

Arquitectura del procesador 8086-8088



Monografias.com

Arquitectura en pipeline

Los procesadores previos al 8086, estaban limitados en su desempeño por la necesidad de realizar los dos pasos principales de ejecución del procesador:

Fetch/Execute, en forma secuencial. Es decir, no se puede ejecutar una instrucción hasta que se traiga de memoria (Fetch); y no podían traerse instrucciones de memoria mientras ejecutaba una instrucción, pues el procesador estaba ocupado.

Resumiendo, un alto porcentaje del tiempo, el procesador estaba ocupado haciendo Fetch, cuando su función debiera ser ejecutar las instrucciones. La capacidad de ejecutar instrucciones sólo se ocupaba en un bajo porcentaje.

Arquitectura en pipeline

- Para solucionar ésto, Intel desarrolló la arquitectura en *pipeline* del *Fetch/Execute*, en la cual simplemente se divide la tarea en dos secciones:
una encargada del *Fetch* (BIU)
y otra del *Execute* (EU).
 - De esta manera, existen circuitos separados para cada función, los cuales trabajan en paralelo.
 - Si bien el proceso aún es secuencial, solamente al principio se requiere desperdiciar tiempo en el *Fetch*.
 - A partir de ahí, *Fetch* va adelante del *Execute*, y trae instrucciones al procesador mientras este ejecuta las anteriores.

Arquitectura en pipeline

Para solucionar ésto, Intel desarrolló la arquitectura en pipeline del Fetch/Execute, en la cual simplemente se divide la tarea en dos secciones:

una encargada del Fetch (BIU)

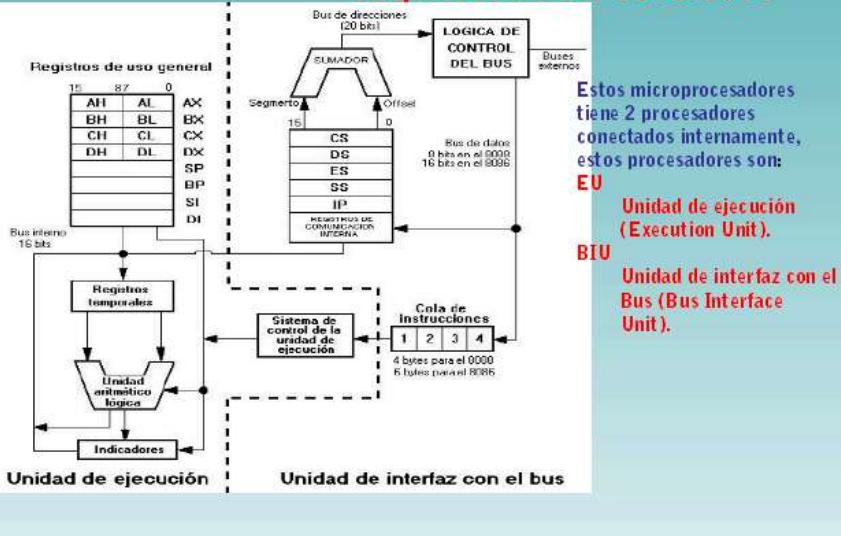
y otra del Execute (EU).

De esta manera, existen circuitos separados para cada función, los cuales trabajan en paralelo.

Si bien el proceso aún es secuencial, solamente al principio se requiere desperdiciar tiempo en el Fetch.

A partir de ahí, Fetch va adelante del Execute, y trae instrucciones al procesador mientras este ejecuta las anteriores.

Arquitectura del 8086/8088



Arquitectura del 8086/8088

Estos microprocesadores tiene 2 procesadores conectados internamente, estos procesadores son:

EU

Unidad de ejecución (Execution Unit).

BIU

Unidad de interfaz con el Bus (Bus Interface Unit).

