

Arquitectura de computadores II

Paralelismo en instrucciones y procesadores superescalares

Mayo de 2022

1 Segmentación de instrucciones

2 Paralelismo

3 Ejemplos

4 Preguntas

Contenido

1 Segmentación de instrucciones

2 Paralelismo

3 Ejemplos

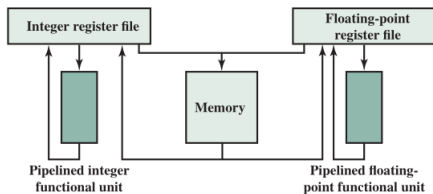
4 Preguntas

Segmentación

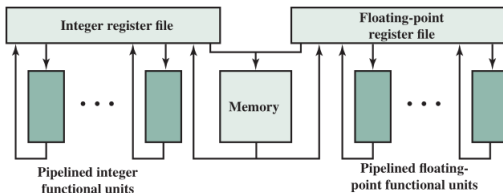
¿Que es segmentación?

1. Las instrucciones pueden ser ejecutadas simultáneamente de acuerdo a sus etapas: FI,DI y EI.
2. Similar a cadena de producción en fábrica
3. Consiste en partir una instrucción en etapas
4. Este proceso es conocido como segmentación de cause o **pipe-line**

Segmentación



(a) Scalar organization



(b) Superscalar organization

Segmentación

Conceptos

Una instrucción puede ser partida en dos etapas

1. Ciclo de captación
2. Ciclo de ejecución

Estos pueden ser partidos en varias etapas, todo el flujo de etapas de la instrucción es llamado **proceso de la instrucción**.

Segmentación

Conceptos

A su vez un ciclo de captación puede ser partido en varias etapas:

1. Captar instrucción (FI)
2. Decodificar instrucción (DI)
3. Calcular operandos (EI): Cálculo de direcciones efectivas y captar operandos

Segmentación

Conceptos

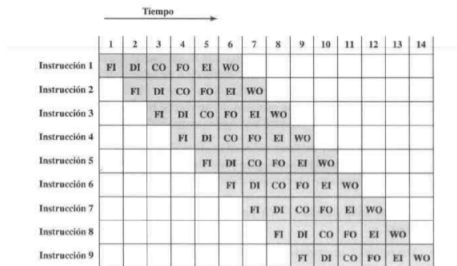
A su vez un ciclo de ejecución puede ser partido en varias etapas:

1. Ejecutar instrucción (EI)
2. Escribir operando (WO)

Segmentación

Conceptos

Debido a que cada etapa sólo requiere algunos elementos del procesador, se puede ejecutar "simultáneamente" más de una instrucción tal y como se muestra en la siguiente figura:



Segmentación

Conceptos

Sin embargo, existen problemas cuando se presentan saltos (llamado a subrutina). Como se puede observar en la siguiente figura, donde existe un salto en la instrucción 3 a la instrucción 15:

	Tiempo →							← Penalización debida al salto						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instrucción 1	FI	DI	CO	FO	EI	WO								
Instrucción 2		FI	DI	CO	FO	EI	WO							
Instrucción 3			FI	DI	CO	FO	EI	WO						
Instrucción 4				FI	DI	CO	FO							
Instrucción 5					FI	DI	CO							
Instrucción 6						FI	DI							
Instrucción 7							FI							
Instrucción 15								FI	DI	CO	FO	EI	WO	
Instrucción 16									FI	DI	CO	FO	EI	WO

Segmentación

Conceptos

Para enfrentar los problemas relacionados con los saltos, se diseñan estrategias para asegurar el flujo estable de las instrucciones, entre las cuales se encuentran:

1. Flujos múltiples
2. Precaptar el salto
3. Buffer de bucles
4. Predicción de saltos
5. Salto retardado

Segmentación

Flujos múltiples

En esta estrategia se intenta resolver el problema de que cuando hay un salto, se debe escoger alguna de las dos instrucciones y se puede hacer una elección equivocada, una solución es duplicar el proceso y hacer que se ejecuten las dos instrucciones en todos los caminos posibles. El problema que se presenta retardos por el proceso adicional que se realiza.

Segmentación

Precaptar el salto

Cuando se identifica un salto, se precapta la instrucción destino del salto y la siguiente después del salto, se almacena esta ultima y se ejecuta en el momento en que se ha finalizado el salto.

Buffer de bucles

Se almacenan las n instrucciones captadas anteriormente.

