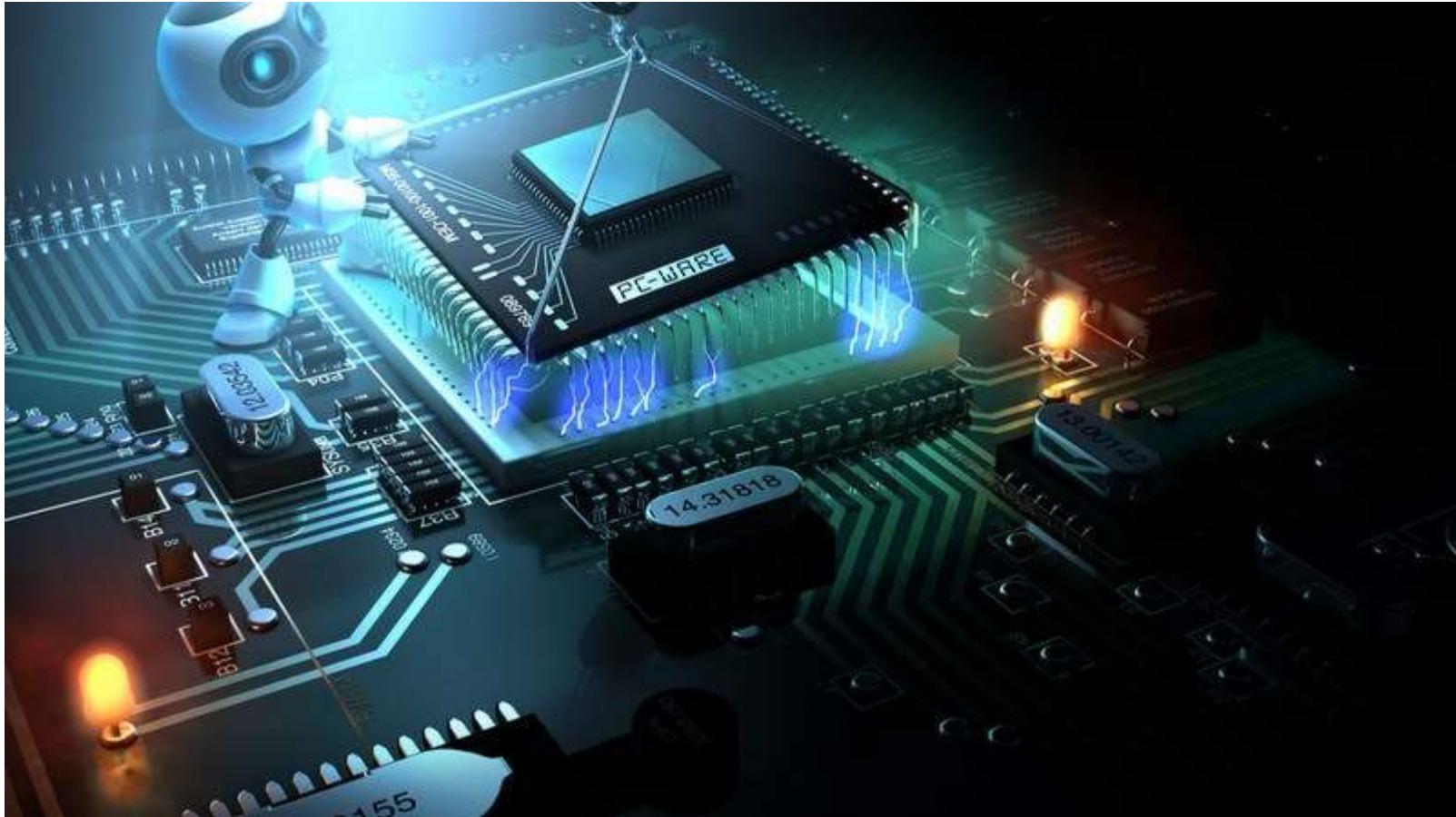


# LENGUAJE ENSAMBLADOR

## (7° A-ICI)



Mtro. en Ing. Armando Álvarez Fdez.  
Ago-2024

# LENGUAJE ENSAMBLADOR

## INTRODUCCIÓN

ALGUNAS MALAS REPUTACIONES O IMPRESIONES ERRONEAS QUE LA GENTE TIENE DEL LENGUAJE ENSAMBLADOR:

- Es difícil de aprender.
- Es duro de leer o entender.
- Es difícil de depurar.
- Es difícil de mantener.
- Es difícil de escribir.
- Ocupa mucho tiempo el programar.
- Los compiladores mejorados eliminan la necesidad de lenguaje ensamblador.
- Las computadoras actuales son tan rápidas que eliminan la necesidad de usar ensamblador.
- Si quieres más velocidad, mejora el algoritmo, en lugar de usar el ensamblador.

## **IMPRESIONES ERRONEAS O MALA REPUTACIÓN DEL LENGUAJE ENSAMBLADOR:**

- Las PC<sub>s</sub> de hoy tienen tanta memoria que no es importante utilizar ensamblador.
- El lenguaje ensamblador no tiene portabilidad.
- Es un lenguaje imperfecto y con muchas carencias.

# **LO QUE SI ES CORRECTO SABER DEL LENGUAJE ENSAMBLADOR**

## **USAR ENSAMBLADOR ES...**

- VELOCIDAD** (Los programas se agilizan hasta 5 o 10 veces más rápidos).
- ESPACIO REDUCIDO** (los programas se hacen mas cortos, Ejem. hasta 50% menos gasto de memoria).
- CAPACIDAD** (se pueden hacer cosas que son difíciles o imposibles en Leng. de alto nivel).

# USAR ENSAMBLADOR Es...

## ● CONOCIMIENTO (Expertice)

(Tu conocim-de Leng- ensamblador  
te ayudará a escribir, mejores  
Programas aún en Leng-de alto  
nivel.

→ Vale la pena aprender  
Leng- Ensamblador. Una  
vez que lo hayas apren-  
dido bien, probable-  
mente te asombraras de  
usarlo más de lo que  
habías pensado.

# ¿Qué es un CPU (Central Processing Unit) Y cuales sus componentes básicos?

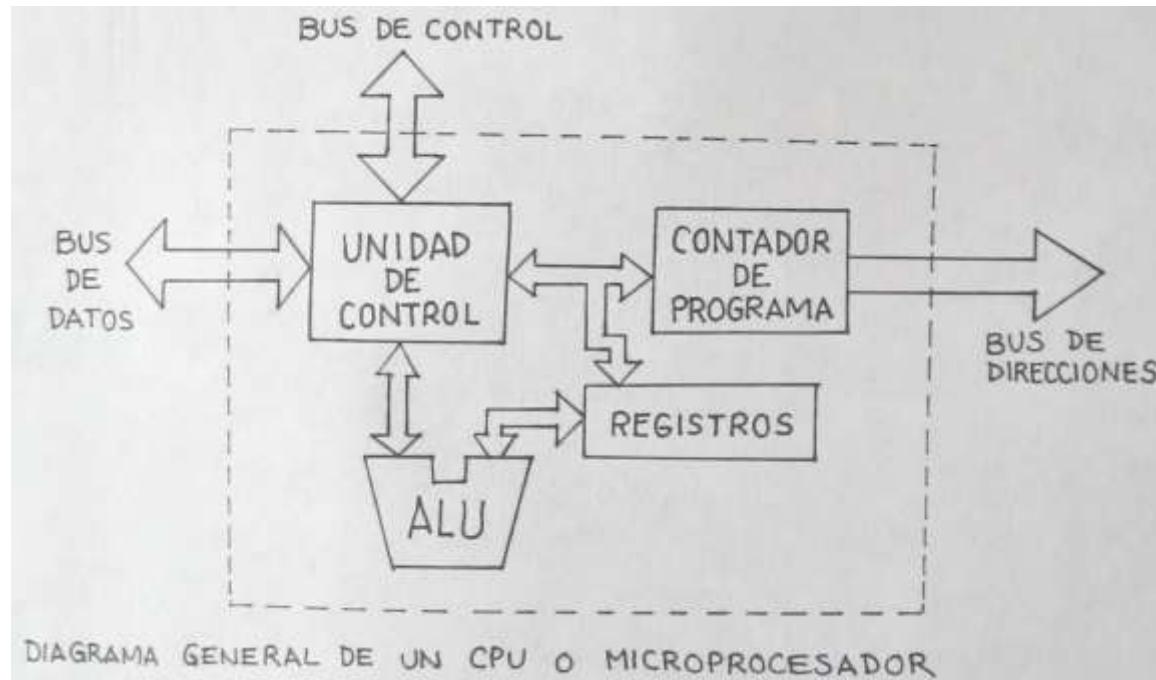
(Unidad Central de Procesamiento ó Microprocesador)

## MICROPROCESADOR (CPU)

Dispositivo electrónico, lógico secuencial capaz de procesar y ejecutar instrucciones (lógicas y aritméticas) codificadas digitalmente.

### PARTES QUE LO COMPONEN:

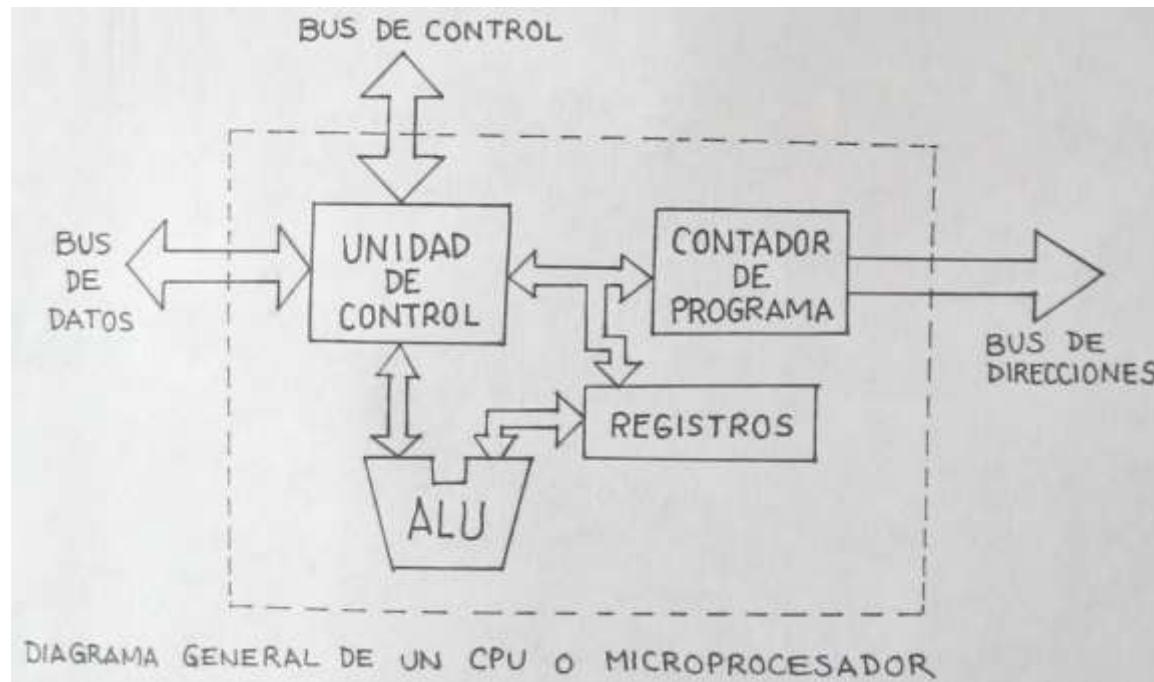
- 1) Unidad lógica aritmética.
- 2) Contador de programa.
- 3) Registros.
- 4) Bus de direcciones.
- 5) Bus de datos.
- 6) Bus de control.
- 7) Unidad de control.



**UNIDAD DE CONTROL.**- Es la encargada de traer y decodificar las instrucciones así como controlar los demás componentes internos.

**ALU (Unidad lógica aritmética).**- Es la encargada de realizar las operaciones lógicas y aritméticas de las instrucciones.

**REGISTROS.**- Memorias temporales de almacenaje de datos y resultados parciales de cálculos.



**CONTADOR DE PROGRAMA.**- Registro en el cual se guarda la dirección de memoria de la instrucción que se está ejecutando, el cual se va incrementando o ajustando cada vez a la dirección nueva correspondiente.

**BUS DE DATOS.**- Terminales del CPU en donde se indica que dirección de memoria o puerto periférico se está accediendo.

**BUS DE CONTROL.**- Conjunto de terminales mediante las cuales el CPU controla sus funciones técnicas de operación.

# UNIDAD TEMÁTICA I

## EVOLUCIÓN DE LOS

## MICROPROCESADORES DE **INTEL**

# UNIDAD TEMÁTICA I

## EVOLUCIÓN DE LOS MICROPROCESADORES DE INTEL

1971

El CPU 4004

- Fué el 1º CPU de Intel
- Un CPU de 4 bits (Bus de datos)
- Con 4096 Espacios de memoria  
(localidad)
- Con 45 Instrucciones



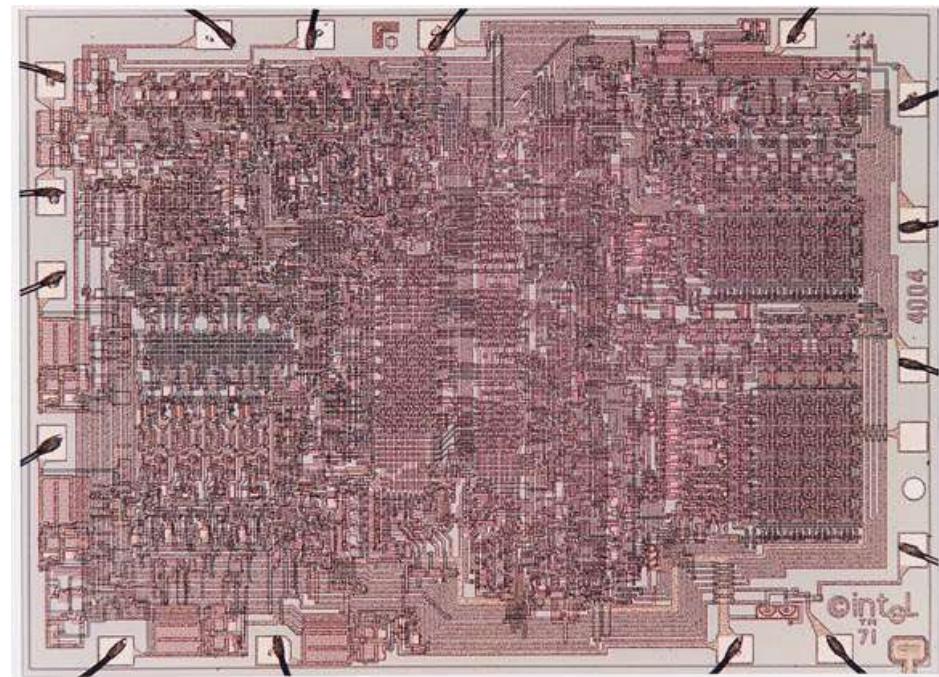
INTEL

- fundada en 1968

- Gordon E. Moore, Robert Noyce y Andy Grove

- Se llamo → NM Electronics y luego INTEGRAL Electronics  
y Japonesa, Busicom

- 50 mil Instrucc/seg.



# Fundación de INTEL



Robert Noyce

Gordon Moore

Andy Grove



Intel's early headquarters in Mountain View



The next big step was the move to Santa Clara, now called SC1

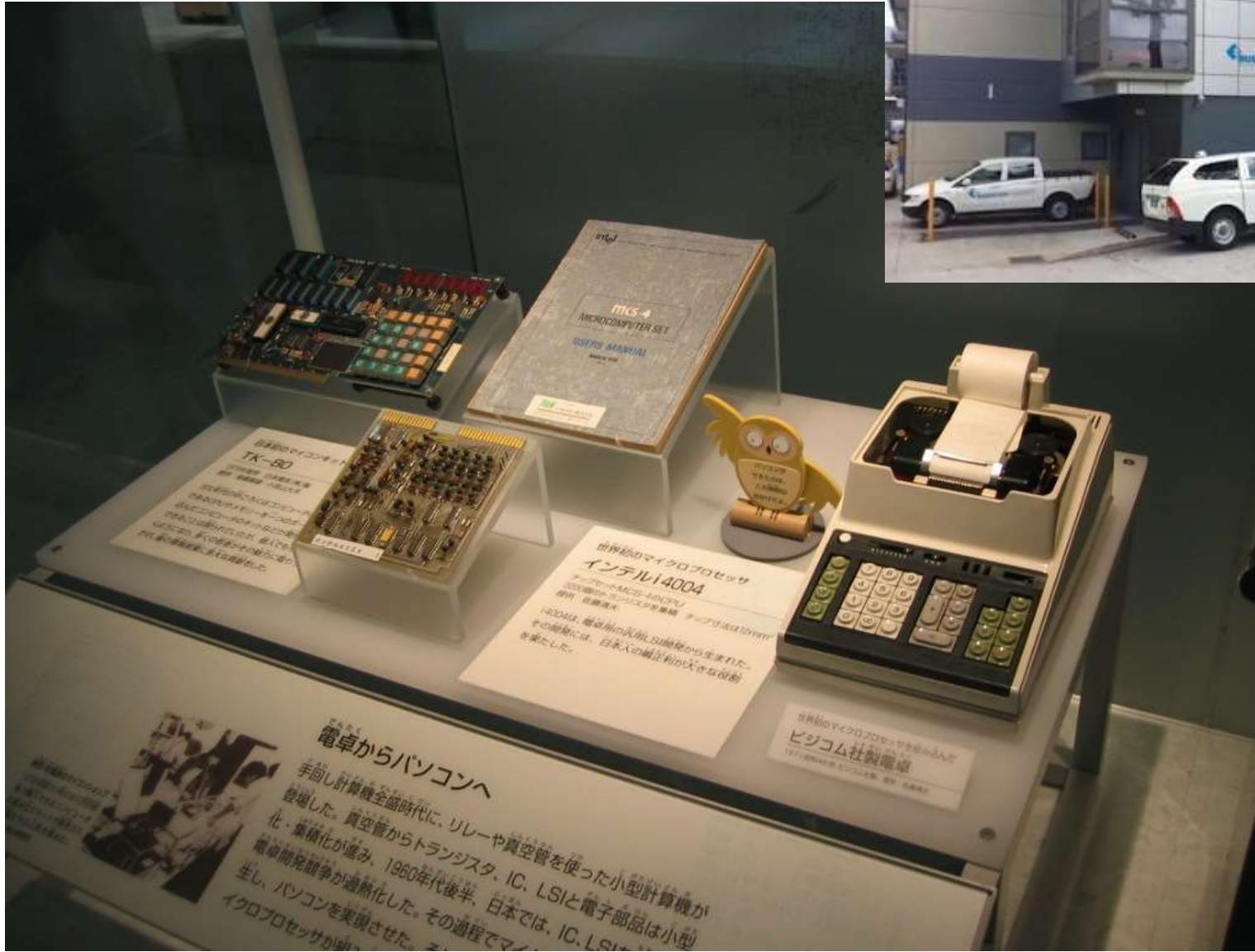


Business may look easy from a historical perspective, but it never is. There are always pensive moments where hard decisions must be made . . . decisions that could make or break the company. This shot was taken at SC1.

# Inicios de Intel



# EMPRESA JAPONESA BUSICOM QUE USO EL PRIMER CPU 4004



# EVOLUCIÓN DE LOS MICROPROCESADORES DE INTEL

1971

## El CPU 4040

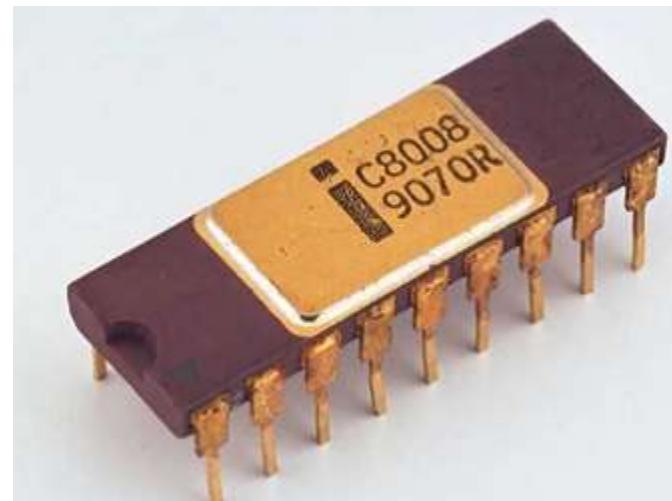
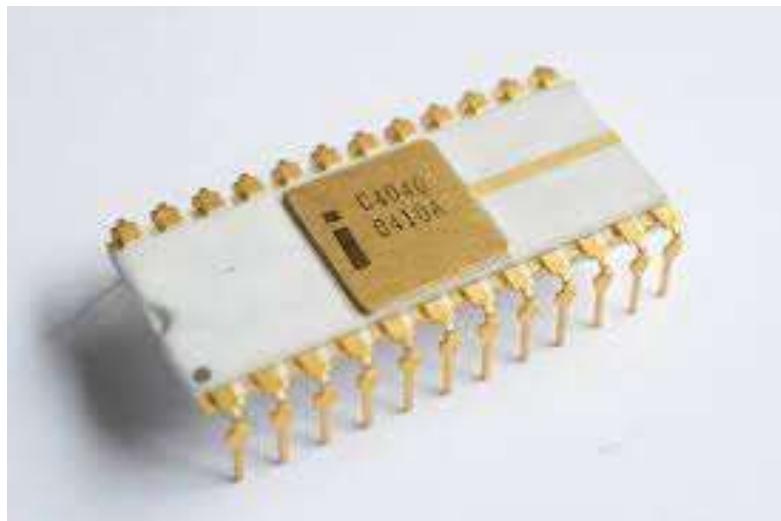
- Una mejora en velocidad del 4004
- Mantenía los 4 bits de datos y misma Capacidad de direccionamiento.

✓ 1971

## EL CPU: 8008

- Una vers. extendida del 4004 a 8 bits.
- Capacidad de dirección de 16K de memoria.
- Con 48 instrucciones.

- Era algo lento pues se basaba mucho en la arquitectura del 4004.



# EVOLUCIÓN DE LOS MICROPROCESADORES DE INTEL

1973 EL CPU = 8080

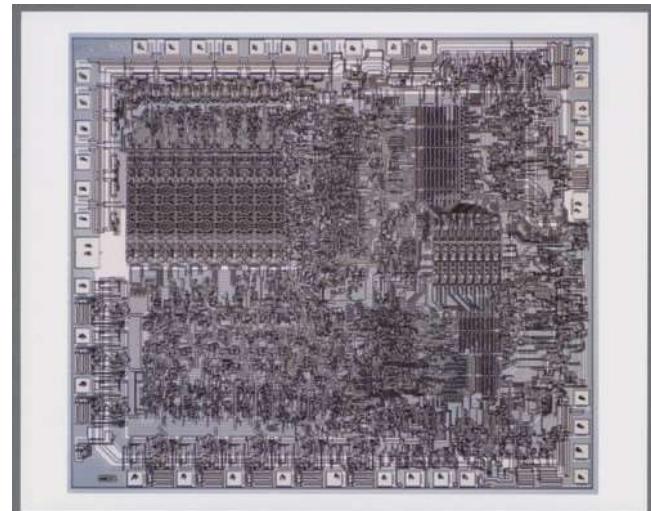
- ✓ - El primer CPU moderno diseñado para 8 bits.
- ✓ - 10 veces más rápido que el 8008
  - 500,000 Instrucc./seg.
- ✓ - Compatibile con la lógica TTL
- ✓ - Capaz de direcc de 64KB de memoria.



© www.cpu-world.com

OTRAS CÍAS ENTRAN AL MERC. DE CPUs de 8 BITS  
Como:

- Motorola = MC6800
- Mos Technology = 6502
- Zilog = Z-8 y después Z-80
- National Semicond = IMP-8



1977

## EL CPU 8085

- Una vers. actualizada del 8080
- ✓ - Más veloz que el 8080
- ✓ - 769,230 Instruc/seg.
- ✓ - Con reloj interno y mayor frecuencia
- ✓ - Con Controlador interno de Sistema.
  - 8 bits de datos

- Por estos fechas apareció Zilog  
y su Z-80, con Leng.-de Máquina  
compatible con el 8085.



# EVOLUCIÓN DE LOS MICROPROCESADORES DE INTEL

- 1978  
+ 1979

1978  
≈ 1979  
EL CPU 8086 / 8088

✓ - Procesador de 16 BITS

✓ - 2.5 MIPS (mill. de Instr por segundo)  $t=400\text{ns}$

✓ - Capacidad de direccionamiento de  $1MB = 1024KB$

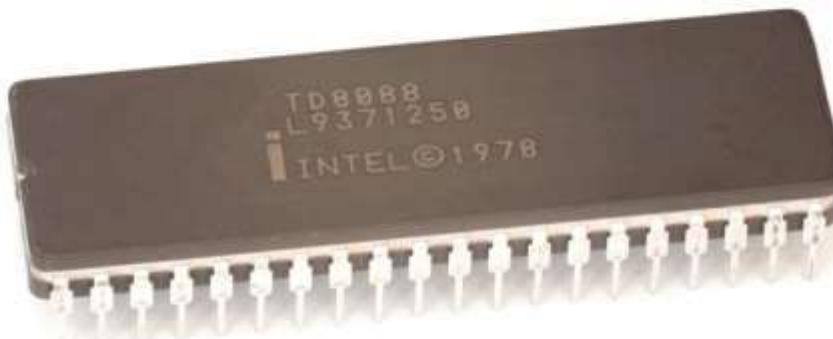
✓ - Un pequeño cache de 4 a 6 Bytes, clave para su alta velocidad.

✓ - Tenía una mayor cantidad de registros internos.  
✓ - para mejorar el Software.

✓ - En 1981 IBM decide utilizar un 8088 en sus PCs XT.

-- El 8086 y el 8088 ejecutan el mismo conjunto de instrucciones. Internamente son idénticos.

-- Excepto que exteriormente se diferencian en que el 8086 tiene un bus de datos de 16 bits y el del 8088 es de solo 8 bits, por ello, el 8086 era más rápido.



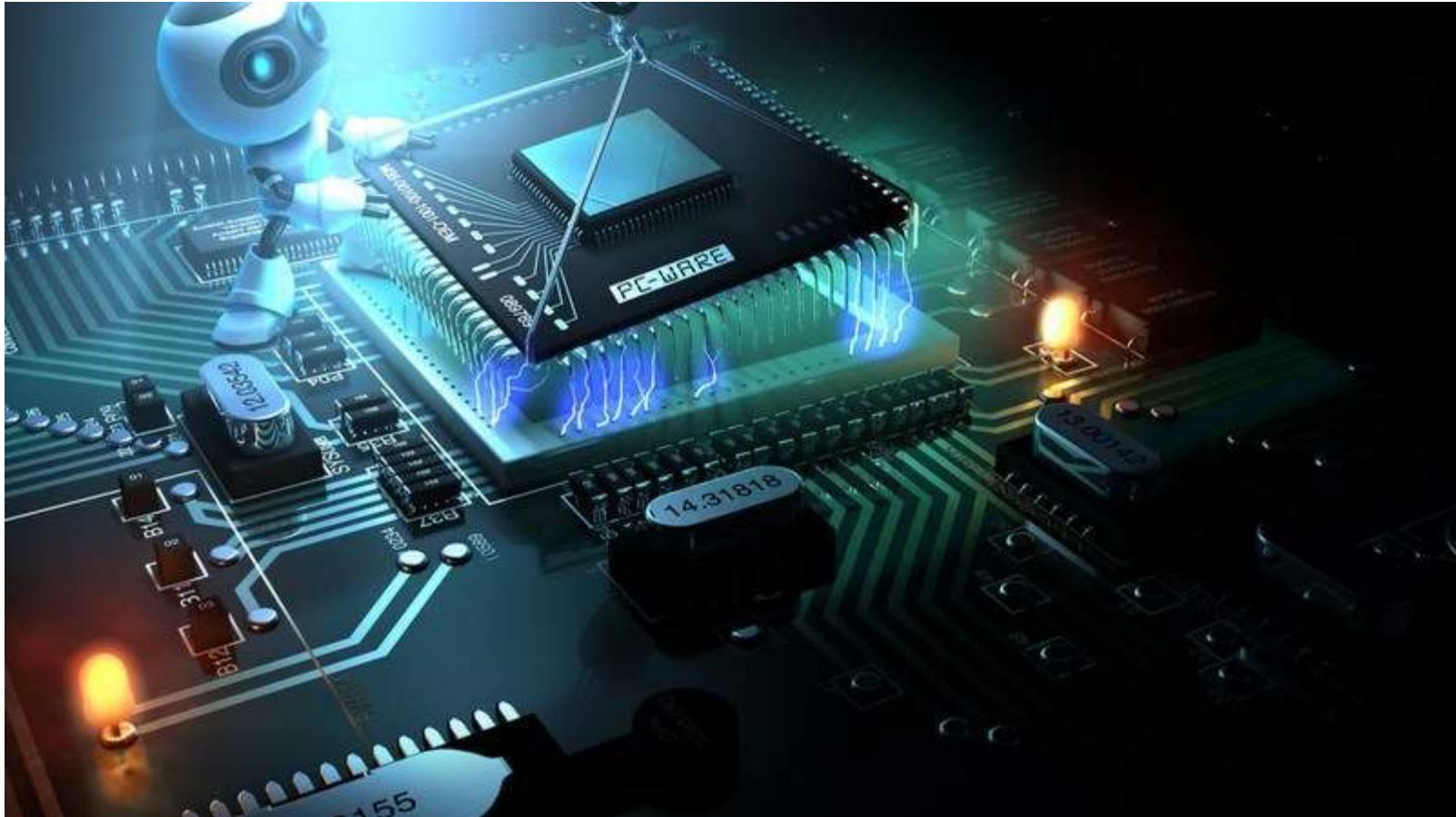


# **APENDICES**



# LENGUAJE ENSAMBLADOR

## (7° A-ICI)



Mtro. en Ing. Armando Álvarez Fdez.  
Ago-2024

# EVOLUCIÓN DE LOS MICROPROCESADORES DE INTEL

- 1978  
+ 1979

1978  
≈ 1979  
EL CPU 8086 / 8088

✓ - Procesador de 16 BITS

✓ - 2.5 MIPS (mill. de Instr por segundo)  $t=400\text{ns}$

✓ - Capacidad de direccionamiento de  $1MB = 1024KB$

✓ - Un pequeño cache de 4 a 6 Bytes, clave para su alta velocidad.

