

Abgabetermin: 4. Kalenderwoche (20.01.2014 - 24.01.2014)  
 Maximal **36** Punkte können erreicht werden.

## 1. Aufgabe (5 Punkte)

Gegeben ist folgende Multiplexerschaltung:

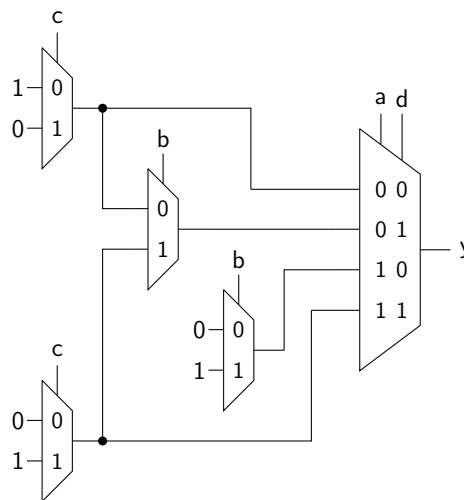


Abbildung 1: Realisierung einer Funktion  $y$  mittels Multiplexern

- Lesen Sie die Funktion  $y$  aus der Multiplexerschaltung ab (**ohne Vereinfachung**).
- Formen Sie die in Aufgabenteil (a) ermittelte Funktion in eine DNF um und übertragen Sie  $y$  in ein KV-Diagramm. Wie lautet die minimal KNF von  $y$ ?

**Hinweis:** Nutzen Sie für Ihre Abgabe die auf ISIS bereitgestellte Lösungsvorlage!

## 2. Aufgabe (13 Punkte)

Gegeben ist folgendes Schaltnetz:

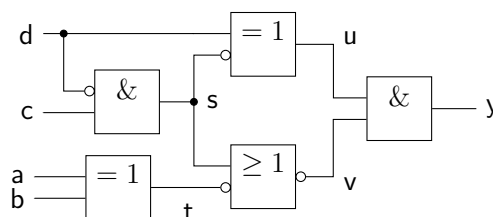


Abbildung 2: Schaltnetz der Funktion  $y$

- (a) Lesen Sie die Signale  $s$ ,  $t$ ,  $u$ ,  $v$  und  $y$  aus dem Schaltnetz ab (**ohne Vereinfachung**). Stellen Sie dabei die Funktionen zunächst nur in Abhängigkeit der jeweiligen Gattereingänge auf. Substituieren Sie anschließend die Zwischensignale soweit, dass jede Formel nur noch von den Schaltnetzeingängen  $a$ ,  $b$ ,  $c$  und  $d$  abhängig sind.
- (b) Minimieren Sie nun die Funktion  $y$  durch Äquivalenzumformungen und algebraische Vereinfachungen zu einer kompakten DNF. Eine Optimierung mittels KV-Diagramm ist unzulässig.
- (c) Vervollständigen Sie folgendes Timingdiagramm:

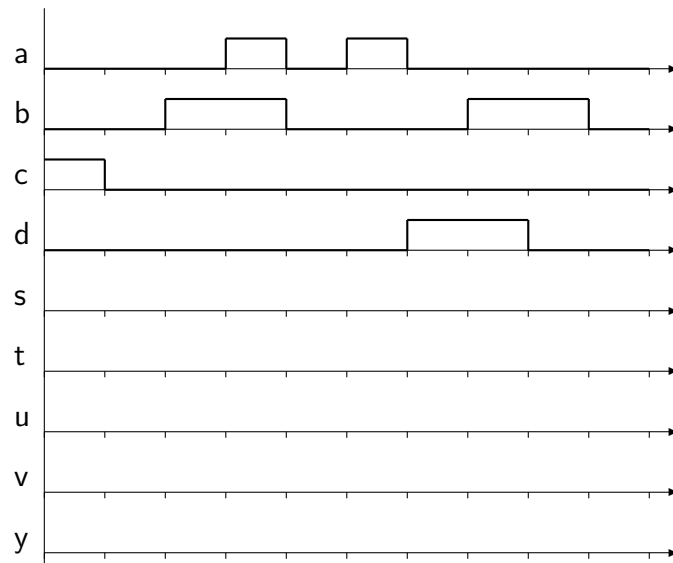


Abbildung 3: Timingdiagramm

**Hinweis:** Nutzen Sie für Ihre Abgabe die auf ISIS bereitgestellte Lösungsvorlage!

### 3. Aufgabe (9 Punkte)

In Abbildung 4 ist eine bereits programmierte PROM-Struktur dargestellt.

- (a) Übertragen Sie die Funktionen  $y_1$  und  $y_2$  direkt aus dem PROM in ein KV-Diagramm.
- (b) Die beiden Funktionen sollen nun möglichst effizient in ein gegebenes PLA eingetragen werden (siehe Abbildung 5). Minimieren Sie  $y_1$  und  $y_2$  entsprechend und geben sie die ermittelten minimalen Funktionen als DNF an.
- (c) Tragen Sie  $y_1$  und  $y_2$  in die PLA-Struktur ein.

