ASIGNATURA: ESTRUCTURA DE COMPUTADORES

I.TELECOMUNICACION

PROFESORA: JUANA Mª LOPEZ

DEPARTAMENTO DE AUTOMATICA

CAPITULO PRIMERO

ARQUITECTURA DEL MICROPROCESADOR 8086

ARQUITECTURA DEL MICROPROCESADOR 8086.

1. INTRODUCCION.

Los procesadores de 16 bits fueron una nueva generación de microprocesadores desarrollados para reemplazar o completar a las microcomputadoras de 8 bits de los años setenta, que fueron las que comenzaron la revolución de las microcomputadoras.

El 8086 fue diseñado para trabajar con lenguajes de alto nivel, disponiendo de un soporte hardware con el que los programas escritos en dichos lenguajes ocupan un pequeño espacio de código y pueden ejecutarse a gran velocidad. Esta concepción, orientada al uso de compiladores, se materializa en un conjunto de facilidades y recursos, y en unas instrucciones entre las que cabe destacar las que permiten efectuar operaciones aritméticas de multiplicar y dividir, con y sin signo; las que manejan cadenas de caracteres, etc.

En su momento, el 8086 junto con el 8088 fueron los microprocesadores más empleados dentro de su categoría, especialmente desde que IBM los adoptó para la construcción de su computadora personal. Muchos fabricantes de microordenadores utilizaron esta familia microcomputadora para fabricar equipos de tipo profesional. Hoy en día, la utilización del 8086 es más reducida, quedando principalmente orientado a la enseñanza, como base de los microprocesadores de la última generación.

Antes de pasar a hacer una descripción más detallada de la arquitectura interna del microprocesador, vamos a destacar brevemente las principales características del 8086:

- El 8086 dispone de instrucciones especiales para el tratamiento de cadenas de caracteres.
- Los registros del 8086 tienen una misión específica, por lo que se podría decir que cada uno de ellos tiene su propia personalidad, aunque varios comparten tareas comunes.
- El encapsulado del 8086 está formado por 40 patillas, simplificando así el hardware, aunque por contra, es necesario la multiplexación del bus de datos con el de direcciones.

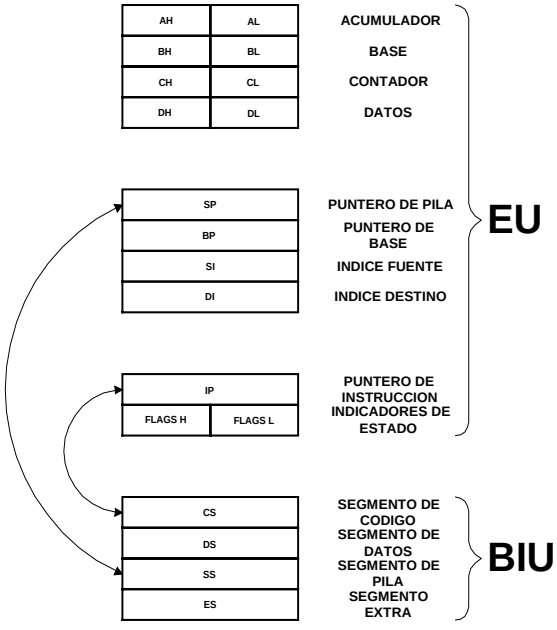
- El 8086 dispone de un conjunto de registros, denominados ‘cola de instrucciones’, en el cual se van almacenando de forma anticipada los códigos de las instrucciones, consiguiendo que este aumente su velocidad de trabajo.

- Las 20 líneas del bus de direcciones sólo permiten direccionar una memoria de 1 Megabyte.
- El 8086 requiere una señal de reloj exterior, siendo 5 y 8 Mhz las frecuencias típicas de funcionamiento.
- El 8086 dispone de una arquitectura "pipe line", es decir, que la CPU puede seguir leyendo instrucciones en los tiempos en que el bus no se utiliza.

2. ARQUITECTURA INTERNA DEL 8086.

Este microprocesador esta dividido en dos sub-procesadores. Por un lado está la "Unidad de Ejecución" (EU) encargada de ejecutar las instrucciones, la cual posee una ALU (unidad aritmético-lógica) con un registro de estado con varios flags asociados y un conjunto de registros de trabajo, y por otro está la "Unidad de Interfaz de bus" (BIU) encargada de la búsqueda de las instrucciones, ubicarlas en la cola de instrucciones antes de su ejecución y facilitar el direccionamiento de la memoria, es decir, encargada de acceder a datos e instrucciones del mundo exterior.

El 8086 contiene 14 registros de 16 bits, de los cuales, unos pertenecen a la EU, que normalmente se suelen usar para direccionamiento, y otros pertenecen a la BIU.



JUEGO DE REGISTROS DEL 8086

Los registros del 8086 podrían clasificarse en tres grupos de acuerdo con sus funciones. El grupo de datos, que es esencialmente el conjunto de registros aritméticos; el grupo de apuntadores, que incluye los registros base e índices y también el contador

de programa y el puntero de pila; y el grupo de registros de segmento, que es un conjunto de registros base de propósito especial.

El grupo de registros de datos o registros generales son registros de 16 bits, pudiéndose usar cada uno de ellos como dos registros de 8 bits. Aun siendo registros de uso general tiene asignadas unas operaciones específicas. Así, por ejemplo, el AX es el acumulador de 16 bits y usándolo a veces provoca que el ensamblador produzca un lenguaje máquina codificado en muy pocos octetos. Se emplea en multiplicaciones, divisiones, entradas/salidas, etc.; el registro BX, se utiliza como registro base para el direccionamiento de memoria; el registro CX, se utiliza como contador y almacenaje de datos y el registro DX, se utiliza para almacenar datos de 16 bits. Puede pensarse que es una extensión del registro AX para multiplicaciones y divisiones con 16 bits. Otra de sus funciones específicas es para almacenar la dirección de E/S durante algunas operaciones de E/S.

El grupo de apuntadores, es decir, punteros e índices está formado por los registros IP, SP, BP, SI, DI.

Los registros puntero son dos:

- IP como registro puntero de instrucciones conocido principalmente como contador de programa. Este contiene un valor de 16 bits que es un desplazamiento sobre la dirección del registro CS (segmento de código) que más adelante detallaremos.

- SP como registro de pila. El registro BP actúa como base de la dirección de la pila.

Los registros puntero de instrucciones (IP) y puntero de pila (SP) se encargan del control de flujo del programa.

Los registros SI y DI actúan como índices asociados al registro DS (segmento de datos).

El grupo de registros de segmento está formado por los registros CS, SS, DS y ES.

- CS (segmento de código).
- DS (segmento de datos).
- SS (segmento de pila).
- ES (segmento extra).

La importancia de dichos registros queda reflejada en la estructura de la memoria con la técnica de segmentación, que principalmente radica en el que el espacio total de memoria se divide en trozos de 64K bytes, que reciben el nombre de “segmentos”.

Las ventajas de utilizar registros de segmento son:

- Permite una capacidad de memoria de hasta 1 megabyte, aunque la dirección asociada a una instrucción sea sólo de 16 bits.
- Permiten que las partes de un programa, instrucciones, datos y pilas, tengan un tamaño mayor de 64K, mediante la utilización de más de un segmento para código, datos o pila.
- Facilitan la utilización de áreas separadas para un programa, sus datos y la pila.
- Permiten colocar un programa y sus datos en diferentes áreas de memoria cada vez que se ejecute.

