH PSW op code= F5

Pone la palabra de estado del procesador (un registro y registro de bandera) en la pila. Ninguna bandera es afectada.

PUSH B op code = C5

Pone el par de registros BC en la pila. Ninguna bandera es afectada.

POP PSW op code= F1

Obtiene la palabra de estado del procesador (un registro y registro de bandera)

desde la pila. Las banderas Z, S, P, CY y AC son afectadas.

POP B op code= C1

Obtiene el par de registros BC desde la pila. Ninguna bandera es afectada.

El microprocesador 8085 permite escoger un área de la RAM para ser una sección de almacenamiento temporal llamada pila. El elemento clave de la pila es el registro apuntador de pila interno del 8085 (SP). El SP puede estar establecido en el principio de cualquier dirección en memoria, pero es mejor dejarlo igual o menor a la dirección que es seleccionada por el MOS.

La memoria es frecuentemente mostrada con las direcciones inferiores estando en la parte superior del listado, y las direcciones más altas siendo la parte inferior como en el ejemplo de abajo:

Dirección	Datos	Instrucción
	Este no es un programa funcional	
FF01	F3	DI
FF02	21	LXI H, FF20
FF03	20	
FF04	FF	
FF05	22	SHLD FFE9
FF06	E9	
FF07	FF	
:		
:	(Direcciones desde la FF08 a FFCF)	
:		
FFD0	40	
FFD1	23	
FFD2	61	
FFD3	03	
SP →	FFD4	00

Por ejemplo, si SP = FFD4 (como en el ejemplo anterior) y BC = 1234 y una instrucción PUSH B es ejecutada, el registro B es copiado a la dirección de memoria SP-1 (FFD3) y el registro C es copiado a la dirección de memoria SP - 2 (FFD2), luego 2 es substraído de SP de modo que la pila ahora mira así:

	Dirección de Pila	Dirección de Datos	Descripción
			Este no es un programa funcional
			:
			:
	FFD0	40	
	FFD1	23	
SP →	FFD2	34	VALOR DEL REGISTRO C
	FFD3	12	VALOR DEL REGISTRO D
	FFD4	00	

Si el registro A es 56 y las banderas son 78 y una instrucción PUSH PSW es ejecutada entonces la pila se parecerá a esto:

	Dirección de Pila	Dirección de Datos	Descripción
			Este no es un programa funcional
			:
			:
SP →	FFD0	78	VALOR DE LAS BANDERAS
	FFD1	56	VALOR DEL REGISTRO A
	FFD2	34	VALOR DEL REGISTRO C
	FFD3	12	VALOR DEL REGISTRO D
	FFD4	11	

Se puede ver ahora donde viene el nombre de "pila", los datos son "apilados" en memoria con cada instrucción PUSH. Puesto que la pila crece hacia abajo en la memoria es posible que eventualmente se sobrescriba el programa, así que se debe tener cuidado al usar la pila.

La instrucción POP trabaja al opuesto de la instrucción PUSH. Por ejemplo, si una instrucción POP H ha sido ejecutada después de PUSH PSW en el ejemplo previo, entonces los datos en la dirección de memoria SP serán copiados en el registro L y los datos en la dirección de memoria SP + 1 serian copiados en el registro H entonces 2 pueda ser añadido a SP. En otros términos, los datos que estaban en PSW cuando fue empujado a la pila serán copiados en el par de registros HL. Similarmente si PUSH BC ha sido ejecutado seguido por POP HL, entonces B podría ser copiado a H y C podría ser copiada a L.

Cargar el siguiente programa en el TRAINER siguiendo las instrucciones.

	Dirección	Datos	Instrucción
PC →	FFC7	F5	PUSH PSW
	FFC8	C5	PUSH B
	FFC9	F1	POP PSW
	FFCA	C1	POP B
	FFCB	00	
	FFCC	00	
	FFCD	00	
	FFCE	00	
	FFCF	00	
	FFD0	00	
	FFD1	00	
	FFD2	00	
	FFD3	00	
SP →	FFD4	00	(el registro de pila esta aquí)

Cargar el contador de programa con FFC7 de nuevo y cargar al registro A con 01, los registros de bandera con 02, cargar BC con 0304 y hacer lo siguiente:

1) Examinar el registro SP que debe ser FFD4.

2) Examinar los contenidos de la pila y deben ser 0000. Esto es hecho la misma manera como se examina al registro A. Apretar el botón "func" y luego la tecla "9/S.C." y los contenidos de pila serán mostrados en la pantalla "ADDRESS/REGISTER PAIR". Recordar que S.C. no es un registro interno del 8085, es sólo una manera que el MOS provee para observar los dos bytes que están en la parte superior de la pila.

3) Ejecutar paso a paso una vez en la dirección FFC7. Esto empuja a PSW en la pila (al registro A y las banderas).

4) Si se examina SP entonces se encontrará que es ahora FFD2 que es 2 menos que antes.

5) Los contenidos de pila ahora serán 0102 que son el valor de PSW que fueron empujado en la pila. Examinar los contenidos de pila para verificar que esto.

El listado de abajo muestra que es lo que en la pila se ve ahora y el registro PC y los registros SP son mostrados apuntando a sus respectivas direcciones de memoria.

	Dirección	Datos	Descripción
	FFC7	F5	PUSH PSW
PC →	FFC8	C5	PUSH B
	FFC9	F1	POP PSW
	FFCA	C1	POP B
	FFCB	XX	(valor indefinido)
	FFCC	XX	(valor indefinido)
	FFCD	XX	(valor indefinido)
	FFCE	XX	(valor indefinido)
	FFCF	XX	(valor indefinido)
	FFD0	XX	(valor indefinido)
	FFD1	XX	(valor indefinido)
SP →	FFD2	02	Valor de las banderas
	FFD3	01	Valor del registro A
	FFD4	00	

- 7) Ejecutar la instrucción PUSH B por ejecución paso a paso una sola vez.
- 8) Si se examina el SP entonces se encontrará que es ahora FFD0 que es 2 menos que antes y los contenidos de pila serán 0304 que son el valor empujado en la pila por la instrucción PUSH B.
- 9) Al examinar la memoria desde FFD0 a FFD4 verá que los datos son como el listado de abajo. Retornar PC al valor de FFC9.

	Dirección	Datos	Descripción	
	FFC7	F5	PUSH PSW	
	FFC8	C5	PUSH B	
PC →	FFC9	F1	POP PSW	
	FFCA	C1	POP B	
	FFCB	XX	(valor indefinido)	
	FFCC	XX	(valor indefinido)	
	FFCD	XX	(valor indefinido)	(ver nota de abajo)
	FFCE	XX	(valor indefinido)	
	FFCF	XX	(valor indefinido)	
SP →	FFD0	04	Valor del registro C	
	FFD1	03	Valor del registro B	
	FFD2	02	Valor de las banderas	
	FFD3	01	Valor del registro A	
	FFD4	00		

10) Ejecutar paso a paso el POP PSW y observar que SP = FFD2 que ahora es 2 más que antes, luego examinar el registro A.F. (el registro A y registro de bandera) y se verá que es ahora es el valor que fue puesto en la pila por la instrucción PUSH B.

11) Ejecutar paso a paso POP B y se verá que SP regreso al valor de nuevo al que tuvo al comienzo del programa (FFD4) y que el par de registros BC es ahora el valor que fue empujado en la pila por la instrucción de PUSH PSW.

