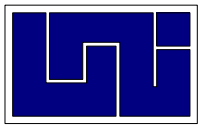

UNIVERSIDAD NACIONAL DE INGENIERIA

RECINTO UNIVERSITARIO SIMON BOLIVAR

Facultad de Electrotecnia y Computación
Departamento de Arquitectura y Sistemas

Folleto de

Arquitectura de Máquinas Computadoras II

UNIDAD III

Curso 2007

Preparado por:

Harriete Martínez Cano

Managua, Agosto 2007

INDICE DE CONTENIDO

2 ORGANIZACION DEL PROCESADOR

2.1 El Procesador

Los procesadores tienen como misión ejecutar operaciones de cómputo. Sabemos del modelo de Von Neumann que estas operaciones se definen como **instrucciones**. El procesador, desde que recibe energía hasta que se apaga ejecuta constantemente instrucciones. Esta función de procesamiento es la más importante en el funcionamiento de una computadora, por lo cual al procesador se le denomina Unidad Central de Procesamiento o CPU. En esta unidad analizaremos las estructuras que requiere un CPU muy sencillo para poder ejecutar una instrucción siguiendo los pasos del ciclo de Instrucción.

2.2 Instrucción y ciclo de instrucción

Como hemos estudiado antes, una instrucción es una orden que emite el programador al procesador para que ejecute una de las operaciones que forman parte de su repertorio. La instrucción debe especificar la operación a realizar y cómo obtener los datos u operandos para realizarla. Para este fin, la instrucción debe tener una estructura lógica denominada formato de instrucción, que se estudiará más adelante.

La secuencia de pasos que sigue el procesador para ejecutar una instrucción se denomina Ciclo de Instrucción. El Ciclo de Instrucción, se pueden organizar en dos bloques o fases de acuerdo a la acción que ocurre. La primera fase, denominada FETCH o de Carga. Corresponde a los pasos necesarios para cargar la instrucción desde la memoria al procesador.

La segunda fase, denominada EXECUTE, corresponde a aquellos pasos que permiten ejecutar la operación en sí y almacenar el resultado:

- FASE FETCH
  - Cargar la siguiente instrucción
  - Incrementar el secuenciador
  - Interpretar la Instrucción
- FASE EXECUTE
  - Cargar los operandos
  - Ejecutar la operación
  - Guardar el resultado
  - Verificar si hay solicitudes de interrupción

2.3 Elementos para estructurar el Procesador

Para poder realizar su función, el procesador, necesita una serie de estructuras. A continuación se expondrá qué tareas requiere llevar a cabo un procesador muy simple y deduciremos qué elementos constructivos se requieren para ejecutar estas tareas, siguiendo las pautas que nos brinda el ciclo de instrucción.

Para estar preparado para realizar sus funciones, el procesador debe:

- **Poder acceder a memoria para leer las instrucciones y los operandos** así como escribir los resultados. Se necesita acceder a la dirección de memoria que apunta el PC para leer la próxima instrucción. De acuerdo a la instrucción, puede requerirse leer de memoria los operandos y/o escribir a memoria, los resultados de la operación.

Para esto, el CPU debe tener un registro conectado al bus de direcciones donde poner la dirección a acceder. A este registro lo llamaremos registro de dirección de memoria o MAR por sus siglas en inglés. También requerimos otro conectado al bus de datos donde poner el valor a escribir o tomar el valor leído desde la memoria. Este lo llamaremos registro de datos de memoria o MDR por sus siglas en inglés. Por supuesto, se necesita una forma de indicarle a la memoria que realice la operación de lectura o escritura y además poder saber cuando la memoria terminó la operación, sobre todo la de lectura. Hasta entonces el procesador puede asumir que en el MDR hay datos válidos. Estas señales se implementan en la unidad de control: READ, WRITE, (que puede ser una sola línea del Procesador R/W, 0 =R; 1 =W, además de alguna línea de habilitación de Memoria.) y MFC (Que en nuestro caso será parte de la lógica de control de Memoria y que indicará al CPU que la función de memoria se ha completado).

