Resumen Segundo Parcial

Arquitectura De Computadores (Universidad Argentina de la Empresa)

Direccionamiento: los campos de direcciones son un formato de instrucciones usual que esta bastante limitado. Existen distintas técnicas de direccionamiento que implican algún compromiso entre el rango de direcciones y/o flexibilidda de direccionamiento de una parte y, por otra, el numero de referencias a memoria y/o la complejidad de calculo de direcciones. Los modos mas comunes de direccionamiento son:

- **Inmediato:** Un dato de 8 o 16 bits se especifica como parte de la instrucción. Puede usarse para definir y utilizar constantes o para fijar valores iniciales de variables. La ventaja es que una vez captada la instrucción, no se requiere una referencia a memoria para obtener el operando, ahorrándose un ciclo de memoria o de cache en el ciclo de instrucción. La desventaja es que el tamaño del numero esta restringido a la longitud del campo de direccionamiento.
- **Directo:** el campo de direcciones contiene la dirección del operando. Requiere una referencia a memoria y no necesita ningún calculo especial. La dirección efectiva (EA) de 16 bits se toma directamente del campo de desplazamiento de la instrucción. El valor constante se tiene que buscar en memoria, en la dirección especificada relativa al DS. Es más lento que los anteriores, pero es el más rápido para ir a memoria, pues ya "sabe" la dirección, la toma de la instrucción y no la tiene que calcular.
- **Indirecto:** La dirección efectiva (EA) está especificada en un registro puntero o un registro índice. El puntero puede ser el registro base BX o el base BP; el registro índice puede ser el Índice Fuente (SI) o el Índice Destino (DI). El operando indica una ubicación de memoria, cuya dirección (sólo la parte desplazamiento) está en SI o en el DI. Es más lento que los anteriores, pues tiene que "calcular" la ubicación. También existe una variante de este que es direccionamiento a multinivel o en cascada.
- **De registros**: es similar al directo. Se diferencia en que el campo de direcciones referencia un registro en un lugar de una dirección de memoria principal. Especifica el operando fuente y el operando destino. Los registros deben ser del mismo tamaño. Usa solamente registros como operandos, es el más rápido.
- **Indirecto con registros**: es análogo al direccionamiento indirecto y análogo al directo. La diferencia estriba en si el campo de direcciones hace referencia a una posición de memoria o a un registro.
- **Con desplazamiento**: nodo potente de direccionamiento que combina las posibilidades de los direccionamientos directo e indirecto con registro. Necesita que las instrucciones tengan dos campos de direcciones (una explicita). El valor de uno de los campos se usa directamente, el otro se refiere a un registro, cuyo contenido se le suma a A para generar al dirección efectiva.
    - Relativo: El registro referenciado implícitamente es el contador de programa (PC), es decir, la dirección de instrucción actual se suma al campo de direcciones para producir el valor EA. La dirección efectiva es un desplazamiento relativo a la dirección de la instrucción. En este modo el operando se especifica como un desplazamiento de 8 bits con signo, relativo al PC (program counter).
    - Con registro base: el registro referenciado contiene una dirección de memoria y el campo de dirección contiene un desplazamiento desde dicha dirección. La diferencia a registro puede ser explicita o implícita.
    - Indexado: el campo de dirección referencia una dirección de memoria principal, y el registro referenciado contiene un desplazamiento positivo desde esa dirección. Puede ser pre-indexado o post-indexado.
- **De pila:** forma de direccionamiento implícito. Las instrucciones maquina no necesitan incluir una referencia a memoria si no que oeran implícitamente con la cabecera de la pila.

