Universität Paderborn

Institut *Elektrotechnik und Informationstechnik*

Fachgebiet *Datentechnik*

Prof. Sybille Hellebrand

Klausur

Digitaltechnik / Grundlagen Technische Informatik

3. August 2015

Punkteverteilung							
Aufgabe	1	2	3	4	5	6	Σ
maximale Punkte	10	15	15	15	15	20	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 2: (VHDL) 15 Punkte

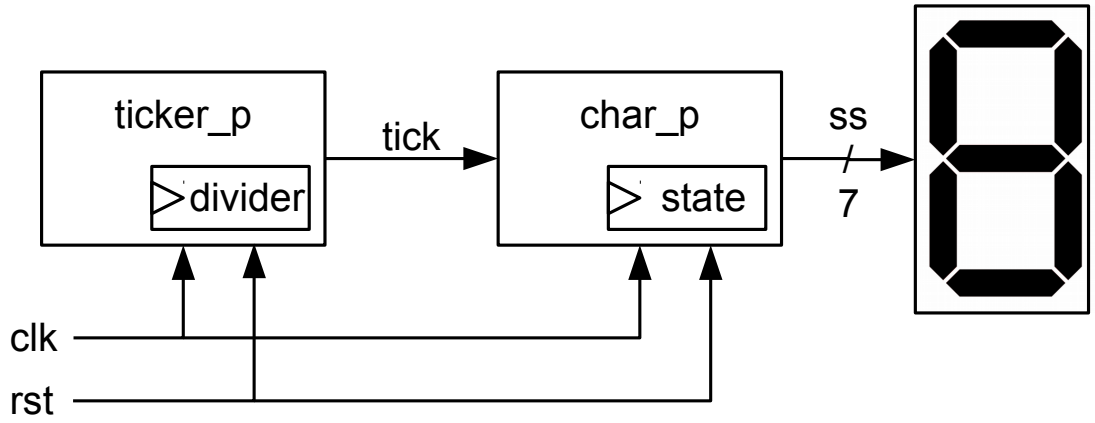


Abbildung 1: Schaltungsübersicht

Auf *einem* 7-Segment Display sollen nacheinander die Buchstaben „A“, „G“, „C“, „E“, „ “(nichts, also ein Leerzeichen) in der angegebenen Reihenfolge angezeigt werden. Der Buchstabe soll jede Sekunde wechseln. Ihre Aufgabe als Hardware-IngenieurIn ist es nun die in Abbildung 1 gezeigten Komponenten zu implementieren die den Buchstabenwechsel umsetzen.

a) 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).

(2 Punkte)

divider: _____

b) Der Signalvektor `ss` ist mit dem 7-Segment-Display verbunden. Abbildung 2 zeigt ein 7-Segment-Display und wie die Buchstaben „A “, „G “, „C “und „E “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. Füllen Sie die Dekodierungstabelle 1 aus, die vom Buchstaben auf die Steuerleitungen `ss` des 7-Segment-Displays umsetzt.

(3 Punkte)

Abbildung 2: 7-Segment Display mit Darstellung der Buchstaben AGCE. Die Nummern geben den Index im ss-Array an.

Buchstabe	ss(0)	ss(1)	ss(2)	ss(3)	ss(4)	ss(5)	ss(6)
’A’							
’G’							
’C’							
’E’							
’ ’							

Tabelle 1: Belegung des Steuersignals `ss`

Abbildung 3: 7-Segment Display mit Darstellung der Buchstaben AGCE. Die Nummern geben den Index im ss-Array an.

Buchstabe	ss(0)	ss(1)	ss(2)	ss(3)	ss(4)	ss(5)	ss(6)
'A'							
'G'							
'C'							
'E'							
' '							

Tabelle 2: Belegung des Steuersignals `ss` (**Ersatz, ungültige Lösung streichen!**)

c) Vervollständigen Sie nun folgenden VHDL-Quelltext. Pro leerer Zeile kann oder muss auch mehr als eine Anweisung eingetragen werden. Ihre Hardware soll auf steigende Taktflanken reagieren.

(10 Punkte)

```
library IEEE; use IEEE.STD_LOGIC_1164.ALL;

entity agce is
    port ( clk, rst : in   STD_LOGIC;
           ss       : out  STD_LOGIC_VECTOR (6 downto 0));
end entity;

architecture ____________________ of agce is

  signal divider : ____________________;
  signal tick: std_logic;
  type state_t is (_____________________________________________________________);
  signal state: state_t;
begin

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

   elsif ____________________________ then

    if divider = ____________________________ then

      ____________________________________________________________
     else

      ____________________________________________________________
    end if;
   end if;
  end process;

  char_p: process(_____________________________) is
  begin
    if rst = '1' then
     state <= BLANK; ss <= "0000000";

    elsif ____________________________ and tick ='1' then
      case state is
        when BLANK => ____________________________________________________

        when A =>     ____________________________________________________

        when G =>     ____________________________________________________

        when C =>     ____________________________________________________

        when E =>     ____________________________________________________

        when others => ____________________________________________________
      end case;
    end if;
  end process;

end _______________;
```

