

**Vorname:** \_\_\_\_\_

**Name:** \_\_\_\_\_

**Matr.-Nr.:** \_\_\_\_\_

**Note:**

28.08.2006, 14.00 Uhr – 16.00 Uhr

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

**Diplomprüfung im Fach**

***"Mikrorechnertechnik"***

## **Musterlösung H06**

Aufgabe:	1	2	3	4	5	6	7	8	9	10	gesamt
Punkte:											
Erreichbare Punktzahl:	10	9	11	10	10	11	9	9	12	9	100

## Aufgabe 1: Speicher und speicherbasierte Anwendungen

10 Punkte

a)

EPROM: electrically programmable read only memory → beschreibbarer Lesespeicher

Ja, ein EPROM kann mittels UV-Licht gelöscht werden.

b)

RAM: random access memory

SRAM: statisches RAM

DRAM: dynamisches RAM

Speicherung der Information in DRAM-Zelle mittels Kondensator.

c)

Die DRAM-Zelle benötigt einen Refresh. Refresh bezeichnet das Auffrischen der gespeicherten Information, da die Information aufgrund von Entladevorgängen ansonsten ohne Refresh verloren gehen würde.

d)

3.75 GByte = 4026531840 Byte = 32212254720 Bit

$$y = (2^x)^2 \cdot 2 \cdot x \Rightarrow x = 15$$

## Aufgabe 2: Zahlendarstellung in Mikrorechnerprogrammen

9 Punkte

a)  $1110\ 0010_b = 2^7 + 2^6 + 2^5 + 2^1 = 226$

b) Als 2er-Komplementdarstellung:  $1110\ 0010_b = -2^7 + 2^6 + 2^5 + 2^1 = -30$   
Als Betrags-Vorzeichendarstellung:  $1110\ 0010_b = -(2^6 + 2^5 + 2^1) = -98$

c)  $1110\ 0010_b = -1 + \frac{2^6 + 2^5 + 2^1}{2^7} = -0,234375$

d)

1100	0100	1001	1000	0000	0000	0000	0000
------	------	------	------	------	------	------	------

=> VZ= negativ

Exponent:  $1000\ 1001_b = 137 \Rightarrow \text{Exp} = 137 - 127 = 10_d$

Mantisse: m=  $001\ 1000\ 0000\ 0000\ 0000\ 0000$

$$\begin{aligned}\text{Dezimalwert} &= (-1)^1 \cdot [1 + m] \cdot 2^{\text{Exp}} \\ &= (-1)^1 \cdot [1 + (2^{-3} + 2^{-4})] \cdot 2^{10} \\ &= -1216\end{aligned}$$

e)

$$\text{Dezimalwert} = -1 + \frac{1}{2} + \frac{1}{2^5} + \frac{1}{2^8} + \frac{1}{2^{11}} + \frac{1}{2^{12}} = -\frac{1901}{2^{12}} = -0,4641$$

f)  $-\left(10 + \frac{3}{16}\right) = (-1)^1 \cdot 10 \frac{3}{16}$

$$\begin{aligned}&= (-1)^1 \cdot \left[2^3 + 2^1 + \frac{1}{8} + \frac{1}{16}\right] = (-1)^1 \cdot [2^3 + 2^1 + 2^{-3} + 2^{-4}] \\ &= (-1)^1 \cdot 2^3 \cdot [1 + 2^{-2} + 2^{-6} + 2^{-7}] \\ &= (-1)^1 \cdot [1 + 2^{-2} + 2^{-6} + 2^{-7}] \cdot 2^{130-127}\end{aligned}$$

