

2.1 Organización General de un Computador

La organización básica de un computador se compone de una unidad de procesamiento (CPU), memoria principal y dispositivos de entrada salida. Estas tres unidades están presentes en cualquier organización de computador y el modelo básico de la funcionalidad y organización fue propuesto por Von Neumann. Un ejemplo sencillo se muestra en el siguiente esquema en una topología de bus único.

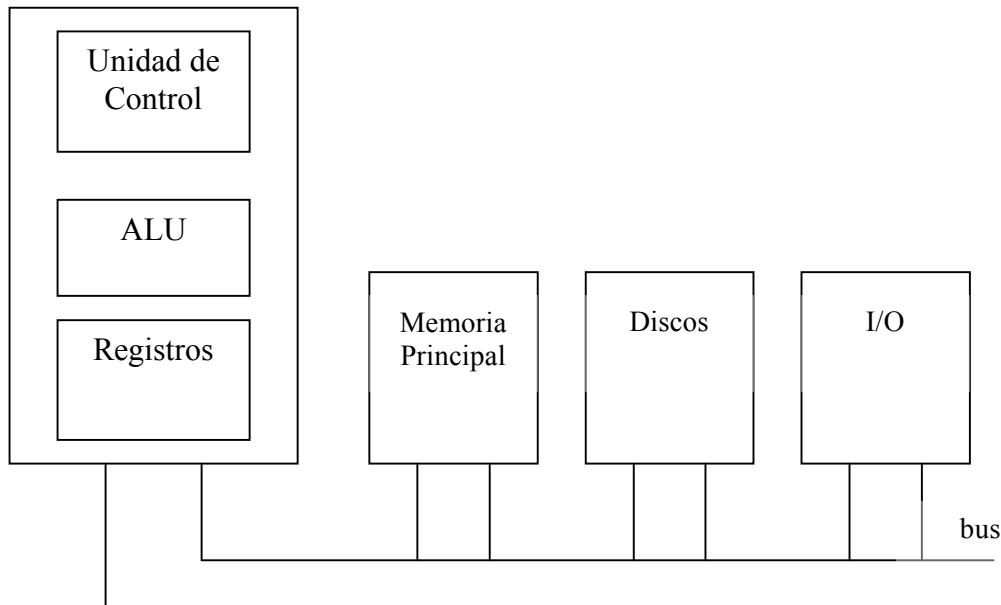


Figura: Organización de bus único

- **CPU:** Esta arquitectura está centrada en la unidad de procesamiento, que ejecuta programas almacenados en la memoria principal (concepto de programa almacenado), por medio de los cuales: extrae, modifica y almacena datos de memoria.
- **I/O:** Los dispositivos de entrada salida son la interfaz del computador con el medio externo distinguiéndose:
 - unidades de salida (impresoras, monitor o display, etc),
 - unidades de entrada (teclado, mouse, etc.),
 - unidades de entrada y salida (discos duros).
- **BUS:** El medio físico de comunicación entre la CPU, la memoria y los dispositivos de I/O es el bus. Éste corresponde a un conjunto de líneas conductoras paralelas donde se transfiere la información binaria entre los dispositivos. Se compone de tres partes:
 - **Bus de direcciones:** permite identificar dispositivos y la unidad o celda direccionable dentro de él (el caso de la memoria principal),
 - **Bus de datos:** transfiere la información entre los distintos dispositivos
 - **Bus de control:** provee la señalización del tipo de acceso (principalmente lectura y escritura).

De esta manera la CPU puede hacer ciclos de lectura y escritura sobre la memoria principal y al mismo tiempo interactuar con los dispositivos de entrada salida. Para lo último la CPU escribe o lee sobre una posición de memoria asociados a estos dispositivos (concepto de dispositivos mapeados en memoria **memory mapped**).

