Name:
Vorname:
Matrikelnummer:
3. Versuch:

Klausur zur Vorlesung Technische Grundlagen der Informatik

Wintersemester 2022 - Dr. Michael Steppat

Hinweise:

- Die Aufgaben sind einzeln zu bearbeiten.
- Bitte jedes Blatt mit Ihrem Namen und Vornamen versehen.
- Jede Aufgabe auf dem Aufgabenblatt und dessen Rückseite lösen. Zusatzblätter dürfen verwendet werden. Diese bitte auch mit Namen versehen.
- Als Hilfsmittel sind nur Papier, Schreibzeug sowie handgeschriebene Aufzeichnungen, 1 DIN A4 Blatt (beidseitig) erlaubt.
- Mobiltelefone sind während der Klausur auszuschalten.
- Die Bearbeitungsdauer beträgt 90 Minuten.
- Viel Erfolg!

Name, Vorname:

1.1 Wandeln Sie die Hexadezimalzahl $B6A7537_{16}$ in eine Dualzahl. Beschreiben Sie kurz Ihre Vorgehensweise. (2 Punkte)

1.2 Wandeln Sie die Dezimalzahl 114 in eine Hexadezimalzahl und in eine Dualzahl. (3 Punkte)

2.1 Addieren Sie die beiden Binärzahlen 1011.1100_2 und 0011.0110_2 und geben Sie den Rechenweg an. (2 Punkte)

2.2 Rechnen Sie die beiden Summanden und das Ergebnis aus Aufgabe 2.1 in Dezimalzahlen um. (3 Punkte)

3.1 Welche Aufgaben übernehmen die Register in einem Prozessorsystem? Wie sind sie mit den restlichen Einheiten des Systems verbunden? (2 Punkte)

3.2 Skizzieren Sie das Rechenwerk eines Prozessorsystem. Welche Ein- und Ausgänge besitzt es. Beschreiben Sie deren Funktion. (4 Punkte)

4. Bei eines digitalen Beleuchtungssystem werden alle 65 Scheinwerfer über eine Adresse angesprochen. Über ein Datensignal kann die Helligkeit von 0-100 eingestellt werden.

4.1 Berechnen Sie die erforderliche Bitbreite für diese Werte. (2 Punkte)

4.2 Zeichnen Sie für die Scheinwerfer Nr. 20 und Nr. 35 das Schaltbild der für die Adressierung notwendigen Komparatoren. (Scheinwerfer Nr. 1 hat die Adresse 0) (4 Punkte)

5.1 Wie kann man mit Hilfe einzelner Volladdierer einen 4-Bit-Addierer aufbauen. Zeichnen Sie hierfür das zugehörige Schaltbild. (3 Punkte)

5.2 Wie kann man bei einer Programmausführung feststellen, an welcher Stelle man sich im Programm befindet? (2 Punkte)

6. Geben Sie für das folgende Assemblerprogramm die Ergebnisse der einzelnen Schritte im Hexadezimalcode an. Die Anfangswerte finden Sie in der ersten Zeile. (5 Punkte)

```
              AX    BX    CX    DX        [a]   [b]
              00    00    00    00        07    03
mov ax, [a]
or ax, 8
mov bx, [b]
div bx
shl ax, 2
add ax, 2
shr ax, 2
xor ax, ax
and ax, 3
mul bx
```

7. Zeichnen Sie für die folgende Wahrheitstabelle die zugehörige Schaltung und geben Sie die zugehörige Gleichung an. (4 Punkte)

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

8.1 Gegeben sind die Binärzahlen 0100.1101_2, 1111.1010_2 und 1111.0011_2.
Geben Sie deren Dezimalwerte im Einer und Zweierkomplement an. (3 Punkte)

8.2 Geben Sie für eine Binärzahl mit 15 Bit den minimalen und den maximalen Wert als Dezimalzahl und als Binärzahl im Zweierkomplement an. (2 Punkte)

8.3 Zeigen Sie , wie die Subtraktion 9-3 im Zweierkomplementcode durchgeführt wird. (2 Punkte)

Lösungen

Die Lösungen werden durch den Autor dieses Dokumentes bereitgestellt. Alle Angaben ohne Gewähr.