Aparte de todos estos registros, el 8086 consta de un registro de estado de estado de 16 bits, aunque algunos de ellos no se utilizan. Cada uno de los bits se denomina indicador o flag, que generalmente, se modifican por las operaciones lógicas y aritméticas.

Estos indicadores son:

- SF (indicador de signo).
- ZF (indicador de cero).
- PF (indicador de paridad).
- CF (indicador de acarreo).
- AF (indicador de acarreo auxiliar).
- OF (indicador de desbordamiento).
- DF (indicador de dirección).
- IF (indicador de interrupción).
- TF (indicador de trap).

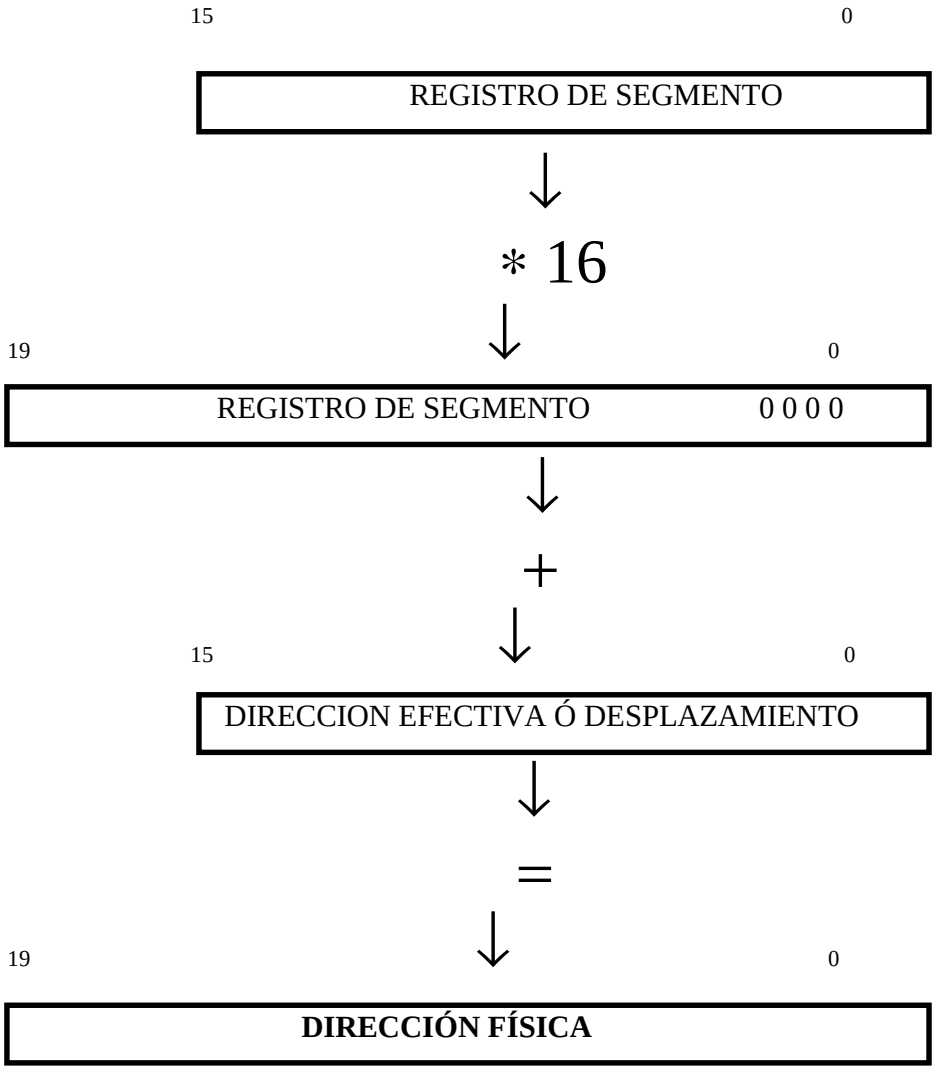
3. ESTRUCTURA DE MEMORIA DE SEGMENTACIÓN.

El 8086 usa un esquema llamado segmentación, para acceder correctamente a un megabyte completo de memoria, con referencias de direcciones de sólo 16 bits, y todo esto gracias a la utilización de registros de segmento que dividen esencialmente el espacio de memoria en segmentos de 64K de longitud, que pueden estar separados entre sí, adyacentes o superpuestos, y que comienzan en una dirección divisible por 16.

La forma en que se completan los 20 bits del bus de direcciones, disponiendo en la CPU, solamente, registros de 16 bits, se consigue de la siguiente manera:

Se parte del contenido de uno de los registros de segmento, que actúan como base. Después, se multiplica por 16 el contenido del registro de segmento, lo que, en binario, significa añadirle 4 ceros a la derecha y convertirlo en una magnitud de 20 bits. Finalmente, se suma un desplazamiento al resultado de la multiplicación anterior. Abreviadamente, la fórmula para calcular una dirección de memoria es:

Dirección Física = 16 * (registro de segmento) + desplazamiento.



De esta manera, sobre la dirección base que apunta el registro de segmento multiplicado por 16, existe un margen de 64K bytes, controlado por un desplazamiento de 16 bits.

Después de haber visto de una forma general el funcionamiento de la estructura de la memoria de segmentación vamos a ver como, más concretamente, el 8086 y en función de los registros de segmento y el resto de registros, calcula las direcciones completas de las instrucciones, posiciones de pila y datos.

Las direcciones completas de las instrucciones y de las posiciones de la pila se forman sumando el contenido de los registros IP y SP con el segmento de código (CS) y el segmento de pila (SS) respectivamente.

La dirección de un dato puede formarse mediante la suma de los contenidos de los registros BX ó BP, los contenidos de SI ó DI, y un desplazamiento. El resultado de este cálculo se denomina dirección efectiva (EA) ó desplazamiento de segmento.

La dirección definitiva del dato, sin embargo, se determina mediante la EA y el registro de segmento apropiado, el segmento de datos (DS), el segmento extra (ES) ó el segmento de pila (SS).

El uso de los diferentes segmentos significa que hay áreas de trabajo separadas para el programa, la pila y los datos. Cada área de trabajo tiene un tamaño máximo de 64K bytes y un mínimo de 0. Dado que hay cuatro registros de segmento, uno de programa (CS), uno de pila (SS), uno de datos (DS) y uno extra (ES) el área de trabajo puede llegar hasta 256K.

El programador puede determinar la posición de estos segmentos, cargando el apropiado registro de segmentación de 16 bits con la dirección de segmento apropiada. Esto se realiza normalmente al inicio del programa, pero se puede fácilmente hacer en cualquier momento mientras al programa se ejecuta, cambiando dinámicamente la dirección de estos segmentos.

Cambiando el desplazamiento, el programador puede acceder a cualquier punto de segmento. Se puede pensar en el segmento como una ampliación de memoria que forma un área de trabajo. El desplazamiento es la única parte de la dirección que aparece normalmente en los programas en lenguaje ensamblador durante las referencias a direcciones del área de trabajo.

