



Tecnicatura Universitaria
en Programación

ARQUITECTURA Y SISTEMAS OPERATIVOS

Unidad Temática N°1:

Arquitectura y Funcionamiento del
Microprocesador

Material teórico
1° Año – 2° Cuatrimestre



Índice

Arquitectura y Funcionamiento del Microprocesador 3

Introducción.....	3
Unidad Central de Procesamiento (CPU).....	4
Unidad Aritmético-Lógica (UAL).....	5
Unidad de Control (μ C)	5
Frecuencia de Señal de Reloj	6
Registros	6
Buses	9
Interfaces de E/S.....	9
Interrupciones.....	10
Interrupciones por Hardware	10
Traps / excepciones	11
Interrupciones por Software	11
Ciclo de una Interrupción	11
E/S para DMA (Acceso Directo a Memoria)	12

Unidades de medida de Capacidad y Velocidad 13

Capacidad de almacenamiento	13
Longitud o ancho de palabra	14
Tiempo de acceso	14
Ancho de banda o velocidad de procesamiento.....	14

Memorias 15

Memorias del tipo RAM	15
Memorias del tipo RAM mas conocidas	15
Memoria RAM FIFO	17
Memorias LIFO.....	17
Memorias ROM	18
Clasificación de las memorias según su tipo.....	19

Placa o tarjeta de memoria.....	20
Decodificador	20
Aplicaciones del decodificar	21
Decodificador BCD a 7 seg	22
Ecuaciones de los decodificadores	23
Manejo de un chip de memoria	23
Diseño de una placa o tarjeta de memoria. Circuito esquemático.....	24
Banco de memoria	27
Microprocesadores	29
Velocidad del procesador	29
Tipo de instrucciones	30
Modos de direccionamiento.....	30
Lenguaje Ensamblador	32
Microprocesador (μ P).....	33
Registros del μ P Intel 8085	33
Diagrama de pines	34
Set de instrucciones 8085	34
Instrucciones 8085	35
Mapa de Direcciones.....	37
Bibliografía	38

Arquitectura y Funcionamiento del Microprocesador

Objetivo Principal	<ul style="list-style-type: none"> •Orientar al alumno a conocer de forma eficiente las características y el funcionamiento del circuito integrado conocido como microprocesador (CPU) para procesar datos en una computadora, y a reconocer sus partes físicas y lógicas
Objetivos Específicos	<ul style="list-style-type: none"> •Entender la organización de un procesador en lo relativo a las unidades funcionales que lo componen: registros, Unidad Aritmética y Lógica (ALU - UAL) y unidad de control (UC). •Entender cómo ejecuta una instrucción un procesador. •Conocer la organización del conjunto de registros del procesador. •Ver los conceptos básicos relacionados con la ALU. •Entender el funcionamiento de la unidad de control del procesador. •Conocer los elementos que componen una CPU, como se fabrica, que materiales se utiliza para su creación. •Conocer los Tipos de conexión, que velocidad alcanzan dichos sistemas y su capacidad de trabajo y almacenamiento. •Reconocer algunos de los principales fabricantes del mercado

Tabla 1: Elaboración propia

Introducción

Al microprocesador se le conoce también como CPU (Central Processing Unit o Unidad Central de Procesamiento), es el cerebro de toda computadora que controla y gobierna todo el sistema.

Consiste en un circuito integrado que interpreta y ejecuta las instrucciones de los programas almacenados en memoria y que además toma los datos de las unidades de entrada, los procesa y los envía a las unidades o periféricos de salida. Es decir, se trata del componente de la computadora que se ocupa del control y el proceso de datos. La potencia de un sistema informático se mide principalmente por la de su CPU.

Entonces, la función principal que tiene es procesar los datos y transferirlos a los otros elementos del computador. Estas tareas se llevan a cabo mediante la ejecución de instrucciones.

La CPU es un elemento capaz de realizar operaciones lógicas y aritméticas además de poder auxiliar en la toma de decisiones.

Permite el procesamiento de información numérica, es decir, información ingresada en formato binario, así como la ejecución de instrucciones almacenadas en la memoria.

También permite el intercambio de datos entre direcciones de memorias y/o periféricos.

Físicamente 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. El lenguaje ensamblador es la forma básica de programar un microprocesador para que sea capaz de realizar las tareas o los cálculos que se requieran. **Se ajusta a la arquitectura de von Neumann.**

Los elementos básicos del microprocesador que estudiaremos son los siguientes:

Conjunto de registros.

Unidad aritmética y lógica.

Unidad de control.

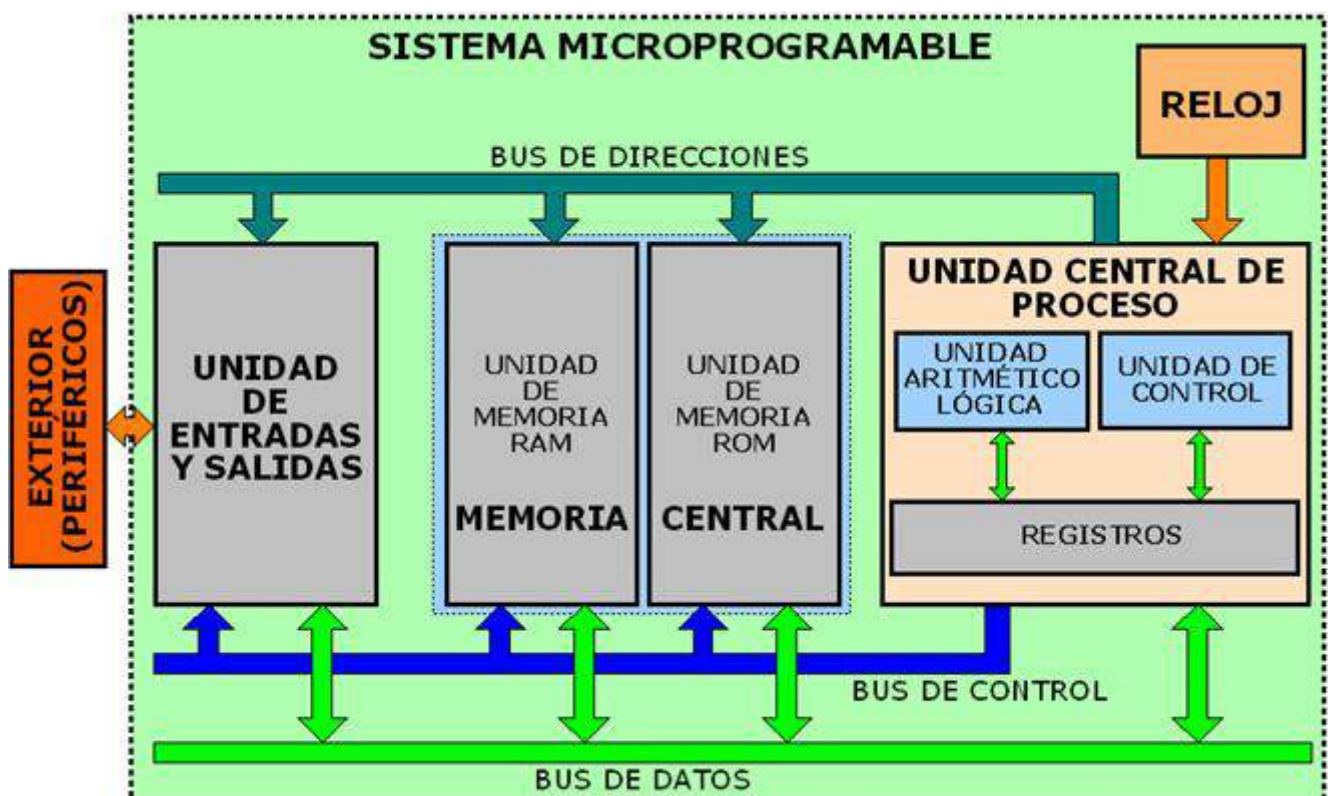


Tabla 2: Arquitectura de Von Newman

Unidad Central de Procesamiento (CPU)

La Unidad Central de Procesamiento es la encargada de leer las instrucciones de la memoria, procesarlas, realizar los cálculos solicitados y guardar en la memoria los resultados de dichas operaciones. También se encarga de establecer las comunicaciones con el exterior a través de las Unidades de Entrada

y Salida. Todas estas funciones las realiza a través de dos unidades: La Unidad Aritmético-Lógica (UAL) y la Unidad de Control (μC). μC

Unidad Aritmético-Lógica (UAL)

La Unidad Aritmético-Lógica es la encargada de realizar las operaciones aritméticas (sumas, restas, etc.) y lógicas (comparar dos números, realizar operaciones del álgebra de Boole, etc.) solicitadas por el programa en ejecución. Su funcionamiento es comandado por la Unidad de Control, es decir, la UAL opera con los datos que recibe siguiendo órdenes de la Unidad de Control.

Unidad de Control (μC)

Sintéticamente, el funcionamiento de la Unidad de Control consiste en extraer de la Unidad de Memoria Principal la instrucción a ejecutar, identificarla, establecer las señales eléctricas correspondientes para ejecutarla y luego procesar la instrucción siguiente.

Por ejemplo, si se trata de una instrucción de procesamiento de datos, la μC establece las señales que indican a la UAL las operaciones aritméticas o lógicas que debe realizar, extrae de la memoria principal los datos implicados por la instrucción; desencadena el tratamiento de dichos datos en la UAL y al finalizar, continúa con la instrucción siguiente. Esta nueva instrucción puede consistir, eventualmente, en almacenar el resultado en la Memoria Principal, para lo cual la μC lee la instrucción, la identifica, la ejecuta estableciendo las señales necesarias para esa instrucción y luego busca la instrucción que sigue. Es decir que, la μC va repitiendo en forma sucesiva estos pasos para cada nueva instrucción, hasta finalizar el programa.

Se puede deducir de lo expuesto, que la función de la Unidad de Control es la de ejecutar instrucciones y dar órdenes al resto de las unidades, controlando de esta manera el funcionamiento de todo el sistema. Envía órdenes externas de lectura/escritura a la memoria y unidades de entrada/salida y envía órdenes internas a la UAL para que esta realice las operaciones correspondientes.

