

Vorname: _____

Name: _____

Matr.-Nr.: _____

Note:

27.02.2007, 14.00 Uhr – 16.00 Uhr

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

Diplomprüfung im Fach

"Mikrorechnertechnik"

Musterlösung
F07

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)

ROM: Wird bei der Herstellung programmiert. Nicht löscherbar.

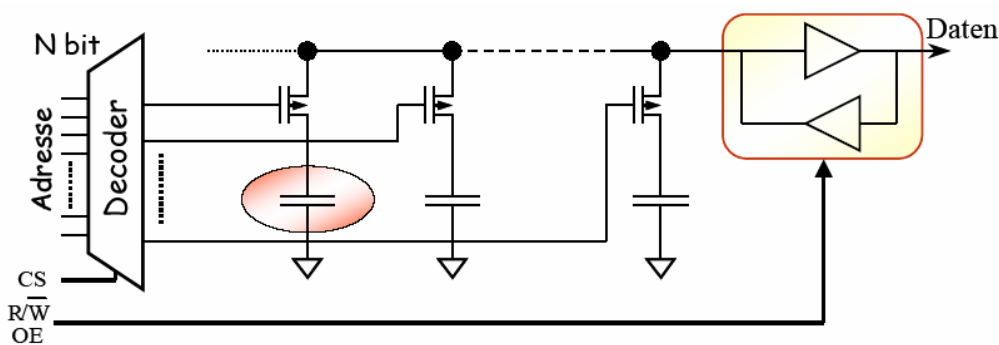
PROM: Vom Anwender einmalig programmierbarer Festwertspeicher. Nicht löscherbar.

EPROM: Vom Anwender mehrmalig beschreibbar. Löscherbar mittels UV-Licht.

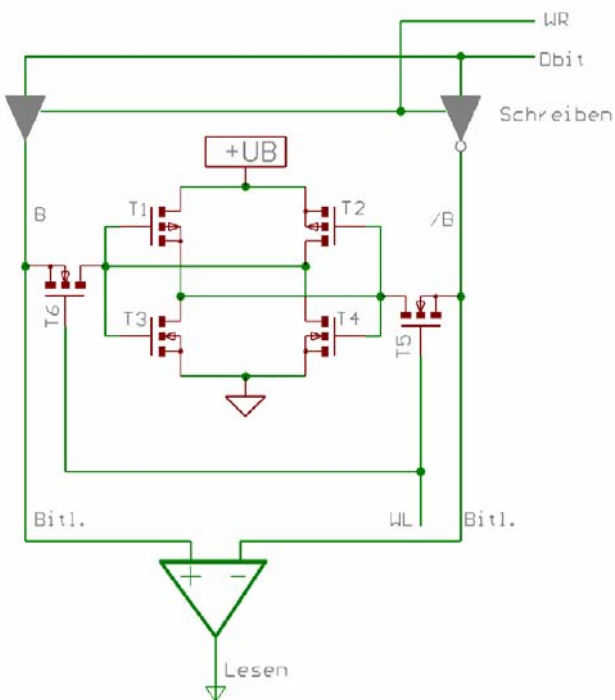
EEPROM: Vom Anwender mehrmalig beschreibbar. Elektrisch löscherbar.

b)

DRAM-Zelle: Speicherung mittels Kondensator. Speicherzelle entlädt sich mit der Zeit, daher Refresh der Daten nötig.



SRAM-Zelle: Speicherung mittels bistabiler Kippzelle. Kein Refresh nötig.



c)

	00	01	10	11
00	0000	0000	0000	0000
01	0000	0001	0010	0011
10	0000	0010	0100	0110
11	0000	0011	0110	1001

d)

Das Ergebnis der Multiplikation wird innerhalb eines Speichers abgelegt. Mittels zweier Adressen, welche die jeweiligen Faktoren repräsentieren, wird der entsprechende Speicherort des Ergebnisses ausgewählt. Die Bitbreite des Ergebnisses beträgt immer die doppelte Adressbreite.

