



**Aufgabe 1 (14 min, 3 + 7 + 2 + 2 Punkte): Karnaugh-Diagramm**

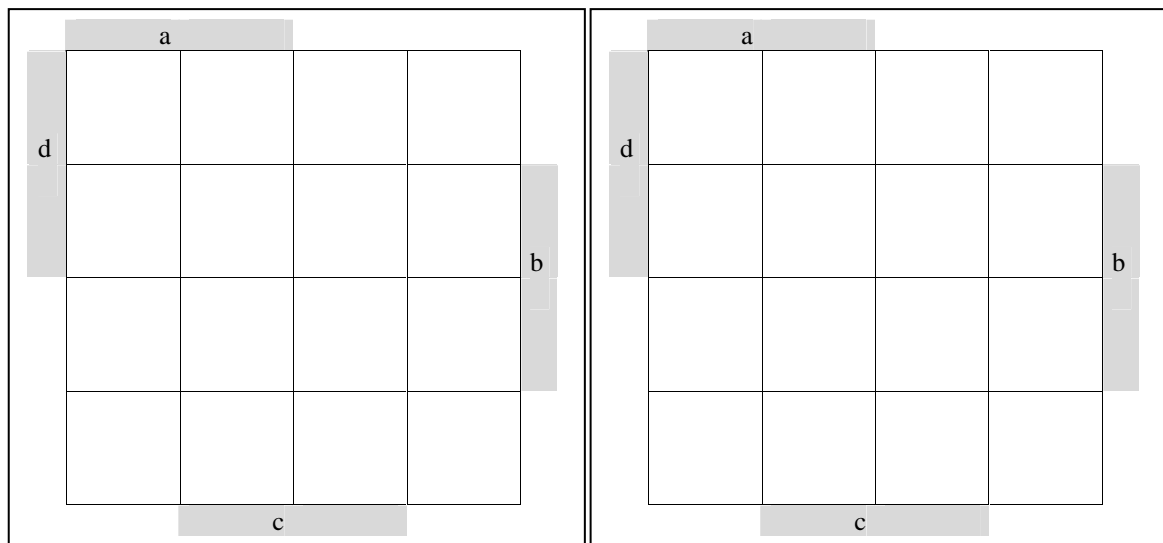
Gegeben sei die boolesche Funktion

$$F = a \bar{c} d + b c d + \bar{a} \bar{c} d + b c \bar{d} + \bar{a} \bar{b} \bar{d} + \bar{a} b \bar{c} \bar{d}$$

- (a) Bestimmen Sie eine minimale Darstellung von F mit Hilfe des Karnaugh-Diagramms: Füllen Sie dazu das Diagramm zunächst entsprechend mit Einsen aus. Markieren Sie dann alle gefundenen **Primimplikanten** und nummerieren Sie diese durch.

Auszufüllendes Karnaugh-Diagramm:

Ersatzdiagramm:



- (b) Bestimmen Sie die **minimalen** Überdeckungsindexmengen  $M\ddot{U}(P_i, x)$ , für alle Primimplikanten. (Lassen Sie die Menge leer, falls  $P_i$  nicht existiert.)

$$M\ddot{U}(P_1, x) = \{ \{1\}, \{3, 6\} \}$$

$$M\ddot{U}(P_2, x) = \{ \{2\}, \{5, 6\} \}$$

$$M\ddot{U}(P_3, x) = \{ \{3\} \}$$

$$M\ddot{U}(P_4, x) = \{ \{4\}, \{2, 3\}, \{1, 5\}, \{3, 5, 6\} \}$$

$$M\ddot{U}(P_5, x) = \{ \{5\} \}$$

$$M\ddot{U}(P_6, x) = \{ \{6\} \}$$

- (c) Geben Sie die sich aus (b) ergebende Überdeckungsfunktion  $\ddot{U}F$  **ohne** weitere Vereinfachungen an.

$$\ddot{U}F = \langle P_3 P_5 P_6 (P_1 + P_3 P_6) (P_2 + P_5 P_6) (P_4 + P_2 P_3 + P_1 P_5 + P_3 P_5 P_6) \rangle$$

- (d) Geben Sie eine minimale Darstellung als minimale boolesche Funktion hier an:

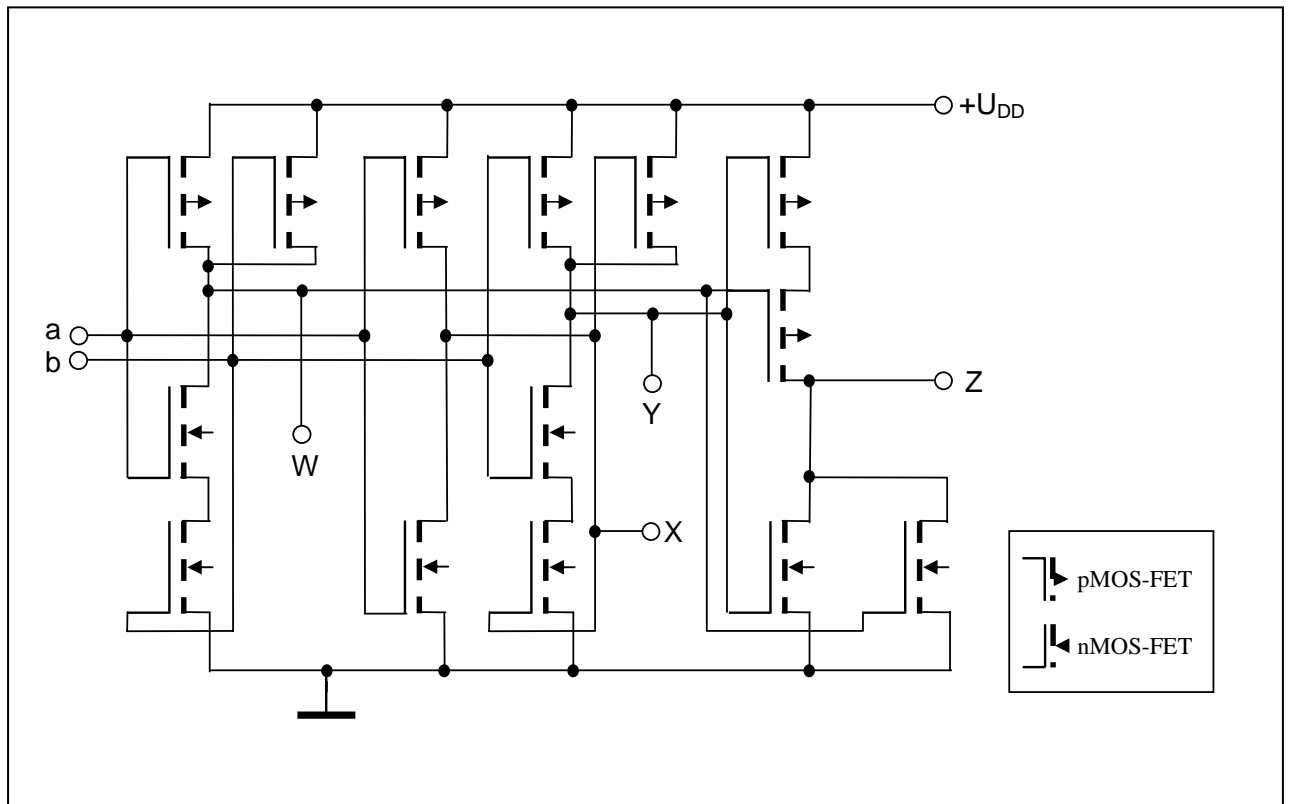
$$F_{\text{Min}} = P_3 + P_5 + P_6 = b c + \bar{a} \bar{d} + \bar{c} d$$

Name:

Matrikel-Nummer:

**Aufgabe 2 (8 min, 8 Punkte): MOS-Technologie**

Gegeben ist die folgende Schaltung mit nMOS- und pMOS-Feldeffekttransistoren.



a) Füllen Sie die folgende Wertetabelle aus:

a	b	W	X	Y	Z
Funktion: (nur 2 Variablen) <sup>1</sup>		$\overline{a \cdot b}$	$\overline{a}$	$\overline{X \cdot b}$	$\overline{W + Y}$
0	0				
0	1				
1	0				
1	1				

<sup>1</sup> Benutzen Sie ggf. auch W, X, Y, Z.