	Dirección	Datos	Descripción
	FFC7	F5	PUSH PSW
	FFC8	C5	PUSH B
	FFC9	F1	POP PSW
	FFCA	C1	POP B
PC →	FFCB	XX	(valor indefinido)
	FFCC	XX	(valor indefinido)
	FFCD	XX	(valor indefinido)
	FFCE	XX	(valor indefinido)
	FFCF	XX	(valor indefinido) (ver nota de abajo)
	FFD0	XX	(valor indefinido)
	FFD1	XX	(valor indefinido)
	FFD2	XX	(valor indefinido)
	FFD3	XX	(valor indefinido)
SP →	FFD4	00	

Nota: Los "valores indefinidos" mostrados en el listado de lenguaje de máquina son los valores que MOS empuja y hace saltar de la pila en cada punto de ruptura o ejecución paso a paso. Esto es hecho sin afectar los valores importantes que el programa almacena en la pila, pero esto afecta a la memoria de pila no utilizada. Sin embargo MOS regresa con el apuntador de pila al mismo lugar, algunos de los bytes debajo de la dirección del apuntador de pila se cambiarán debido a los empujes. Si fuera posible correr el programa a toda velocidad y pararlo en FFCB sin ejecución paso a paso o puntos de ruptura, los valores de memoria podrían ser definidos como sigue:

Dirección	Datos	Descripción
FFC7	F5	PUSH PSW
FFC8	C5	PUSH B
FFC9	F1	POP PSW

	FFCA	C1	POP B
PC →	FFCB	00	
	FFCC	00	
	FFCD	00	
	FFCE	00	
	FFCF	00	
	FFD0	04	Valor del registro C
	FFD1	03	Valor del registro B
	FFD2	02	Valor de las banderas
	FFD3	01	Valor del registro A
SP →	FFD4	00	

Desafortunadamente no hay manera para parar el programa y ver la memoria sin poner un punto de ruptura o usando RST 7 así esto podría ser demostrado.

La pila tiene otros propósitos aparte de almacenar los registros de propósito general. Esto se usa también por la instrucción RST 7 que fue discutida en un laboratorio anterior y las instrucciones de XTHL, CALL y RET que se discutirán más tarde.

SEGUNDA PARTE: USANDO LA INSTRUCCIÓN XTHL.

Instrucciones Utilizadas.

XTHL op code = E3

Cambia a L con el valor en la dirección de memoria SP y cambia H con la dirección de memoria SP + 1. Ninguna bandera es afectada.

PUSH H op code = C5

Poner el par de registros HL en la pila. Ninguna bandera es afectada.

POP H op code = C1

Obtiene el par de registros HL desde la pila. Ninguna bandera es afectada.

MOV A,M op code = 7E

Copia el byte del registro A hacia la ubicación de memoria especificada por el par de registros HL. Ninguna bandera es afectada.

MOV M, A op code = 77

Copia el byte del registro A hacia la ubicación de memoria especificada por el par de registros HL. Ninguna bandera es afectada.

DCR D op code = 15

Decrementa el registro D. Las banderas Z, S, P y AC son afectadas.

Uno de los mejores usos para la instrucción XTHL está en programas donde existe una necesidad de más registros. Se puede cargar el registro HL con un valor y empujarlo en la pila (llamar a este valor HL2) y luego cargar HL con otro valor (llamar a este valor HL1). Después de esto, una instrucción XTHL guardara HL1 en la pila y luego cargar HL con HL2 permitiendo que HL2 se accedido. Si otra instrucción XTHL es ejecutada la pila tendrá HL2 de nuevo y HL igualará HL1 que permitirá a HL1 ser accedido.

Cargar el siguiente programa en el TRAINER siguiendo las instrucciones.

Dirección	Datos	Instrucción
FF01	E5	PUSH H
FF02	21	LXI H, 0000
FF03	00	
FF04	00	
FF05	44	MOV B,H
FF06	4D	MOV C,L
FF07	19	DAD D
FF08	D2	JNC FF0C
FF09	0C	
FF0A	FF	

FF0B	03	INX B
FF0C	E3	XTHL
FF0D	2B	DCX H
FF0E	7C	MOV A,H
FF0F	B5	ORA L
FF10	E3	XTHL
FF11	C2	JNZ FF07
FF12	07	
FF13	FF	
FF14	F1	POP PSW
FF15	FF	RST 7

Cargar el programa en memoria y apretar el botón reset. Cargar el par de registros HL con 0002 y el par de registros DE con FFFF y hacer lo siguiente:

PC ACTUAL

FF01 ejecutar paso a paso y el registro HL será empujado en la pila. Examinar los contenidos de la pila y ellos serán iguales que el registro HL. Este valor en la pila determina cuantas veces la adicción será ejecutada.

FF02 ejecutar paso a paso y HL será cargado con 0000 que es el valor de comienzo de la suma.

FF05 ejecutar paso a paso dos veces y las instrucciones MOV B, H y MOV C, L serán ejecutadas, esto hará que BC=0000 que es la palabra de orden superior del resultado.

FF07 ejecutar paso a paso y la instrucción DAD D causará que DE sea añadido a la suma que es mantenida en HL (examinar HL).

FF08 Puesto que el resultado de la instrucción DAD D es bastante pequeño para encajar en el par de registros HL, la bandera de acarreo no es establecida. Por lo tanto la instrucción JNC FF0C será ejecutada de tal manera que salte por encima de la instrucción INX B. ejecutar paso a paso y PC será FF0C, que es la dirección

de la instrucción siguiente a INX B.

FF0C Examinar los contenidos de la pila y el registro HL luego ejecutar paso a paso. Examinar los contenidos de la pila y el registro HL de nuevo y se verá que la instrucción XTHL ha cambiado sus valores; HL es ahora el número de adiciones que quedan por ejecutar, y el valor en la pila tiene la suma de la adición.

FF0D ejecutar paso a paso y 1 será substraído de HL que indica que ahora existe 1 adición que falta por ejecutar.

FF0E la instrucción en esta dirección y la dirección siguiente son usadas para determinar si HL es 0000. Estas dos instrucciones son necesarias porque la instrucción DCX H no afecta ninguna de las banderas. Ejecutar paso a paso y el registro H será copiado al registro A.

FF0F Ejecutar paso a paso de nuevo y al registro A se le ejecutara una operación OR con el registro L sin cambiar el registro L. Recordar que por definición cuando lógicamente se ejecuta una operación OR en bits, un cero resultara solo si los bits a los cuales se les ha ejecutado la operación OR son ambos 0. Así si el registro A y el L son ambos 0 entonces la instrucción ORA L causará que la bandera cero sea 1, de otra manera la bandera cero será 0. En este caso el valor de L tiene 1 y A es 0 así la bandera cero será 0.

FF10 ejecutar paso a paso y la instrucción XTHL intercambiara el valor en la pila con el valor en HL. Esto hace a HL la suma de nuevo y el valor en la pila es el número de las adiciones que falta por ejecutar.

FF11 Ejecutar paso a paso y la instrucción JNZ será ejecutada que hará que PC retorne a FF07. Hacer notar que la instrucción XTHL ha sido ejecutada después la instrucción ORA L que determina si el par de registro HL es 0000. Puesto que XTHL no afecta el registro de bandera, las banderas que fueron afectadas por la instrucción ORA L son iguales que antes que se ejecutara XTHL.

FF07 Ejecutar paso a paso y la instrucción DAD D será ejecutada que añadirá DE a la suma en HL de nuevo.

FF08 Esta vez el resultado de la adición no podría encajar en el par de registros HL, así que la bandera de acarreo es establecida. Ejecutar paso a paso y la instrucción JNC FF0C no se ejecutará debido a que la bandera de acarreo es 1, en su lugar PC cambiara a la dirección de la instrucción siguiente JNC FF0C.

FF0B ejecutar paso a paso y 1 será añadido al par de registros BC. Esto sucederá en este programa cada vez la bandera de acarreo es establecida por la instrucción previa DAD D.

FF0C ejecutar paso a paso la instrucción XTHL y HL es ahora el número de adiciones que quedan para hacer, y el valor en la pila tiene la suma de la adición.

FF0D ejecutar paso a paso y 1 será substraído del número de adiciones que se deben hacer en el registro HL.

FF0E Ejecutar paso a paso y el registro H será copiado al registro A.

FF0F Ejecutar paso a paso y el registro L se le ejecutara una operación OR con el registro A y debido a que ambos registros son 0 el resultado en el registro A será 0 y la bandera de cero será establecida.

FF10 Ejecutar paso a paso y la instrucción XTHL intercambiara el valor en la pila con el valor en HL. Esto hace a HL la suma de nuevo y el valor en la pila es el número de las adiciones falta por ejecutar (0000).