Interrupciones: Proporcionan una forma de mejorar la eficiencia del procesador, el procesador puede dedicarse a ejecutar otras instrucciones mientras una operación de E/S esta en curso. Después de q estas instrucciones se ejecutaron se devuelve el control al usuario, cuando el dispositivo externo pasa a estar listo para aceptar mas datos del procesador, el modulo de E/S envía una señal de “petición de interrupción” al procesador, el cual suspende la operación que estaba ejecutando y salta a un programa (gestor de interrupción) que da servicio a ese dispositivo y desp sigue con la ejecución del programa original. Es una interrupción en la secuencia normal del funcionamiento, cuando el procesamiento de la interrupción se completa, la ejecución sigue.
En el ciclo de interrupción el procesador comprueba si sucedió alguna interrupción, si no hay el procesador sigue y si hay suspende el programa en curso guardando su contexto y ejecuta la interrupción.
Pueden existir interrupciones múltiples y se pueden seguir dos caminos uno teniendo en cuenta la importancia de los eventos y otra por orden de llegada.
Las interrupciones se dividen en dos tipos:

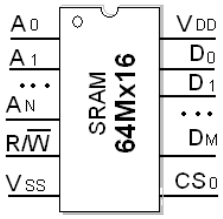
- **Externas:** provocada por un dispositivo externo al procesador. Las dos líneas que pueden señalar interrupciones externas son la líne interrupción (INTR). La línea

NMI reporta la memoria y errores de paridad de E/S. El procesador siempre actúa sobre esta interrupción, La línea INTR reporta las peticiones desde los dispositivos externos.

- **Internas:** resultado de la ejecución de una instrucción INT o una operación de división que cause desbordamiento, ejecución en modo paso a paso o una petición para una interrupción externa, tal como E/S de disco. Los programas por lo común utilizan interrupciones internas, que **no son enmascarables**, para accesar los procedimientos del BIOS y del DOS. El BIOS contiene un extenso conjunto de rutinas de entrada/salida y tablas que indican el estado de los dispositivos del sistema. El DOS y los programas usuarios pueden solicitar rutinas del BIOS para la comunicación con los dispositivos conectados al sistema. El método para realizar la interfaz con el BIOS es el de las interrupciones de software.

Buses: Líneas que interconectan los elementos de un computador, camino de comunicación entre dos o mas dispositivos. Es un medio de comunicación compartido. Si los dispositivos transmiten por el bus al mismo tiempo se produce una colisión.
Algunos dispositivos conectados al bus son activos y pueden iniciar transferencias, mientras que otros son pasivos y esperan solicitudes. Los activos se llaman amos o maestros y los pasivos se llaman esclavos. En ninguna circunstancia la memoria es amo.
Los buses pueden ser únicos (considera a la memoria y a los periféricos como posiciones de memoria, y hace un símil de las operaciones E/S con las de escritura/lectura en memoria. Todas estas equivalencias consideradas por este bus, hacen que no permita controladores DMA (Direct Acces Memory ; de acceso directo a memoria), o dedicados. Estos últimos se dividen en 3 grupos:

- **Buses de datos**: transmite info entre la cpu y los periféricos. dan un camino para transmitir datos entre los modulos del sistema. Estos entran y salen de la memo, cpu y E/S.
- **Buses de direcciones**: identifica el dispositivo destino u origen de la información que se transmite por el bus de datos. Sale de la cpu y entra en las memo y E/S. (n = ln (bits)/ln2)
- **Buses de control:** Controla la transferencia de los buses, los dispositivos conectados al bus y los protocolos handshaking asociados al bus. Sale de la cpu entra a las memo y E/S. (r/W, ack, cs, irq, clock, reset).
- **Buses de alimentación:** Vss, Vdd.



Buses múltiples: se conectan un gran numero de dispositivos a los buses. La multiplicidad se debe a:

- ***Buses de legado*** *(Legacy bus)**: Buses antiguos que se deben seguir soportando*. Ej.: PCI, ISA, RS-232, Centronic, etc.
- ***Buses estándar*** *(Standard Bus)**: son buses que siguen estándares internacionales y populares*. Ej.: PCIe, SCSI, USB, Ethernet, etc.
- ***Buses propietarios*** *(Propietary Bus)**:* están diseñados a medida para maximizar la taza de transferencia de datos entre componentes de sistemas críticos.

