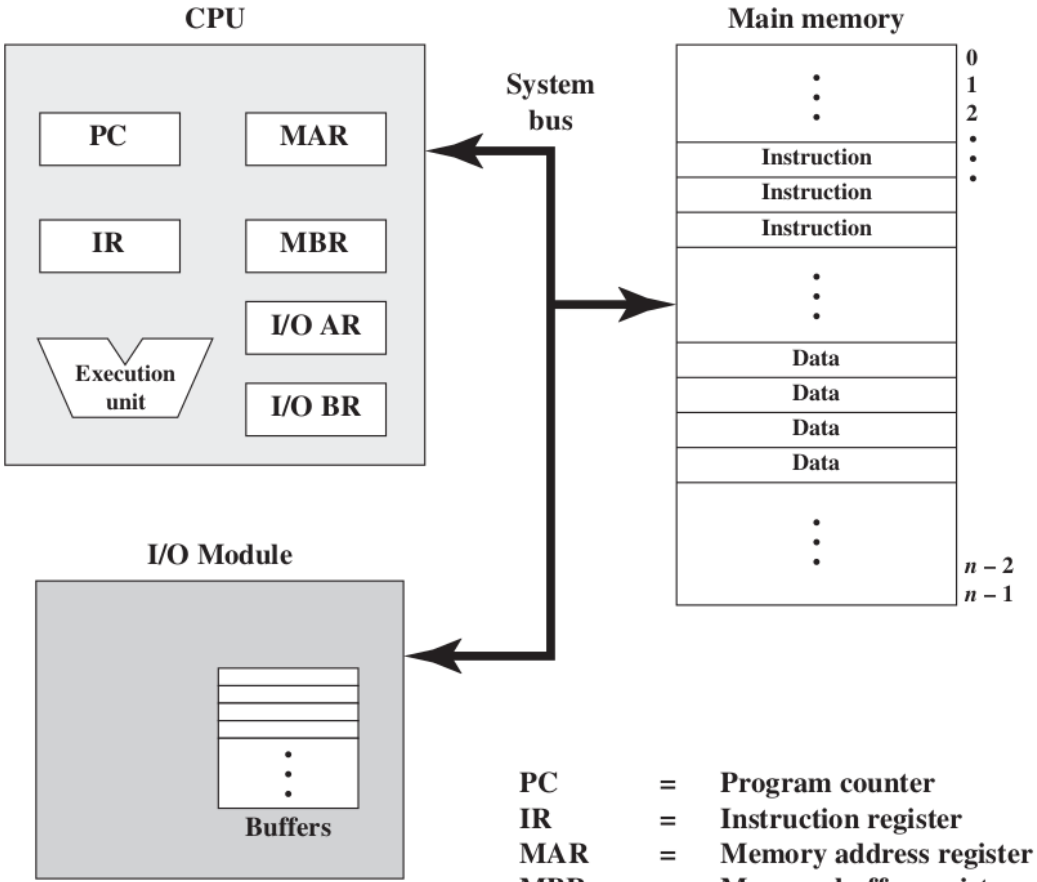
MCBE y Software

Repasando...

