

Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren & Rechnerorganisation

und

Technische Informatik I/II

am 26. Februar 2016, 14:00 – 16:00 Uhr

- Beschriften Sie bitte gleich zu Beginn jedes Lösungsblatt deutlich lesbar mit Ihrem Namen und Ihrer Matrikelnummer.
- Diese Aufgabenblätter werden nicht abgegeben. Tragen Sie Ihre Lösung deshalb ausschließlich in die für jede Aufgabe vorgesehenen Bereiche der Lösungsblätter ein. Lösungen auf separat abgegebenen Blättern werden nicht gewertet.
- Außer Schreibmaterial sind während der Klausur keine Hilfsmittel zugelassen. Täuschungsversuche durch Verwendung unzulässiger Hilfsmittel führen unmittelbar zum Ausschluss von der Klausur und zur Note „nicht bestanden“.
- Soweit in der Aufgabenstellung nichts anderes angegeben ist, tragen Sie in die Lösungsblätter bitte nur die Endergebnisse ein. Die Rückseiten der Aufgabenblätter können Sie als Konzeptpapier verwenden. Weiteres Konzeptpapier können Sie auf Anfrage während der Klausur erhalten.
- Halten Sie Begründungen oder Erklärungen bitte so kurz wie möglich. (Der auf den Lösungsblättern für eine Aufgabe vorgesehene Platz steht übrigens in keinem Zusammenhang mit dem Umfang einer korrekten Lösung!)
- Die Gesamtpunktzahl beträgt 90 Punkte. Zum Bestehen der Klausur sind mindestens 40 Punkte zu erreichen.

Viel Erfolg und viel Glück!

Aufgabe 1 *Schaltfunktionen*

(7 Punkte)

Gegeben sei die Funktionstabelle der Schaltfunktion f :

c	b	a	$f(c, b, a)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1. Geben Sie die disjunktive Normalform (DNF) der Schaltfunktion f an.

1 P.
2. Tragen Sie die Schaltfunktion f in das im Lösungsblatt vorbereitete KV-Diagramm ein. Geben Sie alle Primimplikate von f an und zeichnen Sie die zugehörigen Blöcke im KV-Diagramm ein.
Geben Sie für jedes Primimplikat an, ob es sich um ein Kernprimimplikat, ein Wahlprimimplikat oder ein entbehrlches Primimplikat handelt.
Geben Sie eine konjunktive Minimalform (KMF) der Schaltfunktion f an.

3 P.
3. Die Schaltfunktion f soll mit Hilfe eines 1:8-Demultiplexers und *möglichst wenigen* weiteren Gattern realisiert werden. Geben Sie das zugehörige Schaltnetz an.

1 P.
4. Die Schaltfunktion f soll mit Hilfe eines 2:1-Multiplexers und *möglichst wenigen* weiteren Gattern realisiert werden. Geben Sie das zugehörige Schaltnetz an.

2 P.

Aufgabe 2 *Schaltnetze* (10 Punkte)

Gegeben sei das in Abbildung 1 dargestellte Schaltnetz der Schaltfunktion $y = f(d, c, b, a)$:

