

Prof. Dr.-Ing. Jürgen Teich  
Lehrstuhl für Informatik 12  
(Hardware-Software-Co-Design)  
Friedrich-Alexander-Universität Erlangen-Nürnberg

## Klausur Grundlagen der Technischen Informatik

27. September 2019

Vorname	
Nachname	
Matrikelnummer	

Aufgabe	1	2	3	4	5	$\Sigma$
Max. Punkte	16	16	16	16	16	80
Erreichte Punkte						
Note						

## Organisatorische Hinweise

Bitte sorgfältig lesen und die Kenntnisnahme durch Unterschrift bestätigen

---

- a) Bitte legen Sie Ihren Studentenausweis bereit.
  - b) Als Hilfsmittel sind nur Schreibmaterialien und ein beidseitig handbeschriebenes DIN A4-Blatt zugelassen.
  - c) Verwenden Sie weder Rot- noch Bleistifte.
  - d) Sie können bei der Aufsicht zusätzliche Bearbeitungsblätter anfordern.
  - e) Unleserliches wird nicht bewertet.
  - f) Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet. Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch.
  - g) Die Bearbeitungszeit beträgt 120 Minuten.
- 

### Erklärung

- a) Im Falle einer während der Prüfung auftretenden Prüfungsunfähigkeit zeige ich dies sofort der Aufsicht an und befolge deren Anweisungen. Ich weiß, dass ich die volle Beweislast trage. Ich lasse mir das Formular des Prüfungsamts, das für diese Fälle vorgesehen ist, aushändigen und verfare nach den dort niedergelegten Richtlinien.
- b) Ich weiß, dass im Falle des Täuschungsversuchs oder der Benutzung unerlaubter Hilfsmittel („Unterschleif“) der Prüfungsausschuss die Entscheidung treffen kann, die betroffene Prüfungsleistung als mit „nicht ausreichend“ bewertet gelten zu lassen.
- c) Ich habe die obigen Hinweise zur Kenntnis genommen.

Erlangen, den .....

Unterschrift

Kopiervorlage: nur für Fachschaften

**Aufgabe 1 (Zahlensysteme)**

(16 Punkte)

a) Wie lautet der Wertebereich einer  $n$ -stelligen Binärzahl  $Z_n$  im Zweierkomplement? (2 Punkte)

b) Der Wert einer binären Festkommazahl  $(d_{n-1}d_{n-2} \dots d_m d_{m-1} \dots d_1 d_0)$  mit  $n - m$  Vorkommastellen und  $m$  Nachkommastellen ist definiert als  $\sum_{i \in \{0, 1, \dots, n-1\}} 2^{i-m} \cdot d_i$ . Wie lautet die Menge aller Werte, die solch eine binäre Festkommazahl annehmen kann? (3 Punkte)

c) Konvertieren Sie die Zahl  $139_{10}$  zur Basis 10 in eine vorzeichenlose Septimärzahl, d. h. in eine Zahl zur Basis 7. (3 Punkte)

Kopiervorlage: nur für Fachschaften

- d) Addieren Sie die beiden BCD-codierten Zahlen  $(0110\ 0100\ 1000)_{\text{BCD}}$  und  $(0010\ 0101\ 1001)_{\text{BCD}}$ . Die Addition ist im Binärformat durchzuführen. Berücksichtigen Sie dabei die eventuell entstehenden Pseudotetraden. (3 Punkte)

- e) Wie viele 2-zu-1-Multiplexer benötigen Sie, um jede beliebige Funktion  $f$  mit  $n$  Booleschen Eingangsvariablen als ein Schaltnetz bestehend nur aus 2-zu-1-Multiplexern sowie den Konstanten **wahr** und **falsch** aufzubauen? Begründen Sie Ihre Antwort. (2 Punkte)

- f) Beantworten Sie folgende Auswahlfragen. Jede richtige Antwort gibt einen Punkt, jede falsche Antwort führt zum Abzug eines Punktes, nicht beantwortete Fragen werden nicht gewertet, weniger als null Punkte sind nicht möglich. (3 Punkte)

