

ETI Aufgabenblatt

<https://github.com/LitschiW/ETIPAVorschlaege>

letzte Änderung: 26. Januar 2019

Aufgabenbereich 1: Festkommaarithmetik

a)

Füllen Sie die Tabelle aus:

Dezimalzahl	Vorzeichenbehaftete Binärdarstellung	B-Komplement Darstellung	Hexadecimal Darstellung
42			2A
-8			
	00110000		
	10001101		
		00011010	
		11111111	

(Hinweis: $A_{(16)} = 10_{(10)}$, $B_{(16)} = 11_{(10)}$, $C_{(16)} = 12_{(10)}$, $D_{(16)} = 13_{(10)}$, $E_{(16)} = 14_{(10)}$, $F_{(16)} = 15_{(10)}$)

b)

Konvertieren Sie $93,625_{(10)}$ jeweils in die Binär- und Hexadezimaldarstellung:

(Hinweis: Ihr Ergebnis sollte mehr als 8 Binärstellen enthalten. Das ist in diesem Fall gewollt, Sie müssen nicht kürzen/runden.)

c)

Konvertieren Sie $13_{(10)}$ in die Binärdarstellung und rechnen Sie $93_{(10)} - 13_{(10)}$ mittels binärer Subtraktion:

Aufgabenbereich 2: Fließkommaarithmetik

Für diesen Aufgabenbereich nutzen wir den IEEE 754 Standard für Minifloats.

D.h. wir benutzen eine 8 Bit Darstellung mit einem Vorzeichen-, 3 Mantissen- und 4 Exponentbits.

a)

Wie groß ist der Bias unserer Darstellung?

Was ist der Bias für eine Fließkommazahl
mit einem Exponent der Länge 6?

b)

Konvertieren Sie diese Sonderfälle in Fließkommadarstellung:

$\infty =$

$0 =$

$NaN =$

c)

Bestimmen sie die einzelnen Bestandteile der Fließkommazahl $11011100_{(2F)}$:

$V =$

$E =$

$M =$

d)

Konvertieren Sie $11011100_{(2F)}$ in eine Dezimalzahl:

e)

Addieren Sie $01011100_{(2F)}$ und $01001000_{(2F)}$ mittels Fließkommaarithmetik:

f)

Multiplizieren Sie $01011100_{(2F)}$ und $00111100_{(2F)}$ mittels Fließkommaarithmetik:

g)

Stellen sie 1_{10} in Fließkommaschreibweise da:

Zeigen Sie anhand eines Beispiels, dass man durch das kontinuierliche Addieren von 1_{10} auf eine beliebige Fließkommazahl $F (\neq \infty)$ niemals ∞ erreicht.

Aufgabenbereich 3: Logik und CMOS-Komplexgatter

In diesem Bereiche beschäftigen wir uns mit Logik und CMOS Komplexgattern. Es wird erwartet, dass Sie entsprechen Pull-up und Pull-down Netzwerke zeichnen.

a) Logische Funktionen

Füllen sie folgende Wahrheitstabellen aus:

(Hinweis: $(A \Rightarrow B) \equiv (\bar{A} \vee B)$)

A	B	$A \vee B$
0	0	
0	1	
1	0	
1	1	

A	B	$A \Rightarrow B$
0	0	
0	1	
1	0	
1	1	

A	B	$\overline{A \wedge B}$
0	0	
0	1	
1	0	
1	1	

A	B	$\overline{A \oplus B}$
0	0	
0	1	
1	0	
1	1	

b)

Geben sei folgende Funktionstabelle der Funktion $F(A, B, C) = Q$:

A	B	C	Q
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Welche Normalform der Funktion wäre kürzer?

1. kKNF ☐

2. kDNF ☐

Geben sie die Funktion in der gewählten Normalform an:

$F(A, B, C) =$

c) Allgemeine Fragen zum Thema CMOS:

Wie viele Transistoren benötigt ein OR Komplexgatter?

Wie viele Transistoren benötigt ein NAND Komplexgatter?

Was ist der Unterschied zwischen n-Mos- und p-Mos-Transistoren?

d)

Vereinfachen Sie die Formel $\overline{(C \vee (\overline{C} \wedge A) \vee (\overline{\overline{A} \vee B)})} \wedge \overline{C}$ möglichst stark:

Zeichnen Sie ein (strukturequivalentes) CMOS-Komplexgatter das ihrem Ergebnis entspricht:

e)

Zeichnen Sie folgende Funktion strukturgleich als CMOS-Komplexgatter: $f(x) = \overline{(A \vee B)}$.



Wie viele Transistoren würden Sie benötigen?

Aufgabenbereich 4: Flipflops und Schaltungen

a) Volladdierer

Zeichnen sie einen Halbaddierer auf Gatterebene. Setzen Sie dann 2 Halbaddierer (gekennzeichnet als HA) zu einem Volladdierer zusammen:

Füllen Sie die Funktionstabelle für ein Volladdierer aus:

A	B	C_{in}	S	C_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

b) D-FlipFlop

Was ist der Unterschied zwischen einem D-Latch und einem D-Flip-Flop?

