Universität Paderborn
Institut *Elektrotechnik und Informationstechnik*
Fachgebiet *Datentechnik*
Prof. Sybille Hellebrand

Klausur

Digitaltechnik / Grundlagen Technische Informatik

24. März 2016

Punkteverteilung						
Aufgabe	1	2	3	4	5	Σ
maximale Punkte	20	15	15	19	21	90
erreichte Punkte						

Note:	

Aufkleber

Name:	
Matrikelnummer:	
Studienrichtung:	

Hinweise:
Für die Lösung der Klausuraufgaben sind ausschließlich die Aufgabenblätter zu verwenden. Lösungsangaben außerhalb der Aufgabenblätter („Schmierzettel“, etc.) werden bei der Bewertung nicht berücksichtigt!

Beschriften Sie jede Doppelseite mit Ihrer Matrikelnummer!

Mit Bleistift oder der Korrekturfarbe rot angefertigte Lösungen werden nicht bewertet!

Die Verwendung von „Tipp-Ex“ oder „Tintenkiller“ ist untersagt.

Es ist ein handgeschriebener DIN-A4 Zettel als Hilfsmittel zugelassen!

Es sind keine weiteren Hilfsmittel zugelassen!

Aufgabe 1: (VHDL) 20 Punkte

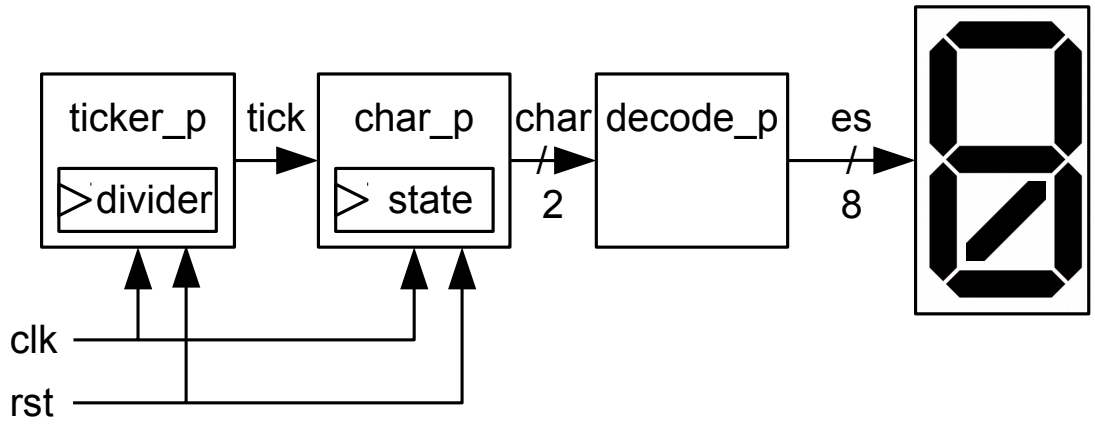


Abbildung 1: Schaltungsübersicht

Auf *einem* 8-Segment Display sollen nacheinander die Buchstaben „V“, „H“, „d“und „L“ in dieser Reihenfolge angezeigt werden. Der Buchstabe soll jede Sekunde wechseln, wobei das „V“für zwei Sekunden zu sehen sein soll. Ihre Aufgabe als Hardware-IngenieurIn ist es nun die in Abbildung 1 gezeigten Komponenten in Prozessen zu implementieren und so den Buchstabenwechsel umzusetzen.

a) Beantworten Sie zunächst folgende Eingangsfragen.

(3 Punkte)

1) Der Eingangstakt `clk` der Schaltung hat eine Frequenz von 2 MHz. Bestimmen Sie den Wert des Teilers `divider`, der zur Taktteilung auf 1 Hz nötig ist (Ergebnis reicht).

divider: _____

2) Welche VHDL-Syntax wird benutzt um eine steigende Taktflanke zu beschreiben? Schreiben Sie ein Beispiel mit dem Signal `takt`.

_____

3) Warum braucht der Prozess `decode_p` die Signale `clk` und `rst` nicht?

_____

b) Vervollständigen Sie den Prozess `ticker_p` der das Taktsignal `clk` nutzt, um das Signal `tick` einmal pro Sekunde für einen Takt auf '1' zu setzen. Pro Zeile kann auch mehr als eine Anweisung eingetragen werden.

(6 Punkte)

```
[...]
signal clk, rst, tick: std_logic;
[...]

ticker_p: process(_____________________________) is
  begin
    if rst = '1' then
      divider <= _____________________;     tick    <= '0';

    elsif ____________________________________ then

      if divider = ___________________________ then

        ____________________________________________________________
      else

        ____________________________________________________________
      end if;
    end if;
end process;
```

