institución pau casals

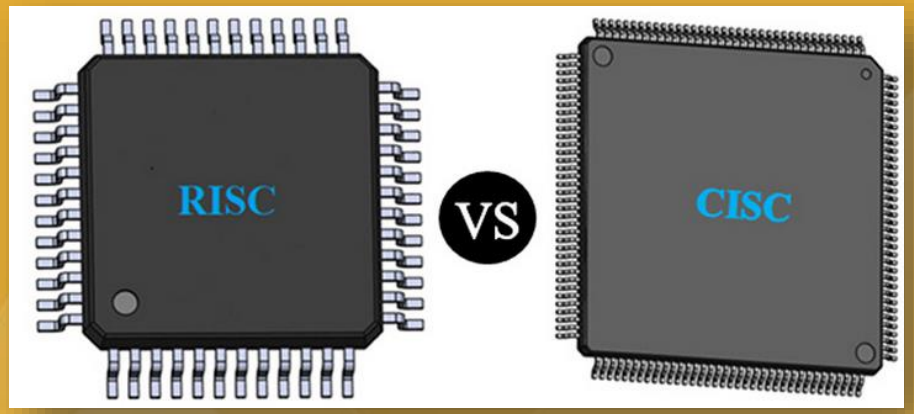
(IFCT0310) ADMINISTRACIÓN DE BASES DE DATOS

- **Computadores para BBDD** -

institución pau casals

Tipos de procesadores en los servidores de BBDD

¿RISC O CISC?



- CISC (Complex Instruction Set Computer)
- RISC (Reduced Instruction Set Computer)

Puntos clave:

- Complejidad de las instrucciones
- Longitud de las instrucciones (variable vs. fija)
- Decodificación de las instrucciones (varios ciclos o sólo uno)
- Complejidad de los compiladores

Conjunto de instrucciones:

WIKIPEDIA
La enciclopedia libre

Conjunto de instrucciones

17 idiomas

Artículo Discusión | Leer Editar Ver historial Herramientas

En este artículo sobre informática se detectaron varios problemas. Por favor, edítalo y/o discute los problemas en la discusión para mejorarlo:

- Necesita ser **wikificado** conforme a las convenciones de estilo de Wikipedia.
- Carece de **fuentes o referencias** que aparezcan en una fuente acreditada.

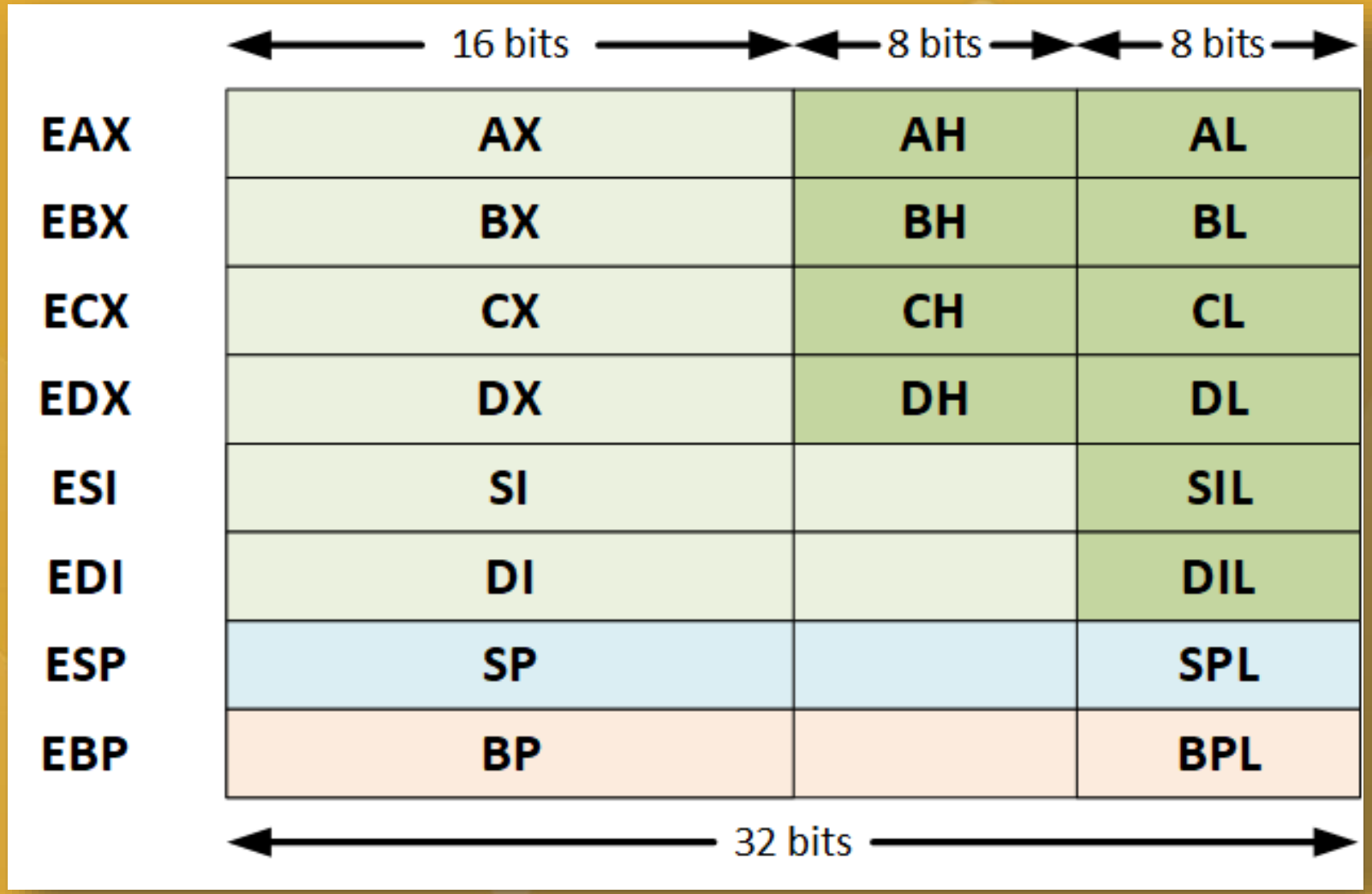
Este aviso fue puesto el 12 de marzo de 2015.