✓ - Tenía una mayor cantidad de registros internos.  
✓ - para mejorar el Software.

✓ - En 1981 IBM decide utilizar un 8088 en sus PCs XT.

-- El 8086 y el 8088 ejecutan el mismo conjunto de instrucciones. Internamente son idénticos.

-- Excepto que exteriormente se diferencian en que el 8086 tiene un bus de datos de 16 bits y el del 8088 es de solo 8 bits, por ello, el 8086 era más rápido.



# EVOLUCIÓN DE LOS MICROPROCESADORES DE INTEL

1983

## EL CPU 80286:

- 16 BITS DE DATOS
- CASI IDENTICO AL 8086 Y 8088
- ✓ - CAPACIDAD DE DIRECCIONAM. DE 16MB.
- ✓ - MISMAS INSTRUCC. QUE 8086 Y 8088 Y ALGUNAS POCAS ADICIONALES PARA MANEJAR LOS 15MB ADICIONALES

- ✓ - 4 MIPS (250ns)
- ✓ - FREC = 8 MHz.



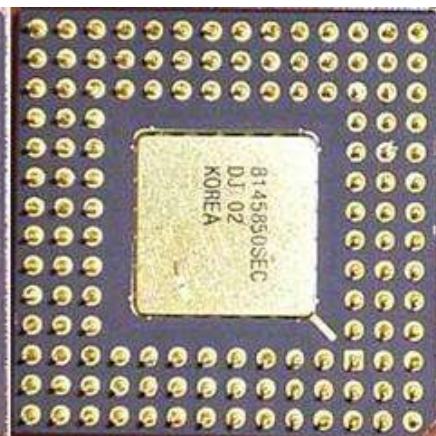
1986 EL CPU 80386:

- ✓ - 32 BITS DE DATOS
- ✓ - 32 BITS DE BUS DE DIRECCIONES. (<sup>CAP. DE DIRECC.</sup> 4GB DE MEMORIA)
- ✓ - CON UNIDAD DE MANEJO DE MEMORIA Y ASIGN. DE MEMORIA (EN HARDWARE, NO SOFTWARE)
- COMPATIBLE CON 8086, 8088 y 80286
- ✓ - REGISTROS DE 32 BITS

SURGEN ALGUNAS VARIANTES:

80386SX → Con 24 BITS DE DIREC. → 16MB. Y DATOS DE 16BITS

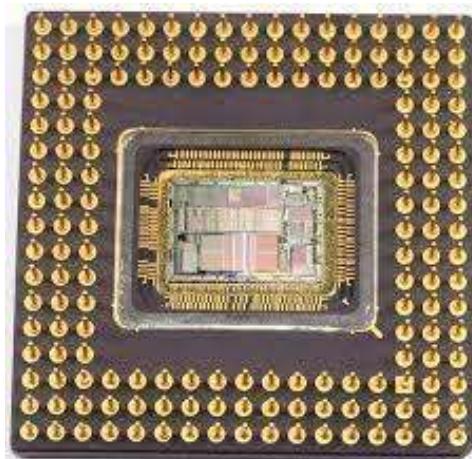
80386 SL → 25 BITS DE DIRECC. (32MB)  
Y B. DATOS = 16 BITS.



1989

### EL CPU 80486:

- CON 32 BITS DE DATOS
- Muy parecido al 80386 excepto por...
- Incorporaba un 80386 (CPU) Y UN 80387 (COPROCESADOR NUMERICO) Y 8KB DE CACHE.
- LA MAYORIA DE LAS INSTRUCC. SE EJECUTABAN EN 1 CICLO DE RELOJ EN VEZ DE 2 (COMO ERA PARA EL 80386)
- 50 MIPS (25ns) Y  $f=50\text{MHz}$ ,  $66\text{MHz}$  Y  $100\text{MHz}$  CON VEL. INTERNAS DEL CPU
- VEL ENTRE CPU Y RAM DE 33MHz.



## 1993 EL CPU PENTIUM: (Originalmente designado como P5 ó 80586 )

- $f = 60$  y  $66 \text{ MHz}$  y después  $100 \text{ MHz}, 120 \text{ MHz}, 133 \text{ MHz} \dots 233 \text{ MHz}$ .
- $110 \text{ MIPS}$ . a  $150 \text{ MIPS}$
- CACHE DE  $16 \text{ KB}$  (  $8 \text{ KB}$  DE INSTRUCC. y  $8 \text{ KB}$  para datos)
- BUS DE DATOS DE  $64 \text{ BITS}$
- CAP. DE DIRECCIONAR ( $4 \text{ GB}$ ) DE MEMORIA.
- CAP. DE MANEJAR NUMEROS DE PUNTO FLOTANTE DE DOBLE PRECISION
- CAP. DE MANEJAR VIDEO DE  $30 \text{ Hz}$
- UN SET DE INSTRUCC. ADICIONAL MMX (MultiMedia Xtension)
- 2 UNIDADES PROCESADORAS DE ENTEROS (Tecnología Superescalar)  
(Ejecuta 2 Instrucc. Independ. simultáneamente)
- Tecnología predictora de saltos
- Con un Coprocesador de Punto flotante integrado.
- 260 Mill. de comp. basado en Pentium y GPPen PowerPC de (Apple+IBM+Motorola) [96% de escrivinhas]



1995

## Pentium-Pro (P6)

- Pensado para el mercado de servidores.
- 21 millones de transistores
- 3 Unidades de Op. con entradas integradas
- 1 Unidad. de Punto Flotante integrada.
- $f = 150\text{MHz}$  y  $166\text{MHz}$
- Contiene 3 máquinas de ejecución de datos.
- Podía ejecutar 3 instrucc. por ciclo de reloj.
- Capas de direccionar 4GB & **64GB** de memoria. (BUS DE DIRECC. DE 36 bits)

(4)

Level 1

- Agrega otra cache de 16KB adicional a los 8K de datos + 8K de Instrucc. de Cache.
- Agrega una cache de Level 2 de 256 KB
- Hardware optimizado para código de 32 bits. (Se vendía con Win NT, en vez de Win 95, por esta razón)



1997

## Pentium II:

- Fue un CPU montado en una pequeña tarjeta electrónica enserrada en una caja plástica que se llamó "SLOT 1". Intel hizo esto porque la caché L2 nueva que llevaba no era lo suficiente rápida para justificar ponerla dentro del CPU.
  - Caché L2 de 512 KB y de 133 MHz
  - El Pentium II fue en esencia un Pentium-Pro, MMX sin caché L2 dentro del microprocesador.
- En 1998 → - Inicialmente el bus de comunicación entre CPU y RAM era de 66 MHz (RAM de 10 ns → PII de 266 MHz a 333 MHz)
- En 1998 → - y después se hizo de 100 MHz (PII de 350 MHz a 450 MHz) (RAM de 8 ns)

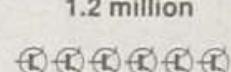
## CPU Pentium II XEON:

- Diseñado para servidores y estaciones de trabajo.
- Con caches de: L1 = 32 KB y L2 = 512 KB, 1 MB ó 2 MB
- Diseñado para trabajar hasta con 4 CPUs en el mismo MoBo.



[www.cpu-world.com](http://www.cpu-world.com)

## INTEL CPUS OVER TIME

	Price at launch/ current price	MIPS at launch/ highest MIPS	Number of transistors
<b>8086</b> June 1978	\$360	.33 .75	29,000 
<b>286</b> February 1982	\$360	1.2 2.66	134,000 
<b>386</b> October 1985	\$299	5 11.4	275,000 
<b>486</b> August 1989	\$950	20 54	1.2 million 
<b>PENTIUM</b> March 1993	\$900 **	112	3.1 million 

\* Discontinued      \*\* Estimated       = 100,000

EJERCICIOS Y NOTAS:

# **ARQUITECTURA INTERNA DE CPUs DE INTEL Y SU EVOLUCIÓN.**

# DIAGRAMAS A BLOQUES DE CPUs 80286 y 80386

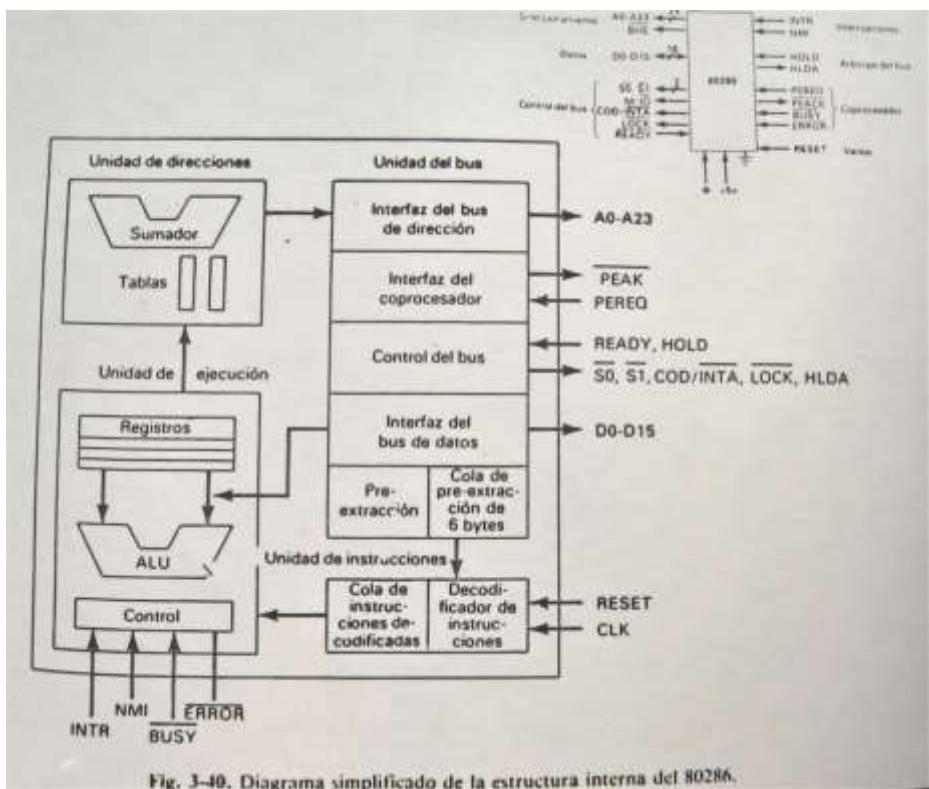


Fig. 3-49. Diagrama simplificado de la estructura interna del 80286.

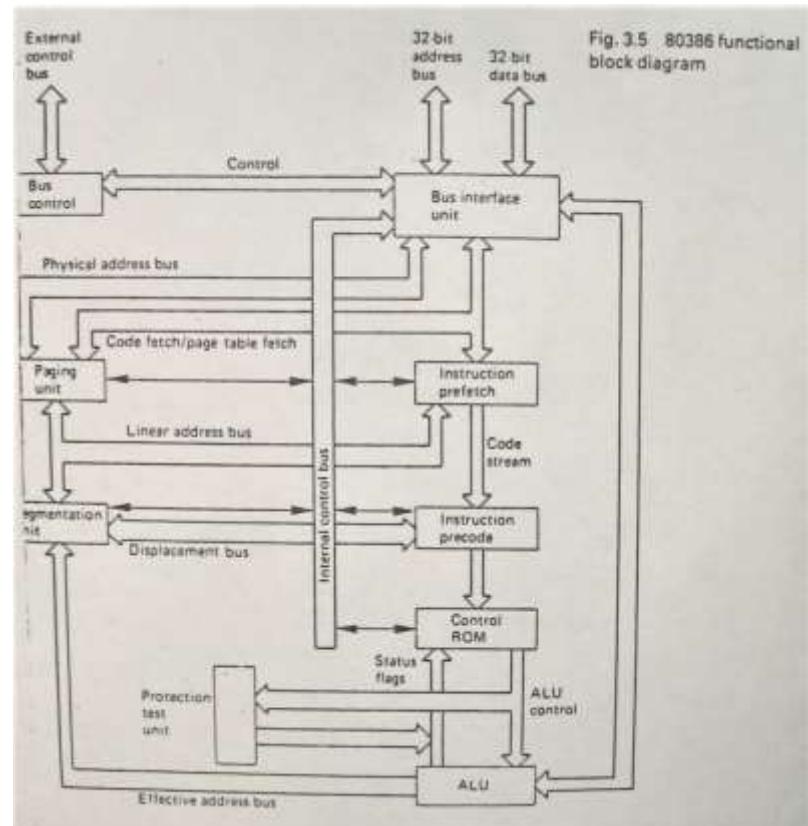
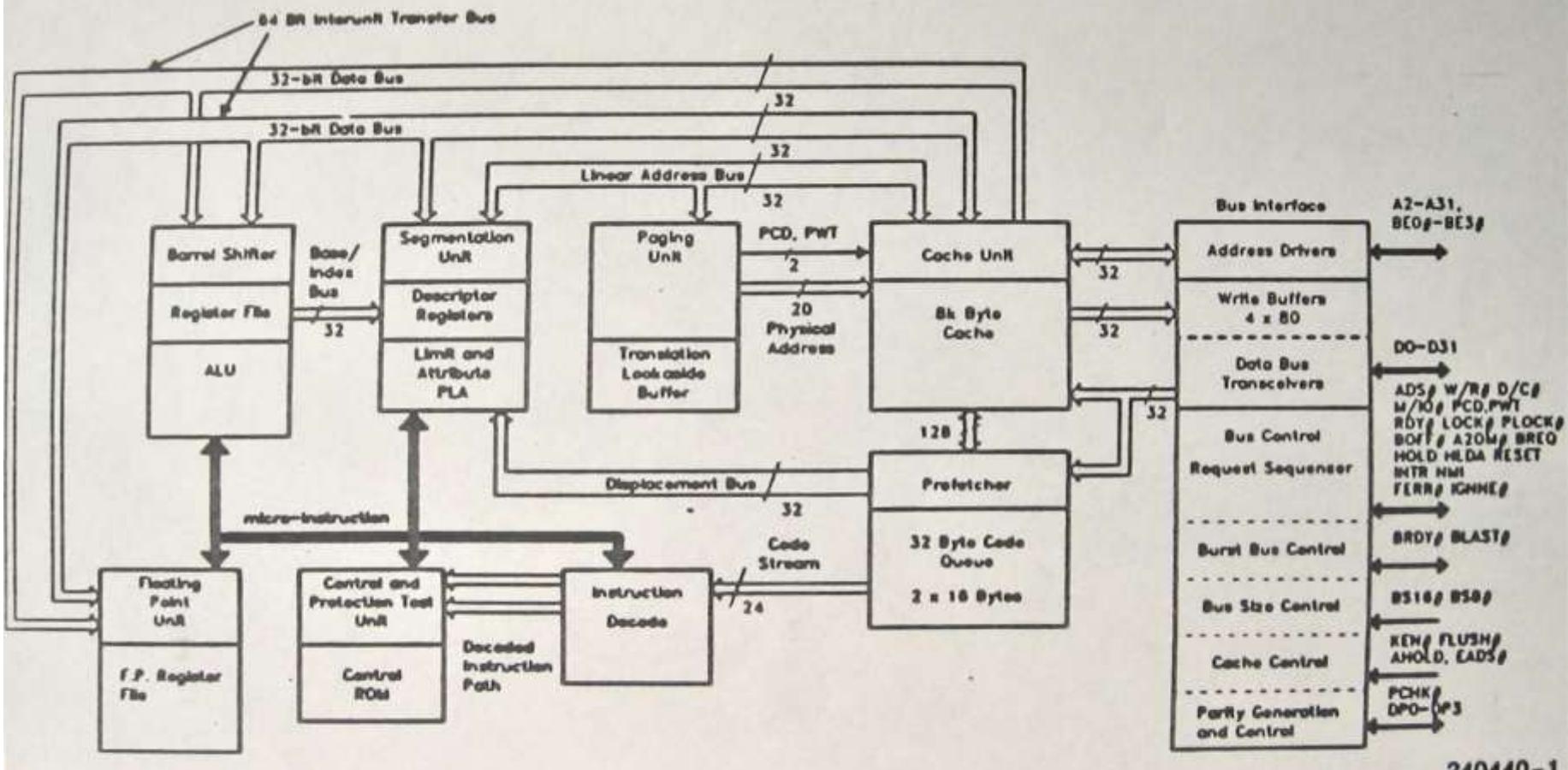


Fig. 3-5. 80386 functional block diagram

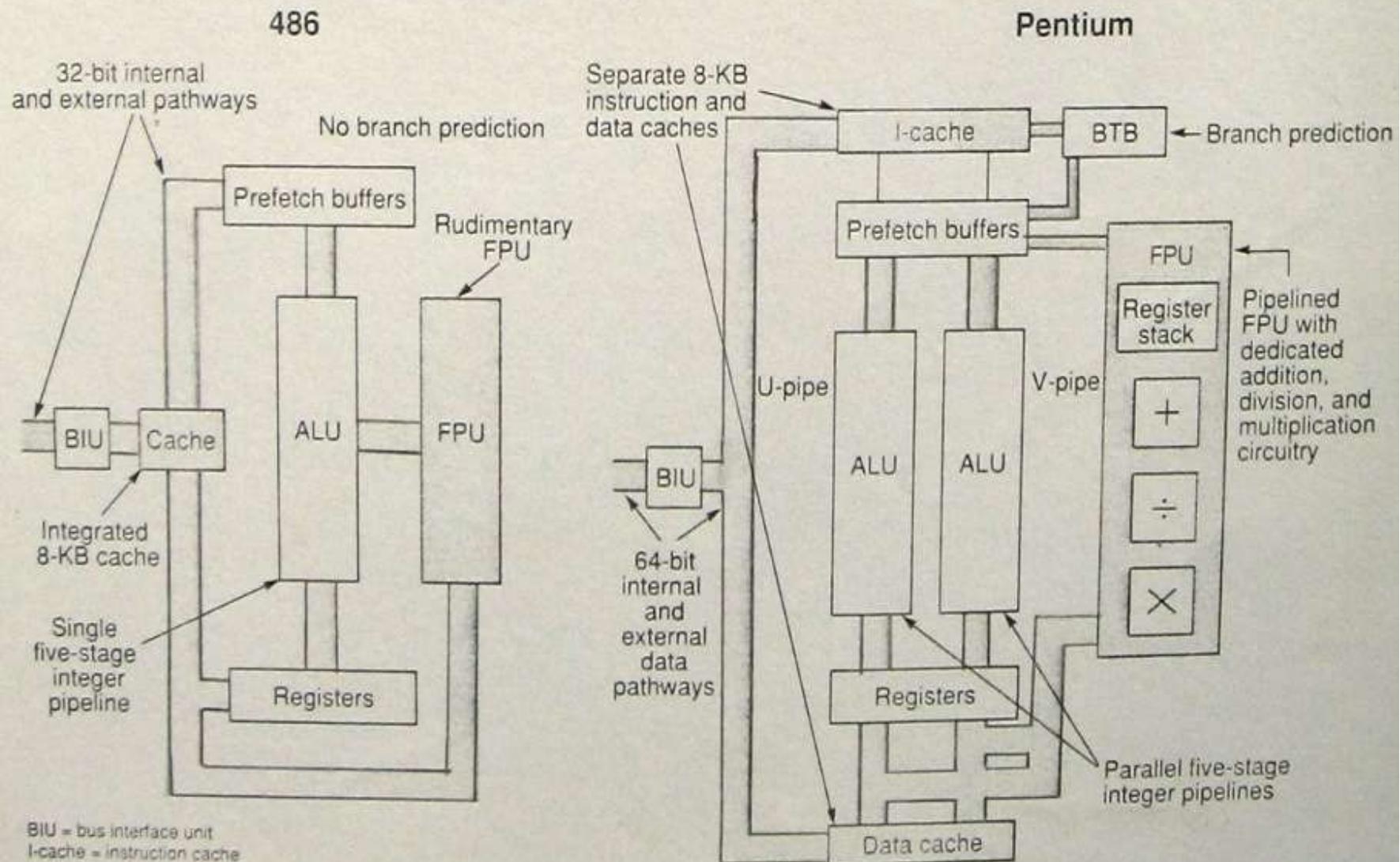
## I486™ Microprocessor Pipelined 32-Bit Microarchitecture



240440-1

RMX, iRMK, 386, 387, 486, i486 are trademarks of Intel Corporation.  
 MS-DOS® is a registered trademark of Microsoft Corporation.  
 OS/2™ is a trademark of Microsoft Corporation.  
 UNIX™ is a trademark of AT&T.

## 486 AND PENTIUM ARCHITECTURES



# EXISTE TODA UNA VARIEDAD DE CPUs DE INTEL QUE DEMUESTRAN MAS DE 50 AÑOS DE EVOLUCIÓN DE LOS MICROPROCESADORES.

Procesadores Intel Core i7 a 64 bits Socket 1366	
Clave:	Procesadores Intel i7 Socket 1366 S/X CORES ((Ventilador NO INCLUIDO))
CPUINT2020	Intel Core i7 EXTREME 390X a 64 bits, 6 CORES de 2.46 GHz con 12 MB de cache, Velocidad de Bus: 6.4 GT/s.
CPUINT2060	Intel Core i7 EXTREME 990X a 64 bits, 6 CORES de 2.33 GHz con 12 MB de cache, Velocidad de Bus: 6.4 GT/s.
Clave:	Procesadores Intel i7 Socket 1366 QUAD CORES
CPUINT2040	Intel Core i7 960 a 64 bits, Quad Core de 3.20 GHz con 8 MB de cache, Velocidad de Bus: 4.8 GT/s.
CPUINT1850	Intel Core i7 850 a 64 bits, Quad Core de 3.06 GHz con 8 MB de cache, Velocidad de Bus: 4.8 GT/s
Clave:	Procesadores Intel i7 Socket 1156
CPUINT1300	Intel Core i7-870 a 64 bits, Quad Core de 2.93 GHz con 8 MB de Cache, Velocidad de Bus: 2.5 GT/s
Intel Core i7-860 a 64 bits, Quad Core de 2.8 GHz con 8 MB de Cache, Velocidad de Bus: 2.5 GT/s (Existencias Limitadas)	
Clave:	Procesadores Intel i7 Segunda Generación Socket 1155
CPUINT548	Intel Core i7-2600K a 64 bits, Quad Core de 3.4 GHz (3.6GHz Turbo Boost) con 8 MB de Cache.
CPUINT2170	Intel Core i7-2600 a 64 bits, Quad Core de 3.4 GHz con 8 MB de Cache.
Clave:	Procesadores Intel i5 Segunda Generación-Socket 1155
CPUINT2200	Intel Core i5-2500 a 64 bits, Quad Core de 3.3 GHz con 6 MB de Cache.
CPUINT2150	Intel Core i5-2400 a 64 bits, Quad Core de 3.1GHz con 6 MB de Cache.
CPUINT2180	Intel Core i5-2300 a 64 bits, Quad Core de 2.8 GHz con 6 MB de Cache.
Clave:	Procesadores Intel i5 Socket 1156
CPUINT2080	Intel Core i5-760 a 64 bits, Quad Core de 2.80 GHz con 8 MB de Cache, Velocidad de Bus: 2.5GT/s
Intel Core i5-750 a 64 bits, Quad Core de 2.66 GHz con 8 MB de Cache, Velocidad de Bus: 2.5GT/s (Existencias Limitadas)	
CPUINT2090	Intel Core i5-660 a 64 bits, Dual Core de 3.60 GHz con 4 MB de Cache, Velocidad de Bus: 2.5 GT/s
CPUINT1600	Intel Core i5-660 a 64 bits, Dual Core de 3.33 GHz con 4 MB de Cache, Velocidad de Bus: 2.5 GT/s
CPUINT1990	Intel Core i5-660 a 64 bits, Dual Core de 3.20 GHz con 4 MB de Cache, Velocidad de Bus: 2.5 GT/s
Clave:	Procesadores Intel i3 Segunda Generación-Socket 1155
CPUINT2230	Intel Core i3-2120 a 64 bits, Dual Core de 3.3 GHz con 3 MB de Cache, Velocidad de Bus: 5 GT/s
CPUINT2240	Intel Core i3-2100 a 64 bits, Dual Core de 3.1 GHz con 3 MB de Cache, Velocidad de Bus: 5 GT/s
Clave:	Procesadores Intel i3 Socket 1156
CPUINT1970	Intel Core i3-540 a 64 bits, Dual Core de 3.06 GHz con 4 MB de Cache, Velocidad de Bus: 2.5 GT/s
Procesadores Intel Core 2 Quad a 64 bits	
Clave:	Procesadores Intel Core 2 Quad Socket (LGA 775) Serie QXXXX FSB 1,333 MHz
CPUINT1750	Core 2 Quad Q9650 a 64 bits, Cuatro Núcleo a 3 GHz con 12 MB de cache Bus de 1,333 MHz (Existencias Limitadas)
Core 2 Quad Q9550 a 64-bits, Cuatro Núcleo a 2.83 GHz con 12 MB de cache Bus de 1,333 MHz (Existencias Limitadas)	
CPUINT328	Core 2 Quad Q9450 a 64 bits, Cuatro Núcleo a 2.66 GHz con 6 MB de cache Bus de 1,333 MHz (Existencias Limitadas)
CPUINT1250	Core 2 Quad Q8400 a 64 bits, Doble Núcleo a 2.66 GHz con 6 MB de cache Bus de 1,333 MHz (Existencias Limitadas)
CPUINT1838	Core 2 Quad Q8400 a 64 bits, Doble Núcleo de 2.66 GHz con 4 MB de cache Bus de 1,333 MHz
CPUINT1790	Core 2 Quad Q8300 a 64 bits, Doble Núcleo de 2.50 GHz con 4 MB de cache Bus de 1,333 MHz
Procesadores Intel Core 2 Duo de Doble Núcleo a 64 bits	
Clave:	Procesadores Intel Core 2 Duo Socket (LGA 775) Serie BX00 FSB 1,333 MHz
CPUINT430	Core 2 Duo E8500 a 64 bits, Doble Núcleo de 3.16 GHz con 6 MB de cache Bus de 1,333 MHz (Existencias Limitadas)
CPUINT440	Core 2 Duo E8400 a 64 bits, Doble Núcleo de 3.00 GHz con 6 MB de cache Bus de 1,333 MHz (Existencias Limitadas)
Clave:	Procesadores Intel Core 2 Duo Socket (LGA 775) Serie TX00 FSB 1,066 MHz
CPUINT1940	Core 2 Duo E7600 a 64 bits, Doble Núcleo de 3.06GHz con 3 MB de cache Bus de 1,066 MHz
CPUINT1810	Core 2 Duo E7500 a 64 bits, Doble Núcleo de 2.93GHz con 3 MB de cache Bus de 1,066 MHz (Existencias Limitadas)
Procesadores Intel Pentium Dual Core de Doble Núcleo	
Clave:	Procesadores Intel Pentium Dual Core Socket (LGA 1155)
CPUINT2130	Pentium Dual Core modelo G850 de 2.9 GHz con 3MB Cache.
CPUINT2270	Pentium Dual Core modelo G840 de 2.8 GHz con 3MB Cache.
CPUINT2280	Pentium Dual Core modelo G420 de 2.6 GHz con 3MB Cache.
Clave:	Procesadores Intel Pentium Dual Core Socket (LGA 775) serie E8XX0 FSB 1066 MHz a 64 bits
CPUINT2260	Pentium Dual Core modelo E5700 de 3.20 GHz con 2MB Cache, Bus de 1066 MHz
CPUINT2290	Pentium Dual Core modelo E6600 de 3.06 GHz con 2MB Cache, Bus de 1066 MHz
CPUINT1930	Pentium Dual Core modelo E5500 de 2.93 GHz con 2MB Cache, Bus de 1066 MHz
Clave:	Procesadores Intel Pentium Dual Core Socket (LGA 775) serie ESXX0 FSB 800 MHz a 64 bits
CPUINT2230	Pentium Dual Core modelo E5300 de 3.2 GHz con 2MB Cache, Bus de 800 MHz
CPUINT2140	Pentium Dual Core modelo E5700 de 3.0 GHz con 2MB Cache, Bus de 800 MHz
CPUINT2110	Pentium Dual Core modelo E5500 de 2.80 GHz con 2MB Cache, Bus de 800 MHz (Existencias Limitadas)
CPUINT1770	Pentium Dual Core modelo E5300 de 2.80 GHz con 2MB Cache, Bus de 800 MHz (Existencias Limitadas)
CPUINT1780	Procesadores Intel Pentium Dual Core Socket (LGA 775)
Pentium Dual Core modelo E2220 de 2.40 GHz con 1MB Cache, Bus de 800 MHz (Existencias Limitadas)	
Procesadores Intel Celeron	
Clave:	Procesadores Intel Celeron Dual Core Socket (LGA 775) serie E1X00 FSB 800 MHz
CPUINT2150	Celeron Dual Core E3500 de 2.7 GHz, con 1 MB Cache, Bus de 800 MHz
CPUINT2160	Celeron Dual Core E3400 de 2.6 GHz, con 1 MB Cache, Bus de 800 MHz
CPUINT2282	Celeron Dual Core E1300 de 2.5 GHz, con 1 MB Cache, Bus de 800 MHz (Existencias Limitadas)
CPUINT1836	Celeron modelo 640 de 2 GHz con 512K Cache, Bus de 800 MHz (Existencias Limitadas)
CPUINT1840	Celeron modelo 430 a 1.8 Ghz con 512 KB de cache con Bus de 800MHz

