

Vorlesungsprüfung aus Digitales Design

20. Januar 2012

Die Arbeitszeit beträgt 2 Stunden. Als Hilfsmittel sind ausnahmslos Schreibzeug, Lineal und (nicht programmierbarer) Taschenrechner erlaubt. Schreiben Sie Ihre Antworten und Lösungen (incl. Lösungsweg!) mit Füllfeder oder Kugelschreiber (nicht rot, KEIN Bleistift!) und streichen Sie alles durch, was nicht zur Beurteilung herangezogen werden soll. Ein Abbruch der Prüfung nach Erhalt der Angaben führt in jedem Fall zu einer Beurteilung.

Tragen Sie Namen, Kennzahl und Matrikelnummer **zu Beginn** der Prüfung in die Tabelle ein und beschriften Sie jedes Blatt, das Sie abgeben möchten rechts oben mit Namen und Matrikelnummer.

Familienname:	Vorname:
Kennzahl:	Matrikelnummer:

Viel Erfolg!

Die nachfolgende Tabelle nicht beschriften!

Beispiel		Mögliche Punkte	Erhaltene Punkte
Theoriefragen		50	
Rechenbeispiel	1	10	
	2	10	
	3	15	
	4	15	
Gesamt		100	

Theoriefragen

Frage 1:

Wozu benötigt man ein Floating Gate?

Frage 2:

Was ist ein Transmission Gate und wie ist es aufgebaut?

Frage 3:

Welchen Unterschied macht es, ob Sie den Ausgang eines Gatters mit einem oder mit 10 Eingängen nachfolgender Gatter belasten?

Frage 4:

Was versteht man unter Electromigration?

Frage 5:

Was versteht man unter einem non-volatile Memory?

Frage 6:

Erläutern Sie das Prinzip des Boundary-Scan!

Frage 7:

Was versteht man im Zusammenhang mit einem PLD unter einem Product Term?

Frage 8:

Wozu benötigt man bei der Fertigung eines ICs die Masken?

Frage 8:

Altert ein Chip auch wenn er nicht in Betrieb ist? Begründen Sie!

Frage 10:

Warum werden bei CMOS-Logik die Funktionen NAND und NOR bevorzugt verwendet und nur selten AND und OR?

Rechenbeispiele

Beispiel 1

Zu entwerfen ist eine Schaltung, die die Zahlen 0 bis 7 (codiert als positive Integer) an einer Kette von 7 Leuchtdioden (LED_1 ... LED_7) so ausgibt, dass bei „0“ keine LED leuchtet, bei „1“ leuchtet LED_1 bei „2“ leuchten LED_1 und LED_2 und bei „7“ leuchten alle LEDs (siehe Abbildung 1.1). Nehmen Sie an, eine LED leuchtet, wenn sie mit ‚1‘ angesteuert wird.

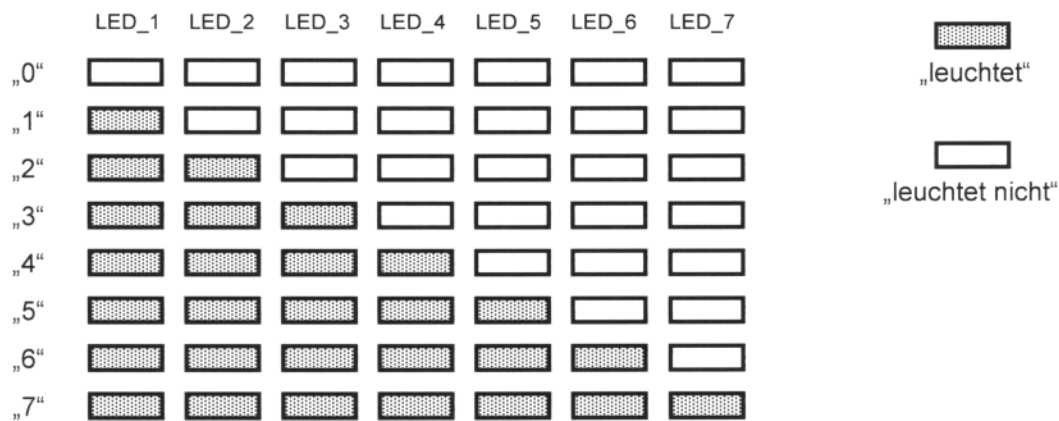


Abbildung 1.1

- (a) Geben Sie die Wahrheitstabelle für alle LEDs an!
- (b) Geben Sie die logische Verknüpfung an, die Sie für LED_4 realisieren müssen!
- (c) Zeichnen Sie das KV-Diagramm für LED_5 und geben Sie die benötigte Funktion in konjunktiver Normalform an!
- (d) Zeichnen Sie das KV-Diagramm für LED_3 und geben Sie die benötigte Funktion in disjunktiver Normalform an!
- (e) Geben Sie ein Schaltbild an, wie Sie die Funktion für LED_1 mittels Multiplexern realisieren können (ähnlich den ACTEL-Bausteinen)

Beispiel 2

In Abbildung 2.1 ist ein analoger Spannungsverlauf U_{in} gegeben.

- (a) Dieses Signal U_{in} wird an einen Eingang eines digitalen Gatters mit Schwellwert $U_{th} = 0,9V$ (ohne Hysterese) gelegt. Ergänzen Sie in Abbildung 2.1 den Spannungsverlauf U_{dig1} nach der Digitalisierung!
- (b) Zwecks besserer Störunterdrückung wird für die Digitalisierung von U_{in} nun ein Schmitt Trigger-Eingang verwendet. U_{dig2} zeigt den resultierenden Verlauf der Ausgangsspannung. Welche Schwellwerte hat dieser Schmitt Trigger? Wie groß ist seine Hysterese (inkl. Einheit)?
- (c) Zeichnen Sie die Kennlinie (Ausgangsspannung über Eingangsspannung) dieses Schmitt Triggers! Beschriften Sie die charakteristischen Spannungswerte und kennzeichnen Sie mit Pfeilen, in welcher Richtung die Kennlinie durchlaufen wird.
- (d) Handelt es sich um einen invertierenden Schmitt-Trigger oder einen nicht-invertierenden? Begründen Sie!