4. SEÑAL DE RELOJ.

Al igual que los mas recientes microprocesadores, el 8086 requiere una única señal de reloj. Este microprocesador no genera su propia señal de reloj siendo necesario la utilización del generador de reloj 8284, que usa un cristal oscilador para determinar la frecuencia de señal. Intercambiando este cristal, se puede seleccionar diferentes velocidades de operación. Intel tiene una versión de 5 MHz y otra de 8 Mhz para el 8086. Estas versiones representan las velocidades más altas, recomendables para este chip.

Para un rendimiento óptimo, el 8086 requiere una señal de reloj que se mantenga a tensión alta una tercera parte del tiempo total de ciclo. Esto significa que el reloj está activo una tercera parte del tiempo y desactivado las dos terceras partes del tiempo.

5. MODOS DE DIRECCIONAMIENTO.

La forma en que se especifica un operando se denomina modo de direccionamiento, es decir, es un conjunto de reglas que especifican la localización (posición) de un dato usado durante la ejecución de una instrucción.

El 8086 tiene 25 modos de direccionamiento o reglas para localizar un operando de una instrucción.

Los modos de direccionamiento más frecuentes son los que calculan la dirección del operando mediante la suma de la dirección base de un registro segmento, multiplicado por 16 y el valor de un desplazamiento.

La gran variedad de direccionamientos proviene de las muchas formas en que se puede determinar el desplazamiento.

En general, los modos de direccionamiento del 8086 se dividen en dos grandes grupos:

1. Modos de direccionamiento de la memoria de programa.
2. Modos de direccionamiento de la memoria de datos.

5.1. Modos de direccionamiento de la memoria de programa.

En la búsqueda de una instrucción, su dirección se obtiene sumando el desplazamiento, contenido en el IP, al valor del registro de segmento CS, multiplicado por 16.

Normalmente al terminar la ejecución de una instrucción, el IP se incrementa, con lo que se pasa a direccionar la siguiente instrucción. Las instrucciones de salto y salto a subrutinas pueden modificar el contenido de IP de tres formas diferentes:

- Por direccionamiento relativo. Al contenido del IP se suma, de forma inmediata, un desplazamiento de 8 a 16 bits, con signo, proporcionado por la misma instrucción.
- Por direccionamiento directo. Se carga, en el IP, una nueva dirección presente en la instrucción.
- Por direccionamiento indirecto. El dato, obtenido por cualquiera de las formas de direccionado de la memoria de datos, es interpretado por las instrucciones de salto, como la dirección a la que se debe saltar.

5.2. Modos de direccionamiento de la memoria de datos.

- Modo inmediato. El operando se proporciona en el byte o bytes que siguen al código de operación de la instrucción.

Ejemplo: ADD CX,385Fh

- Modo de direccionado por registro. Un registro, definido por la instrucción, contiene el operando.

Ejemplo: ADD CX,AX

- Modo directo. El byte o par de bytes que siguen al código OP de la instrucción dan el desplazamiento de 8 ó 16 bits, que, sumado al contenido del registro DS, determina la dirección efectiva en la que se encuentra el dato a transferir.

Ejemplo: ADD CL,TABLA

- Modo directo indexado. El byte o par de bytes que siguen al código OP representan un desplazamiento que se suma al contenido de uno de los registros índice (DI o SI). El contenido de DS se añade al resultado de la suma, con lo que se obtiene la dirección del operando.

Incrementando o decrementando los registros índice, se puede acceder a posiciones de memoria consecutivas.

Ejemplo: ADD CX, [SI+4]

- Modo indirecto. La dirección del operando es el contenido de uno de los siguientes registros: BP, BX, DI o SI.

Ejemplo: ADD CX, [BX]

- Por registro base indexado. El desplazamiento que ha de sumarse a un registro segmento se halla sumando el contenido de un registro índice y un desplazamiento de 8 ó 16 bits, contenido en la instrucción, al contenido de un registro base.

Ejemplo: MOV AX,TABLA[BX][SI]

- Modo relativo a base: El byte o par de bytes que siguen al código OP representan un desplazamiento que se suma al contenido de uno de los registros base (BX o BP). El contenido de DS se añade al resultado de la suma, con lo que se obtiene la dirección del operando.

Ejemplo: MOV AX, [BP]+4

6. FORMATO DE LAS INSTRUCCIONES.

A semejanza con otros microprocesadores, las instrucciones del 8086 pueden clasificarse en tres grupos, según la definición que se utilice para los operandos:

- Sin operando. Por ejemplo, CLI, STI, DAA, WAIT, etc.
- Con un sólo operando. Por ejemplo, JMP, PUSH, CALL, etc.
- Con dos operandos. Por ejemplo, MOV.

Las instrucciones varían de 1 a 6 bytes en longitud. Los desplazamientos y los datos inmediatos pueden tener 8 ó 16 bits dependiendo de la instrucción. El código de operación y el modo de direccionamiento se encuentran en los primeros uno o dos bytes de la instrucción. Estos pueden ir seguidos por:

- Ningún byte adicional.
- Una EA de 2 bytes (sólo para direccionamiento directo).
- Un desplazamiento de 1 ó 2 bytes.
- Un operando inmediato de 1 ó 2 bytes.

- Un desplazamiento de 1 ó 2 bytes seguidos de un operando inmediato de 1 ó 2 bytes.
- Un desplazamiento de 2 bytes y una dirección de un segmento de 2 bytes (sólo para direccionamientos fuera del segmento).

El código de operación y el modo de direccionamiento determinan cual de estas posibilidades es la utilizada. Si un desplazamiento o un operando inmediato son de 2 bytes, siempre aparece primero el byte de menor orden.

El código de operación, que generalmente es el primer byte de la instrucción (aunque hay algunas veces en las que este byte designa un registro y otras en las que 3 bits del código de operación están en el segundo byte), es el encargado de determinar la operación, informando a su vez del tamaño de los operandos. En la mayoría de los códigos de operación hay indicadores especiales de 1 bit. Estos indicadores son:

Bit W. Nos informa del tamaño de los operandos, es decir, si se trata de byte o palabra. Con W = 0 estamos accediendo a un byte y con W = 1 a una palabra.

Bit D. Este bit es utilizado con instrucciones de dos operandos, salvo en los casos en que uno de los operandos debe ser un registro especificado en el campo REG.

Este bit nos informa si el registro especificado en el campo REG se trata del operando fuente u operando destino de instrucción especificada por el código de operación.

Para D = 0, el registro indicado por el campo REG es el operando fuente y para D = 1, el registro indicado por el campo REG es el operando destino.

Bit S. El bit S aparece junto con el bit W en instrucciones de suma inmediata registro / memoria, resta y comparación. El bit S indica el tamaño del dato u operando inmediato, el bit W nos indica el tamaño del dato u operando destino. Si SW es 00, tanto la fuente como destino tienen 8 bits, es decir, se trata de una operación de 8 bits; Si SW es 11 se trata de una operación de 16 bits con un operando inmediato de 16 bits con extensión de signo; Si SW es 01 indica una operación de 16 bits con un operando inmediato de 8 bits con extensión de signo.

Bit V. Utilizado por instrucciones de desplazamiento y rotación para determinar el número de desplazamientos.

Bit Z. Utilizado para instrucciones REP.

A continuación se muestra la combinación de bits para asignación de registros (de segmento o cualquier otro tipo).

