

Vorname: _____

Name: _____

Matr.-Nr.: _____

Note:

27.02.2006 - 10⁰⁰ Uhr bis 12⁰⁰ Uhr

UNIVERSITÄT KARLSRUHE
Institut für Industrielle Informationstechnik
- Prof. Dr.-Ing. habil. K. Dostert -

Diplomprüfung im Fach

"Mikrorechnertechnik"

Musterlösung F06

Aufgabe:	1	2	3	4	5	6	7	8	9	10	gesamt
Punkte:											
Erreichbare Punktzahl:	9	10	9	10	10	10	11	10	10	11	100

Aufgabe 1: Zahlendarstellung in Mikrorechnerprogrammen

9 Punkte

a) $-2^{31} + 2^{28} + 2^{26} + 2^{25} + 2^{22} + 2^{21} + 2^{19} + 2^{16} + 2^{12} + 2^{10} + 2^8 + 2^5 + 2^3 = -1771498200$

b) 96691528

c) 1001 0110 0110 1001 0001 0101 0010 1000_b

=> Fraktal:

$$-1 + \frac{2^{28} + 2^{26} + 2^{25} + 2^{22} + 2^{21} + 2^{19} + 2^{16} + 2^{12} + 2^{10} + 2^8 + 2^5 + 2^3}{2^{31}} = -0,82491813227534$$

d) 1001 0110 0110 1001 0001 0101 0010 1000_b

=> VZ= negativ

Exponent: 0010 1100_b = 44 => Exp = 44 - 127 = -83_d

Mantisse: m = 110 1001 0001 0101 0010 1000

$$\begin{aligned} \text{Zahl} &= (-1)^1 \cdot [1 + m] \cdot 2^{\text{Exp}} \\ &= (-1)^1 \cdot [1 + (2^{-1} + 2^{-2} + 2^{-4} + 2^{-7} + 2^{-11} + 2^{-13} + 2^{-15} + 2^{-18} + 2^{-20})] \cdot 2^{-83} \\ &= (-1)^1 \cdot [1 + 2^{-20} \cdot (2^{19} + 2^{18} + 2^{16} + 2^{13} + 2^9 + 2^7 + 2^5 + 2^2 + 1)] \cdot 2^{-83} \\ &= (-1)^1 \cdot 2^{-20} \cdot [2^{20} + 2^{19} + 2^{18} + 2^{16} + 2^{13} + 2^9 + 2^7 + 2^5 + 2^2 + 1] \cdot 2^{-83} \\ &= -\frac{1909413}{2^{103}} \quad (\approx -1.8828\text{e-}025) \end{aligned}$$

e) $-12\frac{3}{64} = (-1)^1 \cdot 12\frac{3}{64}$

$$\begin{aligned} &= (-1)^1 \cdot [2^3 + 2^2 + \frac{1}{32} + \frac{1}{64}] = (-1)^1 \cdot [2^3 + 2^2 + 2^{-5} + 2^{-6}] \\ &= (-1)^1 \cdot 2^3 \cdot [1 + 2^{-1} + 2^{-8} + 2^{-9}] \\ &= (-1)^1 \cdot [1 + 2^{-1} + 2^{-8} + 2^{-9}] \cdot 2^{130-127} \end{aligned}$$

=> VZ= negativ

Exponent: 130_d = 1000 0010_b

Mantisse: m = 100 0000 1100 0000 0000 0000

=> Gleitkommazahl = 1100 0001 0100 0000 1100 0000 0000 0000

f)

Betrag der Zahl $12\frac{3}{64} = 0000\ 0000\ 0000\ 1100 \cdot 0000\ 1100\ 0000\ 0000$

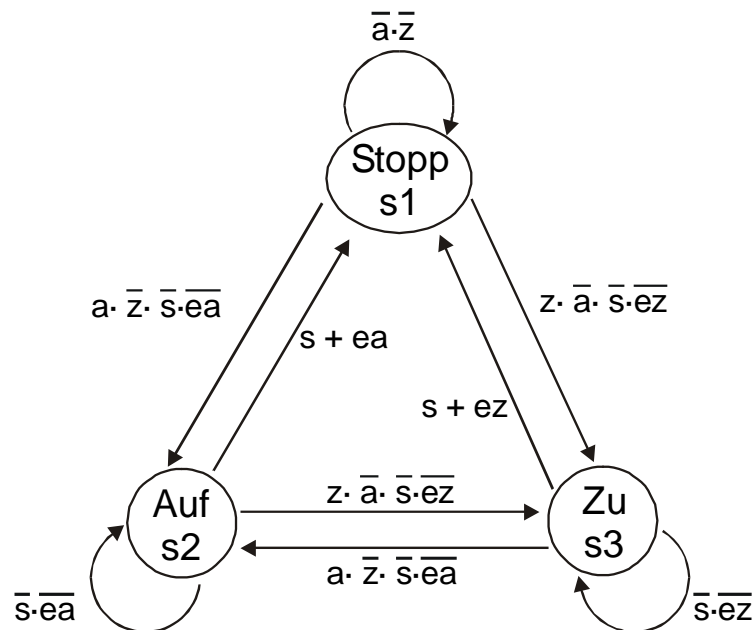
Davon das 2er-Komplement durch Negation und Addition von 1 an der niederwertigsten Stelle, also nicht an der ersten Vorkommastelle:

$$\Rightarrow -12\frac{3}{64} = 1111\ 1111\ 1111\ 0011 \cdot 1111\ 0100\ 0000\ 0000$$

Aufgabe 2: FSM

a), c)

10 Punkte



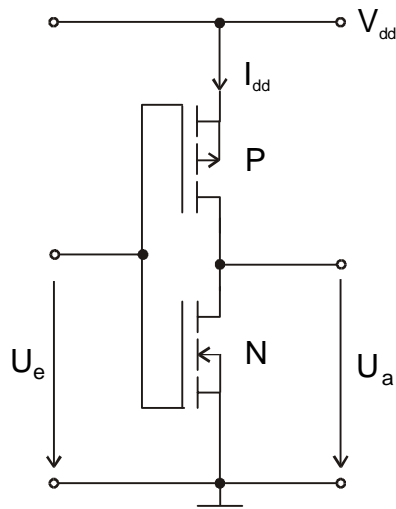
b), c)

relevant =	a,	z,	s,	ea,	ez;	
s1,	1	0	0	0	X,	f2;
s1,	0	1	0	X	0,	f3;
s2,	X	X	0	0	X,	f2;
s2,	0	1	0	X	0,	f3;
s3,	1	0	0	0	X,	f2;
s3,	X	X	0	X	0,	f3;

Aufgabe 3: MOS-Technologie

10 Punkte

a)



b) CMOS-Inverter

c)

$U_e = H$:

Der obere Transistor (p-Kanal) sperrt, der untere Transistor (n-Kanal) leitet; damit liegt der Ausgang an Masse, also auf logisch L.

$U_e = L$:

Der obere Transistor (p-Kanal) leitet, der untere (n-Kanal) sperrt; damit liegt der Ausgang an V_{dd} , also auf logisch H.

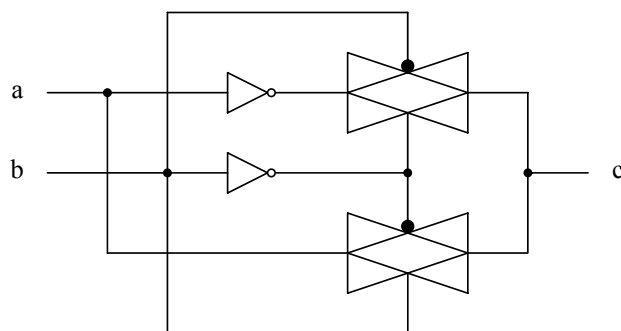
Aufgabe 4: CMOS-Transfergates

10 Punkte

a)

a	b	c
0	0	1
0	1	0
1	0	0
1	1	1

b)



c)

S0	S1	Y
0	0	X1
0	1	X3
1	0	X0
1	1	X2

d) Es handelt sich um einen 4-zu-1-Multiplexer.

10 Punkte

a) Zustandstabelle eines Vollsubtrahierers:

a_i	b_i	c_i	d_i	e_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

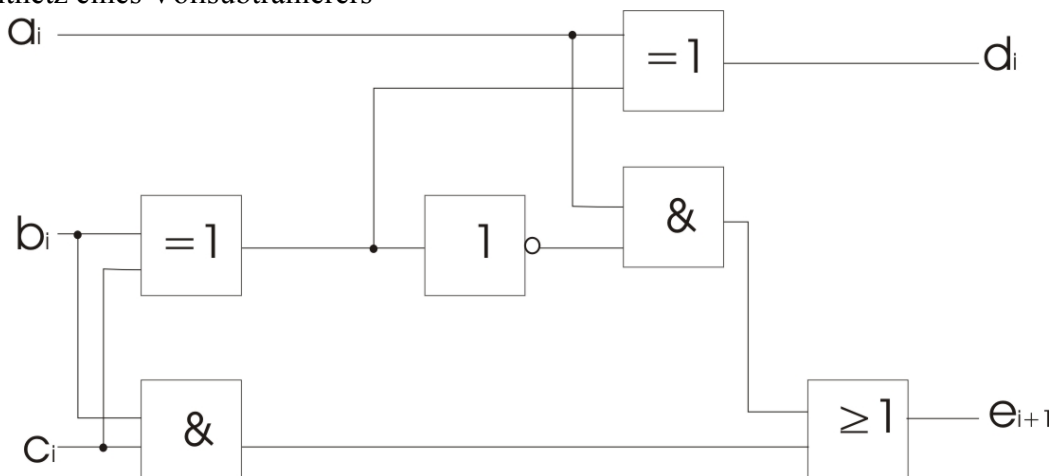
b)