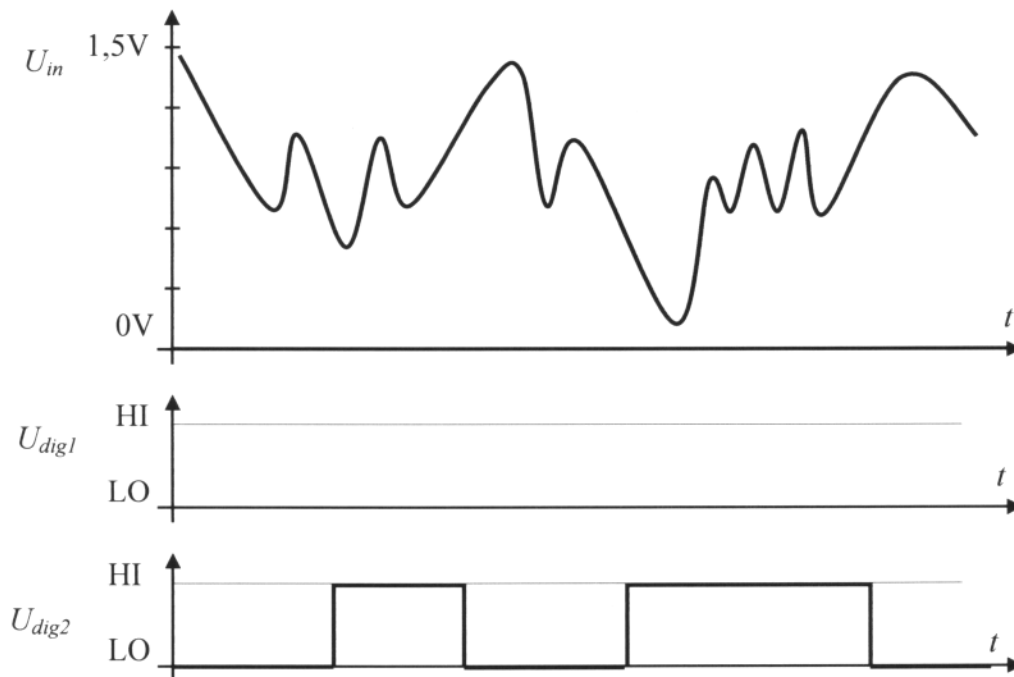


Abbildung 2.1

Beispiel 3

Gegeben ist die in Abbildung 3.1 dargestellte Schaltung mit 200MHz Takt. Ein asynchrones Eingangssignal mit 20MHz liegt am Eingang von Flip-Flop *FF1*. Der Ausgang von *FF1* geht über die kombinatorische Logik *COMB* an den Eingang von Flip-Flop *FF2*. Die Verzögerungszeit der kombinatorischen Logik beträgt $t_{PD} = 1\text{ns}$. Laut Datenblatt haben die beiden Flip-Flops folgende Parameter:

$$t_{SU} = 0,5\text{ns} ; \quad \tau_C = 0,1\text{ns} ; \quad T_0 = 0,1\text{ns} .$$

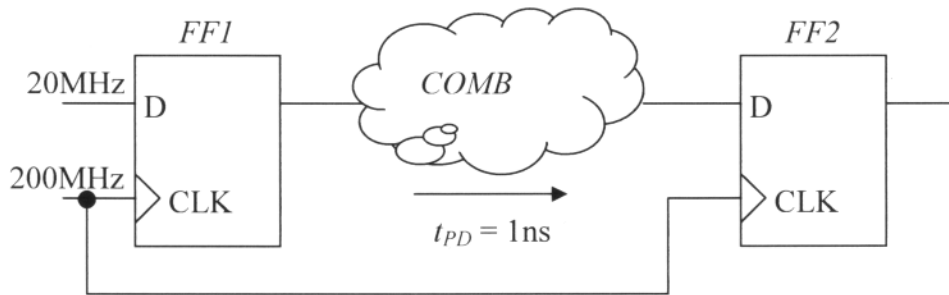


Abbildung 3.1

- Welche MTBU ist bei diesen Nominalwerten zu erwarten?
- Die Schaltung ist Teil eines Chips, der bei 1,2V Versorgungsspannung 5A Strom verbraucht. Welche Leistung verbraucht dieser Chip? Welche Temperatur erreicht der Die bei einer Umgebungstemperatur von 20°C und einem Gehäuse mit $\Theta_{JA}=10\text{K/W}$?
- Ermitteln Sie anhand der Derating Factors nach Tabelle 3.1 die Werte für t_{PD} und t_{SU} bei der nach (b) errechneten Temperatur! Welche Resolution Time t_r erhalten Sie damit? Ist die MTBU damit besser als bei (a)? Begründen Sie!

	1.10 V	1.15 V	1.20 V	1.25 V	1.30 V
-40°	0.77	0.73	0.68	0.64	0.61
0°	1.00	0.93	0.87	0.82	0.78
25°	1.14	1.07	1.00	0.94	0.90
80°	1.50	1.40	1.33	1.26	1.20
100°	1.60	1.49	1.41	1.34	1.28
125°	1.76	1.65	1.56	1.47	1.41

Tabelle 3.1

- Wie verändert sich die Situation, wenn Sie als Versorgungsspannung 1.3V anlegen (nehmen Sie an, dass die dynamischen Verluste dominieren)? Welche Temperatur erreicht der Chip nun? Welchen Derating Factor müssen Sie anwenden? Welche Resolution Time t_r erhalten Sie nun? Ist die MTBU damit besser als bei (c)? Begründen Sie!

Beispiel 4

Gegeben ist folgender VHDL-Code einer State-Machine:

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

```
entity RAM_acc is
  port
  (
    clk, res_n : in  std_logic;
    data       : in  std_logic_vector(7 downto 0);
    addr       : out std_logic_vector(7 downto 0);
    c_sel      : out std_logic;
    o_ena      : out std_logic
  );
end entity RAM_acc;
```

```
architecture good of RAM_acc is
  type STATE_TYPE is (zero, one, two, three);
  signal state, state_next : STATE_TYPE;
  signal addr_int : std_logic_vector(7 downto 0);
  signal addr_next : std_logic_vector(7 downto 0);
  signal rd_val : std_logic_vector(7 downto 0);
begin
  addr <= addr_int;

  sync : process(clk)
  begin
    if clk'event and clk = '1' then
      if res_n = '0' then
        state <= zero;
        addr_int <= (others => '0');
        rd_val <= (others => '1');
      else
        state <= state_next;
        addr_int <= addr_next;
        rd_val <= data;
      end if;
    end if;
  end process sync;

  next_state : process(state)
  begin
    case state is
      when zero =>
        state_next <= one;
      when one =>
        state_next <= two;
      when two =>
        state_next <= three;
      when three =>
        state_next <= zero;
    end case;
  end process next_state;

  output : process(state, addr_int, rd_val)
  begin
    addr_next <= addr_int;
    c_sel <= '1';
    o_ena <= '1';

    case state is
      when zero =>
        addr_next <= rd_val;
      when one =>
        c_sel <= '0';
      when two =>
        c_sel <= '0';
      when others =>
        o_ena <= '0';
    end case;
  end process output;
end architecture good;
```

- (a) Handelt es sich um einen synchronen oder einen asynchronen Reset? Begründen Sie!
- (b) Tragen Sie für die ersten 6 Taktzyklen nach dem Reset den Ablauf der Zustände (State plus zugehöriger Next-State) in Abbildung 4.1 ein! Ergänzen Sie weiters den Zeitverlauf aller Ausgangssignale (treffen Sie für die Eingänge ggf. sinnvolle Annahmen).