(Ersatz, ungültige Lösung streichen!)

```
library IEEE; use IEEE.STD_LOGIC_1164.ALL;

entity agce is
    port ( clk, rst : in    STD_LOGIC;
           ss       : out   STD_LOGIC_VECTOR (6 downto 0));
end entity;

architecture ____________________ of agce is

  signal divider : ____________________;
  signal tick: std_logic;
  type state_t is (____________________________________________________________);
  signal state: state_t;
begin

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

   elsif ____________________________ then

    if divider = ____________________________ then

      ____________________________________________________________
     else

      ____________________________________________________________
    end if;
   end if;
  end process;

  char_p: process(_____________________________) is
  begin
    if rst = '1' then
     state <= BLANK; ss <= "0000000";

    elsif ____________________________ and tick ='1' then
      case state is
        when BLANK => ____________________________________________________

        when A =>     ____________________________________________________

        when G =>     ____________________________________________________

        when C =>     ____________________________________________________

        when E =>     ____________________________________________________

        when others => ____________________________________________________
      end case;
    end if;
  end process;

end _______________;
```

Aufgabe 3: (Logikoptimierung) 15 Punkte

a) Für eine Funktion $f(a,b,c,d)$ sind folgende Minterme gegeben:

$$\mathcal{P} = \{0101,0110,0111,1000,0100,1011,1101,1111\}$$

Bestimmen Sie die Primimplikanten mit dem Quine-McCluskey-Verfahren.

(7 Punkte)

Lösung: _____

b) Bestimmen Sie eine minimale Überdeckung der Minterme aus der unten stehenden Tabelle 3. Verwenden Sie dazu den in der Vorlesung besprochenen Algorithmus. Falls nötig, wählen Sie den Primimplikanten mit dem kleinsten Index als Auswahlheuristik. Führen Sie die verschiedenen Schritte jeweils in einer eigenen Tabelle durch. Notieren Sie die durchgeführten Schritte in den dafür vorgesehenen Feldern. (8 Punkte)

Lösung: _____

P/m	a	b	c	d	e	f	g	h	i
$\mathcal{P}_1$	X				X				X
$\mathcal{P}_2$		X	X	X					X
$\mathcal{P}_3$		X	X						X
$\mathcal{P}_4$	X		X	X					
$\mathcal{P}_5$						X	X		
$\mathcal{P}_6$						X			
$\mathcal{P}_7$						X		X	
$\mathcal{P}_8$					X		X	X	

Tabelle 3: Primimplikantentafel

Aktion:_____

P/m									

Aktion:_____

P/m									

Aktion:____

P/m									

Aktion:____

P/m									

Aktion:____

P/m									

Aktion:____

P/m									

Aktion:____

P/m									

Aktion:____

P/m									

Ersatztafel, ungültige Lösungen streichen!

Aktion:

P/m									

Ersatztafel, ungültige Lösungen streichen!

Aktion:

P/m									

Ersatztafel, ungültige Lösungen streichen!

Aktion:

Aufgabe 4: (Division mit Rückspeichern) 15 Punkte

Abbildung 4 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.