Aufgabe 2: Zahlendarstellung in Mikrorechnerprogrammen

9 Punkte

a) $-1 \leq x < 0,875$

b) $-1 \leq x < 1$

c)

$$Z_{Fi} = \frac{Z_i}{2^{N-1}} = \frac{-z_{N-1} \cdot 2^{N-1} + \sum_{i=0}^{N-2} z_i \cdot 2^i}{2^{N-1}} = -z_{N-1} \cdot 2^0 + \sum_{i=0}^{N-2} z_i \cdot 2^{i-N+1}$$

d) $0,3984375 = 2^{-2} + 2^{-3} + 2^{-6} + 2^{-7} \triangleq 0011001100000000b$

e) $-7,625 = (-1)^1 \cdot 7,625$
 $= (-1)^1 \cdot [2^2 + 2^1 + 2^0 + \frac{1}{2} + \frac{1}{8}] = (-1)^1 \cdot [2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-3}]$
 $= (-1)^1 \cdot 2^2 \cdot [1 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-5}]$
 $= (-1)^1 \cdot [1 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-5}] \cdot 2^{129-127}$

=> VZ= negativ

Exponent: $129_d = 1000\ 0001_b$

Mantisse: m= 111 1100 0000 0000 0000 0000

=> Gleitkommazahl= 1000 0001 0111 0100 0000 0000 0000 0000

f)

Betrag der Zahl $7,625 = 0000\ 0000\ 0000\ 0111 \cdot 1010\ 0000\ 0000\ 0000$

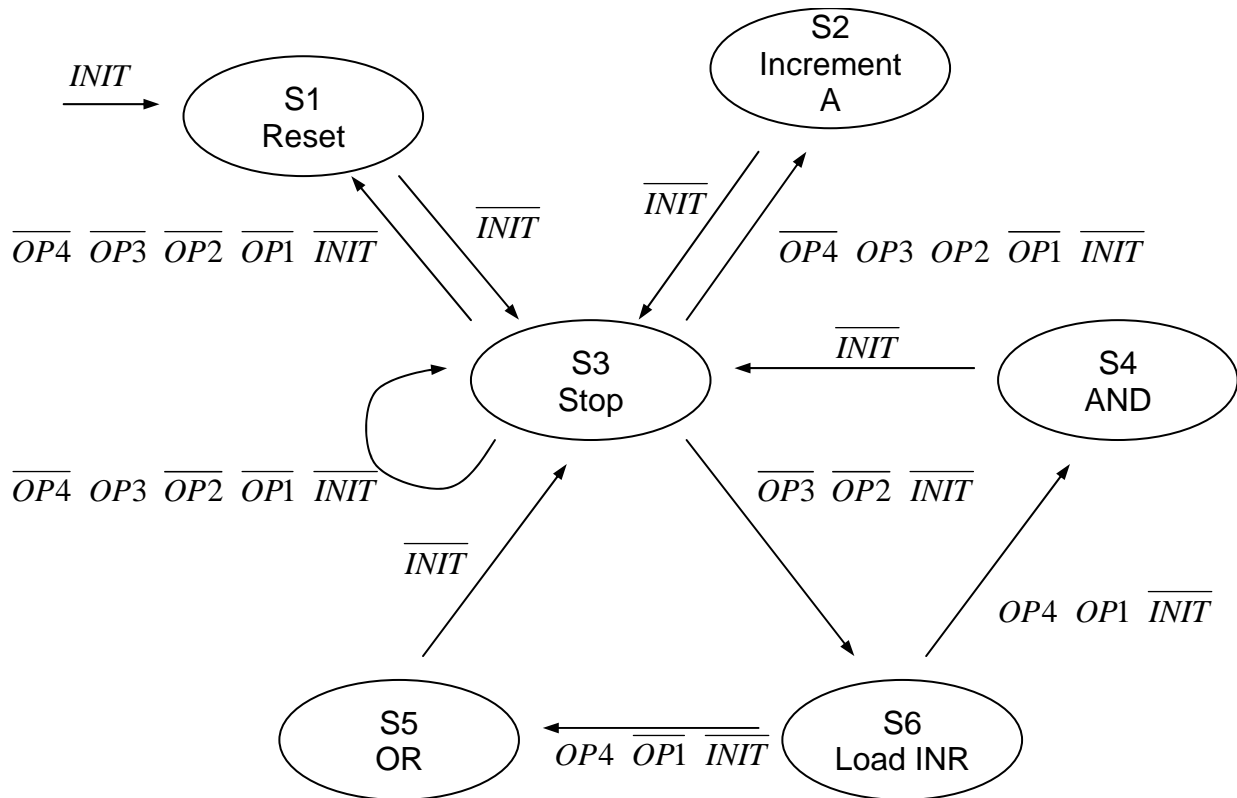
Davon das 2er-Komplement durch Negation und Addition von 1 an der niederwertigsten Stelle, also nicht an der ersten Vorkommastelle:

=> $-7,625 = 1111\ 1111\ 1111\ 1000 \cdot 0110\ 0000\ 0000\ 0000$

Aufgabe 3: FSM

11 Punkte

a)



b) S4, S0

c) S4, S0, S2

d)

1. Daten vom IN-Register in das OUT-Register schreiben.
2. Daten vom Eingang in das IN-Register schreiben, gleichzeitig neue Daten vom Speicher auslesen. Daten aus dem X-Register in das OUT-Register schreiben.

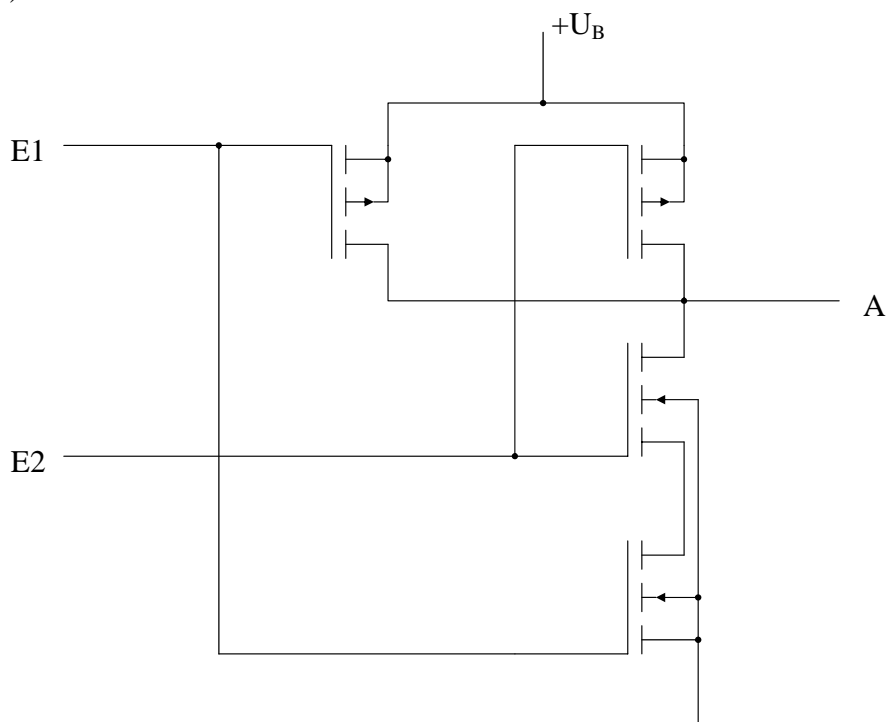
Aufgabe 4: CMOS-Transferegates

10 Punkte

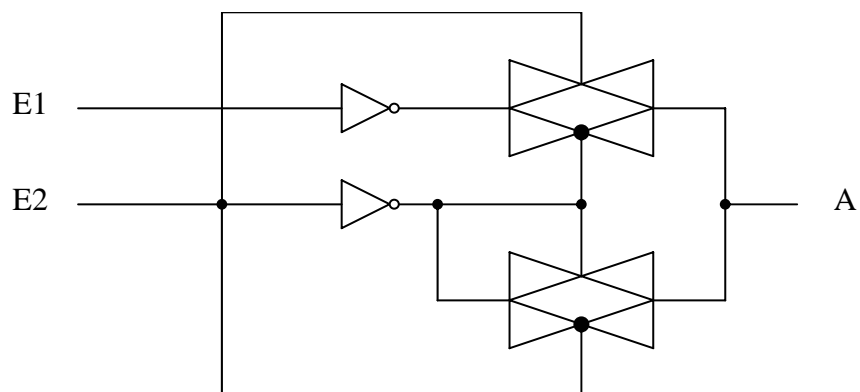
a)

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

b)



c)



d)