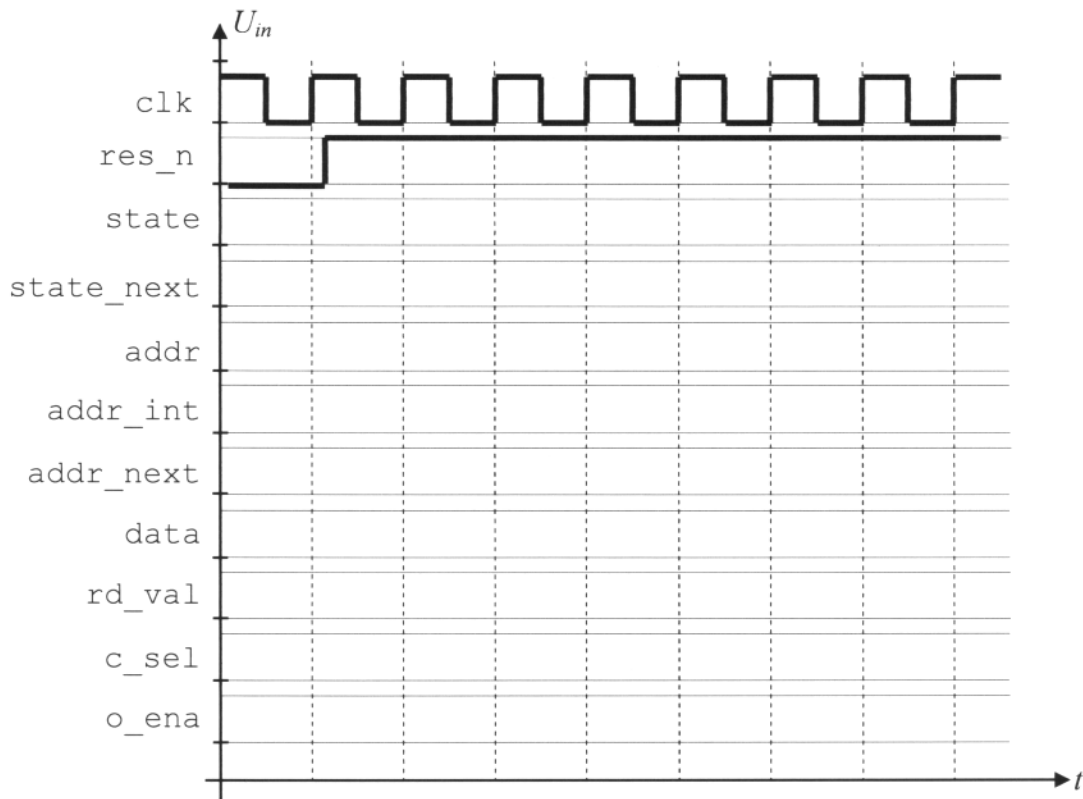


Abbildung 4.1

Gegeben ist weiters ein SRAM mit Read-Timing nach folgendem Datenblatt (Abbildung 4.2)

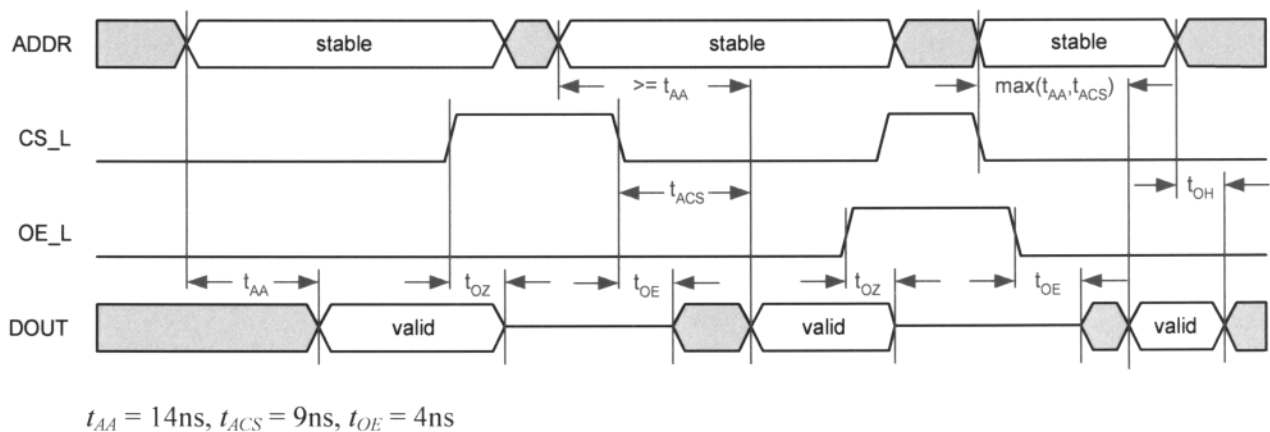


Abbildung 4.2

Angenommen die oben beschriebene State Machine wird mit 200MHz betrieben und wie folgt mit dem SRAM verschaltet: CS_L wird verbunden mit c_sel, OE_L mit o_ena, ADDR mit addr und DOUT mit data.

- (c) Kann für diesen Betrieb sichergestellt werden, dass die Daten des SRAM korrekt von der State Machine übernommen werden (Timing innerhalb der State Machine sei ideal, d.h. keine Delays und keine Setup-Time)? Welches Signal ist problematisch und warum?
- (d) Wie genau müsste man den Code der State Machine ändern, um ein korrektes (bzw. optimiertes) Timing zu erzielen?

Vorlesungsprüfung aus Digitales Design

21. Januar 2010

Die Arbeitszeit beträgt 2 Stunden. Als Hilfsmittel sind ausnahmslos Schreibzeug, Lineal und (nicht programmierbarer) Taschenrechner erlaubt. Schreiben Sie Ihre Antworten und Lösungen (incl. Lösungsweg!) mit dokumentechnen Stiften (Füllfeder, Kugelschreiber, KEIN Bleistift!) und streichen Sie alles durch, was nicht zur Beurteilung herangezogen werden soll. Ein Abbruch der Prüfung nach Erhalt der Angaben führt in jedem Fall zu einer Beurteilung.

Tragen Sie Namen, Kennzahl und Matrikelnummer **zu Beginn** der Prüfung in die Tabelle ein und beschriften Sie jedes Blatt, das Sie abgeben möchten rechts oben mit Namen und Matrikelnummer.

Familienname:	Vorname:
Kennzahl:	Matrikelnummer:

Viel Erfolg!

Die nachfolgende Tabelle nicht beschriften!

Beispiel		Mögliche Punkte	Erhaltene Punkte
Theoriefragen		50	
Rechenbeispiel	1	15	
	2	10	
	3	15	
	4	10	
Gesamt		100	

Theoriefragen

Frage 1:

Wozu benötigt man ein Floating Gate?

Frage 2:

Was ist ein Transmission Gate und wie ist es aufgebaut?

Frage 3:

Welchen Unterschied macht es, ob Sie den Ausgang eines Gatters mit einem oder mit 10 Eingängen nachfolgender Gatter belasten?

Frage 4:

Was versteht man unter Electromigration?

Frage 5:

Erklären Sie den Unterschied zwischen Boundary Scan und Scan Test!

Frage 6:

Warum kann man beim FPGA das Taktsignal nur bei eigens dafür vorgesehenen Pins anlegen?

Frage 7:

Welchen Zweck hat ein Process in VHDL?

Frage 8:

Was versteht man unter Metastabilität?

Frage 9:

Welche Probleme treten beim „exhaustive Test“ in der praktischen Anwendung auf?

Frage 10:

Was versteht man unter Dotierung? Wofür wird sie angewendet?

Rechenbeispiele

Beispiel 1

Abbildung 2.1 zeigt die FET-Realisierung einer logischen Funktion $y(a,b,c)$.

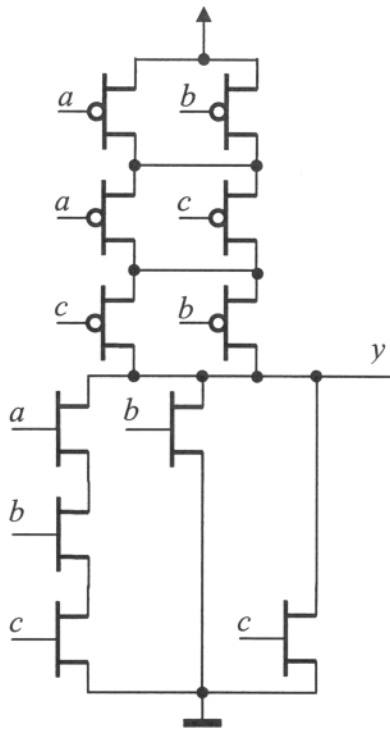
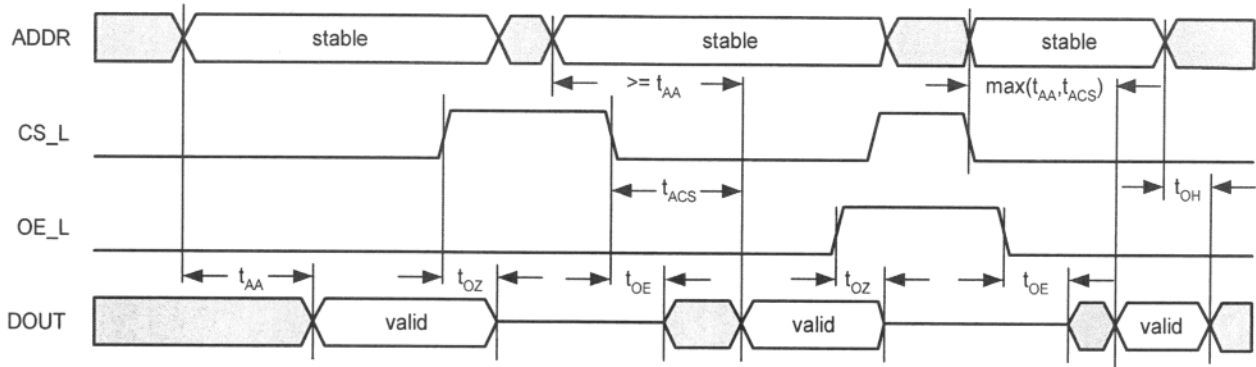