- **Llevar control de la secuencia de instrucciones**: Se necesita un secuenciador de instrucciones, es decir, un mecanismo que vaya indicando una a una las instrucciones a cargar desde la memoria. Para ello, todas las instrucciones del programa a ejecutar deben estar almacenadas de forma secuencial en la memoria. El secuenciador se implementa empleando un registro que inicialmente se hace apuntar a la primera instrucción del programa y luego se incrementará para apuntala a la siguiente y así en lo sucesivo. Este registro de llama contador de programa o PC por sus siglas en inglés. Este enfoque es muy flexible pues el PC, puede incrementarse dentro del procesador una vez cargada la instrucción para que apunte a la siguiente. Otra ventaja, es en el caso de la implementación de saltos o llamados a subrutinas, pues sólo se pone en el PC la dirección destino del salto o la subrutina y así controlamos el flujo de instrucciones en el programa.

- **Guardar la instrucción** en un lugar donde la unidad de control la pueda interpretar. Para que durante la ejecución de la instrucción, se pueda acceder a su contenido en cualquier momento, es necesario otro registro para almacenar la instrucción durante la fase de ejecución. Este registro se le denomina registro de instrucciones o IR por sus siglas en inglés. El IR está conectado directamente a los circuitos de la unidad de control.

- **Poder realizar cálculos aritméticos y lógicos**. Para esto el CPU requiere de una unidad de cálculos aritméticos y lógicos a la que llamaremos ALU por sus siglas en inglés. Realiza todas las operaciones aritméticas, como la suma y la resta. Y lógicas como AND, OR, XOR, NOT, etc. También realiza otras operaciones como comparaciones y operaciones a nivel de Bits.

- **Obtener los operandos para realizar las operaciones**. Estos pueden estar en memoria, ser definidos directamente en la instrucción o residir en un almacenamiento interno del CPU. Es ventajoso mantener los operandos dentro del CPU. Para esto se pueden disponer de registros de propósito general con los cuales pueda la ALU realizar las operaciones.

- **Mantener el estado de la máquina**. Para esto se requieren tener banderas y códigos de condición. Por ejemplo, para comparar dos números se puede restar el primero del segundo y si el resultado fue cero, podemos asegurar que son iguales. Si el resultado de diferente de cero y positivo, el primero es mayor pero si es negativo, el segundo es mayor. Para llegar a estas conclusiones necesitamos almacenar las condiciones de resultados como cero (Z), Signo (S), etc. Estas banderas y códigos se almacenan en un registro llamado F o Flags.

- **Interpretar la instrucción y luego controlar la ejecución de todos los pasos para ejecutarla**. Para esto se requiere de una unidad de control o CU que tenga un decodificador capaz de interpretar los diferentes campos lógicos del formato de instrucciones. Además esta unidad debe poder generar las señales necesarias para que se realicen todos los pasos del ciclo de instrucción adecuadamente.

2.4 Estructura Básica del Procesador

Los elementos constructivos del CPU se organizan de acuerdo al trabajo que realizan. Para estructurar coherentemente el procesador estos elementos se agrupan en unidades que se interconectan para estructurar un todo. Cada elemento se debe comunicar con los otros para poder realizar su función. Esto se logra a través de una estructura de interconexión. Existen dos formas de implementar esta estructura:

- **Uso de conexión directa**: Consiste en realizar la conexión cableada directamente de la salida de un elemento a la entrada del otro. En caso de requerirse lógica de enrutamiento, se emplean los elementos digitales adecuados como multiplexores.

- **Uso de Buses**: Consiste en definir un conjunto de líneas para interconectar múltiples elementos entre sí. Debe, sin embargo, implementarse la lógica de control que garantice el enrutamiento del dispositivo fuente al destino. Cabe mencionar que debe evitarse que dos dispositivos sean fuente en el bus de forma simultánea pues esto provocaría cortocircuitos en las líneas cuando esta esté sometida a dos valores lógicos diferentes.

La figura a continuación muestra un ejemplo de estructura de CPU con una posible interconexión de los elementos mediante buses.

