

Introducción al diseño de un Procesador

Importancia del diseño de un procesador

- El rendimiento de un computador está determinado por el tiempo que la CPU tarda en ejecutar programas:

$$\text{tiempo de CPU} = (\text{instrucciones por programa}) \times \text{CPI} \times (\text{tiempo de ciclo})$$

- El diseño del procesador determina:
 - La duración del ciclo de reloj
 - Número de ciclos de reloj por instrucción, en promedio
- Comúnmente estos dos factores tienen una relación inversa:
 - Procesador monociclo**
 - 1 ciclo por instrucción
 - Tiempo de ciclo largo
 - Procesador multiciclo**
 - Varios ciclos por instrucción
 - Tiempo de ciclo corto

Metodologías para el diseño de un procesador

- Paso 1: Analizar el repertorio de instrucciones** para obtener los requisitos de la ruta de datos
 - La ruta de datos debe incluir tantos **elementos de almacenamiento** como registros sean visibles por el programador. Además puede tener otros elementos de almacenamiento transparentes.
 - La ruta de datos debe incluir tantos tipos de **elementos operativos** como tipos de operaciones de cálculo se indiquen en el repertorio de instrucciones
 - El significado de cada instrucción vendrá dado por un conjunto de transferencias entre registros. La ruta de datos debe ser capaz de soportar dichas transferencias.
- Paso 2: Establecer la metodología de temporización**
 - Monociclo (CPI = 1):** todas las transferencias entre registros implicadas en una instrucción se realizan en un único ciclo de reloj.
 - Multiciclo (CPI > 1):** las transferencias entre registros implicadas en una instrucción se reparten entre varios ciclos de reloj.

Metodologías para el diseño de un procesador

- Paso 3: Seleccionar el conjunto de módulos** (de almacenamiento, operativos e interconexión) que forman la ruta de datos.
- Paso 4: Ensamblar la ruta de datos** de modo que se cumplan los requisitos impuestos por el repertorio, **localizando los puntos de control**.
- Paso 5: Determinar los valores de los puntos de control** analizando las transferencias entre registros incluidas en cada instrucción.
- Paso 6: Diseñar la lógica de control.**

Introducción

- Analizar el repertorio de instrucciones
 - 68000 - CISC
 - Intel (X86) - CISC
 - MIPS – RISC
 - ARM – RISC

Introducción

- Arquitectura MIPS
 - **Microprocessor without Interlocked Pipeline Stages**
 - Creado en 1981 por Hennessy
 - Producido por Silicon Graphics (SGI)
 - Utilizado por:
 - TiVo
 - Windows CE
 - Cisco routers
 - Nintendo 64
 - Sony PlayStation, PlayStation 2, PlayStation Portable.

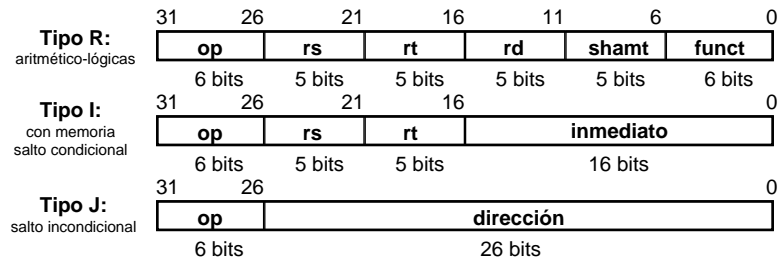
Introducción

- **R2000** (1985): procesador con operaciones multiciclo de multiplicación y división, dentro de un coprocesador matemático, los resultados de estas operaciones eran conseguidos a través de instrucciones particulares. Además, aunque los registros eran de 32 bits podían utilizarse como de 64 para doble precisión.
- **R3000** (1988): añade una cache de 32KB (más tarde 64KB), una MMU para gestionar la memoria virtual, fue el primer procesador MIPS que tuvo éxito en el mercado (se vendieron 1 millón). Por ejemplo, fue utilizado en la **Play Station**, además en los primeros portátiles que utilizaban **Windows CE**.
- **R4000** (1991): aumenta el conjunto de instrucciones a un procesador de 64 bits añadiendo la FPU dentro del chip. Todo esto permite un alto ciclo de reloj (a cambio se reducen las caches a 8KB) y una super-segmentación. **Nintendo64** utiliza una CPU basada en este diseño, así como los primeros routers de **Cisco (36x0 y 7x00)**
- **R8000** (1994): fue el primer procesador super-escalar que podía trabajar a la vez con dos ALUs y dos memorias. Su unidad FPU era perfecta para cálculos científicos, pero duró en el mercado sólo un año.
- **R10000** (1995): tenía caches de 32 KB y su mayor innovación fue la utilización de la ejecución fuera de orden.
- **R20000** (2006): 2 cores ... no ha llegado a producirse

Introducción

- Arquitectura MIPS
 - Todas las instrucciones tienen el mismo tamaño (32 bits)
 - Todas las instrucciones tienen formato similar
 - Código de operación 6 bits
 - Sólo existen 3 formatos de instrucciones
 - Accesos a memoria sencillos
 - Directo registro con desplazamiento

Introducción



El significado de los campos es:

- **op**: identificador de instrucción
- **rs, rt, rd**: identificadores de los registros fuentes y destino
- **shamt**: cantidad a desplazar (en operaciones de desplazamiento)
- **funct**: selecciona la operación aritmética a realizar
- **inmediato**: operando inmediato o desplazamiento en direccionamiento a registro-base
- **dirección**: dirección destino del salto