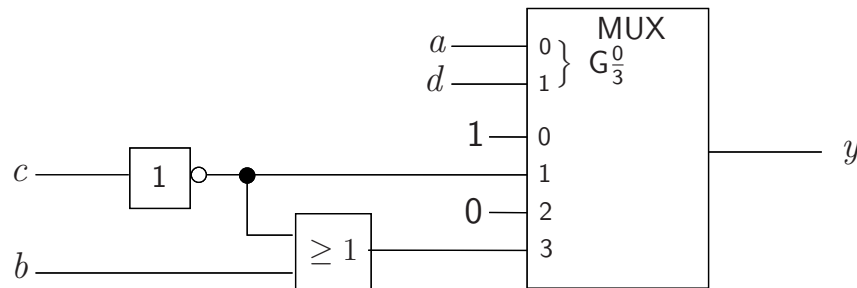


Abbildung 1: Schaltnetz der Funktion $y = f(d, c, b, a)$

1. Geben Sie die Schaltfunktion $y = f(d, c, b, a)$ an. 1 P.
2. Formen Sie die Schaltfunktion y mit Hilfe der Regeln der Schaltalgebra in eine Minimalform um. 1 P.
3. Die Schaltfunktion y soll unter ausschließlicher Verwendung von NAND-Gattern realisiert werden. Formen Sie die im letzten Aufgabenteil gefundene Minimalform entsprechend um und zeichnen Sie das zugehörige Schaltnetz. Die Eingangsvariablen stehen nur nicht negiert zur Verfügung. 3 P.
4. Die Schaltfunktion 2 P.

$$g = \text{NAND}_3(x_1, x_2, x_3) = \overline{x_1 \wedge x_2 \wedge x_3}$$

soll unter ausschließlicher Verwendung von NAND-Gattern mit zwei Eingängen realisiert werden. Wandeln Sie die Schaltfunktion entsprechend um. Zeichnen Sie das Schaltbild.

5. Geben Sie eine CMOS-Realisierung des in Abbildung 2 dargestellten 2:1-Multiplexers an. Es stehen Ihnen dabei zwei Transmission-Gates und ein Inverter zur Verfügung. 3 P.

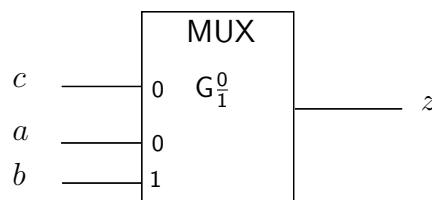


Abbildung 2: 2:1-Multiplexer

Aufgabe 3 Laufzeiteffekte (8 Punkte)

Eine Schaltfunktion $y = g(c, b, a)$ ist durch das Schaltnetz in Abbildung 3 realisiert.

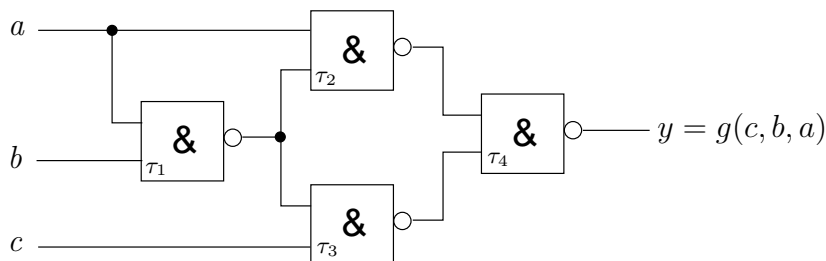


Abbildung 3: Schaltnetz der Schaltfunktion $g(c, b, a)$

1. Geben Sie das endgültige Totzeitmodell des Schaltnetzes an, indem Sie jedem Gatter seinen Verzögerungswert zuweisen und alle Totzeiten zum Eingang des Schaltnetzes verschieben. Tragen Sie die Pfadvariablen in Ihrer Lösung ein und geben Sie die Werte der Pfadverzögerungen an. Die verwendeten Gatter besitzen die Totzeiten τ_1 , τ_2 , τ_3 und τ_4 . 3 P.
2. Übertragen Sie die Schaltfunktion $y = g(c, b, a)$ in das im Lösungsblatt vorbereitete KV-Diagramm. 1 P.
3. Untersuchen Sie folgende Übergänge auf Funktionshasards: 2 P.
 - (a) Übergang 1 $(c, b, a) : (1, 1, 0) \rightarrow (0, 1, 0)$
 - (b) Übergang 2 $(c, b, a) : (1, 1, 0) \rightarrow (0, 0, 1)$

Tragen Sie die Übergänge in das KV-Diagramm aus Aufgabenteil 2 ein. Geben Sie für jeden Übergang an, ob er mit einem Funktionshasard behaftet ist oder nicht. Begründen Sie Ihre Antwort.