La unidad de ejecución (EU)

- La **unidad de ejecución (EU)** se encarga de realizar las **operaciones aritméticas y lógicas**, además de proporcionar las **direcciones lógicas** al **BIU** –una dirección lógica está formada de dos direcciones, la primera indica el segmento en el cual se está trabajando y la otra indica el offset, o desplazamiento, de este segmento; esto es **segment:offset**–.
- La **EU** tiene una **unidad aritmética lógica (ALU)** de 16 bits, un **registro de banderas** y un **conjunto de registros de propósito general**. El **registro de banderas** contiene **6 banderas de estado** (que la **EU** manipula para **indicar el estado del resultado de una operación lógica o aritmética**) y **3 banderas de control** que se **pueden manipular por los programas para alterar las operaciones del procesador**.

La unidad de ejecución (EU)

La unidad de ejecución (EU) se encarga de realizar las operaciones aritméticas y lógicas, además de proporcionar las direcciones lógicas al BIU –una dirección lógica está formada de dos direcciones, la primera indica el segmento en el cual se está trabajando y la otra indica el offset, o desplazamiento, de este segmento; esto es **segment:offset**–.

La EU tiene una unidad aritmética lógica (ALU) de 16 bits, un registro de banderas y un conjunto de registros de propósito general. El registro de banderas contiene 6 banderas de estado (que la EU manipula para indicar el estado del resultado de una operación lógica o aritmética) y 3 banderas de control que se pueden manipular por los programas para alterar las operaciones del procesador .

La Unidad de Interfase con el Bus (BIU)

- Es la responsables de la comunicación externa del procesador. Esta unidad de proceso se encarga de traducir las direcciones lógicas a direcciones físicas de la memoria con ayuda de los registros de segmento DS, SS, ES, CS e IP.
- Para convertir una dirección lógica en una dirección física el BIU corre el valor del registro de segmento 4 posiciones a la izquierda (que es equivalente a multiplicar por 16h) y suma el valor del offset para obtener un valor de 20 bits necesarios, como se muestra en la figura.

La Unidad de Interfase con el Bus (BIU)

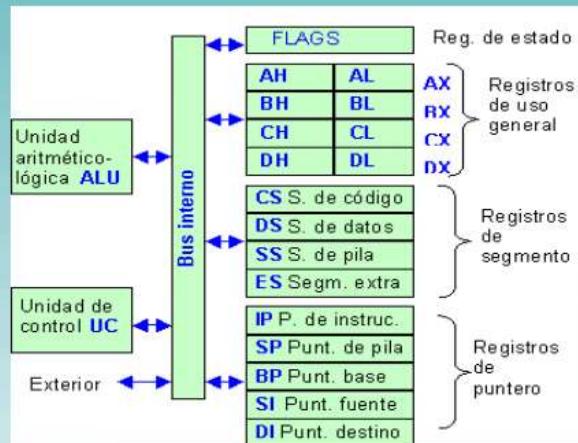
Es la responsables de la comunicación externa del procesador. Esta unidad de proceso se encarga de traducir las direcciones lógicas a direcciones físicas de la memoria con ayuda de los registros de segmento DS, SS, ES, CS e IP.

Para convertir una dirección lógica en una dirección física el BIU corre el valor del registro de segmento 4 posiciones a la izquierda (que es equivalente a multiplicar por 16h) y suma el valor del offset para obtener un valor de 20 bits necesarios, como se muestra en la figura.

MODELO DE PROGRAMACIÓN DEL 8086.

MODELO DE PROGRAMACIÓN DEL 8086.

El modelo de programación del 8086



El modelo de programación del 8086

Registros de datos o de propósito general

AX, BX, CX, DX: pueden utilizarse bien como registros de 16 bits o como dos registros separados de 8 bits (byte superior e inferior) cambiando la X por H o L según queramos referirnos a la parte alta o baja respectivamente. Por ejemplo,

AX se descompone en AH (parte alta) y AL (parte baja). Evidentemente, ¡cualquier cambio sobre AH o AL altera AX!

AX = Acumulador.