———(Ersatz, ungültige Lösung streichen!)———

```
ticker_p: process(_____________________________) is
  begin
    if rst = '1' then
      divider <= _____________________;     tick    <= '0';

    elsif ____________________________________ then

      if divider = ___________________________ then

        ____________________________________________________________
      else

        ____________________________________________________________
      end if;
    end if;
end process;
```

c) Der Prozess `char_p` implementiert eine Zustandsmaschine, welche die Buchstabenfolge kontrolliert. Das Ausgabesignal `char` soll b'00' für „V“ sein, b'01' für „H“, b'10' für „D“ und b'11' für „L“. Pro Zeile kann auch mehr als eine Anweisung eingetragen werden. (5 Punkte)

```
[...]
signal clk, rst, tick: std_logic;  signal char : std_logic_vector(1 downto 0);
[...]

char_p: process(_____) is
  type state_t is (_____);
  signal state: state_t;
begin
  if rst = '1' then
    state <= _____; char <= "00";

  elsif _____ and tick ='1' then
    case state is
      when _____ => _____

      when _____ => _____

      when _____ => _____

      when _____ => _____

      when _____ => _____
    end case;
  end if;
end process;
```

———(Ersatz, ungültige Lösung streichen!)———

```
char_p: process(_____) is
  type state_t is (_____);
  signal state: state_t;
begin
  if rst = '1' then
    state <= _____; char <= "00";

  elsif _____ and tick ='1' then
    case state is
      when _____ => _____

      when _____ => _____

      when _____ => _____

      when _____ => _____

      when _____ => _____
    end case;
  end if;
end process;
```

d) Der Prozess `decode_p` übersetzt vom 2-Bit Vektor `char`, der den Buchstaben kodiert, auf den 8-Bit Vektor `es`, der die Darstellung der Buchstaben kodiert.

Der Signalvektor `es` ist mit dem 8-Segment-Display verbunden. Abbildung 2 zeigt ein 8-Segment-Display und wie die Buchstaben „V“, „H“, „d“und „L“dargestellt werden sollen. Die Abbildung zeigt außerdem welcher Index mit welchem Segment verbunden ist. Bei einer ’1’ leuchtet das Segment auf, bei einer ’0’ bleibt es aus.

Vervollständigen Sie den Prozess. Pro Zeile kann auch mehr als eine Anweisung eingetragen werden.

(6 Punkte)

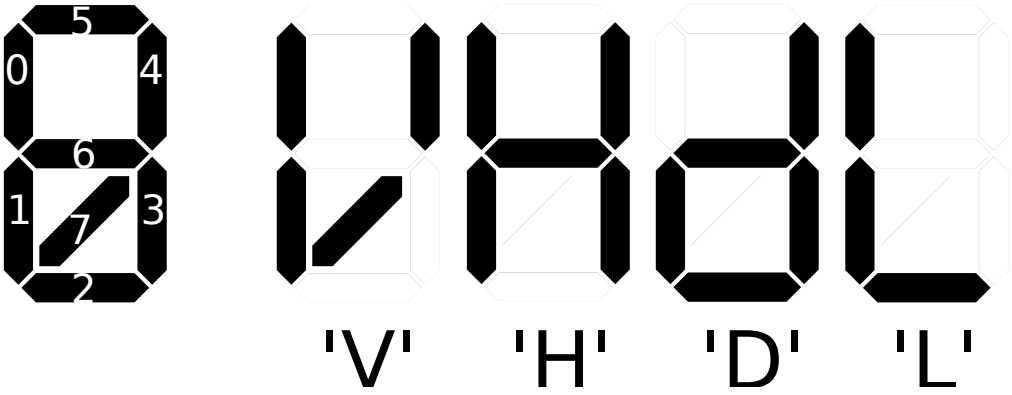


Abbildung 2: 8-Segment Display mit Darstellung der Buchstaben VHdL. Die Nummern geben den Index im `es`-Array an.

```
[...]
signal char : std_logic_vector(1 downto 0);
signal es   : std_logic_vector(7 downto 0);
[...]

decode_p: process(_____) is
begin
  _____

  _____

  _____

  _____

  _____

  _____
end process;
```

———(Ersatz, ungültige Lösung streichen!)———

```
decode_p: process(_____) is
begin
  _____

  _____

  _____

  _____

  _____

  _____
end process;
```