La Unidad de Control recibe señales de un reloj, que es un generador de señales eléctricas en forma de pulsos que sincronizan todas las operaciones elementales de la computadora. El reloj genera los pulsos a intervalos regulares y con ellos marca el ritmo de trabajo de la CPU. Mientras mayor es la cantidad de pulsos generados por segundo más rápido trabaja la computadora. A esta velocidad de trabajo se la llama frecuencia de reloj.

Frecuencia de Señal de Reloj

Se trata de una señal eléctrica de forma rectangular, es decir sucesión indefinida de ceros y unos, que se mide en ciclos por unidad de tiempo. La unidad utilizada es el Hercio, (Hz) y sus múltiplos, Khz, Mhz, Ghz.

Registros

Los registros son, básicamente, elementos de memoria de acceso rápido que se encuentran dentro del procesador. Constituyen un espacio de trabajo para el procesador y se utilizan como un espacio de almacenamiento temporal.

Se implementan utilizando elementos de memoria RAM estática (static RAM). Son imprescindibles para ejecutar las instrucciones, entre otros motivos, porque la ALU solo trabaja con los registros internos del procesador.

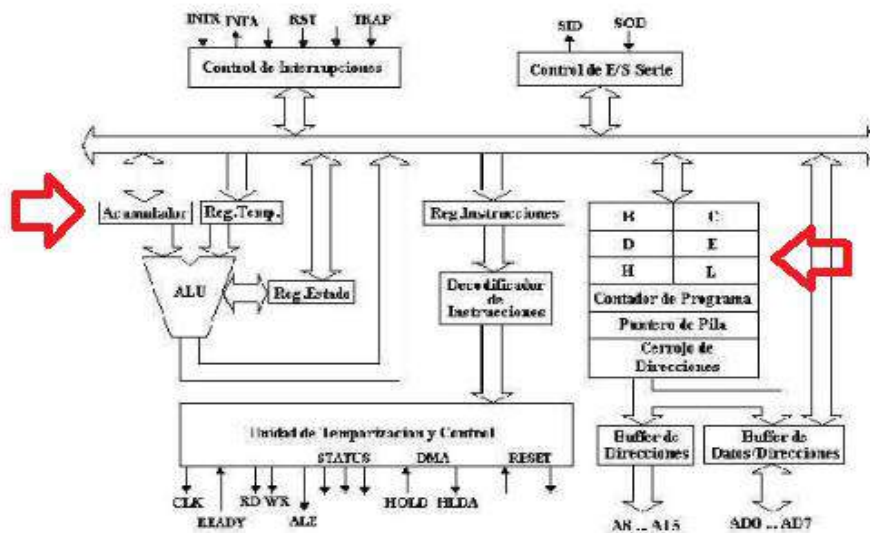


Diagrama de Bloques del Microprocesador Intel 8085

Tabla 3: Diagrama de Bloques del Microprocesador Intel 8085

Los registros del microprocesador se clasifican de la siguiente manera:

Los Registros de Propósito general suelen utilizarse como operandos en las instrucciones del ensamblador. Estos registros se pueden asignar a funciones concretas: datos o direccionamiento.

Los registros de datos se pueden diferenciar por el formato y el tamaño de los datos que almacenan; por ejemplo, puede haber registros para números enteros y para números en punto flotante.

Los **registros de direccionamiento** se utilizan para acceder a memoria y pueden almacenar direcciones o índices. Algunos de estos registros se utilizan de manera implícita para diferentes funciones, como por ejemplo acceder a la pila, dirigir segmentos de memoria o hacer de soporte en la memoria virtual.

Los **registros de instrucción** principales relacionados con el acceso a las instrucciones son:

Program counter (PC): registro contador del programa, contiene la dirección de la instrucción siguiente que hay que leer de la memoria.

Stack Pointer (SP): registro de instrucción, contiene la última dirección de memoria al utilizar instrucción de salto a subrutina. CALL.

RET. Devuelve el estado anterior del stack pointer.

Registros de estado y de control

La información sobre el estado del procesador puede estar almacenada en un único registro denominado registro de estado.

Los bits del registro de estado son modificados por el procesador como resultado de la ejecución de algunos tipos de instrucciones, por ejemplo instrucciones aritméticas o lógicas, o como consecuencia de algún acontecimiento, como las peticiones de interrupción.

Cada bit o conjunto de bits del registro de estado indica una información concreta. Los más habituales son:

Bit de cero (Zero): se activa si el resultado obtenido es 0.

Bit de Acarreo (Carry): se activa si en el último bit que operamos en una operación aritmética se produce transporte; también puede deberse a una operación de desplazamiento.

Bit de desbordamiento (Overflow): se activa si la última operación ha producido un resultado que no se puede representar en el formato que estamos utilizando.

Bit de signo (Sign): se activa si el resultado obtenido es negativo.

Bit de interrupción (Interrupt): indica si las interrupciones están habilitadas o inhibidas.

Bit de modo de operación: indica si la instrucción se ejecuta en modo supervisor o en modo usuario. Hay instrucciones que solo se ejecutan en modo supervisor. Puede activarse en forma auxiliar.

Bit de paridad (Parity): se activa si el resultado obtenido es par.

Bit de interrupción (Trap): indica si se produjo una interrupción por software. Ejemplo: división por cero.

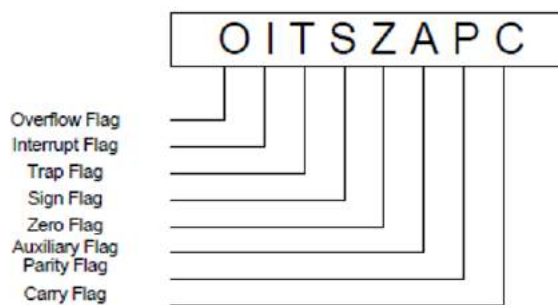


Tabla 4: Registro de Estado

Nivel de ejecución: indica el nivel de privilegio de un programa en ejecución. Un programa puede desalojar el programa que se ejecuta actualmente si su nivel de privilegio es superior.

Los registros de control son los que dependen más de la organización del procesador. En estos registros se almacena la información generada por la unidad de control y también información específica para el sistema operativo. La información almacenada en estos registros no es nunca visible para el programador de aplicaciones.

En resumen...

Se puede describir a una computadora como un conjunto ensamblado de unidades diferentes, cuyo funcionamiento viene dictado por el programa registrado en la Memoria Principal.

La Unidad de Control lee una por una las instrucciones del programa registrado en la Memoria Principal y gobierna la ejecución de dichas instrucciones utilizando las demás unidades según sea necesario.

Cuando la instrucción es una operación de cálculo, la Unidad de Control ordena a la Unidad Aritmético-Lógica que realice la operación.

Si la instrucción es una transferencia de datos con el exterior, la Unidad de Control comanda los Enlaces a Periféricos, para que realicen la conexión con los Periféricos.

Buses

La Unidad de Control necesita comunicarse, además, con la memoria y los periféricos. Esta comunicación la realiza a través de buses externos, que conforman el Bus del Sistema. Este bus está constituido por tres grupos funcionales de líneas denominados:

Bus de Datos: funciona en conjunción con el bus de direcciones y sirve para transmitir instrucciones y datos. Es bidireccional, es decir transmite de la CPU a la memoria y viceversa.

Bus de Direcciones: son líneas de señales para transmitir las direcciones de las posiciones de memoria y de los dispositivos conectados al bus. Puesto que cada línea puede tener dos posibles estados (0 y 1), con n líneas se pueden especificar 2^n direcciones distintas. Es unidireccional, es decir transmite en una sola dirección: de la CPU a la memoria.

Bus de Control: sirve para transmitir las señales de control que coordinan el funcionamiento del computador. Por ejemplo, indica a la memoria si se va a acceder para realizar una operación de lectura o de escritura.

Cuando un dato está preparado para ser transmitido, primero se envía la dirección de destino por el bus de direcciones, a continuación, se envía el dato por el bus de datos, y todas estas acciones están controladas por señales que se envían por el bus de control.

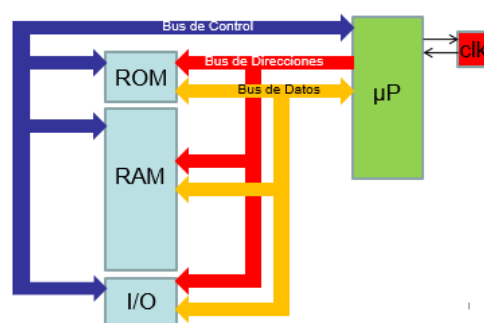


Tabla 5: Buses

Interfaces de E/S

Los controladores o drivers por hardware actúan como intermediarios entre los dispositivos E/S y el sistema. Estos interfaces tienen dos misiones principales: traducir las señales genéricas del procesador a las señales específicas del dispositivo y armonizar de alguna forma la diferencia de velocidad. El hardware de E/S está distribuido sobre una serie de interfaces, conectadas a la placa base. A su

vez, los conectores están conectados al bus del sistema, que sirve como medio de comunicación para intercambios de direcciones, datos y señales de control.

Tipos de E/S

Entrada/salida por consulta o programada: la CPU accede a los registros desde programa, para saber si el dispositivo está listo, se hace una consulta periódica.

Entrada/salida por interrupciones (IRQ): el dispositivo avisa a la CPU cuando está listo, la entrada/salida se hace mediante una rutina de servicio de interrupción.

Entrada/salida por acceso directo a memoria (DMA): el dispositivo accede directamente a la memoria, avisa a la CPU del inicio o final de la operación.

Interrupciones

Las interrupciones son el mecanismo mediante el cual un dispositivo externo (disco duro, impresora, etc) al procesador puede interrumpir el programa que está ejecutando el procesador con el fin de ejecutar otro programa (una rutina de servicio a la interrupción o RSI) para dar servicio al dispositivo que ha producido la interrupción.

¿Por qué Interrumpir?