Dirección de registro	Registros W = 1	Registros W = 0
000	AX	AL
001	CX	CL
010	DX	DL
011	BX	BL
100	SP	AH
101	BP	CH
110	SI	DH
111	DI	BH

Dirección de registro	Registro de segmento
00	ES
01	CS
10	SS
11	DS

Como se puede apreciar en estas tablas se podría producir ambigüedad al representar, por ejemplo, 00 para indicar el registro de segmento ES, 000 para indicar AX y a su byte de menor orden AL , pero esto no sucede ya que el código de operación implica el tipo de registro que indica.

Cuando el código de operación y el modo de direccionamiento ocupan 2 bytes pueden verse representado de las dos formas siguientes:

MOD	COD OP	R/M

MOD	REG	R/M

El primero de los casos es para instrucciones de un sólo operando o bien instrucciones de dos operandos pero uno de ellos implícito en el código de operación.

El segundo de los casos se trata de instrucciones de dos operandos, en cuyo caso uno de ellos viene especificado por el campo REG y el otro operando viene especificado

por los campos MOD y R/M que pueden indicar un registro o bien una posición de memoria.

A continuación se especifica una tabla donde quedan reflejados los modos de direccionamiento y registros de segmento por defecto para varias combinaciones de los campos MOD y R/M.

MOD **R/M**	**00**	**01**	**10**	**11** **W=0**	**W=1**
000 Reg. de segmento	**(BX) + (SI)** **DS**	**(BX)+(SI)+D8** **DS**	**(BX)+(SI)+D16** **DS**	**AL**	**AX**
001 Reg. de segmento	**(BX) + (DI)** **DS**	**(BX)+(DI)+D8** **DS**	**(BX)+(DI)+D16** **DS**	**CL**	**CX**
010 Reg. de segmento	**(BP) + (SI)** **SS**	**(BP)+(SI)+D8** **SS**	**(BP)+(SI)+D16** **SS**	**DL**	**DX**
011 Reg. de segmento	**(BP) + (DI)** **SS**	**(BP)+(DI)+D8** **SS**	**(BP)+(DI)+D16** **SS**	**BL**	**BX**
100 Reg. de segmento	**(SI)** **DS**	**(SI) + D8** **DS**	**(SI) + D16** **DS**	**AH**	**SP**
101 Reg. de segmento	**(DI)** **DS**	**(DI) + D8** **DS**	**(DI) + D16** **DS**	**CH**	**BP**
110 Reg. de segmento	**D16** **DS**	**(BP) + D8** **SS**	**(BP) + D16** **SS**	**DH**	**SI**
111 Reg. de segmento	**(BX)** **DS**	**(BX) + D8** **DS**	**(BX) + D16** **DS**	**BH**	**DI**

Viendo esta tabla podemos apreciar que según la combinación de MOD y R/M tendremos diferentes posibilidades para el segundo operando. Como anteriormente se ha indicado este segundo operando puede ser un registro o una posición de memoria, que principalmente va a estar en función del campo MOD. Si MOD = 11 nos indica que se trata de un registro, pudiendo ver de que tipo de registro se trata según la combinación de R/M. Si MOD es distinto de 11 nos indica una posición de memoria y la dirección efectiva se calcula según las tablas. Si nos fijamos en MOD = 00 esto significa que no hay desplazamiento a menos que R/M = 110, lo que implicaría direccionamiento directo. Si MOD = 01 indica que el tercer byte de la instrucción contiene un desplazamiento de 8 bits que se extiende automáticamente a 16 (extensión de signo) antes de ser usado para calcular la dirección efectiva, y MOD = 10 indica que los bytes tercero y cuarto contienen un desplazamiento de 16 bits.

Podemos observar en la tabla el registro de segmento que es utiliza para cada una de las combinaciones de MOD y R/M. Aunque la dirección efectiva de un operando en memoria está determinada por los campos MOD y R/M, la dirección física de 20 bits

se obtiene, como ya vimos, sumando la dirección efectiva y el contenido de un registro de segmento multiplicado por 16.

Existe un byte especial de prefijo para permitir excepciones a la utilización del registro de segmento dado por la tabla anterior, es decir, nos permite ignorar algunas de estas asignaciones pero no todas, este byte se denomina prefijo de sustitución de segmento. En particular, si la CPU usa la instrucción puntero para ayudarse a apuntar a memoria, es decir, para localizar parte de una instrucción, entonces el registro de segmento de código siempre se usa. Y si la CPU usa el puntero de pila para ayudarse a apuntar a memoria, tanto para introducir como para extraer de la pila, entonces se usa siempre el registro de segmento de pila. El caso del destino en operaciones con cadena es el único en el cual hay una restricción. La combinación del índice de destino (DI) y el segmento extra (ES) se usa siempre para calcular la dirección del destino en cualquier operación con cadenas. El byte de prefijo de sustitución de segmento puede usarse, sin embargo, para obligar a utilizar alguno de los cuatro registros de segmento en el cálculo de la dirección puente de una operación de cadenas, resumiendo, los casos específicos en los que no puede haber sustituciones son:

- El CS se utiliza siempre como registro de segmento para el cálculo de la próxima instrucción que se va a ejecutar.
- Cuando se utiliza SP siempre se utiliza SS como registro de segmento.
- Para operaciones con cadenas siempre se utiliza ES como registro de segmento del operando destino.

Los 24 modos de referenciar la dirección, usados para el acceso de datos, pueden aceptar un byte de prefijo de sustitución de segmento e ignorar sus asignaciones por defecto al segmento, usando cualquier registro de segmento. La asignación por defecto como se puede ver en la tabla es o el segmento de datos (DS) o el de pila (SS) y, de hecho, se usa siempre el segmento de datos a menos que el modo de direccionamiento use el puntero base (BP), en cuyo caso se usa el registro de pila. Intel aconseja utilizar el puntero base para acceder a datos en la pila del sistema y normalmente con llamadas a subrutinas.

Para examinar con más claridad las instrucciones máquina del 8086, vamos a ver unos ejemplos con la instrucción suma (ADD). Una suma (ADD) provoca que el contenido de la posición indicada para el operando fuente sea sumado al contenido de la posición indicada para el operando destino, y que la suma reemplace a ésta última. Esta instrucción puede adoptar uno de los tres formatos que se muestran el la siguiente figura, dependiendo del modo de direccionamiento.

a) Suma de un registro con un registro o con memoria, y almacenamiento del resultado en un registro o en una memoria.

000000DW	MOD	REG	R/M	DESP menor orden	DESP mayor orden

* El campo DESP puede ocupar uno o dos bytes, dependiendo del campo MOD.

b) Suma inmediata con registro (memoria) y colocación del resultado en un registro (memoria).

100000SW	MOD	000	R/M	DESP bajo	DESP alto	DAT bajo	DAT alto

* El campo DESP puede ocupar uno o dos bytes, dependiendo del campo MOD.
* El byte alto del campo DAT es opcional, y estará presente si S:W = 01.

c) Suma inmediata con AX(AL) y almacenamiento del resultado en AX(AL). Caso especial para el acumulador.

0 0 0 0 0 1 0 W	DATO menor orden	DATO mayor orden

* El byte DATO de mayor orden es opcional, dependiendo de si W = 1.