Estos tienen cierras especificaciones en común:

- **Tasa de datos**: Cantidad de datos por unidad de tiempo que el bus puede transferir (varía desde pocos megabytes/segundo a varios gigabyte/segundo).
- ***Nro. máximo de dispositivos:*** Mas dispositivos aumentan la carga del bus, así como la cabecera de arbitraje, ello hace mas lento el bus.
- ***Método de conexión:*** Tiene que ver con el tipo de conector y cable que se debe usar para conectar al bus.
- ***Fortaleza y confiabilidad:*** Tiene que ver con conocer que tan bien se sigue comportando el bus ante la falla de un dispositivo conectado a él, o irregularidades eléctricas como cortocircuitos, que pueden ocurrir antes o durante la transmisión de datos
- ***Parámetros eléctricos:*** Éstos incluyen niveles de voltaje, rangos de corriente, protecciones de sobretensión y cortocircuito, intermodulación y requerimientos de energía.

- ***Cabecera de comunicación:*** Los bits de control y verificación de error se suman a los datos transmitidos para control, coordinación y confiabilidad mejorada.

Los buses también se pueden clasificar por líneas:

- **Dedicadas**: permanentemente asignada a una funcion o a un subconjunto fisico de componentes del computador .su dedicación puede ser física (conectan siempre el mismo subconjunto de modulos) o funcional (hacen siempre la misma funcion). Su venta es qe hay menos disputas por el acceso al bus. Su desventaja es su incremento en tamaño y precio
- **Multiplexada**: la utilizacion de la misma linea tanto para transmitir datos como direcciones utilizando una linea de control adicional. Su desventaja es que las operaciónes de transferencia no pueden ser realizadas en paralelo, lo que reduce las prestaciones. y como ventaja reduce su tamaño y precio.

ARBITRAJE DE BUS: Dado que en un instante dado solo una unidad puede transmitir a traves del bus, debe existir un metodo de arbitraje para tal fin. En ambos metodos de arbitraje el proposito es designar un dispositivo, el procesador o un modulo de entrada y salida como maestro del bus para que pueda asi iniciar la transferencia.Este puede ser:

- **Centralizado**: unico dispositivo denominado controlador de bus o arbitro es el responsable de asignar el control del bus. Este dispositivo puede estar separado o formar parte del procesador. Hay distintos tipo:
    - Prioridad fija: Le entrega la señal de garantia al solicitante de mayor prioridad. Su ventaja es la alta velocidad de respuesta y su desventaja es la iniciación de dispositivos de menor prioridad.
    - Prioridad rotativa: Proporciona tratamiento uniforme a todos los dispositivos. La ventaja es que no existe inanición de dispositivos. La desventaja es que existe respuesta mas lenta a los dispositivos de alta prioridad
    - Encadenamiento circular: Consiste en encadenar los dispositivos de manera que compartan las señales de solicitud y garantía. Asigna prioridad a los dispositivos que tienen mayor cercania al arbitro.

- **Distribuido**: cada modulo dispone de la lógica para controlar el acceso al bus y estos trabajan conjuntamente para compartirlo. Hay distintos tipo:
    - Metodo de autoseleccion: Cada dispositivo pone un codigo de prioridad en el bus y se selecciona el de mayor prioridad.
    - Metodo de deteccion de colision: Multiples dispositivos solicitan el bus, si hay actividad no acceden, de lo contrario acceden pero controlando que no haya colision.

TEMPORIZACION : hace referencia a la forma en que se coordinan los eventos en el bus.

- **Sincronica: l**a presencia de un evento esta determinada por una señal de reloj.El bus incluye una linea de reloj y todos los dispositivos pueden leer esta linea. Todos los dispositivos deben utilizar la misma frecuencia de reloj , por lo que el sistema se adapta al dispositivo mas lento, disminuyendo las prestaciones. Es que es mas facil de implementar.
- **Asincronica:** la presencia de un evento en el bus es consecuencia y depende de que se produzca un evento previo.La ventaja es que en un mismo sistema pueden coexistir dispositivos lentos y dispositivos rapidos sin necesidad de que los rapidos se adapten a los lentos, aumentando las prestaciones del sistema.