Es el registro principal, es utilizado en las instrucciones de multiplicación y división y en algunas instrucciones aritméticas especializadas, así como en ciertas operaciones de carácter específico como entrada, salida y traducción. Obsérvese que el 8086 es suficientemente potente para realizar las operaciones lógicas, la suma y la resta sobre cualquier registro de datos, no necesariamente el acumulador.

Registros de datos o de propósito general

AX, BX, CX, DX: pueden utilizarse bien como registros de 16 bits o como dos registros separados de 8 bits (byte superior e inferior) cambiando la X por H o L según queramos referirnos a la parte alta o baja respectivamente. Por ejemplo,

AX se descompone en AH (parte alta) y AL (parte baja). Evidentemente, ¡cualquier cambio sobre AH o AL altera AX!

AX = Acumulador. Es el registro principal, es utilizado en las instrucciones de multiplicación y división y en algunas instrucciones aritméticas especializadas, así como en ciertas operaciones de carácter específico como entrada, salida y traducción. Obsérvese que el 8086 es suficientemente potente para realizar las operaciones lógicas, la suma y la resta sobre cualquier registro de datos, no necesariamente el acumulador.

Registros de propósito general

BX = Base.

Se usa como registro base para referenciar direcciones de memoria con direccionamiento indirecto, manteniendo la dirección de la base o comienzo de tablas o matrices. De esta manera, no es preciso indicar una posición de memoria fija, sino la número BX (así, haciendo avanzar de unidad en unidad a BX, por ejemplo, se puede ir accediendo a un gran bloque de memoria en un bucle).

CX = Contador.

Se utiliza comúnmente como contador en bucles y operaciones repetitivas de manejo de cadenas. En las instrucciones de desplazamiento y rotación se utiliza como contador de 8 bits.

DX = Datos.

Usado en conjunción con AX en las operaciones de multiplicación y división que involucran o generan datos de 32 bits. En las de entrada y salida se emplea para especificar la dirección del puerto E/S.

Registros de propósito general

BX = Base. Se usa como registro base para referenciar direcciones de memoria con direccionamiento indirecto, manteniendo la dirección de la base o comienzo de tablas o matrices. De esta manera, no es preciso indicar una posición de memoria fija, sino la número BX (así, haciendo avanzar de unidad en unidad a BX, por ejemplo, se puede ir accediendo a un gran bloque de memoria en un bucle).

CX = Contador. Se utiliza comúnmente como contador en bucles y operaciones repetitivas de manejo de cadenas. En las instrucciones de desplazamiento y rotación se utiliza como contador de 8 bits.

DX = Datos. Usado en conjunción con AX en las operaciones de multiplicación y división que involucran o generan datos de 32 bits. En las de entrada y salida se emplea para especificar la dirección del puerto E/S.

Registros de segmento:

Definen áreas de 64 Kb dentro del espacio de direcciones de 1 Mb del 8086. Estas áreas pueden traslaparse total o parcialmente. No es posible acceder a una posición de memoria no definida por algún segmento: si es preciso, habrá de moverse alguno.

CS = Registro de segmento de código (code segment).

Contiene la dirección del segmento con las instrucciones del programa. Los programas de más de 64 Kb requieren cambiar CS periódicamente.

DS = Registro de segmento de datos (data segment).

Segmento del área de datos del programa.

SS = Registro de segmento de pila (stack segment).

Segmento de pila.

ES = Registro de segmento extra (extra segment).