1. Eine Gray-Codierung von Zahlen hat die Eigenschaft, dass sich benachbarte Codewörter, d. h. von einer Zahl  $n$  auf ihren Nachfolger  $n + 1$ , nur in einer einzigen binären Ziffer unterscheiden.  wahr  falsch
2. Mit Und- und Antivalenz-Gattern sowie den Konstanten **wahr** und **falsch** lässt sich jede beliebige Schaltfunktion darstellen.  wahr  falsch
3. Bei der Umwandlung von Ganzzahlen im Binärsystem in Zahlen im Ternärsystem kann es zu Rundungsfehlern kommen.  wahr  falsch

**Aufgabe 2 (Minimierung von Schaltfunktionen)**

(16 Punkte)

- a) Ermitteln Sie alle Primimplikanten einer Schaltfunktion  $f_1(a, b, c, d, e)$ , die durch folgendes Symmetriediagramm spezifiziert ist. Bestimmen Sie alle Kernimplikanten. (4 Punkte)

		$a$				$e$		
		$a$		$a$				
$b$	1	0	0	1	1	0	–	–
	<sub>0</sub>	<sub>1</sub>	<sub>5</sub>	<sub>4</sub>	<sub>24</sub>	<sub>25</sub>	<sub>21</sub>	<sub>20</sub>
	0	1	1	0	0	1	1	0
	<sub>2</sub>	<sub>3</sub>	<sub>7</sub>	<sub>6</sub>	<sub>26</sub>	<sub>27</sub>	<sub>23</sub>	<sub>22</sub>
	0	1	1	0	0	0	0	0
	<sub>12</sub>	<sub>13</sub>	<sub>17</sub>	<sub>16</sub>	<sub>36</sub>	<sub>37</sub>	<sub>33</sub>	<sub>32</sub>
	–	0	0	0	0	0	–	1
	<sub>10</sub>	<sub>11</sub>	<sub>15</sub>	<sub>14</sub>	<sub>34</sub>	<sub>35</sub>	<sub>31</sub>	<sub>30</sub>
		$c$						$d$

- b) Ermitteln Sie mit dem Nelson-Verfahren alle Primimplikanten einer Schaltfunktion  $f_2(a, b, c, d)$ , gegeben durch folgendes Symmetriediagramm. (4 Punkte)

		$a$			
$b$	0	0	1	–	
	<sub>0</sub>	<sub>1</sub>	<sub>5</sub>	<sub>4</sub>	
	1	–	1	0	
	<sub>2</sub>	<sub>3</sub>	<sub>7</sub>	<sub>6</sub>	
	0	1	–	1	
	<sub>12</sub>	<sub>13</sub>	<sub>17</sub>	<sub>16</sub>	
	0	0	1	0	
	<sub>10</sub>	<sub>11</sub>	<sub>15</sub>	<sub>14</sub>	
		$c$			

- c) Sei folgende Überdeckungstabelle der Primimplikanten einer Schaltfunktion  $f_3(a, b, c, d, e)$  gegeben.

k	$p_i$	j				$c_i$
		8	10	12	30	
0	A	×	×			3
1	B	×	×	×		3
2	C			×	×	3
3	D		×		×	2
4	E				×	2

1. Stellen Sie den Petrick-Ausdruck für  $f_3$  auf. (1 Punkt)

2. Bestimmen Sie mit dem Petrick-Verfahren alle kostenminimalen DMFs von  $f_3$ . (3 Punkte)

- d) Erläutern Sie die Vor- und Nachteile einer CMOS-Schaltung gegenüber einer PMOS-Schaltung. (2 Punkte)

- e) Gegeben Sie eine Schaltfunktion  $f_4(a, b, c, d) = a \oplus b + c + \overline{d}$ ;

1. Bestimmen Sie den Ausdruck des Pull-Up-Netzwerks einer CMOS-Schaltung von  $f_4$ . (1 Punkt)

2. Bestimmen Sie den Ausdruck des Pull-Down-Netzwerks einer CMOS-Schaltung von  $f_4$ . (1 Punkt)

**Aufgabe 3 (Automaten)**

(16 Punkte)

