



**Arellano Granados Angel Mariano**  
**218123444**



**Seminario de Traductores de Lenguajes I**  
**I7026 D02**

**Reporte de Actividades 1**  
**Actividades 1 – 2**

## **Actividad 1**

### **Descripción**

Investigar los antecedentes históricos de la familia de los procesadores x86 enfatizando las características de su diseño arquitectónico en una tabla comparativa e incluya el diagrama de la arquitectura de cada procesador.

### **Desarrollo y Resultados**

## PROCESADOR

8086 / 80186

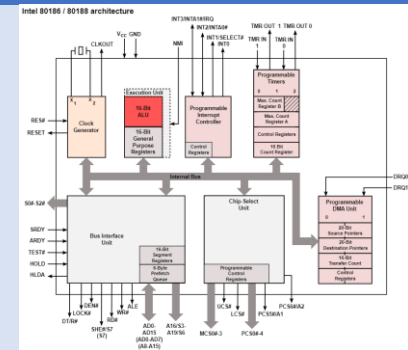


## CARACTERÍSTICAS

### Performance

Max. CPU clock rate	6 MHz to 25 MHz
FSB speeds	6 MHz to 25 MHz
Data width	16 bits
Address width	20 bits

## DIAGRAMA

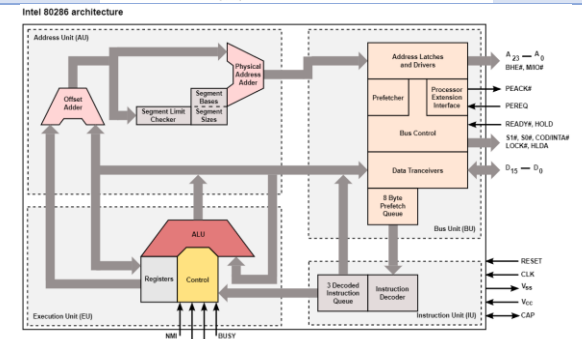


80286



### Performance

Max. CPU clock rate	4 MHz to 25 MHz
FSB speeds	4 MHz to 25 MHz
Data width	16 bits
Address width	24 bits

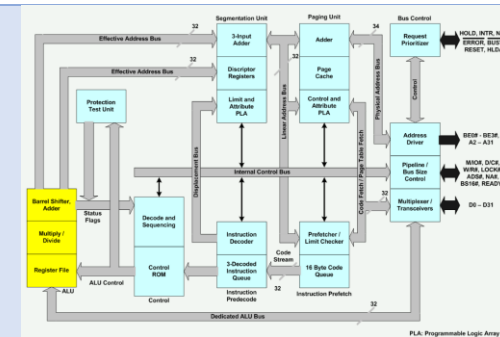


80386

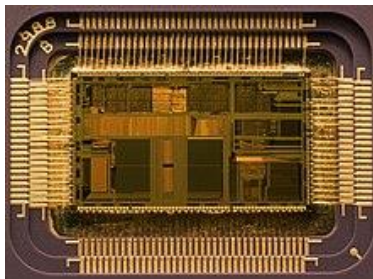


### Performance

Max. CPU clock rate	12 MHz to 40 MHz
Data width	32 bits (386SX: 16 bits)
Address width	32 bits (386SX: 24 bits)

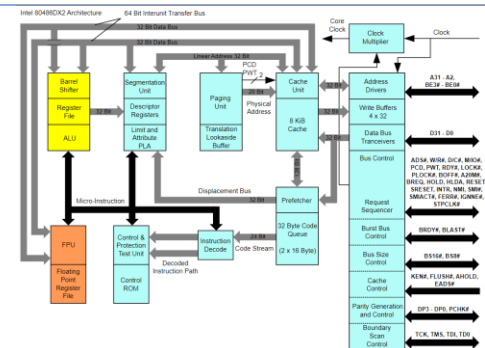


80486



## Performance

Max. CPU clock rate	16 to 100 MHz <sup>[a]</sup>
FSB speeds	16 MHz to 50 MHz
Data width	32 bits <sup>[1]</sup>
Address width	32 bits <sup>[1]</sup>
Virtual address width	32 bits (linear); 46 bits (logical) <sup>[1]</sup>