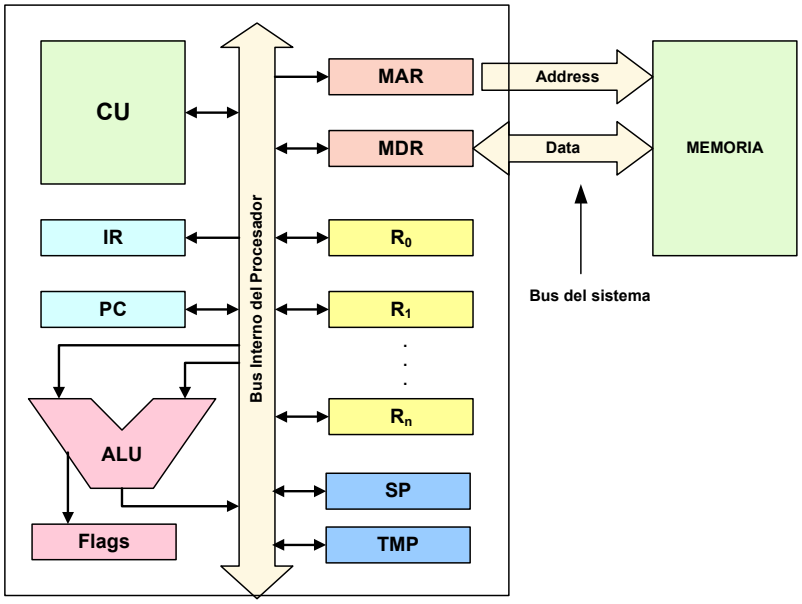

Figura 2.1 Elementos estructurales del procesador.

Modernamente se suele separar la estructura del CPU en dos partes generales: El camino de los datos o Datapath y el control. Conforman el Datapath todos los elementos por los que discurren los datos dentro del CPU, especialmente la ALU, los registros de propósito general, los registros de acceso a memoria y los registros adicionales.

Dentro del Datapath se organizar los órganos de cálculo (ALU y Registros de datos) en una entidad lógica denominada unidad de ejecución. Los registros de acceso al bus y registros auxiliares se suelen organizar en una unidad denominada "de Interfaz o carga".

El control incluye a la unidad de control que se encarga de coordinar y realizar las acciones necesarias para que se ejecuten las instrucciones.

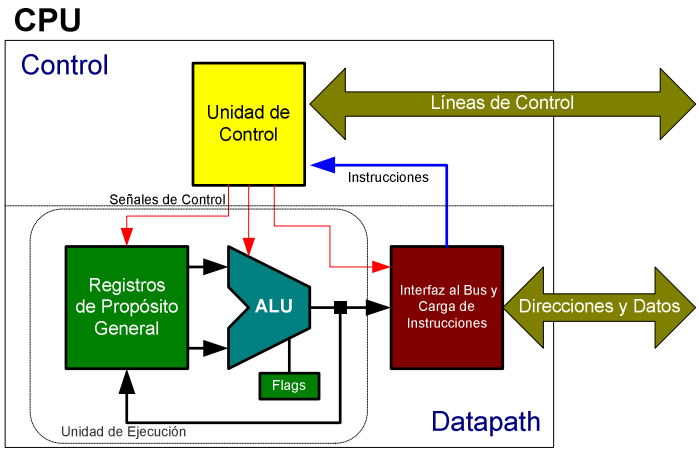

Figura 2.2. Organización del CPU en Datapath y Control

Ejecución de la instrucción:

Habiendo definido una estructura organizativa para nuestro procesador, estudiemos de forma general, los pasos para traer y ejecutar una instrucción basándonos en éstos elementos constructivos:

- El PC tiene siempre la dirección de la próxima instrucción (Cuando la máquina se enciende, éste se inicializa por hardware para apuntar la primera instrucción del programa de arranque). Primero el contenido del PC es transferido al MAR y la unidad de control envía una señal de lectura a la memoria.
- El PC es incrementado por el esquema de secuenciamiento para apuntar a la siguiente dirección.
- Después de un cierto tiempo correspondiente al tiempo de acceso de memoria, la palabra direccionada (la instrucción del programa) es leída de la memoria y cargada en el MDR.
- El contenido del MDR es transferido al IR Ahora la instrucción esta lista para ser decodificada y ejecutada.
- Si dicha instrucción involucra una operación que requiere operandos y estos residen en la memoria (ya que podrían estar en los registros generales del procesador) tienen que ser traídos enviando su dirección al MAR e iniciando un ciclo de lectura.
- Cuando el operando ha sido leído y trasladado de la memoria al MDR, entonces será transferido del MDR a la ALU o a algún registro de trabajo auxiliar para almacenarlo temporalmente mientras se traen otros operandos. La operación se repite para traer todos los operandos, normalmente dos.
- Se ejecuta la operación, mediante la ALU y el resultado es enviado al MDR si debe ser almacenado en memoria o al registro destino en el CPU. Si los operandos o el resultado requiere acceso al subsistema de E/S, se procese de igual forma que si se accede a Memoria, excepto que la UC se encarga de habilitar el dispositivo específico en lugar de la Memoria.