Aufgabe 2: (Logikoptimierung) 15 Punkte

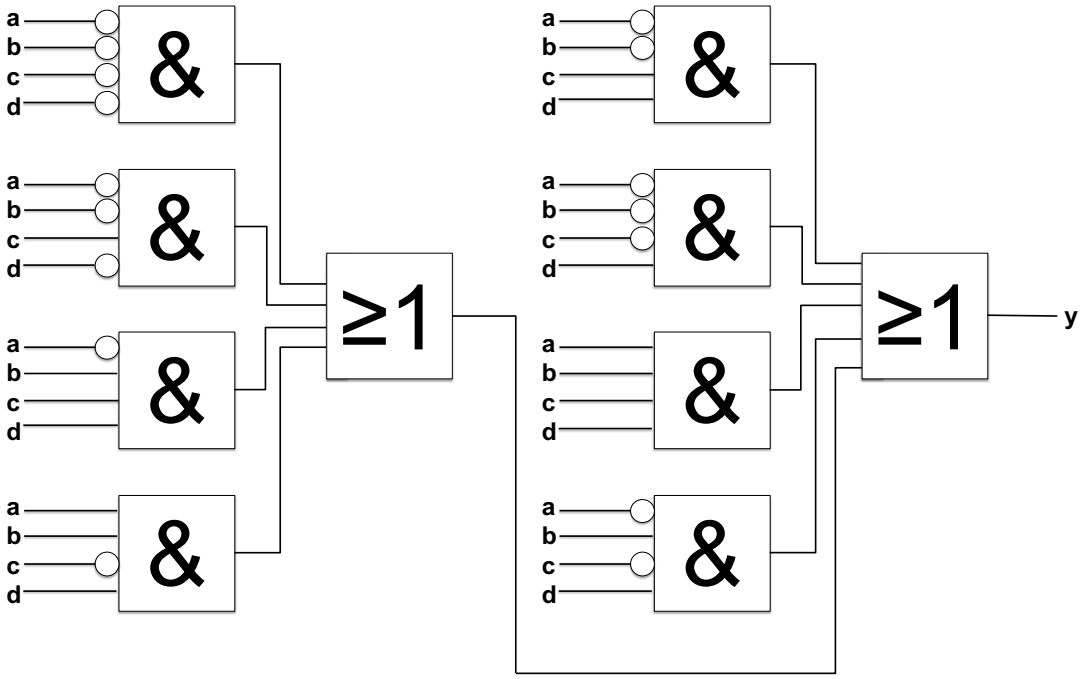


Abbildung 3: Logische Schaltung

Gegeben sei die Schaltung aus Abbildung 3 mit der Schaltungsfunktion $y(a,b,c,d)$. Ihr Ziel ist es, diese Schaltung mit Hilfe eines KV-Diagrammes zu optimieren.

a) Bestimmen Sie die Einsstellenmenge $\mathcal{E}$ der Schaltungsfunktion y und geben Sie die Produktterme als Codewörter *abcd* aus $\{0,1\}^4$ an.

(2 Punkte)

$\mathcal{E} = \{$_____

_____

_____

_____$\}$

b) Finden Sie für die Schaltung aus Abbildung 3 mit Hilfe des KV- Diagramms die Menge der Primimplikanten $\mathcal{P}$ für y und geben Sie diese an. Ermitteln Sie ferner die Menge der essentiellen Primimplikanten und bestimmen Sie anschließend die disjunktive Minimalform $y(a,b,c,d)$.

(7 Punkte)

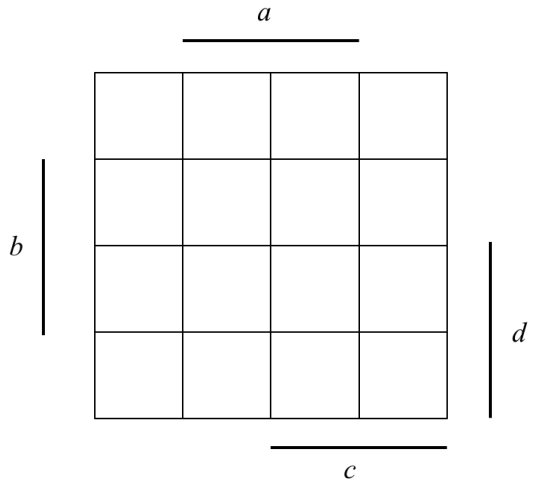


Abbildung 4: KV- Diagramm

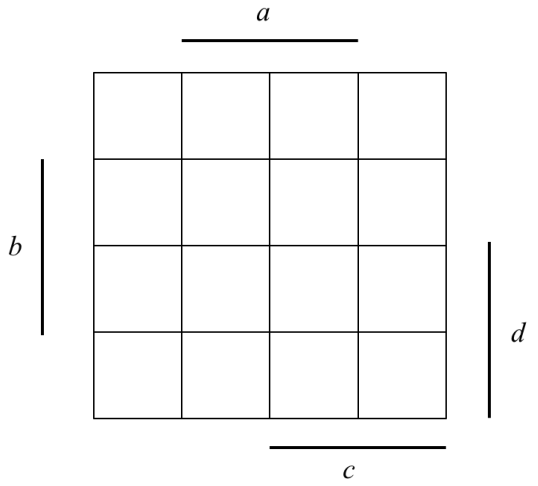


Abbildung 5: Ersatz KV- Diagramm, **ungültige Lösung streichen!**

$\mathcal{P} = \{$_____$\}$

$\mathcal{P}_\mathrm{e} = \{$_____$\}$

Die disjunktive Minimalform lautet

$y(a,b,c,d) =$ _____

c) Gegeben sei die Primimplikantentafel aus Tabelle 1. Berechnen Sie die minimale Summe von Primimplikanten durch Anwendung des Quine-McCluskey Verfahrens (Phase II) und geben Sie diese an. Füllen Sie nach jedem Schritt des Verfahrens eine neue Überdeckungstabelle aus und begründen Sie Ihre Entscheidung kurz. Falls nötig, wählen Sie den Primimplikanten mit den geringsten Kosten als Auswahlheuristik. Gehen Sie davon aus, dass die Implementierungskosten der Primimplikanten durch den Index am Primimplikanten gegeben ist.