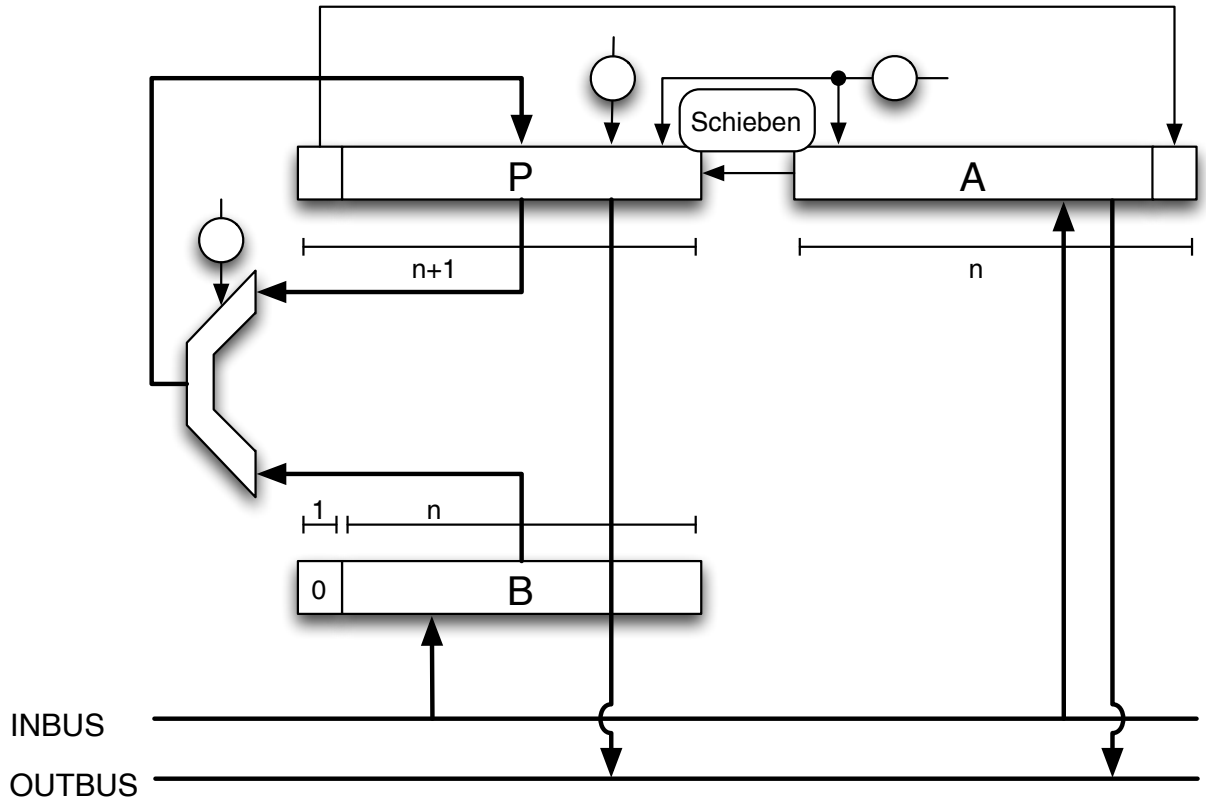


Abbildung 4: Datenpfad eines Dividierers mit einigen Kontrollpunkten

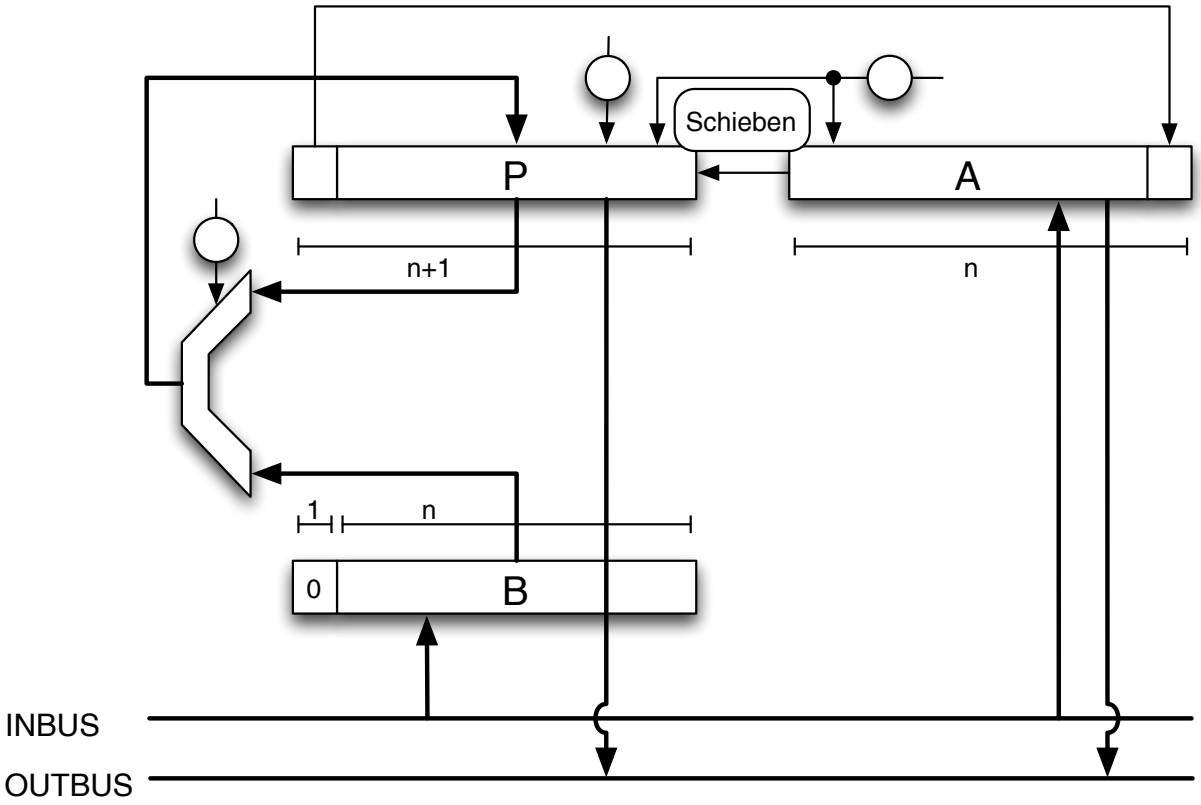


Abbildung 5: Ersatz Dividierer-Datenpfad, **ungültige Lösung streichen!**

a) Ergänzen (und benennen) Sie alle notwendigen Kontrollpunkte im Datenpfad (Abbildung 4 oder 5). 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 6: Schema zur Division mit Rückspeichern.