2.5 Arquitecturas de Acumulador, Pila y Registros.

Dependiendo de la forma en que trae, almacena temporalmente y trata los operandos y el resultado, los procesadores se han clasificado en tres tipos de arquitectura o estilos de procesador: De acumulador, de Registros y de Pila. Cabe mencionar que en la actualidad las máquinas más exitosas combinan características de las tres aunque predominan las máquinas orientadas a registros.

2.5.1 Máquina de Acumulador

Este es el esquema típico de Von Neumann. En un procesador basado en un solo registro denominado acumulador, de acuerdo a su función. Se carece de registros de propósito general para manipular los operados dentro del CPU. Todo el trabajo se realiza el acumulador y la memoria. Hoy día esto parece contraproducente, sin embargo, en el tiempo que esta arquitectura se implementó, la memoria era más rápida que los registros del procesador y estos últimos eran extremadamente caros de producir. Para realizar una operación, el programador debe tomar el primer operando y llevarlo al acumulador, realizar la operación con el acumulador y el contenido de una dirección de memoria (el resultado se guarda en el acumulador) y finalmente transferir el contenido del acumulador a memoria. La figura 2.3 nos muestra una arquitectura típica de acumulador.

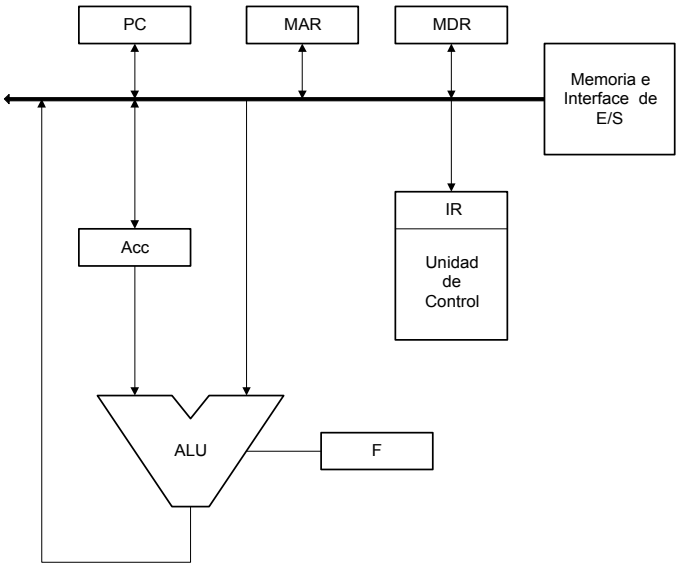

Figura 2.3. Arquitectura de Una máquina de Acumulador.

En este tipo de procesador tenemos dos instrucciones de acceso a memoria: la que carga un dato al acumulador y la almacena el acumulador en memoria. Las instrucciones binarias se realizan con el acumulador como primer operando y un operando en memoria o definido directamente como número. Por ejemplo:

INSTRUCCION	**OPERACIÓN**
LOAD X	Acc ← M(X) ; X es una variable de memoria
LOAD (m)	Acc ← M(m) ; m es una dirección de memoria
LOAD n	Acc ← n ; n es un número entero.
STORE X	M(X) ← Acc ; X es una variable de memoria
STORE (m)	M(m) ← Acc ; m es una dirección de memoria
ADD X	Acc ← (Acc) + M(X) ; X es una variable de memoria
ADD (m)	Acc ← (Acc) + M(m) ; m es una dirección de memoria
ADD n	Acc ← (Acc) + n ; n es un número entero.

Ejemplos de procesadores basados en acumulador son el IAS de Von Neumann, el M6502 y el 6809 de Motorola. Este último cuenta con dos acumuladores A y B.

2.5.2 Máquina de Pila

Una máquina de pila es una computadora en la cual el elemento primario de almacenamiento de datos para la CPU es una pila. Esta se implementa como un área de la memoria controlada por el CPU a la cual no tienen acceso los programas del usuario. En las primeras máquinas de pila, todos los operandos se manejaban mediante la pila, hoy día, sin embargo, la pila se emplea como una estructura de datos auxiliar para el CPU.

La ventaja principal de esta organización es que las manipulaciones de las pilas pueden ser realizadas en una alta velocidad. La alta velocidad es una característica deseable en una computadora que va proporcionar un servicio rápido de interrupciones, por ejemplo la IBM-370.