1.1

Eine Hexadezimalstelle kann vier Binärstellen abdecken. Zur besseren Leslichkeit werden alle vier Stellen ein Trennzeichen gesetzt.
$B_{16} = 1011_2$
$6_{16} = 1010_2$
$A_{16} = 1010_2$
$7_{16} = 0111_2$
$4_{16} = 0100_2$
$5_{16} = 0101_2$
$3_{16} = 0011_2$

$B6A7.4537_{16} = 1011.0110.1010.0111.0100.0101.0011.0111_2$

1.2

$114_{10} = 0111.0010_2$
$0111.0010_2 = 72_{16}$

2.1

Die Zeile mit den kleingedruckten Zahlen stellt den bei der Addition auftretenden Übertrag dar.

$$\begin{array}{cccccccccc} & 1 & 0 & 1 & 1 & . & 1 & 1 & 0 & 0 & _2 \\ + & 0 & 0 & 1 & 1 & . & 0 & 1 & 1 & 0 & _2 \\ & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & {\scriptstyle 1} & & & & \\ \hline & 1 & 1 & 1 & 1 & . & 0 & 0 & 1 & 0 & _2 \end{array}$$

2.2

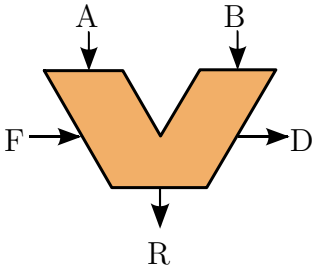
$1011.1100_2 = (1 \times 10^7 + 1 \times 10^5 + 1 \times 10^4 + 1 \times 10^3 + 1 \times 10^2)_{10} = 188_{10}$
$0011.0110_2 = (1 \times 10^5 + 1 \times 10^4 + 1 \times 10^2 + 1 \times 10^1)_{10} = 54_{10}$
$1111.0010_2 = (1 \times 10^7 + 1 \times 10^6 + 1 \times 10^5 + 1 \times 10^4 + 1 \times 10^1)_{10} = 242_{10}$

3.1

Die Register in einem Prozessorsystem dienen zur Zwischenablage der Daten, die direkt von dem Prozessor verarbeitet werden. Der Prozessor ist in der Lage, direkt Operationen auf diesen Registern durchzuführen.
In der Von-Neumann-Architektur sind die Register als Teil des Prozessors mit dem dreigliedrigen (Daten-, Steuer-, Adress-)Bus-System mit anderen Komponenten [des Rechners] verbunden.

3.2

Schematische Darstellung einer arithmetisch-logischen Einheit:

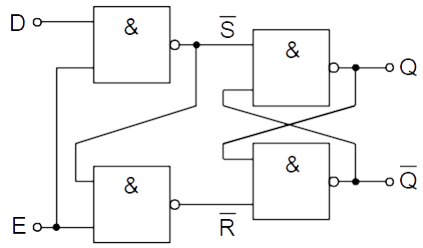


https://commons.wikimedia.org/wiki/File:ALU_symbol.svg, abgerufen am 06/02/2023

A, B: Operanden
F: Operator
D: Statusausgabe (auch Flags genannt)
R: Ergebnis

3.3

Ein Register zum Speichern eines Bits lässt sich beispielsweise mit einem taktpegelgesteuerten D-Flipflop/D-Latch aus NICHT-UND-Gattern realisieren. (siehe unten)