**Hinweis:** Nutzen Sie für Ihre Abgabe die auf ISIS bereitgestellte Lösungsvorlage!

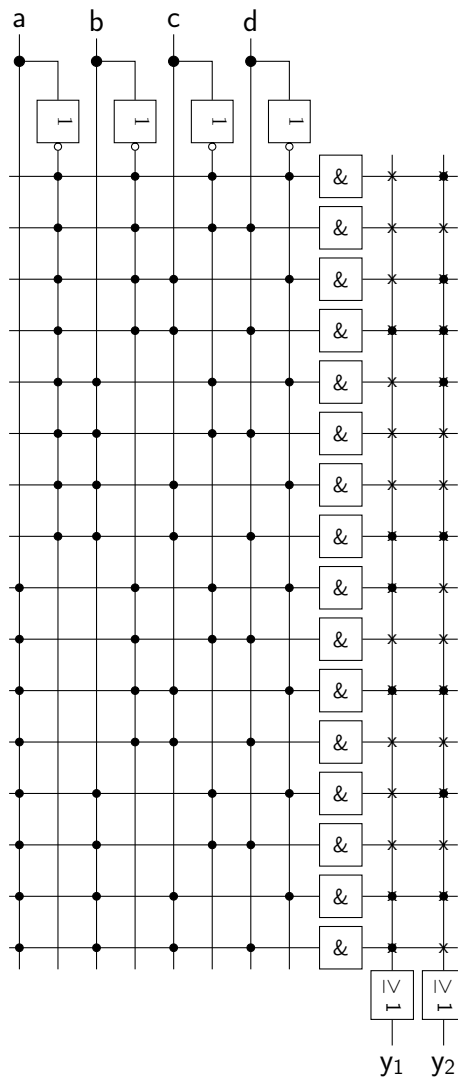


Abbildung 4: Realisierung zweier Funktionen  $y_1$  und  $y_2$  mittels PROM

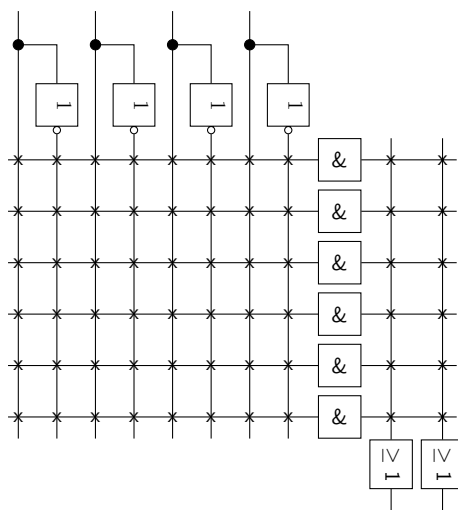


Abbildung 5: PLA-Struktur zur Implementierung von  $y_1$  und  $y_2$

## 4. Aufgabe (9 Punkte)

Folgender VHDL-Quellcode<sup>1</sup> realisiert eine Ihnen unbekannte Funktion:

```
1  entity blackbox is
2      port ( x : in bit_vector (2 downto 0);
3             y : out bit_vector(1 downto 0));
4  end blackbox;
5
6  architecture behave of blackbox is
7      signal k, l, m, t: bit := '0';
8  begin
9
10     k <= '0' when ((x(2) xor x(2)) = '1') else '1';
11     t <= x(1) xor k;
12     y(1) <= m when (x(2) = '0') else
13         l when (x(2) = '1') else '0';
14     l <= t when (not(x(0)) = '1') else not(k);
15     m <= t or (x(0) xor k);
16     y(0) <= (not(x(0)) and ((not(t) xor (not(x(2) nand x(2))))
17         xor k)) or (x(0) and (x(1) xor x(2)));
18 end behave;
```

Quellcode 1: Blackbox (blackbox.vhdl)

- (a) Erstellen Sie eine Testbench, die alle Eingangskombination abdeckt.

**Hinweis:** Geben Sie Ihren ausgedruckten Quellcode für die Testbench ab!

- (b) Übertragen Sie die Simulationsergebnisse in eine Wertetabelle. Welche Funktionalität hat die Blackbox?
- (c) Beschreiben Sie die in Abbildung dargestellte Schnittstelle des Moduls `foo` in VHDL.

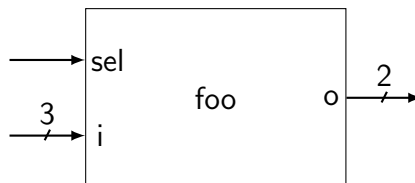


Abbildung 6: Schnittstellenbeschreibung des Moduls `foo`

- (d) Das Modul `foo` realisiert für `sel = 0` das Blackbox-Verhalten. Mit `sel = 1` soll am Ausgang genau das entgegengesetzte Verhalten ablesbar sein. Beschreiben Sie diese Funktionalität auf geeignete Weise in VHDL.

<sup>1</sup>VHDL-Beschreibung der Blackbox `blackbox.vhdl` steht für Sie auf ISIS bereit.