Un **conjunto de instrucciones**, **repertorio de instrucciones**, **juego de instrucciones** o **ISA** (del inglés *instruction set architecture*, «arquitectura del conjunto de instrucciones») es una especificación que detalla las instrucciones que una unidad central de procesamiento puede entender y ejecutar, o el conjunto de todos los comandos implementados por un diseño particular de una CPU. El término describe los aspectos del procesador generalmente visibles para un programador, incluidos los tipos de datos nativos, las instrucciones, los registros, la arquitectura de memoria y las interrupciones, entre otros aspectos. Existen principalmente tres tipos: CISC (*Complex Instruction Set Computer*), RISC (*Reduced Instruction Set Computer*) y SISC (*Simple Instruction Set Computing*).

La arquitectura del conjunto de instrucciones (ISA) se emplea a veces para distinguir este conjunto de características de la microarquitectura, que son los elementos y técnicas que se emplean para implementar el conjunto de **instrucciones**. Entre estos elementos se encuentran las microinstrucciones y los sistemas de caché.

Procesadores con diferentes diseños internos pueden compartir un conjunto de instrucciones; por ejemplo, el Intel Pentium y AMD Athlon implementan versiones casi idénticas del conjunto de instrucciones x86, aunque tienen diseños diferentes.

Conjunto de instrucciones 8, 16, 32 y 64 bits:



Clasificación del conjunto de instrucciones 8086:

1.Instrucciones de transferencia de datos:

1. MOV: Transfiere datos entre registros, memoria o ambos.
2. XCHG: Intercambia el contenido de dos registros o entre un registro y la memoria.

2.Instrucciones aritméticas y lógicas:

1. ADD, SUB, INC, DEC: Realizan operaciones aritméticas como suma, resta, incremento y decremento.
2. AND, OR, XOR, NOT: Realizan operaciones lógicas como AND, OR, XOR y NOT entre operandos.
3. MUL, IMUL, DIV, IDIV: Realizan multiplicación y división, con o sin signo, entre operandos.

3.Instrucciones de control de flujo:

1. JMP: Realiza un salto incondicional a otra parte del programa.
2. JZ, JNZ, JC, JNC, JS, JNS, JO, JNO: Realizan saltos condicionales según el estado de las banderas.
3. CALL: Llama a una subrutina o procedimiento.
4. RET: Retorna de una subrutina.
5. INT: Genera una interrupción del software.

4.Instrucciones de comparación:

1. CMP: Compara dos operandos y establece las banderas según el resultado.
2. TEST: Realiza una operación AND entre dos operandos y establece las banderas según el resultado.