(6 Punkte)

	x_0	x_1	x_2	x_3	x_4	x_5
P_0	**X**		**X**			
P_1	**X**		**X**	**X**	**X**	
P_2				**X**		**X**
P_3					**X**	**X**
P_4	**X**					**X**
P_5		**X**				**X**

Tabelle 1: Überdeckungstabelle

Begründung: _____

	_____	_____	_____	_____	_____	_____
_____						
_____						
_____						
_____						
_____						
_____						

Tabelle 2: Überdeckungstabelle

Begründung: _____

	_____	_____	_____	_____	_____	_____
_____						
_____						
_____						
_____						
_____						
_____						

Tabelle 3: Überdeckungstabelle

Begründung: _____

	_____	_____	_____	_____	_____	_____
_____						
_____						
_____						
_____						
_____						
_____						

Tabelle 4: Überdeckungstabelle

Begründung: _____

	_____	_____	_____	_____	_____	_____
_____						
_____						
_____						
_____						
_____						
_____						

Tabelle 5: Überdeckungstabelle

Begründung: _____

Die Lösung lautet {_____}.

	_____	_____	_____	_____	_____	_____
_____						
_____						
_____						
_____						
_____						
_____						

Tabelle 6: Ersatz Überdeckungstabelle, **ungültige Lösung streichen!**

Begründung: _____

	_____	_____	_____	_____	_____	_____
_____						
_____						
_____						
_____						
_____						
_____						

Tabelle 7: Ersatz Überdeckungstabelle, **ungültige Lösung streichen!**

Begründung: _____

Aufgabe 3: (Automaten) 15 Punkte

a) Führen Sie auf die unten angegebene Automatentafel des Mealy-Automaten $M = (\{a,b\},\{0,1\},\{z_0,\ldots,z_7\},\delta,\lambda)$ eine Reduktion nach dem Ginsburg/Huffman-Verfahren durch. Schreiben Sie in die Tabellen das Zwischenergebnis nach jedem Schritt des Verfahrens. Geben Sie anschließend die Automatentafel des reduzierten Automaten an (Ersatztabellen finden Sie auf Seite 18).

(9 Punkte)

Automatentafel des Automaten M:

δ/λ	a	b
z_0	$z_7/1$	$z_1/0$
z_1	$z_7/1$	$z_0/0$
z_2	$z_5/1$	$z_1/0$
z_3	$z_6/1$	$z_4/1$
z_4	$z_6/1$	$z_3/1$
z_5	$z_4/1$	$z_2/1$
z_6	$z_3/1$	$z_2/1$
z_7	$z_5/1$	$z_2/1$

δ		

δ		

δ		

δ		

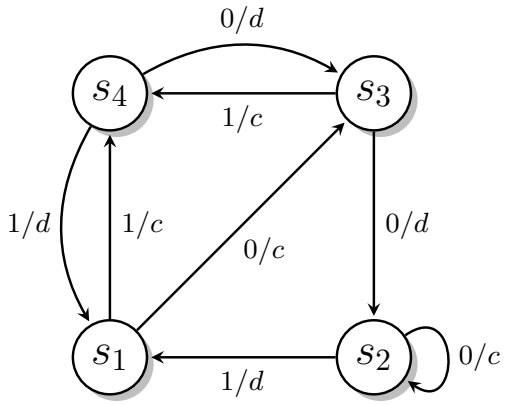
Automatentafel des reduzierten Automaten:

δ/λ		

b) Zeichnen Sie den Graphen des reduzierten Automaten.

(2 Punkte)

c) Gegeben sei die folgende graphische Darstellung des Mealy-Automaten D.



Geben Sie die Automatentafel eines zu D äquivalenten Moore-Automaten $M = (I,O,S,\delta,\mu)$ an. Führen Sie ausschließlich die kanonische Umwandlung durch. Nehmen Sie weder an D noch an M Optimierungen vor.

(4 Punkte)

	μ	δ	

Ersatztabellen für Aufgabenteil a), **ungültige Lösungen streichen!**

Aufgabe 4: (Division mit Rückspeichern) 19 Punkte

Abbildung 6 zeigt den Datenpfad eines sequentiellen Dividierers der mit Rückspeichern arbeitet um zwei n-bit Zahlen zu dividieren. Über den INBUS werden nacheinander die beiden Zahlen a und b geschickt und in das jeweilige Register geschrieben. Der Inhalt von Register A wird dann ganzzahlig durch die Zahl in Register B geteilt, das ganzzahlige Ergebnis a/b steht anschließend in Register A, während Register P den Rest der Division enthält. Beide Registerinhalte (A und P) werden nach der Berechnung nacheinander auf den OUTBUS gelegt.

