



# Entwurf digitaler Schaltungen

## Große Übung 4

Prof. G. Kemnitz, Dr. C. Gieseemann

Institut für Informatik, Technische Universität Clausthal  
6. Juni 2019

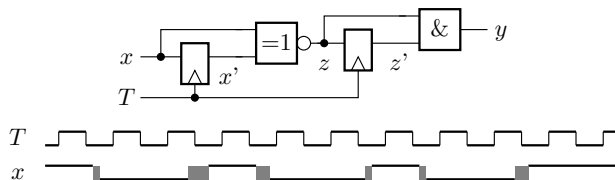
## Aufgabe 4.1: Extraktion des Signalflussplans

```
signal x, tmp, acc, y: std_logic_vector(3 downto 0);
signal op: std_logic_vector(1 downto 0);
signal T: std_logic;
...
process(T)
begin
  if rising_edge(T) then
    case op is
      when "00" => acc <= x;
      when "01" => acc <= acc + tmp;
      when "10" => acc <= acc - tmp;
      when others => null;
    end case;
    tmp <= x;
  end if;
end process;
y <= acc;
```

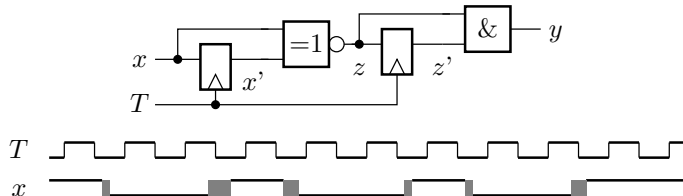
Bilden Sie den Berechnungsfluss durch eine Schaltung aus Registern, Rechenwerken und Multiplexern nach.

## Aufgabe 4.2: Logikfunktion mit Zwischenabtastung

Bestimmen Sie für die gegebene Schaltung und den gegebenen Eingangessignalverlauf die Signalverläufe der Zwischensignale  $x'$ ,  $z$ ,  $z'$  und des Ausgangssignals  $y$ .



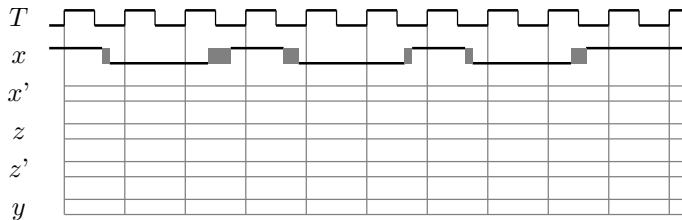
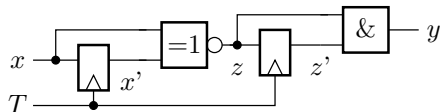
(Vorhalte-, Halte- und Verzögerungszeiten der Gatter und Speicherzellen seien gegenüber der Taktperiode vernachlässigbar.)



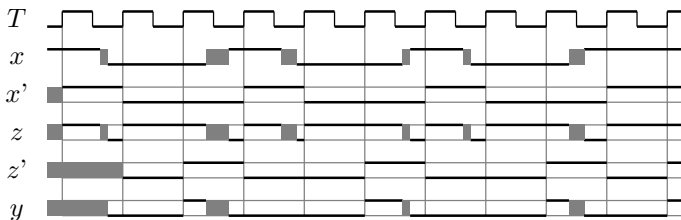
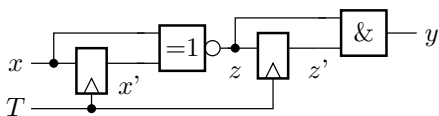
## Hinweise:

- Der Zeitverlauf der unabgetasteten Eingabe wirkt ohne zeitliche Ausrichtung bis auf das Ausgabesignal.
- Wenn ein Eingang der UND-Verknüpfung 0 und der andere ungültig ist, ist die Ausgabe 0.

## Lösung

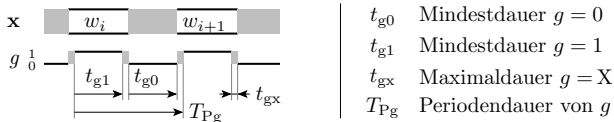


## Zur Kontrolle



## Aufgabe 4.3: Asynchrone parallele Übertragung

Bei einer asynchronen Übertragung seien die mit dem Signal  $x$  übertragenen Datenworte immer dann gültig, wenn das mit übertragene Gültigkeitssignal  $g \neq 0$  ist:



- 1 Mit welcher Taktfrequenz muss die Empfangsschaltung die beiden Signale mindestens abtasten<sup>1</sup>?
- 2 Wie ist das Gültigkeitssignal für die abgetasteten Datenworte zu bilden, damit es für jedes abgetastete gültige Datenwort genau einen Takt lang aktiv ist?

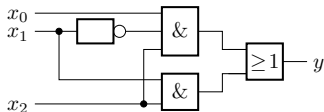
<sup>1</sup>Zeit für  $g = 0$  gleich Zeit für  $g = 1$

## Aufgabe 4.4: Schaltungsumformung

- 1 Suchen Sie nach Vereinfachungsmöglichkeiten:

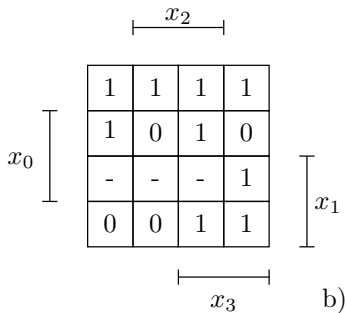
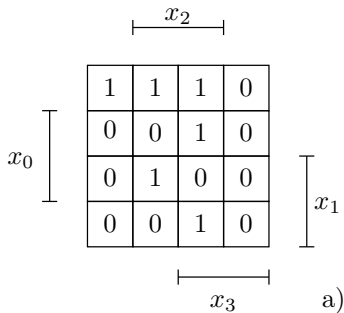
$$x_1x_2(x_2 \vee x_3) \vee x_4(x_1x_2 \vee x_2)$$

- 2 Wandeln Sie die nachfolgende Schaltung in eine Schaltung aus NAND-Gattern und Invertern um.





## Aufgabe 4.5: Vereinfachung mit KV-Diagrammen



## Aufgabe 4.6: Schaltungsminimierung nach Quine und McCluskey

Gegeben ist die Menge Minterme, für die der Funktionswert »1« ist:

$$K \in \{100000, 100100, 101010, 101110, 111110, 110000, 011000, 101011, 101111, 101000, 101001\}$$

- 1 Erstellen Sie die quineschen Tabellen.
- 2 Stellen Sie die Tabelle der Primterme auf.
- 3 Suchen Sie eine minimale Abdeckungsmenge.



## Aufgabe 4.7: VHDL, Wertetabelle, ROBDD, Schaltung

```
signal  a, b, c, y: std_logic;  
...  
y <= a xor b xor c;
```

- 1 Stellen Sie die Wertetabelle für die Schaltung auf.
- 2 Entwickeln Sie aus der Wertetabelle das unreduzierte OBDD für die Abfragereihenfolge a-b-c.
- 3 Entwickeln Sie mit Hilfe der Vereinfachungsregeln – Verschmelzung und Knotenelimination – das zugehörige ROBDD.
- 4 Entwickeln Sie aus dem ROBDD eine optimierte Schaltung.