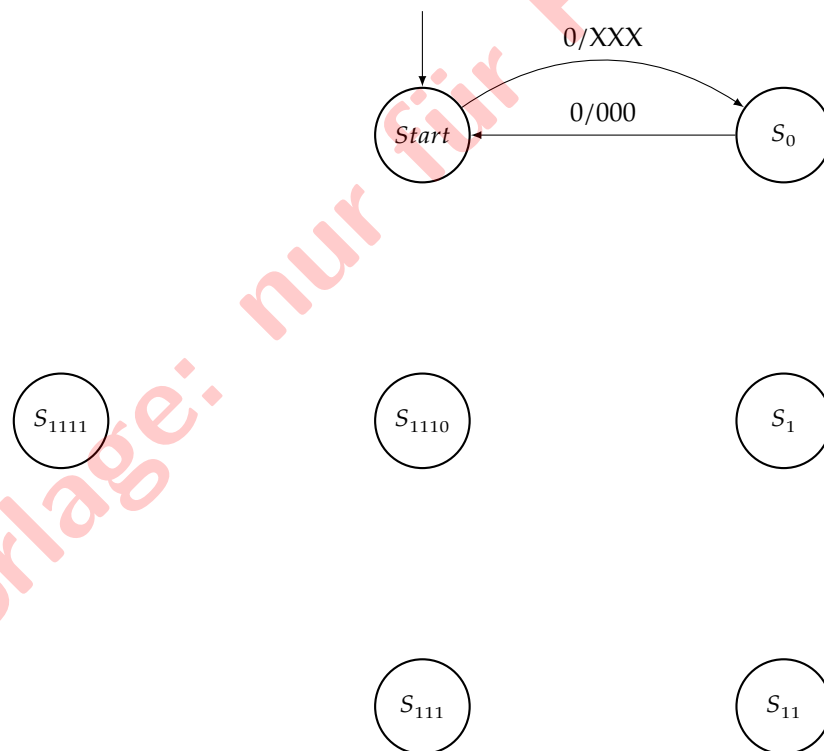
- a) Im Folgenden soll ein Automat entworfen werden, der zur Decodierung von Huffman-codierten Wörtern benutzt werden kann. Der Wortschatz besteht aus acht Codewörtern, die folgendermaßen codiert wurden:

Originalwort	000	001	010	011	100	101	110	111
Huffman-Codierung	00	01	10	110	11100	11101	11110	11111

Der Decodierer erhält als Eingabe ein Huffman-codiertes Wort, wertet dies bitweise beginnend beim hochwertigsten Bit aus und gibt das decodierte Originalwort aus. Dies kann als Mealy-Automat beschrieben werden, indem die Wurzel als Zustand *Start* repräsentiert wird und zusätzlich für jeden inneren Knoten des Huffman-Baums ein Zustand  $S_{id}$  eingeführt wird. Hierbei steht *id* für die bislang bearbeiteten Bits auf dem Pfad von der Wurzel des Huffman-Baums zum jeweiligen inneren Knoten. Jeder Zustandsübergang lässt sich als Paar *Eingabebit/Originalwort* ausdrücken, wobei für unfertige Codewörter XXX ausgegeben wird. Jede Decodierung startet und endet im Startzustand *Start*.

*Hinweis: Die Decodierung des Originalworts 000 für die Eingabe 00 ist bereits in den Automaten eingezeichnet.*

1. Vervollständigen Sie den obig beschriebenen Zustandsautomaten. (3 Punkte)



2. Begründen Sie, ob der obige Mealy-Automat in einen Moore-Automaten umgewandelt werden kann, und geben Sie, falls möglich, die dazu nötigen Maßnahmen an. (1 Punkt)

3. Wie viele Zustände weist ein nach obigem Schema erstellter Mealy-Automat für folgende Huffman-Codierung auf? (1 Punkt)

Originalwort	A	B	C
Huffman-Codierung	11	0	10



4. Für einen weiteren Satz an Codewörtern ergibt sich die folgende Automatentafel. Vervollständigen Sie die Automatentafel unter Verwendung von taktflankengesteuerten D-, JK- bzw. T-Flipflops. (3 Punkte)