La siguiente figura muestra el código en el lenguaje máquina para dos instrucciones ADD, las cuales suman los contenidos de los registros BH y CL, y colocan el resultado en CL. En la primera, D = 1 indica que REG = 001 = CL es donde será almacenada la suma. MOD = 11, por lo que R/M designa un registro, el registro 111 = BH. En la segunda instrucción, D = 0, lo que hace que REG = !!! = BH sea la fuente. De nuevo MOD = 11, y R/M designa a un registro que en este caso es el 001 = CL.

	D	**W**	**MOD**	**REG**	**R/M**
000000	**1**	**0**	**11**	**001**	**111**

D = 1 Indica que el destino es un registro.
W = 0 Operandos de byte (suma en 8 bits).
MOD = 11 La fuente será un registro que quedará determinado por R/M.
REG = 001 Indica el registro CL.

R/M = 111 Indica el registro BH.

a) REG indica destino.

	D	**W**	**MOD**	**REG**	**R/M**
000000	**0**	**0**	**11**	**111**	**001**

D = 0 Indica que la fuente es un registro.
W = 0 Operandos de byte.
MOD = 11 La fuente será un registro que quedará determinado por R/M.
REG = 111 Indica el registro BH.
R/M = 001 Indica el registro CL.

b) REG indica fuente.

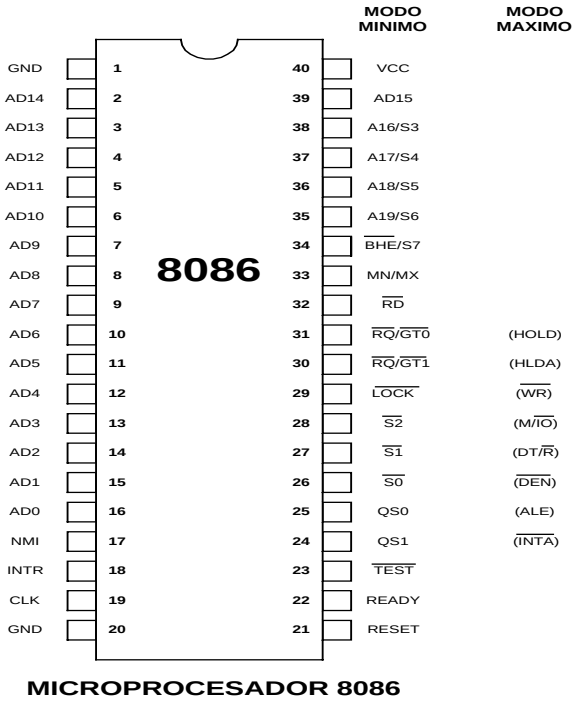
El siguiente ejemplo indica una suma de una posición de memoria con un registro. La dirección efectiva se obtiene sumando los contenidos de BX y DI al desplazamiento de 16 bits (W = 1), que es 2345. Si (BX) = 0892 y (DI) = 59A3, entonces EA = 0892 + 59A3 +2345 = 857A.

	D	W	MOD	REG	R/M	DESPLAZAMIENTO
000000	**0**	**1**	**10**	**010**	**001**	**01000101 00100011**

D = 0 Indica que la fuente es un registro.
W = 1 Operandos de palabra.
MOD = 10 La fuente que será memoria quedará determinada por el campo R/M.
REG = 010 Indica el registro DX.
R/M = 001 Indica una posición de memoria que será:
EA = (BX) + (DI) + desplazamiento de 16 bits.

c) Suma de un registro con memoria.

7. DIAGRAMA DE CONEXIONADO DEL 8086. SEÑALES Y TERMINALES.



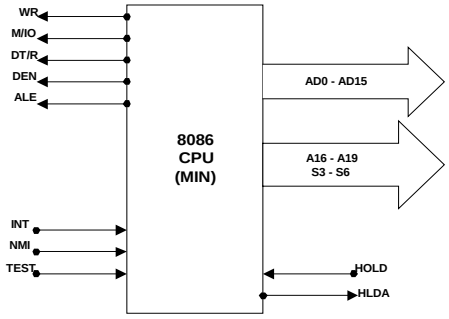
MICROPROCESADOR 8086

El 8086 puede configurarse de dos formas distintas: el modo máximo y el modo mínimo. El modo queda determinado al colocar el terminal MN/MX a tierra o a la tensión de alimentación.

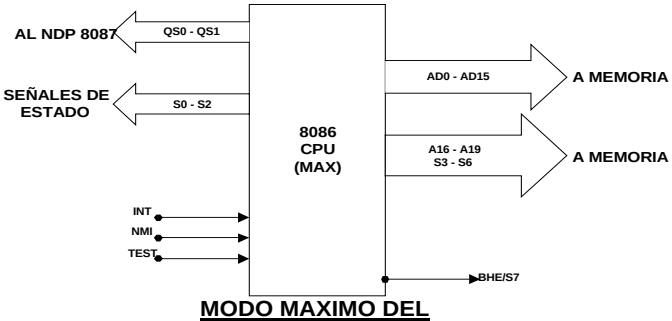
En modo mínimo no admite la multitarea, mientras que en modo máximo es capaz de soportar un bus local, para ampliar directamente el 8086, y un bus de sistema MULTIBUS, que permite configuraciones con varios procesadores, más concretamente el 8086 debe estar en modo máximo si quiere trabajar en colaboración con el procesador de datos 8087 y el procesador de entrada / salida 8089.

En el modo máximo, el 8086 depende de otros chips adicionales como es el controlador de bus 8288 para generar el conjunto completo de señales de control de bus. El modo mínimo permite al 8086 trabajar de una forma más autónoma.

La figura siguiente muestra un esquema de ambos modos.



MODO MINIMO DEL 8086



MODO MAXIMO DEL 8086

En ambos modos, las señales del 8086 se pueden agrupar de la siguiente manera:

- Alimentación.
- Reloj.
- Control y estado.
- Direcciones.
- Datos.

Hay tres terminales para la alimentación: tierra (GND) en los terminales 1 y 20, y una tensión de entrada de 5 voltios (Vcc) en el terminal 40. El terminal de tierra es tierra a la vez para la alimentación y para las señales.

Cuenta con una entrada de la señal de reloj (CLK) en el terminal 19.

El 8086 cuenta con 20 bits de dirección. Los 4 bits más significativos de la dirección comparten terminales con algunas de las señales de estado. Los 16 bits menos significativos son multiplexados tanto para las direcciones como para los datos, es decir,

en ciertos instantes tales terminales conducen parte de una dirección, y en otros son transmitidos los datos.

Estos terminales pueden llevar información de una dirección e información sobre el estado y los datos. El latch 8282 está diseñado para seleccionar la información sobre la dirección de dichos terminales en el instante preciso e ignorar lo referente al estado y los datos.

Hay varios grupos de control y señales de estado.

El terminal **MN / MX** controla si el procesador está en modo mínimo o máximo, conectándolo a tierra o a una tensión de 5 voltios.

Del **S0 al S7** son señales de estado en los terminales 26, 27, 28, 38, 37, 36, 35, 34 respectivamente. En ciertos momentos son salidas del procesador. En otros momentos aparecen otras señales distintas en los mismos terminales. Mirando el estado pueden decirse cosas tales como el tipo de acceso al bus (lectura o escritura, memoria o E/S), el registro de segmento en uso y el estado del sistema de interrupciones. **S0, S1 y S2** son sólo accesibles en modo máximo, en cuyo caso se introducen en los chips controladores de bus 8288. Estas señales decodifican el estado del procesador, de acuerdo con la siguiente tabla.