4. Nehmen Sie an, dass der Übergang $(c, b, a) : (1, 1, 1) \rightarrow (1, 0, 0)$ mit einem dynamischen Strukturhasard behaftet ist. Tragen Sie den Übergang wiederum in das KV-Diagramm ein. Geben Sie ein Schaltnetz in disjunktiver Form an, bei dem der obige Übergang frei von dynamischen Strukturhasards ist. Begründen Sie Ihre Antwort. 2 P.

Aufgabe 4 Schaltwerke (11 Punkte)

In Tabelle 1 ist die kodierte Ablaufabelle eines Mealy–Automaten dargestellt, der durch ein Schaltwerk mit T–Flipflops realisiert werden soll. Das Schaltwerk hat vier Zustände, zwei Eingangsvariablen e_1, e_0 und zwei Ausgangsvariablen y_1, y_0 . Die Zustände sind mit den Zustandsvariablen q_1, q_0 dual kodiert.

q_1^t	q_0^t	e_1^t	e_0^t	q_1^{t+1}	q_0^{t+1}	y_1^t	y_0^t	T_1^t	T_0^t
0	0	–	–	1	1	0	0		
0	1	0	–	1	0	1	1		
0	1	1	–	0	1	1	–		
1	0	0	–	1	1	0	1		
1	0	1	0	0	0	0	1		
1	0	1	1	1	0	1	0		
1	1	0	0	0	1	0	0		
1	1	1	0	1	1	–	1		
1	1	–	1	0	0	0	0		

Tabelle 1: Kodierte Ablaufabelle

- Bestimmen Sie mit Hilfe der im Lösungsblatt vorgegebenen KV–Diagramme die konjunktiven Minimalformen (KMF) der Ausgabefunktionen y_1^t und y_0^t . 2 P.
- Bestimmen Sie mit Hilfe der im Lösungsblatt vorgegebenen KV–Diagramme die disjunktiven Minimalformen (DMF) der Ansteuerfunktionen T_1^t und T_0^t . 3 P.

Hinweis: Vervollständigen Sie hierzu zunächst die im Lösungsblatt wiederholt angegebene Ablaufabelle.

In Abbildung 4 ist ein flankengesteuertes JK–Flipflop mit einem asynchronen „low“-aktiven Rücksetzeingang dargestellt.

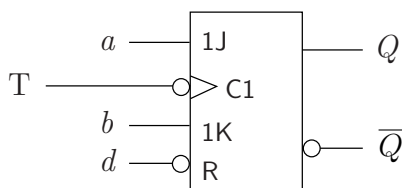


Abbildung 4: Das JK–Flipflop

- Vervollständigen Sie das im Lösungsblatt angegebene Zeitdiagramm, indem Sie den Verlauf von Q angeben. 2 P.

Gegeben ist das in Abbildung 5 dargestellte Schaltwerk. Das Schaltwerk ist mit flanken-gesteuerten D-Flipflops realisiert. Es besitzt den Eingang x und den Ausgang y .

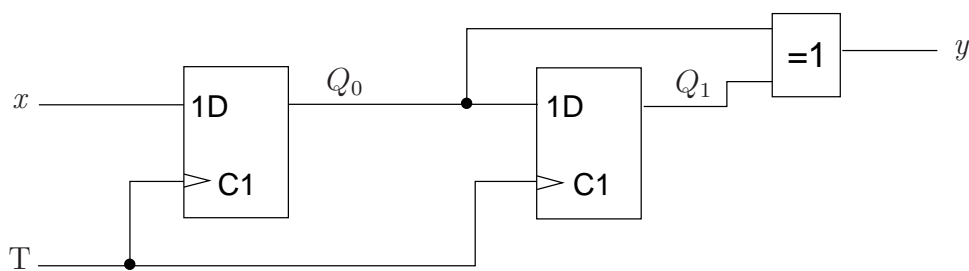


Abbildung 5: Das Schaltwerk

4. Vervollständigen Sie die Verläufe der Signale Q_0 , Q_1 und y für den im Lösungsblatt angegebenen Verlauf von x im Zeitintervall zwischen 0 und 16. 3 P.
5. Beschreiben Sie verbal die Funktion des Schaltwerkes. 1 P.