Zustandsname	Aktueller Zustand			Eingabe $i$	Nachfolgezustand			Ansteuerfunktion			
	$q_2$	$q_1$	$q_0$		$q'_2$	$q'_1$	$q'_0$	$D_2$	$J_1$	$K_1$	$T_0$
<i>Start</i>	0	0	0	0	0	0	1				
<i>Start</i>	0	0	0	1	0	1	0				
$S_0$	0	0	1	0	0	0	0				
$S_0$	0	0	1	1	0	0	0				
$S_1$	0	1	0	0	0	1	1				
$S_1$	0	1	0	1	1	0	0				
$S_{10}$	0	1	1	0	0	0	0				
$S_{10}$	0	1	1	1	0	0	0				
$S_{11}$	1	0	0	0	1	0	1				
$S_{11}$	1	0	0	1	0	0	0				
$S_{110}$	1	0	1	0	0	0	0				
$S_{110}$	1	0	1	1	1	1	0				
$S_{1101}$	1	1	0	0	0	0	0				
$S_{1101}$	1	1	0	1	0	0	0				

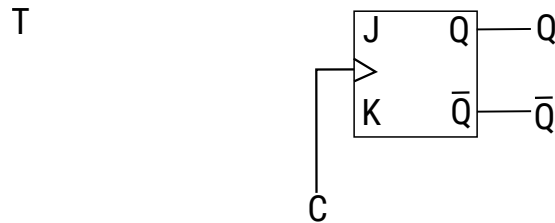
5. Zeichnen Sie ein Schaltwerk ohne Ausgabefunktion für den in Teilaufgabe 1 realisierten Automaten. Verwenden Sie dazu folgende minimierte KMF der Ansteuerfunktion  $J_1$ :

$$KMF_{J_1} = (\bar{i} + q_1)\bar{q}_0(q_1 + q_2 + i)$$

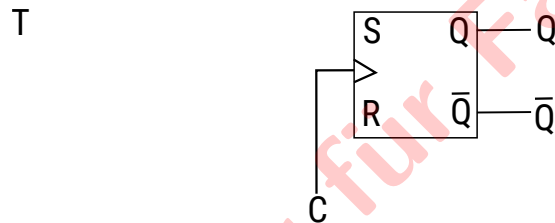
Gehen Sie davon aus, dass die benötigten Ansteuerfunktionen  $D_2$ ,  $K_1$  und  $T_0$  als Eingangssignale zur Verfügung stehen. (3 Punkte)

- b) Transformieren Sie das folgende Active-High RS- und JK-Flipflop jeweils in ein T-Flipflop, indem Sie das Eingangssignal T geeignet verbinden und bei Bedarf Logikgatter hinzufügen. (3 Punkte)

1.



2.



- c) Erläutern Sie den entscheidenden Nachteil eines RS-Flipflops, der bei JK-Flipflops nicht mehr auftritt. (1 Punkt)

- d) Die Zustände eines Automaten sollen mithilfe von Flipflops modelliert werden. Wie viele Flipflops werden dazu benötigt, wenn der Automat 34 Zustände aufweist? (1 Punkt)

**Aufgabe 4 (Codierung und Rechnerarithmetik)**

(16 Punkte)

a) In einem Quantencomputer werden (7,4) Hamming-Codes eingesetzt, um die Quanteninformation vor Rauschen zu schützen. (7,4) Hamming-Codes bestehen aus 4 Informationsstellen  $x_3x_2x_1x_0$  und 3 Prüfstellen  $y_2y_1y_0$ .

1. Welche Arten von Fehlern können mit (7,4) Hamming-Codes erkannt beziehungsweise korrigiert werden? (2 Punkte)

2. Geben Sie die Konstruktionsvorschrift der 3 Prüfstellen für (7,4) Hamming-Codes an. Ergänzen Sie anschließend die Prüfstellen zur Codierung der gegebenen Informationsstellen. (2 Punkte)

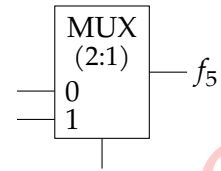
Codewort	Informationsstellen				Prüfstellen		
	$x_3$	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$
1. CW	0	0	0	0			
2. CW	0	1	1	0			

b) Folgende unvollständige Wahrheitstabelle zeigt die Umwandlung einer 3-Bit-Binärzahl  $b_2b_1b_0$  in 3-Bit-Graycode  $g_2g_1g_0$ .

