Karlsruhe Institute of Technology
Institut für Anthropomatik und Robotik (IAR)

Rechnerorganisation im Wintersemester 2017/18

Aufgaben zu den Tutorien in der Woche vom 13. bis 17. November 2017

Prof. Dr. Wolfgang Karl
Haid-und-Neu-Str. 7
Dr.-Ing. Ömer Terlemez
Adenauerring 2, Geb. 50.20
Email: ti@ira.uka.de
Web: http://ti.ira.uka.de

Lernziele:

- C-Sprache: Datentypen, Operatoren, Kontrollstrukturen, Zeiger(arithmetik) usw.
- Grundlagen der MIMA-Architektur (Aufbau, Befehlsformat, Mikrobefehlsformat etc.)
- Mikroprogrammierung

Aufgabe 1

Gegeben sei folgender C-Code, der auf einer 64-Bit-Platform ausgeführt wird:

```
#include <stdio.h>

int main()
{
    printf("sizeof(int) =  %lu\n", sizeof(int));
    printf("sizeof(long) = %lu\n", sizeof(long));

    int int_array[] = {1, 2, 3, 4, 5, 6};
    long long_array[] = {1, 2, 3, 4, 5, 6};

    int* int_pointer = int_array;
    long* long_pointer = long_array;

    printf("\nint_pointer  = %d\n", (int)int_pointer);
    printf("long_pointer = %d\n", (int)long_pointer);

    int_pointer++;
    long_pointer++;

    printf("\nint_pointer  = %d\n", (int)int_pointer);
    printf("long_pointer = %d\n", (int)long_pointer);

    int_pointer += 2;
    long_pointer += 3;

    printf("\n*int_pointer  = %d\n", *int_pointer);
    printf("*long_pointer = %ld\n", *long_pointer);

    printf("\n*int_pointer  = %d\n", *(int_pointer + 2));
    printf("*long_pointer = %ld\n", *(long_pointer + 2));

    return 0;
}
```

Vervollständigen Sie die Ausgabe:

```
sizeof(int) =  4
sizeof(long) = 8

int_pointer  = 1606416256
long_pointer = 1606416208

int_pointer  = ...
long_pointer = ...

*int_pointer  = ...
*long_pointer = ...

*int_pointer  = ...
*long_pointer = ...
```

Aufgabe 2

Geben Sie das Mikroprogramm für die Holphase, und Ausführungsphase des MIMA-Befehls

```
OR  a           # akku OR <a>  -> Akku
```

1. In Register-Transfer-Anweisung Schreibweise
2. In binärer Schreibweise

Aufgabe 3

Das bisher reservierte 8. Bit im Mikrobefehlsformat der MIMA sei mit D bezeichnet und wird als Kennzeichen dafür verwendet, dass die Adresse des nächsten Mikrobefehls aus dem Befehlsteil B_{23}-B_{16} ermittelt werden muss.
Wie sieht dann der Mikrobefehl für die Dekodierung (6. Takt) aus?

Aufgabe 4

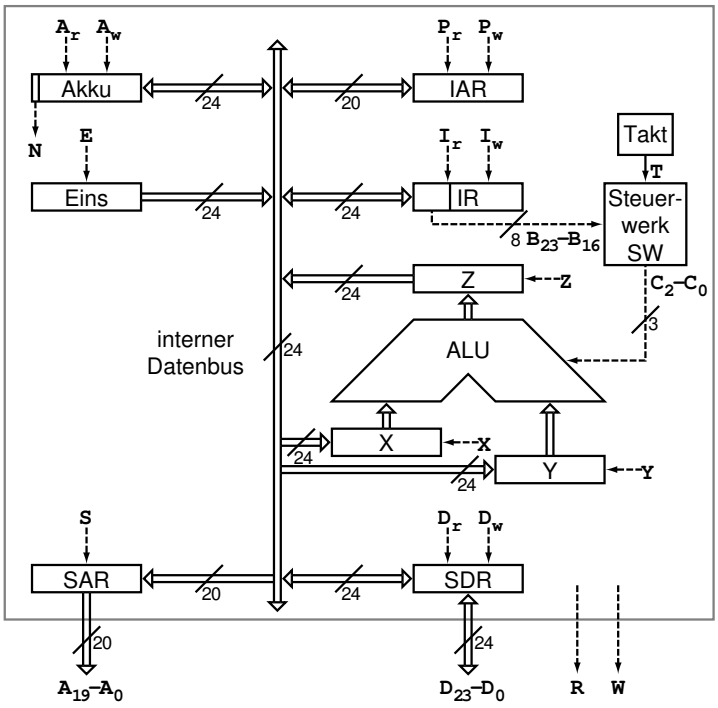
Geben Sie das Mikroprogramm für die Ausführungsphase des Maschinenbefehls für den bedingten Sprung (JMN) an (ab dem 7. Takt, also nach der Holphase und der Dekodierphase).

Verwenden Sie das 9. Bit (Bez.: B) entsprechend Aufgabe 2.

Aufgabe 5

Schreiben Sie ein Mikroprogramm in Register-Transfer-Schreibweise, das den Inhalt des Akkumulators als Zweierkomplement wieder im Akkumulator speichert. Beginnen Sie dabei ab Takt 7 der Befehlsabarbeitung.

Architektur der MIMA



$c_2c_1c_0$	*ALU Operation*
0 0 0	tue nichts (d.h. Z -> Z)
0 0 1	X + Y -> Z
0 1 0	rotiere X nach rechts -> Z
0 1 1	X AND Y -> Z
1 0 0	X OR Y -> Z
1 0 1	X XOR Y -> Z
1 1 0	Eins-Komplement von X -> Z
1 1 1	falls X = Y, -1 -> Z, sonst 0 -> Z

OpCode	*Mnemonik*	*Beschreibung*
0	LDC c	c -> Akku
1	LDV a	<a> -> Akku
2	STV a	Akku -> <a>
3	ADD a	Akku + <a> -> Akku
4	AND a	Akku AND <a> -> Akku
5	OR a	Akku OR <a> -> Akku
6	XOR a	Akku XOR <a> -> Akku
7	EQL a	falls Akku = <a>:-1 -> Akku sonst : 0 -> Akku
8	JMP a	a -> IAR
9	JMN a	falls Akku < 0 : a -> IAR
F0	HALT	stoppt die MIMA
F1	NOT	bilde Eins-Komplement von Akku -> Akku
F2	RAR	rotiere Akku eins nach rechts -> Akku

Register

Akku: Akkumulator
X: 1. ALU Operand
Y: 2. ALU Operand
Z: ALU Ergebnis
Eins: Konstante 1
IAR: Instruktionsadreßregister
IR: Instruktionsregister
SAR: Speicheradreßregister
SDR: Speicherdatenregister

Steuersignale vom SW

– für den internen Datenbus

A_r: Akku liest
A_w: Akku schreibt
X: X-Register liest
Y: Y-Register liest
Z: Z-Register schreibt
E: Eins-Register schreibt
P_r: IAR liest
P_w: IAR schreibt
I_r: IR liest
I_w: IR schreibt
D_r: SDR liest
D_w: SDR schreibt
S: SAR liest

– für die ALU

c_2–c_0: Operation auswählen

– für den Speicher

R: Leseanforderung
W: Schreibanforderung

Meldesignale zum SW

T: Takteingang
N: Vorzeichen des Akku
B_{23}–B_{16}: OpCode-Feld im IR

Befehlsformate