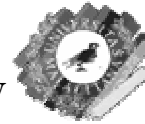




Tema 5: Procesadores superescalares y VLIW



- Introducción
- Superescalares
- VLIW
- Especulación
- Ejemplos

Dept. Arquitectura de Computadores

Arquitectura de Computadores

Universidad de Málaga

5.1 Introducción Sistemas con emisión múltiple

- Objetivo:
 - ✓ Reducir el CPI por debajo de la unidad
- Implicaciones
 - ✓ Emitir y completar más de una instrucción por ciclo
 - ✓ Modificar el pipeline
 - ✓ Explotar paralelismo nivel de instrucción (ILP)
 - Más instrucciones independientes se pueden ejecutar en paralelo
- Dos esquemas:
 - ✓ Superescalares
 - La unidad de control determina qué instrucciones se pueden emitir en un ciclo
 - Emisión dinámica: el nº de instrucciones emitidas por ciclo es variable
 - ✓ Very Long Instruction Word (VLIW)
 - El compilador empaqueta las instrucciones que se pueden emitir en un ciclo
 - Emisión estática: el nº de instrucciones emitidas es fijo (n)

Introducción

Superescalares

VLIW

Especulación

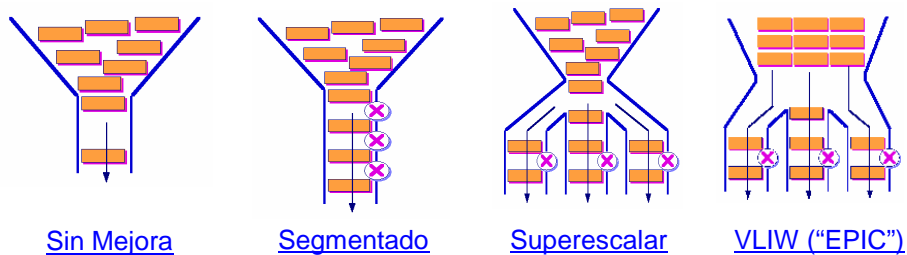
Ejemplos

2

5.1 Introducción

Resumen de técnicas

- Segmentación y supersegmentación
- Planificación SW o HW (Tomasulo) en la ejecución de instrucciones
- Superescalares
- VLIW



Introducción

Superescalares

VLIW

Especulación

Ejemplos

3

5.2 Superescalares

Ejemplo

- Características:
 - ✓ Procesador superescalar de factor 2: tiene dos cauces
 - ✓ Restricciones:
 - Cauce 1: para instrucciones enteras, carga, almacenamiento o salto
 - Cauce 2: para instrucciones punto flotante
 - ✓ Dos instrucciones que cumplan las condiciones se pueden emitir, ejecutar y completar al mismo tiempo.

Instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
sub r4,r3,r5	BI	DI	EJ	M	PE												
addd f0,f8,f6	BI	DI	S1	S2	S3	S4	M	PE									
sw 0(r7), r5		BI	DI	EJ	M	PE											
subd f8, f6, f10		BI	DI	S1	S2	S3	S4	M	PE								
ld f4,0(r2)			BI	DI	EJ	M	PE										
addd f2,f12,f6			BI	DI	S1	S2	S3	S4	M	PE							
slli r5, r6, #2				BI	DI	EJ	M	PE									
addd f6, f6, f10				BI	DI	S1	S2	S3	S4	M	PE						

Introducción

Superescalares

VLIW

Especulación

Ejemplos

4

5.2 Superescalares

Ejemplo

- Necesidad de leer dos instrucciones por ck → bus con cache l más ancho
 - ✓ Con la cache de datos no hace falta: las instr. PF no acceden a mem.
- Si no es posible encontrar dos instrucciones para emitir:
 - ✓ Se emite sólo una, y en el otro cauce se pierde un ck
 - ✓ Si tenemos planificación SW el compilador debe reordenar las instrucciones para emparejar instrucciones enteras con instrucciones PF

Instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
sub r4,r3,r5	BI	DI	EJ	M	PE												
addd f0,f8,f6	BI	DI	S1	S2	S3	S4	M	PE									
sw 0(r4), r5		BI	DI	EJ	M	PE											
		X	X	X	X	X	X	X	X								
slli r5, r6, #2			BI	DI	EJ	M	PE										
addd f2,f12,f6			BI	DI	S1	S2	S3	S4	M	PE							
ld f4,0(r2)				BI	DI	EJ	M	PE									
addd f6, f6, f10				BI	DI	S1	S2	S3	S4	M	PE						

Introducción

Superescalares

VLIW

Especulación

Ejemplos

5

5.2 Superescalares

Ejemplo

- Ventajas de tener un cauce para instr. enteras y otro para PF:
 - ✓ Se simplifica la implementación del HW
 - ✓ No hay riesgo estructural en DI ni en PE ya que usan bancos de reg. distintos
 - ✓ No hay dependencias entre instrucciones enteras y PF **SALVO EN:**
 - Load/Store en registros PF e instrucciones de copia de registros pf/int (movfp2i y movi2fp)
 - Para estos casos hay que suponer que el banco de registros PF tiene
 - Tres puertos de lectura → elimina la posibilidad de riesgo estructural en DI

Instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
sub r4,r3,r5	BI	DI	EJ	M	PE												
addd f0,f8,f6	BI	DI	S1	S2	S3	S4	M	PE									
sd 0(r4), f8		BI	DI	EJ	M	PE											
subd f8, f6, f10		BI	DI	S1	S2	S3	S4	M	PE								
ld f4,0(r2)			BI	DI	EJ	M	PE										
addd f2, f4, f6			BI	DI			S1	S2	S3	S4	M	PE					
slli r2, r3, #2				BI	DI	EJ	M	PE									
addd f6, f6, f12				BI			DI	S1	S2	S3	S4	M	PE				

Introducción

Superescalares

VLIW

Especulación

Ejemplos

6

5.2 Superescalares

Consideraciones

■ Inconvenientes

- ✓ Necesitamos cortocircuitos no solo dentro del mismo cauce sino también entre cauces distintos
- ✓ Riesgos estructurales: segmentar unidades funcionales PF y varios puertos de escritura en el banco de registros PF
- ✓ El retardo del salto (latencia 1ck) afecta no a 1 sino a 3 instrucc.

Instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
sub r4,r3,r5	BI	DI	EJ	M	PE												
multd f0,f8,f6	BI	DI	P1	P2	P3	P4	P5	M	PE								
ld f6, 0(r3)		BI	DI	EJ	M	PE											
subd f8, f8, f10		BI	DI	S1	S2	S3	S4	M	PE								
lw r4,0(r2)			BI	DI	EJ	M	PE										
addd f2,f12,f8			BI	DI	S1	S2	S3	S4	M	PE							
ld f2, 0(r4)				BI	DI	EJ	M	PE									
addd f6, f6, f10				BI	DI	S1	S2	S3	S4	M	PE						

Introducción

Superescalares

VLIW

Especulación

Ejemplos

7

5.2 Superescalares

Con planificación SW

■ Desenrollando 5 veces un bucle y planificando para el superescalar

```
ini:  ld    f0, 0(r1)
      addd  f4, f0, f2
      sd    0(r1), f4
      subi  r1, r1, #8
      bnez  r1, ini
```



□ 12ck / 5elementos = 2.4ck/elemento

- En segmentado y desenrollando 4 veces obteníamos 3.5ck/elem.

□ Rendimiento limitado por la imposibilidad de balancear todas las instrucciones enteras y PF

- Hay muchas instrucciones enteras sin pareja

Instrucción entera	Instrucción PF	ck
ini: ld f0,0(r1)		1
ld f6,-8(r1)		2
ld f10,-16(r1)	addd f4,f0,f2	3
ld f14,-24(r1)	addd f8,f6,f2	4
ld f18,-32(r1)	addd f12,f10,f2	5
sd 0(r1),f4	addd f16,f14,f2	6
sd -8(r1),f8	addd f20,f18,f2	7
sd -16(r1),f12		8
subi r1,r1,#40		9
sd 16(r1),f16		10
bnez r1,ini		11
sd 8(r1),f20		12

Introducción

Superescalares

VLIW

Especulación

Ejemplos

8

5.2 Superescalares Con planificación HW

- Algoritmo de Tomasulo emitiendo dos instrucciones por ck a las ER

Instrucción	Estado de las Instrucciones			
	It.	Emisión	Ejecución	Escritura
ld f0, 0(r1)	1	1	2	3
add f4, f0, f2	1	1	4-7	8
sd 0(r1), f4	1	2	9	10
subi r1, r1, #8	1	3	4	5
bnez r1, ini	1	4	6	7
ld f0, 0(r1)	2	5	6	7
add f4, f0, f2	2	5	8-11	12
sd 0(r1), f4	2	6	13	14
subi r1, r1, #8	2	7	8	9
bnez r1, ini	2	8	10	11

- Pocas emisiones dobles (sólo una operación PF por iteración)
- Se necesitan 4ck por elemento (no hemos quitado subi ni bnez)

Introducción Superescalares VLIW Especulación Ejemplos 9

5.2 Superescalares con Tomasulo Sin limitaciones en el tipo de instrucciones

- Se emiten dos instrucciones por ciclo (de cualquier tipo)