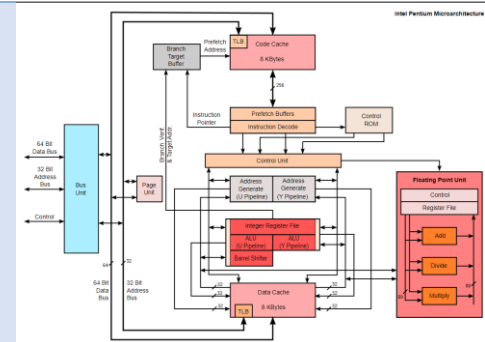


Pentium (80586)



## Performance

Max. CPU clock rate	60–300 MHz
FSB speeds	50–66 MHz
Bus de datos	64 bits

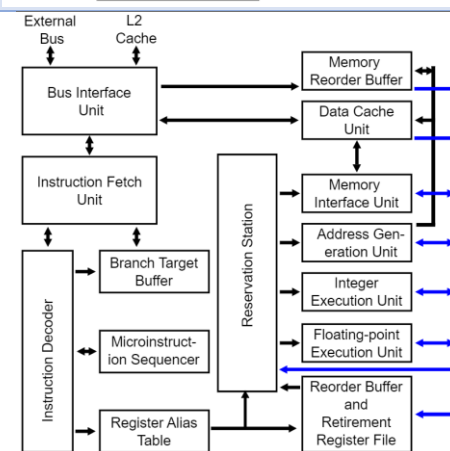


Pentium Pro (6x86)



## Performance

Max. CPU clock rate	150 MHz to 200 MHz
FSB speeds	60 MHz to 66 MHz
Bus de direcciones	36 bits



## Reflexión

Tras investigar las características arquitectónicas de los procesadores de la familia x86 pude observar la gran mejora de rendimiento que tenía cada nueva generación, así como aun no entender a detalle los diagramas ver que cada vez tenían más componentes.

## Actividad 2

### Descripción

Investigar tres instrucciones de cada una de las siguientes categorías, incluya una descripción y un ejemplo.

Aritméticas (ADC, ADD, DEC, DIV, IDIV, IMUL, MUL, NEG, SBB, SUB, XADD)

Conversión de ASCII a BCD (AAA, AAD, AAM, AAS, DAA, DAS)

Cambio de bit (RCR, RCL, ROL, ROR, SAL, SAR, SHL, SHR, SHLD, SHRD)

Comparación (BSF/BSR, BT/BTC/BTR/BTS, CMP, CMPS, CMPXCHG, CMPXCHG88, TEST)

Transferencia de datos (LDS, LEA, LES, LODS, LSS, MOV, MOVS, MOVSX, MOVZX, SOPS, XCHG, XLAT)

Operaciones de banderas (CLC, CLD, CLT, PUSHF, SAHF, STC, CMC, LAHF, POPF, STD, STI)

Operaciones lógicas (AND, OR, NOT, XOR)

Ciclos (LOOP, LLOPE, LOOPZ, LOOPNE, LOOPNZ, LOOPNEW, LOOPNZW)

Operaciones de Cadenas (CMPS, LODS, MOVS, REP, REPE, REPZ, REPNE, REPNZ, SCAS, STOPS)

Y además incluya los tipos de datos que emplean los microprocesadores de la familia x86.

## Desarrollo y Resultados

INTRUCCIÓN	DESCRIPCIÓN	EJEMPLO
ARITMATECAS		
DEC	Resta 1 al operando de la instrucción y almacena el resultado en el mismo operando.	DEC EAX DEC R9
INC	Suma 1 al operando de la instrucción y almacena el resultado en el mismo operando.	INC AL INC R9
XAND	Lleva a cabo una negación aritmética del operando, es decir, hace el complemento a 2 del operando especificado.	NEG RCX NEG DWORD [variable]
CONVERSIÓN DE ASCII A BCD		
AAA	Ajusta el resultado binario de una instrucción ADD o ADC	ADD al, '2' AAA
AAS	Va después de una instrucción SUB o SBB que resta un valor decimal desempaquetado de otro, y almacena el resultado en AL.	SUB al, val2 AAS
DAA	convierte una suma binaria producida por ADD o ADC en AL, a formato decimal empaquetado.	ADD al, 48h DAA
CAMBIO DE BIT		
ROL	Lleva a cabo una rotación de los bits del operando destino a la izquierda, es decir, hacia al bit más significativo; rota tantos bits como indica el operando fuente	ROL RAX, CL ROL RAX,1
SAL	Lleva a cabo un desplazamiento a la izquierda de los bits del operando destino; desplaza tantos bits como indica el operando fuente.	SAL RAX, CL SAL RAX,1
SAR	Lleva a cabo un desplazamiento a la derecha de los bits del operando destino; desplaza tantos bits como indica el operando fuente	SAR RAX, CL SAR RAX,1
COMPARACIÓN		

