

Arquitectura de Computadores I

Luis Alberto Chavarría-Zamora

ITCR

lachavarria@tec.ac.cr

11 de marzo de 2020

Contenido

1 ILP

2 Segmentación

3 Referencias

Paralelismo

Técnica de programación e implementación en la que se pretende realizar operaciones simultáneamente, con el fin de reducir tiempos de ejecución, en un procesador.

Existen diferentes tipos de paralelismo:

- Paralelismo a nivel de bit.
- Paralelismo a nivel de instrucciones.
- Paralelismo a nivel de datos.
- Paralelismo a nivel de tareas.
- Paralelismo a nivel de hilos.

Paralelismo a nivel de instrucción (ILP)

ILP

ILP

Segmentación

Pipeline

Referencias

Según Hennessy Patterson:

“...overlap the execution of instructions and improve performance...”

“... is a measure of how many of the instructions in a computer program can be executed simultaneously...”



Paralelismo a nivel de instrucción (ILP)

ILP

ILP

Segmentación

Pipeline

Referencias

Técnica de **paralelismo** basada en la ejecución simultánea de instrucciones.

Posee dos enfoques:

- Paralelismo por hardware (dinámico) - ejecución
- Paralelismo por software (estático) - compilación

Tipos de ILP:

- Segmentación Pipeline
- Ejecución fuera de orden (OoOE)
- VLIW
- CPU's superescalares

ILP - Bloque básico

Corresponde a una sección de código secuencial que no presenta ramificaciones (branches) hasta el final del mismo. En un bloque básico, el flujo de control es **secuencial** y no se detiene hasta terminar el bloque.

ILP - Bloque básico

Corresponde a una sección de código secuencial que no presenta ramificaciones (branches) hasta el final del mismo. En un bloque básico, el flujo de control es **secuencial** y no se detiene hasta terminar el bloque.

```
w = 0;  
x = x + y;  
y = 0;  
if( x > z )  
{  
    y = x;  
    x++;  
}  
else  
{  
    y = z;  
    z++;  
}  
w = x + z;
```

Source Code

```
w = 0;  
x = x + y;  
y = 0;  
if( x > z )
```

```
y = x;  
x++;
```

```
y = z;  
z++;
```

```
w = x + z;
```

Basic Blocks

Tiempo de ejecución en el peor de los casos (WCET)

Tiempo máximo que tarda en ejecutar un código en un hardware específico. Fundamental en sistemas de tiempo real.

- Análisis estático, típicamente.
- x_i puede tener restricciones estructurales y/o dadas por el programador.

Dado un programa con N bloques básicos, donde cada bloque B_i , que posee un tiempo de ejecución máximo c_i , se ejecuta un número de veces x_i , el WCET es:

Tiempo de ejecución en el peor de los casos (WCET)

Tiempo máximo que tarda en ejecutar un código en un hardware específico. Fundamental en sistemas de tiempo real.

- Análisis estático, típicamente.
- x_i puede tener restricciones estructurales y/o dadas por el programador.

Dado un programa con N bloques básicos, donde cada bloque B_i , que posee un tiempo de ejecución máximo c_i , se ejecuta un número de veces x_i , el WCET es:

$$\sum_{i=1}^N c_i \cdot x_i$$

Mejoras ILP mediante Hardware

ILP

ILP

Segmentación

Pipeline

Referencias

Técnicas que permiten hacer *hardware* capaz de procesar mas instrucciones simultáneamente

- Segmentación Pipeline.
- OoOE.
- VLIW.
- Superescalar.
- Especulación*.
- Renombramiento de registros.

Mejoras ILP mediante Hardware

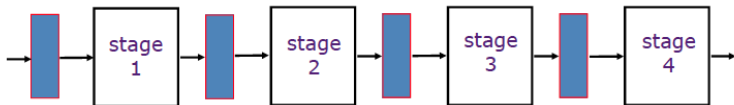
Técnicas que permiten hacer *hardware* capaz de procesar mas instrucciones simultáneamente

- Segmentación Pipeline.
- OoOE.
- VLIW.
- Superescalar.
- Especulación*.
- Renombramiento de registros.

*** Tarea moral: *Spectre and Meltdown.***

Segmentación - Pipeline

Técnica utilizada en el diseño de CPUs para aumentar el rendimiento, mediante la separación de las etapas en el proceso de ejecución de una instrucción.



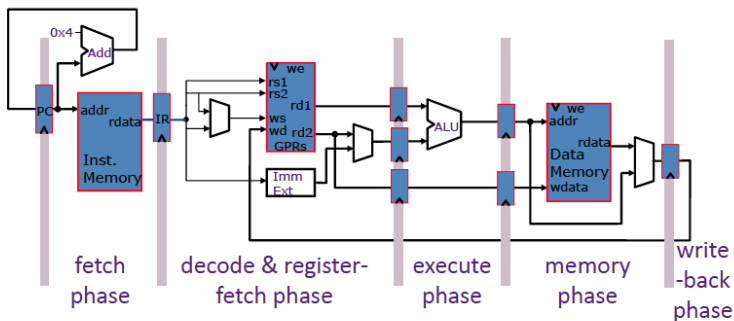
Segmentación - Pipeline

El objetivo del diseñador del pipeline es balancear la longitud de cada etapa del pipeline.

$$\frac{\text{Time per instruction on unpipelined machine}}{\text{Number of pipe stages}}$$

De esta forma se obtiene un *speedup* teórico de N (número de etapas).