Segmentación

Predicción de saltos

Es intentar predecir si un salto se va a producir

1. Predecir que nunca se salta
2. Predecir que siempre se salta
3. Predecir según código de operación
4. Conmutador saltar/no saltar
5. Tabla de historia de saltos

Segmentación

Salto retardado

Se reordenan las instrucciones de tal forma, los saltos se presenten después del momento deseado.

Contenido

1 Segmentación de instrucciones

2 Paralelismo

3 Ejemplos

4 Preguntas

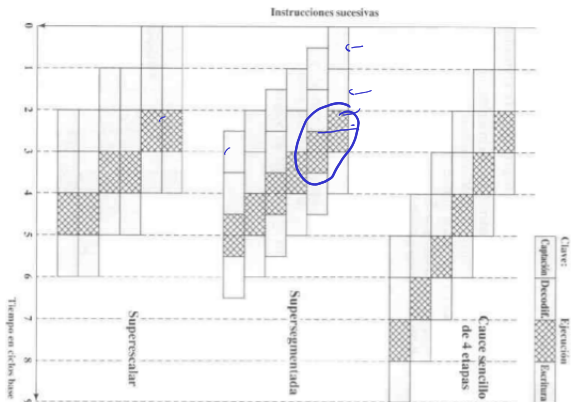
Superescalar vs supersegmentado

- Un procesador superescalar utiliza múltiples causas de instrucciones
- Un procesador supersegmentado, aprovecha que algunas etapas de las instrucciones requieren poco tiempo de realización, por lo que dobla la velocidad de procesamiento

Paralelismo

Superescalar vs supersegmentado

En la siguiente figura se puede observar la diferencia entre superescalar y supersegmentado:



Paralelismo

Limitaciones superescalar

- Dependencia de datos
- Dependencia de procedimientos
- Conflicto recursos
- Dependencia en salidas
- Antidependencia

$$R2 = R2 + R1$$
$$R3 = R2 + R3$$

$I_2: R3 = R3 + 1$
 $I_2: R4 = R3 + R5$
 $I_3: R3 = R2 + R8$
 $+4 = R9 = R3 + R9$

Paralelismo

Limitaciones superescalar

Dependencia de datos

```
ADD EAX , ECX  
MOV EBX , EAX
```

Paralelismo

Limitaciones superescalar

Dependencia de las salidas (write after write [WAW] dependency)

```
I1 : R3 ← R3 op R5  
I2 : R1 ← R3 + 1 ←  
I3 : R2 ← R1 + 1 ←  
I4 : R7 ← R3 op R4
```

Paralelismo

Limitaciones superescalar

Antidependencia (write after read [WAR] dependency)

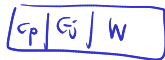
I1 : R3 \leftarrow R3 op R5
I2 : R1 \leftarrow R3 + 1
I3 : R3 \leftarrow R5 + 1 \leftarrow
I4 : R7 \leftarrow R3 op R4

Paralelismo

Nivel de paralelismo

1. Paralelismo a nivel de instrucción: Las instrucciones son independientes y pueden ser ejecutadas en paralelo
2. Paralelismo a nivel de máquina: La CPU es capaz de tomar ventaja del paralelismo en las instrucciones.

Paralelismo



Políticas de emisión de instrucciones

Para realizar el diseño de políticas de emisión de instrucciones se tienen en cuenta las siguientes ordenaciones:

- Orden en que se captan las instrucciones
- Orden en que se ejecutan las instrucciones
- El orden en que las instrucciones actualizan registros y la memoria

Paralelismo

Políticas de emisión de instrucciones

En base a lo anterior se crean las siguientes políticas de instrucciones

- **Emisión en orden y finalización en orden:** Se ejecutan las instrucciones en el orden exacto de llegada
- **Emisión en orden y finalización desordenada:** Se toma en cuenta la dependencia entre salidas, por ejemplo que un recurso sea requerido por una instrucción y otra lo solicite, esta debe esperar a que el recurso esté disponible
- **Emisión desordenada y finalización desordenada:** Con la emisión en orden sólo se codifica hasta que exista un punto de conflicto, en este caso se siguen codificando instrucciones y ejecutando.

Paralelismo

$I_1 \quad R3 = R3 \text{ op } R5$
 $I_2 \quad R1 = R3 + 1$
 $I_3 \quad R3 = R5 \text{ op } R4$
 $R8$

Renombramiento de registros

Debido a la emisión y finalización desordenada de instrucciones. Debido a que pueden existir conflictos una estrategia de solución es la duplicación de recursos, la técnica es conocida como **renombramiento de registros**. Consiste en que se asigna dinámicamente los registros a las instrucciones en diversos momentos de tiempo, con esto se evita problemas en el uso de estos recursos.