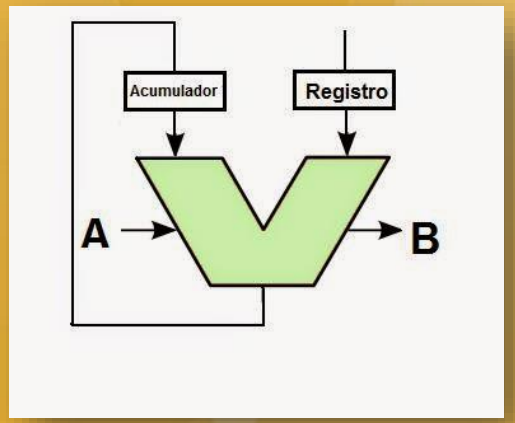
5.Instrucciones de entrada/salida:

1. IN: Lee datos de un puerto de entrada.
2. OUT: Escribe datos en un puerto de salida.

Ejemplo de instrucción de CPU 8 bits (lenguaje ensamblador):

MOV A, #25h

Esta instrucción mueve el valor hexadecimal 25 al registro acumulador A del procesador 8051. En este caso, "A" es el acumulador y "#25h" es el valor que se va a mover.



MOV R1, R0

Esta instrucción mueve el contenido del registro R0 al registro R1 del procesador 8051. "R1" y "R0" son registros de propósito general.

Ejemplo de instrucción de CPU 8 bits (bin. – cód. máquina):

MOV A, #25h → "108" → 0110 1100

- **0110** representa el código de operación para la instrucción MOV A, inmediato.
- **1100** es el valor inmediato 25h que se moverá al registro A.

MOV R1, R0 → "145" → 1001 0001

- **1001** representa el código de operación para la instrucción MOV de registro a registro.
- **0001** representa los registros R1 y R0, donde el contenido de R0 se moverá a R1.

Ejemplo de instrucción de CPU 8 bits (bin. – cód. máquina):

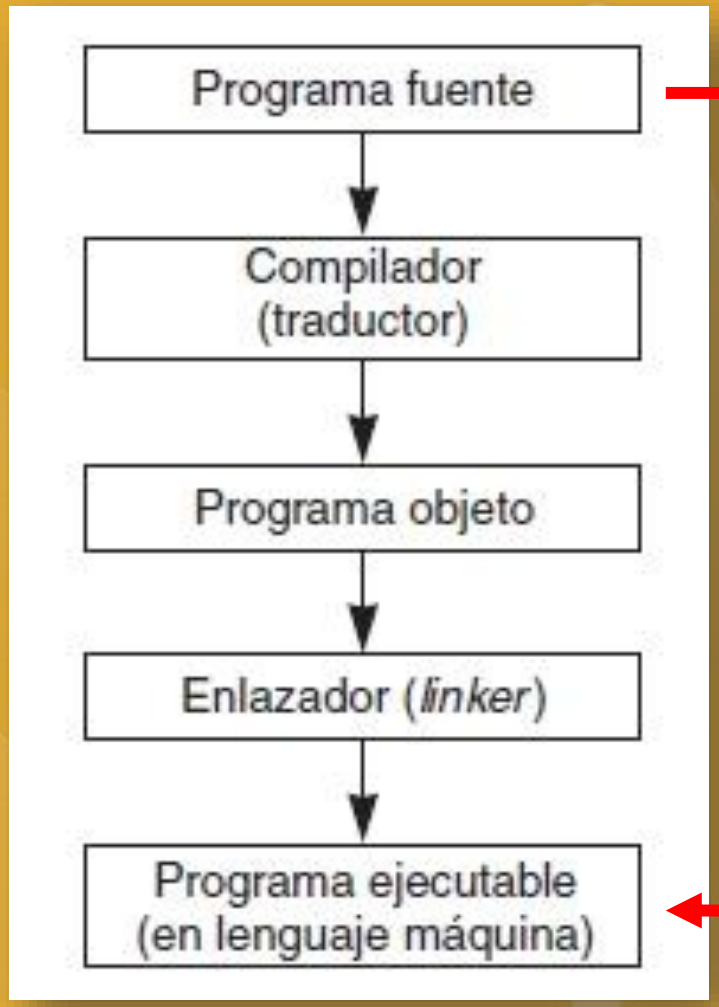
MOV A, #25h → "108" → 0110 1100

- **0110** representa el código de operación para la instrucción MOV A, inmediato.
- **1100** es el valor inmediato 25h que se moverá al registro A.

MOV R1, R0 → "145" → 1001 0001

- **1001** representa el código de operación para la instrucción MOV de registro a registro.
- **0001** representa los registros R1 y R0, donde el contenido de R0 se moverá a R1.

COMPILACIÓN:




```
int sum(int a, int b) {
    return a + b;
}
```

```
sum:
    mov eax, DWORD PTR [esp+4]
    add eax, DWORD PTR [esp+8]
    ret
```

```
10001011 01000100 00100100 00000100
00000011 01000100 00100100 00001000
11000011 00000000
```

SIMULACIÓN:

https://lab.xitrus.es/VonNeumann/


institución pau casals