Name:

Matrikel-Nummer:

**Aufgabe 3 (14 min, 4 + 10 Punkte): Automaten und VHDL**

Gegeben ist die untenstehende VHDL-Beschreibung für einen Mealy-Automaten.

- (a) Geben Sie in dem dafür vorgesehenen Kasten den Zustandsübergangsgraphen an.
- (b) Füllen Sie für die gegebene Eingangsfolge den *Trace* des Automaten in der Tabelle auf der folgenden Seite aus.

```
Entity AUTOMAT is
  Port ( CLK : In std_logic;
        RES : In std_logic;
        X : In std_logic;
        Y : Out std_logic );
end AUTOMAT;

architecture BEHAVIORAL of AUTOMAT is

  type fsm_state is (C1, C2, C3, C4);
  signal CS, NS: fsm_state;

begin
  SEQ: process (CS, X)
  begin
    case CS is
      when C1 =>
        if X = '0' then
          Y <= "0" ;
          NS <= C3 ;
        else
          Y <= "1" ;
          NS <= C3 ;
        end if;
      when C2 =>
        if X = '0' then
          Y <= "0" ;
          NS <= C2 ;
        else
          Y <= "0" ;
          NS <= C1 ;
        end if;
      when C3 =>
        if X = '0' then
          Y <= "1" ;
          NS <= C1 ;
        else
          Y <= "0" ;
          NS <= C2 ;
        end if ;
    end case ;
  end process ;
```

```
MEM: process(CLK, RES)
begin
  if (RES = '1') then
    CS <= C1 ;
  else
    if (CLK'event and CLK = '1') then
      CS <= NS;
    end if;
  end if;
end process ;

end BEHAVIORAL ;
```

*Lösung zu (a):*

Name:

Matrikel-Nummer:

Fortsetzung zu Aufgabe 3:

Lösung zu Aufgabenteil (b):

Füllen Sie für die gegebene Eingangsfolge den **Trace** des Automaten in der Tabelle aus.

CLK	0	0	1	0	1	0	1	0	1	1
RES	1	0	0	0	0	0	0	0	0	0
X	0	1	1	0	0	0	0	1	1	0
Y	0	1	0	1	0	0	1	0	0	0
NS	C1	C3	C2	C1	C3	C3	C1	C2	C1	C2
CS	C1	C1	C3	C3	C1	C1	C3	C3	C2	C2

Ersatztable zu Aufgabenteil (b): (Streichen Sie ggf. die obige Lsg. durch)

CLK	0	0	1	0	1	0	1	0	1	1
RES	1	0	0	0	0	0	0	0	0	0
X	0	1	1	0	0	0	0	1	1	0
Y										
NS	C1									
CS										

**Aufgabe 4 (8 min, 2 + 6 Punkte): Division**

- (a) Rechnen Sie die Werte  $X = 15$ ,  $Y = 4$  und  $-Y$  in das **2er-Komplement** unter Verwendung von 5 Bit inklusive Vorzeichen um. **Geben Sie jeweils die Werte aller 5 Bits an** (das höchstwertigste Bit steht links).