Abbildung 2.1

- (a) Leiten Sie anhand dieses Schaltbildes her, welchen Wert der Ausgang y für folgendes Bitmuster am Eingang annimmt: $(a,b,c) = (1,0,0)$! Kennzeichnen Sie alle leitenden Transistoren!
- (b) Was passiert beim Anlegen des Musters $(a,b,c) = (0,1,0)$? Begründen Sie!
- (c) Modifizieren Sie den n-Stack so, dass sich (in Verbindung mit dem vorhandenen p-Stack) eine korrekt funktionierende logische Schaltung ergibt! Geben Sie deren Funktion y in Form einer Booleschen Gleichung an!
- (d) Zeichnen Sie das KV-Diagramm dieser Funktion y !
- (e) Bezeichnen Sie die Funktion korrekt als AOIxxxx bzw. OAIxxxx !

Beispiel 2

Gegeben ist ein SRAM mit Read-Timing nach folgendem Datenblatt (Abbildung 3.1)



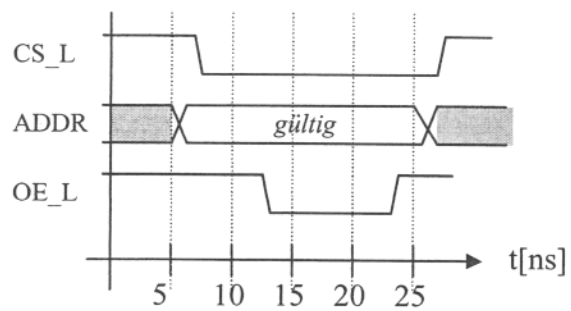
$$t_{AA} = 15\text{ns}, t_{ACS} = 10\text{ns}, t_{OE} = 5\text{ns}$$

Abbildung 3.1

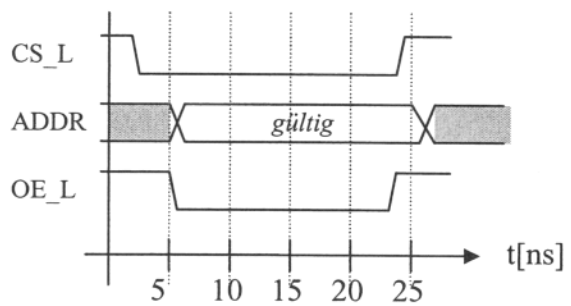
Gegeben sind weiters die Ausgangsmuster von drei State-Machines, die einen Zugriff auf dieses SRAM implementieren sollen (Abbildung 3.2).

- Geben Sie zu jeder Implementierung an, ob sie geeignet ist, einen gültigen Zugriff durchzuführen. Begründen Sie jeweils Ihre Antwort!
- Tragen Sie bei den korrekten Implementierungen den Zeitpunkt ein, ab dem auf die Daten zugegriffen werden kann!
- Machen Sie bei den mangelhaften Implementierungen einen Korrekturvorschlag: Welche Flanke müsste wie verschoben werden, damit der Zugriff gültig wird?

(1)



(2)



(3)

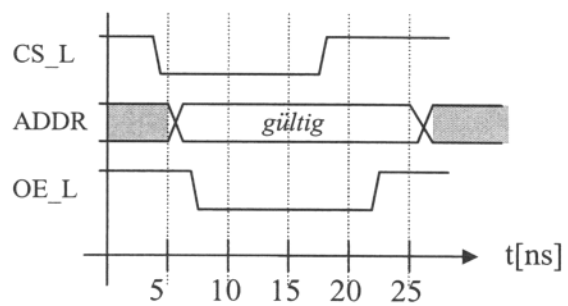


Abbildung 3.2

Beispiel 3

Gegeben ist die Schaltung in Abbildung 1. Arbeiten Sie mit dem single-Stuck-at Fehlermodell.

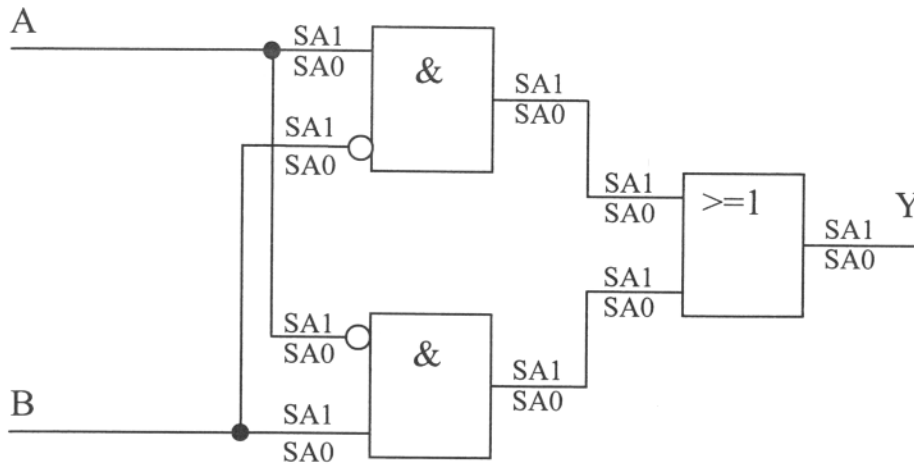


Abbildung 1

- Geben Sie Wahrheitstabelle und KV-Diagramm für diese Schaltung an!
Welche Grundfunktion erfüllt die Schaltung ?
- Wie viele Testvektoren benötigen Sie für einen Exhaustive Test dieser Schaltung?
Geben Sie die Vektoren an!
- In Abbildung sind alle möglichen Stuck-at Fehler eingetragen (SA0, SA1). Streichen Sie jene Fehler durch ein Kreuz durch („X“), die Sie aufgrund von Fehleräquivalenz nicht weiter zu berücksichtigen brauchen!
- Streichen Sie weiters jene Fehler durch einen einfachen Schrägstrich durch („/“), die Sie aufgrund von Fehlerdominanz nicht weiter zu berücksichtigen brauchen!
- Ermitteln Sie alle Testvektoren, die geeignet sind, den SA1-Fehler am Ausgang des oberen UND-Gatters zu erkennen! Beschreiben Sie die für die Ermittlung nötigen Schritte!
Ermitteln Sie ebenso alle Vektoren für SA1 am Ausgang des unteren UND-Gatters sowie für SA0 am Ausgang des OR-Gatters!
- Geben Sie einen minimalen Satz von Testvektoren an, der geeignet ist, alle in Abbildung 1 eingetragenen Fehler zu erkennen!

Beispiel 4

Gegeben sind die in Abbildung 4.1 dargestellten Signalverläufe für ein Datensignal D und ein Clock-Signal bzw. Enable-Signal C .