S2	S1	S0	ESTADO DE LA CPU
0	**0**	**0**	RECONOCIMIENTO DE INTERRUPCION
0	**0**	**1**	LECTURA PUERTA I/O
0	**1**	**0**	ESCRITURA PUERTA I/O
0	**1**	**1**	PARO (HALT)
1	**0**	**0**	CODIGO DE ACCESO A INSTRUCCION
1	**0**	**1**	LECTURA MEMORIA
1	**1**	**0**	ESCRITURA MEMORIA
1	**1**	**1**	PASIVO. NO OPERA.

El controlador de bus 8288 genera a partir de estas, otras importantes señales de control.

En el modo mínimo no es preciso el controlador de bus 8288 puesto que el propio procesador genera por si sólo algunas de estas señales de control.

La señal **RD** es una señal de estado generada por el procesador sobre el terminal 32. Indica un ciclo de lectura de memoria o entradas y salidas.

La señal **READY** que se encuentra en el terminal 22, es una entrada de los dispositivos externos (memoria o controladores E/S) y su función es adaptar las velocidades de memoria y periféricos a la CPU. Esta señal pasa a través del generador de pulsos 8284 para sincronizarse con la señal de reloj.

La señal **READY** trabaja de la siguiente forma: Si se ha seleccionado un dispositivo externo para lectura o escritura y todavía no está preparado para completar la transferencia de datos, pone a cero la línea de señal READY. El procesador ve esta señal y añade ciclos extras de << espera >> hasta que dicha señal se pone a su nivel normal, es decir, a 1 indicando que el dispositivo externo está preparado para realizar la transferencia. Acabada la transferencia las actividades del procesador continúan normalmente.

La señal **RESET** que se encuentra en el terminal 21, es otra de las entradas que también pasa por el generador de pulsos 8284 para sincronizarse con la señal de reloj. Se utiliza para inicializar el procesador borrando la cola de instrucciones y ciertos registros tales como los indicadores, segmento de datos (DS), segmento de pila (SS), segmento extra (ES) poniéndolos a cero. El puntero de instrucciones (IP) y el segmento de código (CS) los carga con FFFFH.

Los terminales **NMI** (**N**on-Maskable **I**nterrup : Interrupción no enmascarable), terminal 17, e **INTR** (**INT**errupt **R**equest : petición de interrupción), terminal 18, son parte del sistema de interrupciones del 8086. Un pulso en el terminal NMI provoca una interrupción especial, llamada *interrupción tipo 2.* Una señal en el terminal INTR causará una respuesta de interrupción de tipo general. El término << no enmascarables>> se refiere al hecho de que la interrupción generada por el terminal NMI no se puede activar o desactivar vía un software a la CPU. Las interrupciones generales por INTR pueden desactivarse vía software.

El terminal **BHE / S7** se utiliza como ayuda en la interfaz de los dispositivos de 8 bits con el bus de datos de 16 bits. Su funcionamiento es el siguiente: Si la línea de dirección 0 es 0 (indicando una dirección par), la señal BHE especifica si se está direccionando una palabra entera o un sólo byte. BHE igual a 0 significa que se trata de una palabra, BHE igual a 1 significa que es un byte. Si la línea de dirección 0 es 1 (indicando una dirección impar), el 8086 direcciona siempre un byte y no una palabra. En este caso BHE es 0.

La señal **M / IO** (terminal 28) informa al sistema cuando el microprocesador requiere acceso a la memoria o al espacio de E/S, es decir, indica la realización de una operación sobre memoria ó sobre entrada / salida.

El terminal **WR** de selección de escritura (terminal 29) indica que los datos están disponibles en las líneas de datos, es decir, se trata de la señal que indica un ciclo de escritura de la CPU.

El terminal **DT / R** (terminal 27) cuya misión es la recepción y transmisión de datos, es decir, para controlar la transferencia de datos, el 8086 precisa de la colaboración del circuito auxiliar 8286 (8287). Este se gobierna por la señal DT / R, que indica el sentido del movimiento de la información (transmisión o recepción).

DEN (terminal 26) igual que la anterior sirve para controlar la transferencia de datos y mas concretamente confirma la validación de los datos.

ALE (terminal 25). Esta señal activa el latch 8282 cuando viene una dirección por las líneas AD0-AD15, es decir, sirve para controlar el multiplexado de datos y direcciones.

INTA (terminal 24). La función de esta señal es el reconocimiento de interrupciones. (Se hablará más en el apartado de interrupciones).

Las señales **HOLD** y **HLDA** son parte del propio sistema de control de bus del 8086. **HOLD** (terminal 31) indica la petición del bus por un periférico exterior. Cuando otro procesador o un aparato como un controlador DMA quiere acceder al control del bus, manda una señal al 8086 a través de la línea HOLD. Cuando está preparado para hacerlo, el 8086 pone sus líneas de datos / direcciones y la mayoría de las líneas de control en el estado de alta impedancia. Al mismo tiempo, envía la señal **HLDA** (terminal 30) para indicar que el bus está libre. El otro aparato puede usar ahora el bus. Cuando finaliza con el bus, envía una señal a la línea HOLD. Inmediatamente después de recibir la señal, el 8086 reanuda el uso del bus.

Las señales **RQ / GT1** y **RQ / GT0** (terminales 30 y 31 respectivamente) se corresponden con las señales HOLD y HLDA del modo mínimo. Se utiliza para liberar el bus y reconocer la acción.

Las señales **QS1** y **QS0** (terminales 24 y 25 respectivamente) son señales de estado de las colas de instrucciones. Sólo son utilizables en modo máximo. El procesador de datos 8087 utiliza estas señales para coordinarse con el 8086.

La señal **TEST** (terminal 23) se utiliza para enlazar el 8086 con un procesador paralelo, sincronizando el procesador principal con los otros.

8. CICLO DEL BUS DEL 8086.

El 8086 se comunica con los elementos externos a través del bus del sistema. Para transferir datos o buscar instrucciones, realiza los llamados " *ciclos de bus* " . El ciclo de bus del 8086 consta de 4 periodos de reloj, llamados estados T. Durante el

primer estado (T1), el 8086 extrae una dirección por las 20 líneas del bus multiplexado de direcciones, datos y estado. Esta dirección se considera válida cuando se produce un flanco descendente en la señal ALE. Esta última, en un sistema mínimo, es generada por el 8086, mientras que en uno máximo se produce en el controlador de bus 8288.

La señal S2-M/IO, indica si se está realizando un acceso a memoria o a dispositivos de entrada / salida.

Durante el segundo estado (T2), la dirección del bus multiplexado desaparece, y las señales S3, S4, S5 y S6 toman el control por las 4 líneas de más peso. Estas aportan la información siguiente:

S4	S3	
0	**0**	***Dirección relativa al segmento extra.***
0	**1**	***Dirección relativa al segmento Stack.***
1	**0**	***Dirección relativa al segmento de código CS o ninguna.***
1	**1**	***Dirección relativa al segmento de datos DS.***

S5 indica el estado del flag de interrupción; cuando S6 es 0, significa que la CPU está actuando sobre el bus.

Al mismo tiempo, en un ciclo de lectura, las 16 líneas de menos peso del bus multiplexado quedan en estado de alta impedancia, mientras que si el ciclo es de escritura, el dato aparece por estas líneas.