Aufgabe 5 Rechnerarithmetik

(9 Punkte)

Hinweis: Geben Sie in dieser Aufgabe *immer* den Rechenweg an.

1. Wie viele Prüfbits sind für eine Einzelbit-Fehlerkorrektur in 200 Bit Datenwörtern erforderlich? 1 P.
2. Was ist der Unterschied zwischen einem *Carry-Ripple*-Addierer und einem *Carry-Lookahead*-Addierer? Wovon hängt die Additionszeit beim *Carry-Ripple*-Addierer ab? 2 P.
3. Wie viele Bits sind *mindestens* notwendig zur Darstellung der Zahl -70 als Zweierkomplementzahl? 2 P.
 - Geben Sie -70 als Zweierkomplementzahl mit minimaler Bitanzahl an.
 - Geben Sie -70 als 16-Bit-Zweierkomplementzahl an.
4. Gegeben sei die folgende 32 Bit Folge 4 P.

1000 0011 0101 1000 0000 0000 0000 0101

Was stellt diese Folge dar, wenn sie interpretiert wird als

- (a) BCD-Zahl.
- (b) Vorzeichenlose Dualzahl. Geben Sie den dezimalen Wert an.
- (c) Gleitkommazahl im IEEE-754-Standard in einfacher Genauigkeit. Geben Sie den dezimalen Wert an.

Hinweis: Sie brauchen die Zweier-Potenzen nicht explizit auszurechnen.

Aufgabe 6 *MIPS-Assembler*

(9 Punkte)

1. Schreiben Sie die folgenden Kontrollstrukturen in MIPS-Assembler um. Die Variablen *i*, *j* und *k* stehen in den Registern *\$s1*, *\$s2* und *\$s3*.

3 P.

```
(a)  if ( i == j )
      k = i - j;
```

```
(b)  if ( i != j )
      k = i - j;
      else
      k = j;
```

```
(c)  if ( j < i )
      k = 1;
      else
      k = 0;
```

2. Führen Sie den folgenden MIPS-Code aus und geben Sie den Inhalt der Register- und Speicherinhalte nach der Ausführung an. Verwenden Sie die im Lösungsblatt angegebenen Tabellen.

4 P.

	Registersatz		Hauptspeicher	
	Register	Inhalt	Adresse	Inhalt
addi \$t3, \$0, 0x24	\$t0	0x30	\$0x20	0x24
lw \$t1, 0x04(\$t3)	\$t1	0x20	\$0x24	0x30
add \$t4, \$t3, \$t0	\$t2	0x300	\$0x28	0x34
sw \$t4, 0x08(\$t3)	\$t3	0x400	\$0x2C	0x40
	\$t4	0x1234	\$0x30	0x44

3. Wie viele Bits sieht das MIPS-Befehlsformat für ein Befehlswort vor?

1 P.

4. Was ist der Unterschied zwischen Maschinensprache und Assemblersprache?

1 P.

Aufgabe 7 *Pipelining*

(10 Punkte)

Gegeben sei folgendes MIPS-Programmstück:

```
S1:  addi  $s1, $zero, 5
S2:  lw    $s0, 0x10010000
S3:  mult  $s1, $s0
S4:  mflo  $s1
S5:  addi  $s0, $s1, -6080
S6:  sw    $s0, 0x10010000
```

1. Geben Sie den Wert der Speicherstelle 0x10010000 nach sequentieller Ausführung des Programmstücks an. 1 P.

Die Speicherstelle 0x10010000 sei vor Ausführung des Programms mit dem Wert 2016 initialisiert.

