Clase Nº10-11
UAL-UC:
UAL: *Su misión en la operación de la computadora. Operaciones. Sus componentes y funcionamiento. Instrucciones. Circuitos: semisumador, sumador total.*
UC: *Su misión en la operación de la computadora. Operaciones. Sus componentes y funcionamiento. Temporización de las señales de control. Buses y sus usos. Procesos de transferencia y de proceso.*

SUMADORES BÁSICOS

Los sumadores son muy importantes no solamente en las computadoras, sino en muchos tipos de sistemas
digitales en los que se procesan datos numéricos. Comprender el funcionamiento de un sumador
básico es fundamental en el estudio de los sistemas digitales. En esta sección se presentan el semisumador
y el sumador completo.

El semi-sumador

Recordemos las reglas básicas de la suma binaria:

0 ¡ 0 ¢ 0

0 ¡ 1 ¢ 1

1 ¡ 0 ¢ 1

1 ¡ 1 ¢ 10

Todas estas operaciones se realizan mediante un circuito lógico denominado ***semi-sumador***.
Un semi-sumador admite dos dígitos binarios en sus entradas y genera dos dígitos binarios en sus salidas: un bit de suma y un bit de acarreo.

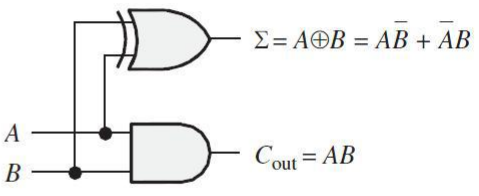
Lógica del semi-sumador. A partir del funcionamiento lógico de un semi-sumador, las expresiones correspondientes a la suma y al acarreo de salida se pueden obtener como funciones de las entradas. Observe que la salida de acarreo (C_{out}) es 1 sólo cuando A y B son 1; por tanto, C_{out} puede expresarse
como una operación AND de las variables de entrada.

A	B	C_{OUT}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Σ = suma
C_{out} = acarreo de salida
A y B = variables de entrada (operandos)

Tabla de verdad de un semi-sumador.

Diagrama lógico de un semi-sumador.



Observe ahora que la salida correspondiente a la suma es 1 sólo si las variables A y B son distintas.

Por tanto, la suma puede expresarse como una operación OR-exclusiva de las variables de entrada.

A partir de las ecuaciones se puede desarrollar la implementación lógica del funcionamiento
de un semi-sumador. La salida de acarreo se produce mediante una puerta AND, siendo *A* y *B* sus dos entra das, y la salida de la suma se obtiene mediante una puerta OR-exclusiva,

Recuerde que la operación OR-exclusiva se implementa con puertas AND, una puerta OR e inversores.

El sumador completo

El segundo tipo de sumador es el ***sumador completo*.Un sumador acepta dos bits de entrada y un acarreo de entrada, y genera una salida de suma y un acarreo de salida.**

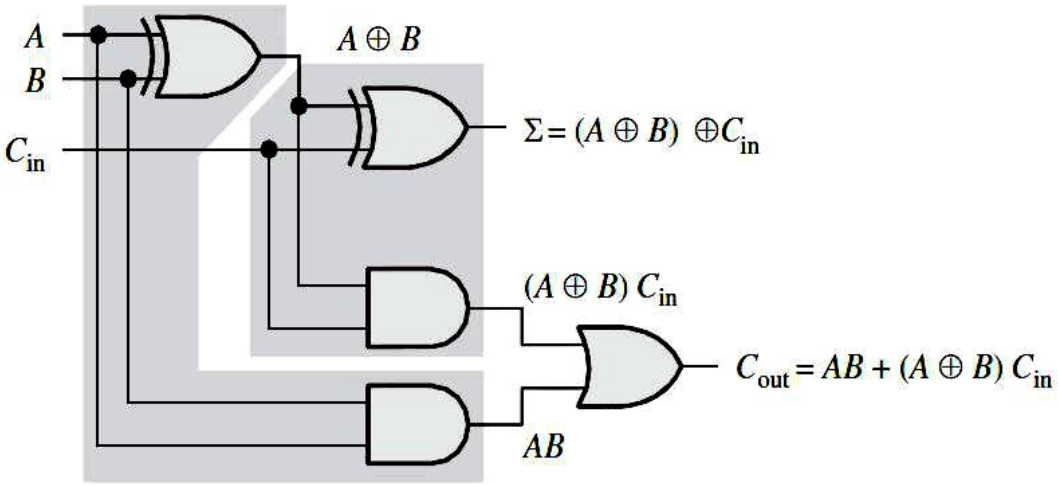
La diferencia principal entre un sumador completo y un semi-sumador es que el sumador completo acepta
un acarreo de entrada.

Lógica del sumador completo. El sumador completo tiene que sumar dos bits de entrada y un acarreo de
entrada. Del semi-sumador sabemos que la suma de los bits de entrada *A* y *B* es la operación OR-exclusiva de esas dos variables. Para sumar el acarreo de entrada (C_{in}) a los bits de entrada, hay que aplicar de nuevo la operación OR-exclusiva, obteniéndose la siguiente ecuación para la salida de suma del sumador completo:

Tabla de verdad de un sumador completo.