- Por resultado de una ejecución de una instrucción: Ej: desbordamiento aritmético ("overflow"), división por cero.
- Por un temporizador interno del procesador: Permite al S.O. realizar ciertas funciones de manera regular.
- Por una operación de E/S: Ej: para indicar la finalización normal de una operación.
- Por un fallo de hardware: Ej: error de paridad en la memoria, pérdida de energía.

Interrupciones por Hardware

Las interrupciones hardware ocurren cuando un dispositivo necesita atención del procesador y genera una señal eléctrica en la línea IRQ que tiene asignada. No están relacionadas con el proceso en ejecución en ese momento. Son conocidas como interrupt request (IRQ).

Traps / excepciones

Son Interrupciones por hardware creadas por el procesador en respuesta a ciertos eventos como:

Condiciones excepcionales: overflow en ALU de punto flotante.

Falla de programa: tratar de ejecutar una instrucción no definida.

Fallas de hardware: error de paridad de memoria.

Accesos no alineados o a zonas de memoria protegidos

Interrupciones por Software

Permiten hacer llamadas directas a una función del sistema operativo. Los procesadores Intel de la gama 808X y compatibles, disponen de una instrucción INT que permite generar interrupciones por software.

Ciclo de una Interrupción

Para atender la interrupción, es necesario llevar a cabo algunas operaciones para transferir el control del procesador a la rutina de servicio de interrupciones, de manera que, después, se recupere con la garantía de que el procesador estará en el mismo estado en el que estaba antes de transferir el control a la rutina de servicio de interrupción:

Almacenar el contador del programa y la palabra de estado (generalmente, se guardan en la pila).

Almacenar la dirección donde empieza la rutina para atender la interrupción en el contador del programa.

Ejecutar la rutina de servicio de interrupción.

Recuperar el contador del programa y la palabra de estado.

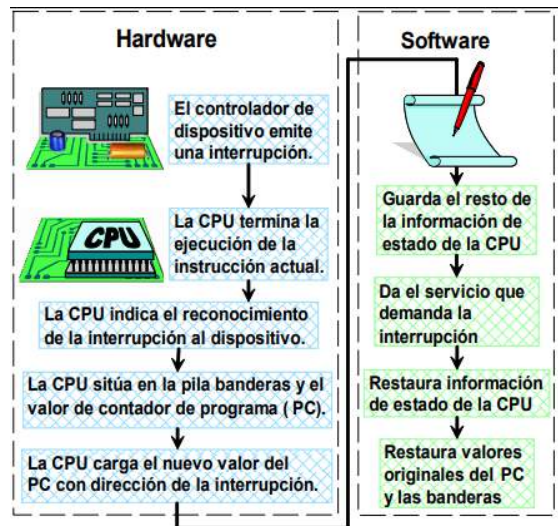


Tabla 6: Procesamiento de una interrupción

E/S para DMA (Acceso Directo a Memoria)

Las técnicas anteriores necesitan la intervención de la CPU para transferir datos entre memoria y módulos E/S. Esto tiene dos inconvenientes:

La velocidad de transferencia está limitada por el tiempo que necesita la CPU para testear y servir al periférico.

La CPU debe ejecutar una serie de instrucciones por cada transferencia E/S.

Cuando se necesita transferir grandes cantidades de datos, el acceso directo a memoria (DMA) es más eficiente. Esto implica la inclusión de un módulo adicional al bus del sistema. El módulo de DMA es capaz de sustituir a la CPU tomando control del bus.

El DMA (Direct Memory Access) es un procesador/controlador especializado en transferencias “muy grandes” desde periféricos a memoria y viceversa.

Es programable. La CPU no realiza ninguna tarea (salvo programar el DMA) ya que la inicialización y transferencia son gobernadas por el periférico.

Para programar el DMA tiene que enviar al menos los siguientes datos:

- Dirección/puerto periférico E/S.
- Posición/dirección en memoria principal.
- Tamaño (número de bytes a transferir).
- Tipo transferencia: lectura o escritura.

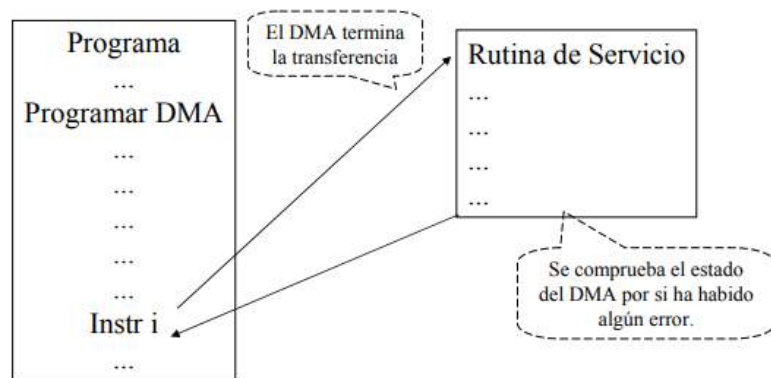


Tabla 7: Ciclo de E/S para DMA

A diferencia del mecanismo por interrupción convencional, una orden DMA puede transferir muchísimos datos de una sola vez. Por lo tanto, el número de interrupciones por byte transferido es mucho menor que con las interrupciones convencionales: se gana en velocidad. El procesador no se encarga de la transferencia de datos.

Unidades de medida de Capacidad y Velocidad

Capacidad de almacenamiento

La capacidad de almacenamiento hace referencia a la cantidad de datos o instrucciones, expresada en bytes, que un dispositivo de memoria puede guardar.

Debido a que el byte es una unidad relativamente pequeña, es habitual expresar la capacidad de almacenamiento con múltiplos. Por ejemplo, en la Memoria Principal o en las unidades de almacenamiento masivo es frecuente expresar la capacidad en GB (Gigabytes).

Unidad	Símbolo	Equivalencias en bytes
Kilobyte	KB	$1 \text{ KB} = 1024 \text{ B} = 2^{10} \text{ B}$
Megabyte	MB	$1 \text{ MB} = 1024 \text{ KB} = 2^{10} \text{ KB} = 2^{10}(2^{10} \text{ B}) = 2^{20} \text{ B}$
Gigabyte	GB	$1 \text{ GB} = 1024 \text{ MB} = 2^{10} \text{ MB} = 2^{10}(2^{20} \text{ B}) = 2^{30} \text{ B}$
Terabyte	TB	$1 \text{ TB} = 1024 \text{ GB} = 2^{10} \text{ GB} = 2^{10}(2^{30} \text{ B}) = 2^{40} \text{ B}$
Petabyte	PB	$1 \text{ PB} = 1024 \text{ TB} = 2^{10} \text{ TB} = 2^{10}(2^{40} \text{ B}) = 2^{50} \text{ B}$
Exabyte	EB	$1 \text{ EB} = 1024 \text{ PB} = 2^{10} \text{ PB} = 2^{10}(2^{50} \text{ B}) = 2^{60} \text{ B}$
Zetabyte	ZB	$1 \text{ ZB} = 1024 \text{ EB} = 2^{10} \text{ EB} = 2^{10}(2^{60} \text{ B}) = 2^{70} \text{ B}$
Yottabyte	YB	$1 \text{ YB} = 1024 \text{ ZB} = 2^{10} \text{ ZB} = 2^{10}(2^{70} \text{ B}) = 2^{80} \text{ B}$

Tabla 8: Múltiplos del Byte

Longitud o ancho de palabra

Una palabra es el conjunto de bits que la computadora utiliza en paralelo al realizar ciertas operaciones internas, como la transferencia de datos entre la CPU y la Memoria Principal. Por lo tanto, la longitud o ancho de palabra indica el número de bits que forman la palabra. Este parámetro es indicativo de la potencia de la computadora ya que, por ejemplo, mientras mayor sea la cantidad de bits que se manejan en paralelo, los operandos que procese la UAL pueden ser números más grandes o mayor la cantidad de datos que se pueden transferir entre CPU y memoria en los procesos de lectura y escritura.

Tiempo de acceso

El tiempo de acceso de la Memoria Principal es el tiempo necesario para realizar una operación de lectura o escritura. Este tiempo se mide como el intervalo que transcurre entre el instante en que se da la dirección concreta de la celda y el instante en que el dato se encuentra disponible (lectura) o ha sido almacenado (escritura).

El orden de magnitud de los tiempos de ejecución de la CPU como así también el de los tiempos de acceso a la Memoria Principal, no guarda relación con los tiempos de la escala humana; por lo que normalmente se expresa la duración de las operaciones en submúltiplos del segundo:

Unidad	Símbolo	Equivalencias en segundos
milisegundo	ms	$1 \text{ ms} = \frac{1 \text{ s}}{1000} = \frac{1 \text{ s}}{10^3} = 10^{-3} \text{ s}$
microsegundo	μs	$1 \mu\text{s} = 10^{-3} \text{ ms} = 10^{-3}(10^{-3} \text{ s}) = 10^{-6} \text{ s}$
nanosegundo	ns	$1 \text{ ns} = 10^{-3} \mu\text{s} = 10^{-3}(10^{-6} \text{ s}) = 10^{-9} \text{ s}$
picosegundo	ps	$1 \text{ ps} = 10^{-3} \text{ ns} = 10^{-3}(10^{-9} \text{ s}) = 10^{-12} \text{ s}$

Tabla 9: Múltiplos del Byte

Ancho de banda o velocidad de procesamiento

El ancho de banda o velocidad de transferencia indica la cantidad de datos (expresados en bytes) transferidos por unidad de tiempo entre un dispositivo y otro.

La unidad de tiempo utilizada generalmente es el segundo. Por ejemplo, decir que el ancho de banda del bus que comunica a la memoria y la CPU es de 133 MB/s significa que en 1 segundo se pueden transferir 133 MB entre la memoria y la CPU.

Memorias

Memorias del tipo RAM

Sin entrar en muchos detalles, qué es la memoria RAM. RAM es un acrónimo del inglés, Random Access Memory o memoria de acceso aleatorio, y sirve para dotar al sistema de un espacio virtual necesario para manejar información y solucionar problemas en cada momento instantáneo. Puedes pensar que es como un papel en el que se toman notas y luego se tira, o siendo más exactos con el ejemplo, se borra y luego se reutiliza. El tamaño de la memoria RAM sería el tamaño de esa hoja de papel, cuanto más grande sea, más número de notas podremos tomar en él. También se entiende que este tipo de memoria son volátiles. Esto implica que se borra la información almacenada si se le quita la alimentación.