Segmentación - MIPS



$$t_C > \max \{t_{IM}, t_{RF}, t_{ALU}, t_{DM}, t_{RW}\}$$

Etapas básicas de un pipeline

ILP

ILP

Segmentación

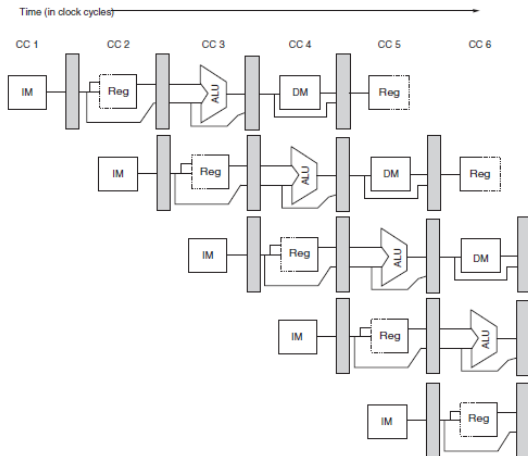
Pipeline

Referencias

- Búsqueda de instrucción (**IF**): Enviar el PC a memoria, traer nueva instrucción, actualizar el PC.
- Decodificación de instrucción (**ID**): "Traducción de instrucción", lectura de registros operandos.
- Ejecución /Dir efectiva (**Ex**): Operaciones en ALU: Memoria efectiva, R-R, R-I.
- Acceso a memoria (**MEM**): Instrucciones L/S.
- Escritura a registros (**WB**): Escritura de resultados R-R o instrucciones L/S a banco de registros.

	Ciclo de reloj								
Número de instrucción	1	2	3	4	5	6	7	8	9
Instrucción i	IF	ID	EX	MEM	WB				
Instrucción i + 1		IF	ID	EX	MEM	WB			
Instrucción i+2			IF	ID	EX	MEM	WB		
Instrucción i+3				IF	ID	EX	MEM	WB	
Instrucción i+4					IF	ID	EX	MEM	WB

Uso de hardware



Ganancia en desempeño

La ganancia en el desempeño debido al pipeline está dada por

$$\begin{aligned}\text{Speedup from pipelining} &= \frac{\text{Average instruction time unpipelined}}{\text{Average instruction time pipelined}} \\ &= \frac{\text{CPI unpipelined} \times \text{Clock cycle unpipelined}}{\text{CPI pipelined} \times \text{Clock cycle pipelined}} \\ &= \frac{\text{CPI unpipelined}}{\text{CPI pipelined}} \times \frac{\text{Clock cycle unpipelined}}{\text{Clock cycle pipelined}}\end{aligned}$$

Ejemplo ARM Cortex R52

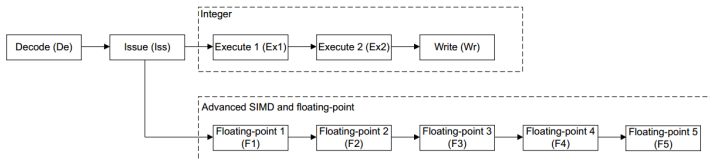
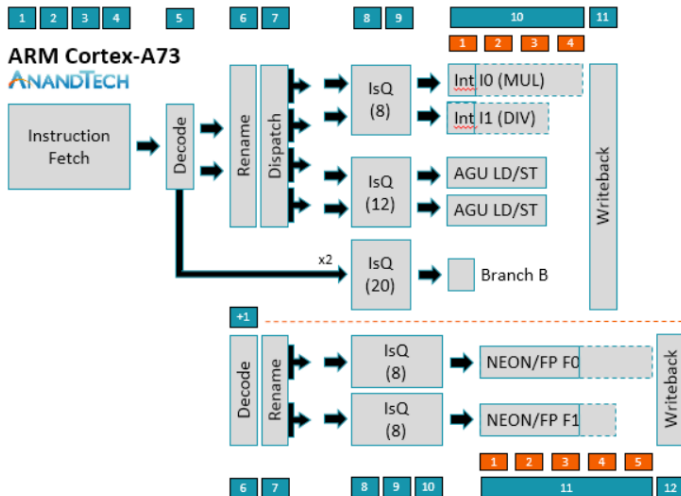


Figure B-1 Cortex-R52 pipeline

Ejemplo ARM Cortex A73



Ventajas y desventajas

Ventajas

- Aumenta el rendimiento del CPU.
- Brinda determinismo en ejecución de instrucciones.

Desventajas

- Etapas e instrucciones lentas afectan el rendimiento general
- Mayor complejidad, más hardware.
- Latencia es ligeramente mayor.
- **Riesgos**

Referencias



J. Hennesy y D. Patterson (2012)

Computer Architecture: A Quantitative Approach. 5th Edition.
Elsevier – Morgan Kaufmann.



J. González y R. García (2019)

Notas de clase de los profesores: Jeferson González y Ronald García.

ARMv8-A Architecture Reference Manual

Intel® 64 and IA-32 architectures software developer's manual
combined volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4