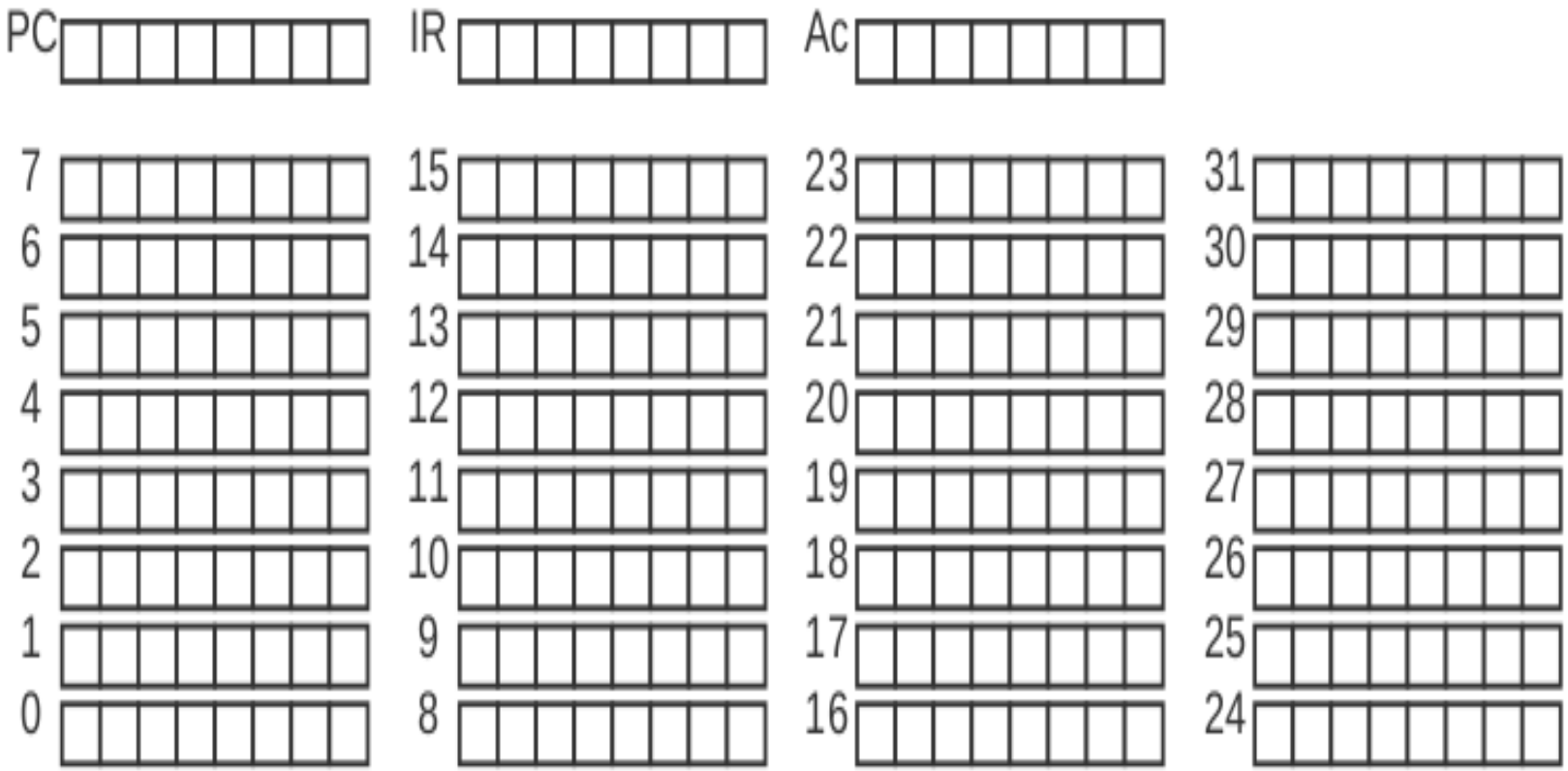


PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

Modelo Computacional Binario Elemental (MCBE)

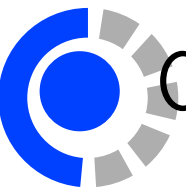
- Hardware:
  - Tres registros de 8 bits: **PC**, **IR** y **AC**.
  - 32 celdas de memoria de 8 bits.
  - Las celda 30 esta mapeada a entrada, y la 31 a salida.
- Software:
  - Posee 8 instrucciones: 2 de carga y almacenamiento, 2 de operaciones aritméticas, 2 de salto, 1 para detener el programa, 1 no op.