1. Vervollständigen Sie zunächst die Tabelle. (2 Punkte)

$b_2$	$b_1$	$b_0$	$g_2$	$g_1$	$g_0$
0	0	0	1	1	1
0	0	1			0
0	1	0			0
0	1	1	0	1	1
1	0	0			1
1	0	1			0
1	1	0			0
1	1	1			1

2. Zeichnen Sie das *Ordered Binary Decision Diagram* der Schaltfunktion  $f_5(b_0, b_1, b_2)$  zur Erzeugung von  $g_0$ , für die Variablenordnung  $b_0, b_1$  und  $b_2$ . Zeigen Sie anschließend am nebenstehenden 2:1-Multiplexer die Realisierung der Funktion  $f_5$  durch passendes Anlegen der Signale  $\overline{b_0}, b_0, \overline{b_1}, b_1, \overline{b_2}$  und  $b_2$  an den Eingängen. (2 Punkte)



- c) Im Folgenden soll ein Schaltnetz für einen 1-Bit Komparator mittels NOR-Gattern realisiert werden.

1. Entwerfen Sie unter ausschließlicher Verwendung von NOR-Gattern je ein Schaltnetz für die im Folgenden gegebenen Funktionen,  $f_{<}(x,y)$ ,  $f_{>}(x,y)$  und  $f_{=}(x,y)$ , für 1-Bit Zahlen  $x$  und  $y$ . Hierfür stehen Ihnen ausschließlich die nicht-negierten Signale  $x$  sowie  $y$  zur Verfügung. (4 Punkte)

$$f_{<}(x,y) = \begin{cases} 1, & \text{wenn } x < y \\ 0, & \text{sonst} \end{cases}$$

$$f_{>}(x,y) = \begin{cases} 1, & \text{wenn } x > y \\ 0, & \text{sonst} \end{cases}$$

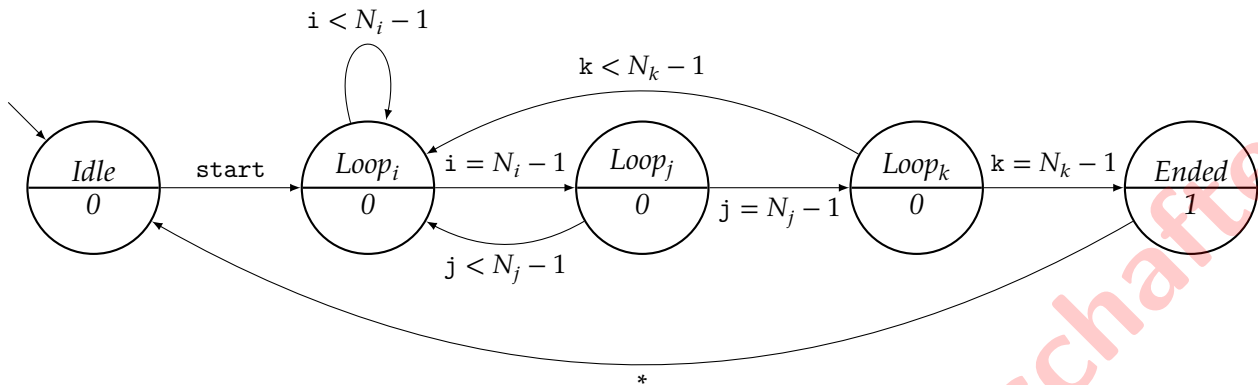
$$f_{=}(x,y) = \begin{cases} 1, & \text{wenn } x = y \\ 0, & \text{sonst} \end{cases}$$



## Aufgabe 5 (VHDL)

(16 Punkte)

Folgender Automatengraph spezifiziert einen Controller für eine dreidimensionale Schleife.