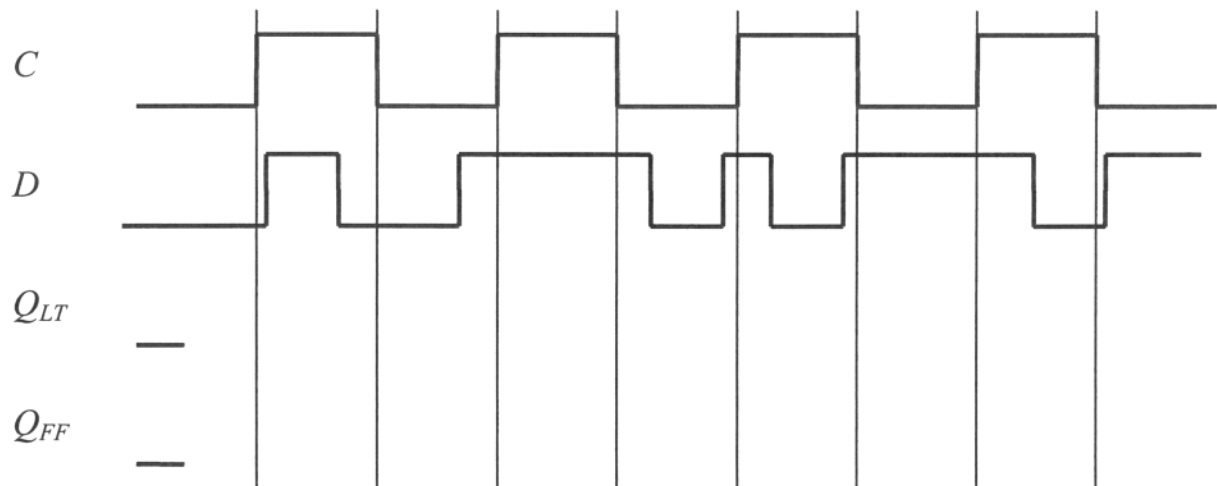


Abbildung 4.1

- (a) Tragen Sie in die Zeile Q_{LT} das zugehörige Ausgangssignal Q eines D-Latch ein! („0“ = „Hold“)
- (b) Tragen Sie in die Zeile Q_{FF} das zugehörige Ausgangssignal Q eines D-Flip-Flop ein! (steigende Flanke ist aktiv)
- (c) Beschreibt der folgende VHDL-Code ein D-Flip-Flop, ein D-Latch oder keines von beiden? Begründen Sie!

```
Architecture BEISPIEL_4 of PRUEFUNG is
begin
  signal CLK, DATA_IN, NEXT, DATA_OUT : std_logic;

  CAPTURE: process (CLK)
  begin
    if CLK'event and CLK = '1' then
      DATA_OUT <= NEXT;
    else
      NEXT <= DATA_OUT;
    end if;
  end process CAPTURE;

  NEXT <= DATA_IN;

end BEISPIEL_4;
```

- (d) Wenn Sie den „else“-Zweig (mit Pfeilen markiert) weglassen – was beschreibt der Code dann?

Vorlesungsprüfung aus Digitales Design

23. April 2008

Die Arbeitszeit beträgt 2 Stunden. Als Hilfsmittel sind ausnahmslos Schreibzeug, Lineal und (nicht programmierbarer) Taschenrechner erlaubt. Schreiben Sie Ihre Antworten und Lösungen (incl. Lösungsweg!) mit Füllfeder oder Kugelschreiber (nicht rot, KEIN Bleistift!) und streichen Sie alles durch, was nicht zur Beurteilung herangezogen werden soll. Ein Abbruch der Prüfung nach Erhalt der Angaben führt in jedem Fall zu einer Beurteilung.

Tragen Sie Namen, Kennzahl und Matrikelnummer **zu Beginn** der Prüfung in die Tabelle ein und beschriften Sie jedes Blatt, das Sie abgeben möchten rechts oben mit Namen und Matrikelnummer.

Familienname:	Vorname:
Kennzahl:	Matrikelnummer:

Viel Erfolg!

Die nachfolgende Tabelle nicht beschriften!

Beispiel		Mögliche Punkte	Erhaltene Punkte
Theoriefragen		50	
Rechenbeispiel	1	15	
	2	15	
	3	10	
	4	10	
Gesamt		100	

Theoriefragen

Frage 1:

Was besagt das Theorem von De Morgan?

Frage 2:

Kann es sinnvoll sein, ein Design aus Kostengründen auf 2 Chips aufzuteilen? Begründen Sie!

Frage 3:

Beschreiben Sie die logische Funktion eines AOI311! Skizzieren Sie seinen Aufbau aus Transistoren!

Frage 4:

Geben Sie die Signal-Resolution-Table für die Funktion XOR und die Zustände $\{0,1,X\}$ an!

Frage 5:

Wozu benötigt man ein Floating Gate?

Frage 6:

Wozu verwendet man eine Antifuse?

Frage 7:

Was versteht man unter Bonding?

Frage 8:

Erläutern Sie, wie physikalische Defekte zu dynamischen Fehlern führen können!

Frage 9:

Warum verursacht ein Scan-Test einen Overhead in der Chipfläche?

Frage 10:

Welche 3 Teile beschreiben eine Design-Unit in VHDL?

Rechenbeispiele

Beispiel 1

Für eine 7-Segment-Anzeige soll eine Ansteuerung realisiert werden: Der Eingang $Z = (z_3, z_2, z_1, z_0)$ ist ein binär codierter positiver Integer-Wert. Dieser Wert soll durch geeignete Ansteuerung der Leuchtsegmente a, b, c, d, e, f, g als Ziffer auf einem Display dargestellt werden. Die Lage der einzelnen Segmente ist in Abbildung 1.1 (rechts) gegeben. Ein Segment leuchtet bei Ansteuerung mit „1“.

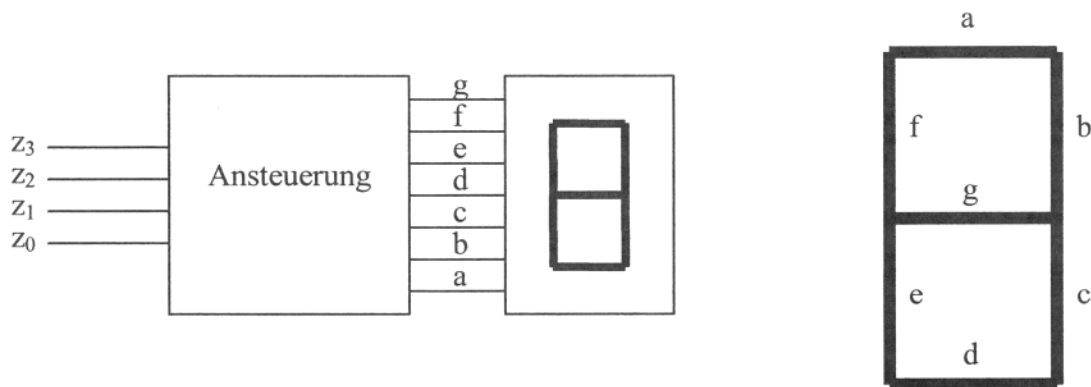


Abbildung 1.1

- Geben Sie die Wahrheitstabelle für die Leuchtsegmente b und c an! (Für Eingänge, die einem Integer größer als 9 entsprechen, kann der Ausgang beliebig sein – tragen Sie daher ein „X“ in die Wahrheitstabelle ein).
- Zeichnen Sie das KV-Diagramm für Leuchtsegment b! Leiten Sie daraus eine konjunktive Normalform ab, die mit möglichst wenig Maxtermen auskommt. Belegen Sie dazu die „X“-Werte geeignet mit „0“ und „1“.
- Skizzieren Sie, wie Sie diese Verknüpfung für b mittels einer PAL-Struktur implementieren würden (Ausgangsinverter ist verfügbar):
Wie viele p-Terms benötigen Sie bei bestmöglicher Auslegung?
Kennzeichnen Sie in Ihrer Skizze alle Verbindungen die programmiert werden müssen mit „0“
- Zeichnen Sie das KV-Diagramm für Leuchtsegment c! Leiten Sie daraus eine disjunktive Normalform ab, die mit möglichst wenig und möglichst kurzen Mintermen auskommt. Belegen Sie dazu wieder die X-Werte geeignet mit „0“ und „1“.
- Leiten Sie eine Realisierung der für c benötigten Verknüpfung mittels Multiplexer-basierter Logik (Prinzip wie im ACTEL-FPGA, ohne Register) her! Zeichnen Sie das Schaltbild Ihrer Lösung!