FF11 Ejecutar paso a paso y el registro PC será cargado con FF14 que es la dirección de la instrucción siguiendo la instrucción JNZ FF07. Esto es porque la bandera cero debe ser 0 para que la instrucción cargue al registro PC con FF07.

FF14 examinar el registro SP luego ejecutar paso a paso y examinarlo de nuevo. Antes de ejecutarlos paso a paso, SP era FFD2 y después de ejecutarlo paso a paso es FFD4, que es el valor que tuvo cuando el programa empezó. La instrucción POP PSW fue usada únicamente para restaurar la pila a su posición de inicio original, el valor que fue cargado en al registro A y registro de bandera no se necesitaba.

FF15 Fin del programa. Examinar los pares de registros BC y HL y deben tener 0001 y FFFE respectivamente, que representa el número 0001FFFE que es el producto de FFFF y 0002.

Probar otros valores para DE y HL (HL debe ser mayor que 0). Tomar nota puesto que este programa usa adiciones repetitivas para realizar multiplicación entre más grande es el valor de HL más tiempo el programa tomará para ejecutar la multiplicación. Por ejemplo, si $HL = FFFF$ el programa tomará aproximadamente 3 segundos para ejecutar la multiplicación.

Anexo 8.11 Guía de Practica N° 11

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 11

Subrutinas

OBJETIVO: Aprender todo lo relacionado a las subrutinas, uso, instrucciones y ejemplos de programas.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

Instrucciones Utilizadas.

CALL <addr> op code = C3

Pone la dirección de la próxima instrucción en la pila y cargar el contador de programa con los dos bytes que sigue al código de operación.

El byte inmediato que sigue al código de operación es el byte menos significativo de la dirección y el otro después de ese es el byte más significativo.

RET op code = C9

Quita una dirección de 16 bits de la pila y lo carga en el contador de programa. Ninguna bandera es afectada.

DCR H op code = 25

Decrementa al registro H. Las banderas Z, S, P y AC son afectadas.

PUSH B op code = C5

Pone al par de registros BC en la pila. Ninguna bandera es afectada.

Las instrucciones CALL y RET permiten la posibilidad de subrutinas. Una subrutina es una parte de un programa que ejecuta una función que es necesitada en muchas partes diferentes del programa principal. La instrucción CALL salta a una subrutina después de poner en la pila la dirección de la instrucción que está siguiendo a CALL. La última instrucción ejecutada de la subrutina será RET que causará que el programa retorne a la dirección que está en la pila. Más adelante en este manual hay ciertos programas de ejemplo que usan subrutinas que son fabricadas en el MOS.

Cargar el siguiente programa en el TRAINER siguiendo las instrucciones.

Dirección	Datos	Instrucción
FF01	21	LXI H, 8000
FF02	00	
FF03	80	
FF04	3E	MVI A,0F
FF05	0F	
FF06	CD	CALL F16
FF07	16	DAD D
FF08	FF	
FF09	2F	CMA
FF0A	CD	CALL F16
FF0B	16	
FF0C	FF	
FF0D	3E	MVI A,55
FF0E	55	
FF0F	CD	CALL F16

FF10	16	
FF11	FF	
FF12	25	DCR H
FF13	C3	JMP FF04
FF14	04	
FF15	FF	
FF16	E5	PUSH H
FF17	F5	PUSH PSW
FF18	D3	OUT 11
FF19	11	
FF1A	2B	DCX H
FF1B	7C	MOV A,H
FF1C	B5	ORA L
FF1D	C2	JNZ FF1A
FF1E	1A	
FF1F	FF	
FF20	F1	PUSH PSW
FF21	E1	POP H
FF22	C9	RET

Cargar el programa en memoria y correrlo; el programa debe mostrar 3 diferentes patrones en los LEDs discretos, uno tras otro, y repite los patrones de nuevo. El retardo entre cada patrón debe disminuir lentamente hasta que los patrones cambien tan rápidamente que no sean notorios.

Luego el retardo se volverá repentinamente mucho más largo y entonces lentamente disminuirá de nuevo. Si el programa está trabajando como describe, apretar el botón reset y hacer lo siguiente:

PC ACTUAL

FF01 Ejecutar paso a paso la instrucción CALL en la dirección FF06 y examinar el contenido de la pila (será 0000) y el apuntador de pila será FFD4

FF06 ejecutar paso a paso y la instrucción CALL FF16 será ejecutada causando que el programa salte a FF16, el principio de la subrutina.

FF16 La instrucción CALL ha puesto la dirección de la instrucción siguiéndola, en la pila. Examinar los contenidos de pila para verificar esto.

Examinar y apuntar los valores de PSW y HL luego ejecutar paso a paso dos veces que primero empujara al par de registros HL luego empujara a PSW en la pila. Poner un punto de ruptura a FF20, luego correr desde la dirección actual.

FF20 Esta es la primera instrucción siguiente al loop de retraso. Comparar los PSW y HL con los valores que apuntó y se verá que han sido cambiados. Ejecutar la instrucción POP PSW paso a paso y el valor original de PSW se restaurará.

FF21 Ejecutar paso a paso de nuevo y la instrucción POP H será ejecutada la cual restaurara el valor original de HL. Notar que el orden de meter (POP) datos de la pila es el opuesto del orden de empujar (PUSH). Esto es porque el último ítem empujado en la pila es el primero que puede ser metido fuera. Metiendo a PSW y HL fuera de la pila ha restaurado la dirección de retorno (FF09) a la parte superior de la pila. Examinar los contenidos de pila para verificar esto.

FF22 ejecutar la instrucción RET paso a paso y ello quitará la dirección de retorno de la pila y saltara a la dirección.

Los contenidos de pila serán 0000 de nuevo con el apuntador de pila a FFD4 después de la instrucción RET.

FF09 Ejecutar paso a paso y el registro A se complementará.

FF0A Ahora que se comprende lo que sucede durante CALL de una subrutina, verificar el contenido de pila para verificar que es 0000. Poner un punto de ruptura a la dirección (FF0D) de la instrucción siguiente a CALL y correr desde la dirección actual.

FF0D La subrutina tiene como la salida el patrón de bits que estaba en el registro A y la pausa de un periodo de tiempo proporcional al valor de HL y retorna a esta

dirección.

Como se puede ver, un CALL de una subrutina actúa como un código de operación definido por un usuario. Una subrutina puede ejecutar una tarea sin cambiar los registros, como en el programa de arriba, o puede cambiar los registros según las reglas que se definen al escribir la subrutina. Por ejemplo, se pudo escribir una subrutina que tome el valor del registro B y lo multiplique por el registro C regrese el resultado en los registros BC. Los programadores normalmente tiene una librería de subrutinas útiles así puedan ser incluidas en sus programas.

Las subrutinas pueden hacer también CALL a subrutinas, y esas subrutinas pueda llamar otros y así sucesivamente. El único límite es la cantidad de memoria que la pila puede usar.

Anexo 8.12 Guía de Practica N° 12

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 12

Uso de Subrutinas del Sistema Operativo MOS

OBJETIVO: Conocer algunas subrutinas que posee el sistema operativo MOS y la funcionalidad de cada una de estas subrutinas para realizar programas aplicados a estos servicios.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

Nota:(Probar esta guía solo en el Primer TRAINER no en el simulador).

PRIMERA PARTE 1: USANDO LAS SUBRUTINAS DE SISTEMA OPERATIVO MONITOR

Esta lección ilustra el uso de una de las subrutinas del Sistema Operativo Monitor. Dichas subrutinas (también llamadas servicios) que se encuentran en la memoria ROM pueden ser ejecutadas al cargar el registro C con el Numero de Servicio y ejecutando un CALL 1000H. También se puede realizar sin escribir un programa o sin afectar al registro PC, al cargar los registros apropiados y presionando "func", luego "2". Todos los registros que no son utilizados como 'entrada' o 'salida' del servicio no serán afectados. Por favor, referirse el apéndice F para obtener la descripción de los servicios de MOS.