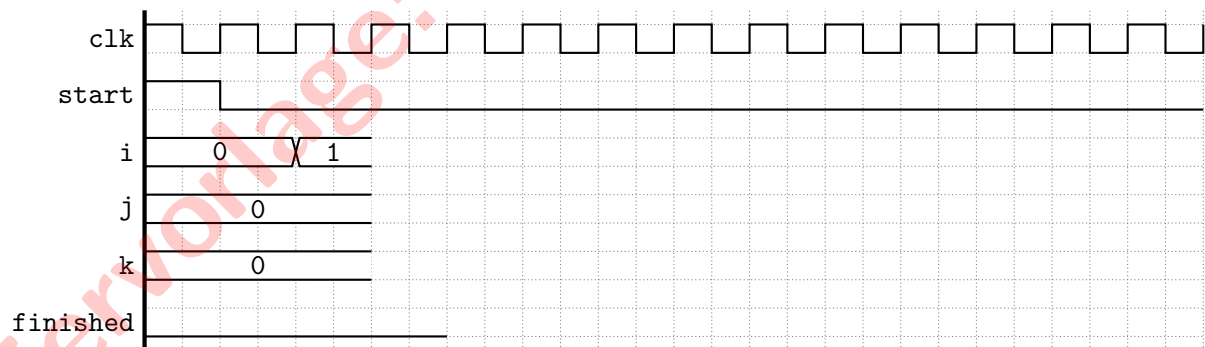


Der Controller hat ein Eingangssignal *start*, ein Ausgangssignal *finished* und führt Buch über drei Zähler, *i*, *j* und *k*, die er nach Erhalt des Signals *start* wie in einer dreifach verschachtelten for-Schleife lexikographisch hochzählt. Die innerste Schleife entspricht *i*, die mittlere *j* und die äußerste *k*. Die Anzahl der Iterationen ist durch  $N_i$ ,  $N_j$  und  $N_k$  gegeben; es gelte  $N_i, N_j, N_k \geq 1$ . Sobald alle Iterationen durchlaufen sind, wird *finished* auf 1 gesetzt.

Die Zähler werden wie folgt aktualisiert: Im Zustand *Idle* werden alle Zähler auf 0 gesetzt. Im Zustand *Loop<sub>i</sub>* wird *i* um 1 erhöht. Im Zustand *Loop<sub>j</sub>* wird *i* auf 0 zurückgesetzt und *j* um 1 erhöht. Im Zustand *Loop<sub>k</sub>* werden *i* und *j* auf 0 zurückgesetzt und *k* um 1 erhöht.

Implementieren Sie den spezifizierten Automaten als *synchrone* Schaltung, die sich *asynchron* zurücksetzen lässt.

- a) Sei  $N_i = 3$ ,  $N_j = 2$  und  $N_k = 1$ . Vervollständigen Sie folgendes Wellenformdiagramm. Alle drei Zähler sind als *signal* implementiert. Der Zustand zu Beginn ist *Idle*. (4 Punkte)



- b) Vervollständigen Sie die folgende *entity*-Deklaration des Schleifencontrollers. (2 Punkte)

```
entity loop_controller is
```

```
end loop_controller;
```

c) Vervollständigen Sie den Rumpf der folgenden architecture.

(9 Punkte)

```
architecture behavioral of loop_controller is
  type states is (idle, loop_i, loop_j, loop_k, ended);
  signal state : states;
  signal i : integer;
  signal j : integer;
  signal k : integer;
  constant N_i : integer := 16;
  constant N_j : integer := 8;
  constant N_k : integer := 4;
begin
```