BTS	La instrucción BTS (prueba y activación de bit) selecciona el bit n en el primer operando, lo copia a la bandera Acarreo y lo activa: BTS baseBit, n	BTS semáforo, 6
CMP	Compara los dos operandos de la instrucción actualiza los bits de resultado según el resultado de la comparación.	CMP R9, RAX
TEST	Realiza una operación lógica 'y' bit a bit entre los dos operandos sin modificar el valor de ninguno de los operandos; actualiza los bits de resultado según el resultado de la 'y' lógica.	TEST R9, RAX
TRANSFERENCIA DE DATOS		
MOV	Copia datos de un operando de origen a un operando de destino.	MOV reg,reg
LEA	Devuelve el desplazamiento de un operando indirecto. Como los operandos indirectos contienen uno o más registros, sus desplazamientos se calculan en tiempo de ejecución.	LEA esi, [ebp-32]
LDS	Carga el contenido de un operando de memoria tipo doble palabra en un registro de segmento y en el registro de destino especificado.	LDS reg, mem
OPERACIONES DE BANDERAS		
CLC	Borra la bandera Acarreo, asignándole un cero.	CLC
POPF	Saca la parte superior de la pila y la coloca en el registro FLAGS de 16 bits.	POPD
PUSHF	mete el registro FLAGS de 16 bits en la pila.	PUSHF
OPRECIONES LOGICAS		
AND	Realiza una operación lógica AND bit a bit entre el operando destino y el operando fuente, el resultado de la operación se guarda sobre el operando destino sobrescribiendo el valor inicial	AND R9, RAX

OR	Realiza una operación lógica OR bit a bit entre el operando destino y el operando fuente; el resultado de la operación se guarda sobre el operando destino, sobrescribiendo el valor inicial	OR R9, RAX
NOT	Lleva a cabo una negación lógica del operando, es decir, hace el complemento a 1 del operando especificado, y complementa todos los bits del operando.	NOT RAX
CICLOS		
LOOP	La instrucción utiliza el registro RCX. Decrementa el valor de RCX, comprueba si el valor es diferente de cero y en este caso realiza un salto a la etiqueta indicada.	MOV RCX, 10 bucle: ; INSTRUCCIONES LOOP bucle
LOOPNZ	es la contraparte de LOOPZ. El ciclo continúa mientras el valor sin signo de ECX sea mayor que cero, y la bandera Cero esté en cero	LOOPNZ destino
LOOPZ	permite que un ciclo continúe mientras esté activa la bandera Cero y el valor sin signo de ECX sea mayor que cero.	LOOPZ destino
OPERACIONES DE CADENA		
LODS	Las instrucciones CMPSB, CMPSW y CMPSD comparan un operando de memoria al que apunta ESI, con un operando de memoria al que apunta EDI	cmps DWORD PTR [esi],[edi]
REP	Repite mientras que ECX > 0	REP movsd
SCAS	Explora una cadena en memoria a la que apunta ES:(E) DI, en busca de un valor que coincida con el acumulador. SCAS requiere que se especifiquen los operandos	SCAS ES:dest



## **Reflexión**

Después de investigar las diferentes instrucciones de los microprocesadores x86 entendí su uso pero al no tener experiencia con el lenguaje ensamblador no logre darles un uso practico en mi cabeza, supongo que el futuro si se me presenta una duda volveré a revisar la tabla para ver si una de las instrucciones me ayuda.