Memorias del tipo RAM mas conocidas

Static RAM (SRAM)

Se trata de uno de los dos tipos básicos de memoria (el otro es DRAM, del que hablaremos a continuación). Comenzó a utilizarse en 1990 y a día de hoy sigue presente en cámaras digitales, routers o impresoras, pero también en la memoria caché de los procesadores o de los discos duros. Es un tipo de memoria que necesita un flujo de energía constante para funcionar, así que al contrario que la RAM dinámica, no necesita estar «refrescándose» para ver qué datos tiene en su interior, y por eso se le llama Static RAM (RAM estática).



Las ventajas de este tipo de memoria es que consume muy poca energía y tiene unos tiempos de acceso muy bajos. Las desventajas incluyen que tienen unas capacidades muy bajas, y unos costes de fabricación bastante elevados.

Dynamic RAM (DRAM)

Es el otro tipo básico de memoria RAM, y se utilizó desde principio de los años 1970 hasta mediados de los años 90. Este tipo de memoria necesita un «refresco» periódico de los datos en su interior porque tienen condensadores que periódicamente se van descargando, y la falta de energía significa pérdida de datos. Por eso se le llama RAM dinámica.



La ventaja de este tipo de memoria es que era más barata de fabricar, y permitía mayores capacidades. Las desventajas, es que tienen unos tiempos de acceso más elevados y consumen más energía.

Synchronous Dynamic RAM (SDRAM)

Este tipo de memoria funciona en sincronía con el procesador, lo que significa que espera a la señal de reloj antes de responder, teniendo como beneficio que permitía al procesador ejecutar órdenes en paralelo. En otras palabras, con este tipo de memoria se puede aceptar una orden de lectura antes de haber terminado de procesar una de escritura. Este proceso, conocido como «pipelining», no afecta al tiempo que se tarda en procesar instrucciones, sino que da la posibilidad de ejecutar varias simultáneamente.



Este tipo de memoria se utiliza desde 1993 hasta día de hoy, tanto en ordenadores como en videoconsolas, y casi todos los siguientes tipos de memoria RAM están basados en este tipo.

Single Data Rate Synchronous Dynamic RAM (SDR SDRAM)

Es un tipo de memoria que vio la luz en 1993 y se sigue utilizando a día de hoy. Es una variante mejorada de la memoria SDRAM que mejora la manera en la que procesa la información de lectura y escritura. «Single Data Rate» significa que se ejecuta una instrucción de lectura y otra de escritura por cada ciclo de reloj del procesador.

La memoria SDR SDRAM es básicamente la segunda generación de memoria SDRAM, y pasó a conocerse simplemente con este nombre cuando se extendió su uso.

Double Data Rate Synchronous Dynamic RAM (DDR SDRAM)

Este tipo de memoria RAM se estandarizó a partir del año 2000, y a partir de aquí surgieron las siguientes generaciones: DDR2, DDR4 y las actuales DDR4.

Opera de la misma manera que la SDR SDRAM solo que el doble de rápido, es decir, es capaz de realizar dos instrucciones de lectura y dos de escritura por cada ciclo de reloj del procesador. Aunque es una versión mejorada de la SDR SDRAM, tiene diferencias físicas pues se amplía el número de pines de 168 a 184. Este tipo de memoria también opera a diferente voltaje (2.5V frente a los 3.3V de la SDR DRAM).

Las siguientes versiones mejoradas son:

DDR2 SDRAM: aunque mantiene el mismo número de operaciones por ciclo de reloj (dos de lectura y dos de escritura), es más rápida porque es capaz de funcionar a mayores velocidades. Las DDR funcionaban a 200 Mhz, mientras que las DDR2 lo hacían a 533 Mhz, con un menor voltaje (1.8V) y más pines (240).

DDR3 SDRAM: múltiples mejoras respecto a las DDR2, que incluyen más velocidad, capacidad, menor consumo (1.5V) y mayor velocidad de funcionamiento (800 Mhz). Aunque tiene el mismo número de pines que la DDR2, estos aspectos hacen que no sean compatibles.

DDR4 SDRAM: mejora de nuevo el rendimiento sobre la DDR3 con mayores velocidades (1600 Mhz), capacidades y funcionan a menor voltaje (1.2V). Este tipo de SDRAM usa 288 pines, así que tampoco es compatible con los anteriores.

Memoria RAM FIFO

Este tipo de memoria es de acceso secuencial. El acrónimo formado por las iniciales de las palabras "First In" "First On" (primero en entrar, primero en salir) que se aplica a un tipo de memorias que se caracterizan por poseer un doble puerto: uno de entrada y otro de salida, que actúan asincrónicamente y en las cuales los datos pueden ser leídos únicamente en el mismo orden que fueron escritos. Es una organización serie-serie porque los datos se leen en serie y se escriben en serie y es FIFO porque a cada pulso de reloj se desplazan una posición los contenidos de todos los registros de desplazamiento, de forma que la palabra que entró primero, es la que también sale primero.

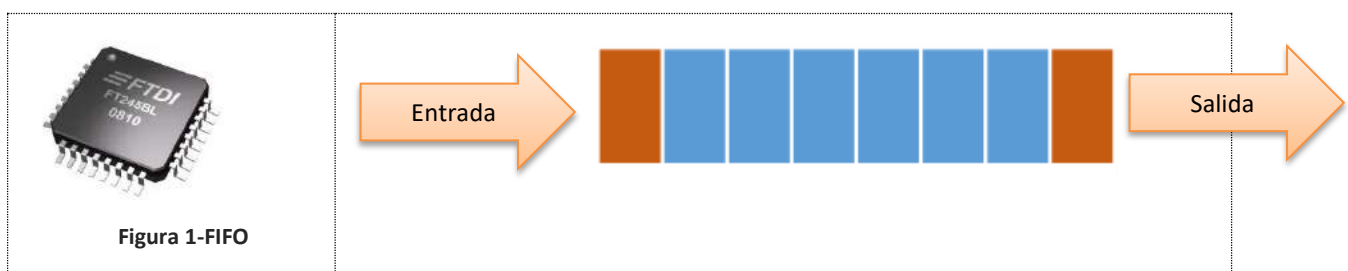


Imagen 1: Elaboración propia

Memorias LIFO

El acrónimo hace referencia de que el último que entra, primero en salir.

Cada pulso de reloj entra una nueva palabra que empuja a las que habían entrado antes aumentando la altura de la "pila". Si cambiamos ahora de lectura a escritura hay que cambiar también el sentido del desplazamiento y aparecen en el primer registro las palabras en orden inverso al que fueron escritas.

Esta organización se llama LIFO (Last-In,First-Out) ya que la última palabra que entró en la memoria es la primera que sale cuando el control pasa de escritura a lectura. También se le suele llamar memorias de pila ("stack") porque las palabras se "amontonan" en forma de pila.

Memorias ROM

En informática, cuando hablamos de memoria ROM (acrónimo de Read-Only Memory, es decir, Memoria de Sólo Lectura), nos referimos a un tipo de almacenamiento empleado en computadoras y otros dispositivos electrónicos, que se caracteriza por ser únicamente de acceso para lectura y nunca para escritura, es decir, que se la puede recuperar pero no modificar o intervenir.

La memoria ROM es de acceso secuencial y su presencia es independiente de la presencia de una fuente de energía. Como se ha dicho, su contenido no puede modificarse, o al menos no de manera simple y cotidiana, y suele contener información introducida en el sistema por el fabricante, de tipo básico, operativo o primario.

Este tipo de memoria opera, además, de manera mucho más lenta que su contrapartida, la RAM (acrónimo de Random Access Memory, es decir, Memoria de Acceso Aleatorio), por lo que su contenido suele volcarse en esta última para ejecutarse más velozmente.

Existen, no obstante, versiones de memoria ROM (conocidas como EPROM y Flash EEPROM) que pueden ser programadas y reprogramadas varias veces, a pesar de que su funcionamiento se rige por las mismas reglas del tradicional. Sin embargo, como su proceso de reprogramación es poco frecuente y relativamente lento, se las continúa llamando del mismo modo.

PROM.

Acrónimo de Programmable Read-Only Memory (Memoria de Sólo Lectura Programable), es de tipo digital y puede ser programada una única vez, ya que cada unidad de memoria depende de un fusible que se quema al hacerlo.

EPROM.

Acrónimo de Erasable Programmable Read-Only Memory (Memoria de Sólo Lectura Borrable y Programable) es una forma de memoria PROM que puede borrarse al exponerse a luz ultravioleta o altos niveles de voltaje, borrando la

información contenida y permitiendo su remplazo. Se pueden regrabar una 10.000 Veces

EEPROM.

Acrónimo de Electrically Erasable Programmable Read-Only Memory (Memoria de Sólo Lectura Borrable y Programable Eléctricamente) es una variante del EPROM que no requiere rayos ultravioletas y puede reprogramarse en el propio circuito, pudiendo acceder a los bits de información de manera individual y no en conjunto. Se pueden regrabar una 1.000.000 Veces.

Memoria FLASH:

Es igual que una EEPROM su diferencia radical en la velocidad de grabado de los datos, además de la tensión utilizada para borrar es de 5v o 3.3v afectada de la memoria. Es la más utilizada actualmente y existe un sin número de variantes.



