

Rechnerorganisation im WS 2020/21

Musterlösungen zum 2. Übungsblatt

Prof. Dr. Jörg Henkel
Dr.-Ing. Lars Bauer
Roman Lehmann, M. Sc.
Haid-und-Neu-Str. 7,
Geb. 07.21 (Technologiefabrik)
Email: roman.lehmann@kit.edu

Lösung 1

(4 Punkte)

1. Die Anzahl der Transistoren pro (Prozessor-) Chip verdoppelt sich alle zwei Jahre. 1 P.
Alternative Interpretationen die sich aus dem Moore'schen Gesetz ergeben: Die Verarbeitungsleistung der Hochleistungsprozessoren verdoppelt sich alle 18 Monate. Oder: Für den gleichen Preis erreicht die Mikroelektronik die doppelte Leistung in jeweils weniger als zwei Jahren (oder die 100 fache Leistung in jeweils 10 Jahren).
2. Ein Mikroprozessor (μP) besteht aus einem Steuerwerk (Leitwerk), einem Rechenwerk (Operationswerk), einer Schnittstelle zum Systembus, einem Adreßwerk, einem On-Chip Cache-Speicher, einer Speicherverwaltungseinheit. 1 P.
3. Bei einem Ein-Chip-Mikrocomputer sind vollständige Mikrocomputer mit Taktgenerator, Speicher, Schnittstellen, evtl. auch Zähler, A/D- bzw. D/A-Wandler usw. in einem einzigen Halbleiter-Baustein integriert. 1 P.
4. Bei der Bezeichnung *n-Bit*-Prozessor gibt *n* die Breite des Datenbusses oder die Breite der allgemeinen Register an. Das ist die Breite der Information, die in einem Schritt von der ALU im Rechenwerk verarbeitet werden kann. Bei einem 64-Bit-Prozessoren können 64-Bit breite Daten in einem Schritt verarbeitet werden. 1 P.

Lösung 2

(4 Punkte)

1. Im Statusregister können je nach Architektur verschiedene Informationen ausgelesen werden. Gebräuchliche Flags sind: 2 P.
 - Carry Flag (Übertrag aus höchstwertigen Bits)
 - Zero Flag (Ergebnis war Null)
 - Overflow Flag (Überlauf bei Operation)
 - Sign Flag (Vorzeichen des Ergebnisses)
 - Even Flag (Ergebnis war gerade)
 - Parity Flag (Ergebnis hat gerade Anzahl an Einsen)

2 P.

2. Beim arithmetischen Rechtsschieben findet eine sogenannte Vorzeichenerweiterung statt, d.h. führende Leerstellen werden mit dem vormalig höchstwertigen Bit aufgefüllt.

Dadurch entspricht das arithmetische Rechtsschieben bei beliebigen Zahlen im Zweierkomplement einer Division durch die entsprechende Zweierpotenz der geschobenen Bits.

Beim logischen Rechtsschieben wird immer mit Nullen aufgefüllt. Es gibt hier bei negativen Zahlen also keine anschauliche arithmetische Interpretation.

Lösung 3

(4 Punkte)

Die Befehlssequenz bildet das Einerkomplement aus dem Inhalt an Speicherstelle 0x2014 und speichert es an Speicherstelle 0x2015. Danach wird das Zweierkomplement gebildet (+1) und an der Speicherstelle 0x2016 gespeichert.

Kodierte Befehlsfolge:

```
0x102014 (0001 0000 0010 0000 0001 0100)
0xF10000 (1111 0001 0000 0000 0000 0000)
0x202015 (0010 0000 0010 0000 0001 0101)
```

Lösung 4

(7 Punkte)

1. Die Lese-Phase (*fetch phase*) erledigt folgende Dinge:

2 P.

- i.) Das InstruktionsAdressRegister in das SpeicherAdressRegister laden und das Lese-Flag auf 1 setzen ($R = 1$).
- ii.) Das InstruktionsAdressRegister in das X Register der ALU laden; das Eins Register in das Y Register laden und beide Werte addieren.
- iii.) Anschließend das berechnete Ergebnis in das InstruktionsAdressRegister speichern.
- iv.) Zum Schluss das aktuelle Datum aus dem SpeicherDatenRegister in das InstruktionsRegister laden.

Damit wird der aktuelle Befehl aus dem Speicher geholt und in das InstruktionsRegister kopiert. Zusätzlich wird das InstruktionsAdressRegister um den Wert 1 erhöht, um im nächsten Zyklus den nachfolgenden Befehl aus dem Speicher laden zu können.

2. Kodierte Mikroprogramm für die Lese-Phase (*fetch phase*):

5 P.

0x00	2 1 0 8 8 0 1	$X = P_w = S = 1; R = 1$
0x01	1 4 0 0 8 0 2	$Y = E = 1; R = 1$
0x02	0 0 0 1 8 0 3	$C_2 - C_0 = 001; R = 1$
0x03	0 A 0 0 0 0 4	$Z = P_r = 1$
0x04	0 0 9 0 0 0 5	$I_r = D_w = 1$

Lösung 5

(6 Punkte)

Bei den Mikroprogrammen handelt es sich um folgende MIMA-Befehle:

- | | |
|--------|---------------------------------------------------------------------------------------|
| 1. EQL | <div style="border: 1px solid black; padding: 2px; display: inline-block;">2 P.</div> |
| 2. JMP | <div style="border: 1px solid black; padding: 2px; display: inline-block;">1 P.</div> |
| 3. LDV | <div style="border: 1px solid black; padding: 2px; display: inline-block;">1 P.</div> |
| 4. LDC | <div style="border: 1px solid black; padding: 2px; display: inline-block;">1 P.</div> |
| 5. STV | <div style="border: 1px solid black; padding: 2px; display: inline-block;">1 P.</div> |

Lösung 6

(2 Punkte)

	wahr	falsch
Alle Maschinenbefehle der MIMA-Architektur haben die Länge von einem Datenwort, sind also 24 Bit lang.	×	
Die Lese- und Dekodierphase des Befehlszyklus laufen unabhängig vom auszuführenden Befehl immer identisch ab.	×	
Um -1 als Konstante in das Akku-Register zu laden, kann der LDC-Befehl (load constant) der MIMA-Architektur verwendet werden.		×
Die MIMA-Architektur kann 2^{24} Speicherzellen adressieren, da das Speicher-AdressRegister (SAR) eine Breite von 24 Bit besitzt		×