Segmento de ampliación para zona de datos. Es extraordinariamente útil actuando en conjunción con DS: con ambos se puede definir dos zonas de 64 Kb, tan alejadas como se deseé en el espacio de direcciones, entre las que se pueden intercambiar datos

Registros de segmento:

Definen áreas de 64 Kb dentro del espacio de direcciones de 1 Mb del 8086. Estas áreas pueden traslaparse total o parcialmente. No es posible acceder a una posición de memoria no definida por algún segmento: si es preciso, habrá de moverse alguno. CS = Registro de segmento de código (code segment). Contiene la dirección del segmento con las instrucciones del programa. Los programas de más de 64 Kb requieren cambiar CS periódicamente. DS = Registro de segmento de datos (data segment). Segmento del área de datos del programa. SS = Registro de segmento de pila (stack segment). Segmento de pila. ES = Registro de segmento extra (extra segment). Segmento de ampliación para zona de datos. Es extraordinariamente útil actuando en conjunción con DS: con ambos se puede definir dos zonas de 64 Kb, tan alejadas como se deseé en el espacio de direcciones, entre las que se pueden intercambiar datos

Registros punteros de pila

SP = Puntero de pila (stack pointer).

Apunta al inicio de la pila. Utilizado en las instrucciones de manejo de la pila.

BP = Puntero base (base pointer).

Es un puntero de base, que apunta a una zona dentro de la pila dedicada al almacenamiento de datos (variables locales y parámetros de las funciones en los programas compilados).

Registros punteros de pila

SP = Puntero de pila (stack pointer). Apunta al inicio de la pila. Utilizado en las instrucciones de manejo de la pila. BP = Puntero base (base pointer). Es un puntero de base, que apunta a una zona dentro de la pila dedicada al almacenamiento de datos (variables locales y parámetros de las funciones en los programas compilados).

Puntero de instrucciones o contador de programa

IP = Puntero de instrucción (instruction pointer).

Marca el desplazamiento de la instrucción en curso dentro del segmento de código. Es automáticamente modificado con la lectura de una instrucción.

Puntero de instrucciones o contador de programa

IP = Puntero de instrucción (instruction pointer). Marca el desplazamiento de la instrucción en curso dentro del segmento de código. Es automáticamente modificado con la lectura de una instrucción.

Registros índices:

SI = Índice fuente (source index).

Utilizado como registro de índice en ciertos modos de direccionamiento indirecto, también se emplea para guardar un valor de desplazamiento en operaciones de cadenas.

DI = Índice destino (destination index).

Se usa en determinados modos de direccionamiento indirecto y para almacenar un desplazamiento en operaciones con cadenas.

Registros índices:

SI = Índice fuente (source index). Utilizado como registro de índice en ciertos modos de direccionamiento indirecto, también se emplea para guardar un valor de desplazamiento en operaciones de cadenas. DI = Índice destino (destination index). Se usa en determinados modos de direccionamiento indirecto y para almacenar un desplazamiento en operaciones con cadenas.

Registro de estado o de indicadores (flags)

- Es un registro de 16 bits de los cuales 9 son utilizados para indicar diversas situaciones durante la ejecución de un programa.
- Los bits 0, 2, 4, 6, 7 y 11 son indicadores de condición, que reflejan los resultados de operaciones del programa
- los bits del 8 al 10 son indicadores de control y el resto no se utilizan.
- Estos indicadores pueden ser comprobados por las instrucciones de salto condicional, lo que permite variar el flujo secuencial del programa según el resultado de las operaciones.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flag	--	--	--	-	OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF

Registro de estado o de indicadores (flags)

Es un registro de 16 bits de los cuales 9 son utilizados para indicar diversas situaciones durante la ejecución de un programa.

Los bits 0, 2, 4, 6, 7 y 11 son indicadores de condición, que reflejan los resultados de operaciones del programa

los bits del 8 al 10 son indicadores de control y el resto no se utilizan.

Estos indicadores pueden ser comprobados por las instrucciones de salto condicional, lo que permite variar el flujo secuencial del programa según el resultado de las operaciones.