$$X = 15_{10} = 01111_2,$$

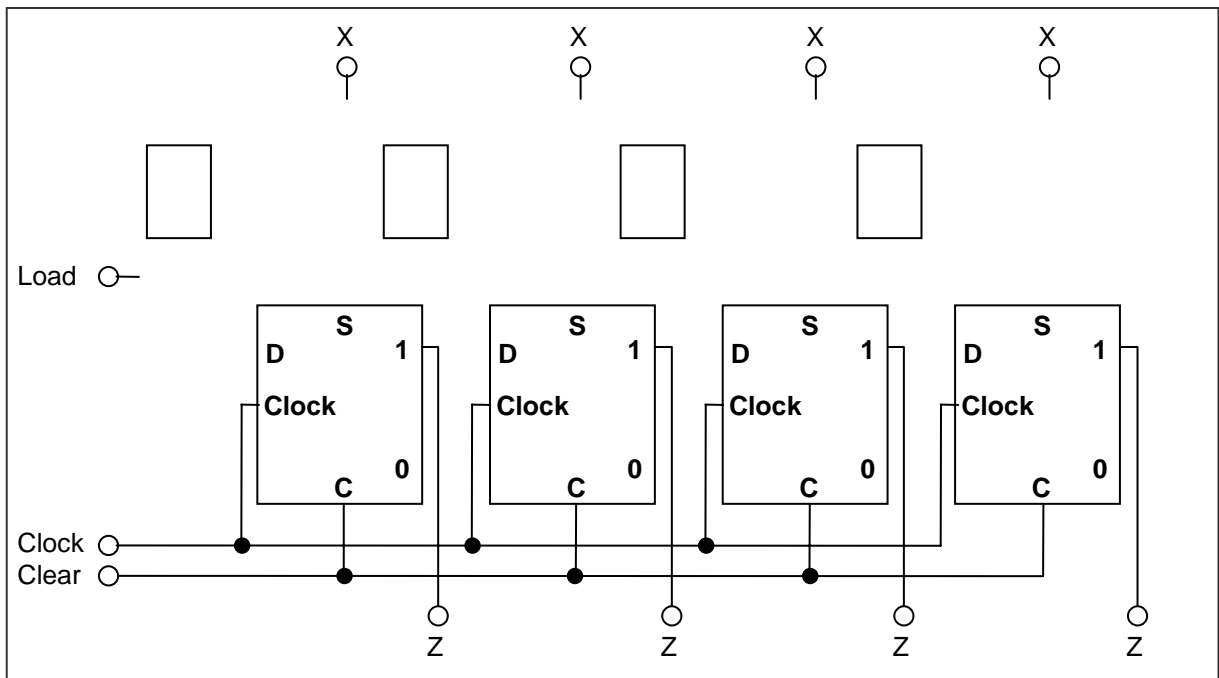
$$Y = 4_{10} = 00100_2, \quad -Y = -4_{10} = 11100_2$$

- (b) Wenden Sie den aus der Vorlesung bekannten "Restoring Algorithmus" zur Festkomma-division auf die Werte  $Q := X = 15$  und  $Y = 4$  an. Das Register Q ist 4 Bit breit, AC besitzt ein 5tes Bit für das intern benötigte Vorzeichen. **Markieren Sie am Ende der Rechnung deutlich wo in den Registern der Quotient bzw. der Rest abzulesen ist!**

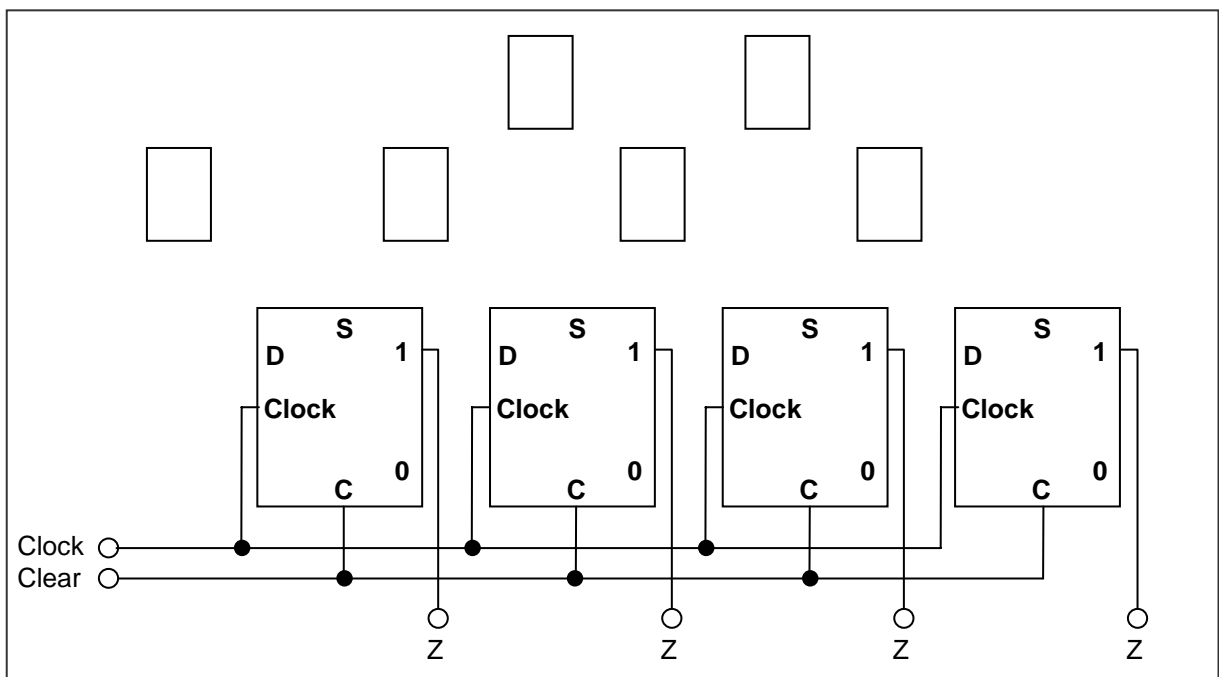
Zyklus	AC (5 Bit Darstellung)	Q (4 Bit Darstellung)	Nächste Operation (Kommentar)
1	00000	1111	Shift
	00001	1110	AC $\leq$ AC - Y
	+11100		
	11101		AC $\leq$ AC + Y
	+00100		
	00001	1110	Shift
2	00011	1100	AC $\leq$ AC - Y
	+11100		
	11111		AC $\leq$ AC + Y
	+00100		
	00011	1100	Shift
3	00111	1000	AC $\leq$ AC - Y
	+11100		
	00011		Q(0) $\leq$ 1
	00011	1001	Shift
4	00111	0010	AC $\leq$ AC - Y
	+11100		
	00011		Q(0) $\leq$ 1
	Rest = 00011	0011 = Quotient	Fertig