Füllen Sie die Funktionstabelle für ein D-Flip-Flop aus:

C	D_t	D_{t+1}

(Hinweis: \neg bezeichnet eine sinkende, \neg eine steigende Flanke)


Zeichnen Sie ein taktgesteuertes D-Latch auf Gatter Ebene. Makieren sie das enthaltene RS-Flipflop:

Zeichnen Sie ein taktgesteuertes D-Flip-Flop. Nutzen Sie D-Latches als vorhandene Bauteile:

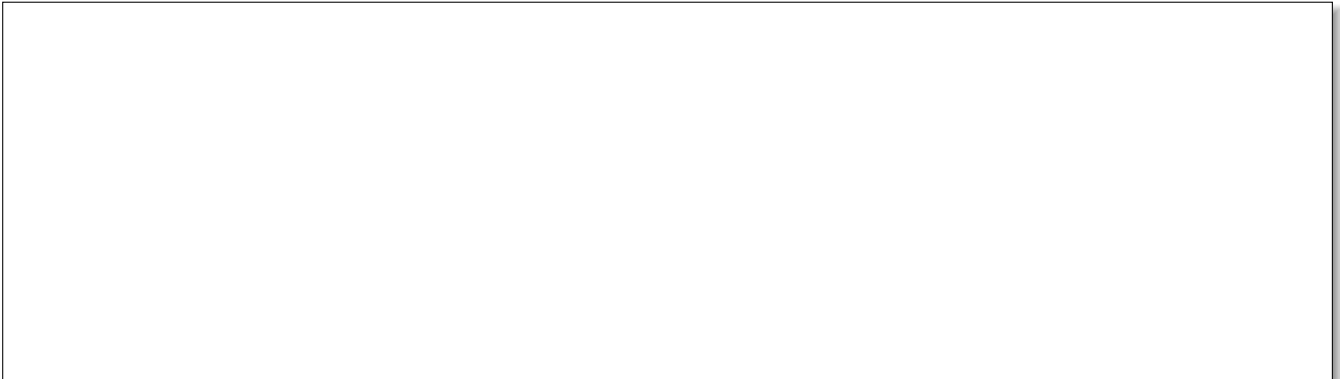


c) Schieberegister

Zeichnen Sie ein 3-Bit-Links-Schieberegister:



Wofür können Schieberegister eingesetzt werden?



Wo kann man Schieberegister auf einem Mikrocontroller finden?

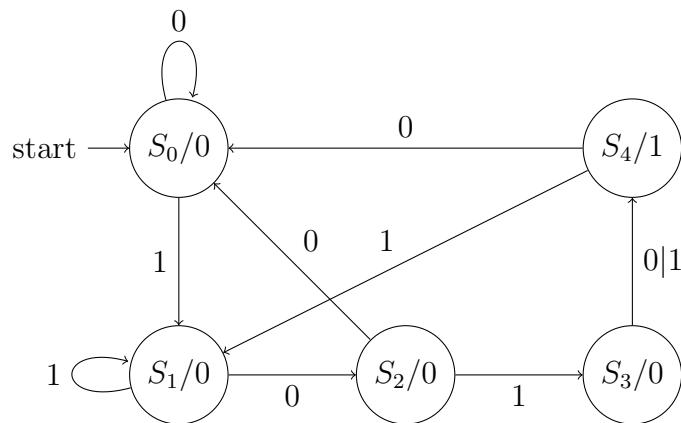
Aufgabenbereich 5: Finite State Machines

a)

Was ist der unterschied zwischen Moore und Mealy Automaten?

b)

Welche Zeichenfolge(n) erkennt folgender Automat:



(Hinweis: bei einem Zustand S_i/X bezeichnet S_i den Zustand und X die Ausgabe in diesem Zustand.)

-
-
-

c)

Entwerfen sie einen synchronen Modulo 4 Zähler. Der Zähler soll bidirektional zählen können, dafür betrachten wir den Eingang dir . Für $dir = 0$ soll vorwärts, bei $dir = 1$ rückwärts gezählt werden.

Zeichnen Sie eine Zustandsdiagramm für diesen Automaten:

(Hinweis: Den Clock Eingang müssen sie zunächst nicht beachten.)

Geben Sie die Zustandsübergangstabelle an:

Takt t			Takt $t + 1$	
dir	s_1	s_0	s_1	s_0
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Geben sie die Zustandsübergangslogik für den Zustand $s = s_1s_0$ an:

$s_1 =$

$s_0 =$

Nun fügen wir dem Automaten 4 1-Bit Ausgänge mit dem Namen ZERO, ONE, TWO, THREE hinzu. Sie sollen entsprechend ihrer Namen den Wert 1 annehmen, wenn der Automat den dazugehörigen Zustand erreicht. (z.B. für $s = 00$ ist ZERO=1, der Rest=0)

ZERO =

ONE =

TWO =

THREE =

Zeichnen sie nun den Automaten auf Gatterebene. Ihnen stehen D-Flip-Flops, Normale-Gatter (AND, OR, NAND, NOR, XOR, NOT) mit beliebig vielen Eingängen zur Verfügung. Beachten Sie, dass sie nun auch den Clock Eingang betrachten müssen:

Aufgabenbereich 6: VHDL

a)

Was ist der Unterschied zwischen einer `architecture` und einem `process` in VHDL?

Was ist der Unterschied zwischen einem `signal` und einem `Port` in VHDL und wo werden diese definiert?

Wofür nutzt man `component`'s in VHDL

Was macht der Befehl `rising_edge(clk)` in einem `process`?

b)

Gegeben sei folgender VHDL Code:

```
process(clk,reset) is
begin
    if clk'event and clk = '1' then
        %write your code here:

    endif;
    if reset'event and reset='0' then
        q_o <= (others => '0');
    endif;
end process;
```

Sie haben weiterhin einen Eingang d_i zur Verfügung, q_o soll einen Ausgang darstellen. Implementieren sie die Funktion eines D-Flip-Flops!

Was für einen Resettyp hat dieses Flip-Flop?

1. synchron aktiv High ☐

2. synchron aktiv Low ☐

3. asynchron aktiv High ☐

4. asynchron aktiv Low ☐