RENDIMIENTO DEL BUS:

- **Ancho del bus:** El ancho se define por el número de líneas del bus. Afecta directamente al desempeño del sistema. El ancho de bus de datos es el Nº de accesos a memoria y el del bus de direcciones es la forma de direccionar.
- **Mayor frecuencia de reloj:** Velocidad de sincronización y transferencia mas altas.
- **Línea de dirección y datos separadas:** no multiplexadas
- **Transferencias en bloque:** menos bytes dedicados a la cabecera del mensaje
- **Control fino del bus:** por ej. liberación del bus mientras se espera que ocurra algo (protocolo de transacción dividida).
- **Arbitraje de bus más rápido.**

Memorias: A diferencia de los circuitos combinacionales donde las salidas dependen unicamente de las entradas, la salida de los circuitos circuitos secuenciales dependen tanto de sus entradas como del valor previo de sus salidas y esto los convierte en ideales para fab

La diferencia entre latch y flip-flop radica en el metodo empleado para cambiar de estado.

Latch SR Los latch son dispositivos logicos de almacenamiento temporal de 2 estados

Latch SR con habilitación; Modificacion del latch SR que solo permite el cambio cuando la señal de reloj lo permite

Latch D: Una forma de resolver el problema que surge cuando S y R son 1 es evitar que pase y para esto

Métodos de acceso a las memorias:

- **Secuencial:** Iniciar al prinicipio y leer en orden. El tiempo de acceso depende de la localización de los datos y de dónde se localizaban previamente.
- **Directo:** bloques individuales, tienen direcciones únicas, El acceso se hace mediante un acceso directo a una vecindad dada, El tiempo de acceso es variable.
- **Random**: Direcciones individuales identifican posiciones exactas. El tiempo de acceso es independiente de la posición o acceso previo.(RAM)
- **Asociativa:** Los datos se localizan por una comparación con los contenidos de una porción del almacenamiento. El tiempo de acceso es independiente de la posición o acceso previo. Ejemplo: Memoria caché

El rendimiento de la memoria depende de:

- **Tiempo de Acceso**: tiempo transcurrido entre presentar la dirección y obtener el dato
- **Tiempo del ciclo de memoria**: "Tiempo de acceso + Tiempo de recuperación"
- **Tasa de transferencia**: velocidad a la cual los datos pueden ser movidos

Registros: porción de memoria que guarda una palabra. Son el punto base sobre el cual trabajan los programas, cuando una instrucción se esta ejecutando dentro del procesador esta usando los registros del procesador (o sea fue traída de memoria, se cumplió todo el ciclo de instrucción pero para ejecutarlo tuvo que haber llevado los datos y todos los elementos que necesita la instrucción a los registros).

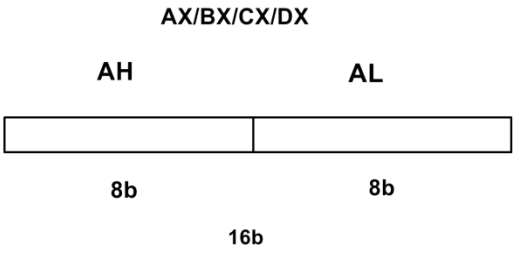
- **Registros de uso general** (AX, BX, CX y DX): más allá de que tienen una función específicamente asignada se pueden usar para cualquier cosa.
- **Registro acumulador** (AX): es donde se guarda el resultado de las operaciones manuales.
- **Registro base** (BX): se usa para operaciones de acceso a memoria entonces es la base del punto de acceso.
- **Registro Contador** (CX): es un registro contador porque mas allá que se puede sumar o restar o conectar la unidad aritmético lógica se usa para operaciones de transferencia, cuenta la cantidad de bytes que se están transfiriendo entre una memoria o un dispositivo de entrada o salida, generalmente trabaja con los dispositivos de E/S.
- **Registro de datos** (DX): opcional, generalmente se usa para guardar datos transitoriamente.
- **Registros de Segmentos** (DS, ES, SS, CS): se guardan direcciones de donde se encuentra en memoria los segmentos.
- **Registro segmento de código** (CS): es la mínima dirección donde arranca el rango de direcciones donde esta cargado el código ejecutable de un programa dentro de la memoria de la máquina.
- **Registro del segmento de pila** (SS): guarda la dirección de base del segmento de pila, es donde cada vez que se entra una interrupción el programa guarda el contenido de todos los registro para que cuando salga de la interrupción y termine de ejecutar el código de la subrutina o de la función pueda seguir con el programa principal.
- **Segmento de datos** (DS): guarda la dirección de base del segmento de memoria donde están guardados los datos que va a usar el programa.
- **Registro del segmento extra** (ES): es un segmento adicional que algunos programas lo pueden usar, otros no, apunto a otro segmento de memoria que es para uso general.