Instrucción	Estado de las Instrucciones			
	It.	Emisión	Ejecución	Escritura
ld f0, 0(r1)	1	1	2	3
add f4, f0, f2	1	1	4-7	8
sd 0(r1), f4	1	2	9	10
subi r1, r1, #8	1	2	3	4
bnez r1, ini	1	3	5	6
inst i+1	1	3	X	X
ld f0, 0(r1)	2	4	5	6
add f4, f0, f2	2	4	7-10	11
sd 0(r1), f4	2	5	12	13
subi r1, r1, #8	2	5	6	7
bnez r1, ini	2	6	8	9

- El salto se emite en el ciclo 3. La BTB predice saltar y emites ld en ck 4

Introducción Superescalares VLIW Especulación Ejemplos 10

5.3 VLIW

- Problema del diseño superescalar: elevado coste HW:
 - ✓ Necesidad de comprobar los CO y operandos de las “n” instr. a ser emitidas
 - ✓ Lógica de detenciones y cortocircuitos muy complicada
- Alternativa: VLIW (Very Long Instruction Word)
 - ✓ El compilador empaqueta múltiples operaciones en una única instrucción



Introducción Superescalares **VLIW** Especulación Ejemplos 11

5.3 VLIW

Ejemplo

- Suponemos 2 UF PF, 2 UF memoria y 1 UF entera/salto
 - ✓ Hasta cinco operaciones por instrucción en el caso mejor
 - ✓ Si cada operación se codifica con 16-32 bits → Instrucciones de 80-160bits
- Desenrollando 7 veces el código ejemplo de la transparencia 8:

Mem1	Mem2	PF1	PF2	Int/salto
ld f0,0(r1)	ld f6,-8(r1)			
ld f10,-16(r1)	ld f14,-24(r1)			
ld f18,-32(r1)	ld f22,-40(r1)	add f4,f0,f2	add f8,f6,f2	
ld f26,-48(r1)		add f12,f10,f2	add f16,f14,f2	
		add f20,f18,f2	add f24,f22,f2	
sd 0(r1),f4	sd -8(r1),f8	add f28,f26,f2		
sd -16(r1),f12	sd -24(r1),f16			subi r1,r1,#56
sd 24(r1),f4	sd 16(r1),f8			
sd 8(r1),f4				bnez r1,ini

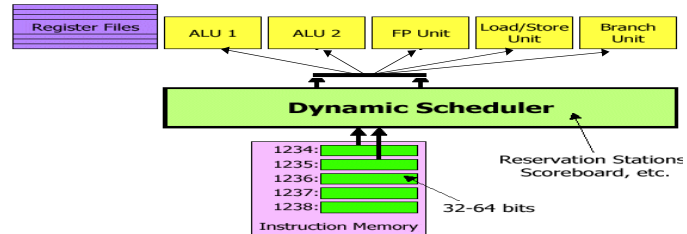
- 9ck/7ele=1.29ck/elemento 23inst/45huecos=51% de eficiencia

Introducción Superescalares **VLIW** Especulación Ejemplos 12

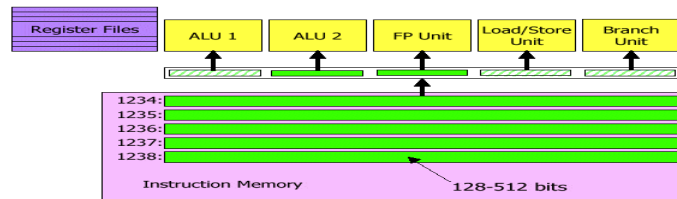
5.3 VLIW

VLIW vs Superescalar

Superescalar



VLIW



Introducción

Superescalares

VLIW

Especulación

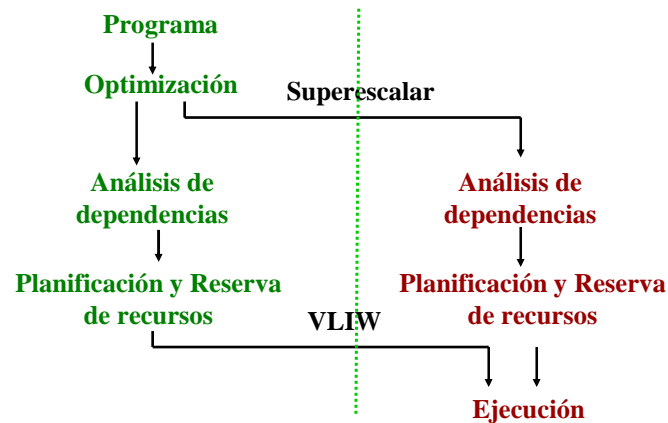
Ejemplos

13

5.3 VLIW

VLIW vs Superescalar

- Diferente división de tareas entre **Software** y **Hardware**



Introducción

Superescalares

VLIW

Especulación

Ejemplos

14

5.3 VLIW

Ventajas e inconvenientes

■ Ventajas

- ✓ La planificación la realiza el compilador
 - Tiene una visión global del código y la planificación sólo se hace una vez en tiempo de compil.
- ✓ Menos requerimientos hardware
 - Menos lógica en la unidad de control. Más silencio para unidades funcionales de ejecución
 - Mayor frecuencia de reloj
- ✓ Arquitectura popular para sistemas empujados (DSP, multimedia)

■ Inconvenientes

- ✓ Tamaño elevado del código por huecos no rellenos
- ✓ Imposible mantener la compatibilidad binaria si cambias la arquitectura
- ✓ Difícil encontrar suficiente ILP para mantener ocupadas todas las UFs
- ✓ Compilador complicado
 - Dependencias de datos en memoria irresolubles en tiempo de compilación
 - Problemas con los riesgos de control por la implementación de predicción estática de salto

Introducción

Superescalares

VLIW

Especulación

Ejemplos

15

5.4 Especulación

■ Objetivo:

- ✓ Extraer más paralelismo en presencia de riesgos de control

■ Hasta ahora...

- ✓ La planificación SW se hacía entre instrucciones del mismo bloque básico
- ✓ Hemos ampliado el tamaño de los bloques básicos mediante
 - Desenrollamiento de bucles
 - Instrucciones condicionales que transforman las dependencias de control en depend. de datos

■ Con especulación...

- ✓ Vamos a mover instrucciones "a través de los saltos"
- ✓ Es decir: planificación de instrucciones entre bloques básicos
- ✓ Vamos a ejecutar instrucciones que a lo mejor no deberían ejecutarse
 - No deben modificar la semántica del programa si se ejecutan cuando no deben
 - No deben provocar excepciones

Introducción

Superescalares

VLIW

Especulación

Ejemplos

16

5.4 Especulación

Ejemplo

- Función: if (A=0) A=B; else A=A+4;

✓ La variable **A** está almacenado en 0(r3) y la variable **B** en 0(r2)

```
lw  r1,0(r3); Carga A
DETENCION
bnez r1,L1 ; test A
lw  r1,0(r2); cuerpo del if
j   L2     ; salta el else
L1: addi r1,r1,#4; el else
L2: sw 0(r3),r1 ; guarda A
```

```
lw  r1,0(r3) ; Carga A
lw  r14,0(r2); Carga espec.
bnez r1,L1   ; salto
j   L2       ;salta el else
L1: addi r14,r1,#4; el else
L2: sw 0(r3),r14 ; guarda A
```

No hemos modificado la semántica

- ✓ Se ha hecho una carga de B de forma especulativa en r14 (otro reg. nuevo)
 - Si bnez no salta → especulación correcta (efectivamente era necesario cargar B)
 - Si beqz salta → especulación incorrecta (no habia que cargar B y de hecho r14 se rescribe)
- ✓ Una instrucción incorrectamente especulada no debe provocar excepciones
 - Si ni siquiera debería de haberse ejecutado, no podemos permitir que provoque excepciones

Introducción

Superescalares

VLIW

Especulación

Ejemplos

17

5.4 Especulación

Soluciones: A. Bits venenosos

- Se añade un bit venenoso al CO de cada instrucción y a cada registro
 - ✓ El compilador pone a 1 el bit venenoso de las instrucciones especulativas
 - ✓ Si una instrucción especulativa produce excepción → envenena reg. destino
 - ✓ Las instrucciones especulativas que leen reg envenenados propagan el veneno al registro en el que escriben
 - ✓ Si una instrucción NO especulativa...
 - Escribe un registro envenenado → resetea el bit venenoso del registro a 0 (cura el reg). La instrucción que desencadenó el proceso estaba incorrectamente especulada
 - Lee un reg envenenado → la instrucción especulativa que desencadenó el proceso esta correctamente especulada → hay que atender la excepción retornando el estado del procesador a la situación en que estaba cuando esa instrucción no se habia completado

```
lw  r1,0(r3) ; Carga A
lw* r14,0(r2); Carga espec.
bnez r1,L1   ; salto
j   L2       ;salta el else
L1: addi r14,r1,#4; el else
L2: sw 0(r3),r14 ; guarda A
```

- Si lw* produce excepción → envenena r14
- Si addi se ejecuta → se cura r14
- Si sw lee un r14 envenenado se atiende la excepción de lw* (y si no, no se atiende).

Introducción

Superescalares

VLIW

Especulación

Ejemplos

18

5.4 Especulación

Soluciones: B. Buffer de futuro

■ El resultado de una instrucción especulativa...

- ✓ No se escribe en registro por si acaso está incorrectamente especulada
- ✓ Por el contrario, se escribe en un buffer de futuro
 - Si la especulación resulta ser correcta → volcado del buffer a los registros
 - Si la especulación resulta ser incorrecta → borras el buffer (como si no la hubieras ejecutado)

■ Ejemplo

```
lw    r1,0(r3) ; Carga A
lw+   r1,0(r2) ; Carga espec.
bnez  r1,L1    ; salto
j     L2       ;salta el else
L1:   addi r1,r1,#4; el else
L2:   sw 0(r3),r1 ; guarda A
```

- Ya no necesitamos el renombre con r14
- lw+ realmente no escribe en r1 sino en buffer
- Dependiendo de si el salto es o no efectivo vamos o no a usar la información del buffer

- ✓ Si el salto no es efectivo → especulación correcta → r1=buffer
- ✓ Si el salto es efectivo → especulación incorrecta → borrar el buffer

Introducción

Superescalares

VLIW

Especulación

Ejemplos

19

5.4 Especulación

Soluciones: C. Tomasulo con especulación

■ Combina tres conceptos importantes:

- ✓ Si una instrucción especulativa provoca excepción, se atiende cuando estas seguro de que está correctamente especulada (similar a bits venenosos)
- ✓ Una instrucción especulativa escribe en registros/memoria cuando estás seguro de que no está incorrectamente especulada (buffer de futuro)
- ✓ Planificación dinámica (Algoritmo de Tomasulo)

■ Ideas clave

- ✓ Las instrucciones se emiten en orden
- ✓ Las instrucciones se ejecutan tan pronto estén sus operandos disponibles, es decir, fuera de orden
- ✓ Las instrucciones se completan (terminan) en orden en la etapa "commit"
 - Cuando una instrucción termina estás seguro de que la anterior ya ha terminado y no va a provocar ninguna excepción.

■ Implementado en todos los procesadores superescalares actuales

- ✓ Del Pentium Pro al Pentium 4, AMD K5-K8, MIPS R10K, 12K, Alpha 21264.

Introducción

Superescalares

VLIW

Especulación

Ejemplos

20

5.4.1 Tomasulo con especulación

Etapas en la ejecución

- **Emisión:**
 - ✓ Si hay ER disponible y un hueco en el BR →
 - Se emite la instrucción y los operandos disponibles (desde reg. o BR) a la ER
 - Si hay operandos no disponibles, en la ER se apunta al la entrada del BR que generará el dato
 - Se ocupa una entrada del buffer (BR). Detención en caso de no disponer de hueco en ER o BR.
- **Ejecución:**
 - ✓ Cuando están disponibles todos los operandos y hay UF libre → Ocupamos la UF (ejecución) y se libera la ER
- **Escritura:**
 - ✓ Cuando la instrucción termina, se libera UF y el resultado se vuelca al CDB
 - Las ER que necesitan el resultado y la entrada del BR que contiene la instr. capturan el resultado
- **Graduación (commit):**
 - ✓ Cuando la escritura ha terminado y se han graduado las instrucc. previas
 - ✓ Instr. no de salto: escribe en reg. o mem y atiende excepciones en su caso
 - ✓ Instr. de salto con predicción incorrecta:
 - Se vacía el BR y se actualiza la información de predicción de salto

Introducción

Superescalares

VLIW

Especulación

Ejemplos

23

5.4.2 Ejemplo

- Ejemplo: ejecución del siguiente código:

```
1 ld      f6, 34(r2)
2 ld      f2, 45(r3)
3 multd   f0, f2, f4
4 subd    f8, f6, f2
5 divd    f10, f0, f6
6 addd    f6, f8, f2
```

Unidades funcionales:

- Un sumador PF. Consume 2ck
- Un multiplicador PF. Consume 3ck
- Un divisor PF: Consume 4ck
- Una unid. entera: Consume 1ck
- Una unid. de mem: Consume 1ck

- ✓ Buffer de reordenación con 10 entradas
- ✓ 3 estaciones de reserva asociadas al sumador PF (add1, add2, y add3)
- ✓ 2 estaciones de reserva asociadas al multiplicador/divisor PF (mult1 y mult2)
- ✓ 6 estaciones/buffers para gestionar las cargas (ld) (load1 a load6)
- ✓ 3 estaciones/buffers para gestionar los almacenamientos (sd) (store1 a store3)
- ✓ A las estaciones de reserva también se les llama **ventana de instrucciones**

Introducción

Superescalares

VLIW

Especulación

Ejemplos

24

5.4.2 Ejemplo

- Tabla completa del estado de las instrucciones y ciclo en que se completa cada etapa

Instrucción	Estado de las Instrucciones				
	Búsq.	DI/Em.	Ejecución	Escritura	Commit
ld f6, 34(r2)	1	2	3	4	5
ld f2, 45(r3)	2	3	4	5	6
multd f0, f2, f4	3	4	6-8	9	10
subd f8, f6, f2	4	5	6-7	8	11
divd f10, f0, f6	5	6	10-13	14	15
addd f6, f8, f2	6	7	9-10	11	16

- Se evitan los riesgos WAR y WAW.
 - ✓ WAR por las ER y porque se escribe en registros en orden. WAW igual
- Excepciones precisas

Introducción Superescalares VLIW Especulación Ejemplos 25

5.4.2 Ejemplo

ck	BI	DI	Vload	Vsum	Vm/d	Mem	SumPF	MulPF	DivPF	BR	Commit
1	1										
2	2	1	1								
3	3	2	2			1					
4	4	3			3	2				1	
5	5	4		4	3					2	1
6	6	5			5		4	3			2
7		6		6	5		4	3			
8				6	5			3		4	
9					5		6			3,4	
10							6		5	4	3
11									5	6	4
12									5	6	
13									5	6	
14										5,6	
15										6	5
16											6

addd usa el resultado de subd aunque no se ha completado.

1	ld	f6, 34(r2)
2	ld	f2, 45(r3)
3	multd	f0, f2, f4
4	subd	f8, f6, f2
5	divd	f10, f0, f6
6	addd	f6, f8, f2

Introducción Superescalares VLIW Especulación Ejemplos 26

5.4.2 Ejemplo Ciclo 2

Estado de los registros	
Campo	BR
F0	
F2	
F4	
F6	#1
F8	
F10	
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	si	LOAD	34	r2			#1
Load2	no						
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	no						
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	si	ld f6,34(r2)	emitido	F6	
2	no				
3	no				
4	no				
5	no				
6	no				

Introducción

Superescalares

VLIW

Especulación

Ejemplos

27

5.4.2 Ejemplo Ciclo 3

Estado de los registros	
Campo	BR
F0	
F2	#2
F4	
F6	#1
F8	
F10	
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	si	LOAD	45	r3			#2
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	no						
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	si	ld f6,34(r2)	ejecución	F6	
2	si	ld f2,45(r3)	emitido	F2	
3	no				
4	no				
5	no				
6	no				

Introducción

Superescalares

VLIW

Especulación

Ejemplos

28

5.4.2 Ejemplo Ciclo 5

Estado de los registros	
Campo	BR
F0	#3
F2	#2
F4	
F6	#1
F8	#4
F10	
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	si	SUB	Mem(34+r2)	Mem(45+r3)			#4
Suma2	no						
Suma3	no						
Mult1	si	MUL	Mem(45+r3)	[F4]			#3
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	si	ld f6,34(r2)	commit	F6	Mem(34+r2)
2	si	ld f2,45(r3)	escritura	F2	Mem(45+r3)
3	si	multd f0,f2,f4	emitido	F0	
4	si	subd f8,f6,f2	emitido	F8	
5	no				
6	no				

Introducción

Superescalares

VLIW

Especulación

Ejemplos

29

5.4.2 Ejemplo Ciclo 6

Estado de los registros	
Campo	BR
F0	#3
F2	#2
F4	
F6	
F8	#4
F10	#5
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	si	DIV		[F6]	#3		#5
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	si	ld f2,45(r3)	commit	F2	Mem(45+r3)
3	si	multd f0,f2,f4	ejecución	F0	
4	si	subd f8,f6,f2	ejecución	F8	
5	si	divd f10,f0,f6	emitido	f10	
6	no				

Introducción

Superescalares

VLIW

Especulación

Ejemplos

30

5.4.2 Ejemplo Ciclo 7

Estado de los registros	
Campo	BR
F0	#3
F2	
F4	
F6	#6
F8	#4
F10	#5
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	si	ADD		[F2]	#4		#6
Suma2	no						
Suma3	no						
Mult1	si	DIV		[F6]	#3		#5
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	no				
3	si	multd f0,f2,f4	ejecución	F0	
4	si	subd f8,f6,f2	ejecución	F8	
5	si	divd f10,f0,f6	emitido	f10	
6	si	addd f6,f8,f2	emitido	f6	

Introducción

Superescalares

VLIW

Especulación

Ejemplos

31

5.4.2 Ejemplo Ciclo 8

Estado de los registros	
Campo	BR
F0	#3
F2	
F4	
F6	#6
F8	#4
F10	#5
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	si	ADD	[#4]	[F2]			#6
Suma2	no						
Suma3	no						
Mult1	si	DIV		[F6]	#3		#5
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	no				
3	si	multd f0,f2,f4	ejecución	F0	
4	si	subd f8,f6,f2	escritura	F8	#1 - #2
5	si	divd f10,f0,f6	emitido	f10	
6	si	addd f6,f8,f2	emitido	f6	

Introducción

Superescalares

VLIW

Especulación

Ejemplos

32

5.4.2 Ejemplo Ciclo 9

Estado de los registros	
Campo	BR
F0	#3
F2	
F4	
F6	#6
F8	#4
F10	#5
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	si	DIV	[#3]	[F6]			#5
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	no				
3	si	multd f0,f2,f4	escritura	F0	#2 x F4
4	si	subd f8,f6,f2	escritura	F8	#1 - #2
5	si	divd f10,f0,f6	emitido	f10	
6	si	addd f6,f8,f2	ejecución	f6	

Introducción

Superescalares

VLIW

Especulación

Ejemplos

33

5.4.2 Ejemplo Ciclo 10

Estado de los registros	
Campo	BR
F0	#3
F2	
F4	
F6	#6
F8	#4
F10	#5
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	no						
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	no				
3	si	multd f0,f2,f4	commit	F0	#2 x F4
4	si	subd f8,f6,f2	escritura	F8	#1 - #2
5	si	divd f10,f0,f6	ejecución	f10	
6	si	addd f6,f8,f2	ejecución	f6	

Introducción

Superescalares

VLIW

Especulación

Ejemplos

34

5.4.2 Ejemplo Ciclo 11

Estado de los registros	
Campo	BR
F0	
F2	
F4	
F6	#6
F8	#4
F10	#5
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	no						
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	no				
3	no				
4	si	subd f8,f6,f2	commit	F8	#1 - #2
5	si	divd f10,f0,f6	ejecución	f10	
6	si	addd f6,f8,f2	escritura	f6	#4 + F2

Introducción

Superescalares

VLIW

Especulación

Ejemplos

35

5.4.2 Ejemplo Ciclo 12

Estado de los registros	
Campo	BR
F0	
F2	
F4	
F6	#6
F8	#5
F10	
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	no						
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	no				
3	no				
4	no				
5	si	divd f10,f0,f6	ejecución	f10	
6	si	addd f6,f8,f2	escritura	f6	#4 + F2

Introducción

Superescalares

VLIW

Especulación

Ejemplos

36

5.4.2 Ejemplo Ciclo 14

Estado de los registros	
Campo	BR
F0	
F2	
F4	
F6	#6
F8	
F10	#5
F12	
.....	
F30	

```

1 ld    f6, 34(r2)
2 ld    f2, 45(r3)
3 multd f0, f2, f4
4 subd  f8, f6, f2
5 divd  f10, f0, f6
6 addd  f6, f8, f2

```

Estaciones de reserva							
Nombre	Ocupado	Op.	V _j	V _k	Q _j	Q _k	Destino
Load1	no						
Load2	no						
Suma1	no						
Suma2	no						
Suma3	no						
Mult1	no						
Mult2	no						

Buffer de reordenación					
ID	Ocup.	Instrucc.	Estado	Destino	Valor
1	no				
2	no				
3	no				
4	no				
5	si	divd f10,f0,f6	escritura	f10	#3 ÷ F6
6	si	addd f6,f8,f2	escritura	f6	#4 + F2

Introducción

Superescalares

VLIW

Especulación

Ejemplos

37

5.4.2 Ejemplo Conceptos principales

■ Ideas clave

- ✓ Los registros/memoria sólo se actualizan cuando se confirma que la instrucción no es especulativa. Mientras se confirma o no, los resultados se guardan en el BR.
- ✓ Una instrucción incorrectamente especulada no altera el estado del procesador (reg/mem) ya que se elimina del BR.
- ✓ Las excepciones se tratan si la instrucción se gradúa → una instrucción incorrectamente especulada no provoca excepción → excep. precisas
- ✓ Cuando una instr. de salto se gradúa se confirma que instrucciones están correctamente especuladas y cuales no
- ✓ El renombre de registros se hace en las ER y en el BR.
- ✓ El BR se implementa como un buffer circular
- ✓ Una estación de reserva se libera en cuanto comienza la ejecución
- ✓ La UF (sumador, multiplicador,...) se libera cuando acaba la ejecución
- ✓ La entrada del BR se liberan cuando la instrucción se gradúa

Introducción

Superescalares

VLIW

Especulación

Ejemplos

38

5.4.2 Ejemplo

Desenrollamiento dinámico de bucles

- Suponemos que el multiplicador consume 4 ciclos

Instrucción	Búsq.	D/Em.	Ejecución	Escritura	Commit
ld f0,0(r1)	1	2	3	4	5
mult f4,f0,f2	2	3	5-8	9	10
sd 0(r1),f4	3	4	10	11	12
subi r1,r1,#8	4	5	6	7	13
beqz r1,ini	5	6	8	9	14
ld f0,0(r1)	6	7	8	9	15
mult f4,f0,f2	7	8	10-14	15	16
sd 0(r1),f4	8	9	16	17	18
subi r1,r1,#8	9	10	11	12	19
beqz r1,ini	10	11	13	14	20

- Después del primer salto las instrucciones en vuelo son especulativas
- Si la predicción de salto falla (ck 9), cancelas todas las instr. siguientes

Introducción Superescalares VLIW **Especulación** Ejemplos 39

5.4.3 Tomasulo con especulación y emisión múltiple de instrucciones (superescalar x2)

ck	BI	DI	Vload	Vsum	Vm/d	Mem	SumPF	MulPF	DivPF	BR	Commit
1	1,2										
2	3,4	1,2	1,2			1					
3	5,6	3,4		4	3	2				1	
4		5,6		4,6	3,5					2	1
5				6	5		4	3			2
6				6	5		4	3			
7				6	5			3		4	
8					5		6			3,4	
9	1	ld		f6, 34(r2)			6		5		3,4
10	2	ld		f2, 45(r3)					5	6	
11	3	multd		f0, f2, f4					5	6	
12	4	subd		f8, f6, f2					5	6	
13	5	divd		f10, f0, f6						5,6	
14	6	addd		f6, f8, f2							5,6

- Emisión, decodificación y commit de hasta 2 inst. por ciclo
- Hemos supuesto: suma (2ck), mult (3ck) y div (4ck)

Introducción Superescalares VLIW **Especulación** Ejemplos 40

5.4.4 Ventajas e inconvenientes

■ Ventajas

- ✓ No hay problemas de dependencias a través de la memoria
- ✓ Se aprovecha de la predicción dinámica de saltos (especulación)
- ✓ Implementa excepciones precisas
- ✓ No es necesario recompilar un código para la nueva arquitectura (planificación dinámica frente a estática)

■ Inconvenientes

- ✓ Elevada complejidad HW
- ✓ Aumentar el ancho del CDB cuando aumenta el factor de superescalaridad
- ✓ Limitado por el grado de IPC que exhibe el programa en ejecución

■ Para el siguiente tema: jerarquía de memoria

- ✓ Importancia del ancho de banda de memoria (más aún en superescalares)
- ✓ Caches no bloqueantes que siguen buscando datos aunque hayan sufridos fallos de cache en instrucciones previas.

Introducción

Superescalares

VLIW

Especulación

Ejemplos

41

5.5 Ejemplos

Procesadores comerciales superescalares y VLIW

✓ Intel

- Pentium (P55C): Superescalar x2 (μOps) sin planificación dinámica
- Pentium Pro a Pentium III: Superescalar x3 (μOps) con planificación dinámica. Supersegmentado
- Pentium 4: Superescalar x3 (μOps) con planificación dinámica. Hipersegmentado
- Itanium e Itanium II: Arquitectura EPIC (VLIW). Arquitectura de 64bits

✓ AMD

- K5 y K6: Superescalar de grado 4 con planificación dinámica.
- K7 (Athlon y Athlon XP): Superescalar x3 (instrucciones nativas)/ x6 (microOps)
- K8 (Athlon 64 y Opteron): Superescalar x3 (instrucciones nativas)/ x6 (microOps). 64 bits

✓ MIPS

- R8000, R10000, R12000 (Todos de 64 bits)

✓ Digital (DEC)

- Alpha 21064, 21164, 21264 (Todos de 64 bits)

✓ Sun

- SuperSPARC y UltraSPARC I, II y III (Todos de 64 bits)

✓ PowerPC

- 601, 604, 620, G3, G4 y G5 (Todos de 64 bits)

Introducción

Superescalares

VLIW

Especulación

Ejemplos

42

5.5 Ejemplos

Intel vs AMD. 5ª Generación

	Pentium			P. MMX	AMD K5	
Referencia	P5, 80501	P54C	P55	P55C	Am5x86	Am5x86
Año	'93	'94	'95	'97	'95	'96
Tecnología (μ)	0.8	0.6	0.35	0.35	0.6	0.35
Frecuencia (MHz)	60/75	100/120	133/200	166/233	75/100	100/166
Frec. Bus (MHz)	50/66	60/66	60/66	66	50/66	66
Nº TRTs (mill)	3.1	3.2	3.3	4.5	4.3	4.3
Nº etapas INT	5			6	5	
Nº etapas FP	8			8		
Nº UFs	3 (2 int, 1pf)			5 (2i,2MMX,1pf)	3(2int,1pf)	
Superescalar	x2 (con fuertes restric.)			x2 (con restric)	x4	
Cache L1 D (on-chip)	8K, 2vías, 32bytes/bloque			16K, 4vías,32B/b	8K	
Cache L1 I (on-chip)	8K, 2vías, 32bytes/bloque			16K, 4vías,32B/b	16K	
Cache L2	Externa, Unificada			Ext. Unific.	Ext. Unific.	

Introducción

Superescalares

VLIW

Especulación

Ejemplos

43

5.5 Ejemplos

Intel 6ª Generación

	P.Pro	Pentium II		Pentium III		
Referencia	P6	Klamath	Deschutes	Katmai	Coppermine	Tualatin
Año	'95	'97	'98	'99	'99	'01
Tecnología (μ)	0.6/0.35	0.35	0.25	0.25	0.18	0.13
Frecuencia (MHz)	150/200	233/300	333/450	450/600	500/1000	1.13/1.4
Frec. Bus (MHz)	66	66	100	100	133	133
Nº TRTs (mill)	5.5	7.5	7.5	9.5	28.1	44
Nº etapas INT	11	11		11		
Nº etapas FP	14	14		14		
Nº UFs	2(2int+2fp)	7(2int+2fp+3MMX)		8(2i+2fp+3MMX+1SSE)		
Superescalar	x3 (μOps)	x3 (μOps)		x3 (μOps)		
Cache L1 D	8K,2v,32B/b	16K,4vías,32B/bloq		16K,4vías,32B/bloq		
Cache L1 I	8K,2v,32B/b	16K,4vías,32B/bloq		16K,4vías,32B/bloq		
Cache L2	4v,32B/b	512K,4vías,32B/bloq		512K,4vías	256K,8vías	512K,8v

Introducción

Superescalares

VLIW

Especulación

Ejemplos

44

5.5 Ejemplos

Intel. 6ª Generación: Económicos y Servidores

	CeleronA	Celeron	CeleronA	P.II Xeon	P.III Xeon	
Referencia	Mendocino	Copperm.	Copperm.	Xeon	Tanner	Cascades
Año	'00	'00	'02	'98	'99	'00
Tecnología (μ)	0.25	0.18	0.13	0.25	0.25	0.18
Frecuencia (MHz)	300/533	800/1100	1.2/1.4G	400/450	500/550	600/933
Frec. Bus (MHz)	66	100	100	100	100	133
Nº TRTs (mill)	19	19	19	7.5	9.5	28.1
Nº etapas INT	11			11	11	
Nº etapas FP	14			14	14	
Nº UFs	7(2int+2fp+3MMX)				8(2i+2fp+3MMX+1SSE)	
Superescalar	x3 (μOps)				x3 (μOps)	
Cache L1 D	16K,4vías,32B/bloq				16K,4vías,32B/bloq	
Cache L1 I	16K,4vías,32B/bloq				16K,4vías,32B/bloq	
Cache L2	128K,4vías,32B/bloq			512K	0.5,1,2M	0.5,1,2M

Introducción

Superescalares

VLIW

Especulación

Ejemplos

45

5.5 Ejemplos

Intel. 6ª Generación. Arquitectura P6

- La misma microarquitectura desde el Pentium Pro al Pentium III
- Nivel ISA IA-32 (conjunto de instrucciones compatible 80x86)
- Sin embargo el núcleo de la arquitectura es RISC
 - ✓ Superescalar con planificación dinámica (Tomasulo con especulación)
- Mecanismo:
 - ✓ Decodificación de las instrucciones IA-32 en μOps RISC de 118bits
 - ✓ Existen dos decodificadores simples y uno complejo
 - Dec. Simples para instrucciones IA-32 simples que se traducen en 1 μOp. Control cableado
 - Dec. Complejo para inst. IA-32 mas complejas.
 - A) Si requiere 4 o menos μOps se generan de forma cableada a partir del CO
 - B) Microprogramado si requiere más de 4 μOps (pueden ser varios cientos de μOps)
 - ✓ Se pueden decodificar hasta dos inst. simples y una compleja tipo A)
 - Es decir, generar 6 μOps al final de la etapa de decodificación
 - ✓ Si la instr. es compleja tipo B) se generas μOps de 6 en 6.

Introducción

Superescalares

VLIW

Especulación

Ejemplos

46

5.5 Ejemplos

Intel. 6ª Generación. Arquitectura P6

■ 11 etapas en el cauce

■ Las 5 primeras:

- ✓ BI y Decodificación

■ Predicción de salto

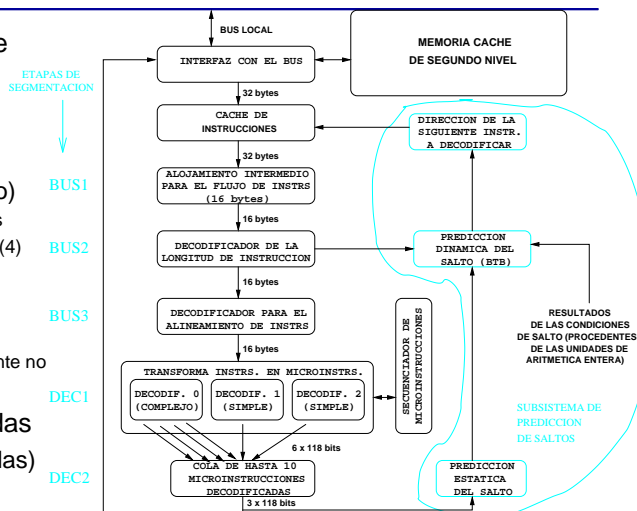
- ✓ Dinámica (90% acierto)
 - BTB, 512 entradas, 4 vías
 - Predictor Adaptativo Pap(4)
 - Si hay fallo en BTB...

- ✓ Estática

- 6 ciclos de penalización
- Salto atrás tomado, delante no
- Si falla... mas de 12ck

■ Las µOps decodificadas

- ✓ Se encolan (10 entradas)
- ✓ Se emiten de 3 en 3



Introducción

Superescalares

VLIW

Especulación

Ejemplos

47

5.5 Ejemplos

Intel. 6ª Generación. Arquitectura P6

■ 6 etapas restantes

■ Tomasulo con especul.

- ✓ Emisión=Aloj+Reor+Prep
- ✓ Ejecución=Ejec
- ✓ Escritura=Esc1
- ✓ Commit=Esc2

■ Buffer de reord. (BRC)

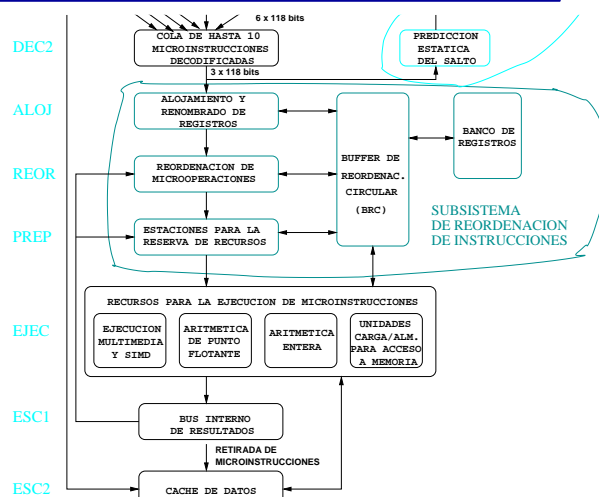
- ✓ 40 entradas
- ✓ 254bits por entrada

■ 5 bancos de est. de res.

■ 7 Unidades funcionales

■ Retira 3 µOps por ck

■ Superescalar x3 (µOps)



Introducción

Superescalares

VLIW

Especulación

Ejemplos

48

5.5 Ejemplos

Intel. 6ª Generación. Arquitectura P6

- Variantes comerciales
 - ✓ Low End: Celeron. Caches L2 más pequeñas (generalmente on-die)
 - El primer Celeron('98, Covington) no tenía cache L2 y fue un fracaso comercial (y técnico)
 - ✓ Mid Range: Pentium X. Caches L2 intermedias (generalmente on-chip)
 - ✓ High End: Xeon. Caches L2 más grandes (generalmente on-die) y SMP
 - El soporte SMP (Symmetric MultiProcessing): placas dual o quad-pentium
 - Caches internas (on-chip) a menos frecuencia que el procesador
 - Caches integradas (on-die) a igual frecuencia que el procesador
- Evolución de las instrucciones multimedia
 - ✓ MMX aparece con el P.MMX y se incorpora en el P.II
 - 57 instrucciones para operaciones SIMD con enteros empaquetados en 64bits
 - ✓ SSE (Streaming SIMD Extensions) o MMX-2. Incluido en el P.III
 - 70 instrucciones adicionales que permiten operaciones SIMD con dos reales s.precis. en 64bits
 - ✓ SSE-2. Aparece con el P.4
 - 144 instrucciones más para op. SIMD empaquetadas en 128 bits (enteros, simple y doble prec.)

Introducción

Superescalares

VLIW

Especulación

Ejemplos

49

5.5 Ejemplos

Intel. 6ª Generación: Pentium M

- Basado en el Pentium III (Arquitectura P6)
- Se comercializa en el 2003
 - ✓ Códigos de referencia: Banias (0.13μ) y Dothan (0.09μ)
- Bajo consumo de potencia
 - ✓ El Pentium 4 puede consumir hasta 100W
 - ✓ El Pentium M consume como máximo 31W
 - Voltaje de alimentación: 0.84V a 600MHz hasta 1.48V a 1.7GHz
- Rango de frecuencias: 600 a 1700MHz
- Transistores:
 - ✓ Banias: 77Millones (incluye L2 de 1M)
 - El Celeron M desactiva 0,5M de L2 → Es un proc. de 0.5M L2, pero tiene 77MTrts
 - ✓ Dothan:144Millones (incluye L2 de 2M)
- Cache L1 de 32K+32K, 8vías, 64Bytes/bloque

Introducción

Superescalares

VLIW

Especulación

Ejemplos

50

5.5 Ejemplos

AMD. 6ª Generación

	K6	K6-2	K6-III
Referencia	K6	Chomper	Sharptooth
Año	'97	'98	'99
Tecnología (μ)	0.35/0.25	0.25	0.25
Frecuencia (MHz)	166/333	266/500	400/600
Frec. Bus (MHz)	66	66/100	100
Nº TRTs (mill)	8.8	9.3	21.3
Nº etapas INT	6		
Nº etapas FP			
Nº UFs	8 (2int + 3MMX + 1fp + 2 3DNow!)		
Superscalar	x4 (μOps)		
Cache L1 D	32K, 2vías, 32Bytes/bloque		
Cache L1 I	32K, 2vías, 32Bytes/bloque		
Cache L2	Externa		256K, 4vías

Introducción

Superscalares

VLIW

Especulación

Ejemplos

51

5.5 Ejemplos

AMD 7ª Generación

	K7/Athlon		Athlon XP			
Referencia	Argon	Thunderbird	Palomino	Thoroughbred A y B		Barton
Año	'99	'00	'01	'02	'02	'03
Tecnología (μ)	0.25	0.25	0.18	0.13		0.13
Frecuencia (GHz)	0.5/1	0.65/1.4	1.33/1.7	1.33/1.8	2/2.4	2.2/2.833
Frec. Bus (MHz)	200	200/266	266	266	266/333	333
Nº TRTs (mill)	22	37	37.5	37.2	37.6	53.9
Nº etapas INT	10		10			
Nº etapas FP	15		15			
Nº UFs	9(3i,1f,2MMX,2-3D,1En)		9(3int,1fp,2MMX,2-3DNow!,1Enhanced3DNow)			
Superscalar	x3 (nativas)/x6(μOps)		x3 (nativas)/x6(μOps)			
Cache L1 D	64K,2vías,32B/bloq		64K,2vías,32B/bloq			
Cache L1 I	64K,2vías,32B/bloq		64K,2vías,32B/bloq			
Cache L2	512K,2vías	256K,16vías	256K,16vías		512K	

Introducción

Superscalares

VLIW

Especulación

Ejemplos

52

5.5 Ejemplos

AMD 7ª Generación

- Cuidado con los nombres comerciales del Athlon XP

Nombre Comercial	Frecuencia Real (MHz)	Frecuencia de bus (MHz)	Nombre Comercial	Frecuencia Real (MHz)	Frecuencia de bus (MHz)
XP 1500+	1333	2x133	XP 2400+	2000	2x133
XP 1600+	1400	2x133	XP 2600+	2133	2x133
XP 1700+	1466	2x133	XP 2700+	2166	2x166
XP 1800+	1533	2x133	XP 2800+	2250	2x166
XP 1900+	1600	2x133	XP 3000+	2416	2x166
XP 2000+	1666	2x133	XP 3200+	2500	2x166
XP 2100+	1733	2x133	XP 3400+	2666	2x166
XP 2200+	1800	2x133	XP 3600+	2833	2x166

- Versiones económicas:

- ✓ Duron (Spitfire): '00; 0.18μ; 0.6/1GHz; 200MHz (bus); 25MTTrts; 64K L2
 - Reduce el tamaño de la cache L2, pero sigue siendo de 16 vías.
- ✓ Duron (Morgan): '01; 0.18μ; 0.95/1.3GHz; 200MHz (bus); 25.2MTTrts; 64K L2
 - Incluye el controlador de memoria, de gráficos y de sonido

Introducción

Superescalares

VLIW

Especulación

Ejemplos

53

5.5 Ejemplos

AMD 7ª Generación: Arquitectura Athlon

- Microarquitectura K7 igual para el Athlon y el Athlon XP
- Conjunto de instrucciones ISA IA-32 con arquitectura RISC
 - ✓ Superescalar con planificación dinámica (Tomasulo con especulación)
- Conversión CISC a RISC
 - ✓ Se pueden decodificar hasta 3 instrucciones nativas IA-32 en paralelo
 - ✓ Cada una puede generar hasta 3 μOps RISC→ Total hasta 9 μOps por ck
 - ✓ Dos tipos de instrucciones
 - Simples: se decodifican por control cableado en el "DirectPath"
 - Complejas: se decodifican por control microprogramado en el "VectroPath"
- Superescalaridad y segmentación
 - ✓ Tres cauces enteros de 10 etapas
 - ✓ Tres cauces flotantes de 15 etapas
 - ✓ Superescalaridad de factor 6 (a nivel de μOps)

Introducción

Superescalares

VLIW

Especulación

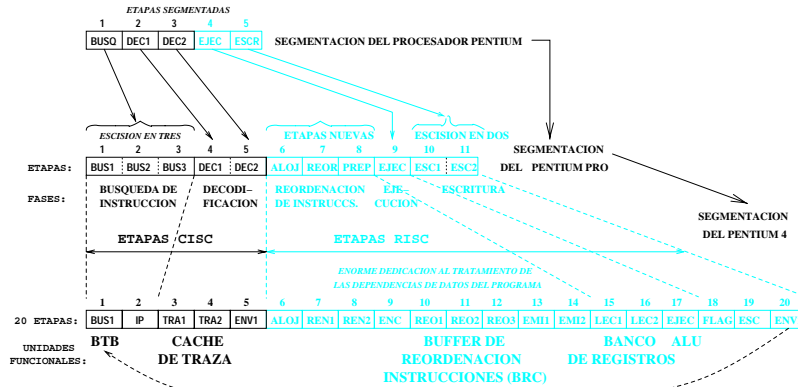
Ejemplos

54

5.5 Ejemplos

Intel 7ª Generación: Arquitectura Pentium 4

- Hipersegmentado (20 etapas). Superescalar x3 (μOps)



- Etapas 5 y 20 consumen 1ck en transporte de información

Introducción

Superescalares

VLIW

Especulación

Ejemplos

57

5.5 Ejemplos

Intel 7ª Generación: Arquitectura Pentium 4

- Conjunto de instrucciones ISA IA-32
- Sólo decodifica 1 instr. IA-32 por ck, que genera de media 3 μOps
- Las μOps se almacenan en una cache de traza (L1I de 12K μOps)

1	cmp
2	br → T1
...	(código no usado)
T1:	3 sub
4	br → T2
...	(código no usado)
T2:	5 mov
6	sub
7	br → T3
...	(código no usado)
T3:	8 add
9	sub
10	mul
11	cmp
12	br → T4

Cache de traza

1	cmp	2	br T1	3	T1: sub
4	br T2	5	mov	6	sub
7	br T3	8	T3: add	9	sub
10	mul	11	cmp	12	br T4

Introducción

Superescalares

VLIW

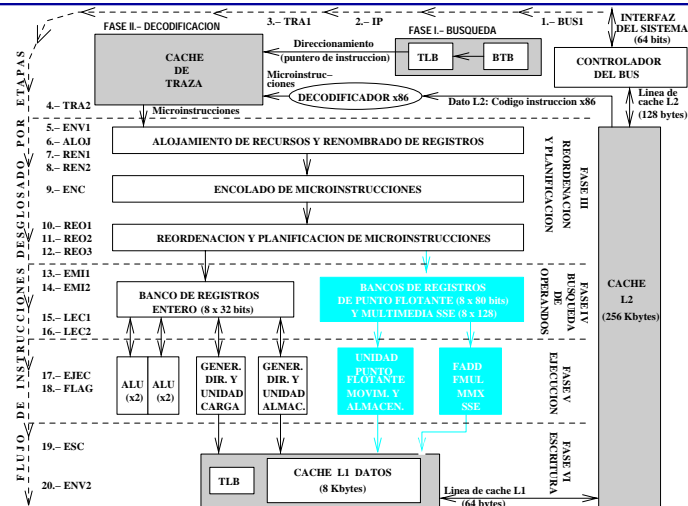
Especulación

Ejemplos

58

5.5 Ejemplos

Intel 7ª Generación: Arquitectura Pentium 4



Introducción

Superescalares

VLIW

Especulación

Ejemplos

59

5.5 Ejemplos

Intel 7ª Generación: Arquitectura Pentium 4

- Buffer de reordenación de 126 entradas
 - ✓ 3 veces más que la arquitectura P6 → mantiene 126 instr. "en vuelo"
- Importancia del predictor de salto (Condición conocida en etapa 18)
 - ✓ BTB de 4096 entradas (8 veces más que el P6). Predictor híbrido
 - ✓ Coste del fallo de predicción de 20 a más de 31ck (depende de L1 traza)
 - ✓ Acierto alrededor del 25%.
- Puntos fuertes
 - ✓ Operaciones aritméticas enteras en 0.5ck
 - ✓ Bus local de 64bits (8bytes) a 400 o 533MHz → 3.2GBs o 4.2GBs
 - ✓ SMT (Symmetric MultiThreading). Emula dos procesadores lógicos
- Puntos débiles
 - ✓ Conjunto de instrucciones IA-32
 - ✓ Arq. desbalanceada: poca superescalaridad y demasiada segmentación
 - ✓ Caches (L1 y L2) pequeñas

Introducción

Superescalares

VLIW

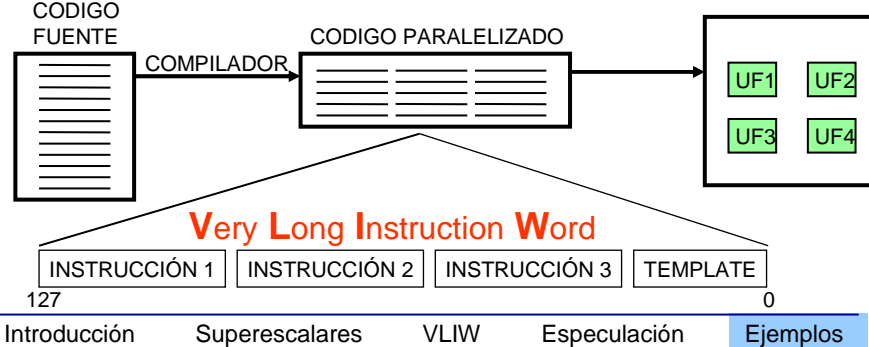
Especulación

Ejemplos

60

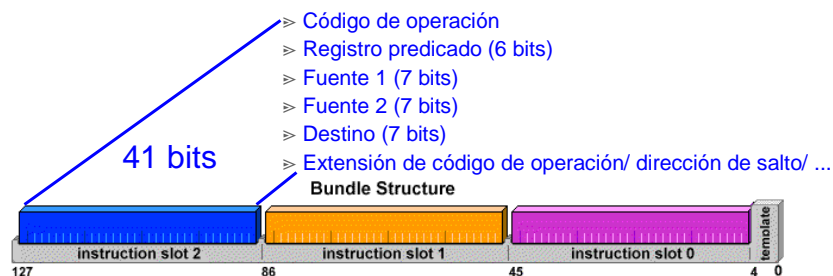
5.5 Ejemplos Intel-HP Itanium

- Problemas de mantener el conjunto IA-32
 - ✓ Hay que traducir (en decodificación) a μ Ops (hasta 3ck de pérdida)
 - ✓ El compilador optimiza el código IA-32 pero no a nivel de μ Ops
- Solución: cambio de nivel ISA al conjunto de instrucciones IA-64
 - ✓ Filosofía VLIW mejorada: EPIC (**Explicitly Parallel Instruction Computing**)



5.5 Ejemplos Intel-HP Itanium

Instrucciones Largas (bundles= 128 bit)



"Bundle" : Es un conjunto de 3 instrucciones y un template.

El campo de template (5 bits)

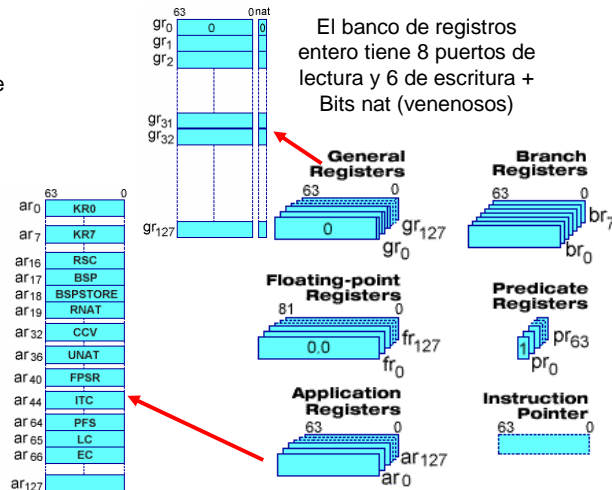
- indica en que unidad funcional se ejecuta cada operación
- indica el grado de paralelismo entre las instrucciones del bundle
- la dependencia entre esta instrucción larga (bundle) y las siguientes.

5.5 Ejemplos Intel-HP Itanium

Registros IA-64

- 128 Registros generales
- 128 Registros coma flotante
- 64 Registros Predicados
- 8 Registros de Saltos
- 128 Registros de aplicación
- Contador de Instr. (IP)

Mediante instrucciones condicionales se eliminan el 50% de los saltos.



Introducción

Superescalares

VLIW

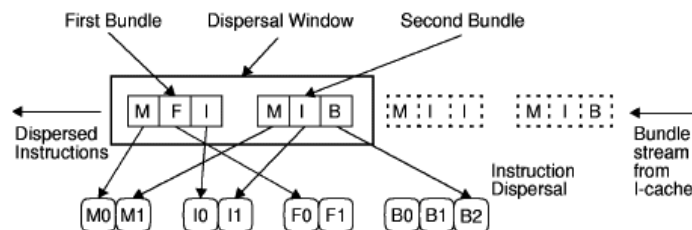
Especulación

Ejemplos

63

5.5 Ejemplos Intel-HP Itanium

- Arquitectura Hardware: procesador segmentado en 8 etapas.
 - ✓ Puertos de emisión (entre paréntesis número de unidades funcionales):
 - 2 inst. Memoria (Itanium). 4 inst. Memoria (Itanium II) (Cache L1D de cuatro puertos)
 - 2 inst. Enteras (6 unidades enteras)
 - 2 inst. PF. (2 unidades punto flotante y una SIMD –multimedia–)
 - 3 inst. Salto. (3 unidades de salto)
 - ✓ Tamaño de ventana: 2 bundles (equivalente a un superescalar x6)
- Distribución de instrucciones desde la ventana de dispersión (para el Itanium)



Introducción

Superescalares

VLIW

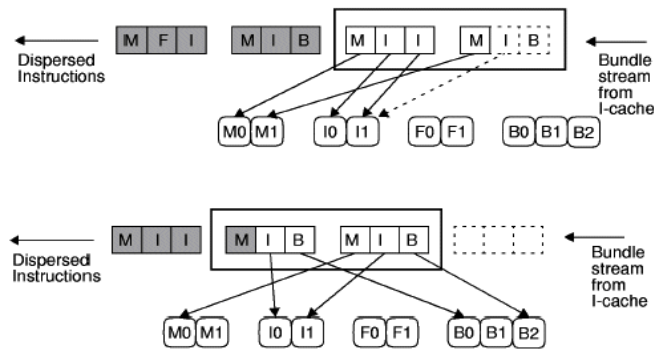
Especulación

Ejemplos

64

5.5 Ejemplos Intel-HP Itanium

- Ejemplos de ejecución con y sin riesgos estructurales:



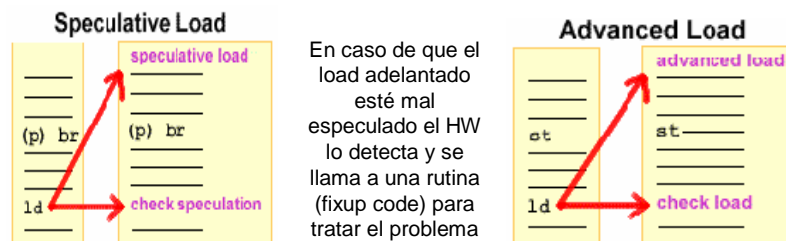
Source: "Itanium™ Processor Microarchitecture Reference", Intel, August 2000

Introducción Superescalares VLIW Especulación Ejemplos 65

5.5 Ejemplos Intel-HP Itanium

- Para ocultar la latencia de memoria se implementan load especulativos

- ✓ Adelantar un load fuera del bloque básico (riesgo de control)
 - Se controlan las excepciones precisas mediante los bits venenosos
- ✓ Adelantar un load a un store (riesgo de datos)
 - El load "adelantado" apunta en una tabla (ALAT) la dirección de la que está cargando
 - Si luego el store escribe en esa misma posición se da cuenta ya que está apuntado en ALAT
 - ALAT = Advanced Load Address Table. Tabla totalmente asociativa.

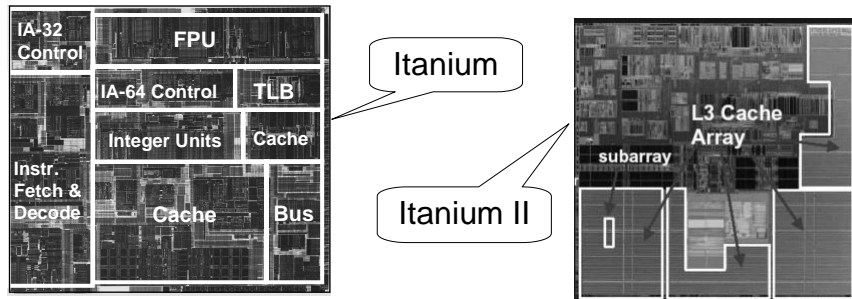


Introducción Superescalares VLIW Especulación Ejemplos 66

5.5 Ejemplos

Intel-HP Itanium

- Predicción de salto:
 - ✓ Predictor en dos niveles con BHT (2,2) de 512 entradas
 - Penalización de 6ck por fallo en predicción
 - ✓ BTB (aquí BTAC: Branch Target Address Cache) de 64 entradas
 - ✓ Pila de direcciones de retorno de subrutina
- Puede ejecutar código IA-32 pero más lento (traducción HW a bundles)



Introducción Superescalares VLIW Especulación Ejemplos 67

5.5 Ejemplos

Intel-HP Itanium

■ Modelos

Nombre	Código	F(GHz)	Tecn.(μ)	MaxPot	Volt	L1(KB)	L2(KB)	L3	F.Bus
Itanium	Merced	0.8	180			16+16,2ck	96,12ck	ext.	266MHz
Itanium II	McKinley	1	180	130W	1.5V	16+16,1ck	256,5ck	3M,11c	400MHz
Itanium II	Madison	1.5	130	130W	1.3V	16+16,1ck	256,5ck	6M	400MHz
ItaniumII	Deerfield(LV)	1	130	62W	1.1V	16+16,1ck	256,5ck	1,5M	400MHz

- ✓ Itanium: Bus de 64bits(2.1GBs); ItaniumII: Bus de 128bits (6.4GBs)
- ✓ Itanium II con L3 de 3MB tiene 221 millones de transistores. 6MB→410MTrts
- Detalles de la cache del Itanium II
 - ✓ L1I: 64Bytes/bloque, 4vías, reemplazo LRU
 - ✓ L1D: 64Bytes/bloque, 4vías, reemplazo NRU. Write Through. 4puertos (2I,2e)
 - ✓ L2: 128Bytes/bloque, 8vías, reemplazo NRU. Write Back
 - ✓ L3: 128Bytes/bloque, 12vías, reemplazo NRU. Escritura Write Back

Introducción Superescalares VLIW Especulación Ejemplos 68

5.5 Ejemplos

Intel roadmap (no oficial)

- Pentium 5: Prescott y Nocona (3.4GHz inicialmente)
 - ✓ De 32 bits. Con bus de direcciones de 40 bits.
 - ✓ Número de threads: 2 (dos procesadores lógicos)
 - ✓ Cache L1 Datos 16K, L1 Inst 16 KmicroOps (de traza). Cache L2: 1MByte
 - ✓ Banco de registros: 256 int, 128 fp.
- Pentium 6: Potomac, Jayhawk y Yamhill (4GHz inicialmente)
 - ✓ De 32 bits. Con bus de direcciones de 48 bits.
 - ✓ Número de threads: 4 (cuatro procesadores lógicos)
 - ✓ Cache L1 Datos 32K, L1 Inst 16 KmicroOps (de traza). Cache L2: 1MByte
 - ✓ Banco de registros: 256 int, 128 fp.
- Pentium 7: Tejas
- Pentium 8: Nehalem
 - ✓ 0.65micras, mas de 100 MTTrs y bus de memoria de 1.2GHz.

Introducción

Superescalares

VLIW

Especulación

Ejemplos

69

5.5 Ejemplos

AMD 8ª Generación: K8

- K8: Athlon64 (MidRange), Athlon64 FX(Gaming), Opteron (High End)

	Athlon 64			Opteron UP, DP y MP		
Referencia	Clawhammer	Paris	San Diego	Sledgehammer		
Año	'03	'03	'04	'03	'03	'03
Tecnología (μ)	0.13	0.13	0.09	0.13	0.13	0.13
SMP	1			1	2	4/8
Cache L2 MB	1	0.256	1	1	1	1

- Opteron a 90nm con códigos Venus (UP), Troy (DP) y Athens (MP)
- Frecuencia: de 1.4 a 2.2GHz en 130nm y más de 3GHz en 90nm
- Transistores: París (67 MTTrs), Clawhammer (106MTTrs)
- Procesador de 64 bits
 - ✓ Nuevo conjunto de instrucciones x86-64 o AA-64
 - ✓ Registros, UF enteras y direcciones de 64 bits

Introducción

Superescalares

VLIW

Especulación

Ejemplos

70

5.5 Ejemplos

AMD 8ª Generación: K8

■ Evolución de los bancos de registros

EVOLUCION ARQUITECTURAL SEGUN EL NUMERO DE BITS

	IA-16: Arq. 16 bits	IA-32: Arq. Intel 32 bits	x86-64: Arquitectura AMD de 64 bits																																								
Registros de propósito general (datos enteros)	<div><div>15870</div><table><tr><td>AX</td><td>AH</td><td>AL</td></tr><tr><td>BX</td><td>BH</td><td>BL</td></tr><tr><td>CX</td><td>CH</td><td>CL</td></tr><tr><td>DX</td><td>DH</td><td>DL</td></tr></table></div>	AX	AH	AL	BX	BH	BL	CX	CH	CL	DX	DH	DL	<div><div>311570</div><table><tr><td>EAX</td><td>AH</td><td>AL</td></tr><tr><td>EBX</td><td>BH</td><td>BL</td></tr><tr><td>ECX</td><td>CH</td><td>CL</td></tr><tr><td>EDX</td><td>DH</td><td>DL</td></tr></table></div>	EAX	AH	AL	EBX	BH	BL	ECX	CH	CL	EDX	DH	DL	<div><div>63311570</div><table><tr><td>RAX</td><td>EAX</td><td>AH</td><td>AL</td></tr><tr><td>RBX</td><td>EBX</td><td>BH</td><td>BL</td></tr><tr><td>RCX</td><td>ECX</td><td>CH</td><td>CL</td></tr><tr><td>RDX</td><td>EDX</td><td>DH</td><td>DL</td></tr></table></div>	RAX	EAX	AH	AL	RBX	EBX	BH	BL	RCX	ECX	CH	CL	RDX	EDX	DH	DL
AX	AH	AL																																									
BX	BH	BL																																									
CX	CH	CL																																									
DX	DH	DL																																									
EAX	AH	AL																																									
EBX	BH	BL																																									
ECX	CH	CL																																									
EDX	DH	DL																																									
RAX	EAX	AH	AL																																								
RBX	EBX	BH	BL																																								
RCX	ECX	CH	CL																																								
RDX	EDX	DH	DL																																								
Regs. de dirección (punteros)	<div><div>15870</div><table><tr><td>SP</td><td></td></tr><tr><td>BP</td><td></td></tr><tr><td>SI</td><td></td></tr><tr><td>DI</td><td></td></tr></table></div>	SP		BP		SI		DI		<div><div>311570</div><table><tr><td>ESP</td><td></td></tr><tr><td>EBP</td><td></td></tr><tr><td>ESI</td><td></td></tr><tr><td>EDI</td><td></td></tr></table></div>	ESP		EBP		ESI		EDI		<div><div>63311570</div><table><tr><td>RSP</td><td></td></tr><tr><td>RBP</td><td></td></tr><tr><td>RSI</td><td></td></tr><tr><td>RDI</td><td></td></tr></table></div>	RSP		RBP		RSI		RDI																	
SP																																											
BP																																											
SI																																											
DI																																											
ESP																																											
EBP																																											
ESI																																											
EDI																																											
RSP																																											
RBP																																											
RSI																																											
RDI																																											
Regs. control (flags estado)	<div><div>150</div><table><tr><td>FLAGS</td></tr><tr><td>IP(PC)</td></tr></table></div>	FLAGS	IP(PC)	<div><div>310</div><table><tr><td>EFLAGS</td></tr><tr><td>EIP (PC)</td></tr></table></div>	EFLAGS	EIP (PC)	<div><div>63310</div><table><tr><td>EFLAGS</td></tr><tr><td>RIP (PC)</td></tr></table></div>	EFLAGS	RIP (PC)																																		
FLAGS																																											
IP(PC)																																											
EFLAGS																																											
EIP (PC)																																											
EFLAGS																																											
RIP (PC)																																											
Ejemplos comerciales	Intel 8086/8088 Intel 80286	Intel 80386 – 80486, Intel Pentium – Pentium 4, (K5, K6 y K7 de AMD análogos)	AMD K8 (Itanium de Intel análogo)																																								

Introducción

Superescalares

VLIW

Especulación

Ejemplos

71

5.5 Ejemplos

AMD 8ª Generación: K8

■ Mantiene la compatibilidad con IA-32. Varios modos de trabajo:

- ✓ Modo extenso de 64bits
 - Direcciones virtuales de 64 bits. Puntero de instrucciones de 64 bits
 - Registros extendidos de 64 bits y registros MMX de 128 bits (los 8 del K7 más 8 adicionales)
- ✓ Modo extenso de compatibilidad con 32 bits
 - Sin necesidad de recompilar puede ejecutar códigos IA-32 bajo un S.O. de 64 bits en modo ext.
 - Renunciando a las nuevas prestaciones: espacio virtual de 4GB y operandos de 16 o 32 bits
- ✓ Modo tradicional de 32 bits: plena compatibilidad x86 a 32 bits

■ Microarquitectura similar al K7

- ✓ Superescalar x6 (μOps). Tomasulo con las mismas entradas en el B. Reord.
- ✓ Segmentación 2 etapas más profunda: 12 etapas int y 17 etapas fp

■ Caches

- ✓ L1 64K+64K, 2vías, 64bytes/bloque
- ✓ L2 unificada, 256K o 1M. 16 vías, 64 bytes/bloque

Introducción

Superescalares

VLIW

Especulación

Ejemplos

72

Bibliografía

- J.L. HENNESSY, D. PATTERSON. Computer Architecture: a quantitative approach. Ed. Morgan Kaufman, 3ed. 2003
 - ✓ Superescalares y especulación: Capítulo 3.
 - ✓ Desenrollamiento y VLIW: Capítulo 4.
- MANUEL UJALDÓN MARTÍNEZ. Arquitectura del PC. Volumen I: Microprocesadores. Ed. Ciencia III S.L. 2003
 - ✓ Ejemplos de procesadores de Intel y AMD: Capítulos 4 al 7.
- Direcciones WEB para los ejemplos de procesadores
 - ✓ www.intel.com, www.amd.com
 - ✓ www.sandpile.org, www.arstechnica.com,
 - ✓ www.chip-architect.com, www.tomshardware.com

Introducción

Superescalares

VLIW

Especulación

Ejemplos

75