Bit	Indicador de:	Uso
CF	Acarreo ("Carry Flag")	Indicador de arrastre del bit de mayor orden, que puede ocurrir en las operaciones aritméticas suma y resta.
PF	Paridad ("Parity Flag")	Si está activo Indica un número par de bits activos (bits cuyo contenido es 1). Esta información es útil cuando el procesador controla transmisiones de datos.
AF	Acarreo auxiliar	Indicador de ajuste en operaciones aritméticas con cantidades BCD
ZF	Cero ("Zero Flag")	Está activo si el resultado de operación es cero o resultado de comparación igual.
SF	Signo ("Sign Flag")	Si está activo indica que el resultado de operación o de comparación son negativos.
TF	Detención ("Trap Flag")	Si está activo, el procesador genera automáticamente una interrupción después de la ejecución de cada instrucción, lo que permite controlar paso a paso la ejecución del programa. Este bit debe estar normalmente inactivo (a 0).
IF	Interrupción ("Interrupt Flag")	Este bit controla el estado del sistema de interrupciones enmascarables . Cuando está activo (1) permite las interrupciones; el estado inactivo (0) las deshabilita.
DF	Dirección ("Direction Flag")	Indica la dirección de las operaciones.
OF	Desbordamiento (OverflowFlag")	Señala desbordamiento aritmético

(Gp:) Bit

(Gp:) Señala desbordamiento aritmético

(Gp:) Desbordamiento (Overflow Flag")

(Gp:) OF

(Gp:) Indica la dirección de las operaciones.

(Gp:) Dirección ("Direction Flag")

(Gp:) DF

(Gp:)

Este bit controla el estado del sistema de interrupciones enmascarables . Cuando está activo (1) permite las interrupciones; el estado inactivo (0) las deshabilita.

(Gp:)

Interrupción ("Interrupt Flag")

(Gp:)

IF

(Gp:) Si está activo, el procesador genera automáticamente una interrupción después de la ejecución de cada instrucción, lo que permite controlar paso a paso la ejecución del programa. Este bit debe estar normalmente inactivo (a 0).

(Gp:)

Detención ("Trap Flag")

(Gp:)

TF

(Gp:) Si está activo indica que el resultado de operación o de comparación son negativos.

(Gp:) Signo ("Sign Flag")

(Gp:) SF

(Gp:) Está activo si el resultado de operación es cero o resultado de comparación igual.

(Gp:) Cero ("Zero Flag")

(Gp:) ZF

(Gp:) Indicador de ajuste en operaciones aritméticas con cantidades BCD

(Gp:) Acarreo auxiliar

(Gp:) AF

(Gp:) Si está activo Indica un número par de bits activos (bits cuyo contenido es 1). Esta información es útil cuando el procesador controla transmisiones de datos.

(Gp:)

Paridad ("Parity Flag")

(Gp:)

PF

(Gp:) Indicador de arrastre del bit de mayor orden, que puede ocurrir en las operaciones aritméticas suma y resta.

(Gp:)

Acarreo ("Carry Flag")

(Gp:)

CF

(Gp:) Uso

(Gp:) Indicador de:

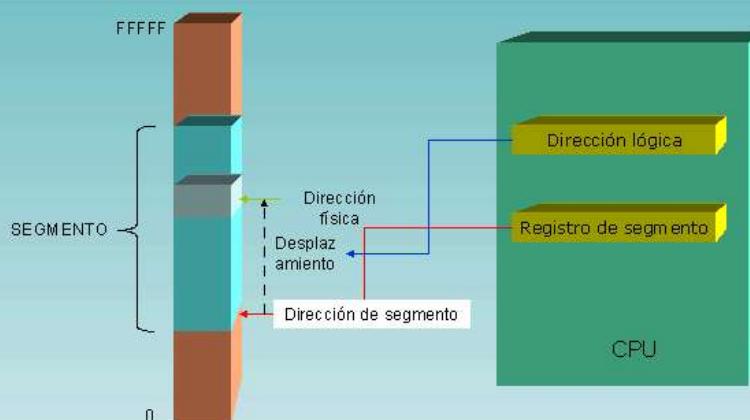
Bit	Indicador de:	Indicativo si bit 1	Indicativo si bit 0
CF	Acarreo	CY ("Carry yes")	NC ("No Carry")
PF	Paridad	PE ("Parity Even") paridad par	PO ("Parity Odd") paridad impar
AF	Acarreo auxiliar	AC ("Auxiliar Carry")	NA ("No Auxiliar")
ZF	Cero	ZR ("Zero")	NZ ("No Zero")
SF	Signo	NG ("Negative") negativo	PL ("Plus") positivo
IF	Interrupción	EI ("Enabled Interrupt") activa	DI ("Disabled Interrupt") desactivada
DF	Dirección	DH ("Down") decremento	UP incremento
OF	Desbordamiento	OV ("Overflow")	NV ("No overflow")