El contenido de los registros se ajusta depende de cada programa.

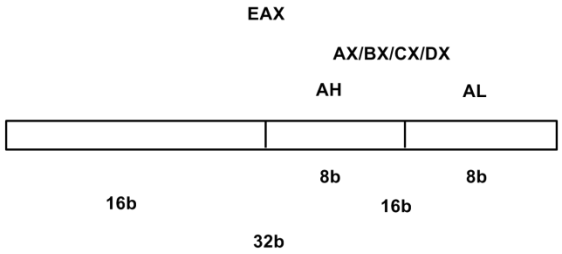
- **Registros punteros** (BP, SP, IP):
  - Registro de puntero de la siguiente instrucción (IP): contiene la dirección de la siguiente instrucción donde esta el siguiente código de operación que tiene que ser ejecutado. Apunta a una dirección desplazada dentro del segmento de código.

- Registro de puntero de la pila (SP): contiene la dirección siguiente donde se va a guardar el próximo valor de la pila. Si quiero leer el ultimo valor de la pila tengo que restarle uno al SP.
- Registro de puntero de base (BP): apunta a una zona intermedia dentro de la pila que es donde se guardan las variables dinámicas (variables que fueron asignadas con valor fijo al inicio de programa, se crean y se destruyen dentro del programa).

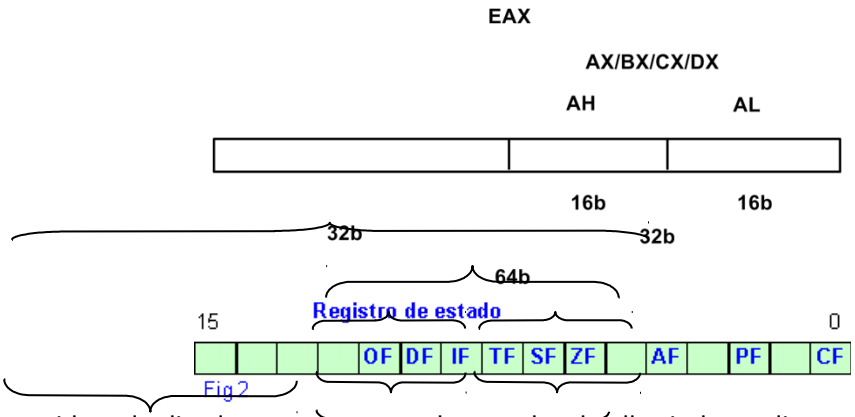
Estos registros se van cargando de acuerdo a las instrucciones del programa.
Los registros AX, BX, DX y CX tienen una particularidad (porque son de uso general):
Dijimos que tiene la longitud de una palabra pero a lo largo de los años esa longitud fue cambiando. Hoy en día hay procesadores de dos familias: que son los de 64 bit o de 32 bits, pero hubo de 16 bits, de 8 bits y también hay procesadores especiales de 256 bits aunque no son los clásicos.
Originalmente el registro era de 8 bits entonces cuando aparecieron los primeros procesadores de 16 bits el registro dejo de ser uno solo y se crearon dos subregistros para mantener la compatibilidad.



Cuando usamos el registro como AL (en el lenguaje de máquina se direcciona AL) se esta usando un registro de 8 bits, cuando usamos AH se esta usando otro registro de 8 bits y cuando lo mencionamos como AX estamos usando un registro de 16 bits. Yo puedo guardar algo en AL, leer algo de AH y después escribir algo en AX por ejemplo, porque se pueden usar independientemente y mezclados instrucción a instrucción sin cambiar nada, solamente el código.
Hasta acá venia todo bien hasta que salió la máquina de 32 bits entonces agregaron otros 16 bits, pero no se mantiene la posibilidad de trabajarlos por separado. Se puede trabajar como un registro de 32 bits, como un registro de 16 (el otro no se puede usar) o como dos de 8 bits.