Figura 2 - EPROM



Figura 3 - EEPROM



Figura 4 - ROM

Imagen 2: Elaboración propia

Clasificación de las memorias según su tipo

Clasificación según su tipo				
Tipo de memoria	Clase	Borrado	Mecanismo de escritura	Volatilidad
RAM	Lectura/Escritura	Eléctricamente por byte	Eléctricamente	Volátil
ROM	Solo Lectura	No es posible	Mediante máscaras	NO Volátil
PROM			Eléctricamente	
EPROM	Luz ultravioleta chip completo			
EEPROM	Eléctricamente por bytes			
Flash				

Tabla 10: Extraído de Memoria rom

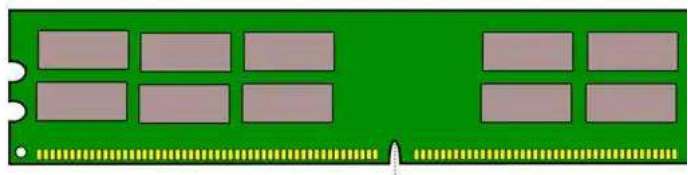
Placa o tarjeta de memoria

Para el diseño de una placa de memoria, se realiza en su primera etapa el diseño del circuito esquemático. En este diseño se especifica los componentes utilizados y las conexiones eléctricas entre los componentes utilizados, logrando en su conjunto, un dispositivo con determinada aplicación.

Para lograr el diseño esquemático de una placa de memoria, se debe conocer algunos componentes y conocimientos previos que se expondrán a continuación.



Memoria tipo DDR



Decodificador

Un decodificador o descodificador es un circuito combinacional, cuya función es inversa a la del codificador, es decir, convierte un código binario (natural, BCD, etc.) de N bits de entrada y M líneas de salida (N puede ser cualquier entero y M es un entero menor o igual a 2^N), tales que cada línea de salida será activada para una sola de las combinaciones posibles de entrada. Normalmente, estos circuitos suelen encontrarse como decodificador / demultiplexor. Esto es debido a que un demultiplexor puede comportarse como un decodificador.

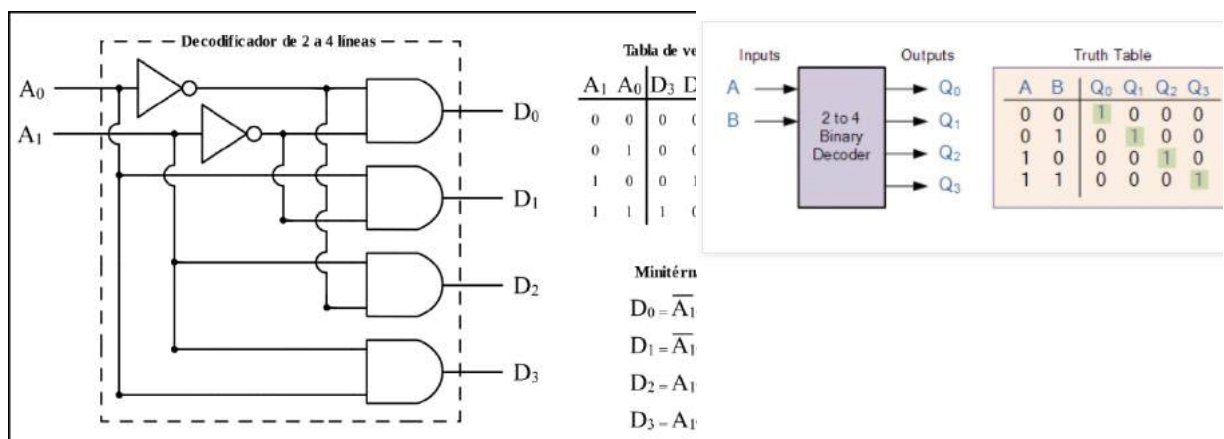


Imagen 3: Elaboración propia

Aplicaciones del decodificar

Su función principal es la de direccionar espacios de memoria. Un decodificador de N entradas es capaz de direccionar 2^N espacios de memoria.

Para poder direccionar 1KB de memoria se necesitarían 10 bits, ya que la cantidad de salidas sería 1024, igual a 2^{10} .

De esta manera:

Con 20 bits se tienen 220 lo que equivale a 1MB.

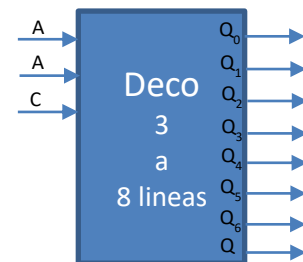
Con 30 bits se tienen 230 lo que equivale a 1GiB

Cada vez que se agrega una línea de dirección, se duplica la capacidad de direccionamiento.

También sirve para realizar funciones lógicas, por ejemplo:

Se quiere obtener de 3 variables de entrada (ABC) la función $F_{a(ABC)} = \sum m_1, m_3, m_5, m_6$

	A	B	C	Q(0)	Q(1)	Q(2)	Q(3)	Q(4)	Q(5)	Q(6)	Q(7)
m ₀	0	0	0	1	0	0	0	0	0	0	0
m ₁	0	0	1	0	1	0	0	0	0	0	0
m ₂	0	1	0	0	0	1	0	0	0	0	0
m ₃	0	1	1	0	0	0	1	0	0	0	0
m ₄	1	0	0	0	0	0	0	1	0	0	0
m ₅	1	0	1	0	0	0	0	0	1	0	0
m ₆	1	1	0	0	0	0	0	0	0	1	0
m ₇	1	1	1	0	0	0	0	0	0	0	1



	A	B	C	F(a)
m ₀	0	0	0	0
m ₁	0	0	1	1
m ₂	0	1	0	0
m ₃	0	1	1	1
m ₄	0	0	0	0
m ₅	0	0	1	1
m ₆	0	1	0	1
m ₇	0	1	1	0

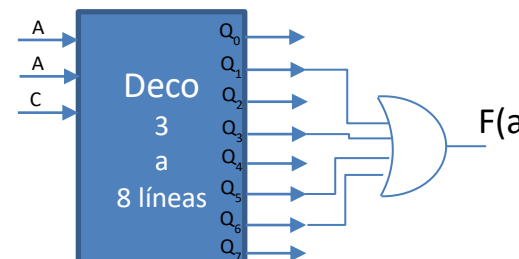
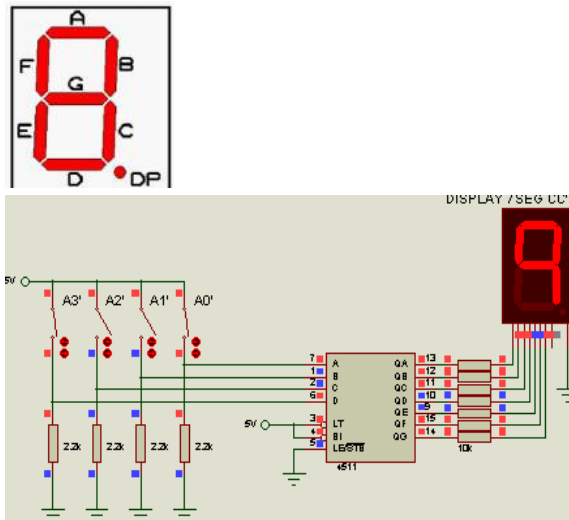


Tabla 11: Elaboración propia

Decodificador BCD a 7 seg

Un decodificador muy conocido es el decodificador BCD a 7 segmentos, donde se traduce en un número binario para ser interpretado por una persona

	A	B	C	D	00	01	02	03	04	05	06	07	08	09
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	1	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	1	1
9	1	0	1	0	1	1	1	1	1	1	1	1	1	1
10	1	0	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	0	0	1	1	1	1	1	1	1	1	1	1
12	1	1	0	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	0	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1



$$\begin{aligned}
 F_{A(ABCD)} &= \sum (0,2,3,5,6,7,8,9) \quad X \geq 10 \\
 F_{B(ABCD)} &= \sum (0,1,2,3,4,7,8,9) \quad X \geq 10 \\
 F_{C(ABCD)} &= \sum (0,1,3,4,5,6,7,8,9) \quad X \geq 10 \\
 F_{D(ABCD)} &= \sum (\text{de deber}) \\
 F_{E(ABCD)} &= \sum (\text{de deber}) \\
 F_{F(ABCD)} &= \sum (\text{de deber})
 \end{aligned}$$

Imagen 4: Elaboración propia

Ecuaciones de los decodificadores

$$Cantidad = Base^{Dígitos}$$

Por ejemplo para un decodificador de 4 entradas, cuantas salidas podemos activar.

$$2^4 = 8$$

Cada salida de un decodificador, puede ser la celda activa de un bit de memoria, esto se puede interpretar que en una memoria debe existir un elemento similar al decodificador para poder seleccionar un bit de memoria. Por ejemplo, un decodificador de 10 entradas, puede seleccionar:

$$2^{10} = 1024 \text{ bit} = 1kb$$

Nota: esta memoria tiene como dato de entrada o salida de 1 bit de información

En el caso de que se tiene la cantidad, y se quiere saber los dígitos para cualquier base, la ecuación es la siguiente:

$$Dígitos = \frac{\ln(Cantidad)}{\ln 2}$$

Por ejemplo, se quiere saber la cantidad de líneas de direcciones (Dígitos binarios) que necesita una memoria de 1GB, la operación es la siguiente:

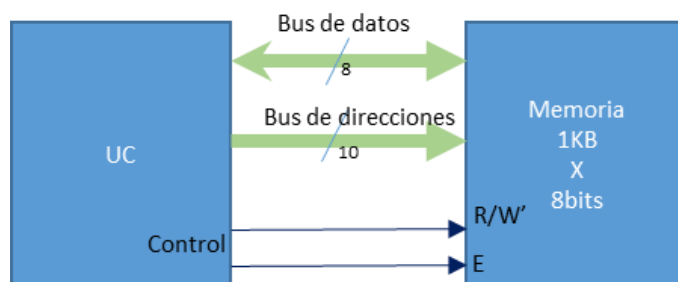
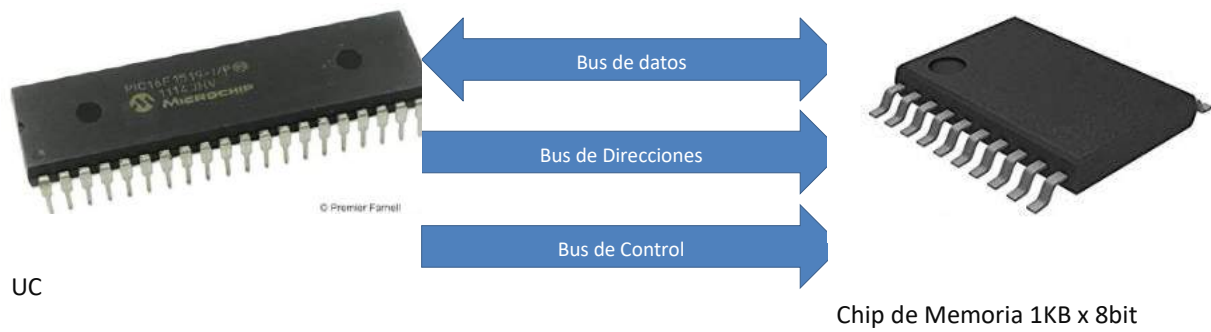
$$Dígitos = \frac{\ln(1GB \times 1024 \times 1024 \times 1024)}{\ln 2} = 30$$

Se necesitan 30 líneas de direcciones, para poder acceder a cualquier parte de la memoria de 1GB.