NV ("No overflow")
 OV ("Overflow")
 Desbordamiento
 OF
 UP incremento
 DN ("Down") decremento
 Dirección
 DF
 DI ("Disabled Interrupt") desactivada
 EI ("Enabled Interrupt") activa
 Interrupción
 IF
 PL ("Plus") positivo
 NG ("Negative") negativo
 Signo
 SF
 NZ ("No Zero")
 ZR ("Zero")
 Cero
 ZF
 NA ("No Auxiliar")
 AC ("Auxiliar Carry")
 Acarreo auxiliar
 AF
 PO ("Parity Odd") paridad impar
 PE ("Parity Even") paridad par
 Paridad
 PF
 NC ("No Carry")
 CY ("Carry yes")
 Acarreo
 CF
 Indicativo si bit 0
 Indicativo si bit 1
 Indicador de:
 Bit

Segmentación de la Memoria en el 8086

Segmentación de la Memoria en el 8086

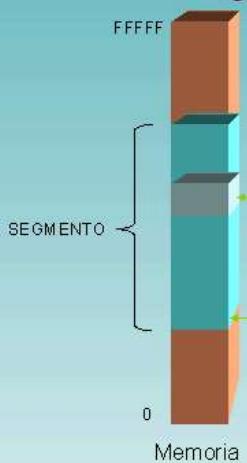
SEGMENTACIÓN



SEGMENTACIÓN

Dirección lógica
Registro de segmento
Desplazamiento
Dirección de segmento
Dirección física
CPU
SEGMENTO
0
FFFFF

Segmento de Memoria



- Un segmento es un área continua de memoria que puede tener hasta 64K-bytes, que debe comenzar en una localidad de memoria cuya dirección sea límite de 16 bytes (cantidad denominada párrafo) y que puede traslaparse con otros segmentos.

Segmento de Memoria

Un segmento es un área continua de memoria que puede tener hasta 64K-bytes, que debe comenzar en una localidad de memoria cuya dirección sea límite de 16 bytes (cantidad denominada párrafo) y que puede traslaparse con otros segmentos.

SEGMENTO

Memoria

0

FFFFF

Segmentos de Memoria

- **El segmento de código** (tiene como base el contenido del registro CS).
- **En este segmento se encuentran las instrucciones que forman el programa. Para acceder a los datos contenidos en él, se usa el registro IP como desplazamiento.**
- **El segmento de datos** (que tiene como base el registro DS).
- **Contiene los datos que utiliza el programa (variables, etc.) Para acceder a los datos contenidos en él, se suele utilizar los registros SI y DI como desplazamiento.**

El segmento de código (tiene como base el contenido del registro CS).

En este segmento se encuentran las instrucciones que forman el programa. Para acceder a los datos contenidos en él, se usa el registro IP como desplazamiento.

El segmento de datos (que tiene como base el registro DS).

Contiene los datos que utiliza el programa (variables, etc.) Para acceder a los datos contenidos en él, se suele utilizar los registros SI y DI como desplazamiento.

Segmentos de Memoria

Segmentos de Memoria

- El segmento de pila (con SS como base).

- En él se desarrolla la pila del programa, utilizada para almacén temporal de datos, llamadas a funciones, etc. Debe estar presente en todos los programas EXE de forma obligada. Se utiliza el registro SP para acceder a los datos de este segmento.

- El segmento extra (con ES como base).

