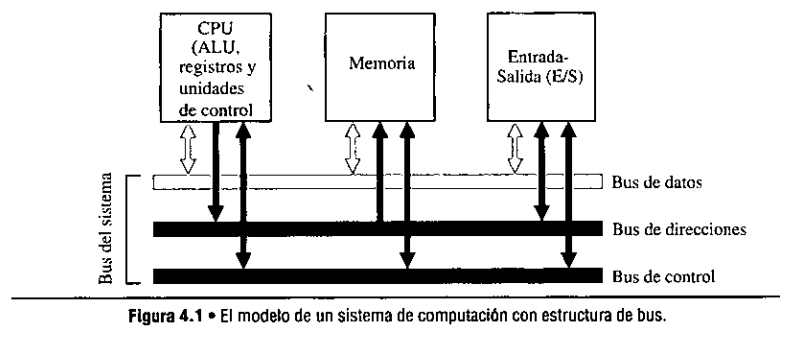
**Capítulo 4**

El propósito de un bus es reducir la cantidad de interconexiones entre la CPU y sus subsistemas. La CPU se interconecta con la memoria y los sistemas de I/O a través del bus del sistema, formado por el bus de datos, el de direcciones y el de control.



El usuario escribe un programa en lenguaje de alto nivel, luego el compilador lo traduce a Assembler, luego el ensamblador lo traduce a lenguaje de máquina para su almacenamiento en disco. Previo a su ejecución, el SO carga el programa de disco a memoria. Durante la ejecución, cada instrucción se carga en la CPU desde la memoria con cualquier dato que sea necesario para ejecutarla. Estas instrucciones se ejecutan en la CPU por lo que las instrucciones y los datos se cargan desde la memoria a los registros de la CPU mientras que los resultados se almacenan en la memoria desde los registros de la CPU.

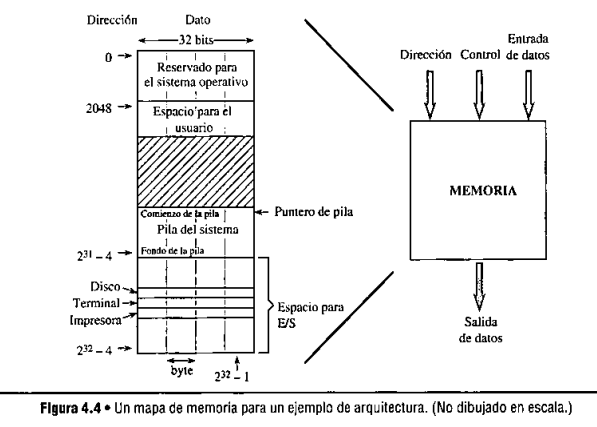
La memoria de una computadora consiste en un conjunto de registros, cada uno de los cuales almacena un byte. Cada registro tiene una dirección de memoria. El tamaño de la palabra depende de cada procesador (32 bits). Una palabra de más de un byte se almacena como una secuencia de bytes.

El mapa de memoria puede diferir entre una implementación y otra. Entonces, algunos programas compilados para el mismo tipo de procesador pueden no ser compatibles entre sistemas.

El espacio para el usuario puede crecer desde la dirección 2048 hasta que se encuentre con la pila. La pila comienza en la dirección 231 - 4 y puede crecer hacia las direcciones inferiores. El espacio de direcciones ente 231 y 232 - 1 se reserva para dispositivos I/O. El mapa de memoria no está compuesto enteramente por memoria real. Pueden existir espacios en los que no haya ni memoria real ni direcciones de dispositivos I/O.

La memoria tiene un espacio de direcciones de 32 bits, lo que significa que el programador puede acceder a un byte de memoria ubicado en cualquier posición en el rango de [0; 232-1].

Una dirección es un puntero a una posición de memoria, la cual contiene un dato. Tanto la dirección como el dato ocupan 32 bits. Una dirección de n bits puede especificar 2n direcciones. Una memoria con un espacio de direcciones de 32 bits tiene una capacidad máxima de 232 bytes o 4 Gb.



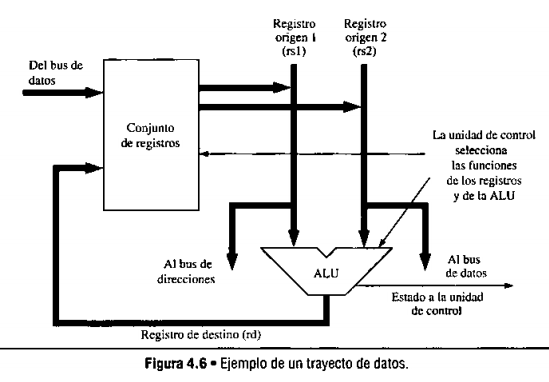
La CPU se divide en dos secciones: la sección de datos (registros + ALU) y la sección de control (unidad de control). La segunda es responsable de la ejecución de las instrucciones del programa (que se almacenan en memoria principal). Existen dos registros que forman la interfaz entre la unidad de control y la unidad de datos: PC (program counter) e IR (instruction register). En PC se guarda la dirección de la instrucción que está siendo ejecutada. Dicha instrucción se rescata de memoria y se almacena en IR, desde donde se la interpreta.

La unidad de control lleva a cabo el ciclo de fetch (búsqueda-ejecución):

1. Buscar en memoria la próxima instrucción a ser ejecutada
2. Decodificar el código de operación de esa instrucción
3. Ejecutar la instrucción
4. Volver a (1)

Los registros de la sección de datos forman una memoria pequeña y rápida, separada de la memoria del sistema, que se utiliza para almacenamiento temporario durante las operaciones de cálculo. La diferencia principal entre los registros y la memoria del sistema es que los registros están en la CPU y por ende es mucho más rápido acceder a ellos.