# Otros Procesadores de Intel, mas recientes:

	Year	Process	Price	Base / Turbo	Cores / Threads	Socket
Core i7-4790K	2013	22nm	\$339	4.0GHz / 4.4GHz	4 / 8	LGA1150
Core i5-4670K	2013	22nm	\$242	3.5GHz / 3.9GHz	4 / 4	LGA1150
Core i3-4350	2013	22nm	\$138	3.6GHz	2 / 4	LGA1150
Pentium G3220	2013	22nm	\$64	3.0GHz	2 / 2	LGA1150
Celeron G1820	2014	22nm	\$42	2.7GHz	2 / 2	LGA1150
Core i7-2700K	2011	32nm	\$332	3.5GHz / 3.9GHz	4 / 8	LGA1155
Core i5-2500K	2011	32nm	\$216	3.3GHz / 3.7GHz	4 / 4	LGA1155
Core i7-870	2009	45nm	\$562	2.93GHz / 3.6GHz	4 / 8	LGA1156
Core i5-760	2009	45nm	\$205	2.8GHz / 3.33GHz	4 / 4	LGA1156
Core 2 Quad Q9650	2008	45nm	\$530	3.0GHz	4 / 4	LGA775
Core 2 Quad Q6600	2007	65nm	\$530	2.4GHz	4 / 4	LGA775
Core 2 Duo E8600	2008	45nm	\$266	3.33GHz	2 / 2	LGA775
Core 2 Duo E6600	2006	65nm	\$316	2.4GHz	2 / 2	LGA775

Procesadores Intel Core i3, i5, i7, i9 Décima Generación				
	Clave	Procesadores Intel Core i9 Décima Generación	Similar	Precio
CPUINT3820		Intel Core i9-11900K Rocket Lake 3.50GHz (5.3GHz Turbo) LGA 1200 16 MB Intel Smart Cache, 8 Núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 500. NO INCLUYE VENTILADOR. Intel Core i9-11900KF Rocket Lake 3.50GHz (5.3GHz Turbo) LGA 1200 16 MB Intel Smart Cache, 8 Núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 500. NO INCLUYE VENTILADOR, NI TARJETA DE VIDEO.	CPUINT3820	\$ 526.35
CPUINT3950		Intel Core i9-10900K Comet Lake 3.70GHz (5.20GHz Turbo) LGA 1200 20 MB Intel Smart Cache. Intel UHD Graphics 630, 10 Núcleos y 20 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. (NO INCLUYE VENTILADOR)	CPUINT3950	\$ 499.95
CPUINT3540		Intel Core i9-10900KF Comet Lake 3.70GHz (5.20GHz Turbo) Frecuencia de la Tecnología Intel Turbo Boost Max 3.0 5.20GHz. LGA 1200 20 MB Intel Smart Cache. Intel UHD Graphics 630, 10 Núcleos y 20 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. NO INCLUYE (VENTILADOR, TARJETA DE VIDEO)	CPUINT3540	\$ 427.63
CPUINT3690		Intel Core i9-10900KA Comet Lake 3.70GHz (5.20GHz Turbo) Frecuencia de la Tecnología Intel Turbo Boost Max 3.0 5.20GHz. LGA 1200 20 MB Intel Smart Cache. Intel UHD Graphics 630, 10 Núcleos y 20 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. (NO INCLUYE VENTILADOR) EDICION LIMITADA AVENGERS	CPUINT3690	\$ 402.60
CPUINT3670		Intel Core i9-10900 Comet Lake 2.80GHz (5.20GHz Turbo) LGA 1200 20 MB Intel Smart Cache. Intel UHD Graphics 630, 10 Núcleos y 20 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400.	CPUINT3670	
CPUINT3580		Intel Core i9-10900F Comet Lake 2.80GHz (5.20GHz Turbo) LGA 1200 20 MB Intel Smart Cache, 10 Núcleos y 20 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. NO INCLUYE GRAFICOS, NECESITA TARJETA DE VIDEO INDEPENDIENTE.	CPUINT3580	\$ 381.70
CPUINT3660		Intel Core i9-10850KA Comet Lake 3.60GHz (5.20GHz Turbo) LGA 1200 20 MB Intel Smart Cache. Intel UHD Graphics 630, 10 Núcleos y 20 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. (NO INCLUYE VENTILADOR) VERSION LIMITADA AVENEGERS	CPUINT3660	\$ 418.00
CPUINT3610		Intel Core i9-10850K Comet Lake 3.60GHz (5.20GHz Turbo) LGA 1200 20 MB Intel Smart Cache. Intel UHD Graphics 630, 10 Núcleos y 20 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. (NO INCLUYE VENTILADOR) VERSION LIMITADA AVENEGERS	CPUINT3610	\$ 400.40
CPUINT3620		Intel Core i9-10850K Comet Lake 3.60GHz (5.20GHz Turbo) LGA 1200 20 MB Intel Smart Cache. Intel UHD Graphics 630, 10 Núcleos y 20 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. (NO INCLUYE VENTILADOR)	CPUINT3620	\$ 393.25
CPUINT3700		Intel Core i9-9900K CoffeeLake 3.60GHz (5.00GHz Turbo) LGA 1151 16 MB Intel® Smart Cache. Gráficos UHD Intel® 630, 8 núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 300. NO INCLUYE (VENTILADOR)	CPUINT3700	\$ 286.00
CPUINT3640		Intel Core i9-9900 CoffeeLake 3.10GHz (5.00GHz Turbo) LGA 1151 16 MB Intel® Smart Cache. Gráficos UHD Intel® 630, 8 núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 300	CPUINT3640	\$ 305.80
	Clave	Procesadores Intel Core i7 Décima Generación	Similar	Precio
CPUINT3760		Intel Core i7-11700 Rocket Lake 2.50GHz (4.90GHz Turbo) LGA 1200 16 MB Intel Smart Cache, 8 Núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 500.	CPUINT3760	\$ 320.10
CPUINT3770		Intel Core i7-11700F Rocket Lake 2.50GHz (4.90GHz Turbo) LGA 1200 16 MB Intel Smart Cache, 8 Núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 500. NECESITA TARJETA DE VIDEO INDEPENDIENTE.	CPUINT3770	\$ 297.00
CPUINT3780		Intel Core i7-11700K Rocket Lake 3.60GHz (5.0GHz Turbo) LGA 1200 16 MB Intel Smart Cache, 8 Núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 500. NO INCLUYE VENTILADOR.	CPUINT3780	
CPUINT3790		Intel Core i7-11700KF Rocket Lake 3.60GHz (5.0GHz Turbo) LGA 1200 16 MB Intel Smart Cache, 8 Núcleos y 16 subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 500. NO INCLUYE VENTILADOR.	CPUINT3790	
CPUINT3680		NECESITA TARJETA DE VIDEO INDEPENDIENTE.		
CPUINT3520		Intel Core i7-10700 Comet Lake 2.90GHz (4.80GHz Turbo). LGA 1200 16 MB Intel Smart Cache. Intel UHD Graphics 630, 8 Núcleos y 16 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400.	CPUINT3760	
CPUINT3530		Intel Core i7-10700K Comet Lake 3.80GHz (5.10GHz Turbo) LGA 1200 16 MB Intel Smart Cache. Intel UHD Graphics 630, 8 Núcleos y 16 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. (NO INCLUYE VENTILADOR)	CPUINT3530	
CPUINT3600		Intel Core i7-10700KA Comet Lake 3.80GHz (5.10GHz Turbo) LGA 1200 16 MB Intel Smart Cache. Intel UHD Graphics 630, 8 Núcleos y 16 Subprocesos. COMPATIBLE SOLO CON MOTHERBOARDS CHIPSET SERIE 400. (NO INCLUYE VENTILADOR) VERSION LIMITADA AVENEGERS	CPUINT3780	

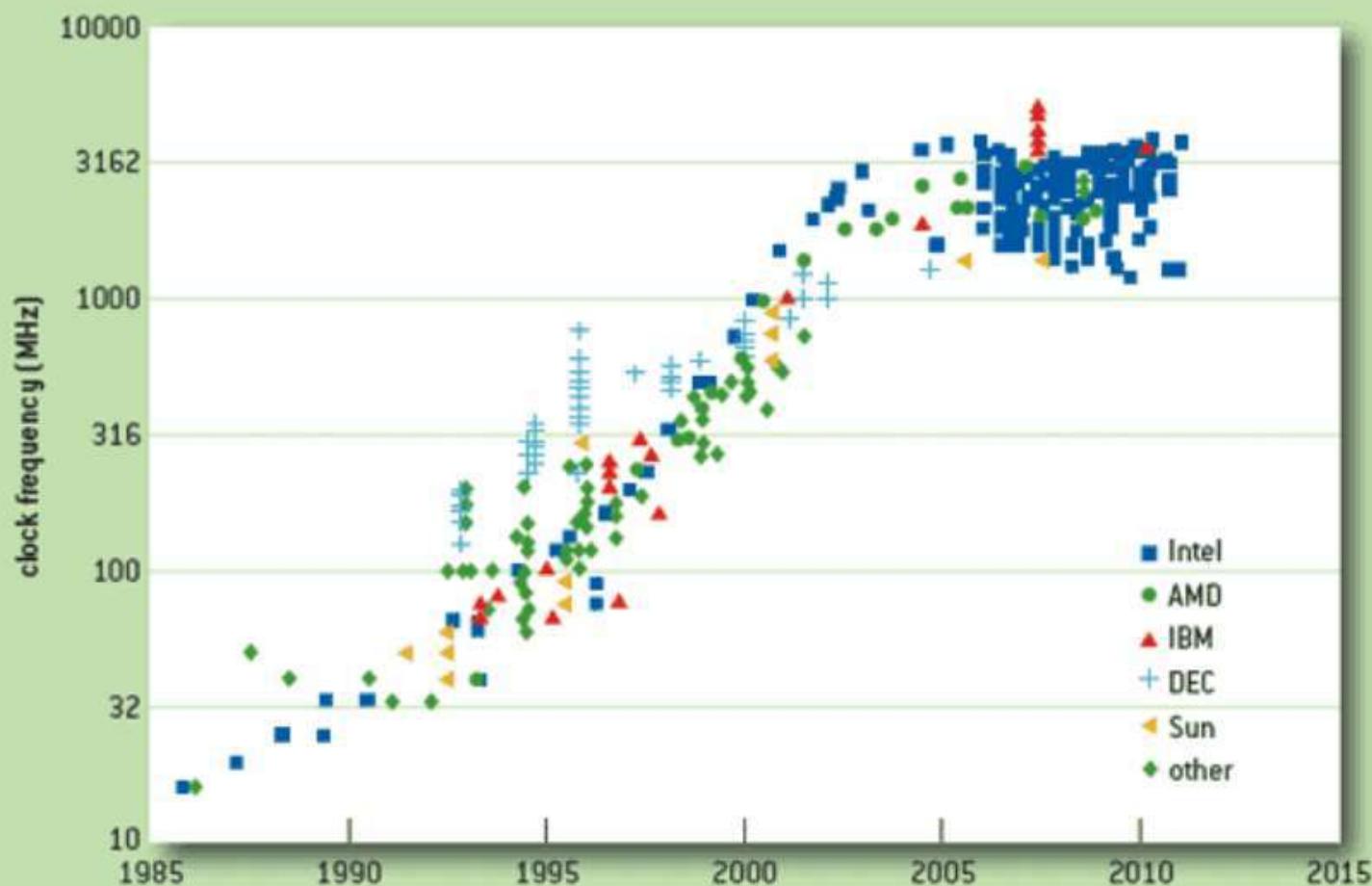
# AMD:

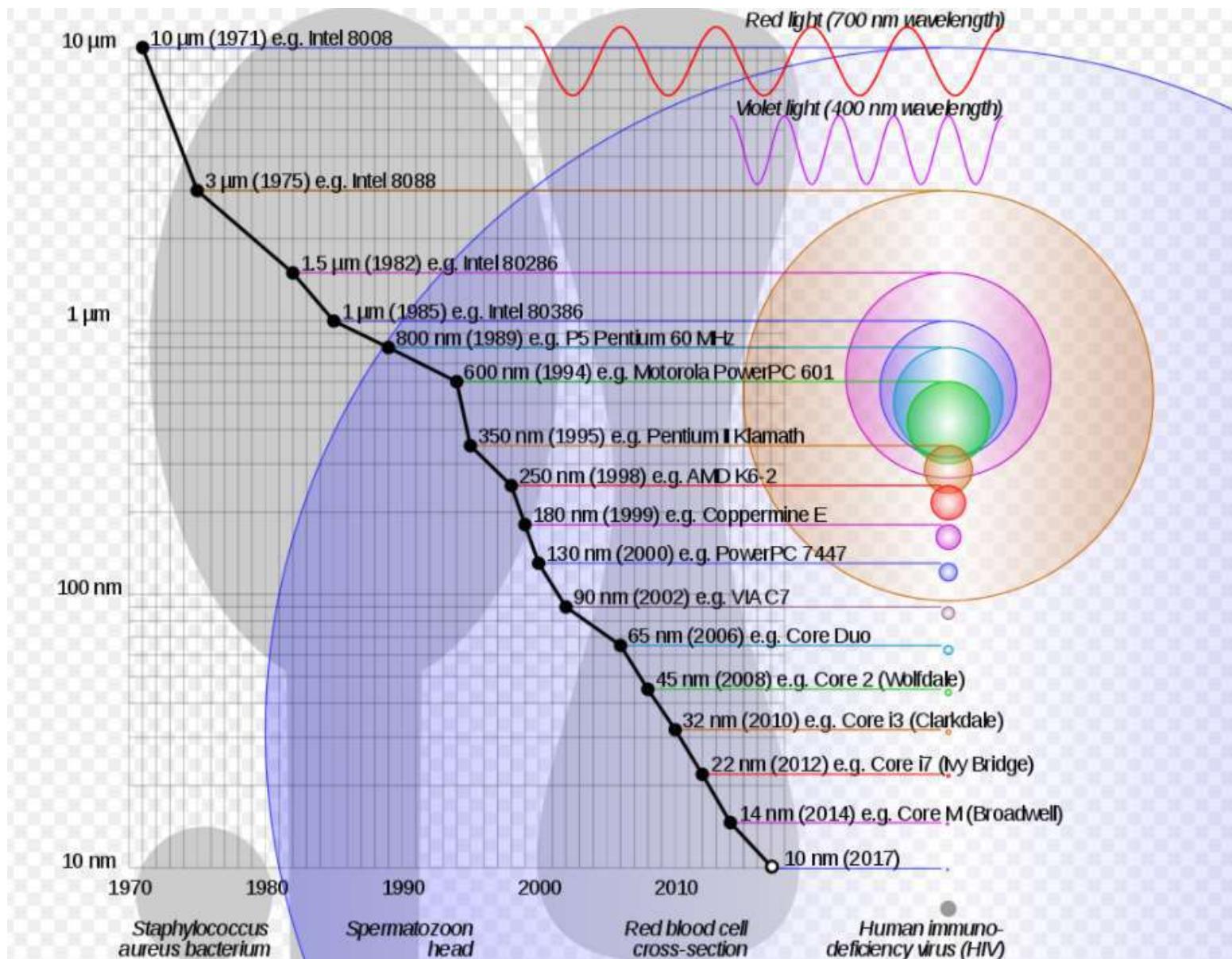
Procesador AMD RYZEN			
Clave	AMD Ryzen Socket AM4 3da Generación	Similar	Precio
CPUAMD2000	Procesador <b>Ryzen 5 3400G</b> CPU con <b>4 núcleos y 8 hilos</b> a <b>3,7 GHz-4,2 GHz</b> , modo normal y turbo, <b>4 MB</b> de caché, GPU Radeon RX Vega 11, de memoria <b>DDR4-2933</b> , TDP de <b>65w</b> , <b>Socket AM4</b> ,Core Technology 12nm Zen+.	CPUAMD2060	\$ 97.35
CPUAMD1990	*Verifica que tu tarjeta madre este preparada para ryzen serie 3000*	CPUAMD1990	\$ 159.50
CPUAMD1980	Procesador <b>Ryzen 5 3600</b> , CPU con <b>6 núcleos y 12 hilos</b> a <b>3,6GHz-4,2 GHz</b> , Cache <b>3MB</b> ,TDP <b>65W</b> , modo normal y turbo, memoria <b>DDR4</b> , <b>Socket AM4</b> *Verifica que tu tarjeta madre este preparada para ryzen serie 3000*	CPUAMD1980	\$ 193.60
CPUAMD1850	AMD <b>RYZEN 5 2600X</b> socket <b>AM4</b> , núcleos de <b>CPU 6</b> , subprocesos <b>12</b> , Reloj base <b>3.6GHz</b> , Reloj de aumento máx. Hasta <b>4.2GHz</b> ,Caché L1 total <b>576KB</b> , Caché L2 total <b>3MB</b> , Caché L3 total <b>16MB</b> , TDP/TDP predeterminado <b>65W</b> ,Velocidad máxima de memoria Up to <b>3200MHz</b> , Tipo de memoria <b>DDR4</b> , Canales de memoria <b>2</b> , Tecnologías compatibles <b>Tecnologías AMD PRO, Display Port Sí, HDMI Sí, INCLUYE VENTILADOR</b> .	CPUAMD1850	\$ 210.21
CPUAMD2080	Procesador <b>AMD Ryzen 5 3600XT, with Wraith Spire cooler Socket AM4</b> . <b>6 núcleos de CPU, 12 hilos</b> , Reloj base <b>3.8GHz</b> , Reloj de aumento máx. Hasta <b>4.5GHz</b> , Caché L1 total, <b>32mb</b> , Caché L2 total <b>3MB</b> , Caché L3 total, <b>16MB</b> , <b>Desbloqueados Sí</b> , CMOS TSMC 7nm FinFET Package AM4 Versión de PCI Express PCIe 4.0, TDP/TDP predeterminado <b>65W</b> , <b>Velocidad máxima de memoria 3200MHz</b> , <b>Tipo de memoria DDR4 (No incluye gráficos)</b>	CPUAMD2080	\$ 264.00
CPUAMD2290	Procesador AMD <b>RYZEN 7 5700G</b> núcleos de <b>CPU 8</b> , hilos <b>16</b> , Reloj base <b>3.8GHz</b> , Reloj de aumento máx. Hasta <b>4.6GHz</b> ,Caché L2 total <b>4MB</b> , Caché L3 total <b>16MB</b> , Desbloqueados No, CMOS 7nm Package <b>AM4</b> , Versión de PCI Express, PCIe 3.0, Solución térmica (MPK) Wraith Spire, TDP/TDP predeterminado <b>65W</b> , Temp. máx. <b>95°C</b> , Compatible con <b>SO Windows 10</b> edición de 64-bits, RHEL x86 edición de 64-bits, Ubuntu x86 edición de 64-bits, El soporte del sistema operativo (SO) variará según el fabricante. Memoria <b>Velocidad máxima de memoria Up to 3200MHz</b> , Tipo de memoria <b>DDR4</b> , Canales de memoria <b>2</b> , Especificaciones de gráficos <b>Frecuencia de gráficos 2000 MHz</b> , Modelo de gráficos <b>Radeon™ Graphics</b> , Cant. de núcleos de los gráficos <b>8</b> , Funcionalidades principales <b>Display Port Sí, HDMI S.(incluye gráficos)</b>	CPUAMD2290	\$ 350.90
CPUAMD2270	Procesador AMD <b>RYZEN 5 5600G RADEON VEGA</b> núcleos de <b>CPU 6</b> , hilos <b>12</b> ,núcleos de <b>GPU 7</b> ,Reloj base <b>3.9GHz</b> , Reloj de aumento máx. Hasta <b>4.4GHz</b> ,Caché L2 total <b>3MB</b> , Caché L3 total <b>16MB</b> , Desbloqueados <b>Sí</b> , CEMOS: <b>TSMC 7nm FinFET Package AM4</b> , Versión de PCI Express, PCIe 3.0, Solución térmica (MPK) Wraith Stealth, TDP/TDP predeterminado <b>65W</b> , Temp. máx. <b>95°C</b> , Compatible con <b>SO Windows 10</b> edición de 64-bits, RHEL x86 edición de 64-bits, Ubuntu x86 edición de 64-bits, El soporte del sistema operativo (SO) variará según el fabricante. Memoria <b>Velocidad máxima de memoria Up to 3200MHz</b> , Tipo de memoria <b>DDR4</b> , Canales de memoria <b>2</b> , Especificaciones de gráficos <b>Frecuencia de gráficos 1900 MHz</b> , Modelo de gráficos <b>Radeon™ Graphics</b> , Cant. de núcleos de los gráficos <b>7</b> , Funcionalidades principales <b>Display Port Sí, HDMI S.(incluye gráficos)</b>	CPUAMD2270	\$ 249.98
CPUAMD2140	Procesador <b>AMD Ryzen 5 5600X, with Wraith Spire cooler Socket AM4</b> . <b>6 núcleos de CPU, 12 hilos</b> , Reloj base <b>3.7GHz</b> , Reloj de aumento máx. Hasta <b>4.6GHz</b> , Caché L1 total, <b>32mb</b> , Caché L2 total <b>3MB</b> , Caché L3 total, <b>16MB</b> , <b>Desbloqueados Sí</b> , CMOS TSMC 7nm FinFET Package AM4 Versión de PCI Express PCIe 4.0, TDP/TDP Solución térmica (PIB), Wraith Stealth, predeterminado <b>65W</b> , <b>Velocidad máxima de memoria 3200MHz</b> , <b>Tipo de memoria DDR</b> , <b>AMD "Zen 3" Core Architecture, Tecnología AMD StoreMI, Utilidad AMD Ryzen™ Master, 4 (No incluye gráficos)</b>	CPUAMD2140	\$ 282.43
CPUAMD1960	Procesador <b>Ryzen 7 3700X</b> , CPU con <b>8 núcleos y 16 hilos</b> a <b>3,6 GHz-4,4 GHz</b> , Cache <b>4MB</b> , TDP <b>65W</b> , memoria <b>DDR4</b> , <b>Socket AM4</b> . *Verifica que tu tarjeta madre este preparada para ryzen serie 3000*	CPUAMD1960	\$ 310.20

FIGURE

7

## Processor Frequency Scaling Over Time





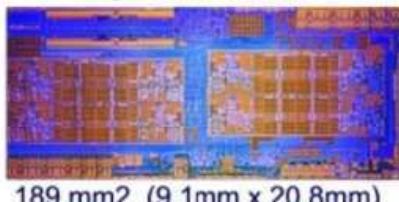
# AMD Ryzen versus Intel Skylake-X

[www.chip-architect.com](http://www.chip-architect.com) 17-06-2017

[www.chip-architect.com](http://www.chip-architect.com) 17-06-2017

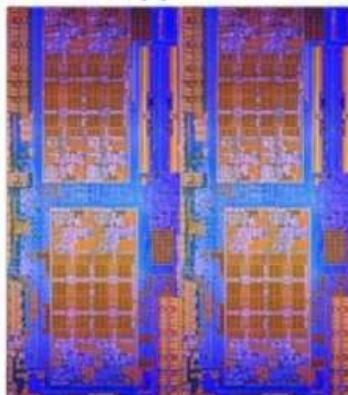


Ryzen 8 cores



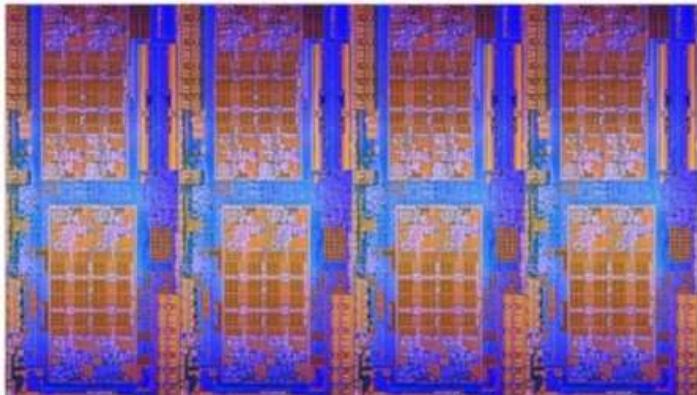
189 mm<sup>2</sup> (9.1mm x 20.8mm)

Threadripper 16 cores

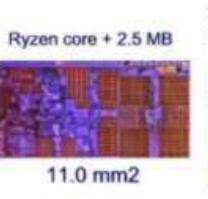


378 mm<sup>2</sup>

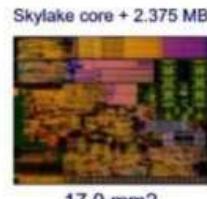
Epyc 32 cores



756 mm<sup>2</sup>

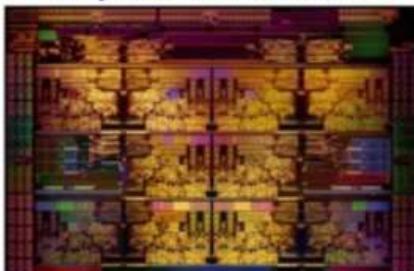


11.0 mm<sup>2</sup>



17.0 mm<sup>2</sup>

Skylake-X 10 cores



322 mm<sup>2</sup> (14.3mm x 22.4mm)