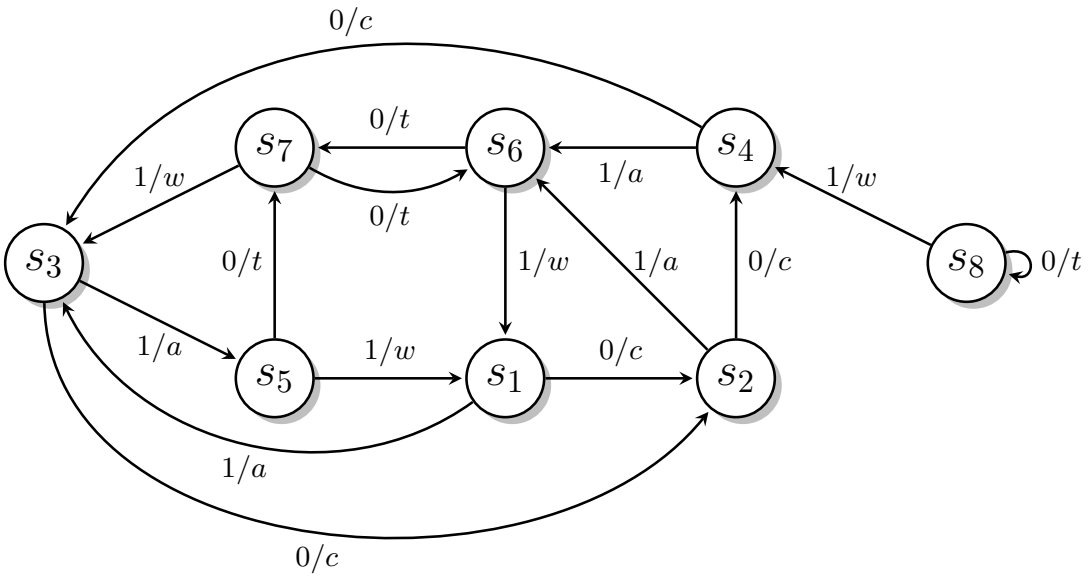
B					
0	0	0	0	1	1

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

Abbildung 7: Ersatzschema zur Division mit Rückspeichern, **ungültige Lösung streichen!**

Aufgabe 5: (Automaten) 15 Punkte

a) Gegeben sei der folgende Mealy-Automat $D = (I, O, S, \delta, \lambda)$ ohne Startzustand.



Ergänzen Sie die fehlenden Angaben zu D. (3 Punkte)

$I =$_____

$O =$_____

$S =$_____

δ/λ		

b) Führen Sie für die unten angegebene Automatentafel des Automaten D' eine Reduktion nach dem Ginsburg/Huffman-Verfahren durch. Schreiben Sie in die Tabellen das Zwischenergebnis nach jedem Schritt des Verfahrens. Geben Sie anschließend die Definition des reduzierten Automaten an.
(Ersatztabellen finden Sie auf Seite 22) (8 Punkte)

Automatentafel des Automaten D':

δ/λ	0	1
z_0	$z_5/11$	$z_0/01$
z_1	$z_7/11$	$z_3/01$
z_2	$z_4/11$	$z_1/01$
z_3	$z_4/11$	$z_1/01$
z_4	$z_7/00$	$z_6/10$
z_5	$z_3/00$	$z_7/10$
z_6	$z_3/00$	$z_5/10$
z_7	$z_2/00$	$z_6/10$

δ		

δ		

δ		

δ		

Definition des reduzierten Automaten:

$I =$_____

$O =$_____

$S =$_____

δ/λ		

c) Zeichnen Sie den Graphen des reduzierten Automaten. (2 Punkte)

d) Wie viele Zustände hat ein zum reduzierten Automaten äquivalenter Moore-Automat, wenn Sie ausschließlich die Umwandlung ohne weitere Optimierungen vornehmen? Begründen Sie Ihre Antwort. (2 Punkte)

Ersatztabellen für Aufgabenteil b). Denken Sie daran, ungültige Lösungen zu streichen!

Aufgabe 6: (RT-Entwurf) 20 Punkte

In dieser Aufgabe soll eine Schaltung zur sequenziellen Berechnung des Skalarprodukts $\langle\vec{x},\vec{y}\rangle := \sum_{i=0}^{n-1} x_i \cdot y_i$ zweier positiver ganzzahliger n-dimensionaler Vektoren $\vec{x} = (x_0,...,x_{n-1})$ und $\vec{y} = (y_0,...,y_{n-1})$ entworfen werden. Die Bitbreite von x_i und y_i beträgt 8.