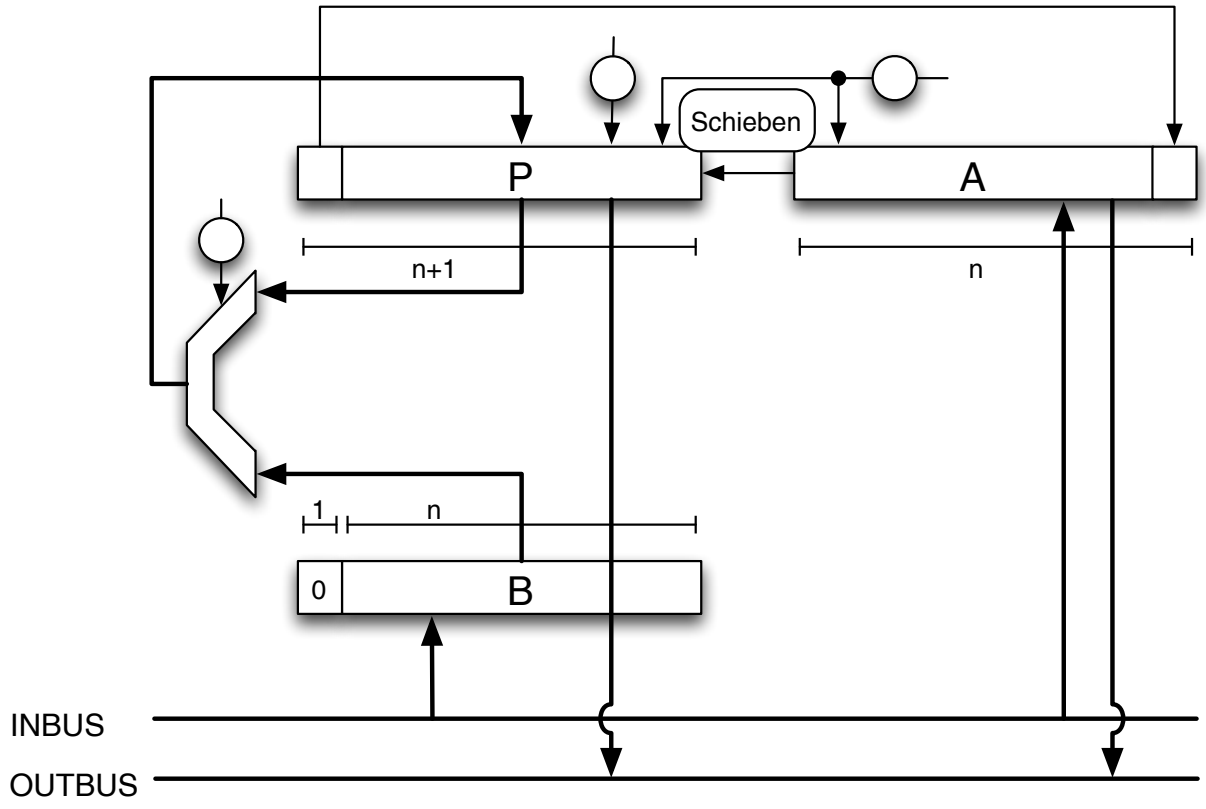


Abbildung 6: Datenpfad eines Dividierers mit einigen Kontrollpunkten

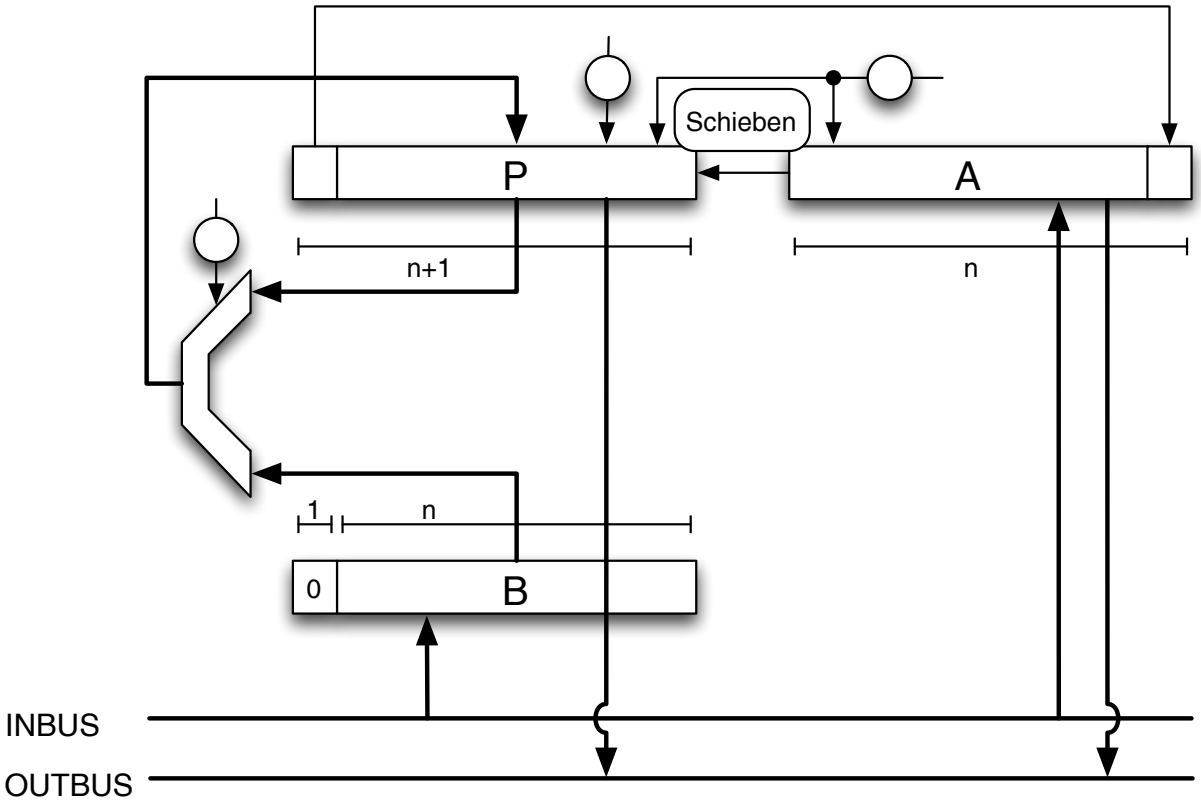


Abbildung 7: Ersatz Dividierer-Datenpfad, **ungültige Lösung streichen!**

a) Ergänzen (und benennen) Sie alle notwendigen Kontrollpunkte im Datenpfad (Abbildung 6 oder 7). Beschreiben Sie die Funktion jedes Kontrollpunktes kurz in der folgenden Tabelle.

(8 Punkte)

Kontrollpunkte:

c0: _____

c1: _____

c2: _____

c3: _____

c4: _____

... _____

... _____

... _____

... _____

... _____

... _____

... _____

Ersatztabelle für Kontrollpunkte, **ungültige Lösung streichen!**

c0: _____

c1: _____

c2: _____

c3: _____

c4: _____

... _____

... _____

... _____

... _____

... _____

... _____

... _____

b) Berechnen Sie per Division mit Rückspeichern den Quotienten a/b für die folgenden fünf-bit Zahlen: $a = 01101_2$, $b = 00011_2$. Füllen Sie hierzu das folgende Schema aus, in welchem A und B bereits die entsprechenden Operanden enthalten.

(11 Punkte)

B					
0	0	0	0	1	1

Nächste Aktion	P						A				
	0	0	0	0	0	0	0	1	1	0	1

Abbildung 8: Schema zur Division mit Rückspeichern.

B					
0	0	0	0	1	1

Nächste Aktion	P						A				
	0	0	0	0	0	0	0	1	1	0	1

Abbildung 9: Ersatzschema zur Division mit Rückspeichern, **ungültige Lösung streichen!**

Aufgabe 5: (RT-Entwurf) 21 Punkte

a) Der in Abbildung 10 spezifizierte Automat soll durch eine speicherprogrammierte Steuerung realisiert werden. Der Startzustand des Automaten ist mit S0 bezeichnet. Tragen Sie die Bus- und Speicherbreiten für die Mikroprogrammsteuerung in Abbildung 11 ein.

Hinweis: Eine '1' am Eingang `load` lädt die jeweilige Sprungadresse in den 2-bit Zähler. (3 Punkte)