Differenz: $d_i = a_i \oplus b_i \oplus c_i = \overline{a_i} \cdot \overline{b_i} \cdot c_i + \overline{a_i} \cdot b_i \cdot \overline{c_i} + a_i \cdot \overline{b_i} \cdot \overline{c_i} + a_i \cdot b_i \cdot c_i$

Entleihung:

$$e_i = \overline{a_i} \cdot \overline{b_i} \cdot c_i + \overline{a_i} \cdot b_i \cdot \overline{c_i} + \overline{a_i} \cdot b_i \cdot c_i + a_i \cdot \overline{b_i} \cdot \overline{c_i} = \overline{a_i}(\overline{b_i}c_i + b_i\overline{c_i}) + b_i c_i = \overline{a_i}(b_i \oplus c_i) + b_i c_i$$

c) Schaltnetz eines Vollsubtrahierers



Aufgabe 6: Mikrocontrollerprogrammierung mit dem 80C51

10 Punkte

a) Unterprogramm

```
START_SUBROUTINE:
    MOV A, R2
    MOV R1, A
    MOV A, R0
LOOP:
    JZ FERTIG
    MUL B, R1
    MUL A, B
    MOV R1, A
    MOV A, R0
    SUBB A, R2
    MOV R0, A
    JMP LOOP
FERTIG:
    RET
```

b) Registerinhalte

Vorher: R0: 6_d, R1: 4_d, R2: 3_d

Nachher: R0: 0000 0000_b, (=0_d)
 R1: 0011 0110_b, (=54_d)
 R2: 0000 0011_b, (=3_d)

c) Registerinhalte

Das Ergebnis steht in R1 und entspricht der Berechnung der Fakultät der Zahl, die sich vor dem Programmstart in R1 befand.

Aufgabe 7: Mikrocontrollerprogrammierung

11 Punkte

- a) Der Aufruf der Interruptroutine soll alle 10 ms erfolgen.

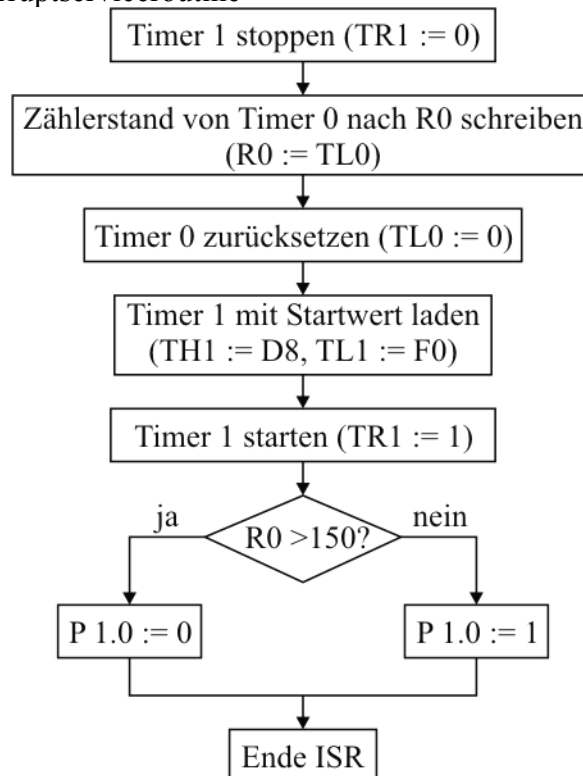
Dafür benötigte Zählschritte für Timer 1: $n = \frac{12 \text{ MHz}}{12} \cdot 10 \text{ ms} = 10.000$

Timer 1 muss als 16 bit-Timer betrieben werden (ohne Autoreload) \Rightarrow Mode 1
Reloadwert: $65.536 - 10.000 = 55.536 = \text{D8F0 h}$

- b) In 10 ms werden durchschnittlich maximal 200 Impulse gezählt
 \Rightarrow Timer 0 als 13 bit-Zähler (Mode 0) oder 16 bit-Zähler (Mode 1),
es werden aber nur 8 bit zum Zählen benötigt.

