

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

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

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

**Note:** \_\_\_\_\_

04.10.2007 - 14<sup>00</sup> Uhr bis 16<sup>00</sup> Uhr

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

**Diplomprüfung im Fach**

***"Mikrorechnertechnik"***

## **Musterlösung H07**

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

## Aufgabe 1: Parallele Multiplikation

10 Punkte

a)

Für die Adresse werden  $2 \times 4 = 8$  Bits benötigt.

Für den Datenausgang werden ebenfalls 8 Bits benötigt, um alle  $2^8$  Speicherstellen auslesen zu können.

b)

Adresse	Inhalt
0000	0000
0001	0000
0010	0000
0011	0000
0100	0000
0101	0001
0110	0010
0111	0011
1000	0000
1001	0010
1010	0100
1011	0110
1100	0000
1101	0011
1110	0110
1111	1001

Tabelle 1.1: Adresse und Speicherinhalt

c)

Jeder der Schritte 1-4 benötigt genau eine Volladdiererlaufzeit; hinzu noch die Zeit für die Gatter-Multiplikationen sowie die Abschlussaddition

$$\Rightarrow \text{Gesamtlaufzeit} = 4 \cdot \tau_{VA} + 1,5 \text{ ns} + 2,5 \text{ ns} = 4 \cdot 2,5 \text{ ns} + 1,5 \text{ ns} + 2,5 \text{ ns} = 14 \text{ ns}.$$

d)

$$f_{\max} = 1 / (2,5 \text{ ns} + 1,5 \text{ ns} + 1 \text{ ns}) = 200 \text{ MHz}$$

e)

Der Register muss alle Ergebnisbits vom Schritt 3 Speichern, d.h. pro HA und VA sind jeweils 2 Bits und für mit X gekennzeichneten nicht veränderten Stellen jeweils 1 Bit

$\Rightarrow$  Register nach Schritt 3 muss 32 Bit lang sein.

**10 Punkte**

c)

A = 10101111  
 Invertieren => 01010000  
 +00000001 => 01010001  
 Kommazahl => 0101,0001  
 =>  $x = 4 + 1 + 0,0625 = 5,0625$   
 => negative Zahl => -5,0625

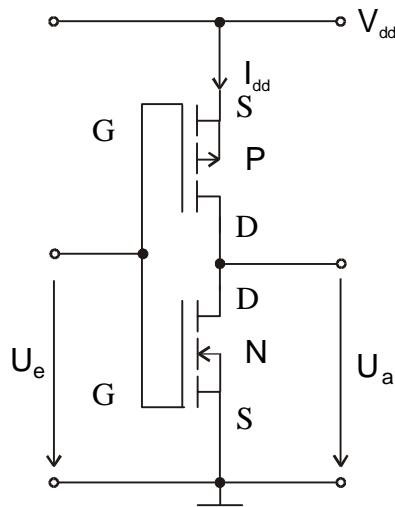
R1 = 10011010b  
Oberstes Bit = Vorzeichen => Dezimalzahl ist  $-1 + 0,125 + 0,0625 + 0,015625 = -0,796875$

[illegible]

### Aufgabe 3: MOS-Technologie

9 Punkte

a)



$U_e = H$ :

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.

b)