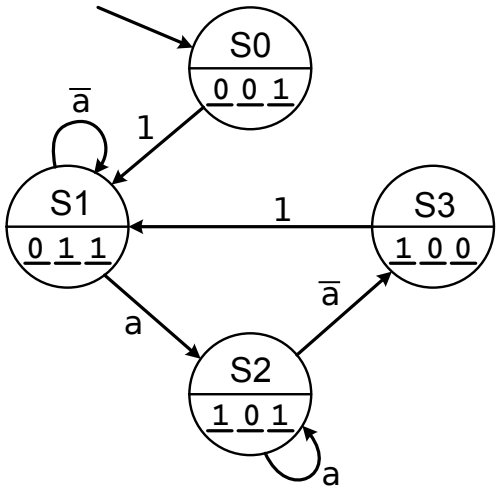


Abbildung 10: Automat

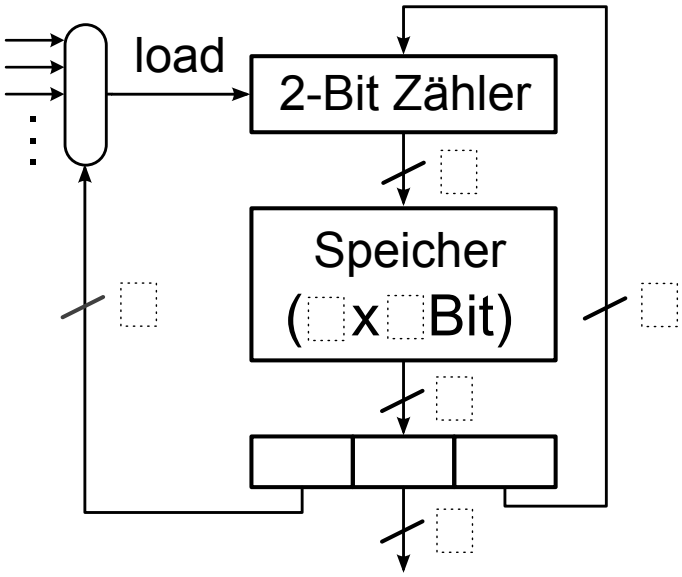


Abbildung 11: Schematische Darstellung einer Mikroprogrammsteuerung

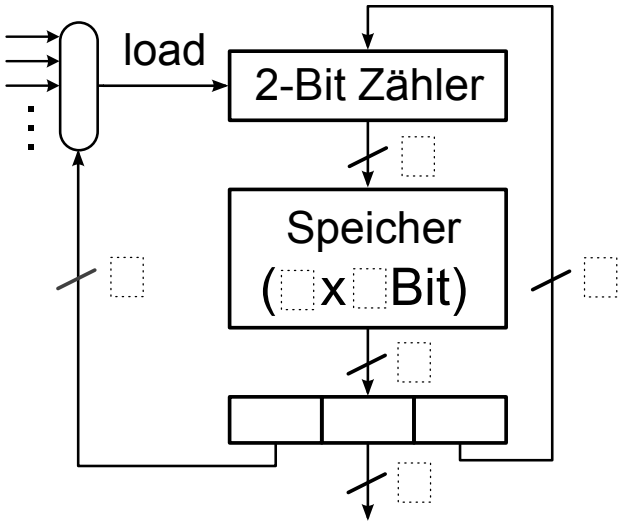


Abbildung 12: Ersatz Datstellung einer Mikroprogrammsteuerung, **ungültige Lösung streichen!**

b) Bestimmen Sie die **notwendigen** Eingangsleitungen des Multiplexers der speicherprogrammierten Steuerung für den Automaten aus Aufgabenteil **a)** und geben Sie eine Binärkodierung dafür an.

(3 Punkte)

Name der Eingangsleitung	Binärkodierung

Tabelle 8: Multiplexer Signale

Name der Eingangsleitung	Binärkodierung

Tabelle 9: Ersatztabelle Multiplexer Signale, **ungültige Lösung streichen!**

c) Kann der Automat in Abbildung 13 als Mikroprogramm mit einem Modulo-3 Zähler realisiert werden? Begründen Sie Ihre Antwort.

(3 Punkte)

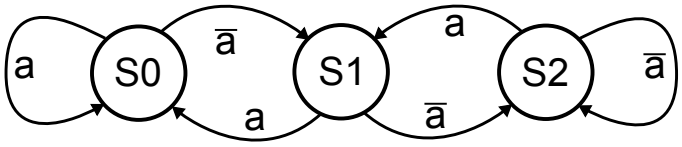


Abbildung 13: Automat

d) Realisieren Sie den Automat in Abbildung 14 als Mikroprogramm. Füllen Sie dazu das Schema in Abbildung 15 aus ohne die Signale am Multiplexereingang zu verändern.

Hinweis: Überlegen Sie zunächst, in welcher Reihenfolge die Zustände durchlaufen werden müssen.

(12 Punkte)