=> VZ= negativ

Exponent:  $130_d = 1000\ 0010_b$

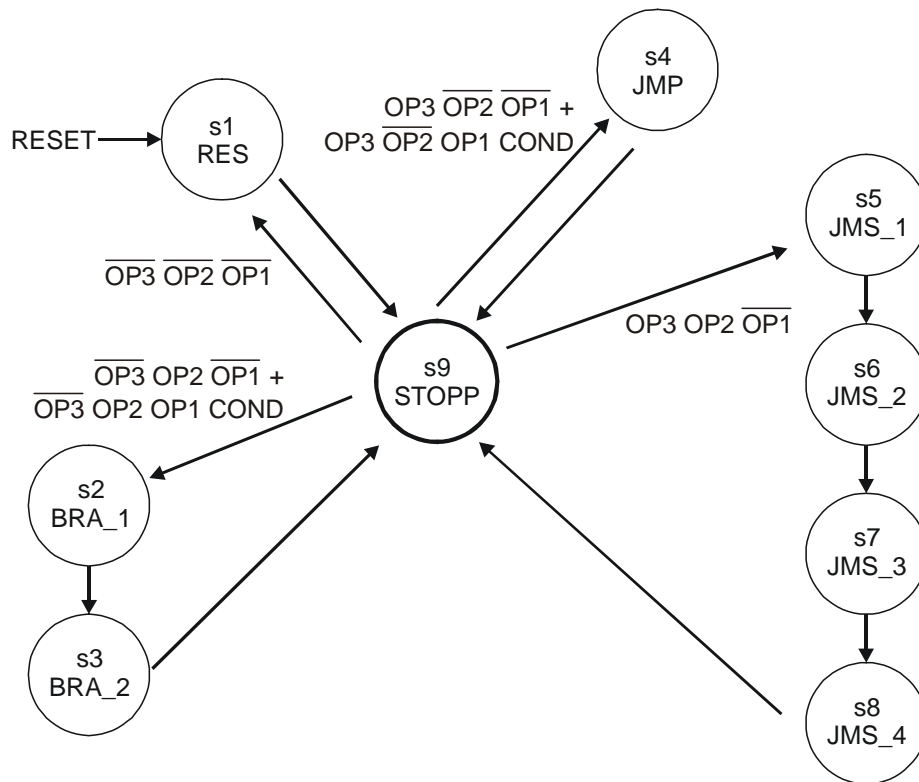
Mantisse: m=  $010\ 0011\ 0000\ 0000\ 0000\ 0000$

=> Gleitkommazahl=  $1100\ 0001\ 0010\ 0011\ 0000\ 0000\ 0000\ 0000$

### Aufgabe 3: FSM

11 Punkte

a), b)



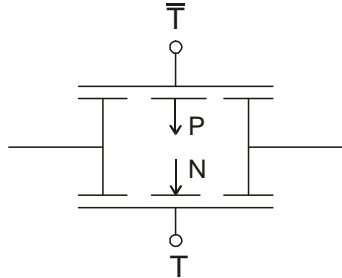
c)

Befehl	OPC3	OPC2	OPC1
unbedingte Verzweigung	0	1	0
Sprung in Unterprogramm	1	1	0
Reset	0	0	0
bedingter Sprung	1	0	1

#### Aufgabe 4: CMOS-Transfergates

10 Punkte

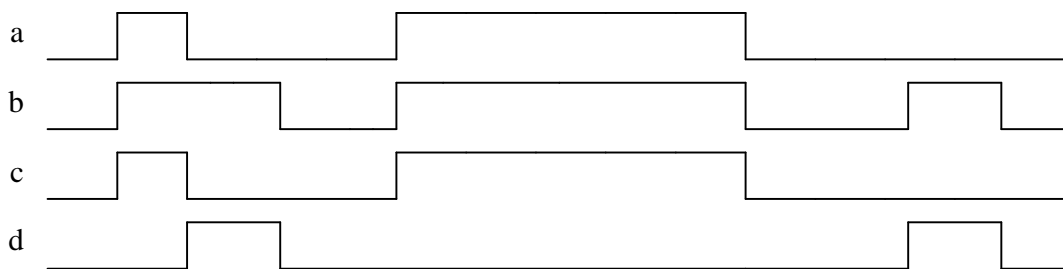
a)



b)

Transfergate leitet für  $T = 1, \bar{T} = 0$   
Transfergate sperrt für  $T = 0, \bar{T} = 1$

c)



d)

Funktion: Halbaddierer

<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

### Aufgabe 5: Arithmetik-Schaltungen

10 Punkte

a) Zustandstabelle eines Volladdierers:

$a$	$b$	$c$	$d$	$e$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

b)  
Volladdierer mit Summenbit  $d$  und Übertragsbit  $e$ .

c)  
Längster Pfad: 6 Gatter  $\Rightarrow$  18 ns.

d)  
$$f = \frac{1}{T} = \frac{1}{18\text{ns}} = 55,56\text{MHz}.$$

## Aufgabe 6: Serielle Schnittstelle des 80C51

11 Punkte

a)

Simplexbetrieb: Informationsfluss nur in eine Richtung möglich  
Halbduplexbetrieb: Informationsfluss in beide Richtungen möglich,  
aber nicht gleichzeitig  
Voll duplex: Informationsfluss in beide Richtungen gleichzeitig möglich

b)

Zählerschritte von Timer 1 bis zum Überlauf:  $100_{16} - 80_{16} = 80_{16} = 128_{10}$   
Eingangstakt von Timer 1:  $f_{osc}/12$   
Eingangsteiler der seriellen Schnittstelle: Timer 1-Überlaufrate dividiert durch  
16 (SMOD = 1) oder 32 (SMOD = 0)

$$\Rightarrow \text{Baudrate (SMOD=0)} = \frac{10 \text{ MHz}}{12} \cdot \frac{1}{32} \cdot \frac{1}{128} = 203,45 \text{ baud}$$

$$\text{Baudrate (SMOD=1)} = \frac{10 \text{ MHz}}{12} \cdot \frac{1}{16} \cdot \frac{1}{128} = 406,90 \text{ baud}$$

$$\text{oder: Baudrate (SMOD)} = \frac{10 \text{ MHz}}{12} \cdot \frac{1}{2^{5-\text{SMOD}}} \cdot \frac{1}{128} = \frac{6510}{2^{5-\text{SMOD}}} \text{ baud} = 203,45 \cdot 2^{\text{SMOD}} \text{ baud}$$

c)

	Bit 7				Bit 0			
SCON	0	1	X/0/1	1	X	X	0	0
TCON	X/0	1	X	X	X	X	X	X
TMOD	0	0	1	0	X	X	X	X
IE	1	X	X	1	0/X	X	X	X

d)

Maximale Baudrate: TH1 = 255 (1 Zählerschritt bis zum Überlauf), SMOD = 1

$$\Rightarrow \text{Baudrate} = \frac{10 \text{ MHz}}{12} \cdot \frac{1}{16} = 52083 \text{ baud}$$

## Aufgabe 7: Mikrocontrollerprogrammierung

9 Punkte

a)

Beim 80C51 handelt es sich um einen 8 bit Mikrocontroller, daher darf der Parameter  $a_3$  maximal 255 betragen.

b)

Die Zufallszahlenfolge lautet:

$X_0 = 3 \Rightarrow X_1 = 64 \Rightarrow X_2 = 21$ .

c) Programmcode

```
;*****
DSEG

FIRSTRND EQU    3H    ; Anfangswert  $X_n$  für Zufallsgenerierung

RANDOM: DS 1

;*****
CSEG
;*****

CALL GETRND

; Zufallszahl initialisieren -----

INITRND:
    MOV     RANDOM, #FIRSTRND    ; festen Anfangswert setzen
    RET

;*****

; Zufallszahl berechnen (Algorithmus nach D.H. Lehmer (1948))-----
;  $X_{n+1} = (a_1 * X_n + a_2) \bmod a_3$ , mit  $a_1 = 37$ ,  $a_2 = 53$ ,  $a_3 = 100$ 

GETRND:
    CALL INITRND                ; den Wert  $X_n$  initialisieren
    MOV     A, RANDOM            ; den Wert  $X_n$  in Akku A laden
    MOV     B, #37
    MUL     AB                  ; das Produkt  $a_1 * X_n$  berechnen
    ADD     A, #53
    MOV     B, #100
    DIV     AB                  ; Modulo-Berechnung
    MOV     RANDOM, B           ; den Wert  $X_{n+1}$  nach RANDOM schreiben
    RET

;*****
```



**Aufgabe 8: Architektur von DSPs und Analyse von Assemblercode****9 Punkte**

a)