Beispiel 2

An einem Chip messen Sie bei einer Versorgungsspannung von $V_{DD}=2,5\text{V}$ einen mittleren Strom von 4A. Im Datenblatt finden Sie für die Kühlung folgende Angaben:

$$\theta_{JC} = 3\text{K/W}, \theta_{JA} = 10\text{K/W}$$

Es ist ein Kühlkörper vorzusehen, der besser als 2K/W spezifiziert ist.

- (a) Angenommen Sie haben einen Kühlkörper, der mit 1K/W spezifiziert ist. Dürfen Sie diesen verwenden?
- (b) Bei Verwendung eines Kühlkörpers mit 2K/W und im Betrieb bei $T_{AMB}=20^\circ\text{C}$:
Welche Temperatur T_C erreicht der Chip (Gehäuse), welche Temperatur T_J der Die?
- (c) Sie haben gerade keinen passenden Kühlkörper zur Hand und betreiben den Chip daher ohne Kühlkörper. Welche Temperatur T_C bzw. T_J erreichen Gehäuse bzw. Die nun?
- (d) Auf wie viel % verkürzt sich durch den Betrieb ohne Kühlkörper die Lebenserwartung des Chips? (Hinweis: Ermitteln Sie dazu den Temperature acceleration Factor)

$$k = 8,6 \cdot 10^{-5} \text{eV/K} ; E_{act} = 0,7 \text{eV}$$

Beispiel 3

In Abbildung 3.1 ist ein analoger Spannungsverlauf U_{in} gegeben.

- (a) Dieses Signal U_{in} wird an einen Eingang eines digitalen Gatters mit Schwellwert $U_{th} = 1,25V$ (ohne Hysterese) gelegt. Ergänzen Sie in Abbildung 3.1 den Spannungsverlauf U_{dig1} nach der Digitalisierung!
- (b) Zwecks besserer Störunterdrückung wird für die Digitalisierung von U_{in} nun ein Schmitt Trigger-Eingang verwendet. U_{dig2} zeigt den resultierenden Verlauf der Ausgangsspannung. Welche Schwellwerte hat dieser Schmitt Trigger? Wie groß ist seine Hysterese (incl. Einheit)?
- (c) Zeichnen Sie die Kennlinie (Ausgangsspannung über Eingangsspannung) dieses Schmitt Trigger! Beschriften Sie die charakteristischen Spannungswerte und kennzeichnen Sie mit Pfeilen, in welcher Richtung die Kennlinie durchlaufen wird.
- (d) Handelt es sich um einen invertierenden Schmitt-Trigger oder einen nicht-invertierenden? Begründen Sie!

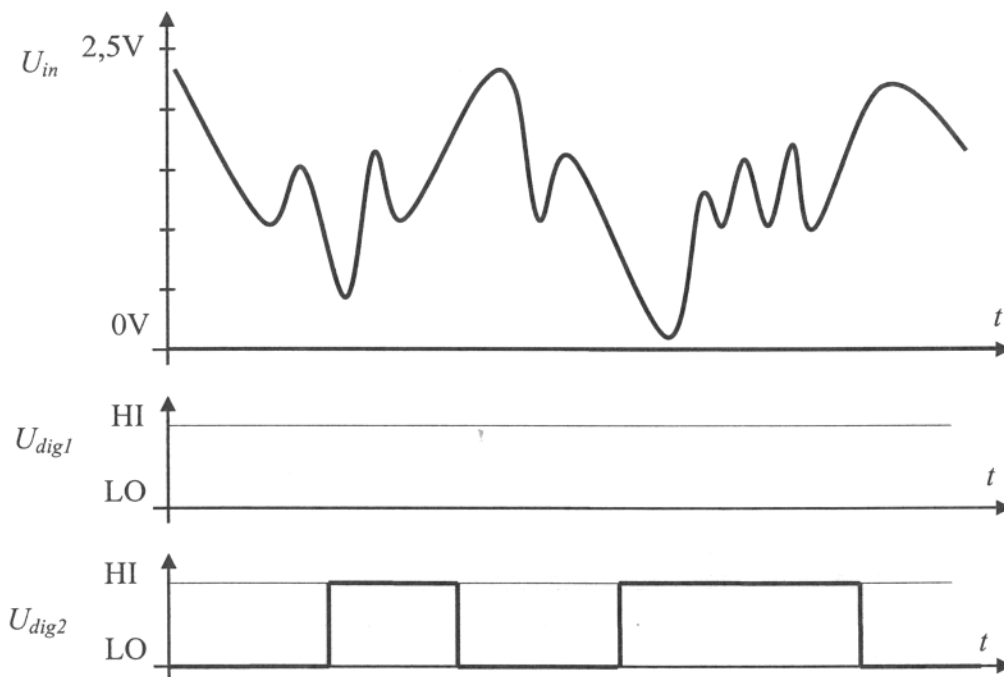


Abbildung 3.1

Beispiel 4

Gegeben ist folgender VHDL-Code einer State-Machine:

```
library IEEE;
use      IEEE.std_logic_1164.all;
use      IEEE.std_logic_UNSIGNED.all;

entity year is
  port(
    flower : in std_logic;
    grass  : out std_logic;
    sun    : out std_logic;
    bird   : in std_logic
  );
end year;
```

```
architecture season of year is
  signal tree, green:
    std_logic_vector(0 to 1);

begin

  autumn: process(tree)
  begin
    case tree is
      when "10" => green <= "11";
      when "00" => green <= "01";
      when "01" => green <= "10";
      when others => green <= "00";
    end case;
  end process autumn;

  spring: process(tree, flower)
  variable smell: std_logic_vector(0 to 3);
  begin
    if flower = '0' then
      smell := "0000";
    end if;
    case tree is
      when "10" => grass <= '1';
                        sun  <= '1';
      when "01" => grass <= '0';
                        smell := smell + '1';
                        sun  <= '1';
      when others => grass <= '0';
                        sun  <= smell(1);
    end case;
  end process spring;

  summer: process(bird, flower)
  begin
    if flower = '0' then
      tree <= "00";
    else
      if bird'event and bird = '1' then
        tree <= green;
      end if;
    end if;
  end process summer;

end season;
```

- (a) Identifizieren Sie Taktsignal und Reset!
- (b) Handelt es sich um einen synchronen oder einen asynchronen Reset? Begründen Sie!
- (c) Ist der Reset hi-aktiv oder lo-aktiv? Begründen Sie!
- (d) Nennen Sie die 3 Funktionsblöcke, die eine State-Machine stets umfassen muß? Welcher Process im obigen Code entspricht welchem Funktionsblock?
- (e) Tragen Sie den Ablauf der Zustände (State plus zugehöriger Next-State) in Abbildung 4.1 ein! Ergänzen Sie weiters den Zeitverlauf von sun, grass und smell !