Un ejemplo de programa utilizando el servicio MOS:

El siguiente ejemplo muestra el número hexadecimal 1234 en los 4 displays de la izquierda:

```
mos    equ 1000h    ; dirección de comienzo de programa
ledhex equ 12h      ; pone a hl en la pila. Este valor es el numero de
                    ; adiciones
.org 0ff01h        ; el programa empieza en esta dirección
mvi     ; ledhex rutina de servicio
c,ledhex
lxi d,1234h       ; carga DE con 1234 hex
call mos           ; llama a MOS para el servicio ledhex
wait:   jmp wait   ; repite hasta que reset es presionado
end      ; así los displays no serán borrados. Normalmente
        ; los programas terminan con rst 7
```

Dirección	Datos	Instrucción
FF01	0E	; MVI C,12
FF02	12	
FF03	11	; LXI C,1234
FF04	34	
FF05	12	
FF06	CD	; CALL 1000
FF07	00	
FF08	10	
FF09	C3	; JMP 0FF09
FF0A	09	
FF0B	FF	

Cargar el programa en memoria, presionar el botón 'reset' y luego ejecutarlo. Se observara el "1234" en la pantalla "ADDRES/REGISTER PAIR" y continuara

mostrándolo hasta presionar el botón reset. Presionar el botón reset y ejecutar paso a paso hasta FF06 el cual representa la dirección de la instrucción CALL 1000. Ejecutar paso a paso otra vez y se observara que en vez de que el programa se detenga en la instrucción de la subrutina que ha sido LLAMADA como en la lección previa, el programa se detendrá en la instrucción que le sigue a la instrucción CALL 1000. La razón de esta inconsistencia se debe a que el MOS comprueba si CALL se detuvo en la memoria ROM o RAM. Si lo hace en la ROM, ejecutara la subrutina a toda velocidad hasta que finalice. Esto se debe a que el método de ejecución paso a paso usado por el PRIMER requiere que las instrucciones sean ejecutadas paso a paso para estar en la RAM. También se demostrara que pueden no observarse los números "1234" cuando la subrutina sea ejecutada. Los números fueron en realidad mostrados, pero tan pronto como ellos fueron mostrados, el MOS retornó al modo de entrada de datos y fueron sobrescritos. Esto ocurre cada vez que se ejecuta paso a paso un servicio que usa las pantallas digitales. Almacena un 0 en FF02 así después que la primera instrucción ha sido ejecutada, el registro C será cargado con 0 en lugar de 12h. Esta vez cuando se corre el servicio de programa 0 será ejecutado en lugar del servicio 12.

Para más los ejemplos de programas que usen servicios MOS, ver las dos lecciones que siguen las cuales usan los servicios ADCIN y PITCH. La última lección usa los servicios KEYIN y LEDHEX.

SEGUNDA PARTE: USANDO PITCH (SERVICIO 10)

NUEVAS INSTRUCCIONES

RRC op code = 0F

Cambia los bits en el registro A hacia la derecha y pone el valor cambiado del bit 0 en la bandera de acarreo y el bit 7. Sólo la bandera de acarreo es afectada.

Este programa traduce el valor del interruptor DIP a un valor de 14 bits que está guardado en el par de registros DE. El valor es pasado al servicio 10 que produce un valor de frecuencia en la bocina inversamente proporcional al valor de DE.

```

dips equ 12h      ; puerto para lectura de los interruptores DIP
mos  equ 1000h   ; dirección de los servicios MOS
      .org 0ff01h
loop: in dips     ; toma el valor del interruptor DIP
      rrc        ; rota el registro A hacia la derecha
      rrc        ; 2 veces puesto que dos bits superiores no son
                  usados
      mov e,a    ; almacena en E temporalmente
      ani       ; enmascara dos bits a 0
      00111111b
      mov d,a    ; guarda en D
      mov a,e    ; toma E de nuevo
      ani       ; enmascara a los 6 bits inferiores a 0
      11000000b
      mov e,a    ; guarda en E
      mvi c,10h ; servicio 10
      call mos   ; genera un campo de acuerdo a DE
      jmp loop   ; lo hace de nuevo
end

```

Dirección	Datos	Instrucción
FF01	DB	; IN 12
FF02	12	
FF03	0F	; RRC
FF04	0F	; RRC
FF05	5F	; MOV E,A
FF06	E6	; ANI EFH
FF07	3F	
FF08	57	; MOV D,A
FF09	7B	; MOV A,E
FF0A	E6	; ANI 0C0H

FF0B	C0	
FF0C	5F	; MOV E,A
FF0D	0E	; MVI C,10H
FF0E	10	
FF0F	CD	; CALL 1000H
FF10	00	
FF11	10	
FF12	C3	; JMP FF01
FF13	01	
FF14	FF	

1) Cambie los 4 interruptores a la izquierda a "encendido" y los 4 a la derecha, a "apagado". Cuando esto se haya hecho, los datos que son introducidos por medio de los interruptores DIP serán 0F en hexadecimal.

2) Apretar reset y cargar el programa en memoria y verificar que fueron cargados correctamente.

3) Ejecutar paso a paso una vez y examinar el registro A para asegurarse es 0F, si no es, hacer el paso 1 de nuevo.

Con el fin de que los interruptores DIP den el rango más amplio de frecuencias el valor de 8 bits de los interruptores DIP debe ser hecho los 8 bits más significativos del valor de 14 bits requerido para generar la frecuencia. El gráfico de abajo muestra como se hace esto. Este gráfico muestra los códigos de operación que son usados para convertir el valor de los interruptores DIP a un valor almacenado en el par de registros DE. A la derecha de los códigos de operación están los valores binarios que serán los registros A, D y E después que el código de operación es ejecutado. En el gráfico, las "X" dicen que el valor no ha sido definido y que son los valores desconocidos.

Código de Operación	Valores de los Registros después de Ejecución de op code		
	A	D	E
IN 12h	00001111b	XXXXXXXXXb	XXXXXXXXXb
RRC	10000111b	XXXXXXXXXb	XXXXXXXXXb
RRC	11000011b	XXXXXXXXXb	XXXXXXXXXb
MOV E,A	11000011b	XXXXXXXXXb	11000011b
ANI 00111111b	00000011b	XXXXXXXXXb	11000011b
MOV D,A	00000011b	00000011b	11000011b
MOV A,E	11000011b	00000011b	11000011b
ANI 11000000b	11000000b	00000011b	11000011b
MOV E,A	11000000b	00000011b	11000000b
:			
:			
:			

Ejecutar paso a paso hasta la dirección FF0D y se verá que el valor de 8 bits del interruptor DIP son ahora los 8 bits superiores del numero de 14 bits en DE. Notar que el par de registros DE es siempre 16 bits, pero acá se refiere como un número de 14 bits puesto que el servicio 10 de MOS ignora los dos bits superiores del registro DE.

	VALOR
Interruptores DIP	00001111
Registro DE	0000001111000000

Después que DE ha sido cargado con el valor de los interruptores DIP, el registro C es cargado con 10h que es el número del servicio y luego un CALL 1000h sea hecho el cual producirá la frecuencia de la bocina. Después del CALL hay un JMP para comenzar el programa de nuevo y continuará repitiendo indefinidamente.

Apretar el botón reset y correr el programa. Mover los interruptores DIP y se oirá el cambio de frecuencia de en la bocina.

TERCERA PARTE: USANDO ADCIN (SERVICIO 9)

NUEVAS INSTRUCCIONES

STC op code = 37

Induce a la bandera de acarreo a 1. Sólo la bandera de acarreo es afectada

JZ <addr> op code = CA

Si la bandera Z es 1 empieza a ejecutar las instrucciones en la dirección de 2 bytes siguiendo la instrucción, de lo contrario empezar a ejecutar la instrucción después de esta. El primer byte siguiente al código de operación es el byte menos significativo de la dirección, el segundo byte es el byte más significativo de la dirección. Ninguna bandera es afectada.

Esta lección tiene como propósito para usarse por aquellos con experiencia en la electrónica básica. Por favor, se debe proceder a la lección próxima si no se tiene entrenamiento de electrónica básica.

Este programa leerá el convertidor de analógico a digital y luego dividirá el número por 8 y producirá una gráfica de barra en la salida digital LED que es proporcional a la entrada de voltaje analógica.

