

24

Aufgabe 1

Bevor die Modell-CPU den ersten Takt-Zyklus der Fetch-Phase startet, halten die internen Register die in Zeile 1 der Tabelle 2 angegebenen Werte. Die Informationen über die Inhalte benötigter Speicher-Zellen sind in Tabelle 1 wiedergegeben (sämtliche Zahlenwerte sind zur Basis 16 angegeben). Eine Interruptanmeldung liegt vor, d. h. FGO oder FGI ist gesetzt!

Füllen Sie Tabelle 2 aus, d. h. bestimmen Sie die Inhalte der gegebenen Register jeweils zum Zeitpunkt nach der aktiven Flanke eines Takt-Pulses der Fetch-, Adressberechnungs-, Execute- und Interrupt-Phase (eine Interruptanmeldung liegt vor, IEN und FGI gesetzt)!

Fügen Sie ferner in Tabelle 1 die jeweilige Adresse und den Inhalt der zusätzlich verwendeten Speicher-Zellen hinzu!

Adresse	Inhalt
1000	220499
149A	14AB52
FE2F	141001

Tabelle 1: Inhalte einiger Speicher-Zellen

PC	MAR	MBR	C/AC	IX	SP	OPR	OPX	AR	F	R	Takt / Phase vorher
1000	4CEA	29AC9F	0/14AB52	119432	FE30	7	0	1	0	0	t ₀ Fetch
1001	1000	220499									t ₁
						2	0499	2			t ₂
										1	t ₃
		22149A									t ₀ Address Calc.
											t ₁
											t ₂
									1	0	t ₃
	149A										t ₀ Execute
		14AB52									t ₁
											t ₂
										1	t ₃
0	FE2F	141001			FE2F						t ₀ Interrupt
											t ₁
											t ₂
									0	0	t ₃

Tabelle 2: CPU-Register

7 (16 Punkte)

Aufgabe 2

1. Welche der folgenden beispielhaften Mikrobefehle sind auf der Mic-1 Architektur durchführbar, welche sind **nicht** durchführbar? Im Falle der Nichtdurchführbarkeit ist dies bitte zu begründen!

a. $H = MBRU \ll 1$

nicht ausführbar, Umkehrschreib nur in 8-9,1 (2)

b. $MBR = H - PC$

nicht ausführbar
MBR kann in Mic-1 nicht beschrieben werden (2)

c. $LV = SP + 2; rd$

nicht ausführbar
LV muss kleiner als SP sein
+2 kein ALU-op (4)

d. $H = MBRU \text{ OR } H$

Ausführbar (2)

e. $MDR = TOS = MDR \text{ AND } H; wr$

Ausführbar (2)

2. Wie lautet der binäre Mikrocode für den folgenden beispielhaften Mikrobefehl?

$SP = MAR = SP - 1; rd$

(Folgeadresse = 0x04C)

Pop 0x57 (2)

Aufgabe 3

(8 Punkte)

1. Was ist der Unterschied zwischen Übertrag (Carry-Out) und Überlauf (Overflow)?

~~Überlauf ist, wenn der darzustellende Bereich überschritten wird.~~

Überlauf ist, wenn sich das MSB ändert und somit bei Vorzeichenbehafteter Rechnung außerhalb des Bereichs ist. Übertrag ist, wenn die miteinander addierten Zahlen ein Ergebnis haben, was außerhalb des Bit-Bereiches liegt.

2. Was ist der Unterschied zwischen logischem und arithmetischem Schieben?

Logisches Schieben ist nicht vorzeichenbehaftet (es wird mit 0 aufgefüllt). Arithmetisches Schieben ist vorzeichenbehaftet, ~~erweitert~~ (es wird mit dem MSB aufgefüllt).

Aufgabe 4

(12) (12 Punkte)

Auf der Integer-Instruktionen-Teilmenge der Java Virtuellen Maschine (IJVM) soll folgende JAVA-Anweisung ausgeführt werden:

$$e = (i - n) + (2 - k) - 9$$

Die Offsets der lokalen Variablen im LV-Bereich seien $i = 3$, $e = 5$, $n = 6$ und $k = 9$!

Wie lautet das symbolische Java Assembler Programm sowie der hexadezimale Java Byte Code?