Tres buses conectan el datapath con el bus del sistema. Esto permite la transferencia de los datos entre la memoria principal y los registros. Otros tres buses conectan los registros con la ALU. Estos buses permiten la búsqueda simultanea de dos operandos almacenados en los registros para ser procesados por la ALU y luego devolver el resultado a los registros.

La ALU implementa una variedad de operaciones de uno y dos operandos. La unidad de control selecciona las operaciones y los operandos, que son transferidos desde los registros a través de dos buses. La salida de la ALU se transfiere a través de otro bus hacia los registros nuevamente.

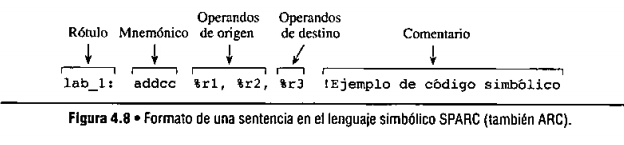
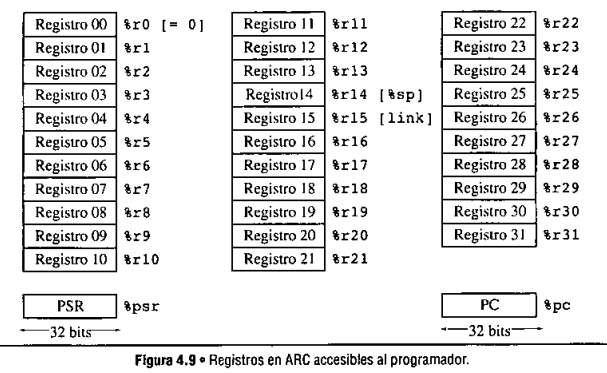
El conjunto de instrucciones es la colección de instrucciones que un procesador puede ejecutar. Estos conjuntos pueden ser radicalmente diferentes de uno a otro procesador. Las diferencias pueden ser:

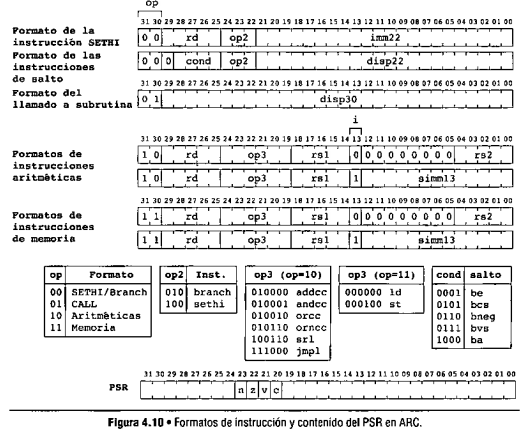
* Tamaño de las instrucciones
* Tipo de operaciones permitidas
* Tipo de operandos sobre los que se permite operar
* Tipo de resultados entregados

Un programa escrito en un lenguaje de alto nivel se puede ejecutar en distintos procesadores siempre y cuando sea recompilado.

Un compilador es un programa que traduce programas escritos en lenguajes de alto nivel a lenguaje de máquina. En el proceso de compilación de un programa, primero se traduce el código fuente a lenguaje simbólico (assembly lenguaje), luego de lo cual un programa ensamblador (assembler) traduce este lenguaje simbólico a código de máquina. Estas traducciones se producen en tiempo de compilación y tiempo de ensamblado respectivamente. El programa objeto resultante puede vincularse con otros programas objeto en el tiempo de enlace (link time). El programa objeto se carga en memoria principal desde el disco en el tiempo de carga (load time) y se ejecuta desde la CPU en tiempo de ejecución (run time).

Se analizará un subconjunto del procesador SPARC llamado ARC (A RISC (Reduced Instruction Set Computer) Computer). Es una máquina de 32 bits que puede manejar tipos de datos de 32 bits, pero toda la información se almacena en memoria en forma de bytes. Respeta el mapa de memoria de la página anterior. ARC tiene diferentes tipos de datos, pero consideraremos únicamente el entero de 32 bits. Cada entero se almacena en memoria como un conjunto de cuatro bytes. El almacenamiento es big endian. ARC tiene 32 registros de uso general de 32 bits cada uno, así como un PC y un IR. Existe un registro de estado del procesador (PSR) que contiene información acerca del estado del procesador, incluidas las flags. El tamaño de todas las instrucciones es de 32 bits. Además, es una máquina de arquitectura carga-descarga (load-store). Las únicas instrucciones permitidas para el acceso a memoria cargan un valor en un registro o almacenan en memoria el contenido de un registro. Todas las operaciones aritméticas se ejecutan sobre datos que estén en registros, y los resultados también se guardan en ellos.





Las instrucciones andcc, orcc y orncc setean el flag:

* z = 1 si todos los bits del resultado son 0.
* n = 1 si el bit más significativo es 1.
* c = v = 0 porque no realizan una operación aritmética sino lógica.

srl 🡪 desplaza un registro a la derecha y pone ceros en los bits más significativos.

sra 🡪 desplaza un registro a la derecha y almacena una copia del bit más significativo del contenido original en los bits vacíos creados por el desplazamiento. Esto es que extiende el signo del operando, preservando su signo aritmético.

addcc 🡪 suma de 32 bits en complemento a 2.

.dwb n (define word block) reserve un bloque de palabras de 4 bytes. El contador de posiciones se desplaza hasta ubicarse delante del bloque, de acuerdo con la cantidad de posiciones obtenidas al multiplicar por cuatro el argumento (n).