

IFU

Tema 2 – Arquitecturas de Ord Digital

Arquitectura Von Neumann

Primeros tiempos -- Ordenadores con electró. compleja muy susceptible a fallos y usaban 1 sist. de programación cableado.

En el 49 B. Newman propuso 2 conceptos:

- Almacenar la secuencia de instrucciones (Prog.) en 1 memoria interna + accesible.
- Usar el sist. binario.

Así idea una máquina con un reducido conjunto de instrucciones y está formada por 3 partes:

Memoria Principal Unidad Central Unidades

O de E/S

Central Procesos

CPU

Es el cerebro y realiza las operaciones aritmético-lógicas y controla las Transf. de información a realizar.

MEMORIA CENTRAL

Es el lugar de almacenaje temporal de:

- Las instrucciones que ejecuta la CPU
- Los datos que se precisan para la ejecución del programa.
- Los resultados que devuelve la CPU

UNIDADES E/S

Comunican la máquina con el exterior (Introduces progr. y datos obtienes resultados).

Los ordenadores modernos, a pesar de las constantes innovaciones introducidas de electrónica y HW , todavía conservan la estructura básica de la arquitectura de Von Newman.

Unidad Central de Proceso (CPU)

Rige todo el sistema:

- Gobierna la memoria.
- Se encarga de las operaciones de E/S.

Se compone de 2 partes:

- UC (Unidad de Control)
- ALU (Unidad Aritmético-Lógica)

UC

Se encarga de sincronizar, dirigir y controlar todos los demás elementos del sistema. Su principal f(x) es la de ejecutar las distintas instrucciones que forman 1 programa. Está formado por 5 partes:

- RI (Registro de Instrucción)

Contiene la instrucción que actualmente se está ejecutando y se divide en 2 partes:

- Campo de Operación (OP)
- Campos de Direccionamiento (DIR)
- CP (Controlador de Programa)

Contiene la dirección de memoria donde está alojada la próxima instrucción a ejecutar.

- Decodificador

Coge del RI, el campo de operación y lo manda decodificado al secuenciador. Un decodificador consta de 2 salidas por entrada.

- Secuenciador.

Almacena todas las posibles instrucciones soportadas por el procesador y tb se encarga de generar las señales de control para gobernar el flujo del programa. Contiene una memoria ROM con el microcódigo del procesador. Se encarga de aplicar dicho microcódigo asociado adicha instrucción.

- Reloj.

Sincroniza los demás elementos de la CPU. (HZ)

ALU

Es un circuito lógico que nos permite realizar operaciones lógicas, aritméticas y de desplazamiento que necesita realizar la CPU. Consta de 3 partes:

- Circuito ALU.

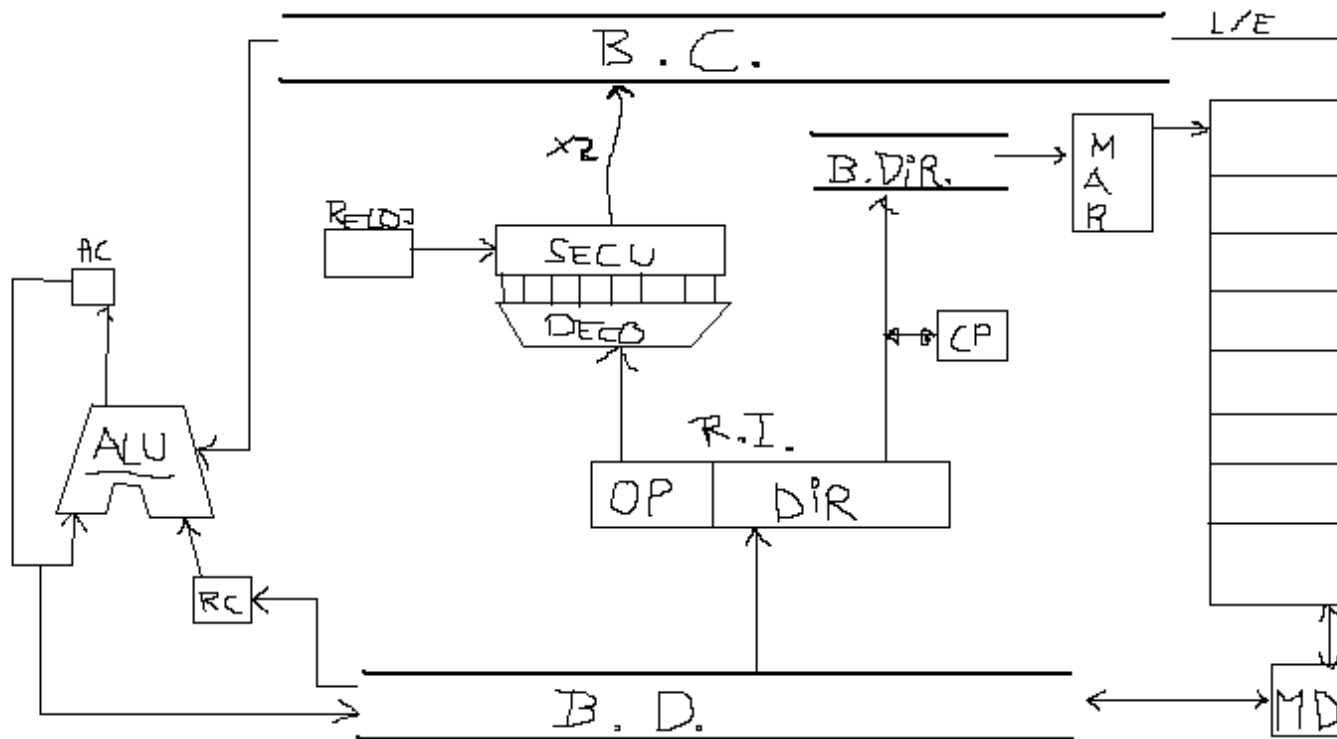
Circuito lógico donde se llevan a cabo los cálculos indicados por la UC.