Una fuente de voltaje variable, entre el rango de 0 a +5 voltios, debe ser conectada a la entrada analógica del conector externo digital E/S CN3 (este está sobre y a la izquierda de la pantalla ADDRESS/REGISTER PAIR). Esto puede hacerse simplemente con un potenciómetro de 10K al conectar el “wiper” a la entrada analógica y las otras dos conexiones a +5V y la otra a tierra. El conector CN3 +5 voltios en el pin 22, entrada analógica en el pin 20 y tierra en pin 18.

Se debe estar seguro que no existe ningún tipo de energía al PRIMER al unir el potenciómetro al conector de E/S. También se debe estar seguro que no haya

ningún corto en las conexiones al unir el potenciómetro a CN3. Puede ser mejor para usar un conector de apareamiento, o envolver las conexiones de los cables. Una vez que las conexiones apropiadas han sido hechas y verificadas, aplicar la fuente de poder al PRIMER y cargar el programa siguiente.

```

leds      equ 11h          ; puerto de salida para la salida digital LED
mos       equ 1000h       ; dirección CALL del MOS
          .org 0ff01h     ; dirección de comienzo del programa
start:    mvi c,9         ; servicio 9
          call mos        ; tomar ADC
          mov a,l         ; colocar en registro A
          rrc             ; rota A hacia la derecha = dividido entre 2
          rrc             ; dividido entre 2
          rrc             ; dividido entre 2
          ani 00001111b   ; borra los 3 bits superiores
          jz zerout      ; si A = 0 entonces enviar hacia afuera
          mov c,a         ; c = numero de leds que se encenderán
          mvi a,0        ; A = 0
loop:     stc             ; establece el acarreo
          rar             ; y rotarlo en el registro A
          dcr c          ; decrementa el contador
          jnz loop       ; si c > 0, loop de nuevo
zerout:   out leds       ; enviar el patrón de barra a los leds
          jmp start      ; hacer todo de nuevo

```

Dirección	Datos	Instrucción
FF01	0E	MOV C,9
FF02	09	
FF03	CD	CALL 1000h
FF04	00	
FF05	10	

FF06	7D	MOV A,L
FF07	0F	RRC
FF08	0F	RRC
FF09	0F	RRC
FF0A	E6	ANI 01FH
FF0B	1F	
FF0C	CA	JZ FF18
FF0D	18	
FF0E	FF	
FF0F	4F	MOV C,A
FF10	3E	MVI A,0
FF11	00	
FF12	37	STC
FF13	1F	RAR
FF14	0E	DCR C
FF15	C2	JNZ LOOP
FF16	12	
FF17	FF	
FF18	D3	OUT 11
FF19	11	
FF1A	C3	JMP FF01
FF1B	01	
FF1C	FF	

Apretar el botón reset y correr el programa. Se verá que la grafica de barra se dibuja se muestra en la salida digital LED cambian proporcionalmente con la tensión de entrada.

Anexo 8.13 Guía de Practica N° 13

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 13 Uso de Instrucciones de Comparación

OBJETIVO: Adquirir los conocimientos sobre la forma de uso y funcionalidad de las instrucciones de comparación en los programas, utilizando dichas instrucciones en los programas a desarrollar para una mayor asimilación de los estudiantes

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

Nota: (Probar esta guía solo en el Primer TRAINER no en el simulador).

NUEVAS INSTRUCCIONES

RLC op code = 07

Cambia los bits en el registro A hacia la izquierda y pone el valor cambiado del bit 7 en la bandera de acarreo y el bit 0. Sólo la bandera de acarreo es afectada.

CMP L op code = BD

Subtrae el registro L del registro A y pone las banderas condicionales como corresponde sin cambiar el registro A. La bandera de acarreo es 1 si $A < L$ y la bandera cero es 1 si $A = L$. Las banderas Z, S, P, CY y AC son afectadas.

CPI <byte> op code = FE

Subtrae el byte siguiendo al código de operación del registro A y establece la banderas condicionales como corresponde sin cambiar el registro A. La bandera de acarreo es 1 si $A < \text{byte}$ y la bandera cero es 1 si $A = \text{byte}$. Las banderas Z, S, P, CY y AC son afectadas.

Las instrucciones de comparación pueden también ser usadas para decir si un valor es mayor o menor que otro valor. El programa siguiente compara el registro H y el registro L y causa la salida digital LED para mostrar diferentes patrones dependiendo de los tamaños relativos de H y L. Si H es mayor que L entonces los dos LED de la izquierda brillaran, si L es mayor que H entonces los dos LED derechos brillaran y si $H = L$ entonces los dos LED del medio brillaran.

```
    leds    equ 11h          ; puerto de salida para la salida digital LED
           .org 0ff01h

loop:   mov a,h             ; pone H en A
        cmp l              ; pone las banderas de acuerdo al valor de A
                               y L
        jc lgreat          ; si CY=1 entonces L>A, así que salta
        jz equal          ; si Z=1 entonces A=L, así que salta
                               ; en este punto en el programa H no es
                               ; menos que L
                               ; y no es igual a L, así que H es mayor que L
        mvi                ; patrón para H>L
        a,00111111b
        jmp patout         ; enviar a la salida el patrón

lgreat: mvi                ; patrón para L>H
        a,11111100b
        jmp patout         ; enviar a la salida el patrón

equal:  mvi                ; patrón para H=L
        a,11100111b
```

patout: out leds ; muestra el patrón de bits
rst 7 ; fin del programa

Dirección	Datos	Instrucción
FF01	7C	MOV A,H
FF02	BD	CMP L
FF03	DA	JC FF0E
FF04	0E	
FF05	FF	
FF06	CA	JZ FF13
FF07	13	
FF08	FF	
FF09	3E	MVI A,3F
FF0A	3F	
FF0B	C3	JMP FF15
FF0C	15	
FF0D	FF	
FF0E	3E	MVI A,FC
FF0F	FC	
FF10	C3	JMP FF15
FF11	15	
FF12	FF	
FF13	3E	MVI A,E7
FF14	E7	
FF15	D3	OUT 11
FF16	11	
Ff17	FF	RST 7

Cargar el programa en memoria y apretar el botón reset entonces hacer lo siguiente:

PC ACTUAL

FF01 cargar HL con FFE0 así H será mayor que L luego ejecutar paso a paso y H será copiada al registro A.

FF02 ejecutar paso a paso y A será comparado con L y las banderas serán establecidas como corresponden. Según la definición de la instrucción CMP L si A es mayor que L entonces la bandera cero y la bandera de acarreo pueden ser ambas 0 (examinar el registro de bandera para verificar esto).

FF03 ejecutar paso a paso y PC apuntará a entonces a la instrucción siguiendo la instrucción JC FF0E. Recordar que debido a que la bandera de acarreo no es 1 la instrucción JC FF0E no se ejecutará.

FF06 ejecutar paso a paso y PC apuntará a la instrucción siguiente JZ FF13 debido a que la bandera cero debe ser 1 antes que el salto ocurra.

FF09 ejecutar paso a paso y el registro A será cargado con el valor que será mostrado en la salida digital LED. Ejecutar paso a paso dos veces más y el programa saltará a la instrucción OUT 11 y ello mostrará el patrón de bits del registro A.

FF17 fin del programa.

Apretar el botón reset y cargar el par de registros HL con 22C0 así el registro H será menos que el registro L y hacer lo siguiente:

FF01 ejecutar paso a paso dos veces para copiar H en A y comparar A con L. Examinar el registro de bandera y se verá que la bandera de acarreo tiene 1 y la bandera cero es 0 porque A es menor que L.

FF03 ejecutar paso a paso y puesto que la bandera de acarreo es 1 la instrucción JNC FF0E ocurrirá.

FF0E ejecutar paso a paso tres veces y A estará cargado con el patrón de bits que indica que H es menor que L, el programa saltará a la instrucción OUT 11 que

mostrará el patrón de bits en el registro A.

FF17 fin del programa

Apretar el botón reset y cargar el par de registros HL con 1010 así H será igual a L.

FF01 ejecutar paso a paso dos veces para copiar H en A y comparar A con L de nuevo. Examinar el registro de bandera y se verá que la bandera de acarreo es 0 porque A no es menor que L y la bandera cero es 1 porque A es igual que L.

