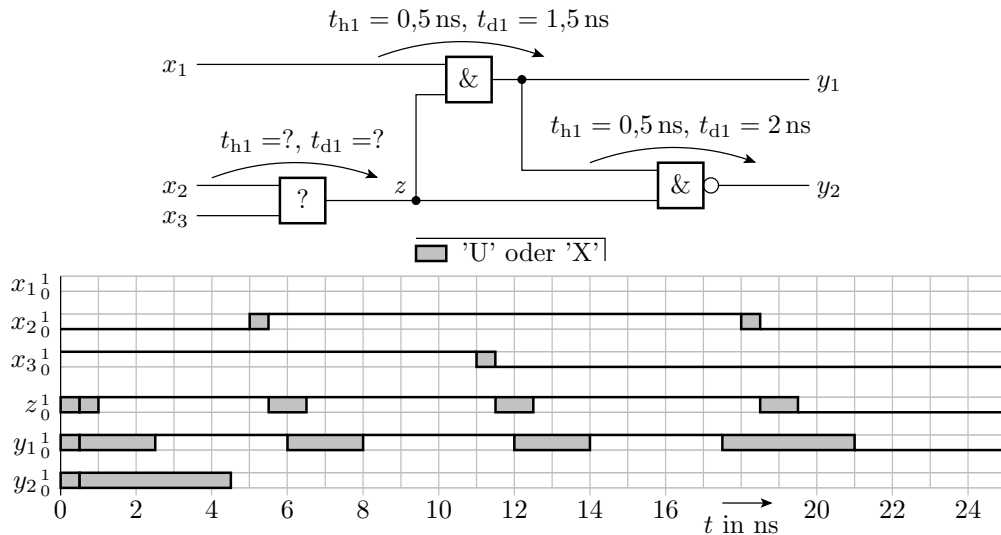


### Prüfungsklausur Grundlagen der Digitaltechnik

**Hinweise:** Die Bearbeitungszeit beträgt 90 Minuten. Schreiben Sie die Lösungen, so weit es möglich ist, auf die Aufgabenblätter. Tragen Sie Namen, Matrikelnummer und Studiengang in die nachfolgende Tabelle ein. Zum Bestehen sind  $\geq 20$  Punkte erforderlich. Geben Sie die Aufgabenblätter zum Schluss mit ab (ZPHÜ – Zusatzpunkte Hausübungen).

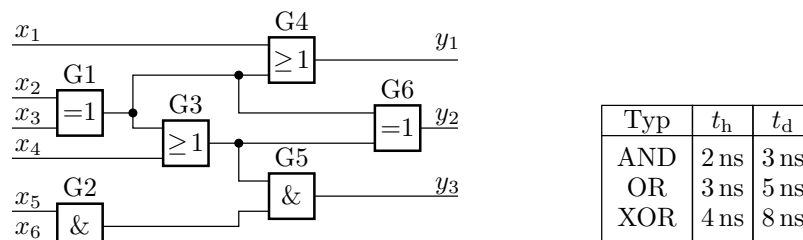
| Name | Matrikelnummer | Studiengang | Punkte | ZPHÜ | Note |
|------|----------------|-------------|--------|------|------|
|      |                |             |        |      |      |

**Aufgabe 1:** Gegeben ist die nachfolgende Schaltung mit den Halte- und Verzögerungszeiten.



- Bestimmen Sie aus dem Signalverlauf von  $z$  die logische Funktion, die Gatterhaltezeit  $t_{h1}$  und die Gatterverzögerungszeit  $t_{d1}$  vom ersten Gatter. 2P
- Ergänzen Sie die Signalverläufe für  $x_1$  und  $y_2$ . 2P
- Notieren Sie die VHDL-Anweisungen zur Erzeugung der Signalverläufe  $x_1, x_2, x_3$  und  $z$ . 2P