Durante los estados T1 y T2, se genera la información referente a la dirección del dato, el sentido de la transferencia (si entra o sale de la CPU) y si es lectura o escritura, para lo cual se activan las señales DEN, DT / R, RD Y WR.

Durante el tercer período de reloj, T3, las 4 líneas de más peso del bus de direcciones siguen conteniendo la información sobre el estado. En una operación de escritura, el dato sigue presente en las 16 líneas de menos peso del bus multiplexado. En un ciclo de lectura, la información existente en las 16 líneas de menos peso del bus multiplexado es considerado válido. Si el dispositivo no es capaz de transferir datos a la velocidad requerida, deberá indicarlo, introduciendo un nivel 0 por la línea READY. Esta señal es reconocida en el último semiciclo de T·, denominado TW. Cuando esto ocurre, la CPU entra en un estado de espera (WAIT). Cuando haya terminado la transferencia, el dispositivo mandará un nivel 1 por READY y comenzará la ejecución del cuarto período de reloj, T4.

En el estado T4, todas las líneas de control de memoria y entradas y salidas se desconectan del bus del sistema. Sobre el bus del sistema aparece un ciclo, constituido

por una serie de eventos asíncronos que seleccionan el dispositivo o posición de memoria, mediante una dirección junto a una señal de lectura o escritura que acompaña al dato.

El 8086 únicamente ejecuta un ciclo de bus cuando comienza la búsqueda de una instrucción o cuando un operando debe transferirse, entre el 8086 y los dispositivos de entrada y salida o memoria.

9. LAS INTERRUPCIONES.

Una estructura de interrupción es una forma de que el procesador provea un servicio rápido y uniforme para la E/S, correcciones y ciertos tipo de error. En general, el procesador continúa con su trabajo habitual hasta que ocurre una interrupción, en cuyo momento salva su estado actual (puntero de instrucción, segmento de código e indicadores) ejecuta una rutina especial, y entonces vuelve a lo que estaba haciendo antes. Se puede ver la interrupción como una llamada a una subrutina y la rutina especial de interrupción como el cuerpo de la subrutina. Las principales diferencias entre subrutinas e interrupciones son:

- Las subrutinas son llamadas únicamente por instrucciones software, mientras que las interrupciones pueden ser invocadas tanto por software como por hardware.
- Las subrutinas únicamente deben salvar la dirección de retorno, mientras que las interrupciones guardan dicha dirección y el estado de todos los indicadores.
- Las subutinas necesitan tener un medio de pasar datos al programa principal y viceversa, mientras que esto no es necesario para las interrupciones.

La estructura de interrupciones del 8086 utiliza una tabla de 256 posiciones de 4 octetos cada una, la cual está en el inicio de la memoria. Cada una de estas posiciones de la tabla de interrupciones puede cargarse con un puntero a diferentes rutinas de la memoria principal. Estos punteros contienen el nuevo contenido del segmento de código (2 octetos) y el puntero de instrucciones (2 octetos) para la rutina que puede estar localizada en cualquier parte de la memoria. A cada uno de estos punteros de 4 octetos se le asigna un número del 0 al 255, según su posición en la memoria. A este número se le llama ***tipo***. Al tipo de interrupción 0 se le asigna la posición de memoria 0 y así sucesivamente hasta la posición 1.020. En general, al tipo *n* se le asigna la posición *4n* de memoria.

Cada tipo de interrupción puede ser llamado tanto por hardware como por software. Esto hace posible probar vía software las interrupciones hardware. Hay un bit de control, llamado indicador de interrupción (IF) que controla si el 8086 responde o no a las interrupciones externas. Este bit puede activarse o desactivarse por medio de las órdenes de activar interrupciones (STI) y el de borrar interrupción (CLI). Esto abre y cierra la puerta para interrupciones de la CPU.

Las interrupciones hardware actúan de la siguiente forma:

Cuando un dispositivo externo necesita servicio, produce una señal en la línea de petición de interrupción (INTR) del 8086. Si el 8086 puede responder (interrupción activa), envía una señal de *recibido*, bien de forma directa (en modo mínimo), o bien vía el controlador de bus 8288 (en modo máximo).

El dispositivo externo indica, mediante un octeto en el bus de datos, que tipo de interrupción desea. El 8086 usa este número para localizar el puntero de la tabla de interrupciones. A continuación, el 8086 salva las condiciones de sus indicadores y un puntero en la pila con la dirección de retorno, y carga el segmento de código y el puntero de instrucciones desde la tabla de interrupción. Esto hace que el procesador ejecute la rutina de servicio. Al final de dicha rutina debe haber una instrucción de retorno de interrupción (IRET).Esta interrupción restaura los indicadores, el segmento de código y el puntero a la dirección de retorno. Se debe usar la instrucción IRET ya que una instrucción retorno *normal* no restauraría los indicadores, por lo que no sería adecuada para estos propósitos (de otra manera se rompería la pila). En este punto, el procesador ha vuelto a sus operaciones normales.

Hay una clase especial de interrupciones llamadas ***interrupciones no enmascarables.*** El término *no enmascarable* se refiere al hecho de que esta clase de interrupciones no puede desactivarse limpiando el indicador de interrupciones. Esta clase de interrupciones genera una interrupción del tipo 2; o sea, su puntero se encuentra en la dirección 2 * 4 = 8 de memoria. Las interrupciones no enmascarables se reservan para casos de emergencia como fallos de potencia o errores de memoria.

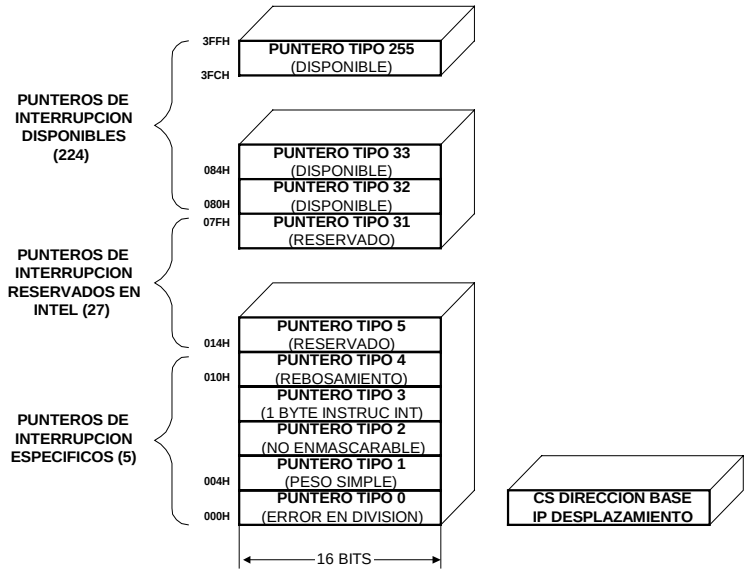


TABLA DE PUNTEROS DE INTERRUPCION DEL 8086

10. LOS CHIPS DE SOPORTE DEL 8086.

En este apartado vamos a ver una serie de chips que proveen al 8086 de circuitería para tres tipos de funciones:

- Lógica para generación de pulsos de reloj.
- Lógica para la interfaz de bus.
- Lógica para la interfaz de controladores.

Las funciones de estos chips son vitales para el sistema, como la sincronización, la conexión de los procesadores con el resto del sistema y la conexión de la computadora con el mundo exterior.