FF03 ejecutar paso a paso a la instrucción JC FF0E y no será ejecutada porque la bandera de acarreo es 0.

FF06 ejecutar paso a paso y la instrucción JZ FF15 será ejecutada porque la bandera cero es 1.

FF13 ejecutar paso a paso tres veces y A estará cargado con el patrón de bits que indica que H es igual que L, el programa saltará a la instrucción OUT 11 que mostrará el patrón de bits en el registro A.

FF17 fin del programa.

Anexo 8.14 Guía de Practica N° 14

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 14

Uso de Interrupciones

OBJETIVO: Entender la funcionalidad y el uso de las diferentes interrupciones en el microprocesador 8085, al desarrollar algunos programas.

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

Nota: (Probar esta guía solo en el Primer TRAINER no en el simulador).

NUEVAS INSTRUCCIONES

EI op code = FB

Habilita las interrupciones después que la siguiente instrucción se ejecute.
Ninguna bandera es afectada.

DI op code = F3

Inhabilita las interrupciones después que la instrucción DI haya sido ejecutada.
Ninguna bandera es afectada.

RIM op code = 20

(Ver enciclopedia de conjunto de instrucciones)

SIM op code = 30

(Ver enciclopedia de conjunto de instrucciones)

El 8085 tiene 5 pines, llamados TRAP, RST 5.5, RST 6.5, RST 7.5 e INTR, dedicados a ser fuentes de interrupciones. El programa que sigue ilustra el uso de la interrupción RST 7.5. Si un 1 aparece en el pin de RST 7.5 y la interrupción no ha sido inhabilitada (inhabilitando la interrupción es explicado más tarde) entonces el programa que está corriendo actualmente será "interrumpido" y la dirección de la instrucción que sería ejecutada como siguiente será empujada en la pila. El microprocesador luego salta a dirección 003C donde se encuentran las instrucciones de MOS que obtienen una dirección de una ubicación reservada de la RAM y luego salta a esa dirección. Esta ubicación que mantiene la dirección al que el microprocesador saltará será llamada un vector. El vector permite la ocurrencia de una interrupción para correr un programa específico que puede ser localizado en cualquier parte de la RAM. Programas de este tipo son llamados rutinas de servicios de interrupción (ISRs).

Las ISR deben siempre retornar los registros que utilizan con mismos valores que estos tuvieron cuando la interrupción ocurrió. Esto se puede hacer empujando los registros y luego metiéndolos antes de retornar al programa interrumpido. Después de haber empujado los registros, el propósito real de las ISR puede ser realizado, que en el caso de este programa, será incrementar un valor que está en la memoria. Antes de salir de la ISR, todos los registros que fueran empujados, debieron ser metidos, luego una instrucción de EI (habilitar interrupción) seguida por una instrucción RET debe ser ejecutada. La instrucción de EI es necesaria porque siempre que una interrupción ocurra, el 8085 ejecuta de forma automática un DI (inhabilitar interrupción). La instrucción de RET retornará el contador de programa a la sección del código que fue interrumpida.

Se puede ocasionar que el 8085 ignore (inhabilite) todas las interrupciones, excepto TRAP, a través de la instrucción DI. Esto es usado en el programa siguiente para mantener las interrupciones de ocurrir mientras que un vector es hecho para la interrupción RST 7.5, mientras que el contador 8155 está siendo la

establecido y mientras que la máscara de la interrupción RST 7.5 es habilitada completamente a través de la instrucción SIM.

La instrucción SIM tiene dos propósitos: para controlar el pin SOD (esto no será usado en esta lección), y para proporcionar una interrupción de control. Usar esta instrucción permite que las interrupciones RST 7.5, RST 6.5 y RST 5.5 puedan ser inhabilitadas individualmente. Antes de usar la instrucción SIM para controlar interrupciones, el registro A debe ser cargado con el siguiente valor binario de 8 bits:

00011xyz

x es el bit de máscara para RST 7.5

y es el bit de máscara para RST 6.5

z es el bit de máscara para RST 5.5

Un bit de máscara de 1 inhabilita la interrupción correspondiente. Si el bit de máscara es 0, entonces la interrupción se habilitará si un EI ha ocurrido ya, o tan pronto como un EI es ejecutado.

El pin de la interrupción RST 7.5 es diferente que los otros pines de interrupciones en que tiene un flip flop conectado a él. Cuando un pulso aparece en el pin de RST 7.5, el flip flop es establecido. Si el flip flop es establecido y la interrupción es habilitada, se vuelve habilitada, el flip flop será restablecido y la ISR se ejecutará.

La capacidad del flip flop para recordar que una solicitud de interrupción ha ocurrido mientras que la interrupción esta inhabilitada ayuda al microprocesador a evitar fallar solicitudes de interrupción. Al habilitar la interrupción RST 7.5 por primera vez en un programa se deberá limpiar el flip flop. Esto se hace; haciendo que el bit cuatro del registro A 1 antes de ejecutar la instrucción SIM. En este programa se habilitara solo la interrupción RST 7.5 y restablece su flip flop, así el valor binario 00011011 (1B hex) pueda ser cargado en el acumulador antes de ejecutar SIM.

El programa de abajo, establecerá el vector de la interrupción RST 7.5 para saltar al ISR en FF20, inicializa el contador 8155 para enviar una onda cuadrada de

20hz al pin de RST 7.5 y luego habilita la interrupción RST 7.5. Después de eso, entrará en un loop que carga al registro A con los datos en la dirección FF2E y envía los datos hacia los LEDs discretos. El ISR interrumpirá este loop 20 veces por segundo y cada vez incrementara el byte en la dirección FF2E.

Digitar en el programa y correrlo. Se verá que los LED muestran el valor creciente de los datos en la dirección FF2B. Apretar el botón reset y poner un punto de ruptura en FF20, (el principio del ISR) entonces correr el programa. La ejecución se alojará en FF20, y si se observan los contenidos de pila (S.C.) el valor será la dirección de la siguiente instrucción que el loop principal ejecutará. Si la interrupción se produjo durante la instrucción LDA entonces S.C. será FF1B, si durante la instrucción OUT entonces S.C. será FF1D o si durante la instrucción JMP; S.C. será FF18. Ejecutar paso a paso una vez, entonces poner el punto de ruptura en FF20 de nuevo y luego correrlo a toda velocidad de nuevo. Después de las paradas de programa, verificar el valor de S.C.; puede ser cualquier de las tres direcciones mencionadas arriba.

```

vec7hlf equ 0ffe9h    ; vector para la interrupción 7.5
leds    equ 11h      ; puerto de los LEDs discretos
                    ; puertos del contador 8155
timerlo equ 14h      ; byte menor del contador
timerhi equ 15h      ; byte mayor y modo del contador
cmdreg  equ 10h      ; registro de comando

.org 0ff01h
di      ; deshabilita las interrupciones
lxi h,ticsub ; hl = dirección del ISR
shld vec7hlf ; guarda en vector
mvi a,0
out timerlo ; los 8 bits menores del contador son 0
mvi a,7ch  ; establece la onda cuadrada del 8155 en modo
            de salida
out timerhi ; con el contador en 20hz
mvi a,0cdh

```

```

        out cmdreg    ; habilita el contador
        mvi          ; un bit 0 habilita la interrupción
        a,11011b
        sim          ; limpia la máscara de interrupción 7.5
        ei           ; habilita las interrupciones
        ; este es el loop principal que será interrumpido por la ISR
loop:   lda cntout   ; contador para la salida de los LEDs
        out leds    ; despliega el valor en el registro A
        jmp loop    ; salta a loop
        ; este ISR incrementa cntout
ticsub: push psw    ; guarda A y las banderas
        lda cntout  ; toma el valor del contador
        inr a       ; lo incrementa
        sta cntout  ; lo guarda de nuevo
        pop psw     ; restaura A y las banderas
        ei         ; rehabilita las interrupciones
        ret        ; continua desde el punto de interrupción
cntout ds 1        ; reserva 1 byte para el contador
end