A	B	C_{in}	C_{OUT}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_{in} = acarreo de entrada. Algunas veces se designa como *CI*.
C_{out} = acarreo de salida. Algunas veces se designa como *CO*.
Σ = suma
A y *B* = variables de entrada (operandos)



Circuito lógico de un sumador completo (cada semi-sumador se representa por un área sombreada)

REGISTROS DE LA CPU:

IP: Puntero de Instrucciones (también llamado PC o CP "contador de programa"). Es un registro que permite localizar la próxima instrucción a ejecutar.
AX: Registro acumulador. Almacena valores y resultados de operaciones realizadas por la UAL.
RI: Registro de Instrucción. Almacena la instrucción que se está ejecutando.
RDI: Registro de Direcciones (MAR: *Memory Address Register*). Allí se forman las direcciones, que serán enviadas por las líneas de dirección (bus de direcciones) donde se va a escribir en memoria o leer desde ella. El número de direcciones que se pueden direccionar con una CPU depende del tamaño del MAR. Si el MAR tiene n bits de tamaño entonces se podrán direccionar un máximo de 2n palabras.
RDA: Registro de datos (MDR: *Memory Data Register*). Almacena los datos que se van a escribir o los que se leyeron de la memoria.
REGISTRO DE ESTADO: Formado por los "flags" o "indicadores" asociados a la UAL.
REGISTROS DE USO GENERAL: Utilizados por la UAL para almacenar diversos datos.

UNIDAD ARITMÉTICO LÓGICA[1]

- Realiza operaciones con datos de acuerdo al programa en curso

Operaciones — elementales — sumador / restador
Complejas — pasos elementales

↓

Velocidad

Unidad Control

[1] Arquitectura de computadores - J. M. Angulo - Editorial Paraninfo (Páginas 85 a 100)

Envía la información a procesar a la Unidad Aritmético Lógica, envía el código que selecciona la operación a realizar.

Unidad Aritmético Lógica (Estructura)

- Uno o más operadores — circuitos electrónicos que realizan una función aritmética o lógica.
- Banco de registros — almacenamiento de datos (8 a 16 dígitos)
- Acumuladores — depósito del resultado que origina el operador soporta información en numerosas operaciones
- Señalizadores de estado — señalizan ciertas condiciones de la última operación realizada por la unidad aritmética.
  - Z — Cero se pone a 1 si el resultado fue 0
  - N — Negativo se pone a 1 si el resultado fue negativo
  - C — Acarreo se pone a 1 si tiene acarreo
  - V — Desbordamiento se pone a 1 si el resultado no cabe en el lugar que le corresponde (overflow-sobrepasamiento)

Secuenciador — (Componente de la UC) A veces una instrucción se descompone en varias operaciones elementales. Este genera señales apropiadas para desarrollar las distintas microinstrucciones que conforman la instrucción.
Para cada instrucción se indica el operador que interviene, la operación a efectuar y los registros que participan.(Ver figura página 4)

Operadores

Circuitos eléctricos que realizan una o varias operaciones lógicas o aritméticas, tales como AND, OR, XOR, Suma, Resta.

- Generales — Realizan distintos tipos de operaciones.
- Especializados — Restringen a una operación muy determinada, por ejemplo, sumas y restas con coma flotante. Los computadores generales tienen un solo operador.
- Combinacionales — Suministran una salida que es función del estado de las entradas**.** No dispone de elementos de memoria. Tiempo igual a la suma de retardos de las puertas lógicas que deben atravesar las señales desde las entradas hasta la salida.
- Secuenciales — Requieren varias fases para obtener el resultado.

Tiene elementos de memoria para almacenar información a transmitirse entre fases.
Contador.
Definición de un algoritmo con el que se halla el resultado y contiene las etapas necesarias y las funciones que efectúa cada una de ellas.

Cualquier operación puede realizarse:

- Por circuito combinacional de una sola fase
- Por circuito secuencial que genera sus propias fases
- Por circuito secuencial cuyas fases las genera la Unidad de Control
- Mediante un programa.

- Paralelo — Realiza la operación correspondiente sobre todos los dígitos de los operandos.
- Serial — Trabaja dígito a dígito
  Es secuencial.
  Requiere tantas fases como dígitos

Operaciones de la Unidad Aritmético Lógica

- De desplazamiento
  Corrimiento de bits de una palabra, dato, registro a derecha o izquierda.
  .
  - Computadores sencillos — desplazamiento a derecha o izquierda de una posición
  - Modernos — múltiples en ambos sentidos

Desplazamientos lógicos.

- Los vectores de los extremos se completan con ceros
- Pérdida de información
- A veces se rellena con uno

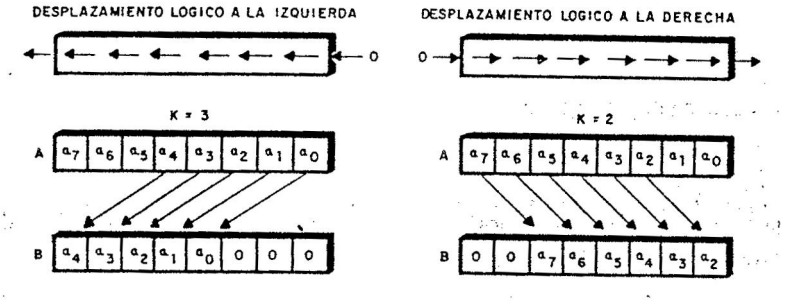


Fig. 3-5. Expresión gráfica de los desplazamientos lógicos a izquierda y derecha con ejemplos de aplicación.

Desplazamientos circulares

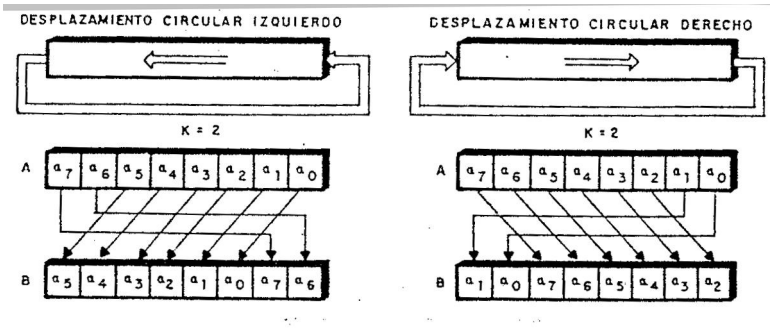


Fig. 3-6. Representación gráfica de los desplazamientos circulares en ambos sentidos, con ejemplos de aplicación.

