Digitale Systeme SS2019

Übungsblatt 4

Aufgabe 1 **(3 Punkte)**

Gegeben sei folgender Assemblerbefehl einer 2-Adressmaschine:

```
ADD 713(R0), (86)
```

In angegebenen Befehl werden folgende Adressierungsarten verwendet:

- n(Rx) : registerindiziert; mit n als Indexwert und RX als Register X
- (addr) : speicherdirekt mit addr Speicheradresse

Die Berechnungen erfolgen nach folgendem Schema:

- `Operand2 = Operand2 Operation Operand1`

Notieren Sie eine minimal kurze Assembler-Befehlsfolge, die diesen Befehl ersetzt!

Die von Ihnen dabei verwendbaren Adressierungsarten seien:

- RX - registerdirekte Adressierung
- (RX) - registerindirekte Adressierung
- #n - unmittelbare Adressierung

Verwenden Sie nur die Befehle `ADD, SUB, MOV`! Außer Register `R0` darf zusätzlich `R1` modifiziert werden!

Aufgabe 2 **(3 Punkte)**

Gegeben sei das folgende Programm einer 1-Adressmaschine. X und Y seien Label für Speicherstellen. `Rx` seien Universalregister. `#N` bezeichne einen Unmittelbaroperanden.

1	LDA	(X)
2	STA	R0
3	MUL	R0
4	ADD	R0
5	DIV	#2
6	STA	(Y)

(a) Welche Funktion F(X) implementiert dieses Programm?

(b) Implementieren sie diese Funktion für eine 0-Adressmaschine in einem möglichst kurzen Assemblerprogramm.

(c) Implementieren sie diese Funktion für eine 2-Adressmaschine in einem möglichst kurzen Assemblerprogramm.

(d) Implementieren sie diese Funktion für eine 3-Adressmaschine in einem möglichst kurzen Assemblerprogramm.

Hinweise:

Der Operand in der Speicherstelle X darf nicht zerstört (überschrieben) werden. Die Angabe des Namens eines Operators entspricht einer speicherdirekten Adressierung.

„(x)“ bedeute speicherdirekte Adressierung.

Verwenden Sie ausschließlich die folgenden Befehle:

- `PUSH` (load from memory to TOS), `POP` (store from TOS to memory), `ADD`, `MUL`, `SUB`, `DIV` sowie `ROT` (rotate Stack: a|b|c $\rightarrow$ c|a|b) und `DUP` (duplicate TOS) für die 0 - Adressmaschine

- `LDA` (lade Operanden in Akkumulator), `STA` (speichere Akkumulator in Operanden), `ADD, MUL, SUB, DIV` für eine 1 - Adressmaschine, Verwenden Sie bei Bedarf die Allzweckregister `R0 ... R7`

- `MOV, ADD, MUL, SUB, DIV` für eine 2 oder 3 - Adressmaschine, Verwenden Sie bei Bedarf die Allzweckregister `R0 ... R7`

- Befehle mit unmittelbar adressierten Operanden sollten ein # verwenden (z.B. `MUL #2,R1` oder `PUSH #2`)

Operandenanordnung bei arithmetischen Operationen:

0-Adressmaschine: `TOS = TOS Operation (TOS-1)`

1-Adressmaschine: `Akkumulator = Akkumulator Operation Operand`

2-Adressmaschine: `Operand2 = Operand1 Operation Operand2`

3-Adressmaschine: `Operand3 = Operand1 Operation Operand2`

Aufgabe 3 **(4 Punkte)**

Gegeben seien folgende Register- und Speicherinhalte eines Rechners: (alle Zahlen sind in Dezimaldarstellung)

	Ort	Inhalt
Register:	R1	0002
	R2	2000
	R3	1000
Speicher:	MEM[0002]	4000
	MEM[1000]	1000
	MEM[2000]	2000
	MEM[3000]	3000
	MEM[4000]	1000

Gegeben seien 3 Befehle des Prozessors, hier in Assembler-Mnemonic angegeben.

1. `ADD 1000(R3), R2`
2. `SUB (R1), R3`
3. `DIV #4000,R2`

Nehmen Sie eine Wortbreite von 16 Bit an. Jeder Befehl kann 1, 2 oder 3 Worte umfassen. Speicheradressen, Unmittelbar-Operanden oder Indizes belegen jeweils ein extra Wort. Das allgemeine Format der Mnemonics lautet:
`Befehl Operand1, Operand2.`
Operationen werden wie folgt ausgeführt:
`Operand2 = Operand1 operation Operand2`

Die Adressierungsarten seien wie folgt notiert:
`Rx` bezeichnet die Adressierungsart „Register direkt“
`#n` bezeichnet „unmittelbar“
`N(Rn)` bezeichnet „Register indiziert“
`(Rn)` bezeichnet „Register indirekt“

(a) Wie verändern sich die Register- und Speicherinhalte wenn jeder Befehl unabhängig von den anderen, d.h. stets mit initialen Werten, ausgeführt wird? Füllen Sie die Tabelle aus!

	R1	R2	R3	MEM[0002]	MEM[1000]	MEM[2000]	MEM[3000]	MEM[4000]
	0002	2000	1000	4000	1000	2000	3000	1000
1.								
2.								
3.								

(b) Wie verändern sich die Register- und Speicherinhalte während der Ausführung der 3 Befehle als Programm? Füllen Sie die Tabelle aus!

	R1	R2	R3	MEM[0002]	MEM[1000]	MEM[2000]	MEM[3000]	MEM[4000]
	0002	2000	1000	4000	1000	2000	3000	1000
1.								
2.								
3.								

(c) Wie viele Speicherzugriffe für Befehls- und Datenzugriffe erfolgen bei der Ausführung der Befehle?

Befehl	Anzahl der Speicherzugriffe
1.	
2.	
3.	