```

Dirección	Datos	Instrucción
FF01	F3	DI
FF02	21	LXI H,FF20
FF03	20	
FF04	FF	
FF05	22	SHLD FFE9
FF06	E9	
FF07	FF	
FF08	3E	MVI A,0
FF09	00	
FF0A	D3	OUT 14

FF0B	14	
FF0C	3E	MVI A,7C
FF0D	7C	
FF0E	D3	OUT 15
FF0F	15	
FF10	3E	MVI A,CD
FF11	CD	
FF12	D3	OUT 10
FF13	10	
FF14	3E	MVI A,1B
FF15	1B	
FF16	30	SIM
FF17	FB	EI
FF18	3A	LDA FF2B
FF19	2B	
FF1A	FF	
FF1B	D3	OUT 11
FF1C	11	
FF1D	C3	JMP FF18
FF1E	18	
FF1F	FF	
FF20	F5	PUSH PSW
FF21	3A	LDA FF2B
FF22	2B	
FF23	FF	
FF24	3C	INR A
FF25	32	STA FF2B
FF26	2B	
FF27	FF	
FF28	F1	POP PSW
FF29	FB	EI
FF2A	C9	RET
FF2B	00	(no un código de operación, pero un valor del contador)

Anexo 8.15 Guía de Practica N° 15

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



GUÍA DE PRÁCTICA N° 15 Escribiendo Programas Propios.

OBJETIVO: Que los estudiantes aprendan a codificar programas escritos por ellos, utilizando las herramientas e instrucciones aprendidas en las guías anteriores, fomentando el interés del estudiante en el área de microprogramación

METODOLOGÍA

Cada estudiante, previo estudio de la guía propuesta, procederá a desarrollarla de manera grupal, durante el tiempo programado y asignado para las prácticas correspondientes dentro del laboratorio, asistido por al menos un instructor responsable de dicha práctica.

PROCEDIMIENTO.

Nota: (Probar esta guía solo en el Primer TRAINER no en el simulador).

NUEVAS INSTRUCCIONES

LXI D, <palabra> op code = 11

Cargar la pareja de registros DE con la palabra (una palabra = 2 bytes) a la que sigue el código de operación. El byte al que sigue el código de operación va en el registro E y el byte después de ese en el registro D. Ninguna bandera es afectada.

Ahora que se tiene un conocimiento básico de la mayor parte de las instrucciones del 8085, es posible el escribir sus propios programas. El primer paso es decidir lo que quiere que el programa haga. Si lo que se quiere hacer algo similar a lo que

alguno de los programas en las lecciones previas hacen, se puede sólo modificar el programa en la lección para convenir con las necesidades. Se pueden combinar también programas de lecciones diferentes para hacer un programa.

Suponiendo que se decida hacer un programa de calculadora que añade los dígitos hexadecimales que se aprietan en el teclado numérico y luego muestra el total en la pantalla cuando se teclé una tecla aparte de un dígito hex. Para empezar la escritura del programa en sus propias palabras, en español. Esto es llamado "pseudo-código". Una versión de pseudo código del programa es lo siguiente:

1. hacer el total 0
2. leer el teclado numérico
3. si la tecla apretada no fue un dígito hex, ir al paso 6
4. agregar la tecla al total
5. ir al paso 2
6. mostrar el total
7. ir al paso 1

Ahora convertir el programa anterior a lenguaje ensamblador.

1. El primer paso requiere que el total sea hecho 0. Debe decidir que registro o par de registros tendrá el total. El par de registro DE trabajaría bien para esto. Para hacer al par de registro DE 0 se puede usar la instrucción LXI D, 0.
2. El próximo paso es leer el teclado numérico. Esto fue hecho en la lección 23 e involucra dos instrucciones de lenguaje ensamblador: MVI C, KEYIN y CALL MOS.
3. El paso tres, verificar que el número que fue apretado no es un dígito hex. Una manera fácil para hacer esto es ver si el número es mayor que F hex y esto se puede hacer con una instrucción de comparación. Recordar que en una

instrucción de comparación, si el número a ser comparado con el registro A es más grande, la bandera de acarreo es establecida. Puesto que el servicio KEYIN retorna el número de la tecla en el registro L es necesario cargar al registro A con 0F hex que es el valor con el que se quiere comparar. Para cargar al registro A con 0F y compararlo con el registro L se debe usar las dos siguientes instrucciones de lenguaje ensamblador: MVI A, 0Fh y CMP L. A fin de saltar al paso 6 cuando L es mayor que 0F hex (en cuyo caso la bandera de acarreo es establecida) una instrucción JC DSPLAY se puede usar. La etiqueta DSPLAY es usada porque no se sabe la dirección del código de pantalla todavía.

4. Existe varias formas diferentes para añadir el valor de la tecla que está en L al total que está en DE. Una de las formas más eficientes son cargar el registro H con 0 y añadir el registro DE al registro HL luego cambiar el registro DE con el registro HL así el total estará en DE de nuevo. Esto se puede hacer con las instrucciones siguientes de lenguaje ensamblador: MVI H, 0, DAD D, y XCHG.

5. El paso 5 puede ser traducido fácilmente a una instrucción de salto. Se dará el nombre a la dirección de salto la etiqueta " RDKEY " puesto que salta a las instrucciones que leen las teclas. Por lo tanto el lenguaje ensamblador para esta instrucción será JMP RDKEY.

6. Para mostrar el total que está en el registro DE se puede usar el servicio ledhex (servicio 12), que muestra el valor de hex en el par de registros DE en las pantallas "ADDRESS/REGISTER PAIR". Así las instrucciones de lenguaje ensamblador para esto serán: MVI C, LEDHEX y CALL MOS.

7. El último paso puede ser fácilmente traducido para la instrucción de lenguaje ensamblador JMP START, donde START es una etiqueta indicador a la primera instrucción.

Hacer una lista de las instrucciones de lenguaje ensamblador que han sido dadas arriba. Empezar las instrucciones con una instrucción de ORG seguida por las instrucciones EQU necesarias para definir los valores de "mos", "keyin" y "ledhex" y poner las etiquetas "start" y "rdkey" en el campo de etiqueta antes de las instrucciones apropiadas. También añadir comentarios a cada línea para

mejorar el interés y poner una instrucción de END al final. Ello debe parecerse a lo siguiente:

```

                .org 0ff01h
mos      equ 1000h    ; dirección de comienzo de las subrutinas MOS
keyin    equ 0bh     ; numero de servicio keyin
ledhex   equ 12h     ; numero de servicio ledhex
start:   lxi d,0     ; limpia el total en DE
cmdreg   equ 10h     ; registro de comando
rdkey:   mvi c,keyin ; C = numero de servicio keyin
         call mos    ; carga a L con la tecla del teclado numérico
         mvi a,0fh   ; valor máximo de las teclas
         cmp l       ; compara la tecla del teclado numérico
         jc dsplay   ; despliega el valor si L>F
         mvi h,0     ; limpia el registro H
         dad d       ; agrega el total a HL
         xchg        ; coloca HL en DE
         jmp rdkey   ; toma otra tecla
dsplay:  mvi c,ledhex ; c = numero de servicio keyin
         call mos    ; total a desplegar en DE
         end

```

Para cambiar el programa de lenguaje ensamblador a lenguaje de máquina, conseguir un poco de papel rallado y en la línea superior poner los títulos:

DIRECCIÓN

DATOS

INSTRUCCIÓN

En la primera línea bajo el título de dirección, poner la dirección hexadecimal del principio del programa (en este caso FF01). La instrucción LXI D, 0 ira bajo la cabecera de titulo "instrucción" y el código de operación para la instrucción (11) ira bajo el título "datos". Según la definición de la instrucción, los datos que serán

cargados en el par de registros DE seguirá el código de operación en las dos direcciones de memoria próximas. Puesto que los datos que están cargado en DE son 00, poner en la línea siguiente la dirección FF02 y poner 00 bajo el título "datos". Hacer lo mismo en la línea siguiente excepto que debe poner FF03 bajo el título "dirección". Ver que cada línea representa una dirección de memoria igual como en la lista de lenguaje de máquina en las lecciones previas. La lista debe parecerse a esto hasta ahora.