Desplazamientos aritméticos

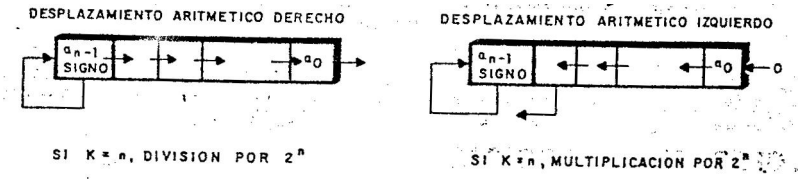


Fig. 3-7. En los desplazamientos aritméticos se mantiene invariable el bit de signo y las posiciones sobrantes se rellenan con ceros, lo que equivale a dividir o multiplicar el valor inicial por una potencia de 2. En la división se rellenan a la izquierda con el bit de signo.

- Invariabilidad del bit del signo
- Relleno de posiciones sobrantes con 0

↓

División o multiplicación del valor inicial por una potencia de 2

- Para multiplicar por potencias de 2 un número negativo, representado en complemento a 1, se deben introducir “unos” por la derecha.

Desplazamientos concatenados.

Son desplazamiento que afectan a un conjunto concatenado de dos o más elementos.

A. Dos registros
B. Un registro con el biestable de acarreo
C. Un registro con el biestable de signo

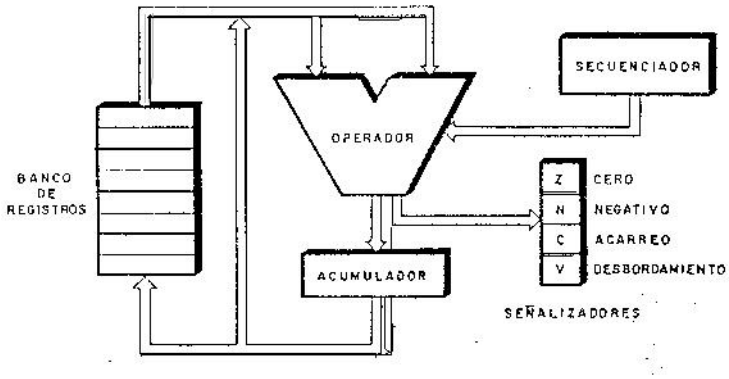


Fig. 3-1. Estructura general por bloques de la Unidad Aritmética.

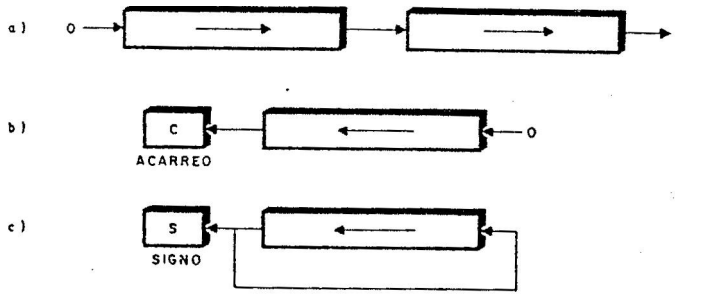


Fig. 3-8. Ejemplos de desplazamientos concatenados: a) sobre dos registros, b) un registro con el señalizador de acarreo y c) un registro con el señalizador de signo.

EJEMPLO DE DESPLAZAMIENTO ***(en este ejemplo se nombra como desplazamiento a la derecha – SHR- cuando se agrega un cero a la derecha, o sea, a la inversa de lo descripto anteriormente. De todas maneras, lo importante es comprender el concepto de para qué se producen operaciones de desplazamiento)****:*

¿Para qué se necesita una operación de desplazamiento de bits?

Ya que las operaciones de multiplicación y división dentro del microprocesador consume mayor tiempo de procesamiento, por lo tanto, cuando se realizan varios cálculos que conllevan multiplicación y división es más factible realizar calculos con desplazamiento hacia la izquierda o derecha (SHL o SHR – shift right o shift left) para no realizar calculos con MUL y DIV (operaciones de multiplicación y división). Para ello se expone un ejemplo fácil de comprender:

Supongamos que:
Deseo multiplicar: 25 x 2 = 50

- *25 en decimal es: 11001 en binario*
- *50 en decimal corresponde con: 11001**0** en binario*

Puede observar ambos valores en binario. ¿Qué notó de diferencia?
*Pues, como observó se le agregó un cero a la DERECHA 11001**0***
Entonces para operaciones de multiplicación necesita desplazar hacia la derecha n veces como sea necesario, por lo tanto, las operaciones corresponden con SHR.
Para simplificar la explicación se puede decir, que para dividir necesita desplazar hacia la izquierda, que corresponde con la operación SHL.

Importante la aclaración de que las operaciones de desplazamiento se realizan con el segundo operando que serán sobre 2^n (potencias de base 2 con los valores: 2 ,4 , 8, 16, 32, …)

En binario:

	operando 1	***operacion***	***operando 2***	***resultado***
◦	*11001*	*shr*	*10*	*110010*
◦	*110010*	*shl*	*10*	*011001*
◦	*11001*	*shr*	*100*	*1100100*

En decimal:

	operando 1	***operacion***	***operando 2***	***resultado***
◦	*25*	*shr*	*2*	*50*
◦	*50*	*shl*	*2*	*25*
◦	*25*	*shr*	*4*	*100*

Resumen:

- ***multiplicar**: numero shr 2^n ———> desplazar hacia la derecha*
- ***dividir:** numero shl 2^n <——— desplazar hacia la izquierda*

Operaciones Lógicas

A cada bit del operando origen se le aplica una función lógica, independiente de los restantes bits, lo que significa que el resultado de aplicar un operador lógico a un bit no afecta el resultado de los demás bits.