**Aufgabe 5 (8 min, 4 + 4 Punkte): Flip-Flops und Register**

- (a) Vervollständigen Sie die nachfolgende Schaltung, so dass ein **4-Bit Parallel-Seriell-Wandler** entsteht, der die seriellen Daten am Ausgang  $Z_3$  ausgibt.



- (b) Vervollständigen Sie die nachfolgende Schaltung, so dass ein synchroner **4-Bit Zähler modulo 16** entsteht.



**Hinweis:** Zeichnen Sie fehlende Signalleitungen (Verbindungen) in die Schaltung ein. Geben Sie außerdem an, welche Gatter-Funktionen die dargestellten Rechtecke darstellen. Füllen Sie dafür das Schaltsymbol entsprechend aus! **Die Gatter haben jeweils maximal 2 Eingänge!** (Sie können die Gatterfunktionen für AND, OR, NAND, NOR, XOR, XNOR und INVERTER benutzen.)

**Aufgabe 6 (8 min, 4 + 4 Punkte): VHDL**

Eine Steuerung eines Hochregalmotors soll durch den Baustein `X_CONTROL` realisiert werden. Das Signal `X_ON` darf nur auf '1' gesetzt werden, wenn die aktuelle X-Adresse `X_IST(0:2)` ungleich der anzufahrenden X-Adresse `X_SOLL(0:2)` ist und wenn der Eingabewert `X_START` gesetzt ist.

- a) Geben Sie die VHDL-Beschreibung der Schnittstelle dieses Steuerbausteins an. Benutzen Sie als Datentyp `std_logic` bzw. `std_logic_vector`.

```
entity X_CONTROL is
  Port (
    X_START : In  std_logic;
    X_IST    : In  std_logic_vector (0 to 2);
    X_SOLL   : In  std_logic_vector (0 to 2);
    X_ON     : Out std_logic;
  );
end X_CONTROL;
```

- b) Vervollständigen Sie die Verhaltensbeschreibung für den Baustein `X_CONTROL`.

```
architecture BEHAVIORAL of X_CONTROL is
begin
  X_CONTROL: process(X_IST, X_SOLL, X_START)
begin
```

```
  if ((X_START = '1') and
      (X_IST /= X_SOLL))
  then
    X_ON <= '1';
  else
    X_ON <= '0';
  end if;
```

```
end process X_CONTROL;
end BEHAVIORAL;
```

```
configuration CFG_X_CONTROL_BEHAVIORAL of X_CONTROL is
  for BEHAVIORAL
  end for;
end CFG_X_CONTROL_BEHAVIORAL;
```