- Su uso es opcional, y en él se encuentra un segmento definido por el usuario y que, regularmente, contiene datos adicionales. Al igual que ocurre con el segmento de datos, para acceder a los datos contenidos en él, se suelen utilizar los registros SI y DI.

Segmentos de Memoria

El segmento de pila (con SS como base).

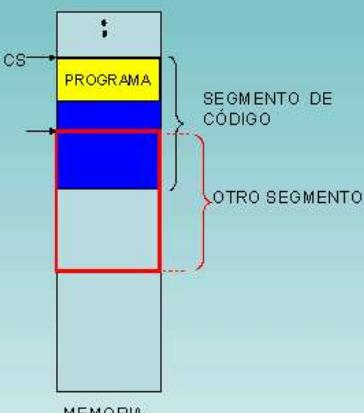
En él se desarrolla la pila del programa, utilizada para almacén temporal de datos, llamadas a funciones, etc. Debe estar presente en todos los programas EXE de forma obligada. Se utiliza el registro SP para acceder a los datos de este segmento.

El segmento extra (con ES como base).

Su uso es opcional, y en él se encuentra un segmento definido por el usuario y que, regularmente, contiene datos adicionales. Al igual que ocurre con el segmento de datos, para acceder a los datos contenidos en él, se suelen utilizar los registros SI y DI.

SEGMENTACIÓN

TRASLAPE DE SEGMENTOS



SEGMENTACIÓN

(Gp:) CS

(Gp:) DS

(Gp:) SS

(Gp:) ES

(Gp:) STACK

(Gp:) SEGMENTO EXTRA DE DATOS

(Gp:) SEGMENTO DE DATOS

(Gp:) SEGMENTO DE CÓDIGO

CS

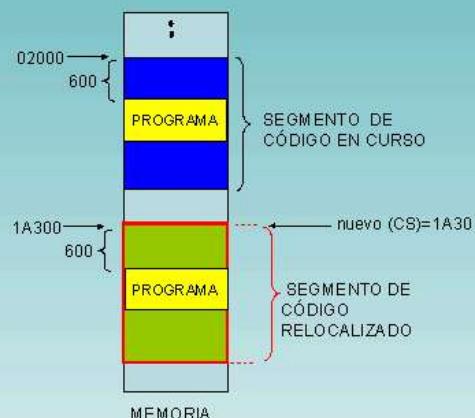
OTRO SEGMENTO

SEGMENTO DE CÓDIGO

PROGRAMA

SEGMENTACIÓN

Relocalización de un programa usando el registro CS



SEGMENTACIÓN

02000

SEGMENTO DE CÓDIGO RELOCALIZADO

SEGMENTO DE CÓDIGO EN CURSO

PROGRAMA

MEMORIA

Relocalización de un programa usando el registro CS

PROGRAMA

600

1A300

600

nuevo (CS)=1A30

Dirección Física y Dirección Lógica

Cada localidad de memoria tiene dos tipos de direcciones :

Dirección Física y Dirección Lógica o efectiva

Una dirección Física es el valor único de 20 bits que identifica la localización de cada byte en el espacio de memoria de 1 Mbyte.

Los programas tratan con dirección Lógicas mas que con la dirección Física.

Para cualquier localidad de memoria, el valor base del segmento ubica el primer byte del contenido del segmento y el valor del offset es la distancia, en bytes , de la localización destino del principio del segmento.

Los valores de la base del Segmento y del offset son cantidades no signadas de 16 bits; la dirección del byte más bajo del segmento tiene un offset 0.