Funktion: Halbaddierer

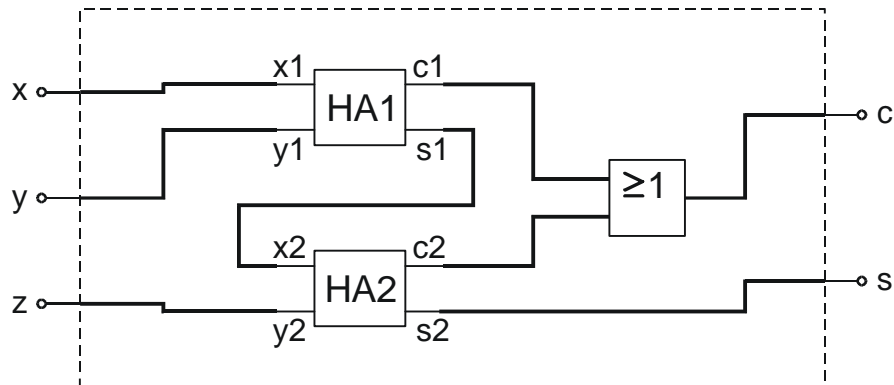
a	b	c	d
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Aufgabe 5: Arithmetik-Schaltungen

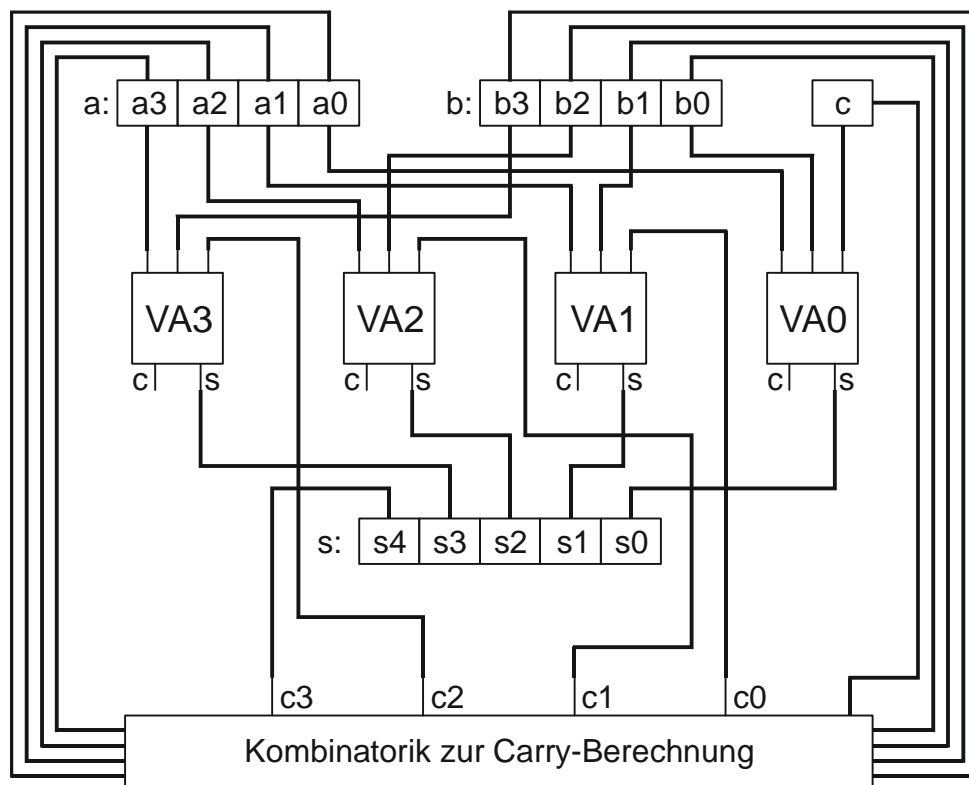
10 Punkte

a)

eine Möglichkeit:



b)



c)

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

$$\Rightarrow \text{Gesamtlaufzeit} = 4 \cdot \tau_{VA} + 1 \text{ ns} + 2 \text{ ns} = 4 \cdot 2 \text{ ns} + 1 \text{ ns} + 2 \text{ ns} = 11 \text{ ns}.$$

$$d) f_{\max} = 1 / (2 \text{ ns} + 1 \text{ ns} + 0,75 \text{ ns}) = 266,667 \text{ MHz}$$

e) Das 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 das Register muss 40 Bit lang sein.