Skylake-X 18 cores



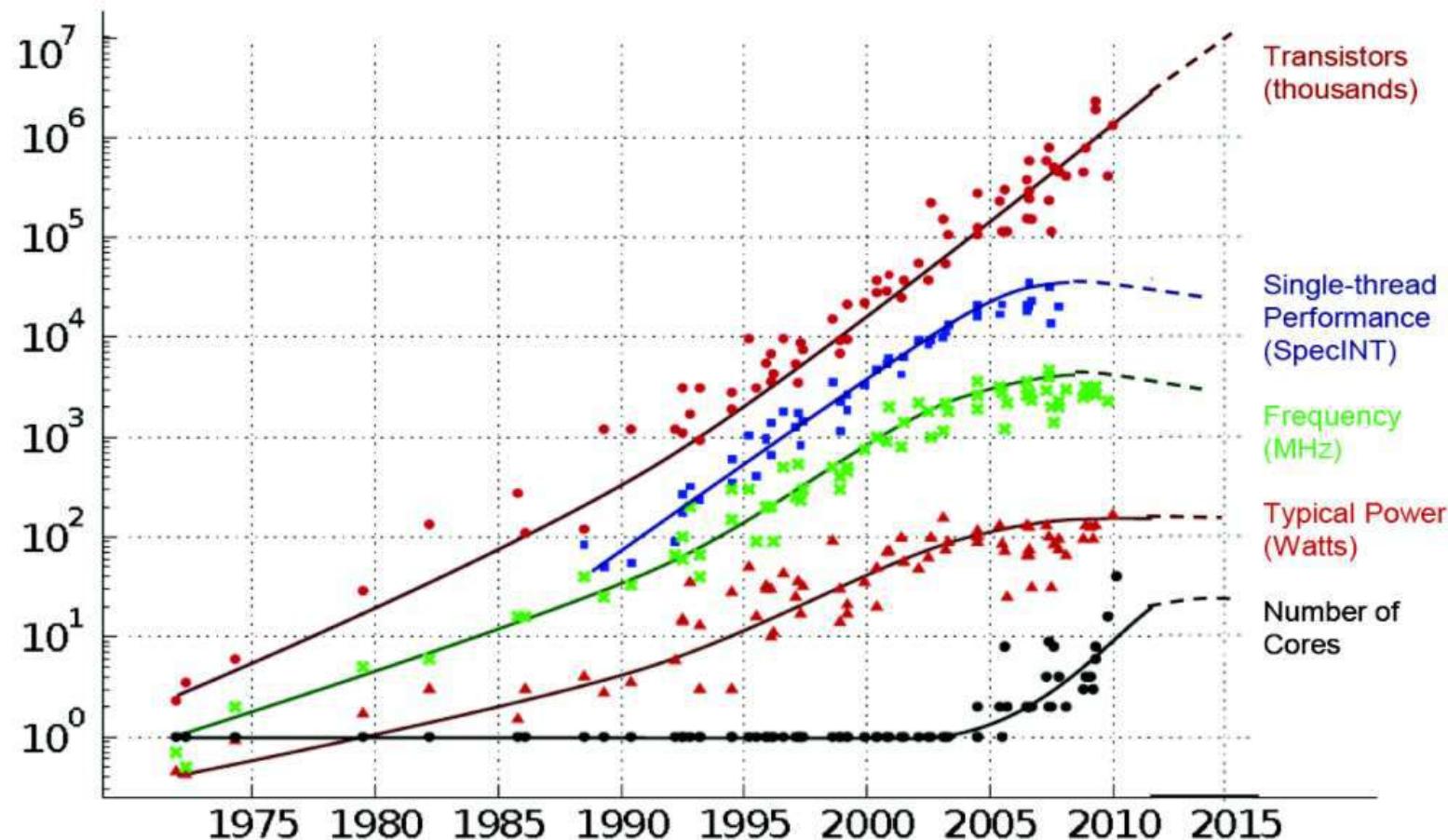
484 mm<sup>2</sup> (21.6mm x 22.4mm)

Skylake-X 28 cores



698 mm<sup>2</sup> (21.6mm x 32.3mm)

# 35 YEARS OF MICROPROCESSOR TREND DATA

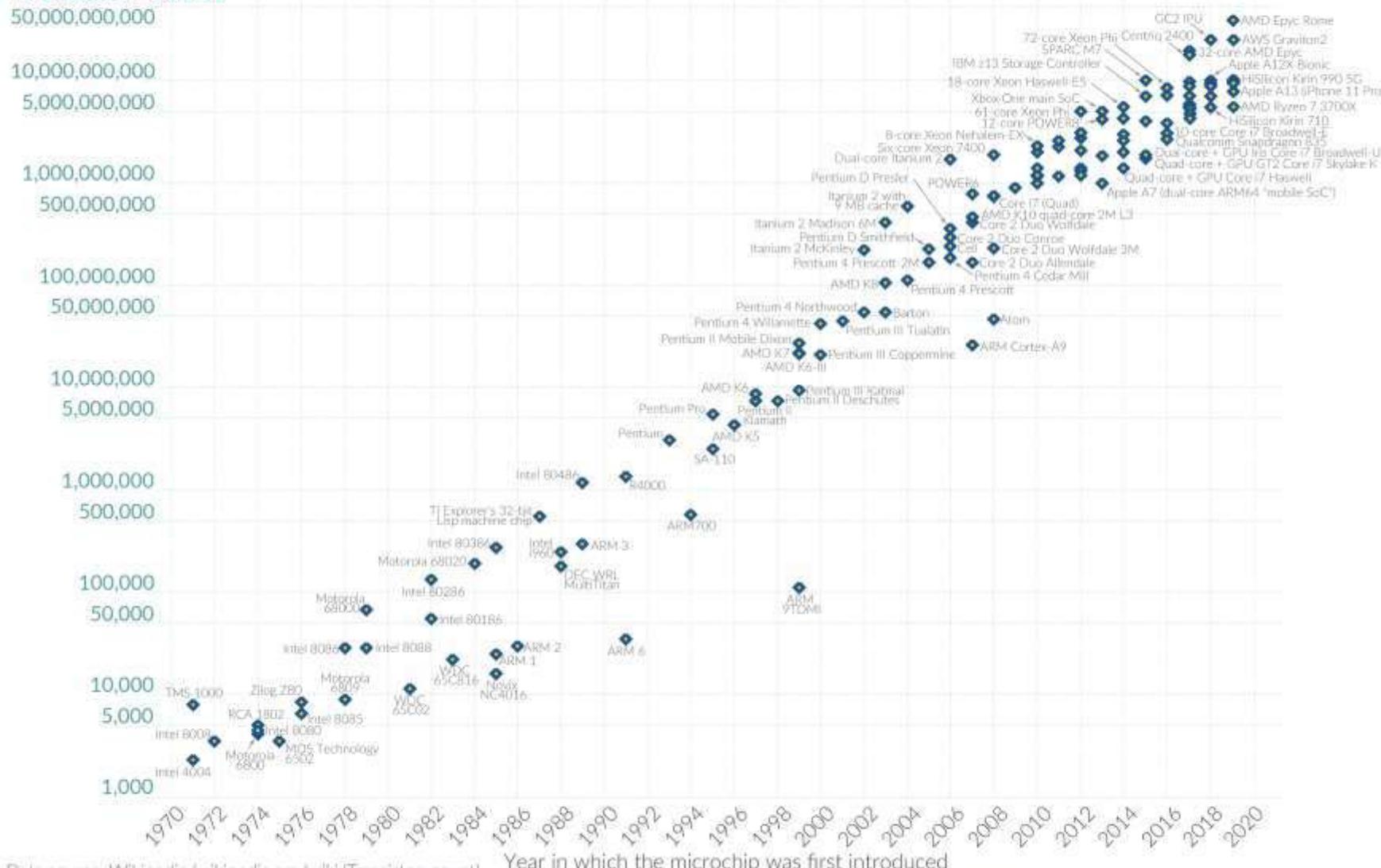


Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten  
Dotted line extrapolations by C. Moore

# Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

## Transistor count





# Corporativo de Intel hoy











# **APENDICES**

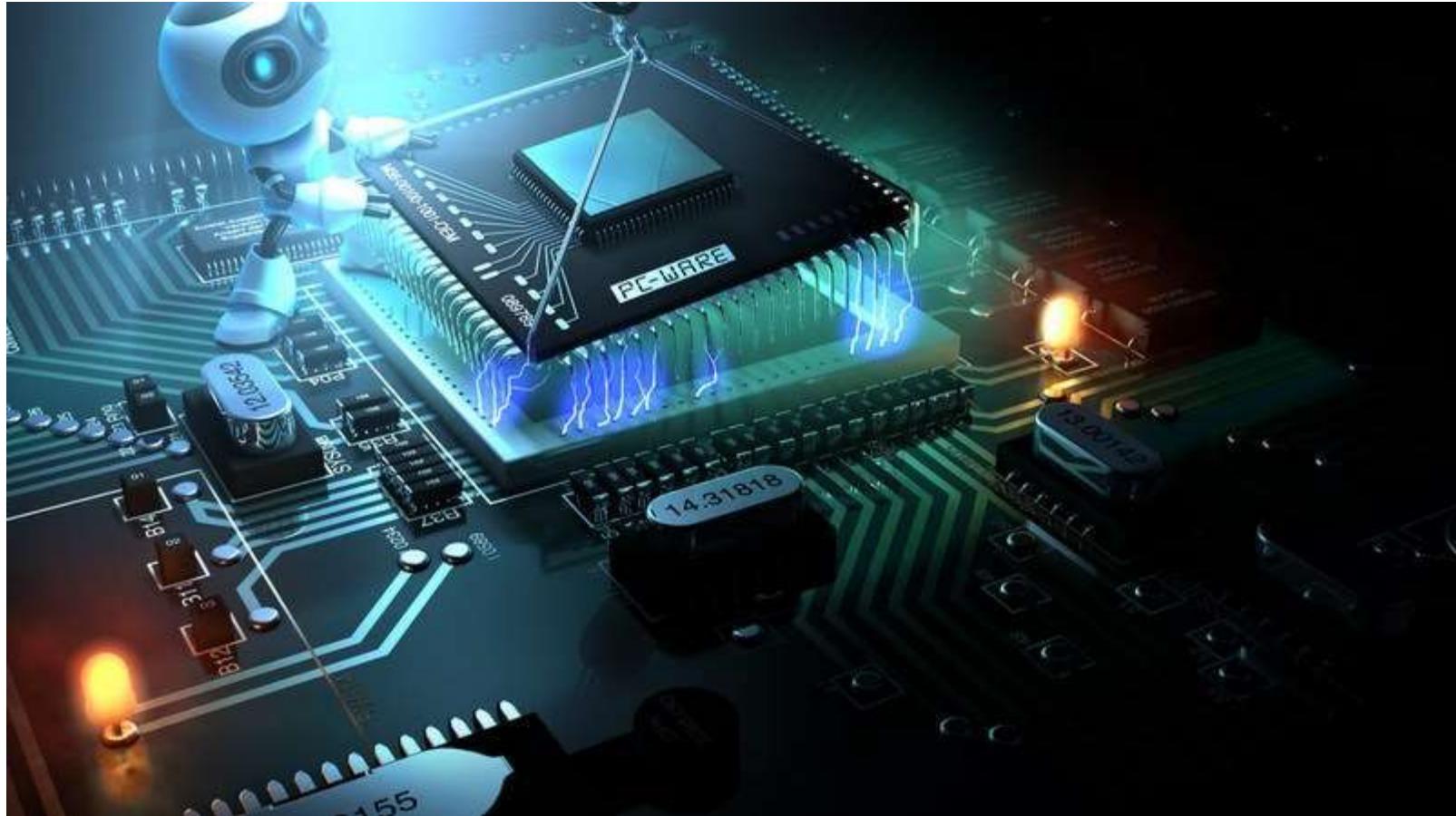
## EJERCICIOS Y NOTAS:

## EJERCICIOS Y NOTAS:



# LENGUAJE ENSAMBLADOR

## (7°-ICI)

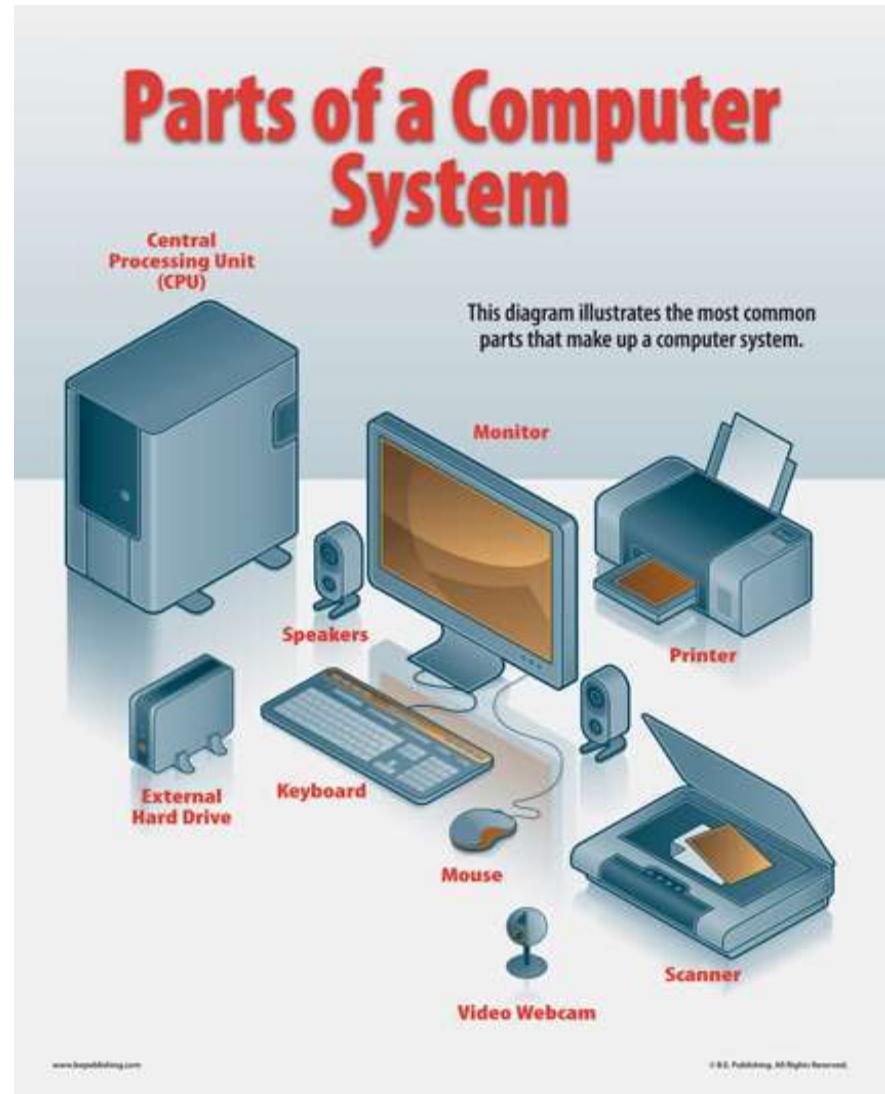


**Mtro. en Ing. Armando Álvarez Fdez.**  
**Ago-2024**

# FUNDAMENTOS DE UN SISTEMA DE COMPUTO

## ¿Qué es el sistema de Computo?

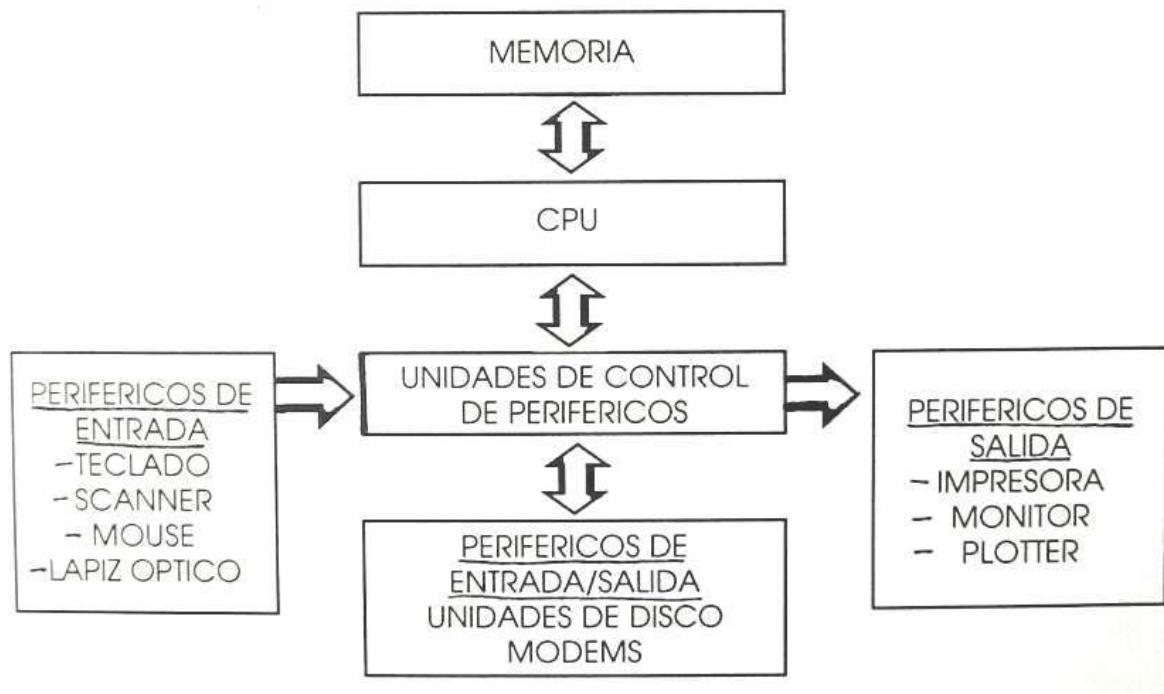
Un sistema de Computo es un conjunto de dispositivos integrados que ingresan, generan, procesan y almacenan datos e información. Los sistemas de Computo se construyen actualmente alrededor de al menos un dispositivo de procesamiento digital (CPU) y todo su hardware de soporte, así como de programas de software y del talento humano que los vuelve productivos.



# ¿Por qué se llama sistema de Computo?

Se hace referencia a la computadora como un sistema porque está formada por componentes integrados (entrada, salida, almacenamiento y CPU) que trabajan juntos para realizar los pasos necesarios en el programa que se está ejecutando.

## ARQUITECTURA GENERAL DE UNA COMPUTADORA



Como se puede observar, en primer lugar tenemos al CPU (Central Processing Unit) ó unidad central de procesamiento, que es el corazón ó el cerebro de todo sistema y la encargada de procesar los datos de un programa.

El CPU requiere de un lugar de almacenaje de información o memoria, donde serán guardados el programa de computación así como los datos o información generada. Como será visto mas adelante, la memoria se divide en dos tipos: Memoria RAM (Memoria volátil o de acceso aleatorio), generalmente para el usuario y sus programas, así como Memoria ROM (Memoria no volátil o de solo lectura), para el uso del hardware así como funciones elementales y técnicas de operación, generalmente poco accesadas o conocidas por el usuario convencional.

# COMPONENTES PRINCIPALES DE HARDWARE EN UN SISTEMA DE COMPUTO

Procesador

Almacenamiento (RAM y ROM)

Unidades de control de dispositivos periféricos

Dispositivos de Entrada,

Dispositivos de Salida

Dispositivos de Entrada/Salida



## *Partes Internas*



## *Partes Externas*



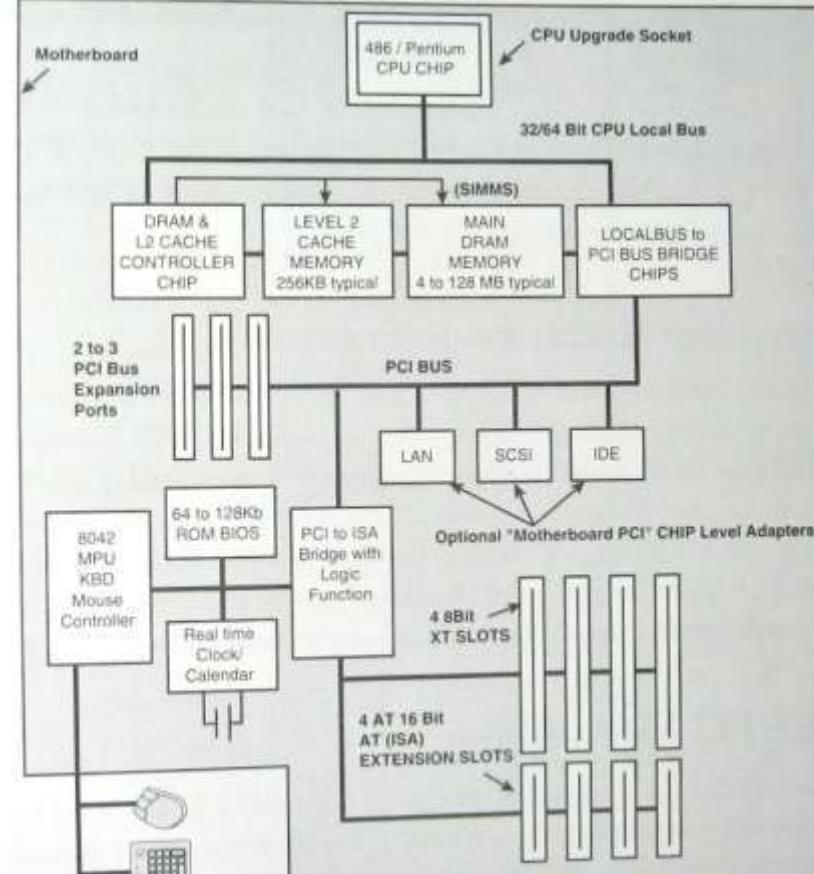
<http://personal.globaled.com/computacion-para-el-odontologo>

# Hardware

## The physical parts of a computer system



# TARJETA MADRE (MOTHERBOARD)



\*One ISA Slot is placed close to one PCI Slot for share use.



# **MICROPROCESADORES**

## **CISC Y RISC**

## MICROPROCESADORES RISC Y CISC.

Acrónimos:

RISC (Reduced Instruction Set Computers)

CISC (Complex Instruction Set Computers)

### RISC

Es un tipo de arquitectura especial en el diseño de los CPU's, que se basa en la implementación de un conjunto de instrucciones reducido y optimizadas al máximo, de tal manera que se ejecuten en un ciclo de reloj.

## ANTECEDENTES DE LA TECNOLOGIA RISC.

Anteriormente el diseño de CPU's se había centrado en CISC. Se buscaba a ver quien ponía mas cosas en un CPU.

**PORCENTAJES DE OCURRENCIA DE INSTRUCCIONES EN DIFERENTES LENGUAJES.**

TIPO DE INSTRUCCIONES	% FORTRAN	% C	% PASCAL	% PROMEDIO
-ASIGNACION	51	38	45	47
-CONDICIONALES	10	43	29	23
-LLAMADAS A PROCEDIMIENTOS	5	12	15	15
-ITERACIONES	9	3	5	6
-GOTO	9	3	0	3
-OTRAS	16	1	6	7

**CONCLUSION:**

"LOS CISC OFRECEN MAS DE LO QUE EN REALIDAD SE REQUIERE"

<b>ARQUITECTURAS ASPECTOS</b>	<b>CISC</b>	<b>RISC</b>
<i>Significado</i>	Computadoras con un conjunto de instrucciones reducido.	Computadoras con un conjunto de instrucciones complejo
<i>Aplicación</i>	Aplicada en ordenadores domésticos	Utilizada para entornos de red
<i>Características</i>	Instrucciones muy amplias.	Instrucciones de tamaño fijo.  Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.
<i>Objetivos</i>	Permitir operaciones complejas entre operandos situados en la memoria o en los registros internos	Posibilitar la segmentación y el paralelismo en la ejecución de instrucciones y reducir los accesos a memoria.
<i>Ventajas</i>	Reduce la dificultad de crear compiladores. Permite reducir el costo total del sistema. Mejora la compactación de código. Facilita la depuración de errores.	La CPU trabaja más rápido al utilizar menos ciclos de reloj. Reduciendo la ejecución de las operaciones. Cada instrucción puede ser ejecutada en un solo ciclo del CPU
<i>Microprocesadores Basados en:</i>	<ul style="list-style-type: none"> <li>• Intel 8086, 8088, 80286, 80386, 80486.</li> <li>• Motorola 68000, 68010, 68020, 68030, 6840</li> </ul>	MIPS Technologies  IBM POWER  PowerPC de Motorola e IBM  SPARC y UltraSPARC

# CUADRO COMPARATIVO

### PROBLEMAS QUE IMPIDEN LA EFICIENCIA DE UN CISC

MAYOR NUMERO DE = MAYOR TIEMPO GASTADO EN DECODIFICAR  
INTRUCCIONES. INSTRUCCIONES.

MAYOR NUM. DE MODOS = MAYOR TIEMPO GASTADO EN DETERMINAR  
DE DIRECCIONAMIENTO. = DIRECCIONES.

SE IMPLEMENTAN FUNCIONES QUE CASI NO SE USAN.

**CISC**  
(complex  
instructions)

010010111001001110111111001111  
1110001101001100100110011001111  
00011010100111010001111001111

010010111001001110111111001111  
1110001101001100100110011001111  
0001101010011101000111100111010  
001101001100100110011001111000  
11010100111010001111001111010001  
101001100100110011001111000110  
10100111010001111001111010

10100100

**RISC**  
( simple  
instructions)

01001011100100111  
101111110011111

0111100100100111  
0111111100111011

0111100111110100  
1011101110100111

0101110101011111  
1100111001001111

111010111101011111  
0111111100111101

CISC	RISC
Instrucciones multiciclo	Instrucciones de único ciclo
Arquitectura memoria-memoria	Arquitectura registro-registro
Instrucciones largas, Código con menos líneas	Instrucciones cortas, Código con más líneas
Se enfatiza la versatilidad del repertorio de instrucciones	Se añaden instrucciones nuevas sólo si son de uso frecuente y no reducen el rendimiento de las más importantes
Reduce la dificultad de implementar compiladores	Compiladores complejos

## HISTORIA.

Se considera que el primer CPU RISC surge cuando John Cocke (entre 1974 y 1975), publica un trabajo dentro de IBM y desarrollan la computadora 801 de IBM en 1975 con técnicas de RISC. (El trabajo se difundió hasta 1982). Por otro lado de 1980 a 1982 en Berkeley David Paterson y Carlo Sequin lanzaron su RISC I y RISC II.

### **REGLA DE ORO DE RISC:**

La regla más importante de RISC es sacrificar todo para reducir el ciclo de tiempo de la trayectoria de datos.

### COMPARACION

#### RISC

- Instrucc. sencillas en un ciclo
- Instrucc. ejecutadas por hardware completamente.
- Pocas instrucc. y modos.
- La complejidad está en el compilador.

#### CISC

- Instrucc. complejas en varios ciclos.
- Instrucc. interpretadas por el microprograma.
- Muchas instrucc. y modos.
- La complejidad está en el microprograma.

# RISC vs. CISC

- En realidad es difícil de comparar ambas arquitecturas puesto que tienen objetivos diferentes, una apuesta por la sencillez del hardware (RISC) y la otra por la aceleración en instrucciones (CISC).:

CISC	RISC
Énfasis en el Hardware (Velocidad)	Énfasis en el Software (Sencillez y rapidez en el Pipeline)
Incluye instrucciones complejas multi-reloj	Instrucciones con reloj simple
Instrucciones complejas (Más de 1000)	Instrucciones sencillas (Menos de 100)
Instrucciones que trabajan de memoria a memoria	Instrucciones que trabajan de registro a registro
Carga y almacenamiento incorporados en la misma instrucción	La carga y el almacenamiento son instrucciones independientes
El tamaño del código es pequeño	El tamaño del código es muy grande
Muchos ciclos de reloj por segundo	Pocos ciclos de reloj por segundo
Pocos registros para almacenar los resultados	Muchos registros para almacenar resultados
Alto costo de producción	Bajo costo de producción

## POWER PC (CPU RISC)

El 3 de julio de 1991 IBM, Apple, y Motorola firmaron un acuerdo para desarrollar el CPU Power PC.

Por una parte los creadores de Power PC. aseguran que las aplicaciones en DOS, MAC, ALUX, trabajarán tan rápido como los actuales. También se espera que posteriormente los desarrolladores de Software y Hardware desarrollen aplicaciones nativas en el sistema CSF/1 UNIX. Tanto Apple, IBM y Motorola seguirán trabajando en sus sistemas anteriores y aún mejorandolos.

### LA FAMILIA POWER PC.

- 601.- Primer POWER PC; 32 bits; 3 capas superescalares; 50,66 y 80 MHZ.  
Desempeño: 85 SPECint 92, 105 SPECfp 92 a 80 MHZ;1993.
- 602.-32 bits; 3 capas superescalares; 66 y 80 MHZ;Desempeño: 75 SPECint, 85 SPECfp a 80 MHZ;1993
- 604.- CPU veloz de punto flotante de doble presición; 32 bits; 4 capas superescalares.  
(Esta disponible aproximadamente el segundo semestre de 1994).
- 620.- CPU de 64 bits; 6 capas superescalares.  
(Esta disponible el 1er. semestre de 1995).