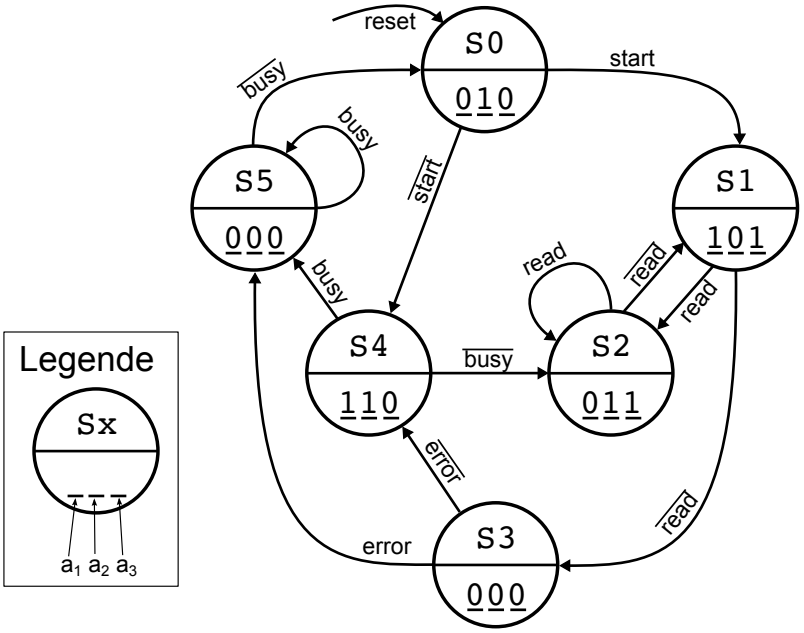


Abbildung 14: Automat

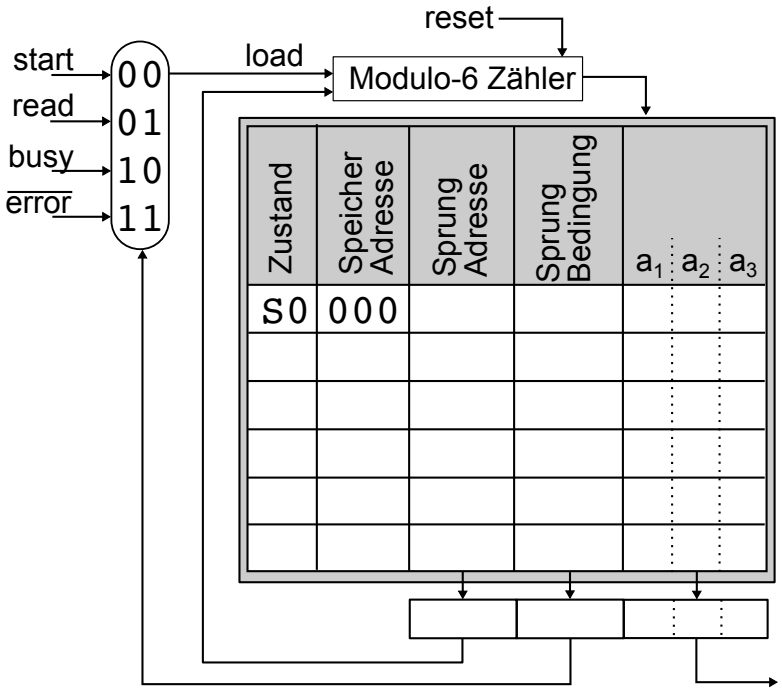


Abbildung 15: Mikroprogramm Speicher

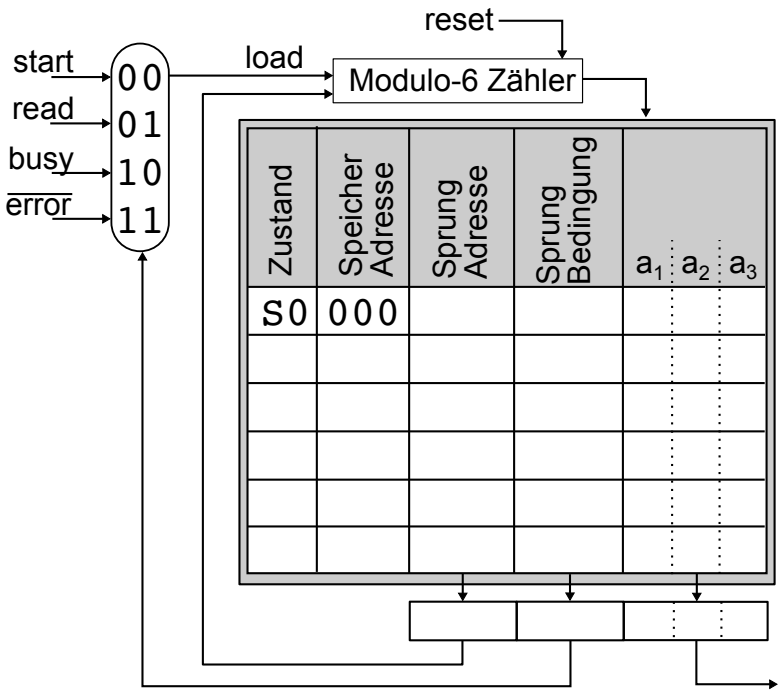


Abbildung 16: Ersatz Mikroprogramm Speicher, **ungültige Lösung streichen!**

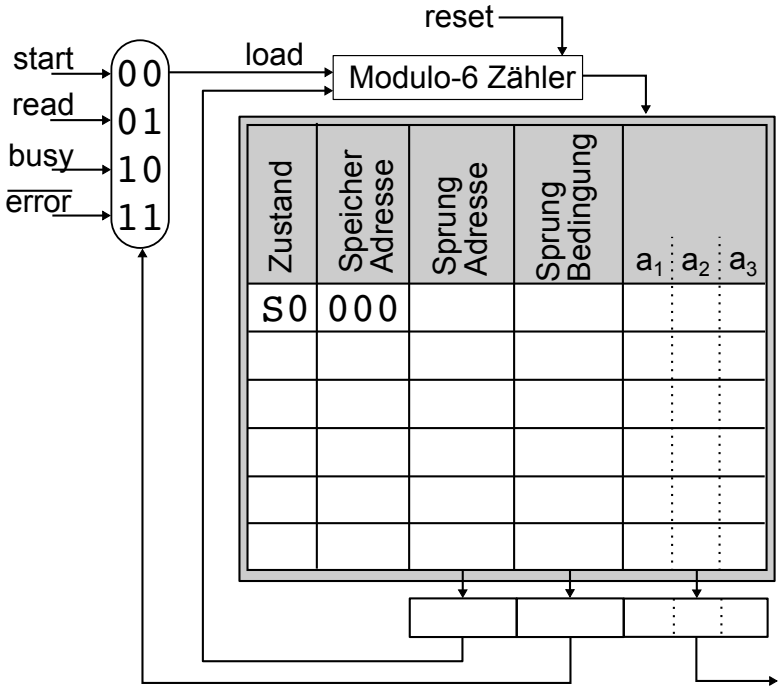


Abbildung 17: Ersatz Mikroprogramm Speicher, **ungültige Lösung streichen!**