DIRECCIÓN	DATOS	INSTRUCCIÓN
FF01	11	LXI D,0
FF02	00	
FF03	00	

Para convertir el resto del programa a lenguaje de máquina probablemente será fácil de conseguir los códigos de operación de la tabla 5-2 al final del apéndice C que buscarlos en las lecciones previas. El código de operación para la instrucción MVI C,KEYIN se puede proporcionar en la primera columna mnemónica y su código de operación (0E) se pueda encontrar en la columna a la izquierda de esta. Se puede notar que el mnemónico en la tabla es MVI C,D8. El D8 no es el numero D8 hex, significa que el 8085 espera 8 bits (un byte) de datos para seguir el código de operación. En el caso de la instrucción que se está traduciendo en estos momentos, los datos que siguen el código de operación son el valor asignado a "keyin" por la instrucción EQU que es 0B hex.

En la misma columna mnemónica, debajo de MVI C, D8 está la instrucción LXI D, D16 que ya se tradujo al lenguaje de máquina. El D16 significa que el 8085 esperará 16 bits (dos bytes) para seguir el código de operación de la instrucción. También dentro de la tabla se verán las instrucciones que tenga la abreviación "Adr" (como JMP Adr). Estas instrucciones también esperan dos bytes de datos para seguir el código de operación, pero en este caso los datos representan una dirección. El byte que sigue el código de operación es el byte menos significativo de la dirección y el que esta después de ese es el byte más significativo. Si se encuentra "Adr" como parte de una instrucción, la mayor parte del tiempo es

necesario esperar hasta que el programa está casi completamente traducido antes que se pueda saber los valores a usar para "Adr". Ir hacia adelante y poner las direcciones en el campo de dirección, pero dejar en blanco el campo de datos hasta más tarde. Abajo esta una lista de la versión en lenguaje de máquina del programa con la "Adr" desconocida en blanco.

DIRECCIÓN	DATOS	INSTRUCCIÓN
FF01	11	LXI D,0
FF02	00	
FF03	00	
FF04	0E	MVI C,0B
FF05	0B	
FF06	CD	CALL 1000
FF07	00	
FF08	10	
FF09	3E	MVI A,0F
FF0A	0F	
FF0B	BD	CMP L
FF0C	DA	JC
FF0D		
FF0E		
FF0F	26	MVI H,0
FF10	00	
FF11	19	DAD D
FF12	EB	XCHG
FF13	C3	JMP
FF14		
FF15		
FF16	0E	MVI C,12
FF17	12	
FF18	CD	CALL 1000
FF19	00	

FF1A	10	
FF1B	C3	JMP
FF1C		
FF1D		

Para encontrar los valores para reemplazar "Adr" en las instrucciones de salto listadas arriba, se debe determinar los valores de las etiquetas usados por las instrucciones. La etiqueta "start" apunta a la primera instrucción en el programa en la dirección FF01, la etiqueta "rdkey" apunta a la segunda instrucción del programa en la dirección FF04 y la etiqueta "dsplay" apunta a la instrucción en la dirección FF16. Llenar las direcciones para cada etiqueta y tendrá la siguiente lista:

DIRECCIÓN	DATOS	INSTRUCCIÓN
FF01	11	LXI D,0
FF02	00	
FF03	00	
FF04	0E	MVI C,0B
FF05	0B	
FF06	CD	CALL 1000
FF07	00	
FF08	10	
FF09	3E	MVI A,0F
FF0A	0F	
FF0B	BD	CMP L
FF0C	DA	JC FF16
FF0D		
FF0E		
FF0F	26	MVI H,0
FF10	00	
FF11	19	DAD D
FF12	EB	XCHG

FF13	C3	JMP FF04
FF14		
FF15		
FF16	0E	MVI C,12
FF17	12	
FF18	CD	CALL 1000
FF19	00	
FF1A	10	
FF1B	C3	JMP FF01
FF1C		
FF1D		

Cargar el programa en memoria y apretar el botón reset y correrlo. Apretar una o una combinación de tecla en hex y ellos serán añadidos en conjunto (las teclas no son mostrados conforme se van apretando). Los datos actuales en la pantalla continuarán siendo mostrados hasta que apretar una tecla no-dígito y entonces el total se mostrará. Cuando primero se corre el programa " Func." será mostrado hasta que se apriete una tecla no-dígito.

Tener en cuenta las limitaciones de memoria cuando se está escribiendo un programa. Si no tiene una de las opciones de actualización del PRIMER, la memoria disponible es de FF01 a FFD3 (211 bytes). Si se está usando la pila en su programa, recordar que 2 bytes son usados con cada PUSH o CALL y cuando una interrupción ocurre. Esto reduce aun más la cantidad de memoria disponible.

Entre más habilidoso se vuelve al escribir programas se puede ser capaz de saltar la fase de pseudo código, o aún la fase de lenguaje ensamblador sin hacer muchas equivocaciones. Cuando se escribe un programa propio, no debe desalentarse si ello no hace el trabajo, la primera versión de un programa raramente trabaja. Se puede necesitar usar la ejecución paso a paso para descubrir porque el programa no está trabajando de la forma en que se espera. Se encontrará con que cuando se está comenzando, la mayor parte de los problemas de programación se causaron por un conocimiento incompleto de la forma en que una instrucción trabaja. Por este motivo es bueno para el usuario el estudiar la enciclopedia de conjunto de

instrucciones INTEL incluida en el apéndice C. Por ejemplo, una equivocación común es esperar la ejecución de una instrucción DCX (decrementar el par de registros) para establecer las banderas de condición. La definición de esta instrucción dice que ninguna bandera es afectada. Así si esta instrucción ha sido usada en un loop condicional para controlar cuantas veces el loop ocurrió, el loop puede ejecutarse infinitamente.

Si se tiene acceso a una PC compatible, se puede querer desarrollar su propio ambiente de programación 8085 que acelerara el proceso de programación grandemente.

Este entorno requiere lo siguiente: un editor de texto simple, un ensamblador 8085 que genere archivos INTEL en hex, un emulador terminar con capacidad de subir el archivo a cargar, y finalmente un PRIMER con una opción estándar o deluxe (o al menos un puerto serie) y MOS versión 2.5 o mayor. En este entorno el usuario escribe sus programas con el editor de texto y genera el archivo hex ensamblando el programa. Apretando " Func." Y luego " 3 " habilitara entonces el puerto serial del PRIMER para recibir un archivo hex. Después de esto, la pantalla muestra "REC.." indicando que el PRIMER está listo para recibir los datos. Después que un final de guardado es recibido (un registro en que el que el cuarto par de dígitos siguiente a los dos puntos es 01), el PRIMER retornará al modo de entrada.

Si algún error ocurre mientras se está recibiendo el archivo hex "Err.." será mostrado seguido por un número hex. Los bits en este número, después de ser convertido a binario indican los siguientes errores:

# BIT	ERROR
0	(no usado)
1	Error de checksum
2	Ningún carácter hex fue encontrado
3	Carácter de escape encontrado
4-7	(no usado)

Apretar una tecla después del mensaje de error pondrá la parte posterior del PRIMER en el modo de entrada. Al recibir un archivo hex se puede abortar en

cualquier momento al restablecer el PRIMER o enviando un carácter de escape (1Bh) al puerto de serial del PRIMER. Al enviar un carácter de escape resultará en un "Err..08 ".

El programa es cargado en la memoria del PRIMER usando la opción de carga(o envío) del terminal emulador para enviar el archivo hex al PRIMER. En este entorno el programa es rápida y exactamente cargado, eliminando el proceso tedioso de ensamblarlo a mano el programa y teclearlo en el teclado numérico del PRIMER.

Si se tiene la opción estándar o deluxe, otra ayuda de programación es EMOS que es un sistema operativo en que toda la interacción es a través del puerto serie y está destinado a ser utilizado con una terminal de datos o un terminal emulador corrido en un PC. Es un MOS mejorado con características tales como borrar, llenar, o mover memoria, ver contenidos de registro, leer entre puertos, escribir los puertos de salida, suma y resta decimal y hex, rastreo de programa, ejecutar paso a paso, y se ha construido un desensamblador .