Bit	7	6	5	4	3	2	1	0
IE	1	X	X	X	1	X	0/X	X
TMOD	0	0	0	1	0	1	0	0/1
TCON	0	1	0	1	X	X	X	X

- c) Flussdiagramm der Interruptserviceroutine

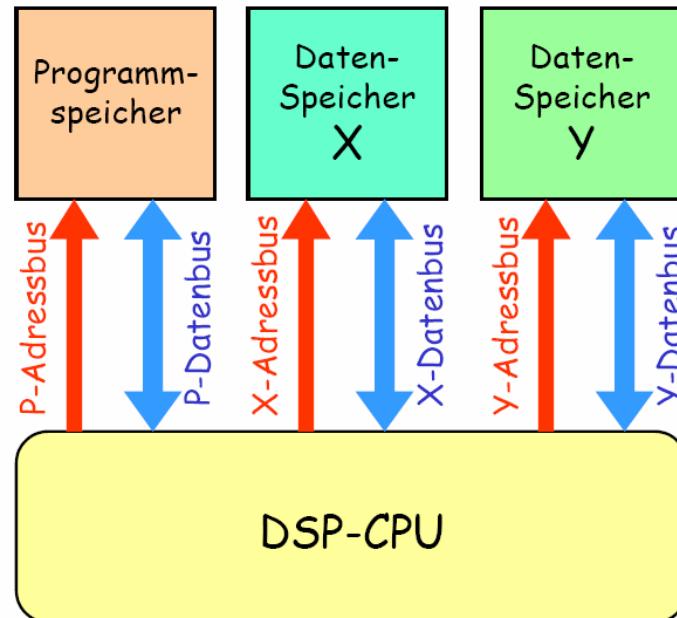


Aufgabe 8: Architektur von DSPs und Analyse von Assemblercode

11 Punkte

a) Harvard-Architektur

b)



c) Modify-Register. M2 wirkt sich auf R2 aus.

d) Es wird ein Ringspeicher der Länge 257 initialisiert.

e)

	R0 (S)	N0	R0 (E)	A	Bezeichnung
MOVE X:(R0),A	\$152	beliebig	\$152	\$150	Indirekt ohne Pointermanipulation
MOVE X:(R0)-N0,A	\$153	\$2	\$151	\$14F	Indirekt mit Postdekrement mit Offset
MOVE R0+N0,A	\$153	\$1	\$153	\$154	Direkt mit Offset

Aufgabe 9: Beschreibung von Schaltungen mit VHDL

10 Punkte

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity mux_4_1 is
port (s: in  std_logic_vector(1 downto 0);
      e: in  std_logic_vector(3 downto 0);
      a: out std_logic);
end mux_4_1;

architecture mux_4_1_arch of mux_4_1 is

begin

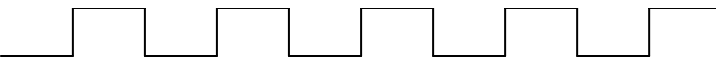
process(s)
begin
    case s is
        when "00" => a <= e(0);
        when "01" => a <= e(1);
        when "10" => a <= e(2);
        when "11" => a <= e(3);
    end case;
end process;

end mux_4_1_arch;
```

Aufgabe 10: Eigenschaften und Analyse von Schaltungen mit VHDL


11 Punkte

a)

clk 

x	7	3	6	1	8
y1	7	3	6	1	8
y2	8	4	7	2	9
y3	7	3	6	1	8

b)

clk 

x	7	3	6	1	8
y1	X	7	3	6	1
y2	X	X	8	4	7
y3	7	3	6	1	8

c) Signale, Variable, Konstanten.

d) Nachdem Schlüsselwort Process folgt in Klammern die sogenannte Empfindlichkeitsliste (Sensitivity List). Sie beinhaltet die Signale, auf deren Änderung der Prozess wartet, d.h. er wird nur dann aktiviert, wenn eine solche Änderung auftritt.

d) Mit VHDL wird eine Hardware beschrieben. Im Gegensatz zu herkömmlichen Sprachen wie C, mit denen eine Software beschrieben wird, können mit VHDL parallele und sequentielle Vorgänge beschrieben werden. Mit herkömmlichen Programmiersprachen zur Entwicklung von Software können nur sequentielle Abläufe entwickelt werden.