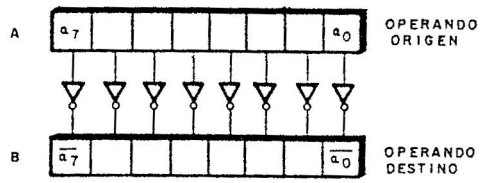


Fig. 3-9. Forma gráfica de realizar la función lógica inversión. Se trata de una operación monádica, porque sólo necesita un operando para producir el resultado.

Operaciones lógicas comunes

- Negación o inversión lógica ($\overline{A}$) (*monádica*)
- Suma lógica u OR (A + B) (*diádica*)
- Producto lógico o AND (A · B)
- XOR, EOR o OR Exclusiva. ($A \oplus B$)

- NOR
- NAND

Operaciones Aritméticas

Cambio de signo

Representación en binario puro	→	Modificación del signo
Representación en complemento a uno	→	Negación lógica
Complemento a dos	→	Negación lógica + suma de unidad

Extensión de signo

Cuando se traslada un valor numérico a la memoria, a un registro, a un bus, a un operador de mayor número de bits, los bits sobrantes deben rellenarse de forma que no cambie el significado.

- Binario c/signo — El bit del signo se traslada al bit de mayor peso del destino, rellenando 0 los bits sobrantes.

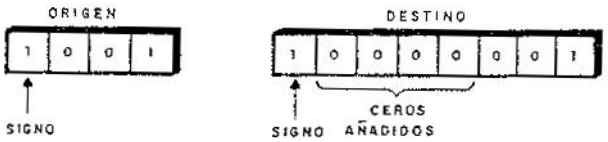


Fig. 3-12. Extensión de signo para números binarios con signo.

- Complemento a uno o a dos — Los bits sobrantes de más peso se rellenan con el mismo valor del bit del signo.

Fig. 3-13. Ejemplos de extensión de signo en números expresados en complemento a uno o a dos.

Adición y sustracción

Similares	
Resta	suma si se toma el sustraendo en forma de complemento a dos
Suma	más importante

Sumador elemental

Operador consistente en un circuito combinacional capaz de sumar dos dígitos binarios más el posible acarreo previo de la etapa anterior produciendo el dígito suma y el de acarreo para la siguiente (ver *sumador completo*)

Sumador paralelo: Consta de tantos sumadores elementales encadenados, como bits tengan los operandos.

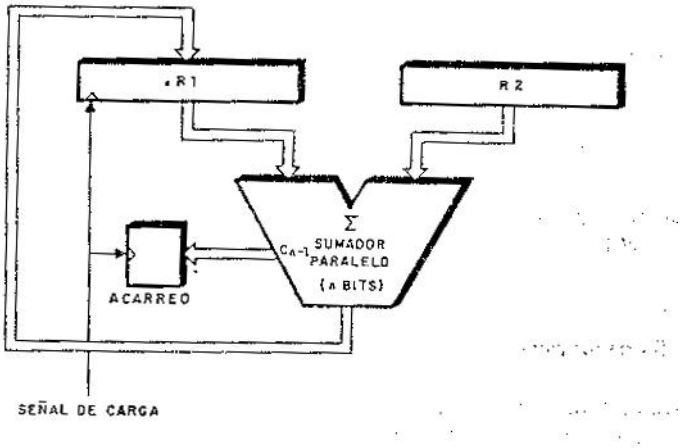


Fig. 3-18. Esquema de utilización práctica del sumador paralelo.

Sumador serie: Un único sumador elemental va sumando cada pareja de bits de los registros que contienen los sumandos.

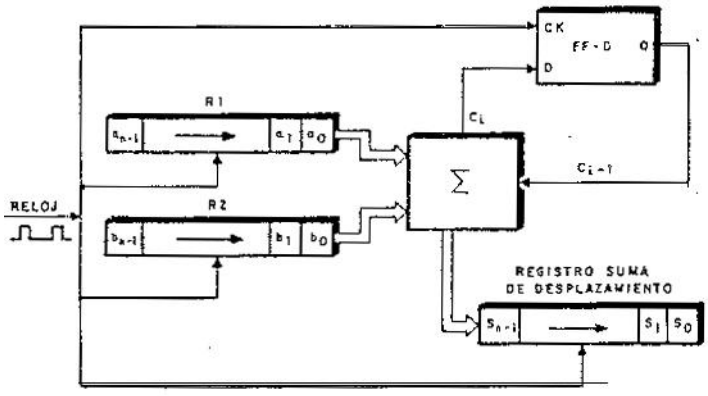


Fig. 3-19. Esquema de un sumador serie.