# ALGUNOS OTROS CPUs RISCs:

## Low-end and mobile systems [edit]

By the beginning of the 21st century, the majority of low-end and mobile systems relied on RISC architectures.<sup>[46]</sup> Examples include:

- The ARM architecture dominates the market for low power and low cost embedded systems (typically 200–1800 MHz in 2014). It is used in a number of systems such as most Android-based systems; the Apple iPhone and iPad, Microsoft Windows Phone (former Windows Mobile), RIM devices, Nintendo Game Boy Advance, DS, 3DS and Switch, Raspberry Pi, etc.
- IBM's PowerPC was used in the GameCube, Wii, PlayStation 3, Xbox 360 and Wii U gaming consoles.
- The MIPS line (at one point used in many SGI computers) was used in the PlayStation, PlayStation 2, Nintendo 64, PlayStation Portable game consoles, and residential gateways like Linksys WRT54G series.
- Hitachi's SuperH, originally in wide use in the Sega Super 32X, Saturn and Dreamcast, now developed and sold by Renesas as the SH4.
- Atmel AVR used in a variety of products ranging from Xbox handheld controllers and the Arduino open-source microcontroller platform to BMW cars.
- RISC-V, the open-source fifth Berkeley RISC ISA, with 32- or 64-bit address spaces, a small core integer instruction set, and an experimental "Compressed" ISA for code density and designed for standard and special purpose extensions.

## Desktop and laptop computers [edit]

- IBM's PowerPC architecture was used in Apple's Macintosh computers from 1994, when they began a switch from Motorola 68000 family processors, to 2005, when they transitioned to Intel x86 processors.<sup>[47]</sup>
- Some chromebooks use ARM-based platforms since 2012.<sup>[48]</sup>
- Apple uses self-designed processors based on the ARM architecture for its lineup of desktop and laptop computers since its transition from Intel processors,<sup>[49]</sup> and the first such computers were released in November 2020.<sup>[40]</sup>
- Microsoft uses Qualcomm<sup>[50]</sup> ARM-based processors for its Surface line.

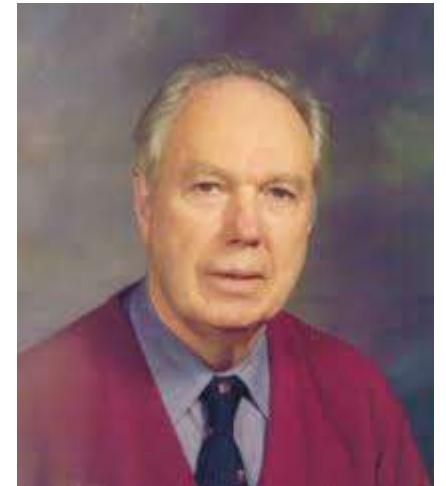
## Workstations, servers, and supercomputers [edit]

- MIPS, by Silicon Graphics (ceased making MIPS-based systems in 2006).
- SPARC, by Oracle (previously Sun Microsystems), and Fujitsu.
- IBM's IBM POWER instruction set architecture, PowerPC, and Power ISA were and are used in many of IBM's supercomputers, mid-range servers and workstations.
- Hewlett-Packard's PA-RISC, also known as HP-PA (discontinued at the end of 2008).
- Alpha, used in single-board computers, workstations, servers and supercomputers from Digital Equipment Corporation, then Compaq and finally Hewlett-Packard (HP)(discontinued as of 2007).
- RISC-V, the open source fifth Berkeley RISC ISA, with 64- or 128-bit address spaces, and the integer core extended with floating point, atomics and vector processing, and designed to be extended with instructions for networking, I/O, and data processing. A 64-bit superscalar design, "Rocket", is available for download. It is implemented in the European Processor Initiative processor.

# **TAXONOMÍA DE FLYNN**

# TAXONOMÍA DE FLYNN

La taxonomía de Flynn es una clasificación para las computadoras con arquitectura paralela, propuesta por el profesor emérito de la Universidad de Stanford **Michael J. Flynn**, la cual clasifica a las mismas atendiendo a la **cantidad de instrucciones** y **flujo de datos concurrentes** en un instante de procesamiento.



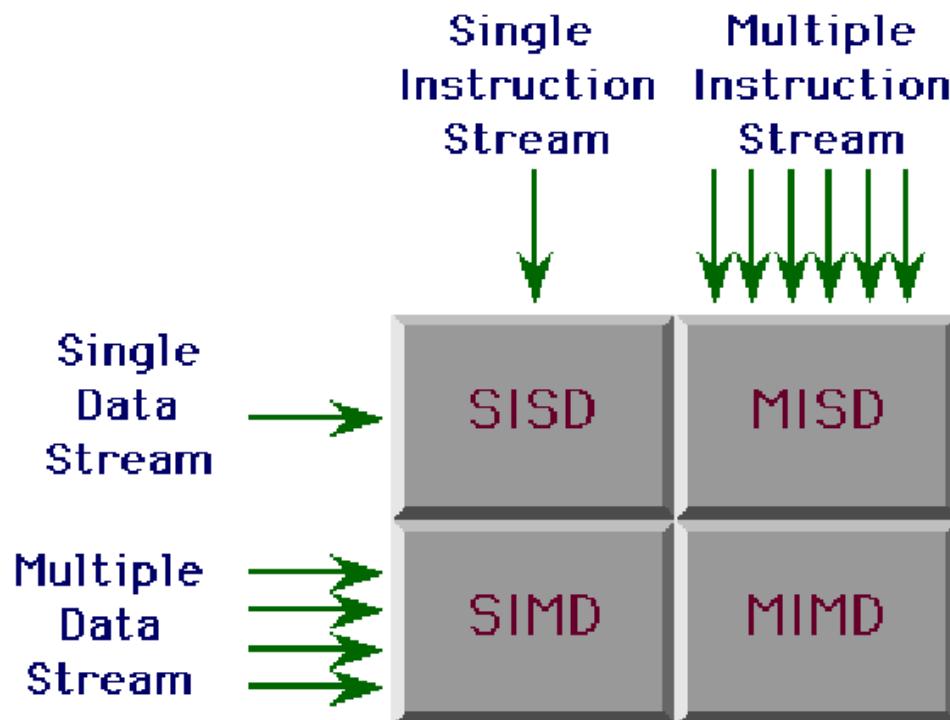
## Taxonomía de Flynn

Es la clasificación más extendida del paralelismo:

- Distingue entre instrucciones y datos
- Estos pueden ser simples o múltiples

		Datos	
		Simples	Múltiples
Instrucciones	Simples	SISD	SIMD
	Múltiples	MISD	MIMD

# TAXONOMÍA DE FLYNN



# SISD:(Single Instruction Single Data)

Esta clasificación se refiere a las **computadoras tradicionales y secuenciales** en las cuales una instrucción a la vez se ejecuta sobre **un único dato** cada ciclo de reloj.

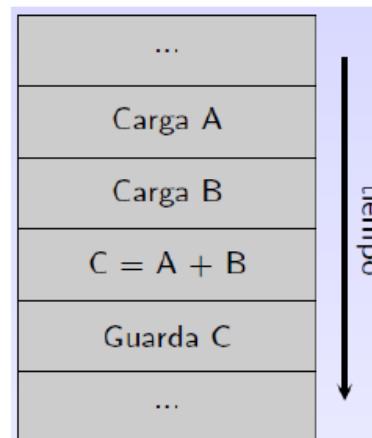
Los datos en cuestión se almacenan en una única memoria en la cual se **usan técnicas como la segmentación** para evitar errores de fragmentación interna. Un ejemplo sencillo de estas computadoras son los antiguos mainframe basados en la arquitectura de Von-Neumann.

## *SISD: Single Instruction, Single Data*

Características del modelo SISD:

- La CPU procesa únicamente **una instrucción por cada ciclo de reloj**
- Únicamente **un dato es procesado en cada ciclo de reloj**
- Es el **modelo más antiguo de computadora y el más extendido**

Ejemplo: la mayoría de las computadoras, servidores y estaciones de trabajo



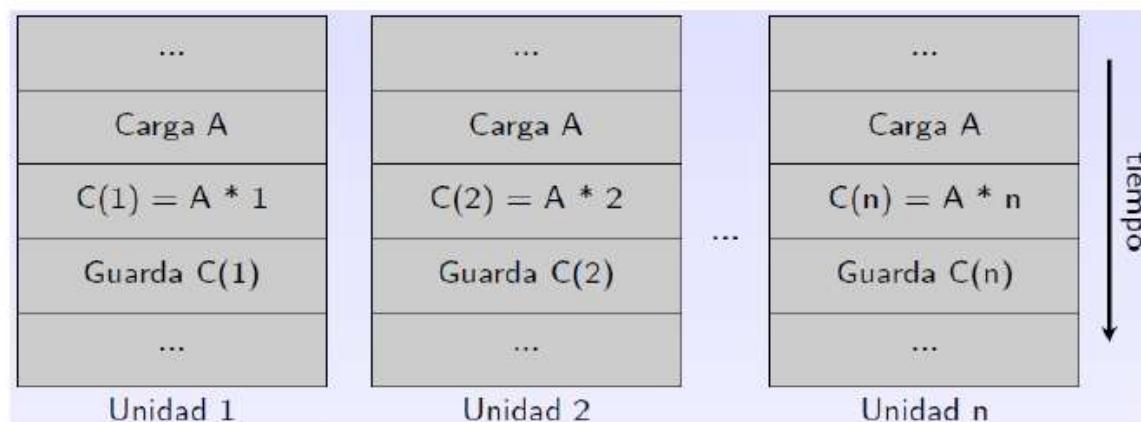
Constan de una única unidad de control de memoria (CU, control unit) y una única unidad aritmético-lógica (ALU), por tanto, sólo ejecuta una única instrucción en un único dato por ciclo de reloj. Corresponde a la arquitectura de von Neumann original.

# MISD: (Multiple Instruction Single Data)

Arquitectura que se refiere a **múltiples instrucciones** ejecutándose **sobre un único dato**. Comúnmente se considera esta **arquitectura poco práctica** ya que en tiempo de ejecución la efectividad del **parallelismo requiere un múltiple flujo de datos** y, además, el acceso concurrente a un mismo dato en memoria puede ocasionar que un **CPU tenga que esperar** a que el recurso(**dato**) esté disponible para poder acceder a él.

## *MISD: Multiple Instruction, Single Data*

- Características del modelo MISD:
- Cada unidad ejecuta una instrucción distinta
- Cada unidad procesa el mismo dato
- Aplicación muy limitada en la vida real



Las pocas computadoras de este tipo, que existen generalmente se utilizan para tolerancia a fallos, donde varios chequeos se hacen en paralelo y un resultado general se guarda en una variable en memoria.

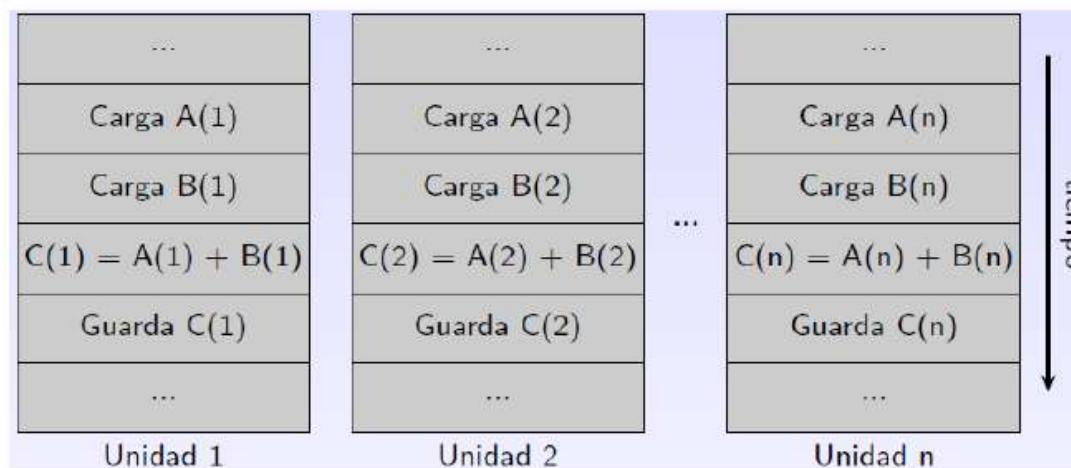
# SIMD:(Single Instruction Multiple Data)

Esta arquitectura representa la ejecución de una misma instrucción sobre un conjunto de datos. La misma es comúnmente vista en ciclos de programación que ejecutan una misma instrucción una y otra vez sobre datos de un arreglo o conjunto de datos. En la arquitectura SIMD estos datos son procesados por múltiples CPU que ejecutan la misma instrucción sobre una parte del conjunto o arreglo, cada uno, hasta llegar a procesar la totalidad de los mismos.

## ***SIMD: Single Instruction, Multiple Data***

### **Características del modelo SIMD:**

- Todas las unidades ejecutan la misma instrucción
- Cada unidad procesa un dato distinto
- Todas las unidades operan simultáneamente



Dos tipos de arquitecturas SIMD son:

**Procesadores vectoriales (vector processors).** Son procesadores cuyas instrucciones trabajan no en un dato a la vez (escalar), sino en arreglos de datos (vectores). Por ejemplo, un procesador tradicional puede calcular la suma de dos registros y guardar el resultado en uno de ellos. Un procesador vectorial tiene varias ALU y puede calcular la suma (una instrucción) de un arreglo de números en un ciclo de reloj si el tamaño del arreglo es menor o igual que su cantidad de ALU.

**Unidades de procesamiento gráfico (GPU, graphics processing units).** Son similares a los procesadores tradicionales con algunas diferencias. Las instrucciones están orientadas al trabajo con gráficas, como suavizar una curva o aplicar efectos a píxeles, puntos, líneas o triángulos. Dado que estos elementos son usualmente independientes, las GPU son arquitecturas altamente paralelas, compuestas de múltiples shaders. Aunque las GPU se suelen clasificar como arquitecturas SIMD, pueden ejecutar instrucciones distintas en distintos datos.

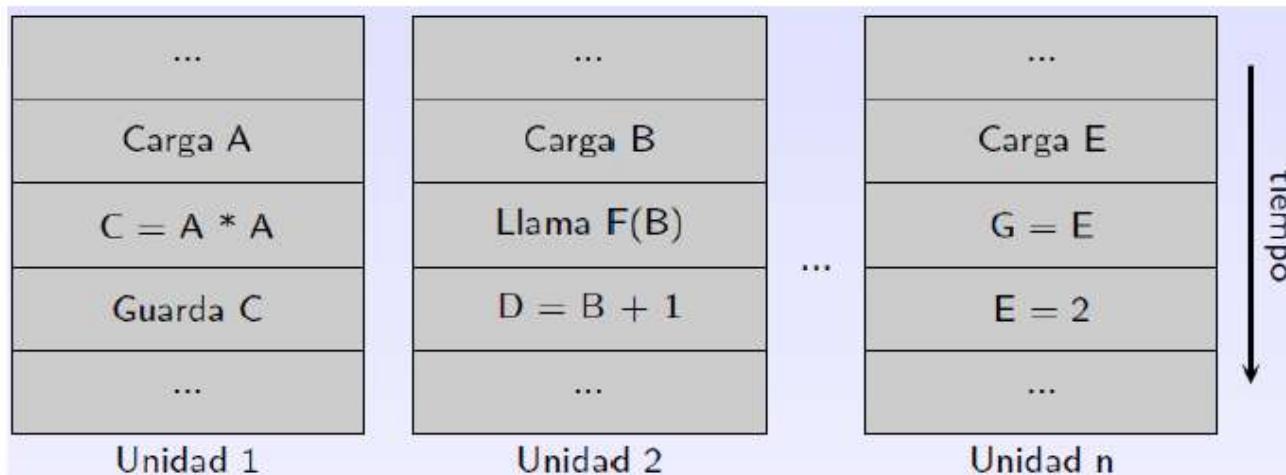
# MIMD: (Multiple Instruction Multiple Data)

Esta arquitectura representa a un conjunto de instrucciones que se ejecutan sobre un conjunto múltiple de datos. La misma es muy usada hoy en día para explotar el paralelismo ya sea con memoria distribuida y memoria compartida o híbridos como los clústers de computadoras. Muchos multiprocesadores modernos (como los de la tecnología Core i de Intel) entran en esta clasificación.<sup>[2]</sup>

## *MIMD: Multiple Instruction, Multiple Data*

### Características del modelo MIMD:

- Cada unidad ejecuta una instrucción distinta
- Cada unidad procesa un dato distinto
- Todas las unidades operan simultáneamente



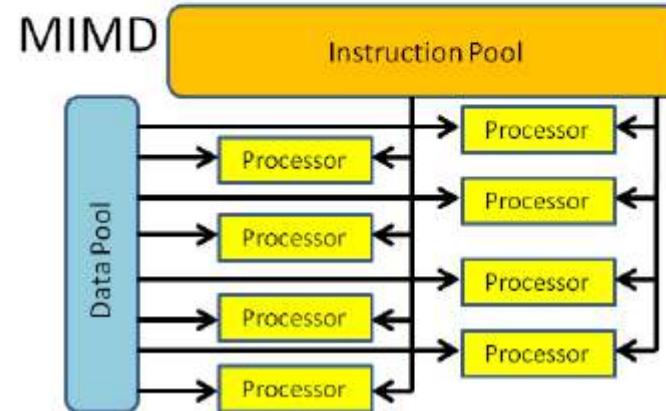
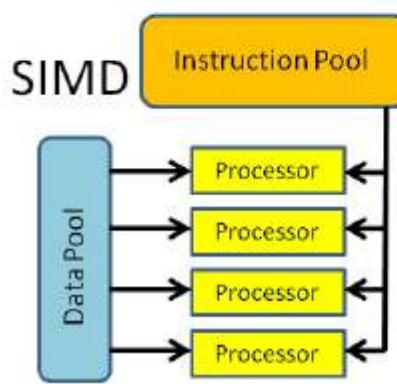
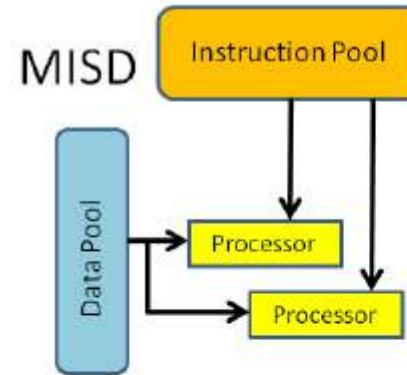
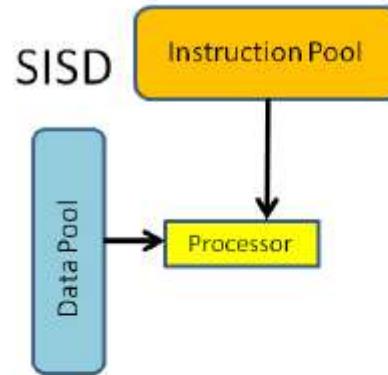
Dos o más núcleos pueden estar simultáneamente ejecutando instrucciones distintas y en datos distintos. Después del 2005 las arquitecturas SISD dejaron de ser las más comunes para dar paso a las arquitecturas MIMD.

Las arquitecturas de múltiples instrucciones y múltiples datos se pueden clasificar en sistemas de memoria compartida y sistemas de memoria distribuida.

#### Fuentes

- DUNCAN, Ralph. A survey of parallel computer architectures. Computer, 1990, vol. 23, no 2, p. 5-16.
- JOHNSON, Eric E. Completing an MIMD multiprocessor taxonomy. ACM SIGARCH Computer Architecture News, 1988, vol. 16, no 3, p. 44-47.

# TAXONOMÍA DE FLYNN DE LA ARQUITECTURA DE COMPUTADORAS

















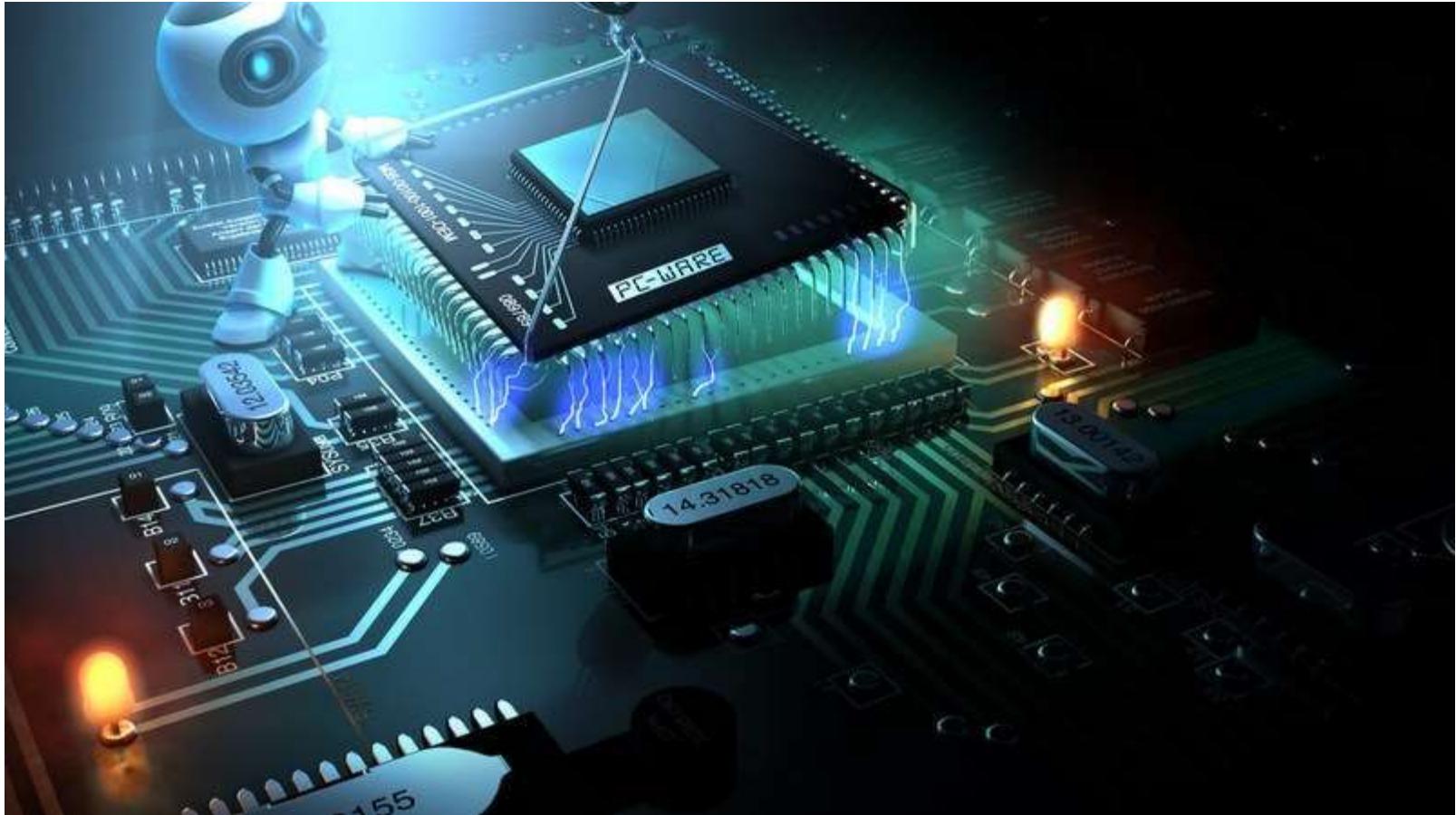


# **APENDICES**



# LENGUAJE ENSAMBLADOR

## (4°B-ISC)



Mtro. en Ing. Armando Álvarez Fdez.  
Ago-2024

# **EL MODELO DE PROGRAMACIÓN DEL CPU INTEL 8086**

## EJERCICIOS Y NOTAS:



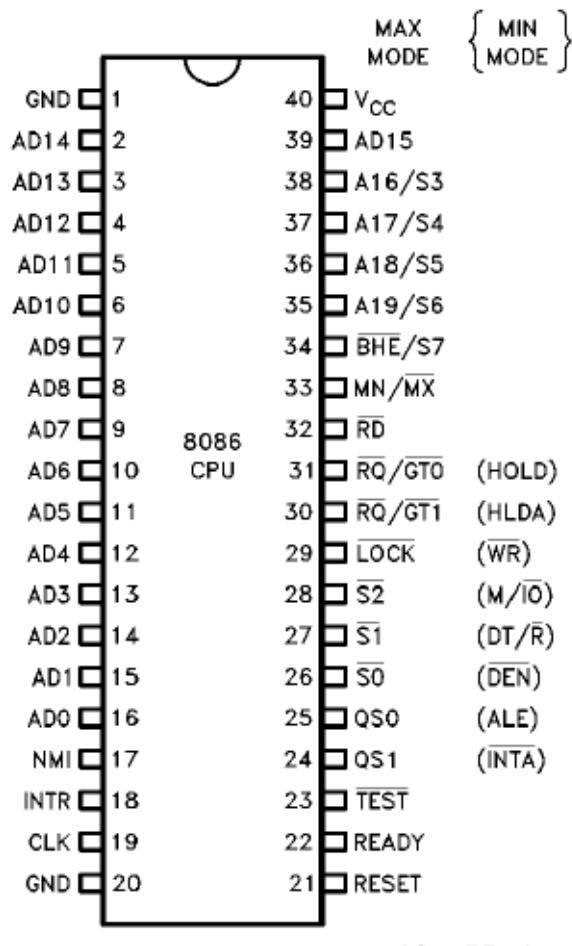
# 8086

## 16-BIT HMOS MICROPROCESSOR

### 8086/8086-2/8086-1

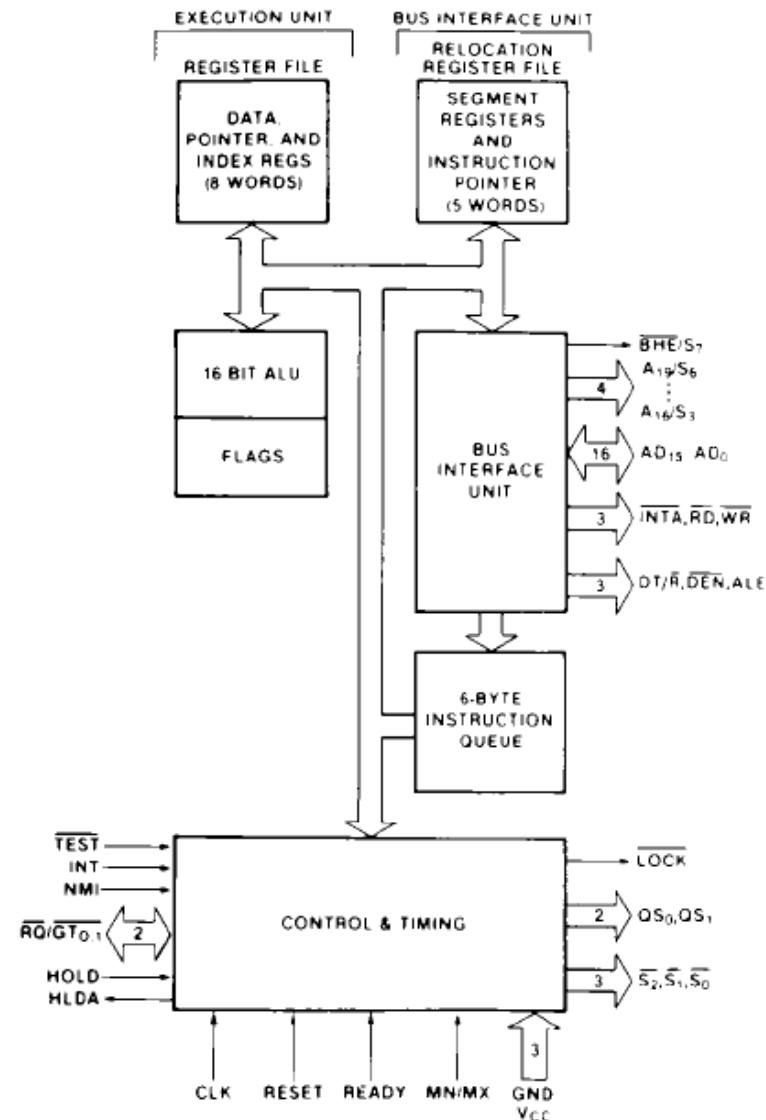
- Direct Addressing Capability 1 MByte of Memory
- Architecture Designed for Powerful Assembly Language and Efficient High Level Languages
- 14 Word, by 16-Bit Register Set with Symmetrical Operations
- 24 Operand Addressing Modes
- Bit, Byte, Word, and Block Operations
- Range of Clock Rates:  
5 MHz for 8086,  
8 MHz for 8086-2,  
10 MHz for 8086-1
- 8 and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal Including Multiply and Divide
- Available in 40-Lead Cerdip and Plastic Package