Register	Inhalt nach Befehl 1	Inhalt nach Befehl 2	Inhalt nach Befehl 3
R0	\$0002	\$0000	\$0000
X0	\$E00000	\$C00000	\$000003
R4	\$0005	\$0005	\$0004
Y0	\$060000	\$000005	\$050000

b)

X0: \$400000 = 0,5 dezimal

Y0: \$400000 = 0,5 dezimal

 $\Rightarrow X0 \cdot Y0 = 0,25$ 

A: \$200000 = 0,25

 $\Rightarrow X0 \cdot Y0 + A = 0,25 + 0,25 = 0,5$ 

A2	A1	A0
\$00	\$400000	\$000000

## Aufgabe 9: Simulation von VHDL-Modellen

12 Punkte

a)  
MAC-Einheit

b)  
entity test is  
end test;

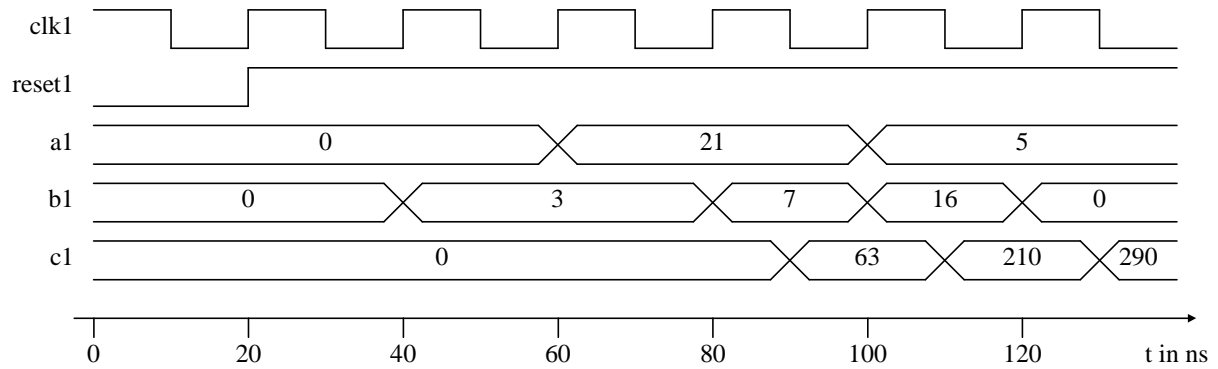
Die Entity einer Testbench muss leer sein.

c)  
architecture test\_arch of test is  
  
signal clk1 : std\_logic;  
signal reset1: std\_logic;  
signal a1, b1: integer range 0 to 127;  
signal c1 : integer range 0 to 145160;  
  
component exam  
port (clk : in std\_logic;  
reset : in std\_logic;  
a, b : in integer range 0 to 127;  
c: out integer range 0 to 145160);  
end component;  
  
begin  
  
exam\_comp : exam  
port map (clk1, reset1, a1, b1, c1);  
  
process  
begin  
clk1 <= '1';  
wait for 10 ns;  
clk1 <= '0';  
wait for 10 ns;  
end process;  
  
reset1 <= '0',  
          '1' after 20 ns;  
  
a1 <= 0,  
      21 after 60 ns,  
      5 after 100 ns;  
  
b1 <= 0,  
      3 after 40 ns,  
      7 after 80 ns,  
      16 after 100 ns,  
      0 after 120 ns;  
  
end test\_arch;

d)

$$3 \cdot 21 + 7 \cdot 21 + 5 \cdot 16 = 290$$

e)



Signale werden um einen Taktschritt verzögert übergeben.

### Aufgabe 10: Logikbeschreibung mit VHDL

9 Punkte

a)

Die Schaltung hat zwei Eingänge vom Typ „bit“ und einen Ausgang von Typ „bit“, die entweder den Zustand '0' oder '1' annehmen können.

b)

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

Verknüpfungsgleichung:  $c = \overline{a \oplus b}$

c)

```
ARCHITECTURE g2 OF K_Gatter IS
  BEGIN
    g <= '1' WHEN d = '1' AND e = '0' AND f = '1' ELSE
      '0';
  END g2;
```