Der Datenpfad (Abbildung 8) besteht aus den Registern x, y, und *result*, sowie aus einem Multiplizierer *mult* und einem Addierer *add*. Die Eingabedaten werden der Reihe nach über einen 8-Bit Bus (INBUS) zur Verfügung gestellt (Abbildung 9).

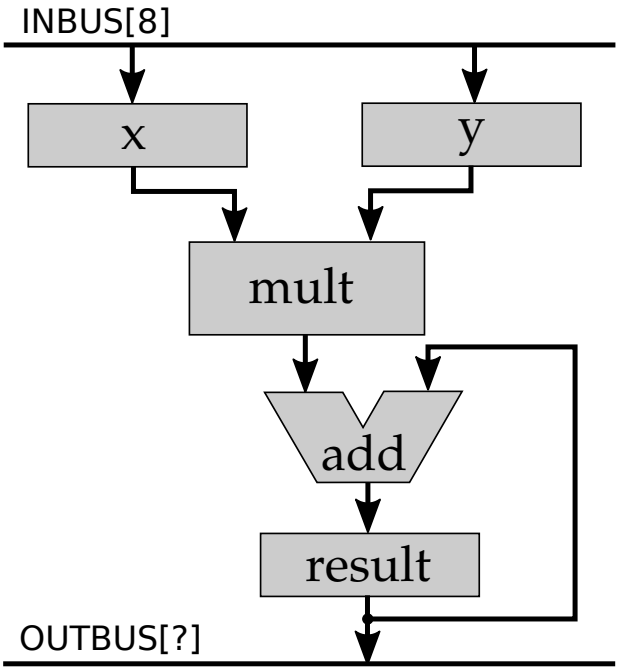


Abbildung 8: Datenpfad zur Berechnung des Skalarprodukts

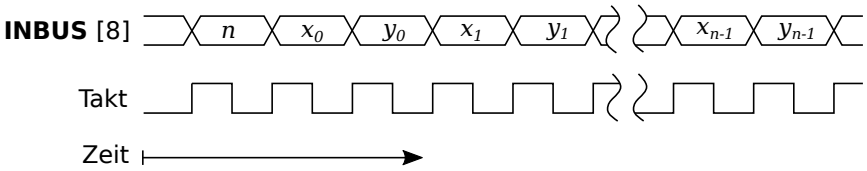


Abbildung 9: Zeitliches Verhalten der Signale

Nach dem Ende der Berechnung soll das Ergebnis an einen zweiten Bus (OUTBUS) ausgegeben werden.

(a) Zeichnen Sie für $n = 5$ die minimalen Busbreiten in die dafür vorgesehenen Kreise im Datenpfad (Abbildung 10) ein, so dass es zu keinem Überlauf kommt und begründen Sie Ihre Entscheidung. Wie viele Bits muss dann das Register `result` speichern können?
Hinweis: Der Addierer kann Operanden mit unterschiedlicher Busbreite korrekt addieren. (5 Punkte)