Manejo de un chip de memoria

Para una memoria del tipo RAM, que se encuentra en un único encapsulado, se debe conocer todas las líneas de bus de addres; bus de datos y las de control. Se debe realizar una determinada secuencia para poderla utilizar como lectura o escritura de datos. Generalmente el control lo realiza un microcontrolador, que es el que necesita de alguna manera almacenar información en una memoria.

Supongamos que tenemos un chip de memoria de 1KB de almacenamiento y el bus de datos es de 8bit. En el diagrama esquemático, la memoria, es utilizada por un μC .



1. Bus de datos. 8 bits (1Byte). Bidireccional porque el dato en la memoria, se puede escribir o leer.
1. Bus de direcciones. 10 línea. Unidireccional. El μC indica que dirección quiere de la memoria leer o escribir.
2. Control R/W'. 1bit. Unidireccional. Se le indica a la memoria si se la quiere leer o escribir. "1" \rightarrow se lee.
3. Control E. Enable. Habilita la memoria.

Imagen 4: Elaboración propia

Diseño de una placa o tarjeta de memoria. Circuito esquemático

Cuando se necesita mayor capacidad de memoria que dispone un solo chip de memoria, se fabrican placas de memoria que contiene varios chip. Se deben conectar entre ellas, mediante diferentes buses. Un bus es una definición compacta de referirnos a varias conexiones eléctricas que tiene un mismo propósito.

Los buses que se utilizan en la placa de memoria son:

Bus de datos

Bus de direcciones (address)

Bus de control

Bus de datos:

Es un grupo de cables conductores que transportan solo datos. El bus de datos es bidireccional porque el flujo de datos en ambas direcciones, desde el microcontrolador (μC) a la memoria y desde la memoria al μC .

La longitud del bus de datos del μC (por ejemplo PIC16F877) es de 8 bits (es decir, dos dígitos hexadecimales), que van desde 00 H hasta FF H. (H denota hexadecimal).

Cuando se trata de una operación de escritura, el μC colocará los datos (que se escribirán) en el bus de datos, cuando sea una operación de lectura, el μC leerá en el bus de datos, los datos provisto por la memoria.

Bus de direcciones:

Es un grupo de cables conductores que transporta solo direcciones. El bus de direcciones es unidireccional porque los datos fluyen en una dirección, desde el μC a la memoria.

Por ejemplo, si la longitud del bus de direcciones del μC es de 16 bits (es decir, cuatro dígitos hexadecimales), que van desde 0000 H a FFFF H, (H denota hexadecimal). El μC puede transferir una dirección máxima de 16 bits, lo que significa que puede abordar 65, 536 (64KB) ubicaciones de memoria diferentes.

La longitud del bus de direcciones determina la cantidad de memoria que un sistema puede direccionar. Tal como un sistema con un bus de direcciones de 32 bits puede direccionar 2^{32} ubicaciones de memoria. Si cada ubicación de memoria contiene un byte, el espacio de memoria direccionable es de 4 GB.

Bus de control:

Es un grupo de cables conductores, que se utiliza para generar señales de tiempo y control para controlar todos los periféricos asociados, el μC usa el bus de control para procesar datos, eso es lo que debe hacer con la ubicación de memoria seleccionada. Algunas señales de control son:

Lectura /Escritura (R/W)

Habilitación (E).

Por ejemplo, si se quiere diseñar una placa de memoria de 16KB x 8 bit, pero se disponen de chip de memoria 4KB x 8 bit, para que un μC pueda disponer de esta cantidad de memoria, se debe realizar el siguiente circuito esquemático.

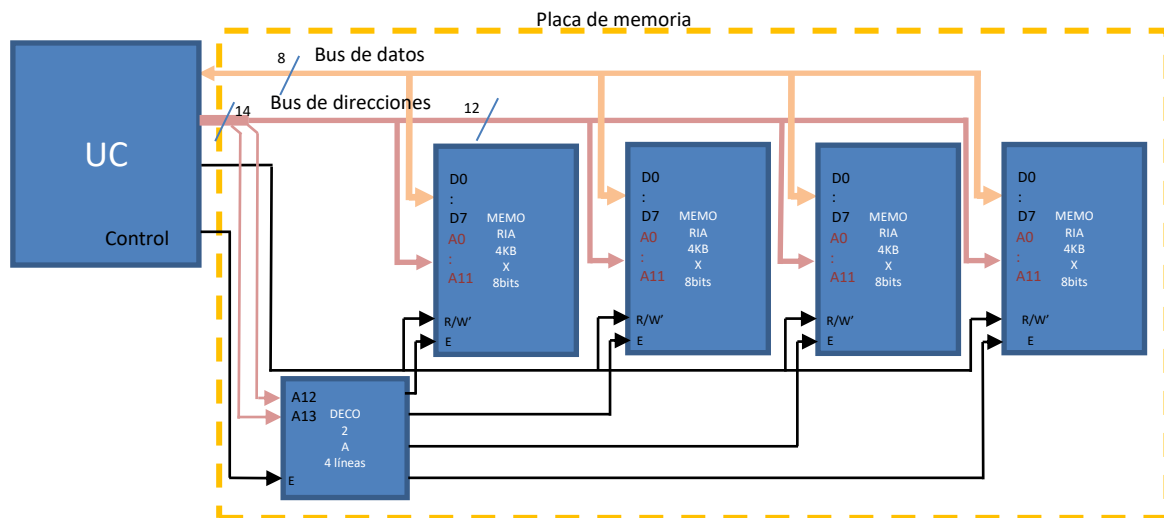


Imagen 5: Elaboración propia

La secuencia del μC para escribir una posición de memoria es la siguiente:

Coloca la posición de memoria en el bus de direcciones.

Coloca el dato en el bus de datos.

Activa la línea de control R/W indicando que va a escribir en la memoria un dato (R/W=0)

Activa con un pulso la señal de E del Decodificador. La duración debe ser acorde a lo que estipula el fabricante del chip de memoria.

Deshabilitar la placa de memoria.

En resume, la principal diferencia entre el bus de direcciones y el bus de datos es que el bus de direcciones ayuda a transferir direcciones de memoria mientras que el bus de datos ayuda a enviar y recibir datos. Es decir, el bus de direcciones se usa para especificar una dirección física en la memoria, mientras que el bus de datos se usa para transmitir datos entre componentes en ambas direcciones. El bus de control, es el encargado de especificar si la placa de memoria está activa, y si se la va a utilizar como lectura/escritura



Figura -SRAM

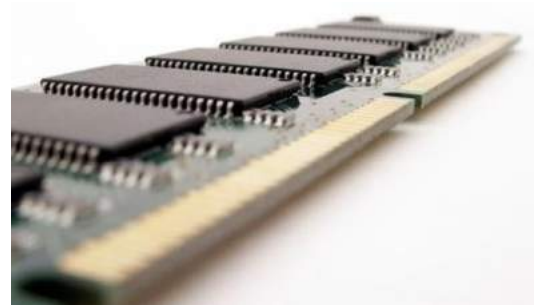


Figura - DDR266

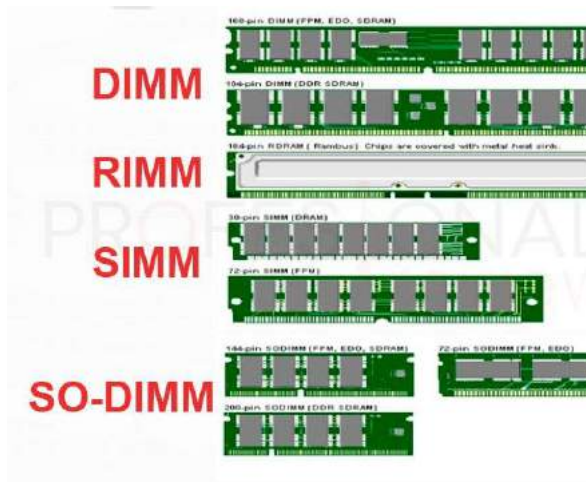


Figura DRAM

Banco de memoria

El banco de memoria, es un diagrama en bloques, que nos muestra la dirección de inicio y la dirección final de la memoria. Cuando hay varias memorias que forman una plaqueta o tarjeta de memoria, aparece lo que se denomina, el paginado de la memoria. Cada página tiene la dirección inicial y final generalmente representada en Hexadecimal que corresponde a su dirección individual para cada chip o integrado de memoria, para poder acceder a su información.

Cabe destacar que el usuario final, que generalmente es un microcontrolador o un microprocesador, no le interesa saber cuántos chips de memoria conforman la placa. Al μC , solo le interesa saber la capacidad total de memoria a la cuál puede utilizar. La misma placa de memoria, administra que chip de memoria está utilizando, dependiendo de la dirección solicitada por el μC .

En resumen, se pueden diseñar placas de memoria con la misma capacidad vista desde en μC y en la placa de memoria nos, podemos encontrar con 4;8;16 chips de memorias en la tarjeta de memoria.