Restador paralelo

Aunque se pueden diseñar restadores elementales, se suele añadir al sumador un circuito que permita llevar a cabo la suma y la resta, seleccionando la operación a efectuar mediante una señal de control, llamada S/R, que se suma si S/R = 0

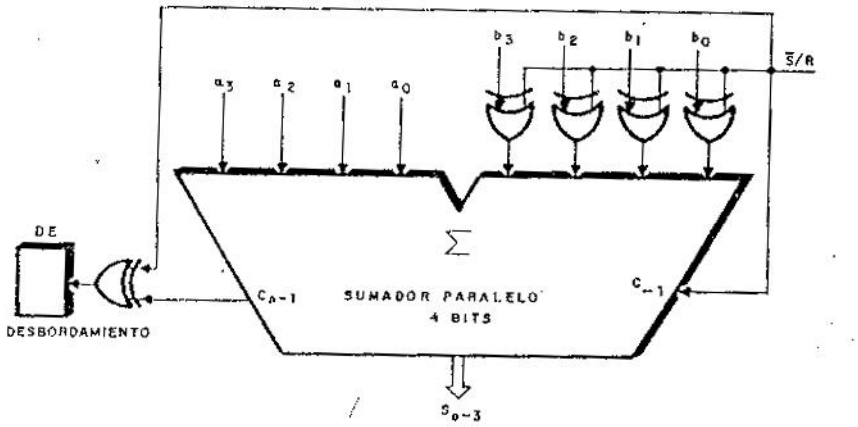


Fig. 3-20. Circuito sumador-restador paralelo. Para que efectúe la resta es necesario que $\overline{S}/R = 1$. En caso contrario, se suman los dos operandos.

Multiplicación[2]

Su fundamento se basa en el método manual de la multiplicación.

Ejemplo

A : 1 0 1 0 1 1

B : 0 0 1 0 1 1

1 0 1 0 1 1	Al encontrar un 1 en B se suma A
1 0 1 0 1 1	Se suma A desplazado, al encontrar un 1 en B
0 0 0 0 0 0	Se suma 0 desplazado, al encontrar un 0 en B
1 0 1 0 1 1	Se suma A desplazado, al encontrar un 1 en B
0 0 0 0 0 0	Se suma 0 desplazado, al encontrar un 0 en B
0 0 0 0 0 0	Se suma 0 desplazado, al encontrar un 0 en B
0 0 1 1 0 1 1 1 0 0 1	Al encontrar un 1 en B se suma A

[2] Arquitectura de computadores - J. M. Angulo - Editorial Paraninfo (Páginas 116 a 119)

Esquema lógico de un multiplicador que usa el algoritmo de suma de desplazamiento. S es la señal que activa R1 para que la suma se cargue en él, cuando $b_1 = 1$. DP indica a B que debe rotar ya a R_1, R_2 y C, que también deben hacerlo. La señal S introduce R_1 a la entrada del sumador, cuando $b_0 = 1$; cuando $b_0 = 0$ no se hace nada.

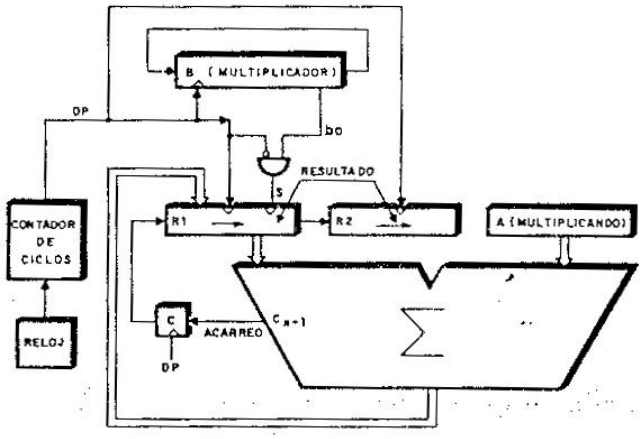


Fig. 3-31. Esquema lógico de un multiplicador que usa el algoritmo de suma desplazamiento. S es la señal que activa R1 para que la suma se cargue en él, cuando $b_1 = 1$. DP indica a B que debe rotar y a R1, R2 y C, que también deben hacerlo. La señal S introduce R1 a la entrada del sumador, cuando $b_0 = 1$; cuando $b_0 = 0$ no se hace nada.

.

División[3]

Ejemplo de mecánica para efectuar la división
Mentalmente se comprueba si el dividendo o dividendo parcial "cabe" dentro del divisor. Si "cabe", se introduce un "1" al cociente y si no, un "0", desplazando una posición a la izquierda el registro que guarda el cociente, si "cabe" (cociente = 1), al dividendo o dividendo parcial se le resta el divisor desplazado a la izquierda totalmente, obteniéndose otro dividendo parcial.

```
                    10110100 | 1001
                  - 1001       10100
Dividendo parcial →  ▒0100100
                  -  0000
Dividendo parcial →  ▒100100
                  -  1001
Dividendo parcial →  ▒00000
                  -  0000
Dividendo parcial →  ▒0000
                     0000
                     0000
```

[3] Arquitectura de computadores - J. M. Angulo - Editorial Paraninfo (Páginas 122 a 123)

Diagrama lógico de un divisor

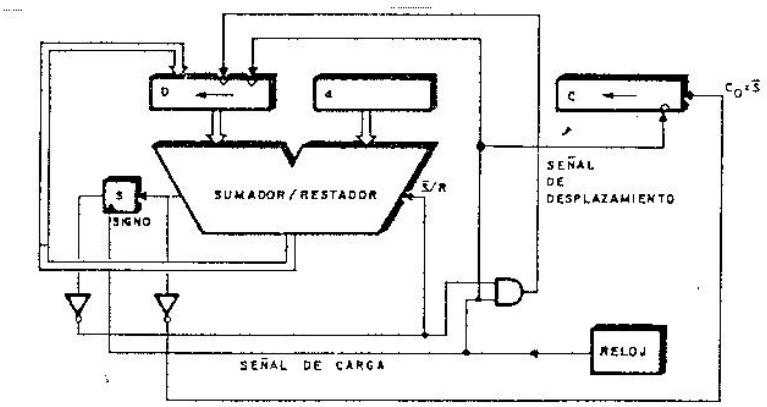


Fig. 3-35. Diagrama lógico de un divisor.