Cuando salieron los 64 bits, por limitaciones de hardware, no se pudo seguir manteniendo este esquema. Perdieron la estructura de los registros de 8 bits (no puede correr ningún programa de 8 bits). En las ultimas pc tampoco se puede correr de 16bits (en los últimos core i) no admiten compatibilidad con programas de 16/8 bits ,Solo corren con 32/64 bits.



AX BX CX DX se los considera duplicados porque se puede usar dos de ellos independientes.

- F (registro de flags/estado): se maneja bit a bit. No guarda dirección de nada. En hoy en día son de 32/64 bits, cada bit significa algo. Se interpretan bit a bit. cuando el bit está en 1 esta activado y en 0 desactivado.

- CF: si cuando termina de hacer una operación y hubo carry me lo indica. Se pone en 1 el bit.
- PF: es el bit de paridad, indica si el resultado en el acumulador tiene resultado par o impar. Se activa cuando es impar.
- AF: Acarreo auxiliar. Tiene que ver con las sumas en formato BCD (valor binario 0 al 9). Si por ejemplo sumo 0100+0110=1010, me da 10, no existe en tabla por lo tanto es un overfloat en la tabla BCD.
- ZF: indica que la operación me dio cero. Me sirve para cuando hago comparaciones ej if (a == b) la maquina carga en ax y bx los valores y los resta,y se fija si el zf da cero, en ese caso son iguales. Se pone 1 cuando la op te da cero.
- SF: bit de signo. Nos dice si es positivo (0) negativo (1).
- TF: cuando esta 1 realiza un ciclo de instrucción de maquina y hasta que no le das enter no sigue con la siguiente instrucción, es para hacer paso a paso el programa (debug).
- IF: 0 bloquea las interrupciones enmascarables, en 1 las permite. Bloquea la atención de interrupciones porque si no las interrupciones son miles y van cayendo por segundo y si no las bloqueo dilata mucho la atención de interrupciones.
- DF: nos da la idea en qué sentido se recorre la memoria, creciente de las direcciones o decreciente, se usa para cuando se hacen operaciones para recorrer vectores o matrices, que recorren direcciones nos dice en qué sentido.
- OF: bit de overflow, cuando se supero tamaño de palabra en acumulador, ej te da 5 bits el resultado. La diferencia con el AF es que este opera con la palabra binaria normal.

Diferencia entre lenguaje de máquina y lenguaje ENSAMBLADOR: Los dos lenguajes son lo primero que se diseñan cuando se va a diseñar un procesador.
Lenguaje de maquina binario puro, se representa para las personas en hexadecimal. Lenguaje ensamblador son nemónicos (representa algo) que al compilarlos se convierten en lenguaje de máquina. La máquina opera exclusivamente con lenguaje de máquina. Los programas no se pueden ejecutar en ensamblador. No hay nada que se ejecute mas rápido que el lenguaje de maquina, ya que no hay que traducir nada. En el assembler es muy complejo buscar errores.
Cuando se diseña el procesador, ahí se define si va a ser RISC o CISC, si usamos instrucciones complejas o reducidas.
Los registros manejan direcciones más grandes que el rango de direcciones que puede manejar el registro propiamente dicho. En los procesadores de 16 bits podemos tener 2 a la 16 direcciones = 65536. Para direccional rangos de memorias mas grandes se trabaja con combinación de registros y se calcula lo que se llama la dirección efectiva (valor que se vuelca al bus de direcciones para que pueda accederse a la ram), a un segmento lo multiplica por 16 y le sumo un desplazamiento que esta en otro registro de 16 bits (todo en hexadecimal, en este caso). Si multiplico un número por su base le agrego un 0 a la derecha.

Unidad de codificación: Se encarga de decodificar la instrucción que se va a ejecutar, interpretar ese código para averiguar el tipo de instrucción a realizar.

Unidad de ejecución: Encargada de consumar la ejecución y para ello activará las señales necesarias y en un orden determinado.