10.1. Generador de pulsos de reloj 8086

Cualquier sistema basado en el 8086 requiere una lógica adicional encargada de generar las señales de sincronización para todo el sistema. El generador de pulsos de reloj 8284 de Intel, junto con el cristal oscilador externo, es un chip diseñado específicamente para estas tareas.

El 8284 es un chip con 18 terminales que se utiliza para generar los pulsos del 8086 y sus periféricos. Los pulsos de reloj determinan la velocidad de funcionamiento del sistema. La velocidad máxima estándar para estos procesadores es una frecuencia de reloj de 5 Mhz, es decir, 200 nanosegundos por ciclo, aunque algunos chips particulares

funcionan a 8 Mhz, o lo que es lo mismo, 125 nanosegundos por ciclo. Tanto los chips estándar, como las versiones especiales más rápidas admiten teóricamente una velocidad máxima de 2 Mhz, aunque en la práctica funcionan incluso a frecuencias menores.

El generador de pulsos 8284 necesita un cristal oscilador, o una señal lógica externa como fuente de frecuencia. La opción se especifica conectando un terminal o bien a tierra, o bien a la fuente de alimentación. La fuente de frecuencia proporciona una frecuencia triple de la señal resultante del 8284. Poniendo un cristal oscilador u otro, la frecuencia obtenida puede variar entre 5 Mhz hasta casi los 8 Mhz.

Para conseguir un rendimiento óptimo de los procesadores, los pulsos de reloj generados por el 8284 se mantienen a tensión alta durante un 33 por 100 del período. Como la frecuencia de la fuente es triple que la de salida del 8284, es relativamente fácil generar este tipo de onda.

Normalmente salen tres señales del generador de pulsos de reloj 8284 hacia el procesador. Estas tres señales son CLK (señal de reloj), RESET y la señal de READY. Las dos últimas señales están sincronizadas con CLK.

La misión de la señal RESET es la de reinicializar los valores de la computadora. Esta señal es imprescindible para solventar los casos en que un programa se mete en un bucle infinito, o que un ruido de alimentación afecte a partes del programa en curso.

La función de la señal READY es la de sincronizar el procesador con los dispositivos externos más lentos. La señal READY va desde el dispositivo externo al procesador, pasando a través del generador de pulsos de reloj. Cuando el procesador quiere acceder a un dispositivo que no está preparado para la transferencia, el dispositivo envía un 0 por la línea READY. Cuando el procesador recibe esta señal, entra en un ciclo de espera hasta que aparece un 1 por dicha línea. Sólo entonces continúa el programa.

El 8284 genera otra señal que es PCLK (reloj periférico), que funciona a la mitad de la frecuencia de la señal CLK, con un ciclo de trabajo del 50 por 100. Está previsto para sincronizar aquella lógica que requiera esta forma de temporización.

10.2. Controlador de Bus 8288

Antes de comenzar a describir con más detalle el controlador de bus 8288 vamos a explicar en lo que consiste la función de dicho dispositivo que no es otra que la de actuar de interfaz entre el procesador y el bus de control.

La lógica de interfaz de bus es necesaria por dos razones:

- Las señales de los procesadores pueden no ser lo suficientemente potentes para controlar el resto del sistema.

- Las señales producidas por los procesadores puede que no correspondan directamente a las señales que necesita el resto del sistema.

El Controlador de Bus 8288, el Transceptor de Datos Octal 8286, y el Latch Octal 8282 se utilizan para solventar estos problemas de interfaz en un sistema 8086 en modo máximo. Además el Selector de Bus 8289 se usa como interfaz de los grupos de procesadores con el bus principal del sistema.

Además de amplificar las señales del 8086 estos chips resuelven el problema del exceso de señales para los 40 terminales del procesador 8086.

Centrándonos en el Controlador de Bus 8288 podemos decir que es un chip de 20 terminales, y que como hemos dicho anteriormente es el encargado de realizar el interfaz entre el procesador y el bus de control. Decodifica las señales de estado S0, S1 y S2 del 8086 en modo máximo y genera un conjunto completo de señales de control, como la de control de lectura en memoria (MRDC), control de lectura de E / S (IORC), control de escritura en memoria (MWTC), control de escritura de E / S (IOWC), latch de direcciones disponible (ALE) y datos disponibles (DEN). Algunas de estas señales de control, como las de lectura y escritura, tienen como destino el bus del sistema, mientras que otras, tales como las de direcciones o datos disponibles, son señales destinadas a los otros chips de interface del procesador con los otros subbuses (el Transceptor 8286 y el Latch Octal 8286). Hay también algunas entradas al 8288 de otros dispositivos. Las siguientes tablas muestran las señales de bus producidas por el Controlador de Bus 8288 en respuesta a las señales de estados de los procesadores.

S2	*S1*	*S0*	*Estados del procesador*	*Ordenes al bus del 8288*
0	**0**	**0**	**Reconocimiento de interrupción**	**INTA**
0	**0**	**1**	**Leer puerto de E / S**	**IORC**
0	**1**	**0**	**Escribir puerto de E / S**	**IOWC**
0	**1**	**1**	**Para**	**ninguna**
1	**0**	**0**	**Código de acceso**	**MRDC**
1	**0**	**1**	**Leer memoria**	**MRDC**
1	**1**	**0**	**Escribir memoria**	**MWTC**
1	**1**	**1**	**Pasivo**	**ninguna**

MRDC	**Orden de lectura en memoria**
MWTC	**Orden de escritura en memoria**
IORC	**Orden de lectura de E / S**
IOWC	**Orden de escritura de E / S**
INTA	**Reconocimiento de interrupción**

10.3. Transceptor 8286

El transceptor 8286 es un chip de 20 terminales cuya misión es servir de interfaz entre el procesador y el bus de datos. Se utiliza como almacenamiento intermedio para los datos que llegan y salen del procesador, y es necesario por varias razones. Por ejemplo, algunas veces las líneas de datos del procesador no pueden suministrar la corriente necesaria para la carga de un dispositivo externo, y tienen que amplificarse antes de llegar al bus del sistema. Otro uso del transceptor es el de convertir las señales bidireccionales que ciertos buses de datos externos necesitan en señales unidireccionales. Una tercera posibilidad es la de ayudar a distinguir las señales de datos sobre las señales de direcciones. Obsérvese que en el 8086 las señales de datos y direcciones están *multiplexadas* en los mismos terminales.

10.4. Latch Octal 8282

El latch Octal 8282 es otro chip también de 20 terminales, encargado de hacer de interfaz entre las líneas multiplexadas de datos / direcciones del 8086 y el bus de direcciones del sistema. El 8282 espera hasta que la información sobre la dirección aparece en tales terminales, y entonces toma la información y la mantiene sobre el bus de direcciones del sistema. Con la línea de Latch de dirección disponible (ALE), el procesador , en modo mínimo, o el Controlador de Bus 8288, en modo máximo, le dice al Latch Octal cuando debe tomar esta información.

Todos estos chips de interfaz pueden llevarse a un estado de alta impedancia (desconectados eléctricamente del bus) cuando otros procesadores o controladores poseen el control del bus.

Existen otros dispositivos tales como el selector de bus 8289 que se utiliza en sistemas grandes basados en el 8086. Realiza una conexión, o interfaz, entre el 8086 y un bus del sistema con otros procesadores conectados.