**Aufgabe 2:** Gegeben sind die nachfolgende Schaltung und die Halte- und Verzögerungszeiten der Gatter:



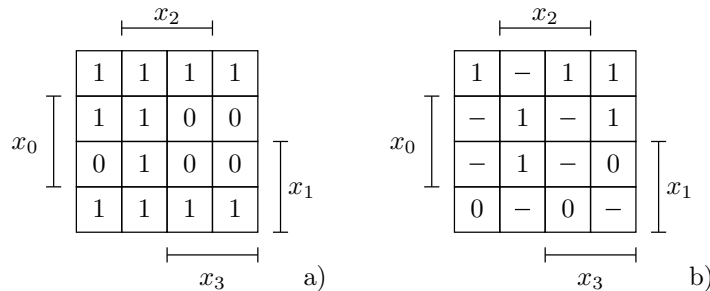
- a) Bestimmen Sie für alle Pfade durch die Schaltung die Haltezeit  $t_h$  und die Verzögerungszeit  $t_d$ . 6P
- b) Wie groß sind die Halte- und die Verzögerungszeit der Gesamtschaltung? 1P
- c) Zeichnen Sie die Schaltung mit Abtastregistern an den Ein- und Ausgängen. Die Register sollen die Vorhaltezeit  $t_s = 2 \text{ ns}$ , die Nachhaltezeit  $t_n = 0$ , die Haltezeit  $t_{hr} = 0,5 \text{ ns}$  und die Verzögerungszeit  $t_{dr} = 2 \text{ ns}$  haben. Wie groß ist die max. Taktfrequenz? 2P

**Aufgabe 3:** Gegeben ist die nachfolgende durch ein FCMOS-Gatter zu realisierende Funktion:

$$y = \overline{b(a \vee \bar{a}\bar{b}) \vee d(b \vee ad) \vee b(\bar{b} \vee b) \vee a(c \vee b) \vee ab(c \vee bd)}$$

- a) Vereinfachen Sie die logische Funktion soweit wie möglich. 1,5P
- b) Stellen Sie die Funktionen  $f_n$  und  $f_p$  des NMOS- und des PMOS-Netzwerks des FCMOS-Gatters auf. 1P
- c) Zeichnen Sie das FCMOS-Gatter als Transistorschaltung<sup>1</sup>. 1,5P

**Aufgabe 4:** Gegeben sind die beiden nachfolgenden KV-Diagramme:



- a) Entwickeln Sie die linke Schaltung einmal nach den Einsen und einmal nach den Nullen. 3P
- b) Entwickeln Sie die rechte Schaltung entweder nach den Einsen oder nach den Nullen. Ersetzen Sie »-« (Don't-Care-Felder) vorher so durch '0' oder '1', dass die Schaltung minimal wird. 1P

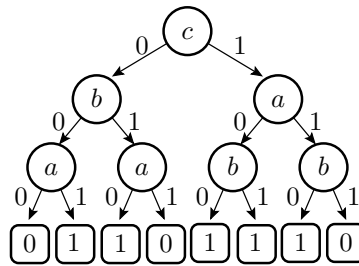
**Aufgabe 5:** Gegeben ist die nachfolgende Menge von Mintermen  $x_5x_4x_3x_2x_1x_0$ , für die die logische Funktion eins ist:

$$K \in \left\{ \underbrace{010\ 101}_a, \underbrace{011\ 111}_b, \underbrace{001\ 111}_c, \underbrace{110\ 000}_d, \underbrace{100\ 000}_e, \underbrace{000\ 000}_f, \underbrace{011\ 110}_g, \underbrace{010\ 000}_h, \underbrace{000\ 100}_i \right\}$$

- a) Stellen Sie nach dem Verfahren von Quine und McCluskey die quineschen Tabellen nullter bis zweiter Ordnung auf. Bezeichnen Sie die Primterme, die in den quineschen Tabellen durch keine Konjunktionen höherer Ordnung abgedeckt sind mit  $P_1$  bis  $P_n$ . 3P
- b) Wählen Sie aus den Primtermen eine minimale Abdeckungsmenge aus und bestimmen Sie den zugehörigen minimierten logischen Ausdruck. 1P

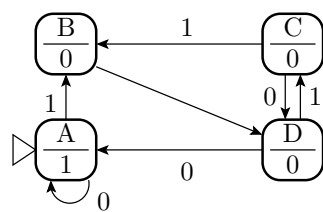
<sup>1</sup>Statt der Inverter genügt es, die entsprechenden Ein- oder Ausgangsvariablen als invertiert anzugeben.