UNIDAD DE CONTROL

Recordamos:

Los bloques componentes de la computadora (CPU, memoria, unidades de E/S) están comunicados entre sí mediante conjuntos de líneas que transportan información binaria del mismo tipo. Son los colectores o **buses**.

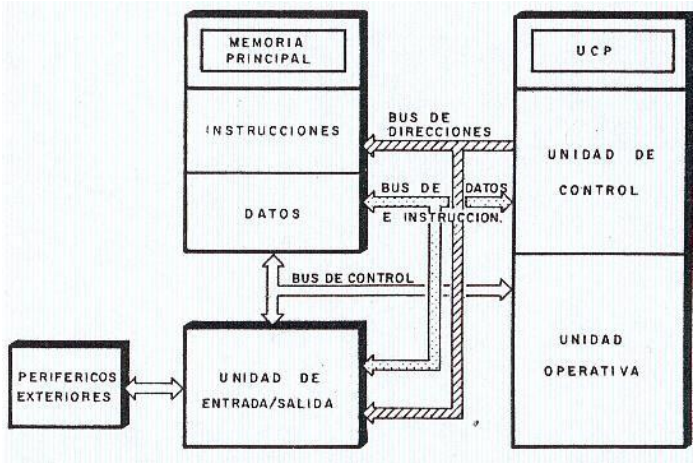
o El colector de líneas digitales es el **bus de direcciones.** Es de carácter unidireccional porque su contenido siempre forma parte de la unidad de control

o Al estar formado por n líneas, tendrá acceso a 2^n posiciones diferentes

o El **bus de datos** es el encargado de transferir información. Es bidireccional y triestado, pues es compartido por todos los elementos del computador. La unidad de control establece cuál es el elemento emisor y cuál actúa de receptor.

o El número de líneas del bus de datos es el que define la palabra de trabajo del computador.

o El **bus de control** es el colector de líneas que transporta señales auxiliares de gobierno y sincronización. Transporta los impulsos de reloj, las señales que indican si la operación es lectura o escritura, las de habilitación de los buffers y registros, las de inicialización, etc.



g. 6-1. Representación de los bloques fundamentales del computador interconectados a través de los buses.

Unidad de control

Interpreta y controla la ejecución de las instrucciones recibidas desde la memoria principal.

Recibe, a través del bus de datos, el código binario de la instrucción en curso, que lo registra convenientemente. Después el decodificador de instrucciones selecciona las posiciones de una memoria ROM, llamada *Memoria de Control*, que corresponde a dicha instrucción y en las que se hallan grabados los códigos de las señales que controlan cada uno de los pasos elementales en que se descompone la instrucción y que reciben la denominación de *microinstrucciones*.

El *secuenciador* se encarga de sacar y distribuir a los elementos del sistema las correspondientes señales de control de cada microinstrucción y, de esta manera, ejecutar ordenadamente la instrucción en curso.

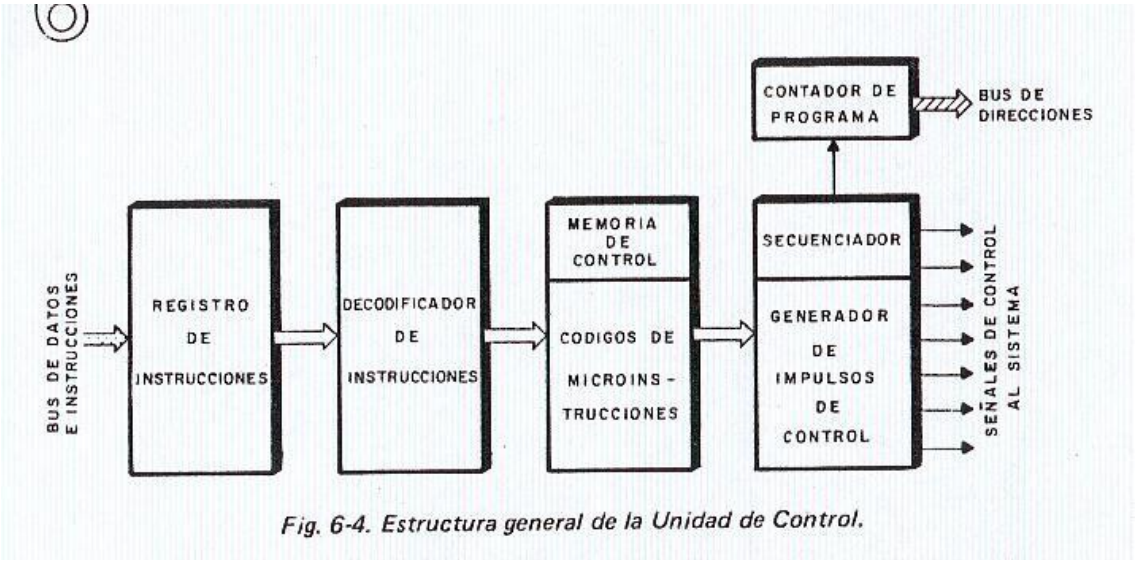


Fig. 6-4. Estructura general de la Unidad de Control.

Dentro de la Unidad de Control se encuentra el Contador de Programa (PC) encargado de mandar por el bus de direcciones la posición de la memoria donde se encuentra la siguiente instrucción. Aunque corrientemente se incrementa en 1, existen instrucciones de bifurcación que permiten variar su contenido de forma diferente, dando lugar a rupturas de programa y toma de decisiones de acuerdo con los resultados que se van obteniendo.

La Unidad de Control resuelve situaciones anómalas o de conflicto que pueden ocurrir en el computador.