- RC (registro de carga)

Hace la f(x) de filtro de toda la información que entra al ALU. Contiene de manera temporal los diversos datos que intervienen en las operaciones que se están realizando.

- AC (Acumulador)

Es un registro donde se vuelca el resultado de la operación realizada ; el cual a su vez puede intervenir en operaciones posteriores.



Por ejemplo si tenemos la instrucción MUL 100 200 300, esta viaja por el bus de datos hasta el RI (MUL se pone en el OP y 100 en DIR) De manera inmediata el código de la siguiente instrucción se vuelca en el CP. El contenido de DIR viaja por el bus de datos hasta el MAR. Una vez allí, el secuenciador, a través de una llamada de E/S sabe el lugar exacto en memoria del registro 100 y vuelca su contenido en el MDR. Luego viaja por el Bus de Datos hasta el RC (registro de carga) y a su vez el ALU lo coloca en el AC (acumulador). De manera inmediata el registro 200 sigue el mismo proceso que el anterior registro hasta llegar al RC. Después el ALU procesa los datos de RC y del AC según se lo indique el secuenciador (que ha mandado la orden pertinente por el Bus de Control). Los datos resultantes viajan por el Bus de Datos hasta el MDR. El registro 300 ya está cargado en el MAR y por una orden de escritura los datos del MDR son guardados en el registro 300 y el código de la siguiente instrucción es procesado para cargar la instrucción indicada en el RI.

Tema 4 – Sistemas de numeración / Conversiones de base

Representación de datos en el ordenador

Desde los 1º ordenadores surge el problema de traducir los datos para que sean comprensibles por el ordenador. Después de que V. Newman saque su modelo es cuando nace la representación en el llamado Sistema Binario (solo admite 2 estados muy diferenciados)

Aquí es cuando surge el término bit (Binary Digit), que es la unidad mínima de información comprensible por un ordenador. A partir del bit se obtienen:

8 bits 1 Byte

1024 Bytes (2^{10}) 1 KB

1024 KB 1 MB

1024 GB 1 Petabyte

Sistemas de numeración

Hay de 2 tipos:

- No posicionales (Romanos)
- Posicionales.

Cualquier número se puede descomponer en el polinomio de potencias:

$$\text{PolPot} = [X_{n-1} * B^{n-1}] + [X_{n-2} * B^{n-2}] + \dots + [X_0 * B^0]$$

Para la parte fraccionaria de un núm el PolPot puede ser expresado:

$$\text{PolPot} = [X_{-n+2} * B^{-n+2}] + [X_{-n+1} * B^{-n+1}] + \dots + X_{-n} * B^{-n}$$

B = Base

Sistema decimal

Base = 10

Símbolos = 0,1,2,3,...,9

Los pesos de los números son potencias de 10 que aumentan de derecha a izquierda.

Pesos enteros = $10^0, 10^1, 10^2, 10^3, \dots, 10^{n-1}$

Sistema Binario

Base = 2

Símbolos = 0 y 1

Peso parte entera = $2^0, 2^1, \dots, 2^{n-1}$

Peso parte fraccionaria = $2^{-1}, 2^{-2}, \dots, 2^{-n-1}$

Este sistema es muy engorroso para representar números largos pero es el más comprensible por la máquina y presenta una aritmética sencilla:

Decimal	Binario
0	0000
1	0001
2	0010

3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Sistema Octal

Permite pasar fácilmente a binario de un modo más compacto ya que cada 3 dígitos binarios son uno octal.

Base = 8

Símbolos = 0, ,7

Sistema hexadecimal

Base = 16

Símbolos = 0,, 9,A,,F

Un dígito hexadecimal son 4 binarios. Las direcciones de memoria suelen venir dadas en este sistema de numeración.

Conversión de Cualquier base decimal

Usamos el polinomio de potencias (multiplicando por la base en la que esté el número)

Conversión de decimal cualquier base

Usaremos el método de divisiones sucesivas. Se va dividiendo el número por la base a la que queremos convertir hasta que el número sea más pequeño que la base.

Resultado = Último cociente + restos en orden inversos a como

fueron obtenidos.

Para pasar la parte fraccionaria de un número decimal a cualquier base se multiplica la parte fraccionaria por la base del sistema que se quiere representar. Con la parte fraccionaria de dicha * volvemos a repetir el proceso hasta que se obtiene un producto entero, si es posible o encontrar periodicidad.

Resultado = partes enteras en orden de aparición (arribaabajo)

Conversión de cualquier base a cualquier base

Resultado = pasar a decimal por el PolPot + divisiones sucesivas