

**Klausur (08.04.2011) :**  
**Technische Grundlagen der Informatik 1**  
**Digitale Systeme**  
**WS 2010/2011**

<b>Vorname</b>	: Max
<b>Name</b>	: Mustermann
<b>Matrikelnummer</b>	: 123456
<b>Klausur-Code</b>	: 007

**Wichtige Hinweise:**

- Mobiltelefone sind auszuschalten
- für die Lösung darf weder Bleistift noch Rotstift verwendet werden
- für die Lösungen sind die Aufgabenblätter zu verwenden
- der Lösungsweg muss nachvollziehbar sein und sich an der Aufgabenstellung orientieren
- Betrugsversuche werden mit einem Nichtbestehen der Klausur geahndet

Mit der Veröffentlichung meines Ergebnisses unter dem oben angegebenen Klausur-Code erkläre ich mich einverstanden (Bitte ankreuzen)

Hiermit erkläre ich, dass alle personenbezogenen Angaben richtig sind, ich die Hinweise gelesen und verstanden habe und ich mich gesundheitlich in der Lage fühle an der Klausur teilzunehmen.

Unterschrift: \_\_\_\_\_

<b>Aufgabe</b>	1	2	3	4	5	6	$\Sigma$
<b>max. Punkte</b>	12	10	13	7	7	11	60
<b>erreichte Punkte</b>							
<b>Korrektor</b>							

### Aufgabe 1 (12 Punkte)

- (a) Kennzeichnen Sie in der folgenden Tabelle die richtigen Aussagen mit **ja** und falsche Aussagen mit **nein**! Wenn Sie sich nicht sicher sind lassen Sie die Zeile unausgefüllt, da falsche Markierungen zu Punktabzug führen.

	ja	nein
Das Tafelauswahlverfahren ergänzt Quine McCluskey ...		
... um die Möglichkeit die minimale Anzahl an Primtermen zu ermitteln!		
... um die Möglichkeit die KKNF und KDNF zu ermitteln!		
Die folgende Kombination an Grundgattern reicht aus um jede beliebige boolesche Funktion abzubilden :		
NAND und NOR		
XOR und NOT		
Die folgenden Bausteine sind wiederprogrammierbar :		
FPGA		
ROM		
EEPROM		
Das QMC-Verfahren findet immer die minimale Primtermüberdeckung.		
Voll- und Halbaddierer besitzen jeweils zwei Ein- und Ausgänge.		
Ein 4-zu-1 Multiplexer hat zwei Steuereingänge.		
Zur Kennzeichnung einer Bereichsüberschreitung bei Operationen auf vorzeichenlosen Dualzahlen, wird auf das <i>Overflow</i> -Bit zurückgegriffen.		
Beim sequentiellen Verarbeiten aufeinanderfolgender Gray-Codes ändert sich jeweils ein Bit.		
2Komplement-Zahlen verwenden zur Darstellung der dezimalen '0' zwei verschiedene Codes.		
Bei vorzeichenbehafteten Dualzahlen zeigt eine führende '0' eine negative Zahl an und eine '1' eine positive Zahl.		
In KV-Diagrammen entsprechen '0'en den Maxtermen der abgebildeten Funktionen.		
Aus Maxtermen bildet man die KDNF einer Funktion.		
Die Tison-Methode bestimmt die minimale Anzahl von Primimplikanten.		
Die Funktionalität eines arithmetischen und eines logischen Linkshift ist die selbe.		
Ein getaktetes D-Flipflop ist während T=1 transparent.		
Ein asynchrones Flipflop arbeitet nie transparent.		

- (b) Im nachfolgenden VHDL-Quellcode ist die Komponente F00 beschrieben. Leider sind fünf Fehler enthalten. Finden und **korrigieren** Sie mindestens vier von diesen!

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

entity F00 is
port (
    a, b, c: input std_logic;
    z: out std_logic);
end entity;