Kopiervorlage: nur für Fachschaften

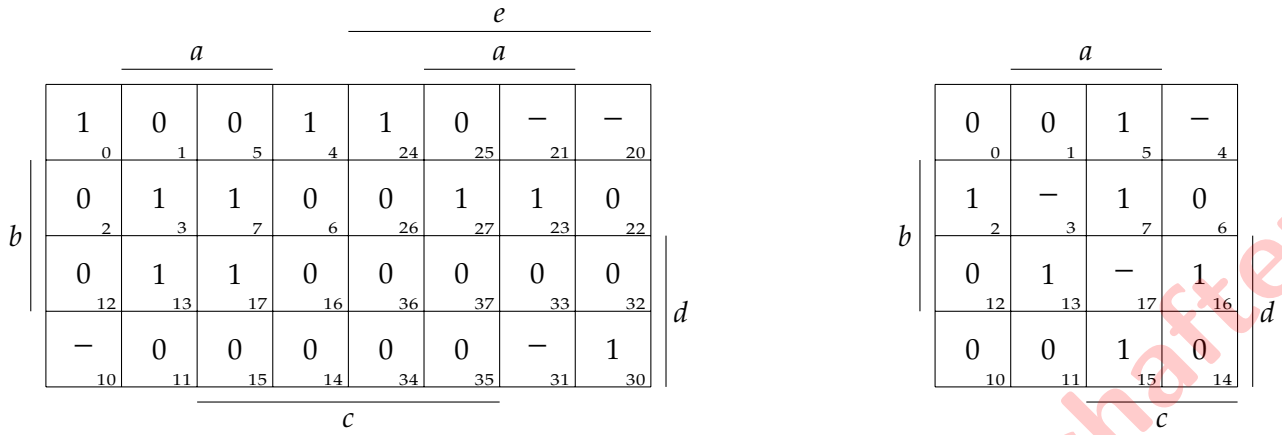
Kopiervorlage: nur für Fachschaften

**end architecture ;**

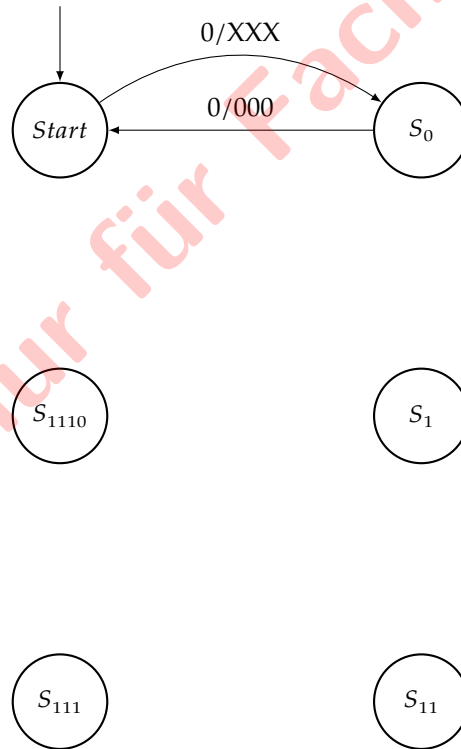
- d) Erläutern Sie, wieso es bei der Entwicklung von Hardware im Vergleich zu Software besonders wichtig ist, deren korrekte Funktion zu verifizieren. (1 Punkt)



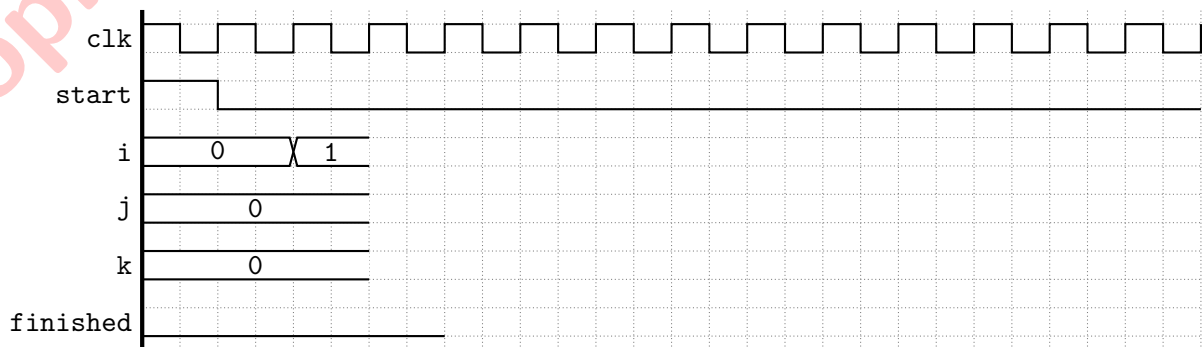
**Aufgabe 2**



**Aufgabe 3**



**Aufgabe 5**



Kopiervorlage: nur für Fachschaften