Dentro de las funcionalidades de un computador se reconoce cuatro tipos básicos:

- **Procesamiento de datos:** Funcionalidad asociada a la unidad aritmética y lógica (ALU) dentro de la organización de la CPU. El procesamiento está estrictamente vinculado a la forma en que la información es codificada.
- **Almacenamiento de datos:** Existe una jerarquía de niveles de almacenamiento en la organización, cada uno con atributos específicos (Jerarquía de Memoria).
- **Transferencia de datos:** Funcionalidad que permite la interacción del sistema con el medio externo.
- **Control:** Lo relacionado a la capacidad de gobernar las unidades dentro de la organización de un computador. La información del control está implícita en las instrucciones y es la CPU la encargada de interpretarlas y llevarlas a cabo.

Por ser la unidad central de procesos (CPU) el centro funcional de esta organización, nos centraremos en explicar en detalle su funcionamiento y de qué manera interactúa con el resto de las unidades, principalmente la memoria.

2.2 Unidad Central de Procesos

Corresponde a la unidad principal dentro de la arquitectura de un computador (dispositivo maestro por excelencia) cuya funcionalidad es la ejecución de programas almacenados en memoria central. En palabras generales esta unidad extrae cada instrucción, la examina, ejecuta y gobierna el flujo de ejecución de dicho proceso, i.e. determina la secuencia de micro-eventos a ejecutar.

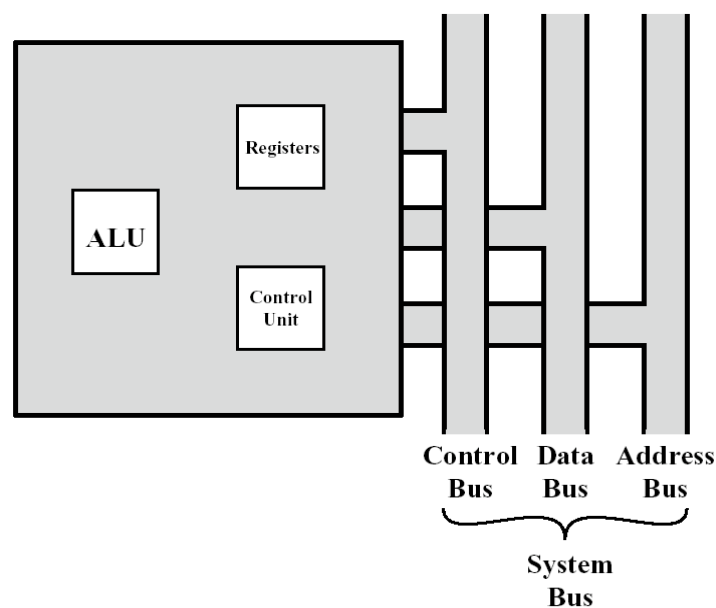


Figura: Bloques básicos en la Organización de una CPU

Sus partes fundamentales son:

Unidad de Control: Extrae la instrucción en curso de memoria principal (nivel de lenguaje de máquina), hace el proceso de decodificación para interpretar su tipo y la naturaleza de sus argumentos. Con esto provee la señalización de control para las restantes etapas de la ejecución de dicha instrucción.

Unidad Aritmética Lógica (ALU): Dispositivo que realiza las operaciones aritméticas (sumas, multiplicaciones) o lógicas (OR, AND, NOT), determinadas por la unidad de control, sobre los argumentos también determinados en el proceso de decodificación. Como se verá, esta unidad se diseña utilizando dispositivos combinacionales y secuenciales (compuertas lógicas)

Registros: Pequeña memoria de alta velocidad integrada dentro de la CPU, que permite almacenar resultados intermedios, información de control y estado en dicho proceso. Dentro de los registros más emblemáticos presentes en prácticamente todas las organizaciones de procesadores se encuentran:

- **PC (Program Counter):** Almacena la información de la posición de memoria de la instrucción a ejecutar por la CPU, su modificación a lo largo del tiempo esta relacionado con el flujo lógico de ejecución de un programa.
- **IR (Instruction Register):** Almacena la información de la codificación en lenguaje de máquina (nivel 2) de la instrucción en curso. Corresponde al contenido de la posición de memoria direccionada por el registro PC al comienzo de la ejecución de una nueva instrucción.
- **STATUS:** Este registro almacena las celdas de estados de la CPU (o **flags**) después de haber ejecutado cierta instrucción. Esta información es indispensable para implementar instrucciones de control de flujo (**branch**) al nivel de lenguaje de máquina.
- **ACUMULADORES:** Registros temporales disponibles para programadores de nivel 2 (lenguaje de máquina) para almacenar variables intermedias en un proceso. Estos pueden ser de propósito general o exclusivamente de datos o direcciones, estos últimos para acceder a argumentos en memoria principal. El tamaño de estos registros (número de bits) y el número de éstos depende del diseño arquitectónico de la CPU.
- **SP:** Almacena la dirección de memoria del comienzo de una pila (*stack*), para almacenar variables locales de los procedimientos. Este espacio memoria temporal se maneja con este registro.

La siguiente figura muestra un esquema más detallado de la organización de una CPU. La figura muestra los buses internos los cuales permiten la transferencia entre las unidades involucradas y la señalización de control que gobierna dichas transferencias, en la cual la **Unidad de Control** actúa como dispositivo maestro. Adicionalmente estos buses permiten la transferencia entre los registros (memoria interna de la CPU) y la ALU. De este esquema se hace evidente que la ALU solo interactúa directamente con su memoria interna (banco de registros)

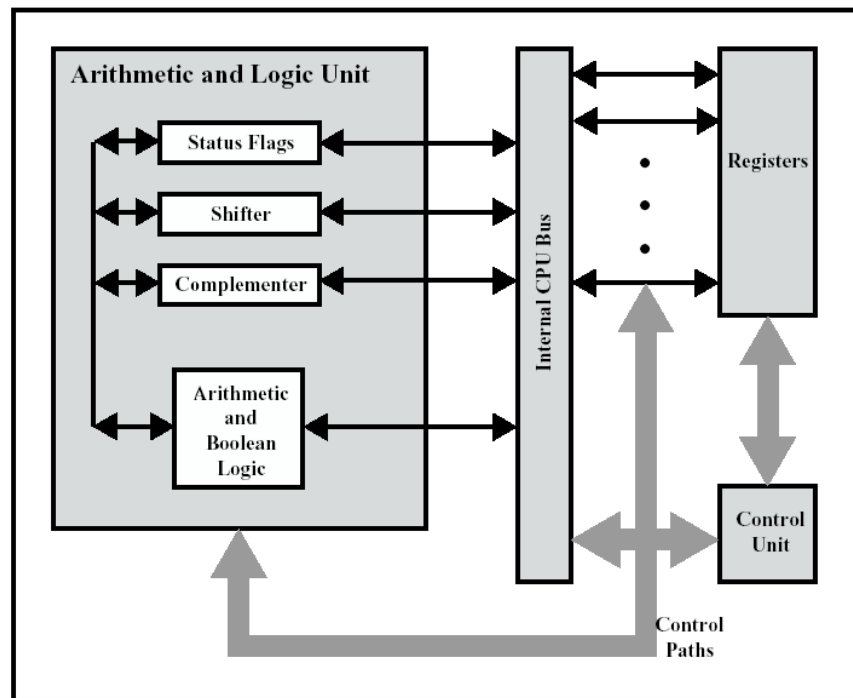


Figura: Estructura Interna de CPU

2.3 Clasificación de Registros

Como se verá mas adelante, los **registros** son el primer eslabón de una jerarquía de memoria en la arquitectura de un computador, correspondiente al nivel de memoria físicamente más próximo a la ALU y el que provee las mejores presentaciones del punto de vista de tiempo de acceso¹. Los registros de la CPU se pueden clasificar en dos grupos:

- **Registros visibles para el programador nivel 2:** Son los registros accesibles al programador de lenguaje de máquina, los cuales son utilizados con el objeto de minimizar los accesos a memoria principal, típicamente registros de entrada salida de la ALU. Estos registros pueden ser de uso general o específico. Por ejemplo hay registros que sólo son utilizados como registros de direcciones, otros que sólo son registros de datos (argumentos de entrada o salida de la ALU). También pueden existir registros de propósito general los cuales son utilizados tanto como datos como para el cálculo de direcciones. Dado que estos registros son visibles para el programador de nivel 2, estas restricciones están estipuladas en la sintaxis de dicho lenguaje². De aquí se ve que el lenguaje de máquina (repertorio de instrucciones interpretado por la CPU) esta condicionado a las restricciones de diseño de su CPU.
- **Registros de Control y estado:** son utilizados por la unidad de control para controlar el funcionamiento de la CPU, ejemplo PC, IR, STATUS. Este último almacena los bits de código de condición, información del estado de ejecución del proceso que permite implementar los saltos condicionales en el flujo de un proceso. Estos bits son fijados por el hardware de la CPU a consecuencia de la última operación de la ALU. Los más emblemáticos son:
 - Signo:** contiene el bit de signo de la última operación
 - Cero:** indica cuando la operación fue cero
 - Acarreo:** indica acarreo de bit en la suma.

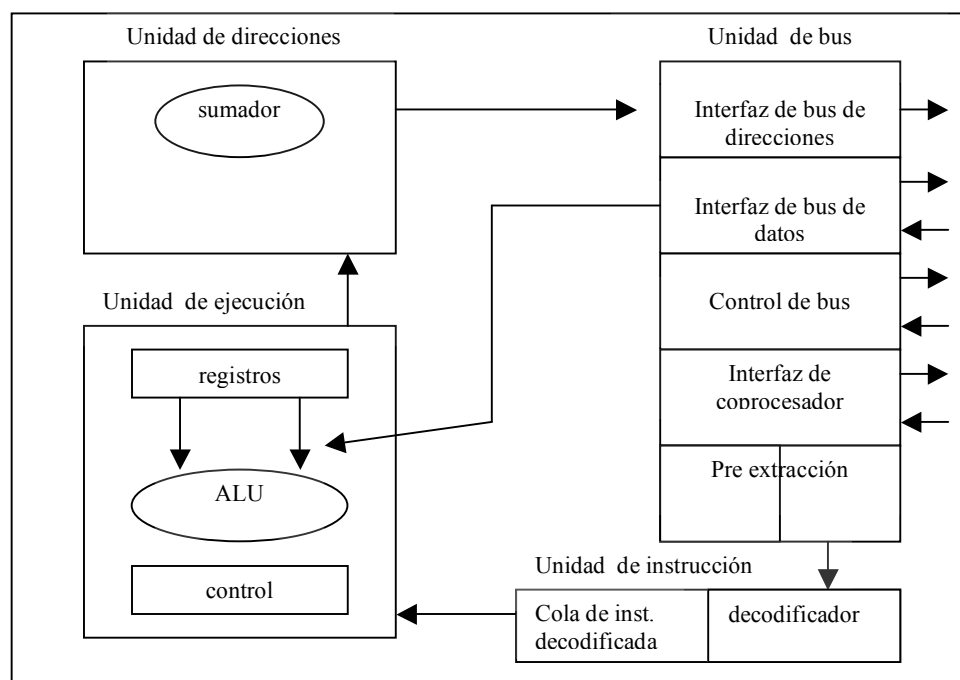
¹ Unidad VI Sistema de Memoria se verá en detalle todos los niveles de memoria, sus propiedades y cual es la razón de diseño en su organización y estructura.

² Estos puntos se verán en detalle en Unidad IV de Lenguaje de Máquina.

Desbordamiento: usado para indicar desbordamiento aritmético (resultado inválido, fuera del rango de representabilidad numérico)³

2.4 Ejemplo Intel 286

A continuación se muestra la arquitectura interna básica de un procesador Intel 80286. En este ejemplo se ve que existen muchas más unidades presentes en el proceso de ejecución de instrucciones. La razón es que existen procesos adicionales que explícitamente no se han mencionado.



Las unidades presentes y su funcionalidad son:

Unidad de instrucciones: toma la palabra asociada a la instrucción codificada en lenguaje de máquina desde la *Unidad de bus* y hace el proceso de “*decodificación*”.