Hardware del MCBE



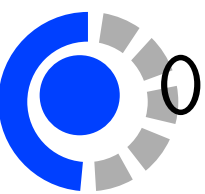
Software del MCBE

```
        LD      CANT
SIGUE:  JZ      FIN
        ST      OUT
        SUB     UNO
        JMP     SIGUE
FIN:    HLT
UNO:    1
CANT:   3
```



¿Dónde se ubica un proceso en MCBE?

```
0000000            LD      CANT
0000001 SIGUE:     JZ      FIN
0000010            ST      OUT
0000011            SUB     UNO
0000100            JMP     SIGUE
0000101 FIN:       HLT
0000110 UNO:       1
0000111 CANT:      3
```



Mnemónicos

- LD y ST
- ADD y SUB
- JMP y JZ
- HLT
- NOP

Rótulos

- Cuando necesitamos hacer referencia a una dirección, como en las operaciones de transferencia o en las aritméticas, el ensamblador nos permite independizarnos del valor de esa dirección y simplemente indicar un nombre simbólico o rótulo para esa dirección.

Rótulos

- Para que el programa quede completo, ese nombre simbólico debe aparecer en algún lugar del programa, al principio de la instrucción, y separado por un carácter : del resto de la línea.

Entrada en MCBE

- La posición 30 es de sólo lectura, es decir, no se pueden escribir contenidos en esa dirección. Cuando leemos esa posición de memoria, la máquina detiene su programa y espera que se introduzca un dato.

Salida en MCBE

- La posición 31 es solamente de escritura. Al escribir un dato en la dirección 31, hacemos que la MCBE escriba ese valor por pantalla.

Lenguajes de programación

- Lenguajes de bajo nivel.
  - Lenguaje máquina.
  - Lenguaje ensamblador.
- Lenguajes de alto nivel (Por ejemplo, Lenguaje C).

Lenguajes de alto nivel

- Paradigma imperativo: se explicita el orden de las acciones:
  - Procedudal (ejemplo: *C*).
  - Orientado a objetos (ejemplo: *SmallTalk*).
- Paradigma declarativo: se declaran propiedades o predicados que deben cumplirse.
  - Lógico (ejemplo: *Prolog*).
  - Funcional (ejemplo: *Lisp*).

Paradigma Procedural

Factorial en *C*

```
int factorial(int n)
{
        int f = 1;
        while (n > 1) {
                f *= n;
                n--;
        }
        return f;
}
```

Paradigma Lógico

Factorial en *Prolog*

```
factorial(0,X):- X=1.
factorial(N,X):- N1=N-1, factorial(N1,X1), X=X1*N.
factorial(N):- factorial(N,X), write(X).
```

Paradigma Funcional

Factorial en *Lisp*

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1))) ) )
```

Ensamblador

- Ahora bien, si escribimos el programa en lenguaje ensamblador, ¿cómo hacemos para que la máquina lo ejecute?

Ensamblador

- Ahora bien, si escribimos el programa en lenguaje ensamblador, ¿cómo hacemos para que la máquina lo ejecute?
- Traducción: Programa ensamblador

Ensamblador

- Ahora bien, si escribimos el programa en lenguaje ensamblador, ¿cómo hacemos para que la máquina lo ejecute?
- Traducción: Programa ensamblador
- Un lenguaje y un programa ensamblador están ligados a un procesador determinado.

ISA: Instruction Set Architecture

- Es el conjunto de instrucciones que ejecuta una determinada CPU o famila de procesadores.

Arquitectura

- La arquitectura de una computadora es un modelo abstracto y se puede definir como lo que necesita saber alquien que programa en el lenguaje ensamblador de esa computadora (instrucciones, modos de direccionamiento, registros).

Familia x86

- x86 es una familia de arquitecturas desarrollada inicialmente por Intel (para el procesador 8086).
- Cualquiera de los procesadores de la familia x86 puede ser programado usando un lenguaje ensamblador x86.

Vamos que falta poco!
Recuerden leer el apunte!