Las pilas más comunes son las de empuje hacia abajo, es decir, que inician en la dirección más alta de memoria y crecen en el sentido de las direcciones de memoria más bajas. La organización de la máquina de pila es mostrada en la figura 2.4 . El registro SP es el puntero a Pila, es un registro que siempre tiene la dirección de la última palabra insertada en la pila: Tope de pila o TOS (Top of Stack). En función de este registro, también se puede obtener el elemento inmediato bajo TOS: Próximo en pila o NOS (Next on Stack).

En el evento de una operación binaria, el primer operando será removido de la pila y mantenido en el registro TEMP. El segundo operando será directamente llevado de la pila a la entrada derecha de la ALU, que en nuestro ejemplo corresponde al puerto x o primer operando. El resultado de la operación es almacenado en la cabeza o tope de la pila.

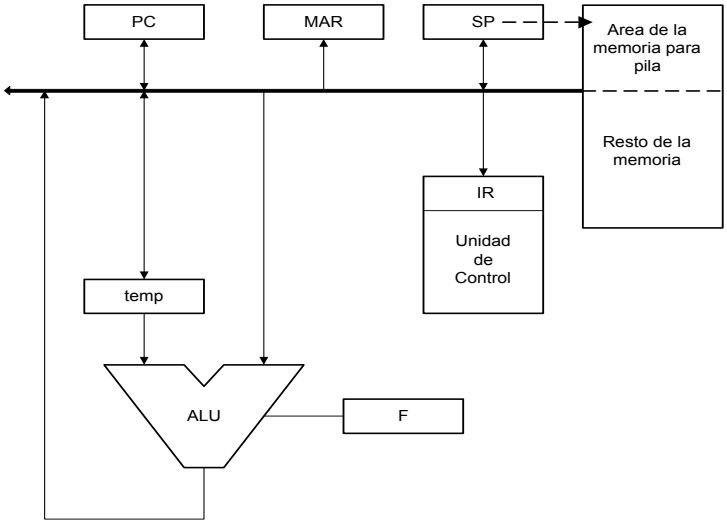

Figura 2.4 Arquitectura típica de una máquina de pila

Instrucciones típicas para una máquina de pila son:

Instrucción	**Operación**
PUSH X	TOS ← M(X)
PUSH (m)	TOS ← M(m)
PUSH n	TOS ← n
POP Z	M(Z) ← TOS
POP (m)	M(m) ← TOS
ADD	(TOS') = (NOS) + (TOS)
SUB	(TOS') = (NOS) - (TOS)
MUL	(TOS') = (NOS) * (TOS)
DIV	(TOS') = (NOS) / (TOS)

NOS: Next on the stack TOS: Top of the stack.

En la actualidad no existen ejemplos de esta organización en forma pura, sin embargo existieron máquinas que se construyeron con esta arquitectura como las grandes computadoras Burroughs B5500, B6500 y B6700 y la minicomputadora HP3000. Una de las ventajas de esta arquitectura es que el compilador sólo necesita convertir a postfija las expresiones algebraicas y luego hacer una traducción directa a ensamblador, empleando PUSH por cada aparición de variable y la operación correspondiente con cada operando. El pop es un caso especial; que corresponde a la asignación y se evalúa al final y además arrastra consigo al miembro izquierdo de la expresión.

2.5.3 Máquina de Registros.

La organización de un procesador de Registros Generales es mostrada en la figura 2.5. En este estilo de diseño respecto al manejo de los operandos, se promueve el uso de muchos registros de propósito general para almacenar los operandos temporalmente dentro del CPU. Esto tiene la ventaja que los datos más frecuentes sólo se cargan una vez desde la memoria.

Una arquitectura típica consta de un banco o fichero de *m* registros de propósito general (R_o... R_{m-1}). Son llamados de esta forma porque en cualquiera de ellos se pueden mantener datos, direcciones de memoria o el resultado de alguna operación aritmética o lógica. El programador puede emplear estos registros para realizar las diferentes operaciones en un

programa. Algunos procesadores tienen algún destino específico para algunos de ellos; el registro F (Flag) o PSW (Processor status Word) es un registro de banderas y/o códigos de condición que mantiene información importante del estado del procesador o del desarrollo del programa. Por ejemplo, cuando el resultado de una operación fue cero (bandera Z = 1) o cuando ocurrió un acarreo (bandera C = 1).

