

Resumen Unidad 2 – Arquitectura de Computadoras

◆ 2.1 Registros visibles para el usuario

Son registros que pueden ser accedidos directamente por los programadores para manipular datos y direcciones. Están diseñados para facilitar la programación y mejorar el rendimiento del procesador.

Tipos de registros:

- **Registros de datos:** Almacenan valores numéricos que se usan en operaciones aritméticas o lógicas.
- **Registros de direcciones:** Guardan direcciones de memoria necesarias para acceder a datos.
- **Registros de propósito general:** Pueden usarse para diversas funciones según el programa (almacenar variables, resultados, etc.).
- **Registros de condición:** Contienen información sobre el resultado de operaciones (como si hubo un desbordamiento o si el resultado fue cero).

✓ **Importancia:** Estos registros permiten un acceso rápido a los datos, lo que optimiza la ejecución de instrucciones y facilita el trabajo de los desarrolladores.

◆ 2.2 Registros de control y de estado

Son registros utilizados internamente por la CPU para tomar decisiones durante la ejecución de instrucciones y supervisar el estado general del sistema.

Registros de control:

- Determinan cómo opera la CPU.
- Incluyen configuraciones de acceso a memoria, manejo de interrupciones y contexto de procesos.

Registros de estado:

- Reflejan el estado actual del procesador.
- Contienen banderas como:
 - **Overflow (desbordamiento)**
 - **Carry (acarreo)**
 - **Zero (resultado cero)**

✅ **Importancia:** Son clave para la estabilidad del sistema, permitiendo que el procesador reaccione ante condiciones específicas o errores.

◆ 2.2.3 Organización de registros en CPUs reales

Tipos comunes:

- **Propósito general:** Para operaciones básicas. Ej.: AX, BX (x86), x0-x31 (RISC-V).
- **Propósito específico:** Incluyen registros como el **PC** (Program Counter) y el **IR** (Instruction Register).
- **Estado y control:** Ej.: FLAGS en x86, CPSR en ARM.
- **Punto flotante:** Para cálculos matemáticos complejos, como los registros FPR.

Ejemplos de arquitecturas:

- **RISC-V:** 32 registros x0-x31, PC y CSR.
- **PowerPC:** 32 registros enteros y 32 de punto flotante.
- **x86:** EAX, EBX, ESP (pila), EIP (instrucción), EFLAGS.

- **ARM:** R0-R15; CPSR para banderas y control.
-

◆ 2.3 Ciclo de instrucción (Fetch-Decode-Execute)

Es el proceso básico mediante el cual una CPU ejecuta instrucciones.

Etapas:

1. **Fetch:** Se busca la instrucción en memoria (usando el PC) y se guarda en el IR.
2. **Decode:** La CPU interpreta la instrucción y prepara los operandos.
3. **Execute:** Se ejecuta la operación (aritmética, acceso a memoria, control, etc.), y el resultado se almacena.

✓ **Importancia:** Cada instrucción sigue este ciclo, y su correcta implementación asegura que el procesador funcione de forma precisa y eficiente.

◆ 2.3.2 Segmentación de instrucciones (Pipeline)

Técnica que divide el ciclo de instrucción en etapas que se ejecutan en paralelo. Similar a una línea de ensamblaje.

Etapas típicas del pipeline:

- Fetch → Decode → Operation → Execute → Write-back

Ventajas:

- Mayor rendimiento.

- Reducción del tiempo total por instrucción.

Desventajas:

- Diseño más complejo y costoso.
 - Mayor latencia si hay interrupciones.
-

◆ **2.3.3 Conjunto de instrucciones**

Conjunto de operaciones que un procesador puede ejecutar. Define cómo interactúa el software con el hardware.

Tipos comunes:

- **Aritméticas:** Suma, resta, etc.
- **Lógicas:** AND, OR, NOT...
- **Transferencia de datos:** MOV, LOAD, STORE...

Formatos:

- Longitud fija, variable, extendida.

RISC vs CISC:

- **RISC:** Instrucciones simples y rápidas. Ej.: ARM, MIPS.
 - **CISC:** Instrucciones más complejas. Ej.: x86.
-

◆ **2.3.4 Modos de direccionamiento y formatos**

Definen cómo se accede a los datos y cómo se estructura una instrucción.

Modos de direccionamiento:

- **Inmediato:** El dato está en la instrucción.
- **Directo:** La dirección del dato está en la instrucción.
- **Indirecto:** La dirección del dato está en un registro.
- **Por registro:** El dato está en un registro.
- **Relativo:** Usado para saltos, se basa en la posición actual.
- **Indexado:** Se calcula con un índice + desplazamiento.

Formatos:

- Longitud fija
 - Longitud variable
 - Extendido
-

◆ 2.4 Casos de estudio de CPUs reales

1. **Intel 8086 (1978):** 16 bits, introdujo la arquitectura x86 y segmentación de memoria.
2. **Intel Pentium (1993):** 32 bits, dos pipelines, ejecución fuera de orden.
3. **AMD Athlon 64 (2003):** Primer CPU de 64 bits para consumidores, controlador de memoria integrado.
4. **Intel Core i7-8700K (2017):** 6 núcleos, alta eficiencia y velocidad.
5. **Apple M1 (2020):** Basado en ARM, alta integración y eficiencia energética, gran impacto en laptops.