<p>Banco 1 4KBx8</p>	0000(h)	$2^{12} = 4096 \text{ Byte} = 4KB$	$2^{14} = 16384 \text{ Byte} = 16KB$
<p>Banco 2 4KBx8</p>	0FFF(h) 1000(h)	$4096_{(d)} = 1000_{(h)}$	$16384_{(d)} = 4000_{(h)}$
<p>Banco 3 4KBx8</p>	1FFF(h) 2000(h)	Como se cuenta desde 0	Como se cuenta desde 0
<p>Banco 4 4KBx8</p>	2FFF(h) 3000(h)	$1000_{(h)} - 1_{(h)} = FFF_{(h)}$	$4000_{(h)} - 1_{(h)} = 3FFF_{(h)}$
<p>Banco 4 4KBx8</p>	3FFF(h)	Dirección inicial 0 (h)	Dirección inicial 0 (h)
		Dirección final FFF(h)	Dirección final 3FFF(h)
		Se le suma una dirección. Esta es la primer dirección del siguiente banco de memoria	
		$FFF_{(h)} + 1 = 1000_{(h)}$	
		La dirección final del segundo banco es $1000_{(h)} + FFF_{(h)} = 1FFF_{(h)}$	

Microprocesadores

Velocidad del procesador

La velocidad del procesador determina la rapidez con la que se realizan las acciones de la computadora, por ejemplo, la velocidad a la que se ejecutarán los programas. Hoy en día, las computadoras personales cuentan con procesadores de diversas marcas como AMD e Intel, entre otras. Estos procesadores son similares, así que una referencia para comparar su velocidad es observar la frecuencia del reloj.

Recordando lo antes dicho, el reloj es un generador de pulsos que sirven para sincronizar las operaciones elementales de la computadora. Se puede decir entonces que la frecuencia de reloj indica la velocidad a la que una computadora realiza sus operaciones más básicas, como la suma de dos números por parte de la UAL.

El término frecuencia de reloj se refiere a la velocidad de funcionamiento del procesador principal, que se expresa en millones de ciclos por segundo (Megahertz o MHz) o en miles de millones de ciclos por segundo (Gigahertz o GHz). Cuando se dice que una computadora es una máquina de 4 GHz, esto significa que la frecuencia de su reloj es de 4 mil millones de ciclos por segundo.

En resumen...

El reloj de la CPU genera señales eléctricas en forma de pulsos a intervalos regulares. Estos pulsos sincronizan todas las operaciones elementales de la computadora.

A la cantidad de pulsos generados por el reloj en 1 segundo se le denomina frecuencia de reloj y se mide en ciclos por segundo (hercios o hertz). En computadoras actuales, este valor se expresa en GHz (gigahertz: miles de millones de ciclos por segundo).

La frecuencia de reloj indica la velocidad del procesador (velocidad con que realiza sus operaciones más básicas).

La velocidad del procesador es un factor importante en el ritmo de trabajo de una computadora. Otro factor que incide en esto es la velocidad de transferencia de la memoria.

El tiempo de ciclo es el tiempo que transcurre entre dos pulsos de reloj.

La frecuencia del reloj y el tiempo de ciclo son cantidades inversas. Por

ejemplo, si el valor de frecuencia es de cuatro ciclos por segundo, el valor del tiempo de ciclo es un cuarto de segundo.

Tipo de instrucciones

Transferencia ----> MOV

Aritméticas ----> ADD, DCR, INR

Lógicas ----> XOR, OR, NAND

Rutina de secuencia (saltos, interrupciones) ----> JUMP, JZ, JNZ, JC

Control de Proceso ----> NOP, CLC

Modos de direccionamiento

Los modos de direccionamiento establecen las formas en que se puede, a nivel de las instrucciones, especificar la dirección de un operando o del lugar donde colocar el resultado de la operación correspondiente a la instrucción. Es decir, “a donde va a ir a buscar el operando o dato”

Los modos de direccionamiento más comunes son:

Implícito

No posee ningún parámetro ya que el propio código de operación se conoce la dirección de el o los operandos a los que se desea acceder con los que se quiere operar.

Ejemplo

CLC, NOP

Inmediato

En este modo en la instrucción se encuentra el propio operando (su valor). Se utiliza típicamente para constantes (ya que las instrucciones no se pueden modificar).

Direccionamiento Inmediato

Cod Operación

Operando

Ejemplo

MOV A, 9

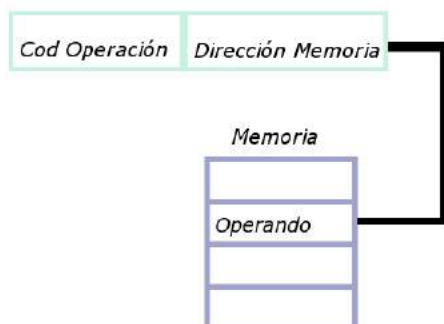
Mueva al registro A, el valor 9

Directo

El campo de operando en la instrucción contiene la dirección en memoria donde se encuentra el operando.

El operando reside en la memoria, se codifica entre corchetes para indicar que es una posición de memoria.

Direccionamiento Directo



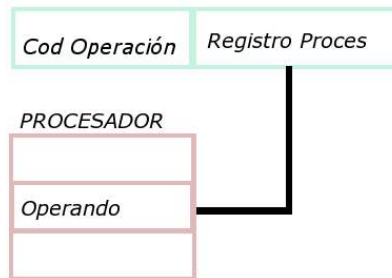
Ejemplo:

MOV A, [3D7C]

Mueve al registro A, el valor que contiene la posición de memoria 3D7C.

Por registro

El valor del operando se encuentra en un registro del microprocesador, por esta razón, esta operación es muy veloz, ya que no interviene la memoria.

Direccionamiento por Registro

Ejemplo

MOV A, B

Mueve al registro A, el valor que posee el registro B

Por registro Indirecto

Mueve un valor a un registro determinado, lo que se encuentra direccionado en un puntero. Como puntero se utiliza los registros HL.

MOV A, HL

Mueve al registro A, el contenido de la posición de memoria indicada en HL

Lenguaje Ensamblador

Es un lenguaje de programación de bajo nivel para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables. Implementa una representación simbólica de los códigos de máquina binarios y otras constantes necesarias para programar una arquitectura dada de CPU y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador. Esta representación es usualmente definida por el fabricante de hardware, y está basada en los mnemónicos que simbolizan los pasos de procesamiento (las instrucciones), los registros del procesador, las posiciones de memoria y otras características del lenguaje.

Un lenguaje ensamblador es por lo tanto específico de cierta arquitectura de computador física (o virtual). Esto está en contraste con la mayoría de los lenguajes de programación de alto nivel, que idealmente son portátiles.

Un lenguaje de programación de bajo nivel es aquel en el que sus instrucciones ejercen un control directo sobre el hardware y están condicionados por la estructura física de la computadora que lo soporta.

Microprocesador (μP)

Todos los microprocesadores utilizan los conceptos ya estudiados en esta unidad. Cada modelo difiere en características tales como capacidad de direccionamiento de memoria, puertos, nro de bits de datos, velocidad de procesamiento, etc.

Utilizamos el microprocesador Intel 8085 por su sencillez a los efectos educativos. No obstante, aplican todos los conceptos estudiados en los microcontroladores que hoy se utilizan en el mercado.

Registros del μP Intel 8085

Cada modelo de microprocesador posee características propias referidas al número y denominación de sus registros de trabajo.

En este caso, el Intel 8085 tiene disponible los registros Acumulador, B y C; D y E; H y L, SP; PC y registro de banderas (Flag Register).

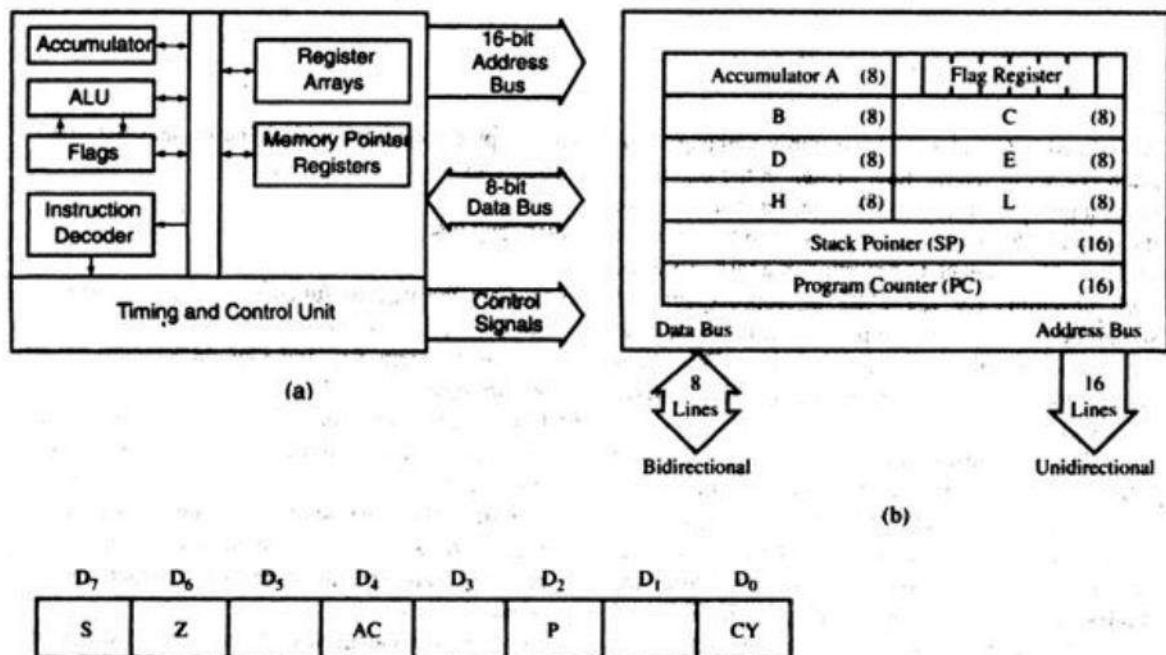
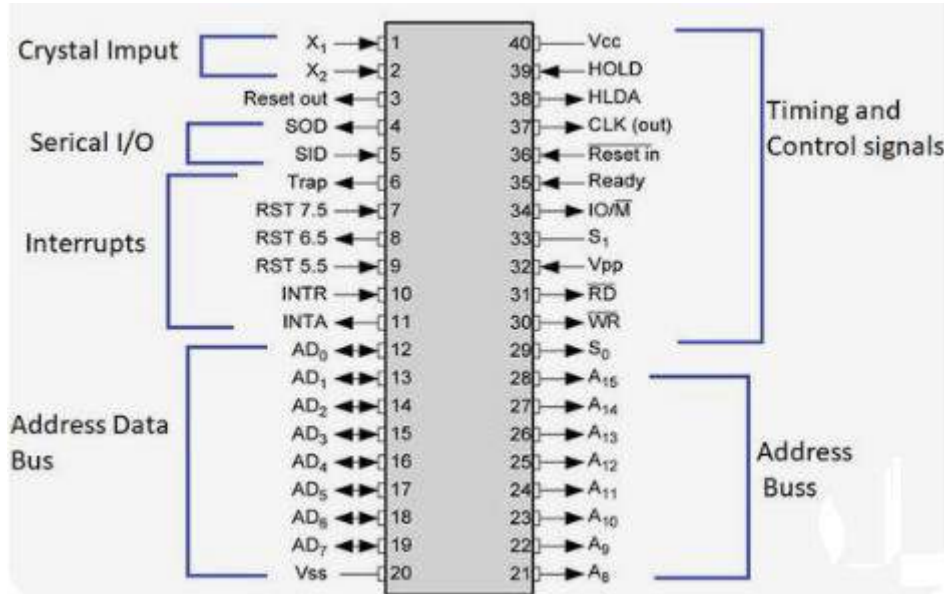