2. Bestimmen Sie alle Datenabhängigkeiten im Programmstück. Ignorieren Sie dabei Datenabhängigkeiten, die Speicherstellen im Hauptspeicher anstelle von Registern betreffen. 3,5 P.

Geben Sie zu jeder Datenabhängigkeit die beiden beteiligten Befehle, das ursächliche Register und den Typ der Datenabhängigkeit an.

3. Das Programm soll in einer DLX-Pipeline ausgeführt werden, für die keine Forwarding-Techniken implementiert sind. Fügen Sie dazu eine minimale Anzahl von Leerbefehlen (NOP-Befehlen) ein, so dass keine der Datenabhängigkeiten zu Konflikten führt. 2 P.

Sie dürfen dabei die Reihenfolge der Befehle *nicht* verändern.

4. Betrachten Sie nun eine DLX-Pipeline mit Forwarding-Techniken, d.h. *Result Forwarding* und *Load Forwarding*. Fügen Sie eine minimale Anzahl von Leerbefehlen (NOP-Befehlen) ein, so dass keine der Datenabhängigkeiten zu Konflikten führt. 2 P.

Sie dürfen dabei die Reihenfolge der Befehle *nicht* verändern.

5. Geben Sie die Anzahl der zur Ausführung des Programms notwendigen Taktzyklen für 1,5 P.

- die sequentielle Ausführung,
- die korrekte Ausführung in einer DLX-Pipeline ohne Forwarding-Techniken und
- die korrekte Ausführung in einer DLX-Pipeline mit Forwarding-Techniken

an. Nehmen Sie für die sequentielle Ausführung an, dass die Ausführung jedes Befehls fünf Taktzyklen benötigt.

Für die Ausführung auf einer DLX-Pipeline analysieren Sie die modifizierten Programmstücke, die Sie in den vorhergehenden Aufgabenteilen erstellt haben.

Aufgabe 8 Cache-Speicher

(12 Punkte)

1. Bei einem Cache-Speicher mit einer Speicherkapazität von 512 KByte ist die Hauptspeicheradresse in ein 15 Bit Tag-Feld, ein 12 Bit Index-Feld und ein 5 Bit Byte-Offset unterteilt. Geben Sie bei der Beantwortung der folgenden Fragen den Lösungsweg an.

(a) Bestimmen Sie die Blockgröße in Bytes.

1 P.

(b) Wieviele Einträge besitzt der Cache-Speicher?

1 P.

(c) Wie ist der Cache-Speicher organisiert?

2 P.

2. Es soll ein 2-fach-assoziativer (*2-way set associative cache*) Cache-Speicher mit 256 Sätzen und einer Blockgröße von 8 Byte realisiert werden. Nehmen Sie an, dass die Hauptspeicheradresse 32 Bit breit ist. Zur Verwaltung eines Cacheblocks wird nur ein Statusbit (*Valid-Bit: V*) verwendet.

3 P.

Bestimmen Sie den insgesamt erforderlichen Speicherbedarf zur Realisierung dieses Cache-Speichers. Geben Sie dabei den Rechenweg an.

3. Gegeben sei ein direkt-abgebildeter Cache-Speicher (*direct mapped cache*) mit einer Speicherkapazität von 64 Byte und einer Blockgröße von 16 Byte. Als Aktualisierungsstrategie wird ein Durchschreibverfahren (*write through policy*) verwendet. Bei dieser Aktualisierungsstrategie wird ein CPU-Datum bei einem *write miss* nur in den Speicher geschrieben. Bei einem *write hit* wird ein CPU-Datum sowohl in den Cache als auch in den Speicher geschrieben.