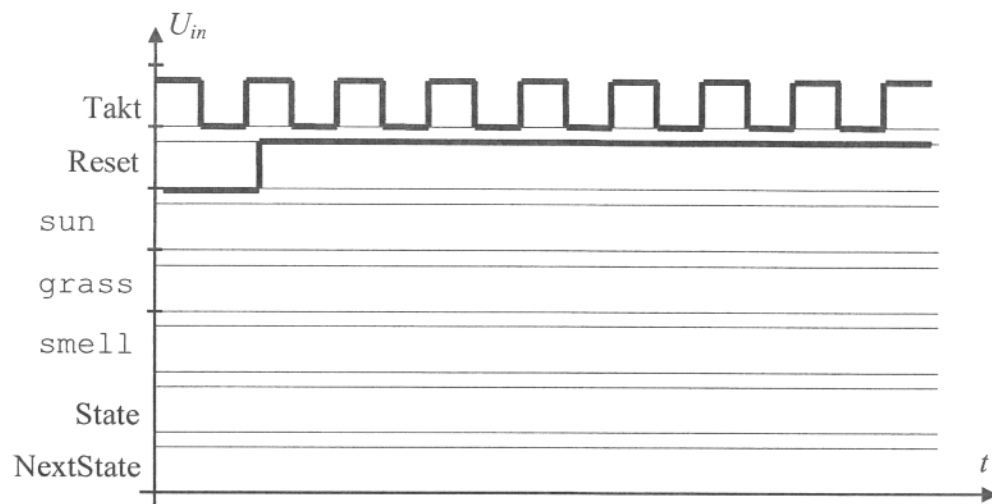


Abbildung 4.1

Vorlesungsprüfung aus Digitales Design

12. Jänner 2007

Die Arbeitszeit beträgt 2 Stunden. Als Hilfsmittel sind ausnahmslos Schreibzeug, Lineal und (nicht programmierbarer) Taschenrechner erlaubt. Schreiben Sie Ihre Antworten und Lösungen (incl. Lösungsweg!) mit Füllfeder oder Kugelschreiber (nicht rot, KEIN Bleistift!) und streichen Sie alles durch, was nicht zur Beurteilung herangezogen werden soll. Ein Abbruch der Prüfung nach Erhalt der Angaben führt in jedem Fall zu einer Beurteilung.

Tragen Sie Namen, Kennzahl und Matrikelnummer **zu Beginn** der Prüfung in die Tabelle ein und beschriften Sie jedes Blatt, das Sie abgeben möchten rechts oben mit Namen und Matrikelnummer.

Familienname:	Vorname:
Kennzahl:	Matrikelnummer:

Viel Erfolg!

Die nachfolgende Tabelle nicht beschriften!

Beispiel		Mögliche Punkte	Erhaltene Punkte
Theoriefragen		50	
Rechenbeispiel	1	15	
	2	10	
	3	15	
	4	10	
Gesamt		100	

Theoriefragen

Frage 1:

Welche Bedeutung hat die Sensitivity-List in einem VHDL Code?

Frage 2:

Welche Strecke legt eine elektromagnetische Welle auf einer Leitung innerhalb von 1ns zurück?

Frage 3:

Handelt es sich bei einem ROM um rein kombinatorische oder sequentielle Logik? Begründen Sie!

Frage 4:

Geben Sie die Signal-Resolution-Table für die Funktion XOR und die Zustände $\{0,1,X\}$ an!

Frage 5:

Was versteht man unter einem non-volatile Memory?

Frage 6:

Erläutern Sie das Prinzip des Boundary-Scan!

Frage 7:

Was versteht man im Zusammenhang mit einem PLD unter einem Product Term?

Frage 8:

Wozu benötigt man bei der Fertigung eines ICs die Masken?

Frage 8:

Altet ein Chip auch wenn er nicht in Betrieb ist? Begründen Sie!

Frage 10:

Warum werden bei CMOS-Logik die Funktionen NAND und NOR bevorzugt verwendet und nur selten AND und OR?

Rechenbeispiele

Beispiel 1

Gesucht ist eine Schaltung, mit der Sie überprüfen können, ob drei Eingangsbits logisch übereinstimmen. Die Schaltung soll also am Ausgang y genau dann 0 ausgeben, wenn die logischen Zustände an den drei Eingängen a, b, c gleich sind, also $(0,0,0)$ oder $(1,1,1)$.

- (a) Geben Sie Wahrheitstabelle und KV-Diagramm für diese Schaltung an!
- (b) Leiten Sie aus dem KV-Diagramm die Gleichung der logischen Verknüpfung einmal in konjunktiver und einmal in disjunktiver Normalform ab!
- (c) Geben Sie an, wie Sie diese Schaltung mittels LUT realisieren können: Wie viele Eingänge und Ausgänge benötigt die LUT, wie werden diese beschaltet (Skizze)? Geben Sie den Speicherinhalt der LUT an!
- (d) Geben Sie eine möglichst einfache Realisierung der Funktion mittels AOI bzw. OAI an (was ist einfacher?). Zeichnen Sie die Schaltung als Realisierung mittels FETs!
- (e) Vergleichen Sie die Implementierungen aus (c) und (d): Wie viele Gatteräquivalente benötigen Sie für (d)? Nehmen Sie an, für eine Speicherzelle der LUT werden 6 Transistoren benötigt, und die restliche Logik sei vernachlässigbar – wie viele Gatteräquivalente benötigt die LUT-Lösung aus (c) ?

Beispiel 2

In Abbildung 2.1 ist die Kennlinie eines Schmitt-Trigger gegeben.

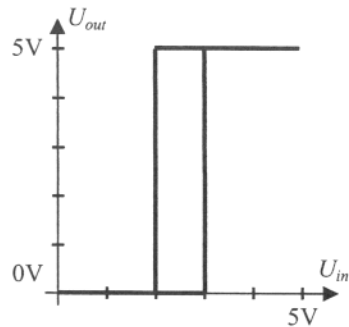


Abbildung 2.1

- Kennzeichnen Sie durch Pfeile in den vertikalen Abschnitten der Kennlinie die Richtung, in der die Kennlinie durchlaufen wird.
- Wie groß ist die Hysterese dieses Schmitt-Trigger (incl. Einheit!)?
- Handelt es sich um einen invertierenden Schmitt-Trigger oder einen nicht-invertierenden? Begründen Sie!

Gegeben ist weiters der Spannungsverlauf U_{in} in Abbildung 2.2:

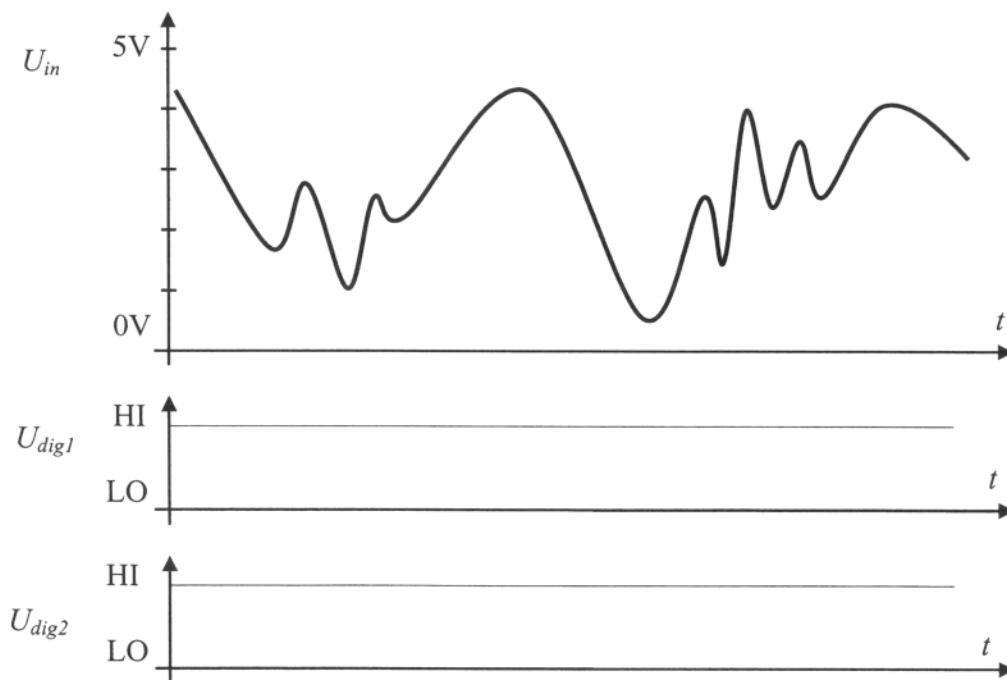


Abbildung 2.2

- Wie sieht die Ausgangsspannung am Ausgang des Schmitt-Trigger aus, wenn Sie die Eingangsspannung nach Abbildung 2.2 anlegen? Tragen Sie den Verlauf als U_{dig1} in Abbildung 2.2 ein!
- Nehmen Sie nun einen Komparator mit fixer Schwellspannung von 2V, d.h. ohne Hysterese, an und stellen Sie dessen Ausgangsspannungsverlauf als U_{dig2} in Abbildung 2.2 dar!