# Chip CPU Intel 8086 y su diagrama a bloques



**40 Lead**

**Figure 2. 8086 Pin Configuration**



**Figure 1. 8086 CPU Block Diagram**

# DESCRIPCION GENERAL DEL 8086 y 8088

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

## EXPLICACION =

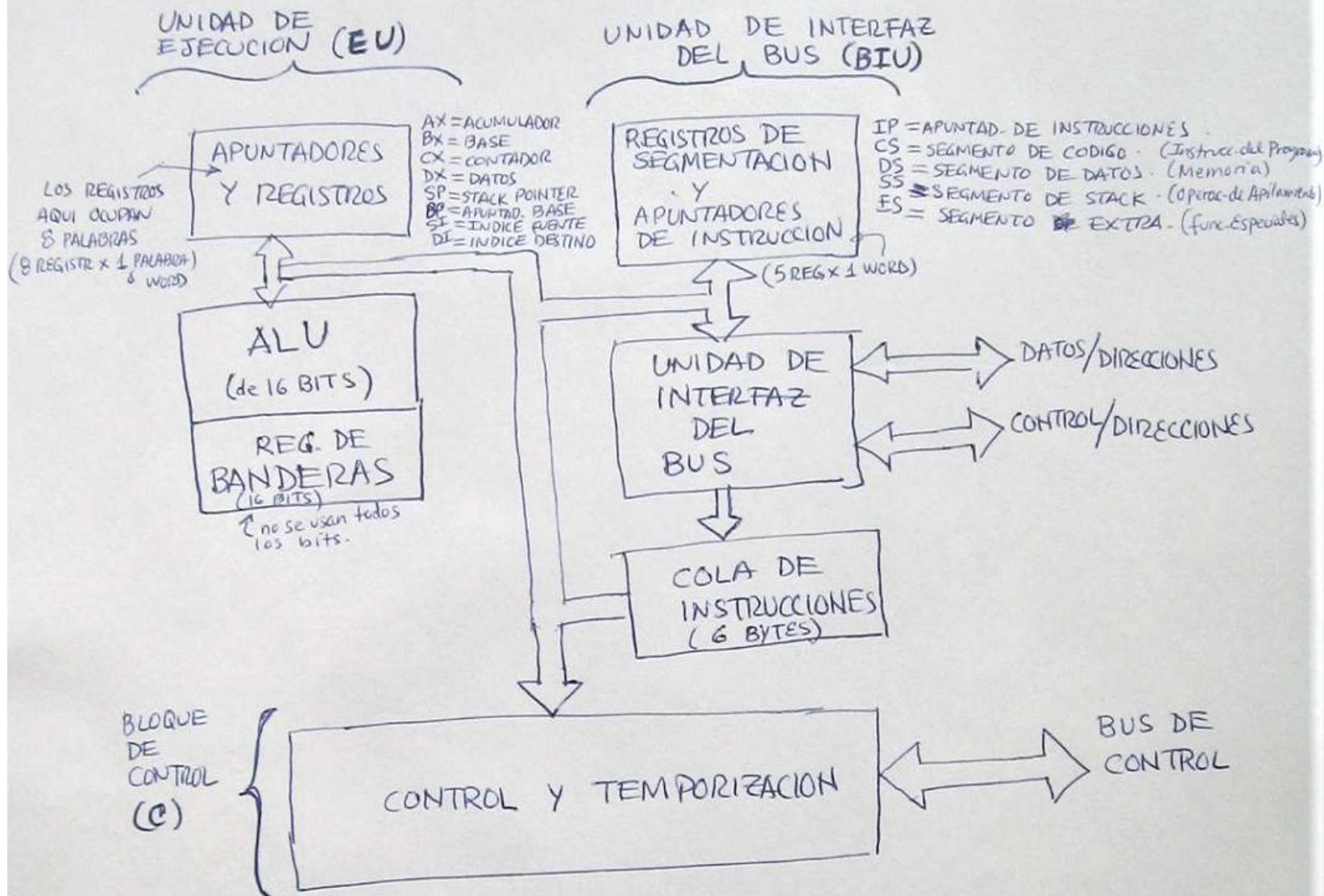
- Consta de 6 Partes.
- Se agrupan en 3 Bloques básicos

- EU (Execution Unit): Realiza las Operaciones Lógicas y Aritméticas y calcula Búsco-efectivas
- BIU (Bus Interface Unit): Manipula los buses de datos y direcciones y controlar cola de instrucciones.
- Control = "Señales de Control y Temporizadores del CPU" (Técnicos)

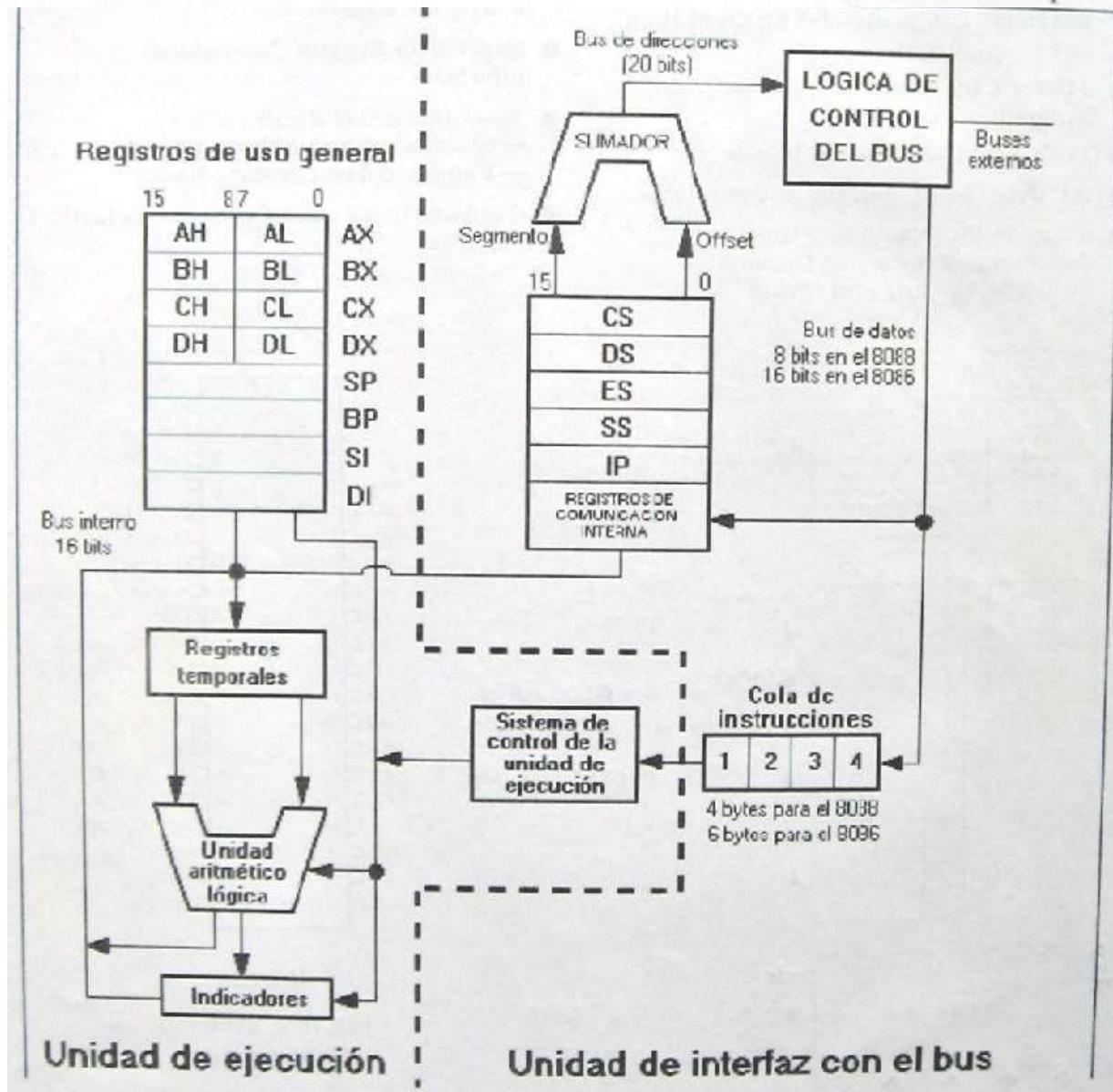
- El CPU tiene 2 modos Fundamentales de operación
    - \* Se ajusta el modo con la entrada PIN (33) llamado MN/MX
- Modo MÁXIMO = Modo de Hardware que maneja más señales por Ejemplo, para interactuar con un Coprocesador Matem. 8087 y Arbitraje del Bus.
  - Modo MÍNIMO = Es un modo Básico de Sistema Mínimo menos sofisticado en Hardware

# DESCRIPCION GENERAL DEL 8086 y 8088

## DESCRIPCION GRAL. DEL CPU 8086 (DIAGR. A BLOQUES)



Las unidades de ejecución son idénticas en ambos microprocesadores, pero las unidades de interfaz del bus son diferentes en varias cuestiones, como se desprende del siguiente diagrama en bloques:



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

## 2. El modelo de programación.

Se describe la estructura de los registros del microprocesador y se explica la forma en que se direcciona la memoria por medio de los registros de segmentos y de los desplazamientos de dirección.

Nombre 32 bits	AH	AX	AL	NOMBRE		
EAX		BH	BX	Acumulador		
EBX		CH	CX	Índice base		
ECX		DH	DX	Contador		
EDX			DL	Datos		
ESP			SP	Apuntador de pila		
EBP			BP	Apuntador de base		
EDI			DI	Índice destino		
ESI			SI	Índice fuente		
32 bits						
16 bits						
EIP EFLAGS	IP	Apuntador de Inst				
	FLAGS	Banderas				
				Código		
				Datos		
				Extra		
				Pila		
				No se le dan nombres especiales		
				GS		

## MODELO DE PROGRAMACION DEL 80X86 (SUS REGISTROS)

NOTA: Las áreas sombreadas EXISTEN SOLAMENTE EN LOS CPUS INTEL 80386, EN ADELANTE.

OBSERVAR QUE EL MODELO DE PROGRAMACION TIENE REGISTROS DE 8, 16 Y 32 BITS.

\* (32-BITS) Sólo existe por reduplicación  
REGISTROS DE 8 BITS:

{AH, AL, BH, BL, CH, CL, DH, DL}

REGISTROS DE 16 BITS:

{AX, BX, CX, DX, SP, BP, DS, SS,  
IP, FLAGS, CS, DS, ES, SS, FS, GS}

REGISTR. EXTENDIDOS DE 32 BITS:

{EAX, EBX, ECX, EDX, ESP,  
EBP, EDI, ESZ, EIP, EFLAGS}



EAX	AH	AX	AL	← ACUMULADOR
EBX	BH	BX	BL	← BASE
ECX	CH	CX	CL	← CONTADOR
EDX	DH	DX	DL	← DATOS
ESP	SP			← STACK POINTER (AVANT DE PILA)
EBP	BP			← APUNTADOR BASE
EDI	DI			← INDICE DESTINO
ESI	SI			← INDICE FUENTE

EIP	IP	← APUNTADOR DE INSTRUCCIONES
EFLAGS	FLAGS	← BANDERAS

Ej: es un registro de 32 bits

LOS REGISTROS FS Y GS NO TIENEN UN NOMBRE ESPECIFICO

CS	← SEGMENTO DE CODIGO
DS	← SEGMENTO DE DATOS
ES	← SEGMENTO EXTRA
SS	← SEGMENTO DE STACK (PILA = APUNTIADOR)
FS	
GS	

EXISTEN REGISTROS DE:

- DE PROPÓSITO GRAL → MULTIPROPÓSITO: {EAX, EBX, ECX, EDX, EBP, EDI, ESI} → Utilizadas para muchos propósitos
- DE PROPÓSITO ESPECIAL → Tienen una finalidad específica o más especializada en algo. {ESP, EIP, EFLAGS, CS, DS, ES, SS, FS, GS}

REGISTROS DE SEGMENTACIÓN

# NOMBRES E INICIALES DE LOS REGISTROS DEL 8086 EN INGLES

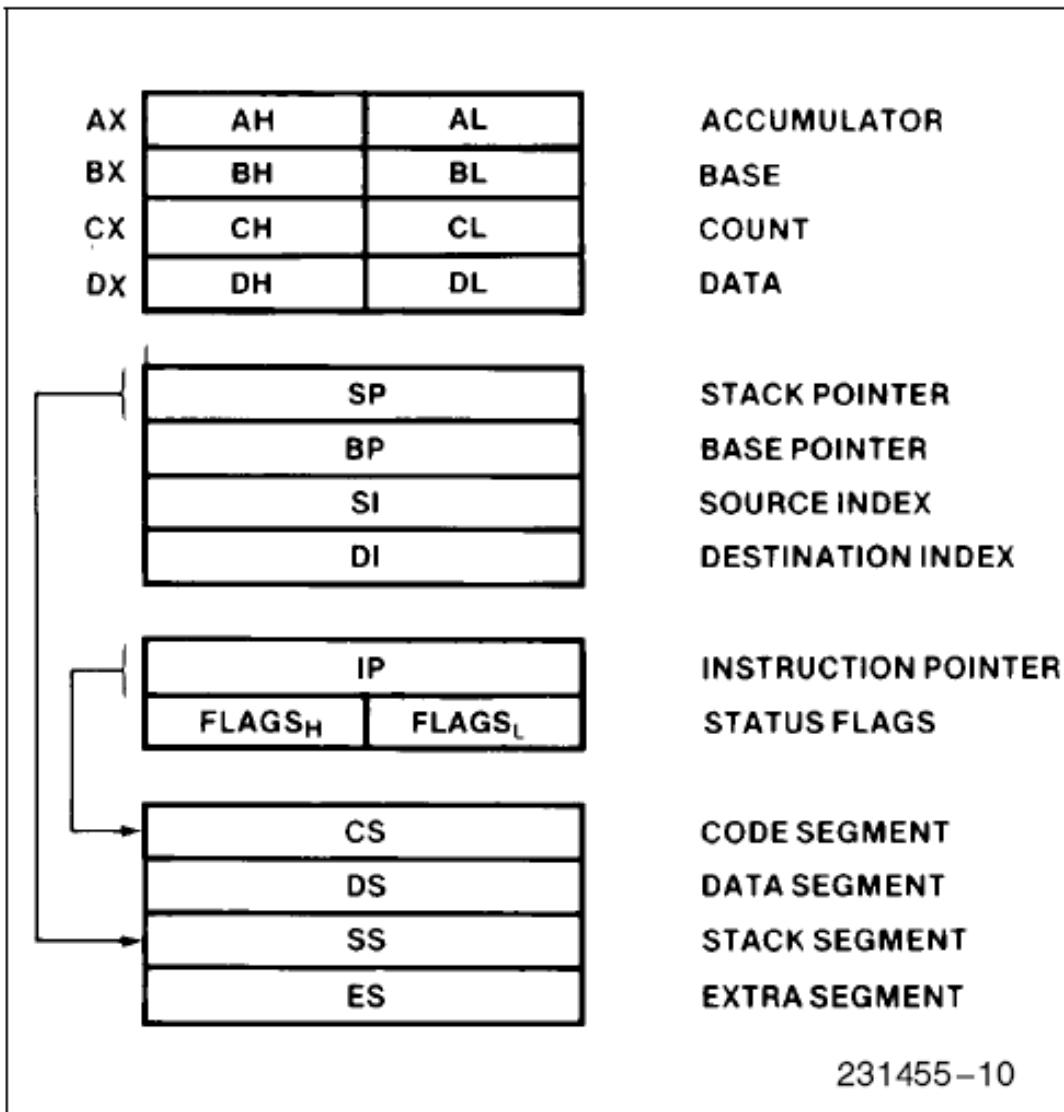


Figure . 8086 Register Model

### *3. Registros del microprocesador*

#### **Registros de propósito general.**

Los *registros de propósito general* se utilizan en la forma en que se desee el programador. Cada registro para uso general se puede direccionar como un registro de 32 bits (EAX, EBX, ECX y EDX), como un registro de 16 bits (AX, BX, CX y DX) o como uno de 8 bits (AH, AL, BH, BL, CH, CL, DH y DL). Se debe tener en cuenta que solo el 80386 en adelante contienen el grupo de registros de 32 bits. Las funciones principales de los registros de propósito general incluyen:

**AX (Acumulador):** Es el acumulador primario, posee dos características diferentes de los otros acumuladores. Todas las operaciones de I/O deben pasar por la parte baja (AL) de este acumulador, las instrucciones que utilizan este registro gastan menos espacio y se ejecutan más rápido.

**BX (Base):** Es un acumulador de propósito general que puede ser utilizado para calcular direcciones, a menudo conserva la dirección base (desplazamiento) de los datos que hay en memoria. Si la memoria es accesada con este registro los programas serán más cortos y correrán más rápidos.

**CX (Contador):** Puede ser utilizado para el conteo de ciertas instrucciones para corrimientos (CL) y rotaciones del numero de bytes (CX) para las operaciones repetidas de cadenas y un contador (CX o ECX) para la instrucción LOOP.

**DX (Datos):** Las instrucciones de I/O lo utilizan para definir la dirección del periférico, algunas multiplicaciones y divisiones también lo necesitan.

### **Registros apuntadores e índices.**

**SP (Apuntador de pila):** Se utiliza para direccionar datos en una pila de memoria LIFO, esto ocurre mas a menudo cuando se ejecutan las instrucciones PUSH y POP, cuando se llama CALL o cuando se regresa RET de una subrutina.

**BP (Apuntador de base):** Es un apuntador que se utiliza para referenciar parámetros, especialmente cuando se programa en un ambiente de múltiples lenguajes.

**SI (Índice de fuente):** Se emplea para direccionar datos fuente en forma indirecta para utilizarlos con las instrucciones de cadenas o arreglos.

**DI (Índice destino):** Se suele emplear para direccionar datos destino en forma indirecta para utilizarlos con las instrucciones de cadenas o arreglos.

**IP (Apuntador de instrucciones):** Se utiliza siempre para direccionar a la siguiente instrucción que va a ejecutar el microprocesador. Esta relacionado con el registro CS que es el segmento de código. Para formar la localidad real de la siguiente instrucción se suma el contenido de IP con CS por 10H. Corresponde al contador de programas (Program Counter o PC) de otros microprocesadores.

## **Registros con segmentos.**

Unos registros adicionales, a los que se da el nombre de registros de segmentos, generan direcciones en la memoria junto con otros registros en el microprocesador. A continuación aparece una lista de cada registro de segmento junto con su función en el sistema:

***CS (Código):*** El segmento de código es una sección en la memoria que tiene los programas y procedimientos utilizados por los programas. El registro de segmento de código define la dirección inicial de la sección de memoria que tiene el código (*Es un registro de 16 bits que define el segmento de las instrucciones*).

***DS (Datos):*** El segmento de datos es una sección en la memoria que contiene la mayor parte de los datos utilizados por un programa, (estos datos pueden ser variables, vectores, matrices, etc).

**ES (Extra o adicional):** El segmento extra o adicional de datos lo utilizan algunas instrucciones para cadenas.

**SS (Pila):** El segmento de pila define la superficie de la memoria utilizada para la pila. La ubicación del punto inicial de entrada a la pila, se determina por el registro apuntador de la pila. El registro BP también direcciona los datos que hay dentro del segmento de pila.

**FS y GS:** Estos registros de segmento adicionales están disponibles en los microprocesadores 80386 en adelante a fin de contar con dos segmentos adicionales de memoria para acceso con los programas.



## EJERCICIOS Y NOTAS:

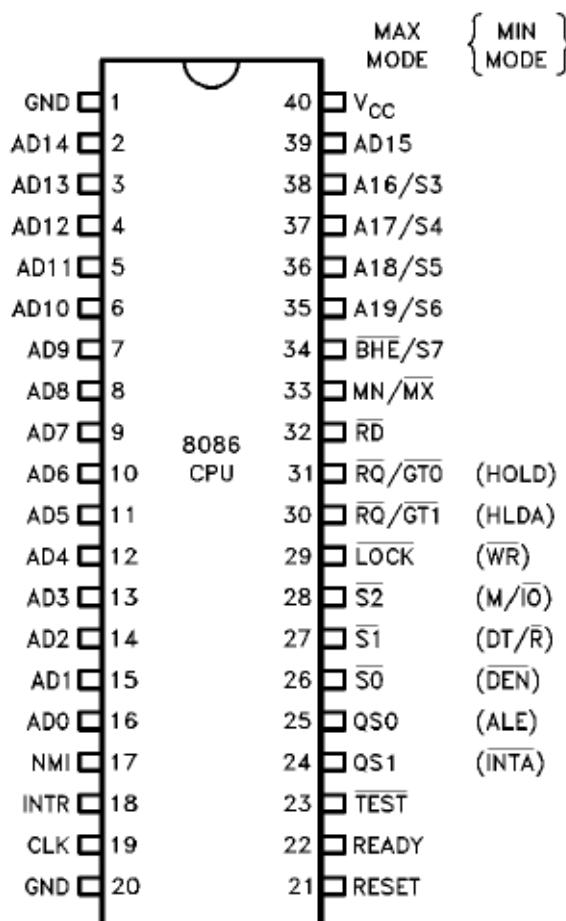
# **APENDICES**



**8086**  
**16-BIT HMOS MICROPROCESSOR**  
**8086/8086-2/8086-1**

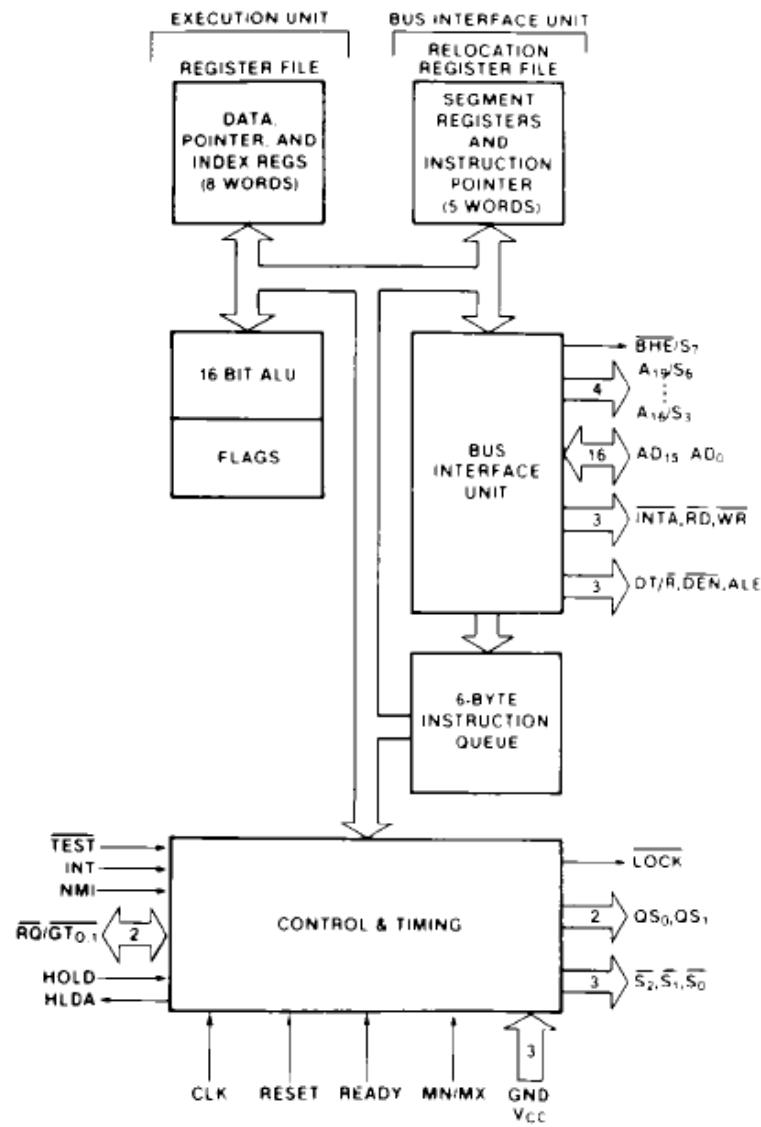
- Direct Addressing Capability 1 MByte of Memory
- Architecture Designed for Powerful Assembly Language and Efficient High Level Languages
- 14 Word, by 16-Bit Register Set with Symmetrical Operations
- 24 Operand Addressing Modes
- Bit, Byte, Word, and Block Operations
- 8 and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal Including Multiply and Divide
- Range of Clock Rates:  
5 MHz for 8086,  
8 MHz for 8086-2,  
10 MHz for 8086-1
- MULTIBUS System Compatible Interface
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range
- Available in 40-Lead Cerdip and Plastic Package  
(See Packaging Spec. Order #231369)

# Chip CPU Intel 8086 y su diagrama a bloques



40 Lead

**Figure 2. 8086 Pin Configuration**



**Figure 1. 8086 CPU Block Diagram**

231455-1

# REGISTROS DE LA UNIDAD DE EJECUCIÓN DE INSTRUCCIONES (EU).

Tienen 16 bits cada uno y son ocho:

1. **AX** = Registro acumulador, dividido en **AH** y **AL** (8 bits cada uno). Usándolo se produce (en general) una instrucción que ocupa un byte menos que si se utilizaran otros registros de uso general. Su parte más baja, **AL**, también tiene esta propiedad. El último registro mencionado es el equivalente al acumulador de los procesadores anteriores (8080 y 8085). Además hay instrucciones como DAA; DAS; AAA; AAS; AAM; AAD; LAHF; SAHF; CBW; IN y OUT que trabajan con AX o con uno de sus dos bytes (AH o AL). También se utiliza este registro (junto con DX a veces) en multiplicaciones y divisiones.
  
2. **BX** = Registro base, dividido en **BH** y **BL**. Es el registro base de propósito similar (se usa para direccionamiento indirecto) y es una versión más potente del par de registros HL de los procesadores anteriores.
  
3. **CX** = Registro contador, dividido en **CH** y **CL**. Se utiliza como contador en bucles (instrucción LOOP), en operaciones con cadenas (usando el prefijo REP) y en desplazamientos y rotaciones (usando el registro CL en los dos últimos casos).

# REGISTROS DE LA UNIDAD DE EJECUCIÓN DE INSTRUCCIONES (EU).

4. **DX** = Registro de datos, dividido en **DH** y **DL**. Se utiliza junto con el registro AX en multiplicaciones y divisiones, en la instrucción CWD y en IN y OUT para direccionamiento indirecto de puertos (el registro DX indica el número de puerto de entrada/salida).

5. **SP** = Puntero de pila (no se puede subdividir).

Aunque es un registro de uso general, debe utilizarse sólo como puntero de pila, la cual sirve para almacenar las direcciones de retorno de subrutinas y los datos temporarios (mediante las instrucciones PUSH y POP). Al introducir (push) un valor en la pila a este registro se le resta dos, mientras que al extraer (pop) un valor de la pila este a registro se le suma dos.