Aufgabe 6: Serielle Schnittstelle des 80C51

11 Punkte

a)

```
CLR C           ; Carry löschen
MOV A, R0       ; A := R0
RLC A          ; A := 2 · A, Überlauf in Carry speichern
MOV 20H.1, C    ; ersten Überlauf in 20H.1 speichern
CLR C          ; Carry löschen
RLC A          ; A := 2 · A, Überlauf in Carry speichern
MOV 20H.0, C    ; zweiten Überlauf in 20H.0 speichern
INC A          ; A := A + 1
MOV R1, A       ; Ergebnis in R1 speichern
MOV R2, 20H     ; Überläufe in R2 speichern
```

b) Operation: $4 \cdot R0 + 1$

c) Ergebnis steht in R2,R1

d) Ergebnis: $4 \cdot C6H + 1 = 4 \cdot 198 + 1 = 793 = 0319H$

Register	R2	R1	R0
Inhalt (hexadezimal)	03	19	C6

Aufgabe 7: Mikrocontrollerprogrammierung

9 Punkte

a)

Bit 7				Bit 0				
SCON	0	1	X	1	X	X	0	0
		Mode 1	(für Mode 2, 3)	receive enable	(für Mode 2, 3)	Interrupt-Flags		

Bit 7				Bit 0				
TMOD	0/1/X	0	1	0	X	X	X	X
		Start durch TR1	Timer	8 bit Autoreload	(für Timer 0)			

b)

Autoreloadwert: EEH \rightarrow Anzahl Zählerschritte = $256 - 238 = 18$

Frequenz zur Baudratengenrierung: $18 \cdot 4800 = 86400$ kHz

Eingangsfrequenz der seriellen Schnittstelle: Timer 1-Überläufe dividiert durch 16

Eingangsfrequenz für Timer 1: $f_{osc}/12$

damit folgt für die Frequenz f_{osc} : $f = 86,4kHz \cdot 12 \cdot 16 = 16,5888MHz$

c) Exakte Baudrate: $\frac{16MHz}{12 \cdot 16 \cdot 17} = 4901,96$ baud

Fehler: $\frac{4901,96 - 4800}{4800} = 21,24 \cdot 10^{-3}$

d) Für die maximale Baudrate muss die Zahl der Zählerschritte minimal werden, d. h. Zählerüberlauf nach einem Schritt.

damit folgt für die zugehörige Baudrate: $\frac{12MHz}{12 \cdot 16} = 62,5$ kbaud

Aufgabe 8: Architektur von DSPs und Analyse von Assemblercode

9 Punkte

a)

```
MOVE    # $1000 , R1           ; Werte von externem ADW einlesen
MOVE    # $40 , R5             ; Pointer auf internen Y-Datenspeicher
```

b)

```
MOVE    X: ( R1 ) , X0         ; Startwert in X0
MOVE    Y: ( R5 ) + , Y0       ; Startwert in Y0
CLR     A                      ; Akku A löschen
DO      #5 , ENDE
MAC     X0 , Y0 , A      X: ( R1 ) , X0      Y: ( R5 ) + , Y0
ENDE
```

c)

A2	A1	A0
\$00	\$0A0000	\$000000

Aufgabe 9: Simulation von VHDL-Modellen

12 Punkte

a)

```
entity klausur is
port (clk, reset: in std_logic;
      result: out std_logic);
end klausur;
```

b)

```
process (clk)
begin
    if reset = '0' then
        counter <= 1;
    else
        if clk'event and clk = '1' then
            if counter < 20 then
                counter <= counter + 1;
            else
                counter <= 1;
            end if;
        end if;
    end if;
end process;
```

c)

```
process (counter)
begin
    if counter < 11 then
        result <= '0';
    else
        result <= '1';
    end if;
end process;
```

d) Takteiler

Aufgabe 10: Logikbeschreibung mit VHDL

9 Punkte

a)

clk						
x	3	4	2	1	6	
y1	30	40	20	10	60	
y2	15	20	10	5	30	
y3	15	20	10	5	30	

b)

clk						
x	3	4	2	1	6	
y1	X	X	35	30	15	
y2	X	15	20	10	5	
y3	15	20	10	5	30	

c) Signale, Variable, Konstanten.

d) Mit VHDL wird 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.