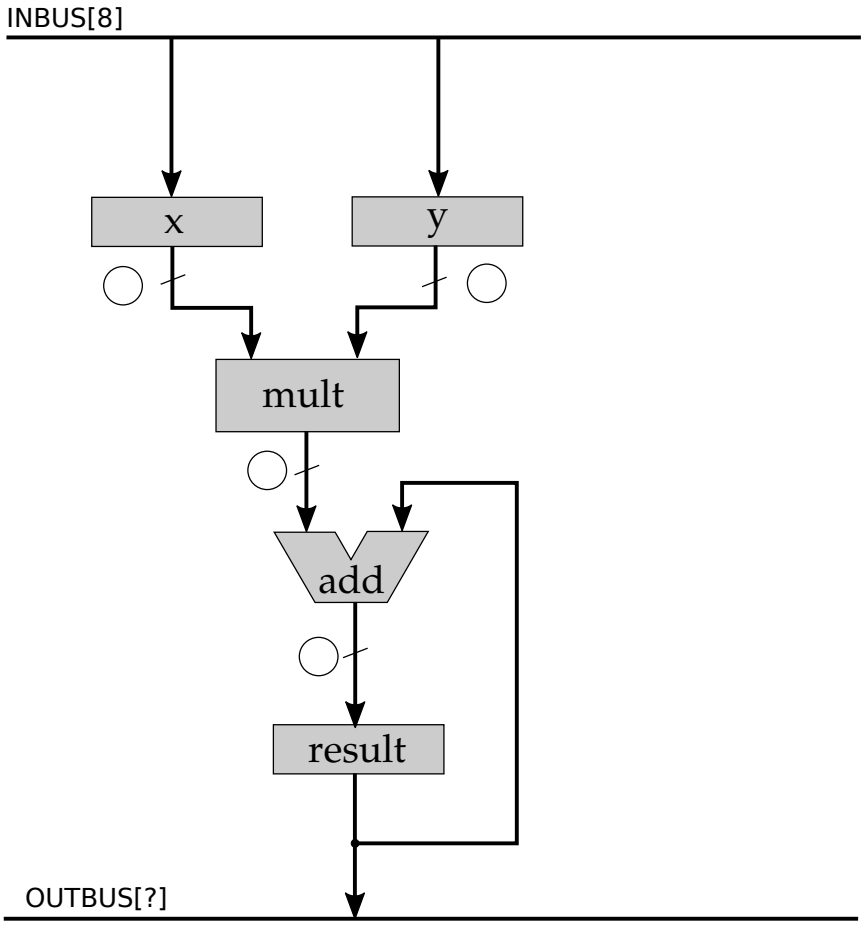


Abbildung 10: Datenpfad zur Berechnung des Skalarprodukts

(b) Nun soll die Steuerung von Ihnen entworfen werden. Zeichnen Sie dazu in den Datenpfad (Abbildung 11) die notwendigen Kontrollpunkte und eventuell weitere notwendige Komponenten (z.B. Zähler) ein. Bennen Sie die Signale und geben Sie kurz deren Funktion an. (5 Punkte)

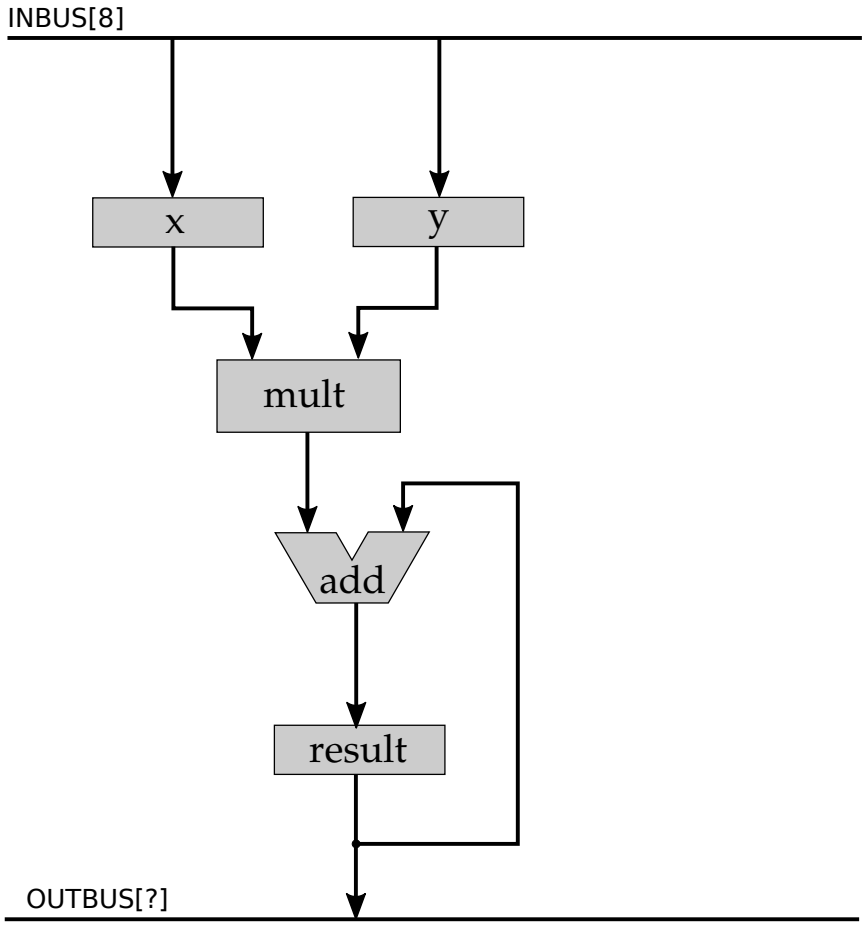


Abbildung 11: Datenpfad zur Berechnung des Skalarprodukts

Für eine andere Komponente sei bereits die Spezifikation der Steuerung gegeben (vgl. Abbildungen 12 und 13).