5 P.

Betrachten Sie die folgenden Lese- und Schreibzugriffe auf die in dezimaler Schreibweise angegebenen Adressen:

Adresse	0	32	96	16	112	32	16	112	64	0
read/write	r	r	r	r	r	r	r	w	w	r
Index	0	2								
Tag	0	0								
Hit/Miss	Miss									

Vervollständigen Sie die obige Tabelle im Lösungsblatt. Verwenden Sie dabei **Miss** für Cache-Miss und **Hit** für Cache-Hit.

Aufgabe 9 *Speicherverwaltung* (8 Punkte)

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer Seitengröße von 2 KByte, 8 virtuellen Seiten und 4 physikalischen Seiten (Frames). Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 2 angegeben.

Virtuelle Seitennummer	Physikalische Seitennummer
0	2
1	0
2	-
3	-
4	1
5	-
6	3
7	-

Tabelle 2: Seitentabelle

1. Skizzieren Sie die Unterteilung der 32 Bit breiten virtuellen Adresse. 1 P.
2. Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen: 4 P.
1024, 2047, 2048, 2102, 4095, 4096, 8192, 8202

Zur Beschleunigung der Adressberechnung soll ein Cache-Speicher als *Translation Lookaside Buffer (TLB)* eingesetzt werden, der die letzten 32 Einträge aus dem Seitentabellen-Verzeichnis und der Seitentabelle speichert.

3. Unter welchen Bedingungen wird eine Beschleunigung der Adressumsetzung durch einen *Translation Lookaside Buffer (TLB)* erreicht? 1 P.
4. Wie breit ist der *Tag* eines Cache-Eintrags? Gehen Sie dabei von einer n Bit breiten virtuellen Adresse, einer m Bit breiten physikalischen Adresse und einer Seitengröße von 4 KByte aus. 2 P.

Aufgabe 10 *Verschiedenes* (6 Punkte)

1. Was versteht man unter dem *Y-Diagramm*? Zeichnen Sie das Y-Diagramm von Gajski und beschriften Sie die „Achsen“ und die verschiedenen Entwurfsebenen. 3 P.
2. Was ist der entscheidende Nachteil eines Befehlsformats mit variabler Länge? 1 P.
3. Wofür steht die Abkürzung „ALU“? 1 P.
4. Wofür steht die Abkürzung „RISC“? 1 P.

Lösungsblätter zur Klausur

Digitaltechnik und Entwurfsverfahren & Rechnerorganisation

und

Technische Informatik I/II

am 26. Februar 2016, 14:00 – 16:00 Uhr

Name:	Vorname:	Matrikelnummer:
-------	----------	-----------------

Digitaltechnik und Entwurfsverfahren/TI-1	
Aufgabe 1	von 7 Punkten
Aufgabe 2	von 10 Punkten
Aufgabe 3	von 8 Punkten
Aufgabe 4	von 11 Punkten
Aufgabe 5	von 9 Punkten

Rechnerorganisation/TI-2	
Aufgabe 6	von 9 Punkten
Aufgabe 7	von 10 Punkten
Aufgabe 8	von 12 Punkten
Aufgabe 9	von 8 Punkten
Aufgabe 10	von 6 Punkten

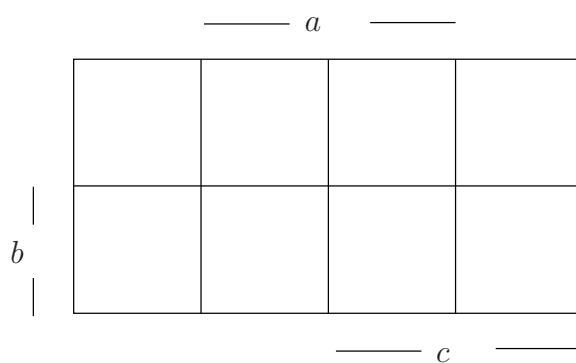
Gesamtpunktzahl:	
-------------------------	--