A: Source, B: Gate, C: Drain, D: Drain, E: Gate, F: Source

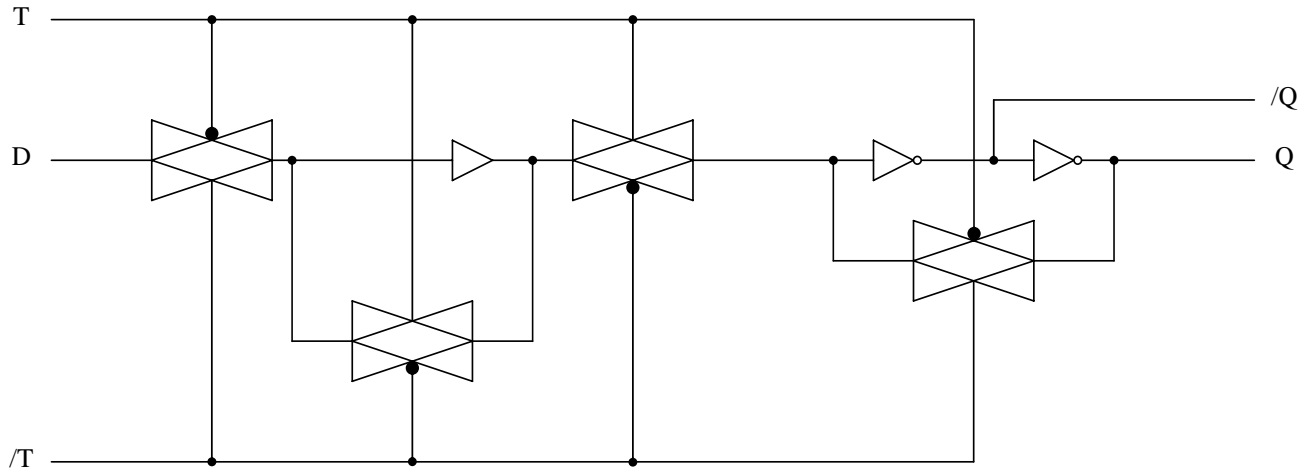
c)

G: n, H: p, I: n, J: p

#### Aufgabe 4: CMOS-Transferrgates

11 Punkte

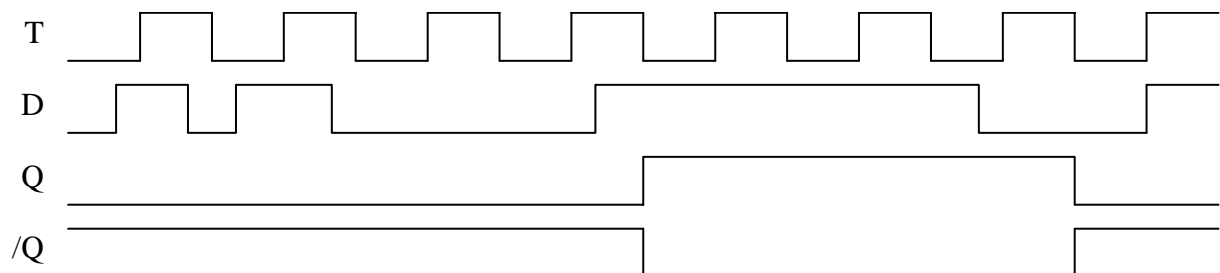
a)



b)

TG1: offen, TG2: gesperrt, TG3: gesperrt, TG4: offen

c)



**Aufgabe 5: Beschreibung einer FSM**

**10 Punkte**

a)

Befehl	OP4	OP3	OP2	OP1	Beschreibung
INC A	1	0	0	1	increment A
ANDL A, INR	1	1	0	1	A=(INR) AND (A)
ORL A, INR	1	1	1	0	A=(INR) OR (A)
NOP	1	1	1	1	no operation

b)

S3, S0

c)

S3, S0, S1

d)

	IN_LOAD	ADR_IN	ADR_LOAD	ALE	READ	WRITE	TR_IN	X_LOAD	ADD	OUT_LOAD
<b>Schritt 1</b>	0	0	0	0	0	0	1	0	0	1
<b>Schritt 2</b>	1	0	0	1	1	0	0	0	1	1

**Aufgabe 6: Assemblerprogrammierung mit dem 80C51**

**10 Punkte**

a)

```
Sub35:
    mov A,R0
    subb A,#35
    mov R0,A
ret
```

b)

```
Divider:
    mov A,R1
    rrc A
    mov R1,A
    mov A,R0
    rrc A
    mov R0,A
ret
```

c)

```
Send:
    jnb TI, Send
    clr TI
    mov SBUF,A
ret
```

## Aufgabe 7: 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:  $100_{16} - 65_{16} = 9B_{16} = 155_{10}$   
Eingangstakt von Timer 1:  $f_{osc}/12$   
Eingangsteiler der seriellen Schnittstelle: Timer 1-Überlaufrate dividiert durch  
16 (SMOD = 1) oder 32 (SMOD = 0)