Abbildung 12: Steuerung des Datenpfads

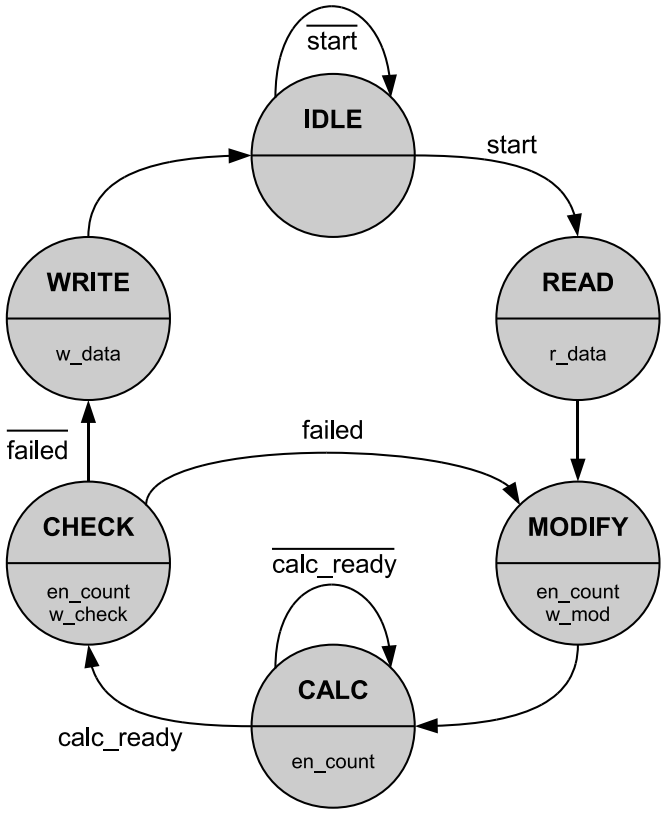


Abbildung 13: Zustandsübergangsgraph

Die Steuerung soll als μprogrammierbare Steuerung implementiert werden. Beantworten Sie zunächst die folgenden Fragen, um das μ-Befehlsformat festzulegen:

(c) Wie viele Adressbits werden benötigt? Begründen Sie Ihre Antwort! (1 Punkt)

(d) Wie viele Bits müssen für die Steuersignale reserviert werden? (1 Punkte)

(e) Welche Sprungbedingungen gibt es? Geben Sie im folgenden für die Sprungbedingung jeweils eine Kodierung mit einer minimalen Anzahl von Bits an. (2 Punkte)

Sprungbedingung	Kodierung

(f) Vervollständigen Sie nun den Speicherinhalt in Abbildung 14 für das μProgramm. Nutzen Sie Ihre Überlegungen aus Aufgabenteil **e)**. Benennen Sie auch die Steuersignale in der Kopfzeile der Tabelle. (6 Punkte)