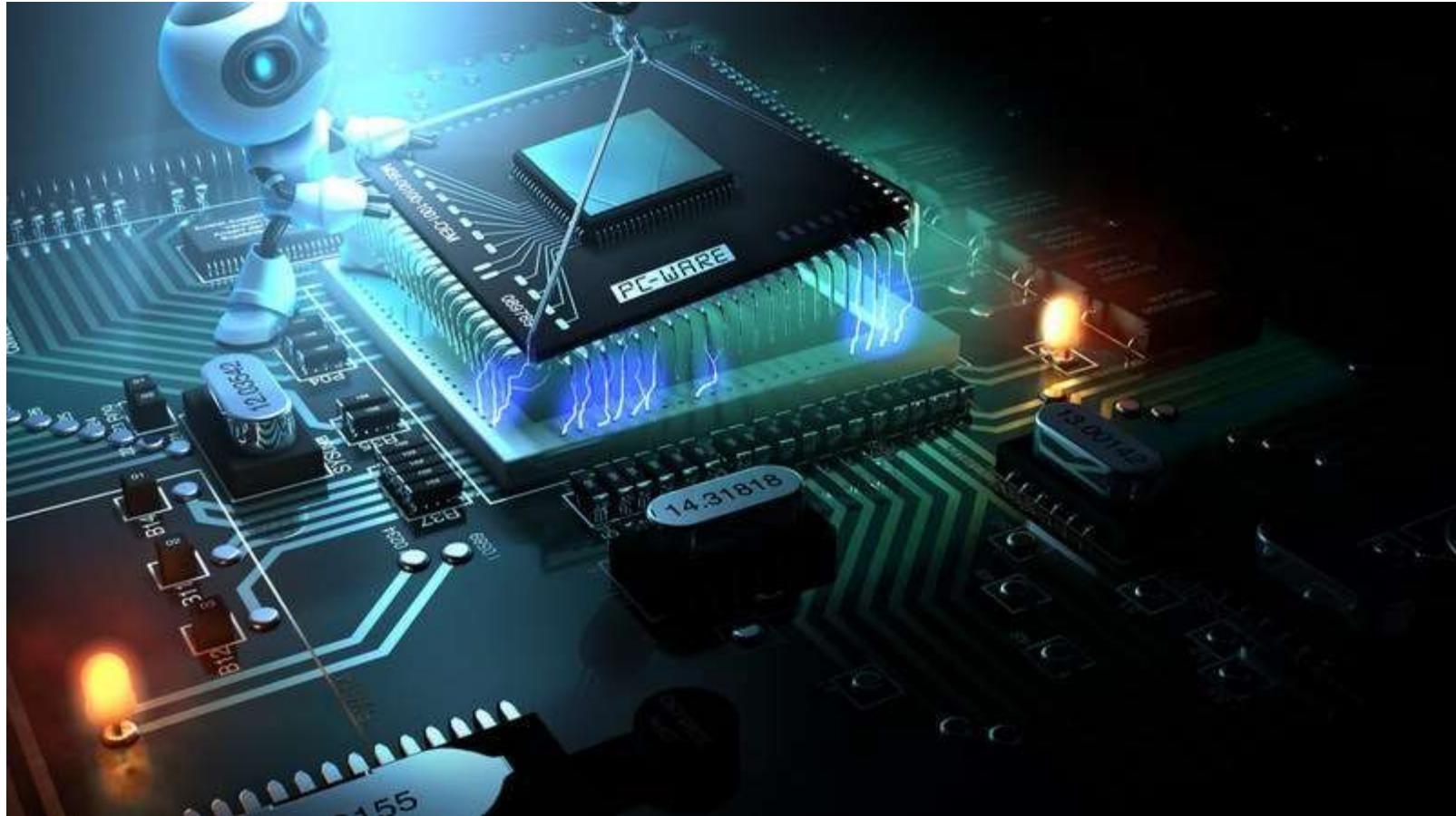
6. **BP** = Puntero base (no se puede subdividir). Generalmente se utiliza para realizar direccionamiento indirecto dentro de la pila.

## EJERCICIOS Y NOTAS:



# LENGUAJE ENSAMBLADOR

## (7° A-ICI)



Mtro. en Ing. Armando Álvarez Fdez.  
Ago-2024

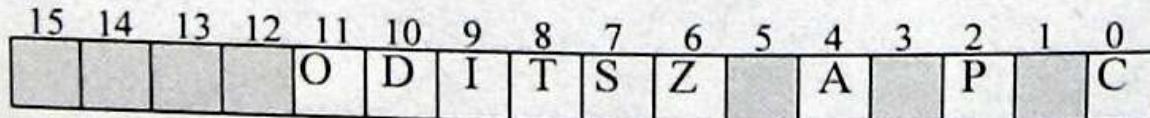
# **REGISTRO DE BANDERAS DEL 8086**

# REGISTRO DE BANDERAS DEL 8086

(También conocido como Registro de Estatus del CPU 8086)

## 4. Banderas de Registro Status (El registro de banderas).

Las banderas indican la condición del microprocesador a la vez que controlan su funcionamiento.



# REGISTRO DE BANDERAS DEL 8086

## Registro de flags



C : acarreo en la suma y arrastre en la resta

P : paridad del dato (0, impar y 1, par)

A : acarreo auxiliar. Indica el acarreo o arrastre entre los bits 3 y 4

Z : indicación de resultado igual a cero

S : indicador de signo del resultado. 0, positivo y 1, negativo

T : trampa. Habilita la característica de depuración del procesador

I : habilitación de interrupciones de hardware

D : selección de incremento o decremento en los índices

O : sobreflujo.

# EXPLICACIÓN DE LAS BANDERAS

**C (Acarreo):** Indica un acarreo después de una suma o un préstamo después de una resta. La bandera de acarreo también indica condiciones de error en ciertos programas y procedimientos. También se utiliza en algunas instrucciones de rotación y desplazamiento.

**P (Paridad):** Es un cero para la paridad impar y un uno para la paridad par. La paridad es un conteo de “unos” expresado en un número par o impar. Por ejemplo, si un número contiene 3 bits con uno binario, tiene paridad impar. Si un numero contiene cero bits de uno se considera que paridad par.

**A (Acarreo Auxiliar):** Indica un acarreo después de una suma o un préstamo después de una resta entre las posiciones de los bits 3 y 4 en el resultado. Este indicador muy especializado se prueba con las instrucciones DAA y DAS para ajustar el valor de AL después de una suma o resta BCD. El microprocesador, no utiliza en otra forma el bit de bandera A.

**Z (Zero):** Indica que el resultado de una operación aritmética o lógica es cero. Si Z=1, el resultado es cero, si Z=0 entonces el resultado no es cero.

**S (Signo):** Indica el signo aritmético del resultado después de una suma o resta. Si S=1 la bandera de signo se activa y el resultado es negativo. Si S=0 la bandera de signo se desactiva y el resultado es positivo. Se debe tener en cuenta que el valor de la posición del bit más significativo se coloca en el bit de signo para cualquier instrucción que afecte las banderas.

**T (Trampa):** Cuando se activa la bandera de trampa se habilita la característica de depuración del microprocesador. Mas adelante aparecen mayores detalles de esta característica.

**I (Interrupción):** Controla el funcionamiento de la terminal de la entrada de INTR (*Interrupción externa*). Si I=1 se habilita la interrupción y si I=0 se deshabilita la entrada INTR.

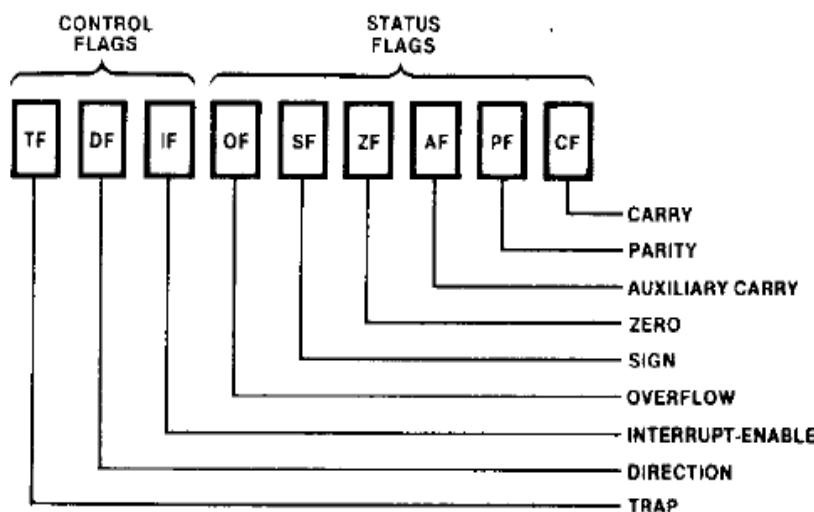
**D (Direccion):** Controla la selección de incremento o decremento de los registros DI o SI durante las instrucciones de cadenas o arreglos. Se utiliza para indicar si las operaciones con string se ejecutarán en forma ascendente o descendente.

**O (Sobreflujo):** Es una condición que ocurre cuando se suman o restan números con signo. Un sobreflujo indica que el resultado ha excedido de la capacidad de la máquina. Por ejemplo, si se suma un 7FH (+127) a 01H (+1) el resultado es 80H (-128). Este resultado representa una situación de sobreflujo señalado por la bandera para la suma con signo. Para operaciones sin signo no se toma en cuenta esta bandera.

# **NOTAS ADICIONALES SOBRE LAS BANDERAS DEL 8086**

## **Flags**

The 8086 and 8088 have six 1-bit status flags (figure 2-9) that the EU posts to reflect certain properties of the result of an arithmetic or logic



**Figure 2-9. Flags**

operation. A group of instructions is available that allows a program to alter its execution depending on the state of these flags, that is, on the result of a prior operation. Different instructions affect the status flags differently; in general, however, the flags reflect the following conditions:

1. If AF (the auxiliary carry flag) is set, there has been a carry out of the low nibble into the high nibble or a borrow from the high nibble into the low nibble of an 8-bit quantity (low-order byte of a 16-bit quantity). This flag is used by decimal arithmetic instructions.
2. If CF (the carry flag) is set, there has been a carry out of, or a borrow into, the high-order bit of the result (8- or 16-bit). The flag is used by instructions that add and subtract multibyte numbers. Rotate instructions can also isolate a bit in memory or a register by placing it in the carry flag.

## **NOTAS ADICIONALES SOBRE LAS BANDERAS DEL 8086**

3. If OF (the overflow flag) is set, an arithmetic overflow has occurred; that is, a significant digit has been lost because the size of the result exceeded the capacity of its destination location. An Interrupt On Overflow instruction is available that will generate an interrupt in this situation.
4. If SF (the sign flag) is set, the high-order bit of the result is a 1. Since negative binary numbers are represented in the 8086 and 8088 in standard two's complement notation, SF indicates the sign of the result (0 = positive, 1 = negative).
5. If PF (the parity flag) is set, the result has even parity, an even number of 1-bits. This flag can be used to check for data transmission errors.
6. If ZF (the zero flag) is set, the result of the operation is 0.

Three additional control flags (figure 2-9) can be set and cleared by programs to alter processor operations:

1. Setting DF (the direction flag) causes string instructions to auto-decrement; that is, to process strings from high addresses to low addresses, or from "right to left." Clearing DF causes string instructions to auto-increment, or to process strings from "left to right."
2. Setting IF (the interrupt-enable flag) allows the CPU to recognize external (maskable) interrupt requests. Clearing IF disables these interrupts. IF has no affect on either non-maskable external or internally generated interrupts.
3. Setting TF (the trap flag) puts the processor into single-step mode for debugging. In this mode, the CPU automatically generates an internal interrupt after each instruction, allowing a program to be inspected as it executes instruction by instruction. Section 2.10 contains an example showing the use of TF in a single-step and breakpoint routine.

# VISTA GENERAL DEL SET DE INSTRUCCIONES DEL 8086

## 6. Juego de instrucciones.

Las categorías de instrucciones descritas en esta sección incluyen: Transferencia de datos, aritméticas, manipulación de bits, cadenas o arreglos, transferencia de programa, etc.

**Transferencia de datos:** Incluye instrucciones para transferencia de datos que transfieren bytes, palabras o dobles palabras de datos entre la memoria y los registros así como entre el acumulador y los puertos de E/S.

Código Operación	Función
IN	Mete datos al acumulador desde un dispositivo de E/S.
LAHF	Carga banderas en AH.
LEA	Carga la dirección efectiva.
LDS	Carga DS y registro de 16 bits con los datos de memoria de 32 bits.
LES	Carga ES y registro de 16 bits con los datos de memoria de 32 bits.
MOV	Carga byte, palabra o doble palabra.
OUT	Saca datos del acumulador a un E/S.
POP	Recupera una palabra de la pila.
POPF	Recupera los indicadores de la pila.
PUSH	Salva las palabras en la pila.
PUSHF	Salva banderas en la pila.
SAHF	Carga AH en las banderas.
XCHG	Intercambia bytes, palabras o dobles palabras.
XLAT	Emplea AL para entrar a una tabla de conversión.

**Aritmética:** Instrucciones para sumar, restar, multiplicar y dividir datos como bytes, palabras o dobles palabras. El sistema suma y resta con el empleo de datos con signo o sin signo y datos BCD o de ASCII. Multiplica y divide números ASCII con signo o sin signo.

Código Operación	Función
AAA	Ajuste ASCII para la suma.
AAD	Ajuste ASCII para la división.
AAM	Ajuste ASCII para la multiplicación.
AAS	Ajuste ASCII para la resta.
ADD	Suma datos entre registros o la memoria y otro registro.
ADC	Suma datos con la bandera de acarreo.
CBW	Convierte byte a palabra.
CMP	Compara los datos.
CWD	Convierte palabra a doble palabra.
DAA	Ajuste decimal de AL después de una suma de BCD:
DAS	Ajuste decimal de AL después de una resta de BCD:
DEC	Decremento.
DIV	División sin signo.
IDIV	División con signo.
IMUL	Multiplicación con signo.
INC	Incrementa.
MUL	Multiplicación sin signo.
NEG	Cambia el signo (complemento a dos).
SBB	Resta con acarreo.
SUB	Resta datos entre los registros y la memoria u otro registro.

**Manipulación de bits:** Se utilizan para controlar los datos hasta el nivel bits. Estas instrucciones incluyen operaciones lógicas, corrimientos y rotaciones.

Código Operación	Función
AND	Y lógica.
NOT	Invertir (complemento a uno).
OR	O lógica. <i>inclusivo</i>
SAR	Corrimiento aritmético a la derecha.
SHL/SAL	Corrimiento a la izquierda.
SHR	Corrimiento lógico a la derecha.
RCL	Rotación a la izquierda con acarreo.
ROL	Rotación a la izquierda.
RCR	Rotación a la derecha con acarreo.
ROR	Rotación a la derecha.
TEST	Operación con el AND lógico, pero solo afectando banderas.
XOR	O exclusivo.

**Instrucciones para cadenas:** Se emplean para manipular cadenas de datos en la memoria. Cada cadena consta, ya sea de bytes o de palabras y tiene hasta 64K bytes de longitud.

Código Operación	Función
CMPS	Comparación entre memoria y memoria.
LODS	Cargar el acumulador.
MOVS	Mover de memoria a memoria.
SCAS	Comparación entre la memoria y el acumulador.
STOS	Almacenar en el acumulador.

**Transferencia de programa:** Incluyen brincos, llamadas (CALL) y para retorno.

*Saltos basados en datos sin signo.*

JE / JZ	Jump if equal / jump if zero
JNE / JNZ	Jump if non equal / jump if non zero
JA / JNBE	Jump if above / jump if no below or equal
JAE / JNB	Jump if above or equal / jump if not below
JB / JNAE	Jump if below / jump if not above or equal
JBE / JNA	Jump if below or equal / jump if not above

### *Saltos basados en datos con signo*

<b>JE / JZ</b>	Jump if equal / jump if zero
<b>JNE / JNZ</b>	Jump if non equal / jump if non zero
<b>JG / JNLE</b>	Jump if greater / jump if non less or equal
<b>JGE / JNL</b>	Jump if greater or equal / jump if non less
<b>JL / JNGE</b>	Jump if less / jump if not greater or equal
<b>JLE / JNG</b>	Jump if less or equal / jump if not greater

### *Pruebas aritméticas especiales.*

<b>JS</b>	Jump if sign
<b>JNS</b>	Jump if not sign
<b>JC</b>	Jump if carry
<b>JNC</b>	Jump if not carry
<b>JO</b>	Jump if overflow
<b>JNO</b>	Jump if not overflow
<b>JP / JPE</b>	Jump if parity / jump if parity even
<b>JNP / JPO</b>	Jump if not parity / jump if parity odd

## *5. Modos de direccionamiento.*

# **INTRODUCCION A LOS**

## ***Modos de direccionamiento***

### **(Primera aproximación de clasificación)**

Modos de direccionamiento. El microprocesador 8086 tiene 25 modos de direccionamiento. El modo de direccionamiento indica la forma en que ha de calcularse la dirección efectiva de los operandos de la instrucción. Los modos de direccionamiento son complicados como para explicarse en detalle cada uno de ellos sin embargo pueden ser clasificados como sigue:

- Directo.
- Segmentado.
- Base.
- Indexado.
- Con offset.

Estos modos de direccionamiento pueden ser combinados de tal forma que conforman así los 25 modos de direccionamiento.

El modo de direccionamiento directo hace referencia a la localidad de memoria en forma inmediata.

El modo de direccionamiento segmentado hace referencia al segmento involucrado para calcular la dirección efectiva.

El modo de direccionamiento de base utiliza el registro base BX para el cálculo de la dirección del operando.

El modo indexado involucra el registro de índice correspondiente en el cálculo de la dirección efectiva del operando.

El offset es un valor de desplazamiento que ha de tener la dirección calculada con los índices anteriores.

Un ejemplo de la forma más general en que se usa el modo de direccionamiento es la siguiente:

```
INC 6[BX][DI]
```

En este modo de direccionamiento la dirección efectiva del operando, en este caso del dato que ha de ser incrementado, es  $(16*DS + BX + DI + 6)$ . Esta instrucción utiliza el modo de direccionamiento segmentado, base, indexado con offset. Cabe hacer notar que no todas las instrucciones admiten todos los modos de direccionamiento, sin embargo, es preferible dejar que con la práctica se aprenda cuales los permiten y cuales no ya que sería un trabajo interminable el tratar de aprenderse todas las instrucciones con todos los modos de direccionamientos permitidos.

# **FIN DE PRESENTACION**



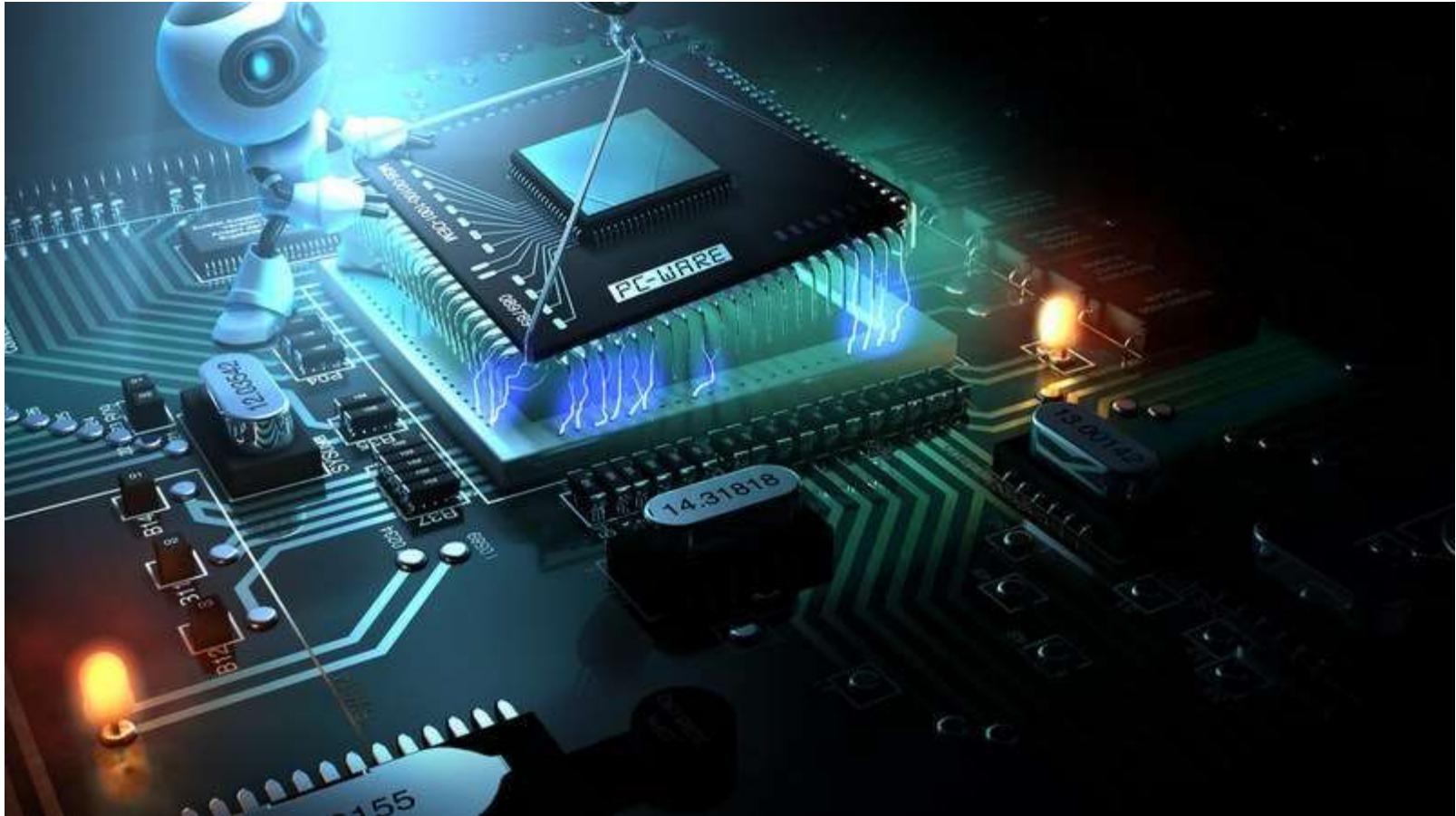
# **APENDICES**

## EJERCICIOS Y NOTAS:



# ENSAMBLADOR

## (7°-ICI)



Mtro. en Ing. Armando Álvarez Fdez.  
Sep-2024

# **MODOS DE DIRECCIONAMIENTO**

**(Continuación)**

## Modos de direccionamiento

De Los procesadores 8086 y 8088

- **Implícito.** El dato está implícito en la propia instrucción. Ej. `STC`, `STD` y `STI`, (Set Carry, Set Direction y Set Interrupts) encienden el flag correspondiente indicado en la propia instrucción. `CBW` (Convert Byte to Word) extiende el bit del signo del registro AL a AX. Ni el AL ni el AX son especificados, puesto que la instrucción `CBW` implicitamente trabaja sobre ellos.
- **Inmediato.** El dato a operar está inmediatamente después del opcode de la instrucción. Ej, `MOV AX, 5`
- **Registro.** El dato está en un segundo registro. Ej. `MOV AX, BX`. Aquí, el dato está en el registro BX
- **Directo.** La dirección del dato está en el campo de la dirección del opcode. Ej. `MOV AX, [100h]`. Aquí se mueve (copia) el contenido de las direcciones 100h y 101h al registro AX. En este caso se mueven dos bytes puesto que AX es de 16 bits. Si fuera `MOV BL, [100h]` se movería solo un byte pues BL es un registro de 8 bits

- **Indirecto.** El dato es especificado mediante una combinación de registros índice y base, y puede haber un desplazamiento
- **Base.** Un registro base (BX o BP) tienen la dirección de donde se tomará el dato. Ej. `MOV AX, [BX]`
- **Índice.** Un registro índice (SI o DI) tienen la dirección de donde se tomará el dato. Ej. `MOV AX, [SI]`
- **Base + Desplazamiento.** El dato se tomará de la dirección apuntada por la suma de un registro base más un desplazamiento. Ej. `MOV AX, [BP + 7]`
- **Índice + Desplazamiento.** El dato se tomará de la dirección apuntada por la suma de un registro índice más un desplazamiento. Ej. `MOV AX, [DI + 7]`
- **Base + Índice.** El dato se tomará de la dirección apuntada por la suma de un registro base más un registro índice. Ej. `MOV AX, [BX + SI]`
- **Base + Índice + Desplazamiento.** El dato se tomará de la dirección apuntada por la suma de un registro base, más un registro índice, más un desplazamiento. Ej. `MOV AX, [BX + SI + 9]`

# NOMBRES E INICIALES DE LOS REGISTROS DEL 8086 EN INGLES

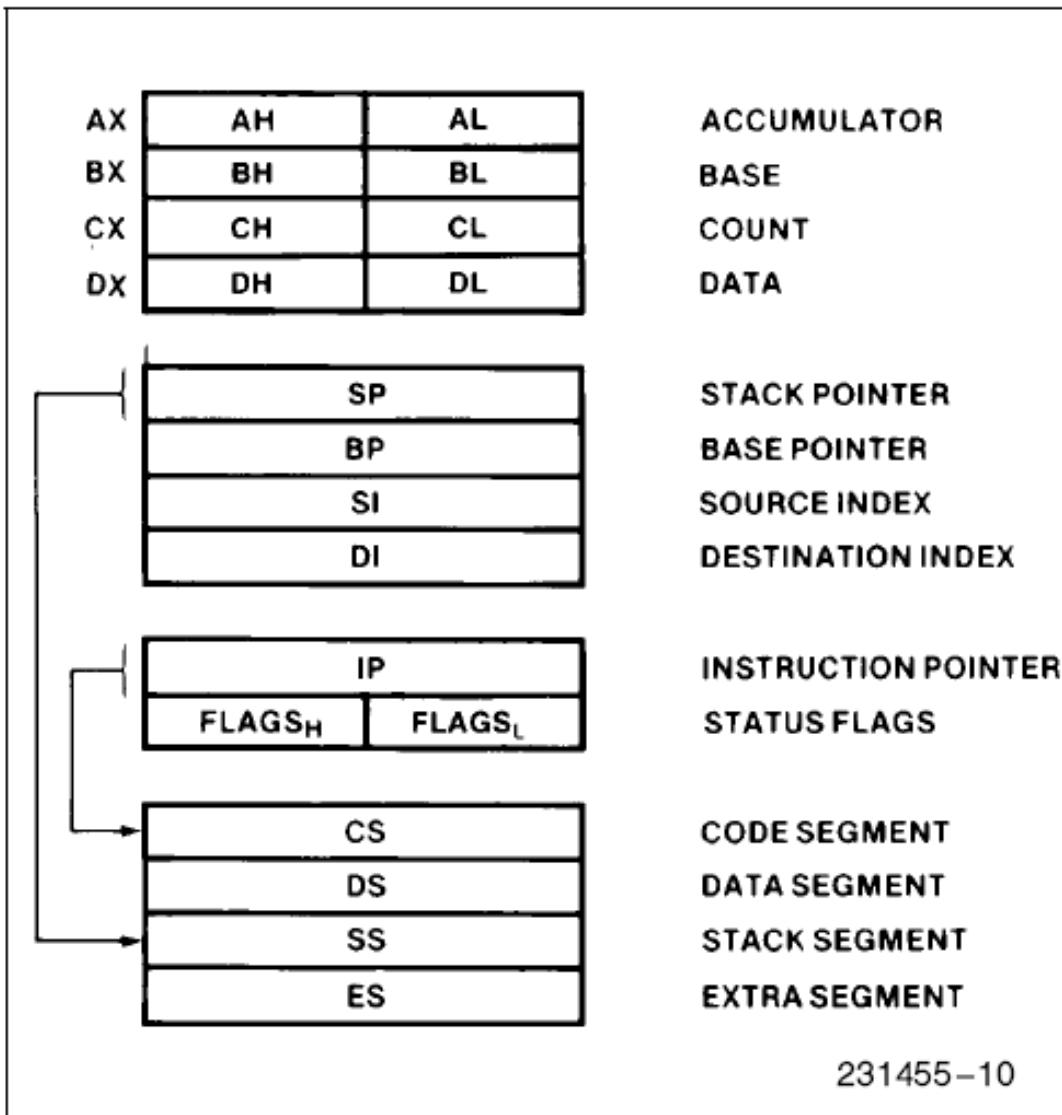


Figure . 8086 Register Model

# Segunda clasificación: (Hacia una mejor aproximación)

## Modos de direccionamiento

De Los procesadores 8086 y 8088

- **Implícito.** El dato está implícito en la propia instrucción. Ej. `STC` , `STD` y `STI` , (Set Carry, Set Direction y Set Interrupts) encienden el flag correspondiente indicado en la propia instrucción. `CBW` (Convert Byte to Word) extiende el bit del signo del registro AL a AX. Ni el AL ni el AX son especificados, puesto que la instrucción `CBW` implicitamente trabaja sobre ellos.
- **Inmediato.** El dato a operar está inmediatamente después del opcode de la instrucción. Ej, `MOV AX, 5`
- **Registro.** El dato está en un segundo registro. Ej. `MOV AX, BX` . Aquí, el dato está en el registro BX
- **Directo.** La dirección del dato está en el campo de la dirección del opcode. Ej. `MOV AX, [100h]` . Aquí se mueve (copia) el contenido de las direcciones 100h y 101h al registro AX. En este caso se mueven dos bytes puesto que AX es de 16 bits. Si fuera `MOV BL, [100h]` se movería solo un byte pues BL es un registro de 8 bits

- **Indirecto.** El dato es especificado mediante una combinación de registros índice y base, y puede haber un desplazamiento
  - **Base.** Un registro base (BX o BP) tienen la dirección de donde se tomará el dato. Ej. MOV AX, [BX]
  - **Índice.** Un registro índice (SI o DI) tienen la dirección de donde se tomará el dato. Ej. MOV AX, [SI]
  - **Base + Desplazamiento.** El dato se tomará de la dirección apuntada por la suma de un registro base más un desplazamiento. Ej. MOV AX, [BP + 7]
  - **Índice + Desplazamiento.** El dato se tomará de la dirección apuntada por la suma de un registro índice más un desplazamiento. Ej. MOV AX, [DI + 7]
  - **Base + Índice.** El dato se tomará de la dirección apuntada por la suma de un registro base más un registro índice. Ej. MOV AX, [BX + SI]
  - **Base + Índice + Desplazamiento.** El dato se tomará de la dirección apuntada por la suma de un registro base, más un registro índice, más un desplazamiento. Ej. MOV AX, [BX + SI + 9]

## *Ejemplos de Modos de direccionamiento*

- *Direccionamiento Inmediato:* Transfiere n bytes o palabras de datos inmediatos hacia el registro o localidad en la memoria en el destino. Ejemplo:

MOV AL, 22H

MOV EBX, 12345678H

MOV AL, 18H

MOV BX,1998H

MOV DS, 0B70H

MOV AX, 0B70H

MOV DS, AX

MOV DL, 0B800H

MOV CX, 1

MOV CX, 0001

# *Ejemplos de Modos de direccionamiento*

- *Direccionamiento Inmediato:* Transfiere n bytes o palabras de datos inmediatos hacia el registro o localidad en la memoria en el destino. Ejemplo:

MOV AL, 22H

MOV EBX, 12345678H

MOV AL, 18H

MOV BX, 1998H

MOV DS, 0B70H

MOV AX, 0B70H

MOV DS, AX

MOV DL, 0B800H

MOV CX, 1

MOV CX, 0001

# *Ejemplos de Modos de direccionamiento*

- **Direccionamiento Inmediato:** Transfiere n bytes o palabras de datos inmediatos hacia el registro o localidad en la memoria en el destino. Ejemplo:

MOV AL, 22H  
MOV EBX, 12345678H

MOV AL, 18H  
MOV BX, 1998H

MOV DS, 0B70H Error, no se puede en direccionamiento inmediato, solo por registro.