Utiliza la siguiente información:

- Instrucción
- Registro de estado con sus señalizadores

  Contiene información sobre
  Resultado de la operación anterior
  Posibles situaciones anómalas o especiales
  Desbordamiento
  Interrupciones
  Errores de paridad

Estas situaciones exigen una acción determinada por parte de la Unidad de Control
La información sobre el estado del procesador se usa para hacer rupturas condicionales en la secuencia del programa, ya sea por medio de instrucciones o por interrupciones externas o situaciones de error.

- Contador de períodos
- Señales de Entrada /Salida
- Proceso: El código de la operación (OP) de la instrucción indica a la Unidad de Control la operación que se debe ejecutar y el modo de direccionamiento utilizado.

Se localizan los operandos
Se mandan a la UAL
Se almacena el resultado

Un ejemplo simple de ejecución de una instrucción luego de decodificada: Una instrucción de suma de dos registros R1 y R2, guardando su resultado a R1.

Luego de decodificada la instrucción, se realiza lo siguiente:

- Se envía al mux de entrada de la ALU las salidas de los registros R1 y R2.
- Se instruye a la ALU que se trata de una operacion de suma.
- La salida de la ALU, a través de otro mux, es conectada a la entrada de carga de datos del registro R1, y se activa también la entrada de control de R1 que habilita la carga de datos.
- Las salidas de la ALU de acarreo, overflow, zero, neg., cargan los correspondientes bits en los flags.

Operaciones elementales y microinstrucciones

Operación elemental: la ejecución de cada instrucción se descompone en una serie ordenada de pequeños pasos.

EJ. lectura de un operando
Incremento del Contador de Programa
Ejecución de una operación aritmética
Suma de la base más el desplazamiento para hallar el valor efectivo de una dirección de memoria

La ejecución de cada operación elemental requiere la activación de un conjunto de señales de control, que genera la Unidad de Control a través de su Secuenciador. En cada ciclo de máquina, generado por un reloj, el Secuenciador envía una serie de señales de control que realizan una o varias operaciones elementales, que configuran una microinstrucción. Una instrucción consta de varias microinstrucciones y cuando se realiza la última de una instrucción, la Unidad de Control se prepara para recibe el código de operación de la siguiente instrucción del programa y comenzar la ejecución de sus respectivas microinstrucciones.

Tipos de operaciones elementales:

Operaciones de transferencia: se comienza seleccionando los elementos que participan (posiciones de memoria o registros), uno de los cuales actúa como origen y otro como destino. Luego se establece un camino de comunicación entre las salidas del origen y las entradas del destino. Finalmente se envía al destino una señal para que se "cargue" con la información que existe en su entrada.
Operaciones elementales de proceso: Tienen un planteamiento básico idéntico a las de transferencia
La información de origen se hace pasar por un operador en su camino hacia el destino. En caso de que la operación sea diádica, los dos orígenes u operandos confluyen en el operador que produce el resultado

Señales de control
Cada uno de los bloques principales del computador son gobernados por las diferentes señales de control.

Temporización de las señales de control.
Períodos y fases
Reloj

En los sistemas digitales, todas las señales se sincronizan con una señal de temporización básica denominada reloj. El reloj es una señal periódica en la que cada intervalo entre impulsos (el período) equivale a la duración del bit.

Los computadores poseen un funcionamiento síncrono, gobernado por un oscilador o reloj. Los flancos de este reloj representan la temporización básica del sistema, puesto que determinan el menor tiempo que puede durar una operación elemental.
Se entiende como **período** la duración de un tiempo elemental determinado por el reloj.

Cada **sentencia** (orden) de un programa en alto nivel es traducida por un programa compilador en una secuencia de **instrucciones** (órdenes más simples que una UC puede ejecutar.

A su vez, la ejecución de cada instrucción se divide en **pasos** aún más simples
Las acciones que debe ordenar/controlar la UC en cada uno de estos pasos están determinadas por combinaciones binarias llamadas "**microcódigos**" que van apareciendo una tras otra en las **líneas de control** con cada uno de los pulsos que constituyen los Mhz
Estas líneas salen de la UC hacia la UAL, los registros y la memoria . La ejecución de una instrucción implica una secuencia de movimientos de transferencia de bytes entre memoria principal y registros de la UCP (o entre estos últimos), establecidos por la UC en un orden determinado, según el código de dicha instrucción. También puede ordenarse una operación en la UAL

Cada segundo puede ejecutarse algunos millones de instrucciones, para lo cual deben sucederse muchos *millones* de estos movimientos de pasaje de direcciones, códigos, datos y resultados, *al ritmo de millones de impulsos eléctricos por segundo* ("**megahertz**", abreviados **MHz**[1]) que le llegan a la UC, generados regularmente por un cristal piezo-eléctrico de cuarzo o "**reloj**" ("*clock*").
Así se habla de microprocesadores (con reloj) de 100 MHz, 1 GHz, etc. En principio, a mayor número de MHz podrán suceder más de estos movimientos por segundo, con lo cual se podrán ejecutar más instrucciones por segundo. Un Pentium de 1 GHz puede ejecutar más de mil millones de instrucciones por segundo (1000 MIPS), y en ciertos casos hasta 3000 MIPS.

La ejecución de una instrucción de máquina, de tipo general se subdivide en cuatro fases:
1. lectura del código de la instrucción. Fase de búsqueda o fetch}
2. lectura de los operandos (datos) y decodificación de la instrucción

3. ejecución de la operación
4. almacenamiento del resultado

Cada fase puede necesitar de uno o varios períodos.
El objetivo de dividir las instrucciones en fases reside en alcanzar una sistematización de su tratamiento para simplificar el diseño de la Unidad de Control.