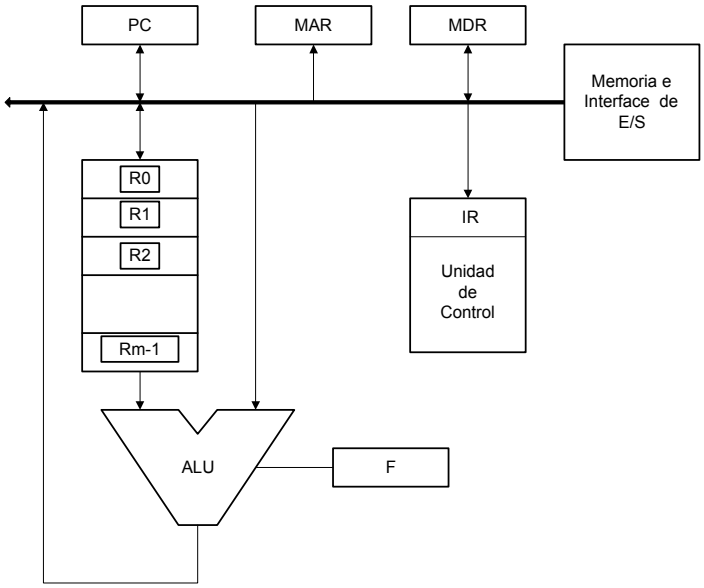

Figura 2.5. Organización Típica de un Procesador de Registros Generales

Con procesadores de registros generales podemos tener instrucciones de dos y tres operandos. En máquinas 2 operandos, el destino o resultado de la operación está implícito en uno de los dos operandos. Este fenómeno se denomina lectura destructiva de operandos, pues el valor del registro que es a la vez operando y destino se pierde por sobre escritura del resultado.

Instrucciones de 3 operandos

Instrucción	Operación
ADD Rd, Rf1, Rf2	; Rd ← Rf1 + Rf2
SUB Rd, Rf1, Rf2	; Rd ← Rf1 - Rf2
MUL Rd, Rf1, Rf2	; Rd ← Rf1 * Rf2
DIV Rd, Rf1, Rf2	; Rd ← Rf1 / Rf2

Instrucciones de 2 operandos

```
MOV Rd, Rf          ; Rd ← Rf
MOV Rd, n           ; Rd ← n | n es un número
MOV Rd, [m]         ; Rd ← M(m)  ; m es una dirección en memoria
MOV [m], Rf         ; M(m) ← Rf  ; m es una dirección en memoria
ADD Rf, Rd          ; Rd ← Rf + Rd
SUB Rf, Rd          ; Rd ← Rf - Rd
MUL Rf, Rd          ; Rd ← Rf * Rd
DIV Rf, Rd          ; Rd ← Rf / Rd
```

Procesadores del tipo de registros de propósito general son la mayoría de las máquinas grandes modernas y los procesadores RISC por excelencia. Ejemplos: IBM 370, VAX-11 de DEC, MC 68000 de Motorota, MIPS en todos sus modelos, SPARC, PowerPC, ARM.

Los diseñadores modernos de computadoras, valorando las virtudes de cada estructura han adoptado maneras de implementar máquinas con todas las ventajas de cada una, eliminando las limitaciones que tenían cada una por separado. Por ejemplo, la familia 80x86 de INTEL, son máquinas orientadas a acumulador con una cantidad no muy abundante de registros de propósitos generales con una pila integrada además.

Para entender mejor cómo varían las instrucciones respecto a cada estilo de CPU, se presenta a continuación un ejemplo que permite hacer comparaciones entre los diferentes tipos. Se tiene la siguiente sentencia en pascal: **D := A + B * C**, Escriba la secuencia de instrucciones en lenguaje de máquina con tres, dos, una y una y cero direcciones.

Tres operandos: (Suponer A está en R1, B en R2 y C en R3, R4 es D)

```
MUL R4, R2, R3          ; D = (B) * (C)
ADD R4, R1, R4          ; D = (A) + (B * C)
```

Dos operandos: (Suponer A está en R1, B en R2 y C en R3, R4 es D)

```
MOV R4, R3      ;D = ( C)
MUL R2, R4      ;D = (B)*(C)
ADD R1, R4      ;D = (A) + (D)
```

Un operando (Acumulador) :

```
LOAD B          ;Acc= (B)
MULT C          ;Acc= (A)*(C)
ADD A           ;Acc= (Acc) + (A)
STORE D         ;D = (Acc)
```

Máquina de Pila:

```
PUSH  A
PUSH  B
PUSH  C
MUL             ;  calcula B*C
ADD             ;  suma A a B*C
POP  D          ;  salva resultado
```