**Aufgabe 6:** Gegeben ist das nachfolgende binäre Entscheidungsdiagramm (BDD):



- Stellen Sie die Wertetabelle dazu auf. 1P
- Entwickeln Sie aus der Wertetabelle das geordnete binäre Entscheidungsdiagramm (OBDD) mit der Abfragereihenfolge a-b-c. 1P
- Minimieren Sie das geordnete binäre Entscheidungsdiagramm zu einem reduzierten geordneten binären Entscheidungsdiagramm (ROBDD). 2P
- Entwickeln Sie aus dem ROBDD die Multiplexerschaltung und vereinfachen Sie diese durch Ersatz der Multiplexer mit konstanten Eingaben durch funktionsgleiche Gatterschaltungen. 1P

**Aufgabe 7:** Gegeben sind der folgende Automatengraph und die Zustandskodierung:



Zustände: A, B, C, D,  
 Codierung: 000 001 010 100  
 Übernahme bei  $T = \lambda$   
 Initialisierung bei  $I = '1'$   
 Eingabe:  $x$  (std\_logic)  
 Ausgabe:  $y$  (std\_logic)

- Füllen Sie die nachfolgende Tabelle für die Übergangs- und die Ausgabefunktion aus. 2P

| Zustand, symbolisch        | A |   | B |   | C |   | D |   |
|----------------------------|---|---|---|---|---|---|---|---|
| Zustand $z_2 z_1 z_0$      |   |   |   |   |   |   |   |   |
| Ausgabe $y$                |   |   |   |   |   |   |   |   |
| Eingabe $x$                | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Folgezustand, symbol.      |   |   |   |   |   |   |   |   |
| Folgezustand $z_2 z_1 z_0$ |   |   |   |   |   |   |   |   |

- Gegeben ist die Eingabefolge in der nachfolgenden Tabelle. Ergänzen Sie die die Zustandsfolge und die Ausgabefolge. 2,5P

|             |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Zustand     | A | A | B | D | C | B | D | A | A | B | D | C | D |
| Ausgabe $y$ |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Eingabe $x$ |   |   |   |   |   |   |   |   |   |   |   |   |   |

- Vervollständigen Sie in der nachfolgenden VHDL-Beschreibung des Automaten in den mit »\*« gekennzeichneten Zeilen die Port- und Signalvereinbarungen, die Beschreibung der Übergangsfunktion und die Ausgabeanweisung. 3,5P

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity Automat is
4  port(
5  * T, I: ...           -- Takt- und Init.-Signal
6  * x: ...             -- Eingangssignal
7  * y: ...             -- Ausgabesignal
8  end entity;
9  architecture Verhalten of Automat is
10 * signal z: ...      -- Zustand
11 begin
12 process (T, I)
13 begin
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 end process;
35 * y <=                -- Ausgabezuweisung
36 end architecture;

```

**Aufgabe 8:** Zeichnen Sie die Schaltung, die der nachfolgende VHDL-Prozess beschreibt:

2P

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity Afg8 is end entity;
4
5  architecture a of Afg8 is
6  signal Takt, Reset, sel, x1, x2, x3, x4: std_logic;
7  signal y: std_logic_vector (2 downto 1);
8  begin
9  process (Reset, Takt)
10 variable z : std_logic_vector(1 downto 0);
11 begin
12 if Reset = '0' then y <="00";
13 elsif rising_edge (Takt) then
14 z(0) := x1 xor x2;
15 z(1):= x3 nand x4;
16 if sel = '1' then
17 y(1) <= z(0);

```

```
18     y(2) <= z(1);
19     end if;
20     end if;
21     end process;
22 end architecture;
```

**Zur Bewertung:**

| Aufgabe             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Summe |
|---------------------|---|---|---|---|---|---|---|---|-------|
| max. Punktezahl     | 6 | 9 | 4 | 4 | 4 | 5 | 8 | 2 | 42    |
| erzielte Punktezahl |   |   |   |   |   |   |   |   |       |