Unidad de control: centro nervioso del ordenador. Desde ella se controlan y gobiernan todas las operaciones. sus funciones básicas son:

- Tomar las instrucciones de memoria
- Decodificar o interpretar las instrucciones
- Ejecutar las instrucciones (tratar las situaciones de tipo interno (inherentes a la propia CPU) y de tipo externo (inherentes a los periféricos)

Para realizar su función consta de los siguientes elementos:

- **Contador de programa**: Contiene la dirección de memoria de la siguiente instrucción a ejecutar.
- **Registro de instrucciones**: Contiene la instrucción que se está ejecutando en cada momento.
- **Decodificador**
- **Reloj:** Proporciona una sucesión de impulsos eléctricos o ciclos a intervalos constantes (frecuencia constante), que marcan los instantes en que han de comenzar los distintos pasos de que consta cada instrucción

• **Secuenciador**: En este dispositivo se generan órdenes muy elementales (microórdenes) que, sincronizadas por los impulsos de reloj, hacen que se vaya ejecutando poco a poco la instrucción que está cargada en el registro de instrucción.

UNIDAD ARITMÉTICO-LÓGICA (ALU): bloque funcional del microprocesador encargado de realizar todas las operaciones matemáticas y lógicas (suma, resta, multiplicación, división y aquellas que trabajan con dígitos binarios, operaciones lógicas: AND, NOR, NOT, NAND, OR, X-OR, etc). A través de un bus interno se comunica con la unidad de control la cual le envía los datos y le indica la operación a realizar. Formada por los siguientes elementos:

• **Circuito operacional :** Contiene los circuitos necesarios para la realización de las operaciones con los datos procedentes de los registros de entrada (REN). Este circuito tiene unas entradas de órdenes para seleccionar la clase de operación que debe realizar en cada momento (suma, resta, etc).
• **Registros de entrada (REN) :** En ellos se almacenan los datos u operandos que intervienen en una instrucción antes de la realización de la operación por parte del circuito operacional. También se emplean para el almacenamiento de resultados intermedios o finales de las operaciones respectivas.
• **Registro acumulador:** Almacena los resultados de las operaciones llevadas a cabo por el circuito operacional. Está conectado con los registros de entrada para realimentación en el caso de operaciones encadenadas. Asimismo tiene una conexión directa al bus de datos para el envío de los resultados a la memoria central o a la unidad de control.
• **Registro de estado (flags):** Se trata de unos registros de memoria en los que se deja constancia algunas condiciones que se dieron en la última operación realizada y que habrán de ser tenidas en cuenta en operaciones posteriores. Por ejemplo, en el caso de hacer una resta, tiene que quedar constancia si el resultado fue cero, positivo o negativo.

Instrucciones: Ciclo de instrucción: conjunto de acciones que se llevan a cabo en la realización de una instrucción. Se compone de dos fases:

- Fase de búsqueda: transfiere la instrucción que se va a ejecutar desde la memoria central a la unidad de control.
- Fase de ejecución: Consiste en la realización de todas las acciones que conlleva la propia instrucción.

Pasos del ciclo de intruccion:

1. **Búsqueda de instrucción.** Copiar PC en MAR *(Registro de Dirección de Memoria)*. Leer la instrucción siguiente y la coloca en el IR *(Registro de Instrucción)*.
2. **Búsqueda del registro / decodificación de la instrucción.** Decodificar la instrucción. Leer operandos (registros). Incrementar el PC.
3. **Dirección efectiva / ejecución.** Se pueden dar los siguientes casos:
    a. **Instrucción de referencia a memoria**: a través de la ALU se construye la dirección efectiva *(Efective Address)*, que se carga en el MAR, y el dato se carga en el MDR (Registro de Datos de Memoria) si estamos ante una instrucción de escritura.
    b. **Instrucción ALU**: se ejecuta la operación de la ALU.
    c. **Instrucción de salto / bifurcación**: se calcula la dirección destino a través de una operación de la ALU. En saltos condicionales, se examina un registro para evaluar la condición, y en su caso se calcula la dirección de salto.
4. **Completar salto /acceso a memoria.** Se pueden dar los siguientes casos:
    a. • **Instrucción de referencia a memoria**: cargar en memoria o leer de ella empleando los registros MAR y MDR.
    b. • **Instrucción de salto**: cargar la dirección de salto en el PC.
5. **Postescritura:** Se procede a escribir el resultado (procedente de la ALU o de MDR) en los registros o la memoria