Beispiel 3

Gegeben ist die in Abbildung 3.1 dargestellte Schaltung: Ein asynchrones Eingangssignal mit mittlerer Frequenz $f_{input} = 10\text{MHz}$ soll in eine synchrone Schaltung mit Taktfrequenz $f_{clk} = 200\text{MHz}$ übernommen werden. Dazu wird das Flip-Flop $FF1$ als einstufiger Synchronizer vorgeschaltet, wobei zwischen Ausgang des Synchronizers und Eingang der synchronen Schaltung ein Routing-Delay $t_{route} = 1,5\text{ns}$ auftritt. Laut Datenblatt haben die beiden Flip-Flops folgende Parameter:

$$t_{SU} = 0,5\text{ns} ; \quad \tau_C = 0,2\text{ns} ; \quad T_0 = 0,1\text{ns} .$$

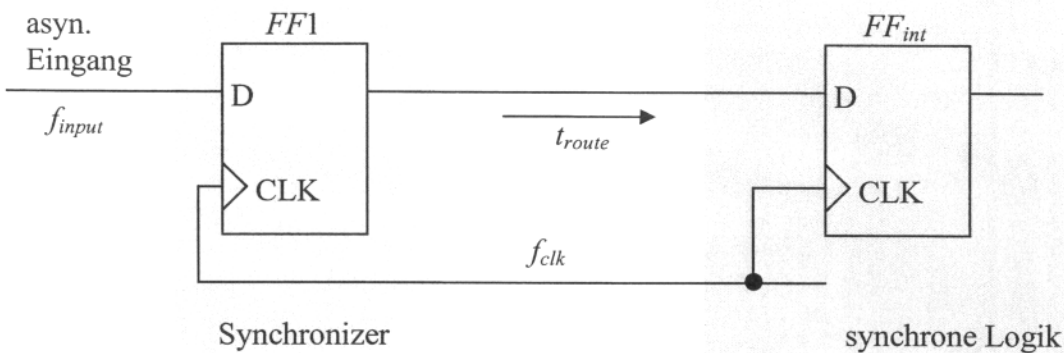


Abbildung 3.1

(a) Wie groß ist die Resolution Time für FF_{int} ? Welche MTBU ist zu erwarten?

Zur Erhöhung der MTBU soll der Synchronizer zweistufig gemacht werden, indem ein weiteres Flip-Flop, $FF2$, mit identischen Eigenschaften zwischen den Ausgang von $FF1$ und den Eingang von FF_{int} geschaltet wird. Damit ergibt sich die in Abbildung 3.2 dargestellte Schaltung. Das Routing Delay zwischen zwei benachbarten Flip-Flops betrage weiterhin $t_{route} = 1,5\text{ns}$.

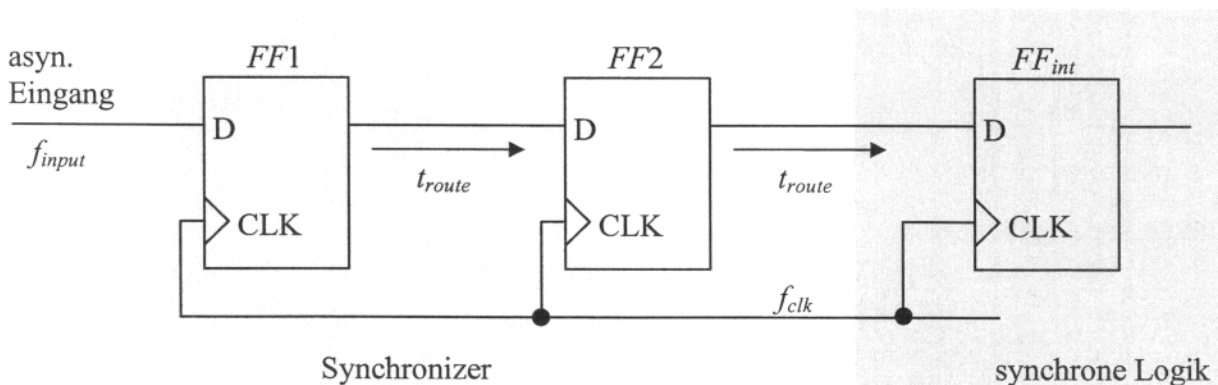


Abbildung 3.2

- (b) Wie groß die Resolution time für $FF2$? Welche MTBU ist an $FF2$ zu erwarten?
- (c) Welche MTBU ergibt sich nun für FF_{int} ? (Hinweis: Beachten Sie bei der Berechnung, dass sich ein asynchroner Eingangsimpuls für FF_{int} nur im Falle eines Upset an $FF1$ ergibt!)

Beispiel 4

Gegeben ist folgender VHDL-Code einer State-Machine:

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_UNSIGNED.all;

entity head is
port(
    mouth: in std_logic;
    nose: in std_logic;
    hair: out std_logic;
    eye: in std_logic;
    ear: in std_logic;
    brain: out std_logic;
);
end head;
```

```
architecture parts of head is
    signal sweat, tears:
        std_logic_vector(0 to 1);
begin
    look: process(sweat)
    begin
        case sweat is
            when "10" => tears <= "11";
            when "00" => tears <= "01";
            when "01" => tears <= "10";
            when others => tears <= "00";
        end case;
    end process look;

    listen: process(ear, eye, tears)
    begin
        if ear'event and ear = '1' then
            if eye = '0' then
                sweat <= "00";
            else
                sweat <= tears;
            end if;
        end if;
    end process listen;

    smell: process(sweat, mouth, nose)
    begin
        brain <= '1';
        if nose = mouth then
            brain <= '0';
        end if;
        case sweat is
            when "10" => hair <= '0';
            when others => hair <= '1';
        end case;
    end process smell;
end parts;
```

- (a) Identifizieren Sie Taktsignal und Reset! Handelt es sich um einen synchronen oder einen asynchronen Reset? Begründen Sie!
- (b) Ist der Reset hi-aktiv oder lo-aktiv? Begründen Sie!
- (c) Welches Signal repräsentiert den Zustand der State-Machine? Geben Sie den Namen jenes Prozesses im obigen Code an, in dem dessen synchrone Aktualisierung erfolgt! Begründen Sie!
- (d) Welches Signal repräsentiert den Folgezustand (Next-State)? Geben Sie den Namen jenes Prozesses im obigen Code an, in dem dessen Aktualisierung erfolgt! Begründen Sie!
- (e) Welche Signale sind Inputs, welche Outputs? Geben Sie den Namen jenes Prozesses im obigen Code an, in dem die Aktualisierung der Outputs erfolgt!
- (f) Handelt es sich um eine State-Machine vom Moore-Type oder vom Mealy-Type? Begründen Sie!