$$\Rightarrow \text{Baudrate (SMOD=0)} = \frac{10 \text{ MHz}}{12} \cdot \frac{1}{32} \cdot \frac{1}{155} = 168,01 \text{ baud}$$

$$\text{Baudrate (SMOD=1)} = \frac{10 \text{ MHz}}{12} \cdot \frac{1}{16} \cdot \frac{1}{155} = 336,02 \text{ baud}$$

$$\text{oder: Baudrate (SMOD)} = \frac{10 \text{ MHz}}{12} \cdot \frac{1}{2^{5-\text{SMOD}}} \cdot \frac{1}{155} = \frac{5376}{2^{5-\text{SMOD}}} \text{ baud} = 168,01 \cdot 2^{\text{SMOD}} \text{ baud}$$

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 8: Digitale Signalprozessoren****9 Punkte**

a)

<b>Register</b>	<b>Inhalt nach Befehl 1</b>	<b>Inhalt nach Befehl 2</b>	<b>Inhalt nach Befehl 3</b>
R0	\$0003	\$0003	\$0003
X0	\$300000	\$200000	\$000005
R4	\$0000	\$0000	\$0003
Y0	\$F10000	\$000000	\$C40000

b)

<b>Befehl</b>	<b>Adressierungart X-Speicher</b>	<b>Adressierungart Y-Speicher</b>
<b>1</b>	Indirect mit Postinkrement	Indirekt mit Postinkrement durch Offset
<b>2</b>	Indirekt mit Offset	Direkt
<b>3</b>	Direkt mit Offset	Indirekt mit Prädecrement

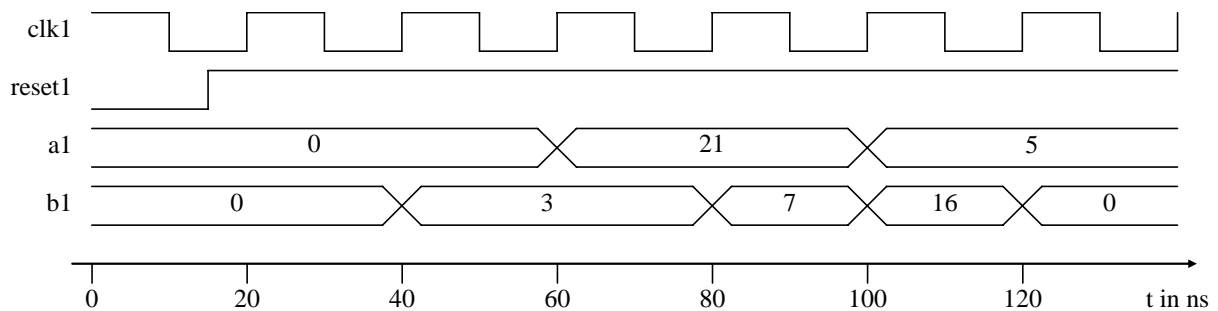
### Aufgabe 9: Logikbeschreibung mit VHDL

10 Punkte

a)

Testbench

b)



c)

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

entity examination is
port (clk    : in  std_logic;
      reset  : in  std_logic;
      a      : in  std_logic;
      b      : in  integer range 0 to 128;
      c      : out integer range 0 to 256);
end examination;

architecture examination_arch of examination is
begin

process(clk, reset)
begin
    if reset = '0' then
        c <= 0;
    else
        if clk'event and clk = '0' then
            c <= a+b;
        end if;
    end if;
end process;

end examination_arch;
```

## Aufgabe 10: Analyse von VHDL-Modellen

10 Punkte

a)

clk					
x	6	20	1	7	13
y1	6	20	1	7	13
y2	11	25	6	12	18
y3	9	23	4	10	16

b)

clk					
x	5	2	16	8	22
y1	U	5	2	16	8
y2	U	U	U	10	7
y3	U	U	8	5	19

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.