Cada localidad de memoria tiene dos tipos de direcciones :

Dirección Física y Dirección Lógica o efectiva

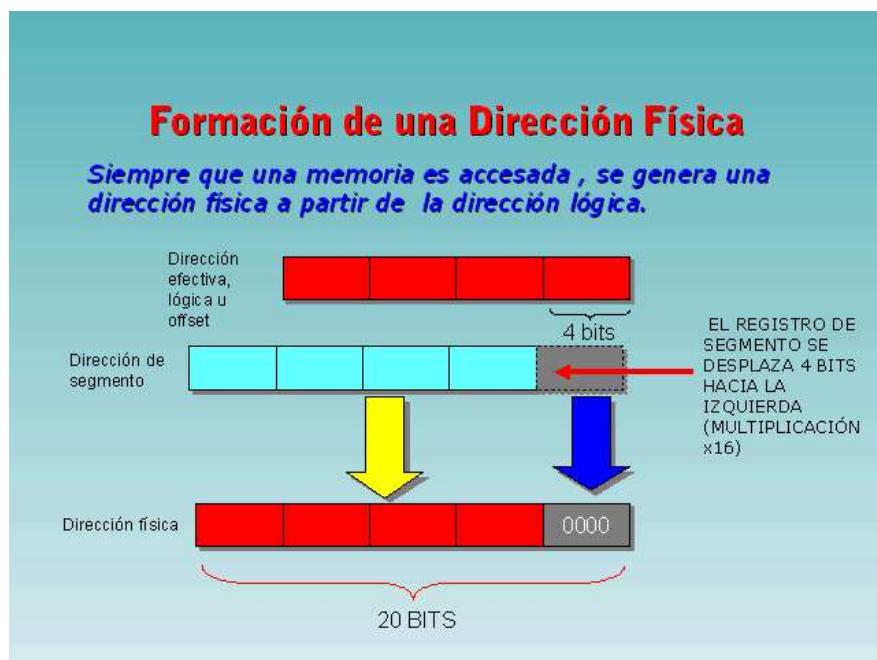
Una dirección Física es el valor único de 20 bits que identifica la localización de cada byte en el espacio de memoria de 1 Mbyte.

Los programas tratan con dirección Lógicas mas que con la dirección Física.

Para cualquier localidad de memoria, el valor base del segmento ubica el primer byte del contenido del segmento y el valor del offset es la distancia, en bytes , de la localización destino del principio del segmento.

Los valores de la base del Segmento y del offset son cantidades no signadas de 16 bits; la dirección del byte más bajo del segmento tiene un offset 0.

Dirección Física y Dirección Lógica



Formación de una Dirección Física

0000

4 bits

Dirección de segmento

Dirección efectiva, lógica u offset

Dirección física

EL REGISTRO DE SEGMENTO SE DESPLAZA 4 BITS HACIA LA IZQUIERDA (MULTIPLICACIÓN x16)

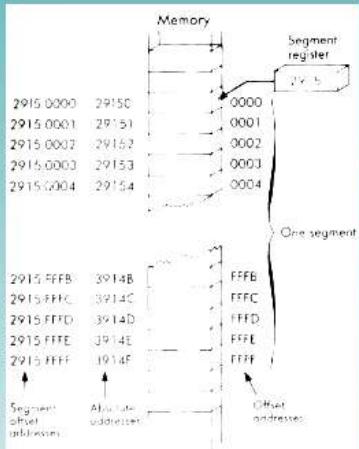
20 BITS

Siempre que una memoria es accesada , se genera una dirección física a partir de la dirección lógica.



Formación de una Dirección Física

Formación de una Dirección Física



Por Ejemplo:

Si la dirección base del segmento es 2915h, entonces las direcciones en este segmento inician en 2915:0000 y van hasta 2915:FFFF

Formación de una Dirección Física

Por Ejemplo:

Si la dirección base del segmento es 2915h, entonces las direcciones en este segmento inician en 2915:0000 y van hasta 2915:FFFF

El Centro de Tesis, Documentos, Publicaciones y Recursos Educativos más amplio de la Red.

[Política de Derechos de Autor](#) | [Normas de la Comunidad](#) | [Nuestro Código de Honor](#) | [Política de Privacidad](#) | [Términos de Servicio](#) |

[Gestionar las preferencias](#)

© Monografias.com S.A.