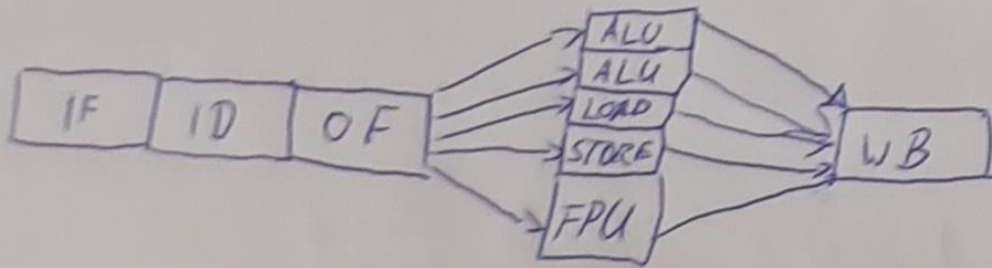
1	ILOAD i	0x15	0x03
2	ILOAD n	0x15	0x06
3	IADDSUB	0x64	
4	IADD BIPUSH 2	0x10	0x02
5	IADD	0x60	
6	ILOAD k	0x15	0x09
7	ISUB	0x64	
8	BIPUSH 9	0x10	0x09
9	ISUB	0x64	
10	ISTORE e	0x36	0x05

28

Aufgabe 5

(32 Punkte)

1. Skizzieren Sie den prinzipiellen Aufbau einer superskalaren Prozessorarchitektur!



6

2. Eine Modell-CPU besitze 8 (sichtbare) Arbeitsregister R0 - R7. Arithmetische Instruktionen benutzen jeweils 3 davon: 2 Quell- und 1 Ziel-Register. Findet die Dekodierung einer Instruktion im Zyklus n statt, dann beginnt die Ausführung dieser Instruktion im Zyklus n+1. Die add (+) Instruktion und die sub (-) Instruktion schreiben im Zyklus n+2 das Ergebnis zurück und komplexere Instruktionen (mul (*), div (/)) im Zyklus n+4. Die Dekodier-Einheit kann 2 Instruktionen pro Zyklus einleiten. „Register Renaming“ wird von der Modell-CPU **nicht** unterstützt! Folgendes Assembler-Programm wird von dieser CPU ausgeführt:

1. R5 = R3 * R2
2. R3 = R1 + R2
3. R6 = R5 / R2
4. R4 = R1 - R6
5. R7 = R3 + R5

Ermitteln Sie die Ausführungszeit (Anzahl der Takte) für die Bearbeitung dieser Instruktionsfolge durch Ausfüllen der beiden folgenden Tabellen für die In-Order- und Out-of-Order-Ausführung!

-2

Zn	Bn	Decodiert	Iss	Ret	zu lesende Register								zu beschreibende Register								
					0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
1	1	$R5 = R3 \cdot R2$	1				1	1										1			
	2	$R3 = R1 + R2$	2			1	2	1										1		1	
2	3	$R6 = R5 \cdot R$	-			1	2	1										1		1	
3						1	2	1										1		1	
4						1	2	1										1		1	
5				1		1	1											1			
				2																	
6			43				1			1										1	
	4	$R4 = R1 - R6$	-				1			1										1	
7							1			1										1	
8							1			1										1	
9							1			1										1	
10				3																1	
				4																1	
				5																1	
	5	$R7 = R3 + R5$																		1	
11			4			1				1										1	
			5			1				1										1	
12				4		1				1										1	
				5																	
13				4						1											1
				5						1											1
14																					
15																					

Tabelle 3: In-Order-Ausführung

(2)

Zn	Bn	Decodiert	Iss	Ret	zu lesende Register								zu beschreibende Register							
					0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	1	$R5 = R3 \cdot R2$	1																	
	2	$R2 = R1 + R2$	2				1	1											1	
2	3	$R6 = R5 / R2$	-				1	2	1										1	
	4	$R4 = R1 - R6$	-				1	2	1						1				1	
3	5	$R7 = R3 + R5$	-				1	2	1						1				1	
				2				1	1										1	
4								1	1										1	
5				1																
6			3					1			1									
			5					1	1		2								1	
7								1	1		2								1	
																			1	
8			5					1			1								1	
9								1			1								1	
10			3																	
11			4					1			1									
12																			1	
13																			1	
14			4																	
15																				

Tabelle 4: Out-of-Order-Ausführung

Aufgabe 6

(8 Punkte)

6

Gegeben sei ein ARM Cortex-M4 Mikroprozessor.

1. Register R1 enthält folgenden Wert:

 $[R1] = 0x00006000$

Was ist bei der Ausführung der folgenden Befehlszeile zu beachten?

BX R1

Man wechselt vom Thumb in den ARM-Modus

Bezug zum M4?

3

2. Welche Aufgabe hat das Link-Register?

Speicherung der aktuellen Rücksprungadresse bei Funktions- und Unterprogrammaufrufen

3