architecture behavioral of BAR is
    signal m, n : bit;
begin
    m <= a nor b nor c;
    n => b and not(c) nor a and c;
    a <= b xor c;
    z <= m or n;
end architecture;
```

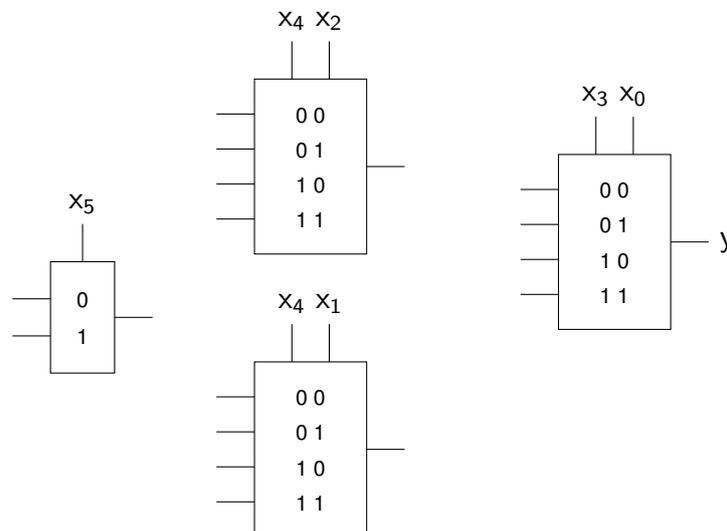
## Aufgabe 2 (10 Punkte)

Gegeben ist die folgende Funktion

$$y = \overline{x_0} \overline{x_2} \overline{x_3} \overline{x_4} \overline{x_5} + \overline{x_0} x_1 x_3 x_4 \overline{x_5} + \overline{x_0} x_2 \overline{x_3} \overline{x_4} + \overline{x_0} x_2 \overline{x_3} x_4 + \overline{x_0} \overline{x_1} x_3 \overline{x_4} + x_0 x_3 \overline{x_5}$$

Formen Sie die gegebene Funktion mit Hilfe von Shannon so um, dass sie auf die vorgegebene Multiplexerstruktur projiziert werden kann und bilden Sie anschließend die umgeformte Funktion unter Verwendung aller Multiplexer auf die Multiplexerstruktur ab.

**Hinweis:** Verwenden Sie dabei die Multiplexer so häufig wie möglich.



### Aufgabe 3 (13 Punkte)

Das nachfolgende VHDL-Quellcodefragment enthält Signalzuweisungen:

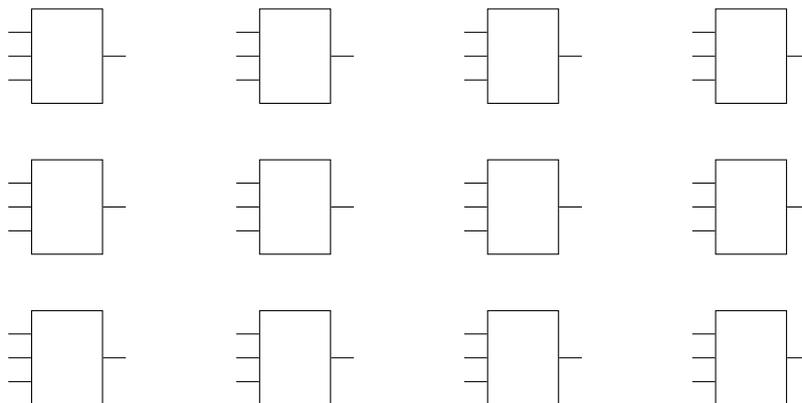
```

...
signal a, b, c, d, v, w, x, y, z : bit;
begin
  v <= not(a or b or c or d);
  w <= b and not(c) nor a and c;
  x <= w or d;
  y <= not(w and not(d));
  z <= not(d) and c and (b nor v);
...

```

(a) Lesen Sie die Funktionen x, y und z in Abhängigkeit von a, b, c und d aus!

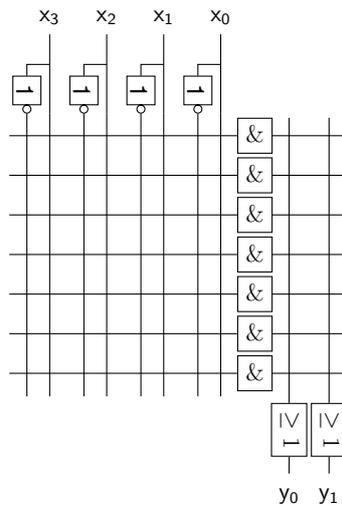
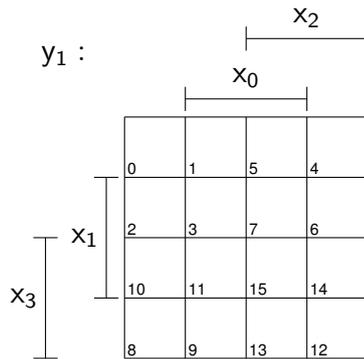
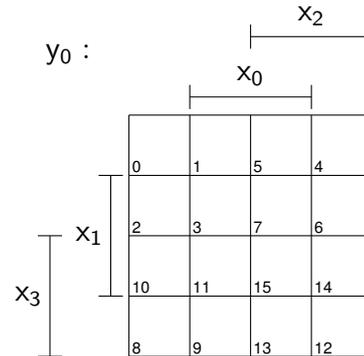
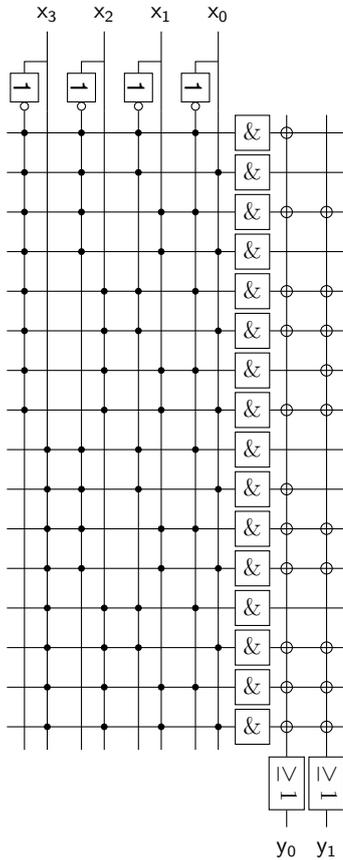
(b) Erstellen Sie für die Funktionen x und y ein Schaltnetz unter ausschließlicher Zuhilfenahme der gegebenen Gatterstruktur. Kennzeichnen Sie die benutzten Gatter mit den der Funktionalität entsprechenden Symbolen (jede Funktion soll direkt durch Gatter abbildbar sein). Setzen Sie wo nötig an den Ein- oder Ausgängen Negationssymbole. Beschalten Sie nicht benötigte Eingänge so, dass die Funktionalität des Gatters sicher gegeben ist.



### Aufgabe 4 (7 Punkte)

Übertragen Sie die Funktionen  $y_0, y_1$  aus dem unten abgebildeten PROM in die angegebenen KV-Diagramme. Minimieren Sie die beiden Funktionen und tragen Sie diese danach möglichst effizient in das gegebene PLA ein.

**Hinweis:** Beim ersten Schritt ist es sinnvoll sich vorher zu überlegen, welche strukturellen Eigenschaften von PROM und KV-Tafeln zusammenhängen.



### Aufgabe 5 (7 Punkte)

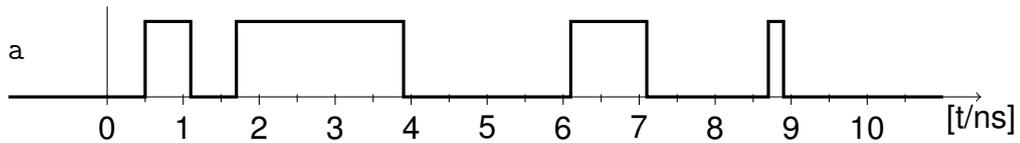
Das nachfolgende VHDL-Quellcodefragment enthält Signalzuweisungen:

```

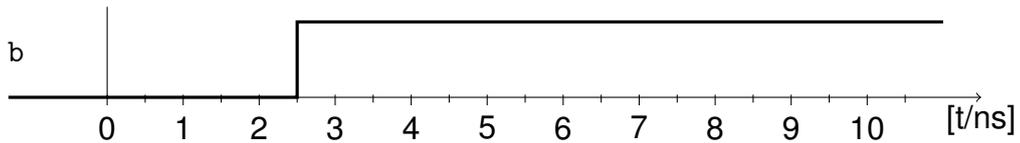
...
signal a, b, o1, o2, o3 : bit;
begin
o1 <= inertial a after 1.5 ns;
o2 <= transport not (a and b) after 1.5 ns;
o3 <= reject 0.5 ns inertial a and b after 1 ns;
...

```

Gegeben ist außerdem der Signalverlauf des Signals a:



sowie der des Signals b:

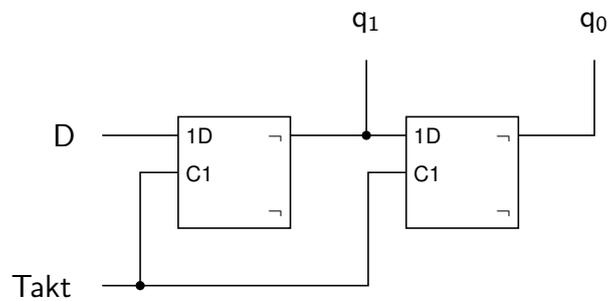


Ordnen Sie den folgenden gegebenen Signalverläufen die passenden Signale zu und tragen Sie jeweils den Namen (o1, o2, o3) in das entsprechende graue Kästchen ein. Kennzeichnen Sie die Signalverläufe, die nicht das Ergebnis einer der aufgeführten Signalzuweisungen sein können, durch ein x im grauen Kästchen:

<input type="checkbox"/>	

## Aufgabe 6 (11 Punkte)

Im nachfolgenden Bild ist die Schaltung eines 2-Bit Schieberegisters aus zwei Master-Slave-Flipflops gegeben. Mit jeder ansteigenden Taktflanke auf der Leitung Takt wird der aktuelle Speicherinhalt  $q_1, q_0$  um eine Stelle nach rechts versetzt und der am Eingang D anliegende Wert vom linken Flip-Flop übernommen.



- (a) Das Schieberegister soll als endlicher Automat betrachtet werden. Um welchen Automatentypen handelt es sich ?
- (b) Stellen Sie die Zustandsübergangstabelle des Automaten auf! Wie viele Zustände kann es maximal geben ?

(c) Interpretieren Sie den Speicherinhalt als Dualzahl  $q_1 \cdot 2^1 + q_0 \cdot 2^0$ . Stellen Sie das Zustandsdiagramm auf, indem Sie Knoten benutzen, die dem dual interpretierten Speicherinhalten entsprechen.

(d) Geben Sie bei Zustand 0 beginnend die längste Folge sich nicht wiederholender Zahlen durch den Zustandsgraphen an ! Ist die Folge eindeutig ?