Cronogramas o diagramas de tiempos
Es una gráfica de formas de onda digitales que muestran la relación temporal real entre dos o más señales, y cómo varía cada señal en relación con las demás

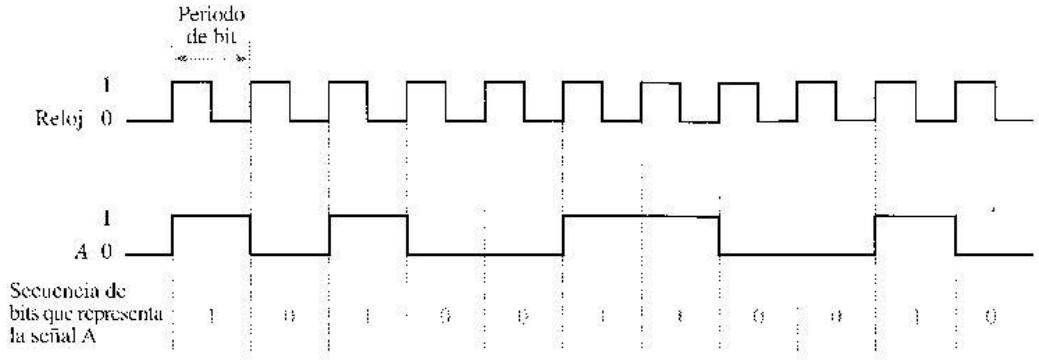


FIGURA 1.10
Ejemplo de una señal de reloj sincronizada con una señal que representa una secuencia de bits.

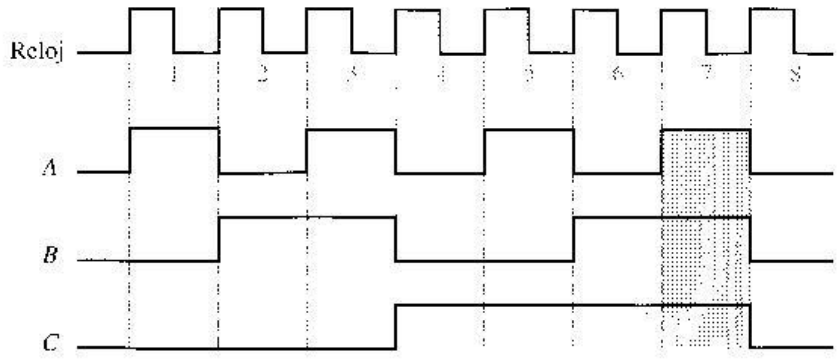


FIGURA 1.11
Ejemplo de un cronograma o diagrama de tiempos.

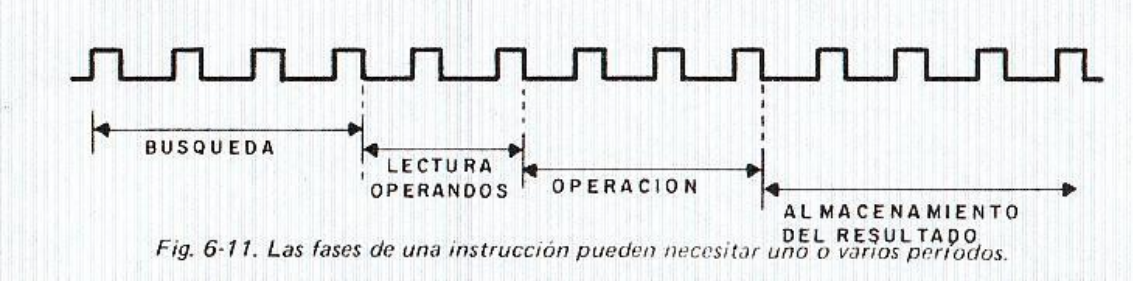


Fig. 6-11. Las fases de una instrucción pueden necesitar uno o varios períodos.

...ón los MIPS y las MFLOPS ?

Un procesador puede ejecutar millones de instrucciones por segundo. Estas últimas palabras se abrevian con la sigla **MIPS** . Este dato puede servir relativamente para comparar, a una misma frecuencia de operación, la performance de un mismo procesador o de distintos procesadores entre sí, ejecutando instrucciones de tipo semejante, siendo que los MIPS de un mismo procesador varían de un programa a otro. La ejecución de distintos tipos de programas es siempre la mejor medida de comparación.

Un Pentium de 1 Ghz en promedio puede terminar de ejecutar hasta 3 instrucciones por ciclo, lo cual extrapolando implicaría unos 3000 MIPS

Dados los requerimientos actuales de las operaciones en punto flotante, la evaluación de la performance de procesadores para las mismas, pasó a ser importante. Los **MFLOPS** –pronunciados "megaFLOPS"– (del inglés *mega floating points operations per second*) son las millones de operaciones en punto flotante por segundo que puede realizar un procesador, calculados como el cociente:

megaflops/megaofps	10^6 *flops*
gigaflops/gigaofps	10^9 *flops*
teraflops/teraofps	10^{12} *flops*
petaflops/petaofps	10^{15} *flops*
exaflops/exaofps	10^{18} *flops*
zettaflops/zettaofps	10^{21} *flops*
yottaflops/yottaofps	10^{24} *flops*

Multiplexor

Circuito lógico que pasa los datos digitales procedentes de varias líneas de entrada a una única línea de salida según una secuencia de tiempos específica. Se puede representar mediante una operación de conmutación electrónica que secuencialmente conecta cada una de las líneas de entrada a la línea de salida.

Demultiplexor

Circuito que pasa los datos digitales procedentes de una línea de entrada a varias líneas de salida según una determinada secuencia de tiempo. Por lo tanto el demultiplexor es un multiplexor invertido.

La multi y demultiplexación se emplean cuando datos procedentes de distintas fuentes se tienen que transmitir a través de una línea hasta una localización distante, y deben redistribuirse a varios destinos.