Unidad de Ejecución: lleva a cabo la instrucción decodificada por la *Unidad de Instrucción*, internamente a esta unidad está el bloque de control que hace el proceso de interpretación de los campos decodificados de la instrucción y gobierna a la ALU y *Unidad de Direcciones* para efectos de manejar la transferencia de datos de la instrucción en curso (*trayectoria de datos*)

Unidad de Bus: realiza todas las operaciones de bus las cuales permiten comunicación con la memoria y los dispositivos de I/O. Está encargada de hacer la interfaz para la extracción de instrucciones y datos desde memoria. Tiene un buffer de instrucciones para extraer por adelantado las instrucciones siguientes (principio de localidad espacial) para evitar que la *Unidad de Ejecución* pierda ciclos de reloj esperando que esta extraiga la instrucción en curso (**pipeline**).

³ Esto puntos se tocarán con mayor detalle, en las unidades siguientes

2.5 Instrucciones

En esta etapa entenderemos como instrucción al set de palabras de código binario que se provee al nivel de lenguaje de máquina (nivel 2). La CPU gobierna la extracción secuencial de instrucciones de un programa desde la memoria principal (programa ejecutable). Por tanto una instrucción se representa físicamente por una trama de bits almacenados en algún bloque de memoria. El tamaño de esta depende del diseño al nivel de lenguaje de máquina, por ejemplo la arquitectura **MIPS** (SPARC5) tiene instrucciones de máquina de 32bit (4byte) independiente de su tipo⁴.

Los campos básicos se ilustran en el siguiente ejemplo:

opcode	rs	rt	rd	adress
--------	----	----	----	--------

- **Opcode:** la codificación de la operación a realizar por la CPU
- **rs:** la codificación del primer registro de la CPU (argumento 1)
- **rt:** la codificación del segundo registro de la CPU (argumento 2)
- **rd:** la codificación del registro de la CPU de destino.
- **adress:** codificación de un direccionamiento local para instrucciones del tipo de transferencia de datos (llevar datos desde la memoria a registros y vice-versa).

Es interesante mencionar que dependiendo del tipo de instrucción es el formato que esta posea, pues pueden tener uno o dos argumentos y dependiendo de eso la validez de sus campos⁵.

El proceso de decodificación o interpretación de campos de una instrucción esta asociado a la Unidad de Control de la CPU. La salida de este proceso es la señalización de control para todas las unidades internas de la CPU (ALU, registros, shift, multiplexores, etc.) para llevar a cabo la ejecución.⁶

2.5.1 Ejecución de Instrucciones

El proceso de ejecutar instrucciones se puede subdividir en una secuencia de pasos lógicos en un ciclo llamado **extrae – decodifica – procesa (EDP)**. A nivel macro se puede decir que ocurren las siguientes instancias en dicho proceso:

1. **(Instruction Fetch)** Extraer la instrucción de memoria (direccionada por el **PC**) y almacenarla en el **IR**. Utiliza las interfaces del bus de datos y direcciones.
2. **(Instruction Decode)** Unidad de control inicia el proceso de decodificación, para determinar el tipo de instrucción (*opcode*). También se determina en esta etapa la naturaleza de los argumentos, por ejemplo si hay que accesarlos de memoria. Paralelamente se incrementa el **PC** para apuntar a la siguiente instrucción.
3. **(Operand Fetch)** Extrae los argumentos desde memoria o registros (si los hay) y carga en los registros de entrada de la ALU.
4. **(Execute)** Ejecuta instrucción (ALU).
5. **(Result Store)** Almacena los resultados en el lugar apropiado (registros internos temporales o memoria)

⁴ En el capítulo de formato de instrucciones se verá los criterios de diseño de las instrucciones de nivel 2 y el caso particular de instrucciones de largo variable.

⁵ El detalle del formato de cada instrucción se verá posteriormente al discutir los tipos de instrucciones de maquina

⁶ El detalle de este proceso se vera en la Unidad III de Microprogramación.

6. Va al paso 1.

Todo el proceso que permite que la ALU finalmente actúe sobre sus registros de entrada y retornar los resultados, se conoce como trayectoria de datos (**datapath**).

Ejemplo 1 (ejecución de instrucción)