Diagrama de pines

Refiere al conexionado eléctrico con el resto de los dispositivos.



Set de instrucciones 8085

El set de instrucciones contiene la totalidad de las instrucciones que un microprocesador es capaz de realizar, con detalles específicos de cada una, como pueden ser nemónico, valor real de cada instrucción en hexadecimal, “modos de direccionamiento” de cada instrucción, banderas afectadas, ciclos de reloj insumidos para su la ejecución de cada instrucción, etc.

CONJUNTO DE INSTRUCCIONES DEL MICROPROCESADOR 8085

Descripción	Código	Nemotécnico	B*	C*	Descripción	Código	Nemotécnico	B*	C*
1 - INSTRUCCIONES PARA TRANSFERENCIA DE DATOS					Mueva el contenido del Reg. C al Reg. A	79	MOV A, C	1	4
MOVIMIENTOS AL REGISTRO B					Mueva el contenido del Reg. D al Reg. A	7A	MOV A, D	1	4
Mueva el contenido del Reg. B al Reg. B	40	MOV B, B	1	4	Mueva el contenido del Reg. E al Reg. A	7B	MOV A, E	1	4
Mueva el contenido del Reg. C al Reg. B	41	MOV B, C	1	4	Mueva el contenido del Reg. H al Reg. A	7C	MOV A, H	1	4
Mueva el contenido del Reg. D al Reg. B	42	MOV B, D	1	4	Mueva el contenido del Reg. L al Reg. A	7D	MOV A, L	1	4
Mueva el contenido del Reg. E al Reg. B	43	MOV B, E	1	4	Mueva el contenido de M al Reg. A*	7E	MOV A, M	1	7
Mueva el contenido del Reg. H al Reg. B	44	MOV B, H	1	4	Mueva el contenido del Reg. A al Reg. A	7F	MOV A, A	1	4
Mueva el contenido del Reg. L al Reg. B	45	MOV B, L	1	4	CARGA INMEDIATA DE LOS REGISTROS				
Mueva el contenido de M al Reg. B*	46	MOV B, M	1	7	Mueva data al Registro A	3E	MVI A, data	2	7
Mueva el contenido del Reg. B al Reg. B	47	MOV B, A	1	4	Mueva data al Registro B	06	MVI B, data	2	7
MOVIMIENTOS AL REGISTRO C					Mueva data al Registro C	0E	MVI C, data	2	7
Mueva el contenido del Reg. B al Reg. C	48	MOV C, B	1	4	Mueva data al Registro D	16	MVI D, data	2	7
Mueva el contenido del Reg. C al Reg. C	49	MOV C, C	1	4	Mueva data al Registro E	1E	MVI E, data	2	7
Mueva el contenido del Reg. D al Reg. C	4A	MOV C, D	1	4	Mueva data al Registro H	26	MVI H, data	2	7
					Mueva data al Registro L	2E	MVI L, data	2	7
					Mueva data a la memoria*	35	MVI M, data	2	10

En la figura se observa la parte inicial del set de instrucciones con la información de tallada. Para realizar los ejercicios de programación en

ENSAMBLADOR es necesario que te ayudes con el set de instrucciones anexo provisto en el material de estudio.

Instrucciones 8085

Las instrucciones se encuentran debidamente detalladas en el set de instrucciones. Para facilitar la comprensión se sugiere que el estudiante cuente con el set de instrucciones del microprocesador que esté trabajando. Las más comunes que se encuentran presentes en todos los microprocesadores son las siguientes:

MOV es la instrucción más simple, refiere al movimiento de datos entre dos sectores, ejemplos.

MOV A, B mueve datos desde el registro B hacia el A

MOV C, 10 mueve el valor 10 al registro C.

MOV A, [2000] mueve el contenido de la dirección de memoria 2000, hacia A

MOV A, [HL] mueve datos desde la dirección de memoria indicada por HL, hacia A

ADD se utiliza para sumar dos valores aritméticos, ejemplos:

ADD C suma el valor de C al acumulador, el resultado quedará en A.

ADD B suma A+B, el resultado en A

ADC a diferencia de ADD, suma con acarreo

ADI es la instrucción ADD, modo de direccionamiento INMEDIATO.

SUB es la operación contraria a ADD, es decir RESTA.

INR es la operación que incrementa en UNO un valor indicado, ejemplos

INR A, incrementa en uno al acumulador.

INR B, incrementa en uno al registro B.

INX B, incrementa en uno a los registros BC, tomándolos como uno de 16 bits.

INR M, incrementa en uno la posición de memoria previamente cargada en los registros HL.

DCR y DCX son las instrucciones contrarias a INR e INX, es decir decremento de valores, ejemplos:

DCR A, decrementa en uno al acumulador.

DCR B, decrementa en uno al registro B.

DCX B, decrementa en uno a los registros BC, tomándolos como uno de 16 bits.

DCX M, decrementa en uno la posición de memoria previamente cargada en los registros HL.

AND OR y XRA realizan esas operaciones lógicas entre registros. Ejemplos:

ORA B, realiza OR entre A y B

AND C, realiza AND entre A y C

CMP – CPI realizan la comparación entre diversos registros o valores fijos, con el acumulador, ejemplos

CPI 25, compara el valor de A con el número 25. Si es menor se enciende el flap de SIGNO

CMP B, compara el valor de A con B. Si es menor A que B, afecta flaps de SIGNO.

LDA, STA y sus variantes.

Load y Storage son instrucciones que se asemejan a las MOV, pero para ser utilizadas en la manipulación de los registros en 16 bits, combinando HyL, ByC, etc.

LXI B, FFFF ; carga de manera inmediata el valor FFFF en los registros B y C.

CALL – RET

Call se utiliza para saltar a una subrutina que se encuentra localizada en una zona fuera de las direcciones del programa principal. Ejemplo.

CALL E894 ; El programa salta a la dirección de memoria E894. El Stack pointer acumulará la dirección de memoria del programa principal. El PC, contador de programa, asume la nueva dirección de memoria, en este caso E894.

En la subrutina RET, que debe localizarse como última instrucción en la subrutina, permite que mediante RET, el SP devuelva al PC la dirección de memoria del programa principal.

Mapa de Direcciones

El mapa de direcciones representa todas las posiciones de memoria que es capaz de direccionar un microprocesador.

Por ejemplo, un microprocesador que posee 16 bits en su bus de direcciones, es capaz de direccionar 2^{16} , es decir 65535, (64K) posiciones de memoria.

Las posiciones iniciales siempre son ocupadas por memoria ROM, ya que el microprocesador al bootear comienza a ejecutar la instrucción ubicada en la posición CERO.

La cantidad de memoria ROM, RAM y puertos de entrada y salida, dependerá del dispositivo en el cual en MP se encuentra funcionando.

MAPA DE DIRECCIONES

Direcciones en Hexadecimal	Direcciones en Binario		Direcciones en Decimal
0000 h	0000.0000.0000.0000 b	ROM	0 d
		ROM	
		ROM	
		ROM	
		ROM	
		RAM	
		RAM	
		RAM	
		RAM	
7FFF h	0111.1111.1111.1111 b	RAM	32767 d
			posiciones libres
		PUERTO DE ENTRADA	
		PUERTO SALIDA	
FFFF h	1111.1111.1111.1111 b		65535 d

Bibliografía

Ginzburg, M.C. 2006. LA PC POR DENTRO. Edit. Biblioteca Técnica Superior. Argentina. 4º Edición.

https://es.wikipedia.org/wiki/Modo_de_direccionamiento

Orenga, Miquel Albert Manonellas, Gerard Enrique. EL PROCESADOR. (FUOC. Fundación per a la Universitat Oberta de Catalunya). PID_00177072

Leiva, Hoesé, Ferrari, Ayestarán, Romera. "Estructura y Funcionamiento de Computadoras I". Facultad de Ciencias Exactas Físicas y Naturales. Departamento de Informática. Universidad Nacional de San Juan.

Williams Stallings - Organización y Arquitectura de Computadores, Capítulos 1 y 3-4-6-9-10-11-12. Editorial Prentice Hall - 5ª edición - 2000. Biblioteca Central Ubicación: 004.22 STA o

Blázquez, J.M. "Interrupciones en la arquitectura INTEL IA32", 2004.



Atribución-No Comercial-Sin Derivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterado su contenido, ni se comercialice. Referenciarlo de la siguiente manera:
Universidad Tecnológica Nacional Facultad Regional Córdoba (S/D). Material para la Tecnicatura Universitaria en Programación, modalidad virtual, Córdoba, Argentina.