MOV AX, 0B70H  
MOV DS, AX

MOV DL, 0B800H Error.  
MOV CX, 1  
MOV CX, 0001

## **Ejemplos de**

- **Direccionamiento por Registro:** Transfiere bytes o palabra desde el registro fuente o localidad de memoria, hasta el registro o localidad destino en la memoria.  
Ejemplo:

MOV CX, DX

MOV AH, DH

MOV BP, BX

MOV AX, BL Error

MOV AL, BL

MOV AH, 0

CBW              Convierte de byte a word, si es positivo llena de ceros, si es neg llena de 1.

- **Direccionamiento Directo:** Transfiere bytes o palabra entre una localidad de memoria y un registro. Ejemplo:

MOV DL, [300H]

MOV DX, [200H]

### **NOTA SOBRE FORMATOS DE ALMACENAMIENTO DE DATOS:**

00	19
01	98
02	

Formato:  
Motorola  
Big Endian

00	98
01	19
02	

Formato:  
Intel  
Little Endian

Si se tiene el número 1998, en los procesadores motorola se guarda de la forma Big Endian, en los procesadores intel se almacena de la forma Little Endian.

## Ejemplos de

- **Direccionamiento por Registro:** Transfiere bytes o palabra desde el registro fuente o localidad de memoria, hasta el registro o localidad destino en la memoria.  
Ejemplo:

MOV CX, DX

MOV AH, DH

MOV BP, BX

MOV AX, BL Error

MOV AL, BL

MOV AH, 0

CBW              Convierte de byte a word, si es positivo llena de ceros, si es neg llena de 1.

- **Direccionamiento Directo:** Transfiere bytes o palabra entre una localidad de memoria y un registro. Ejemplo:

MOV DL, [300H]

MOV DX, [200 H]

## NOTA SOBRE FORMATOS DE ALMACENAMIENTO DE DATOS:

00	19
01	98
02	

Formato:  
Motorola  
Big Endian

00	98
01	19
02	

Formato:  
Intel  
Little Endian

Si se tiene el número 1998, en los procesadores motorola se guarda de la forma Big Endian, en los procesadores intel se almacena de la forma Little Endian.

- **Direccionamiento Indirecto por Registro (Base):** Transfiere bytes o palabra entre un registro y una localidad de memoria direccionada por un registro índice o base. Ejemplo:

```
MOV AX, [BX]  
MOV [BX], AX
```

- **Direccionamiento Base mas Índice:** Transfiere bytes o palabra entre un registro y una localidad de memoria direccionada por un registro base mas un registro índice. Ejemplo:

```
MOV AX, [BX + SI]  
MOV [BX + DI], CX
```

- **Direccionamiento Relativo por Registro:** Transfiere bytes o palabra entre un registro y una localidad de memoria direccionada por un registro índice o base y además un desplazamiento. Ejemplo:

MOV AX, [BX + 4]

- **Direccionamiento Relativo Base más Índice:** Transfiere bytes o palabra entre un registro y una localidad de memoria direccionada por un registro base más un índice mas un desplazamiento. Ejemplo:

MOV AX, [BX + DI + 4]



**EJERCICIOS:** Determine que tipo de direccionamiento es para cada instrucción.

MOV [BX + 4], AX

MOV AX, [BX + SI + 4]

MOV [EBX + 2\*ESI], AX

MOV AX, BX

MOV AX, [1234H]

MOV [BX], AX

MOV BL, 3AH

MOV [BX + SI], AX

**EJERCICIOS:** Determine que tipo de direccionamiento es para cada instrucción.

MOV [BX + 4], AX

MOV AX, [BX + SI + 4]

MOV [EBX + 2\*ESI], AX

MOV AX, BX

MOV AX, [1234H]

MOV [BX], AX

MOV BL, 3AH

MOV [BX + SI], AX

**EJERCICIOS:** Determine que tipo de direccionamiento es para cada instrucción.

MOV [BX + 4], AX	Relativo por registro
MOV AX, [BX + SI + 4]	Relativo base mas índice
MOV [EBX + 2*ESI], AX	Relativo por Índice escalado
MOV AX, BX	Registro
MOV AX, [1234H]	Directo
MOV [BX], AX	Indirecto por registro
MOV BL, 3AH	Inmediato
MOV [BX + SI], AX	Relativo por Base mas índice

## EJERCICIO

*Ejercicio: Implementar un programa en ensamblador que sume la localidad de memoria 200 y la 201, el resultado lo deje en la localidad 202.*

1º OPCIÓN:

```
MOV BX, 200H  
MOV AL, [BX]  
INC BX  
MOV DL, [BX]  
ADD AL, BL  
INC BX  
MOV [BX], AL  
INT 20H
```

2º OPCIÓN:

```
MOV AL, [200H]  
MOV BL, [201H]  
ADD AL, BL  
MOV [202H], AL  
INT 20H
```

*Ejercicio: Implementar un programa en ensamblador que sume la localidad de memoria 200 y la 201, el resultado lo deje en la localidad 202.*

*3º OPCIÓN:*

```
MOV AL, [200H]  
ADD AL, [201H]  
MOV [202H], AL  
INT 20H
```

*4º OPCIÓN:*

```
MOV AX, [200H]  
ADD AL, AH  
MOV [202H], AL  
INT 20H
```

# ORGANIZACIÓN DE LA MEMORIA



# **ORGANIZACIÓN DE LA MEMORIA**

El espacio de direccionamiento de un sistema basado en un microprocesador, se denomina memoria lógica o memoria física. La estructura de la memoria lógica es diferente, en casi todos los casos, que la estructura de la memoria física. La memoria lógica es el sistema de memoria tal como lo ve el programador, mientras que la memoria física es la estructura real en el hardware en el sistema de memoria.

## Memoria lógica.

El espacio básico de la memoria lógica es el mismo en todos los microprocesadores Intel. La memoria lógica se enumera por bytes. En la siguiente ilustración se observa el mapa de memoria lógica de algunos miembros de la familia Intel. Se vera que la única diferencia es que algunos miembros contienen mas memoria que otros. Además, se debe tener en cuenta que la memoria física puede diferir de la memoria lógica en muchos sistemas.

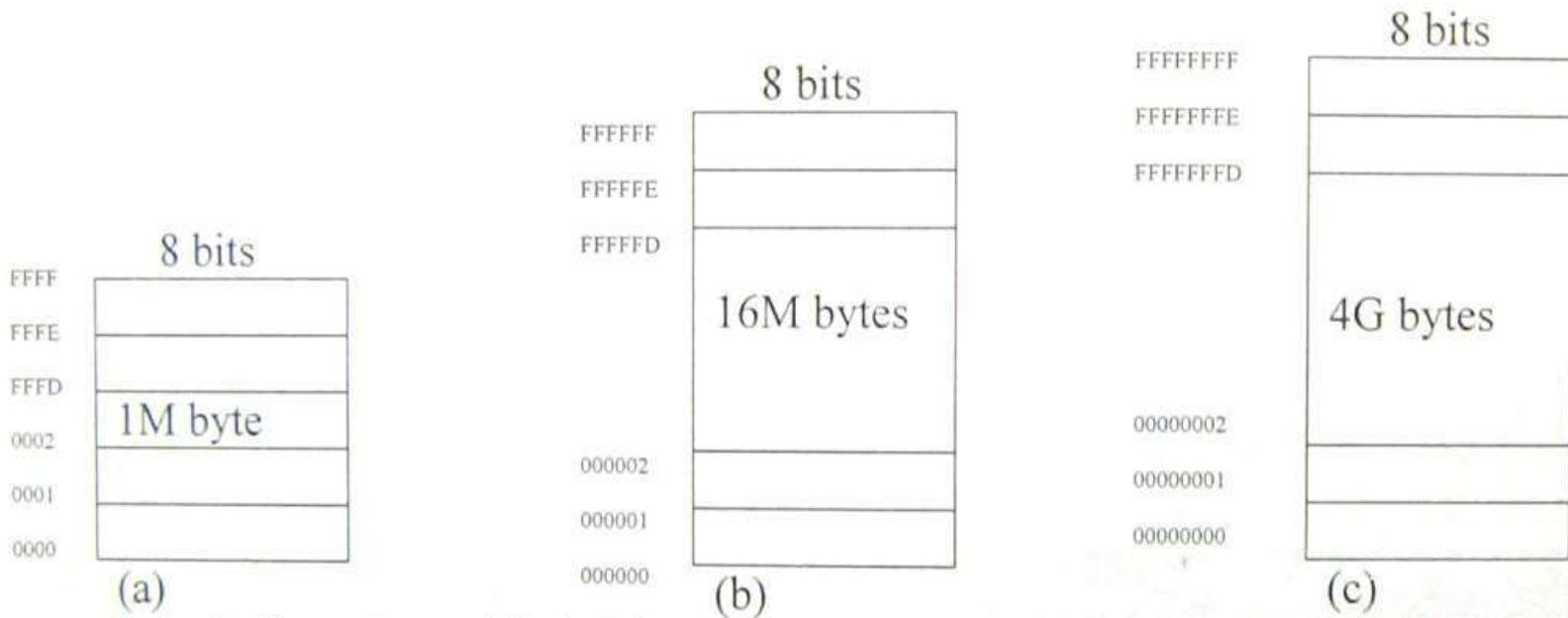


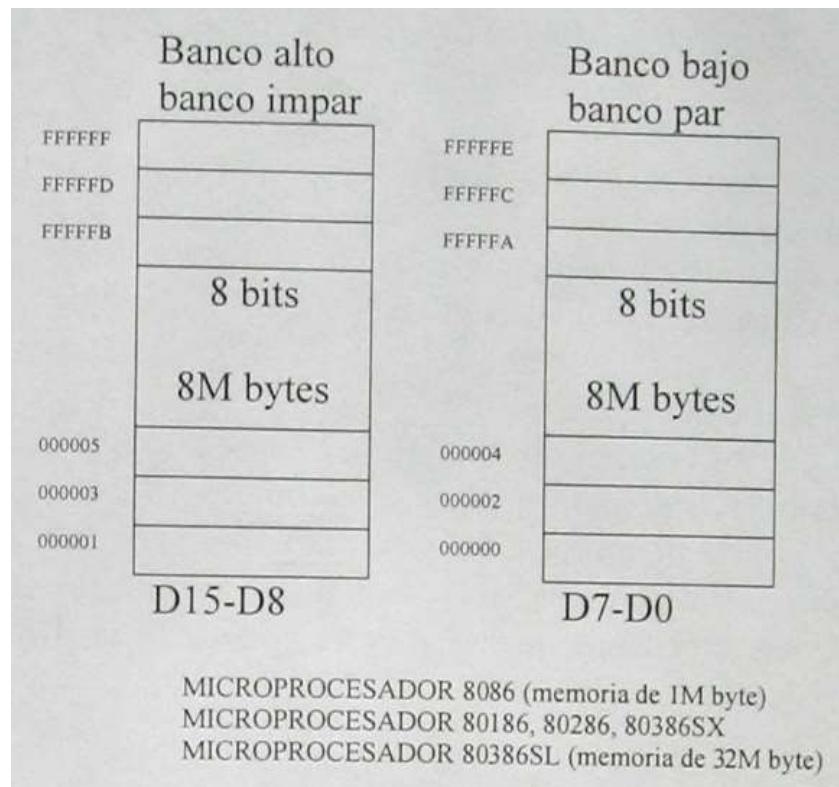
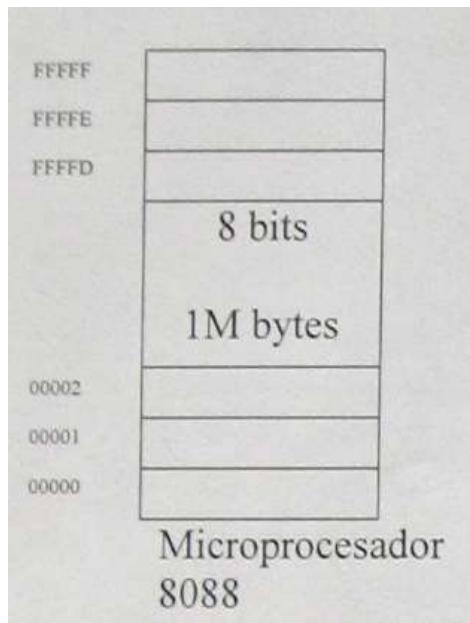
Ilustración: El mapa de memoria lógica de los microprocesadores: (a) 8086/8088/80186 (b) 80286/80386SX (c) 80386DX y 80486

Cuando estos microprocesadores direccionan una palabra de 16 bits en la memoria, se accesan dos bytes consecutivos. Por ejemplo, la palabra en la localidad 00122H se almacena en los bytes 00122H y 00123H; el byte menos significativo se almacena en la localidad 00122H. Si se accesa a una palabra de 32 bits, esta palabra doble la contiene cuatro bytes consecutivos. Por ejemplo. La doble palabra almacenada en la localidad 00120H, se almacena en los bytes 00120H, 00121H, 00122H y 00123H; el byte menos significativo se almacena en 00120H y el byte más significativo en 00123H.

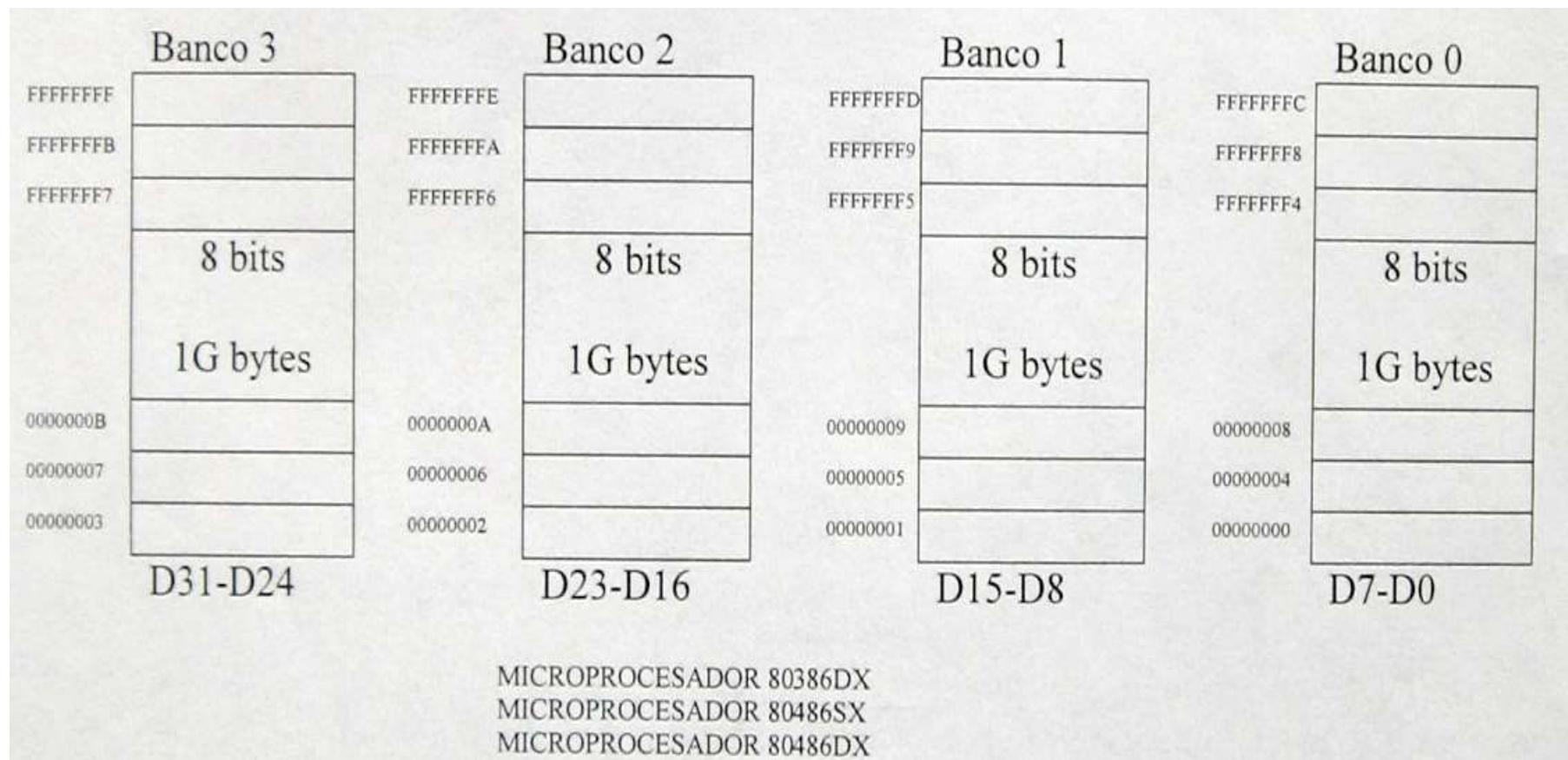
## Memoria Física.

Las memorias físicas de los miembros de la familia Intel difieren del ancho. La memoria del 8088 es de 8 bits de ancho; las memorias 8086, 80186, 80286 y 80386SX tienen 16 bits de ancho; las memorias del 80386dx y 80486 son de 32 bits de ancho. Para la programación, no hay diferencia en el ancho de la memoria porque la memoria lógica siempre es de 8 bits de ancho; pero, como se puede ver en la siguiente ilustración, hay una diferencia para el diseñador del hardware.

La memoria esta organizada en bancos de memoria en todas las versiones del microprocesador excepto el 8088 que tiene un solo banco de memoria. Un *banco de memoria* es una sección de 8 bits de ancho. Los microprocesadores de 16 bits tienen dos bancos de memoria para formar una sección de memoria de 16 bits de ancho, a la cual se direcciona por bytes o por palabras.



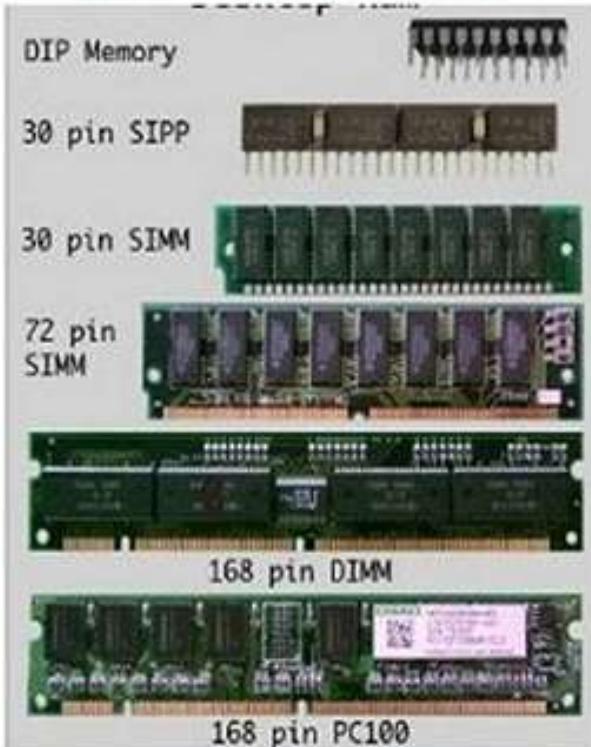
Los microprocesadores de 32 bits tienen cuatro bancos de memoria, pero se les direcciona como bytes, palabras o dobles palabras.



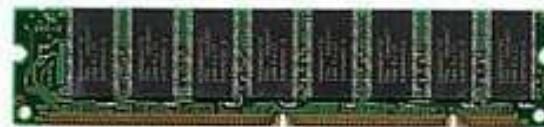
# EJEMPLOS DE BANCOS DE MEMORIA USADOS

## Memoria RAM

### Primeros Modelos de Ram



DIMM



DDR



DDR2



DDR3



# EJEMPLO DE MAPA DE MEMORIA DE LAS PRIMERAS IBM PCs Mod. XT con CPUs INTEL 8088/8086

## PC/XT Memory Map

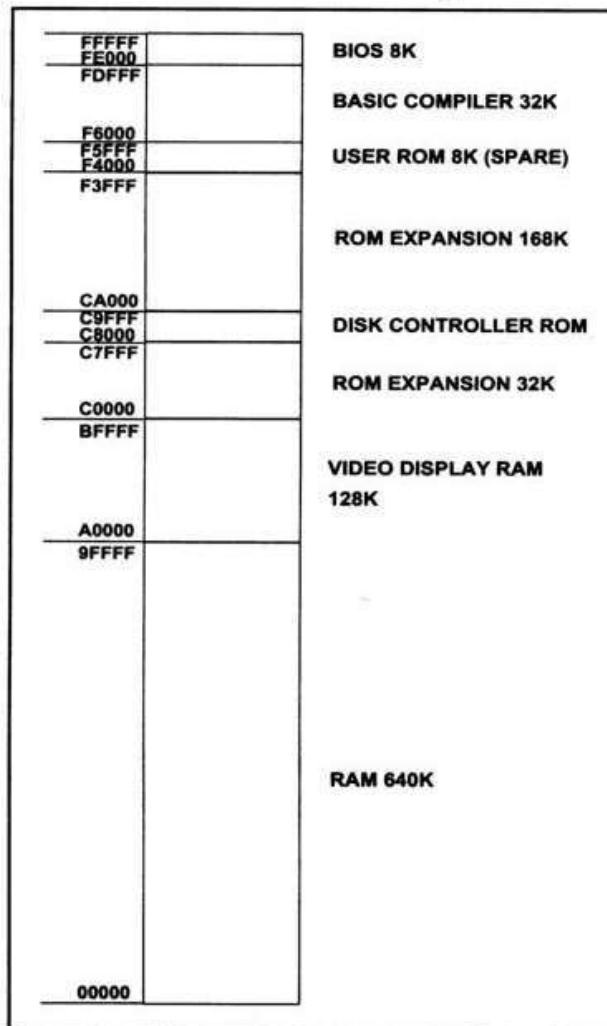
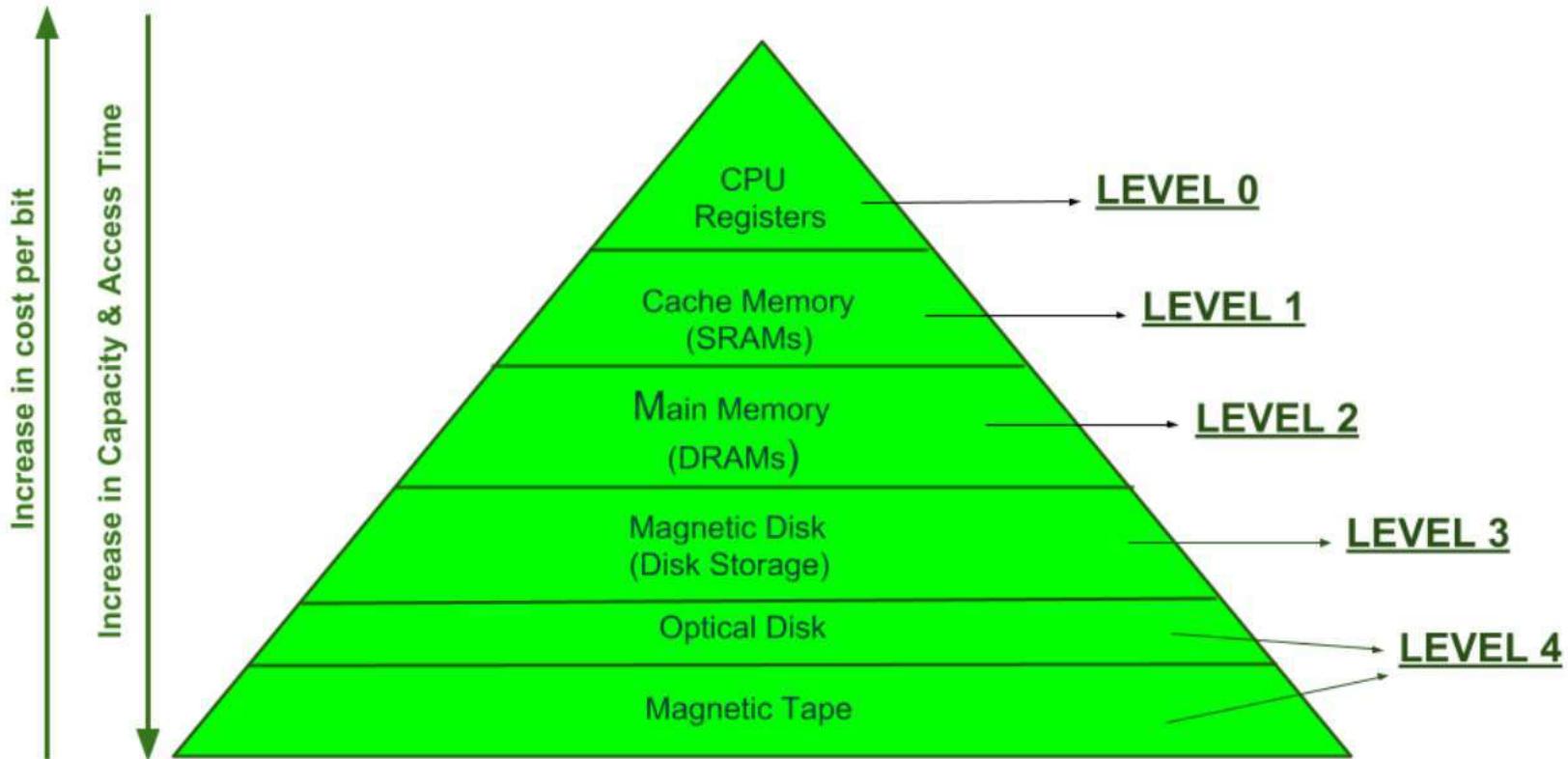


Figure . PC.XT Detailed Memory Map



## MEMORY HIERARCHY DESIGN

## ESTRUCTURA DE SEGMENTACIÓN DE MEMORIA

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 64KB de longitud, que pueden estar separados entre sí, adyacentes o superpuestos, y que comienzan en una dirección divisible por 16.**

La **forma como 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.**



**FIN DE  
PRESENTACIÓN**

# **APENDICES**

## EJERCICIOS Y NOTAS:

