

Zentralübung Rechnerstrukturen: Pipelining und Superskalartechniken

4. Aufgabenblatt – Musterlösung

1 Pipelining

1. Pipeline und Zykluszeit

- a) • Ohne Pipelining: Zykluszeit = Summe aller Stufen
 $\text{Zykluszeit} = 250 \text{ ps} + 100 \text{ ps} + 130 \text{ ps} + 220 \text{ ps} + 50 \text{ ps} = 750 \text{ ps}$
• Mit Pipelining: Zykluszeit = Längste Stufe + Latenz des Pipelineregisters
 $\text{Zykluszeit} = 250 \text{ ps (IF-Stufe)} + 20 \text{ ps} = 270 \text{ ps}$
- b) $\text{SpeedUp} = \frac{\text{average exec time w/o pipeline}}{\text{average exec time w pipeline}} = \frac{\text{CPI} * \text{CycleTime w/o p}}{\text{CPI} * \text{CycleTime w p}} = \frac{1,0 * 750 \text{ ps}}{1,2 * 270 \text{ ps}} \approx 2,31$
- c)

IF	ID + EX	MA + WB
----	---------	---------
- d) Die Zykluszeit wird von der Ausführungszeit längsten Stufe bestimmt
⇒ Aufteilung der längsten Stufe, hier der IF-Stufe
Die Aufteilung einer anderen Stufe würde zu keiner Reduktion der Zykluszeit führen.

2. Pipeline und Kontextwechsel

- a) • CPI-Wert von 1 bedeutet ohne Misses können in 200 Zyklen 200 Instruktionen abgearbeitet werden
⇒ Betrachtung der Berechnung von 200 Instruktionen
– 1 L1 Cache-Miss nach 100 Berechnungszyklen
– 1 L1 und L2 Cache-Miss nach 200 Berechnungszyklen

Insgesamt:

Anzahl an Zyklen: $100 + 10 + 100 + 50 = 260$

$$\text{CPI} = \frac{260}{200} = 1,3$$

- b)
- Pro Thread erneut 200 Instruktionen
 - Bei Cache-Miss Kontextwechsel zu anderem Thread
 - Alternierend 100 Berechnungszyklen pro Thread
 - Letzter Cache-Miss des 2. Threads nicht mehr mit Berechnung (lediglich 1 Instruktion) überlagert

Thread 1:

Anzahl Zyklen: $2 \cdot 100 + 2 \cdot 100 + 1 = 401$

$$CPI_1 = \frac{401}{200} \approx 2$$

Thread 2:

Anzahl Zyklen: $2 \cdot 100 + 2 \cdot 100 + 50 = 450$

$$CPI_2 = \frac{450}{200} = 2,25$$

Also jedes Programm einzeln betrachtet einen schlechteren CPI-Wert.

Aber 400 Instruktionen in 450 Zyklen, statt 520 bei sequentieller Ausführung.

2 Multi-Threading

Befehl	Dispatch (Issue Q Anzahl Befehle)	Issue	Register Fetch	Start Exec	Ende Exec	CDB	Retire (TX)	Retire (TY)
X1	1 (1)	2	3	4	5	6	7	
X2	1 (2)	4	5	6	7	8	9	
X3	2 (3)	4	5	6	9	10	11	
X4	2 (4)	8	9	10	11	12	13	
X5	9 (3)	10	11	12*	12	13		
X6	9 (4)	11	12	13	13	14		
Y1	15 (1)	16	17	18	19	20		21
Y2	15 (2)	18	19	20	22	23		24
Y3	16 (3)	21	22	23	24	25		26
Y4	16 (4)	23	24	25	26	27		28
Y5	24 (3)	25	26	27	29	30		31
Y6	24 (4)	28	29	30	30	31		32
X5	32 (2)	33	34	35	35	36	37	
X6	32 (3)	34	35	36	36	37	38	

- * Im 12. Zyklus wird Cache-Miss erkannt
- Im 13. Zyklus wird eine HW-Exception ausgelöst
- Im 14. Zyklus die Pipeline geleert und somit im 15. Zyklus mit dem Dispatch des Threads Y begonnen

3 Algorithmus von Tomasulo I

#	Instruktion	Issue	Executes	Writes Result
1	mul r2, r1, r1	1	2	6
2	div r4, r4, r2	2	7	15
3	add r1, r4, r4	3	16	18
4	add r2, r4, r3	4	19	21
5	div r1, r2, r3	7	22	30
6	sub r4, r4, r2	8	22	24
7	add r3, r1, r2	9	31	33
8	mul r1, r2, r3	16	34	38
9	add r3, r3, r3	19	34	36
10	sub r4, r4, r1	22	39	41

4 Algorithmus von Tomasulo II

Hinweis: Schauen Sie sich auch die Übungsfolien mit den ausführlichen Erläuterungen an!

Befehl	Takt																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
LD.D F0,0(R1)	IF	ID	IS	M	M	M	WB																
ADD.D F4,F0,F2	IF	ID	IS				EX	EX	WB		M												
S.D 0(R1),F4		IF	ID	IS					M														
ADD R1,R1,#8		IF	ID	IS	EX	WB																	
SUB R3,R1,R2			IF	ID		IS	EX	WB															
BLTZ R3,LOOP			IF	ID				IS	EX	WB													
LD.D F0,0(R1)				IF	ID		IS	M	M		WB												
ADD.D F4,F0,F2				IF	ID			IS			EX	EX	WB										
S.D 0(R1),F4					IF	ID					IS		M	M									
ADD R1,R1,#8					IF	ID				IS	EX	WB											
SUB R3,R1,R2						IF	ID				IS		EX	WB									
BLTZ R3,LOOP						IF	ID						IS		EX	WB							
LD.D F0,0(R1)							IF	ID			IS		M	M	M	WB							
ADD.D F4,F0,F2							IF	ID					IS		IS	EX	WB	EX	M	M			
S.D 0(R1),F4								IF	ID							IS		M	M				
ADD R1,R1,#8								IF	ID								EX	WB	IS	EX	WB		
SUB R3,R1,R2									ID														
BLTZ R3,LOOP							IF	ID													IS	EX	WB

5 VLIW-Prozessoren

Hinweis: Schauen Sie sich auch die Übungsfolien mit den ausführlichen Erläuterungen an!

a)

Slot 1	Slot 2	Slot 3
add r1, r2, r3	sub r5, r3, r5	ld r11, [r12]
ld r3, [r1]	ld r9, [r7]	add r11, r11, r12
mul r3, r3, r3	mul r11, r11, r9	
st [r5], r3	st [r12], r11	

b)

Slot 1 (ALU)	Slot 2 (ALU)	Slot 3 (L/S)
add r1, r2, r3	sub r5, r3, r5	ld r11, [r12]
add r11, r11, r12		ld r3, [r1]
mul r3, r3, r3		ld r9, [r7]
mul r11, r11, r9		st [r5], r3
		st [r12], r11