Paralelismo

Predicción de saltos

Debido a la segmentación de las instrucciones, existen problemas en el cause de las instrucciones debido a los saltos, en procesador superescalares la estrategia utilizada es **salto retardado**

Paralelismo

Ejecución superescalar

Primero se debe analizar la dependencia entre instrucciones para enviarlas a una ventana de ejecución. Cuando se ingresan a esta ventana estas dependencias no son secuenciales (no se ejecutan en el orden de llegada), y son ejecutadas de acuerdo a la pólitica de ejecución que se utilice.

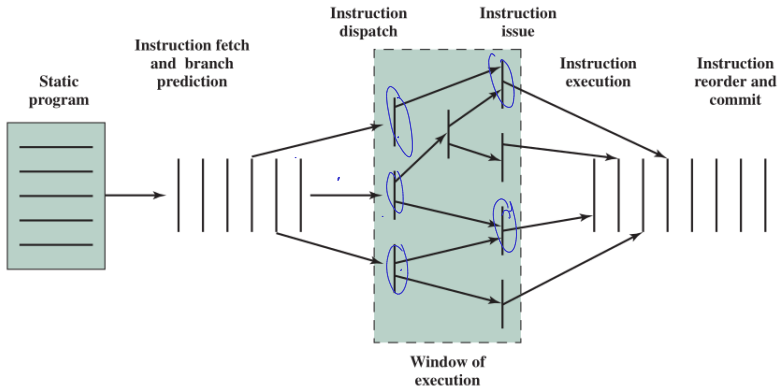
Paralelismo

Implementación superescalar

1. Estrategias predictivas para la captación de instrucciones
2. Lógica para determinar dependencias
3. Recursos para la ejecución de instrucciones múltiples
4. Mecanismos para un correcto orden de los procesos