https://commons.wikimedia.org/wiki/File:FF_NAND-D.png, abgerufen am 06/02/2023

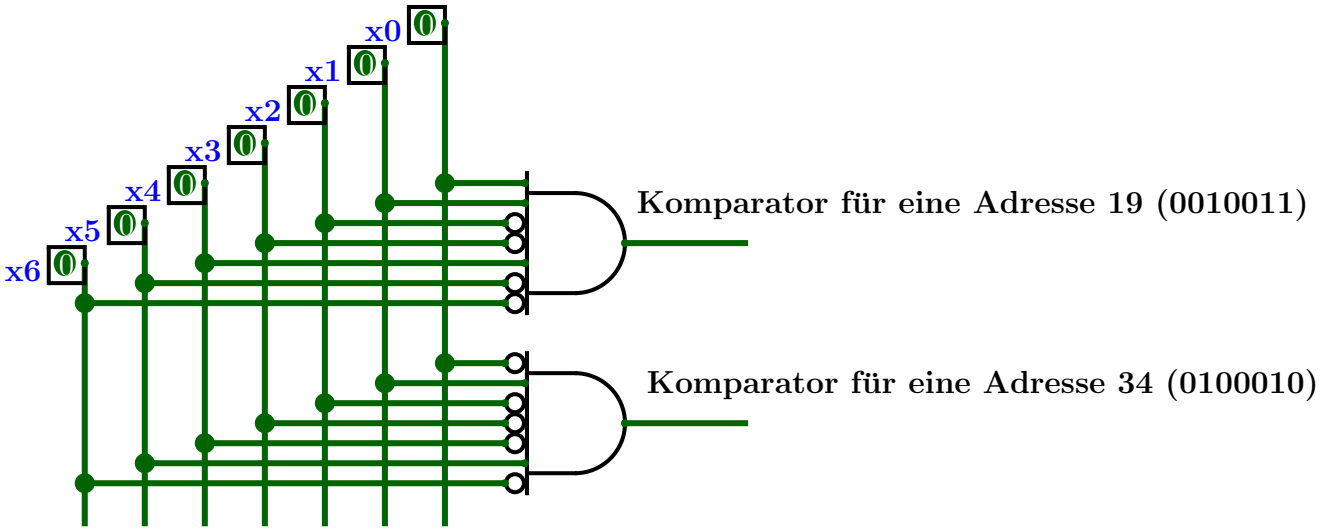
4.1

Bitbreite für Adressierung mit mindestens 65 möglichen Adressen:
$\lceil log_2(65) \rceil = 7$

Bitbreite für Adressierung mit mindestens 101 möglichen Adressen [Anm.: Der Autor geht davon aus, dass sowohl 0 als auch 100 valide Helligkeitsstufen sind]:
$\lceil log_2(101) \rceil = 7$

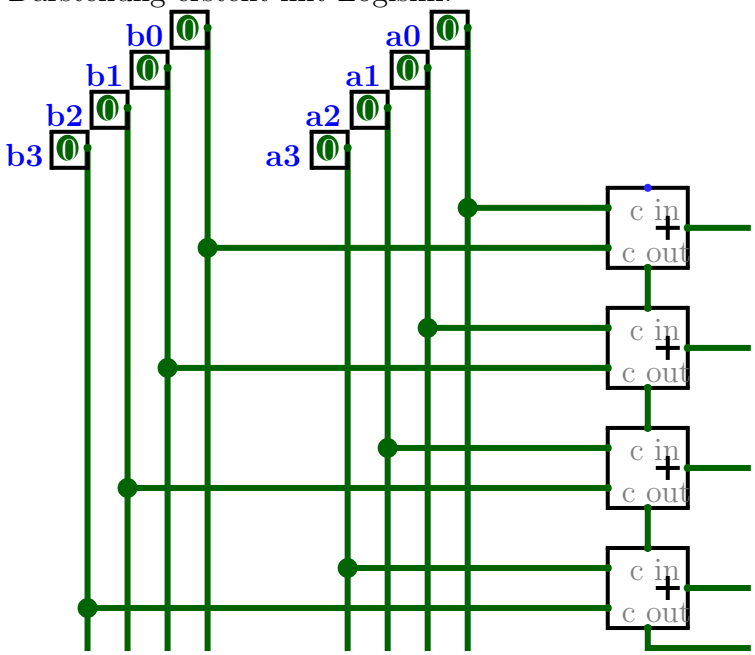
4.2

Der Autor geht hier davon aus, dass die Umwandlung von Dezimalzahlen in das Dualsystem keiner weiteren Erläuterung erfordert.
Darstellung erstellt mit Logisim.



5.1

Darstellung erstellt mit Logisim.



5.2

Der Programmzähler (englisch *Instruction Pointer*) verweist auf die Stelle im Programmspeicher, bei der sich der Prozessor im Programmfluss befindet.

6

	AX	BX	CX	DX	[a]	[b]
	00	00	00	00	07	03
mov ax, [a]	07					
or ax, 8	0F					
mov bx, [b]		03				
div bx	05					
shl ax, 2	14					
add ax, 2	16					
shr ax, 2	05					
xor ax, ax	00					
and ax, 3	00					
mul bx	00					

7

Konjunktive Normalform von Y:
$(A \vee \neg B \vee \neg C) \wedge (\neg A \vee \neg B \vee C))$

8.1: (8-Bit Einer- und Zweierkomplementcode)

0100.1101_2

Einerkomplement: 77_{10}
Zweierkomplement: 77_{10}

1111.1010_2

Einerkomplement: $(-5)_{10}$*
Zweierkomplement: $(-6)_{10}$

1111.0011_2

Einerkomplement: $(-12)_{10}$*
Zweierkomplement: $(-13)_{10}$

* Laut Anmerkungen von Herrn Steppat seien 1111.1010_2=250_{10} und 1111.0011_2=243_{10} im 8-Bit-Einerkomplement. Dann wurden die Binärzahlen weder im Einer- oder Zweierkomplement, sondern als vorzeichenfreie Zahl interpretiert.

8.2

Maximal: $16383_{10} = 011.1111.1111.1111_2$
Minimal: $-16384_{10} = 100.0000.0000.0000_2$

8.3

Unter Verwendung von 5-Bit-Zweierkomplement
$9_{10} = 01001_2$
$(-3)_{10} = 11101_2$

$$\begin{array}{rr} & 01001_2 \\ + & 11101_2 \\ \hline & 00110_2 \end{array}$$