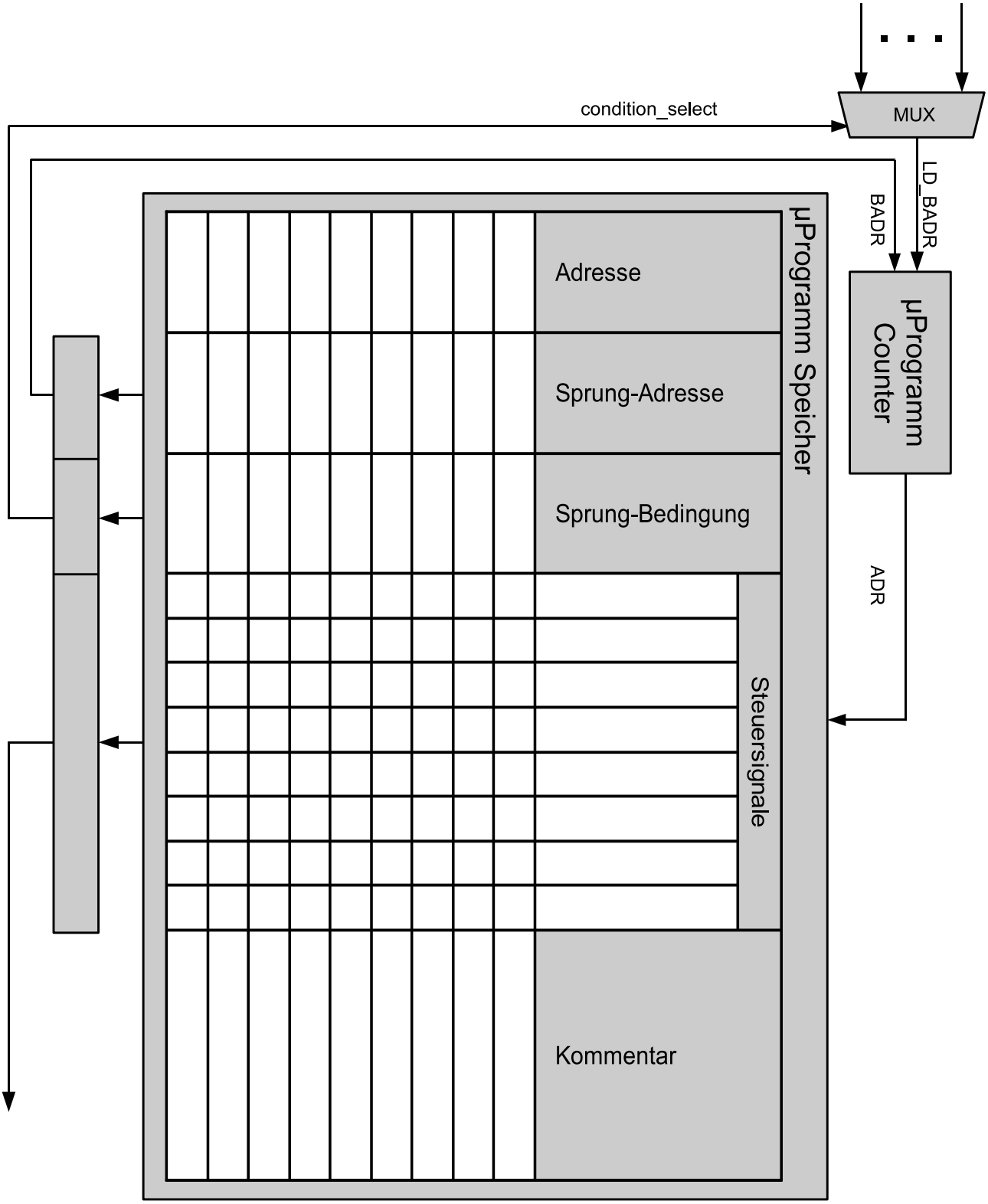


Abbildung 14: Logik des Steuerwerks

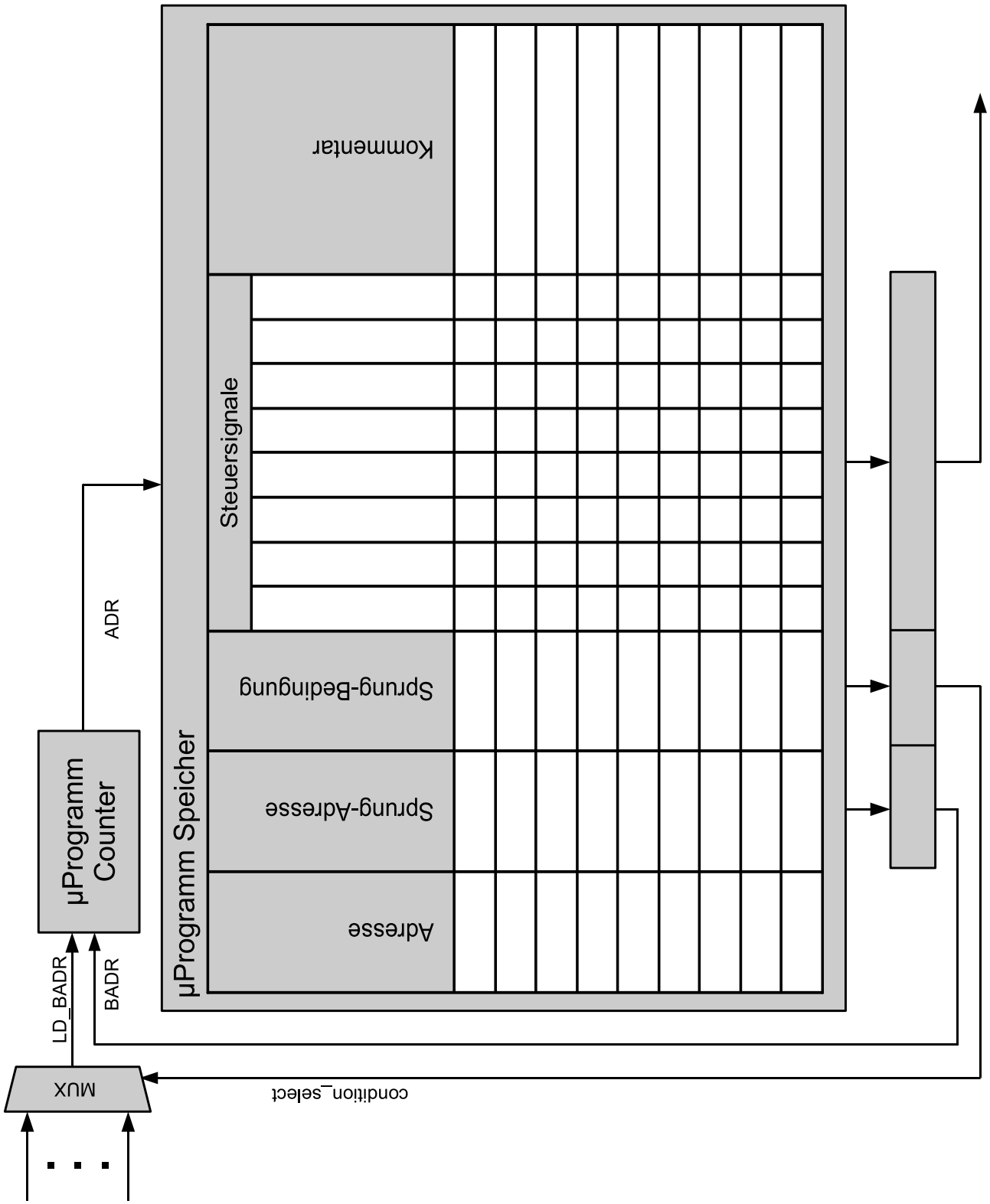


Abbildung 15: Ersatz Logik des Steuerwerks, **ungültige Lösungen streichen!**