Paralelismo

Superscalar



Contenido

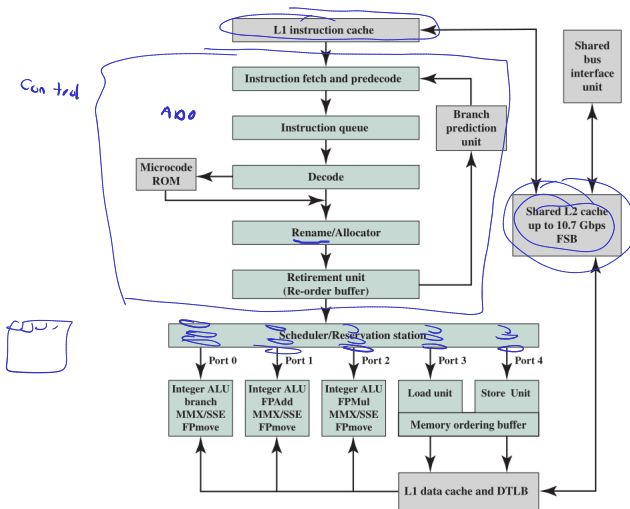
1 Segmentación de instrucciones

2 Paralelismo

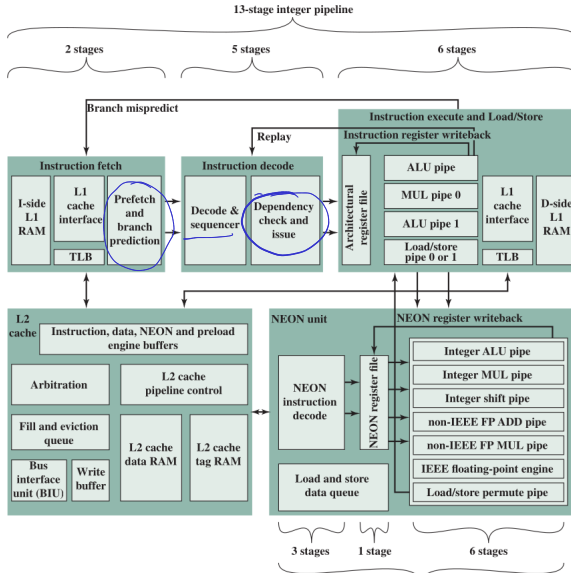
3 Ejemplos

4 Preguntas

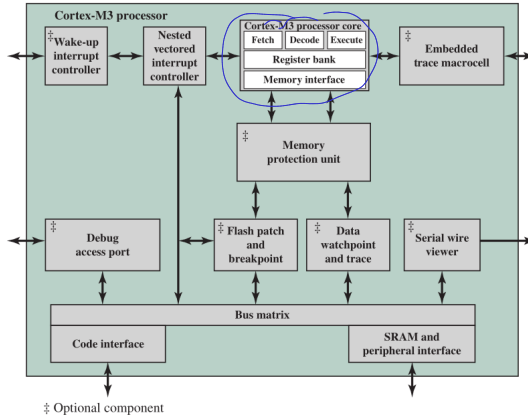
Ejemplos: Intel Core



Ejemplos:ARM Cortex-A8



Ejemplos:ARM Cortex-M3



Contenido

1 Segmentación de instrucciones

2 Paralelismo

3 Ejemplos

4 Preguntas

Preguntas

Pregunta

¿Cuál es la característica esencial del enfoque superescalar en el diseño de procesadores?

Respuesta

Un procesador superescalar es aquel en el que se utilizan múltiples pipelines de instrucciones independientes. Cada canal consta de varias etapas, de modo que cada canal puede manejar varias instrucciones a la vez. Las canalizaciones múltiples introducen un nuevo nivel de paralelismo, permitiendo procesar múltiples flujos de instrucciones a la vez.

Preguntas

Pregunta

¿Qué es el paralelismo a nivel de instrucción?

Respuesta

Es la ejecución simultánea de varias instrucciones, de tal manera se conserve la consistencia del programa en ejecución.

reordenamiento

Preguntas

Pregunta

¿Cuál es la distinción entre el paralelismo a nivel de instrucción y el paralelismo de máquina?

Respuesta

El paralelismo a nivel de instrucción existe cuando las instrucciones de una secuencia son independientes y, por lo tanto, pueden ejecutarse en paralelo mediante el solapamiento. El paralelismo de máquina es una medida de la capacidad del procesador para aprovechar el paralelismo a nivel de instrucción.

El paralelismo de la máquina viene determinado por el número de instrucciones que pueden obtenerse y ejecutarse al mismo tiempo (el número de conductos paralelos) y por la velocidad y sofisticación de los mecanismos que utiliza el procesador para encontrar instrucciones independientes.

Preguntas

Pregunta

Enumere y defina brevemente tres tipos de políticas de emisión de instrucciones superescalares.

Respuesta

1) (In-order issue) Emisión en orden con finalización en orden:
Emitir las instrucciones en el orden exacto que se conseguiría con la ejecución secuencial y escribir los resultados en ese mismo orden.

Preguntas

Pregunta

Enumere y defina brevemente tres tipos de políticas de emisión de instrucciones superescalares.

Respuesta

2) (In-order issue with out-of-order completion) Problema de orden con finalización fuera de orden: Emitir las instrucciones en el orden exacto que se conseguiría con la ejecución secuencial, pero permitir que las instrucciones se ejecuten hasta su finalización fuera de orden.

Preguntas

Pregunta

Enumere y defina brevemente tres tipos de políticas de emisión de instrucciones superescalares.

Respuesta

3) (Out-of-order issue with out-of-order completion) Problema fuera de orden con finalización fuera de orden: El procesador tiene una capacidad de anticipación que le permite identificar instrucciones independientes que pueden ser llevadas a la etapa de ejecución. Las instrucciones se emiten sin tener en cuenta el orden original del programa. Las instrucciones también pueden ejecutarse hasta su finalización fuera de orden.

Preguntas

Pregunta

¿Cuáles son los elementos clave de la organización de un procesador superescalar?

Respuesta

1) Estrategias de obtención de instrucciones que obtienen simultáneamente múltiples instrucciones, a menudo prediciendo los resultados de las instrucciones de bifurcación condicional y obteniéndolas más allá. Estas funciones requieren el uso de múltiples etapas de búsqueda y decodificación en el pipeline, así como una lógica de predicción de bifurcación.

Preguntas

Pregunta

¿Cuáles son los elementos clave de la organización de un procesador superescalar?

Respuesta

2) Lógica para determinar las verdaderas dependencias que implican valores de registro, y mecanismos para comunicar estos valores a donde se necesitan durante la ejecución. 3) Mecanismos para iniciar, o emitir, múltiples instrucciones en paralelo.

Preguntas

Pregunta

¿Cuáles son los elementos clave de la organización de un procesador superescalar?

Respuesta

4) Recursos para la ejecución en paralelo de múltiples instrucciones, incluyendo múltiples unidades funcionales en cadena y jerarquías de memoria capaces de atender simultáneamente múltiples referencias de memoria. 5) Mecanismos para comprometer el estado del proceso en el orden correcto.

Preguntas

¿Preguntas?

Siguiente tema:
Unidad de control