	Note:
--	--------------

Aufgabe 1

1. DNF von $f(c, b, a)$:

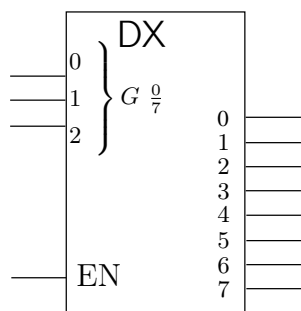
2. KV-Diagramm $f(c, b, a)$:



Primimplikate:

KMF von $f(c, b, a)$:

3. Schaltnetz:



Name:

Vorname:

Matr.-Nr.:

3

4. Schaltnetz:

Aufgabe 2

1. y :

2. Minimalform von y :

Name:

Vorname:

Matr.-Nr.:

4

3. Minimalform von y in NAND-Form:

Schaltnetz:

4. Realisierung von $g(x_1, x_2, x_3)$ mit NAND-Gattern:

Schaltbild:

Name:

Vorname:

Matr.-Nr.:

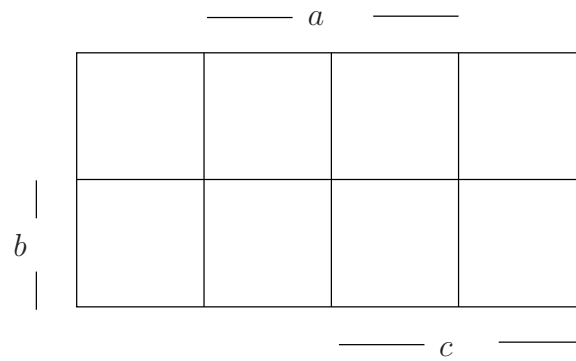
5

5. CMOS-Realisierung eines 2:1-Multiplexers:

Aufgabe 3

1. Totzeitmodell:

2. KV-Diagramm für g :



3. (a) Übergang 1 $(c, b, a) : (1, 1, 0) \rightarrow (0, 1, 0)$

(b) Übergang 2 $(c, b, a) : (1, 1, 0) \rightarrow (0, 0, 1)$

4. Übergang $(c, b, a) : (1, 1, 1) \rightarrow (1, 0, 0)$

Aufgabe 4

1. KMF der Ausgabefunktionen:

 $y_1^t :$

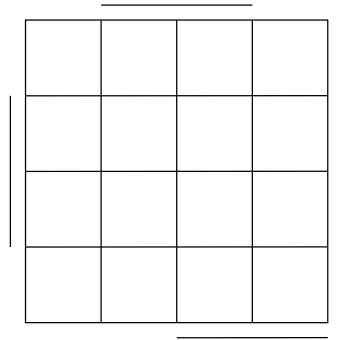
 $y_0^t :$

 $y_1^t =$
 $y_0^t =$

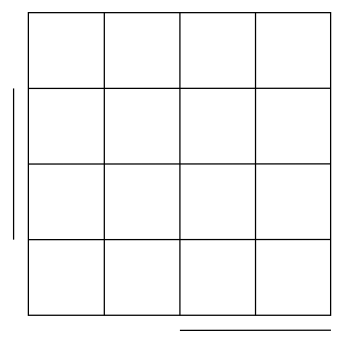
2. DMF der Ansteuerfunktionen:

q_1^t	q_0^t	e_1^t	e_0^t	q_1^{t+1}	q_0^{t+1}	y_1^t	y_0^t	T_1^t	T_0^t
0	0	-	-	1	1	0	0		
0	1	0	-	1	0	1	1		
0	1	1	-	0	1	1	-		
1	0	0	-	1	1	0	1		
1	0	1	0	0	0	0	1		
1	0	1	1	1	0	1	0		
1	1	0	0	0	1	0	0		
1	1	1	0	1	1	-	1		
1	1	-	1	0	0	0	0		

T_1^t :



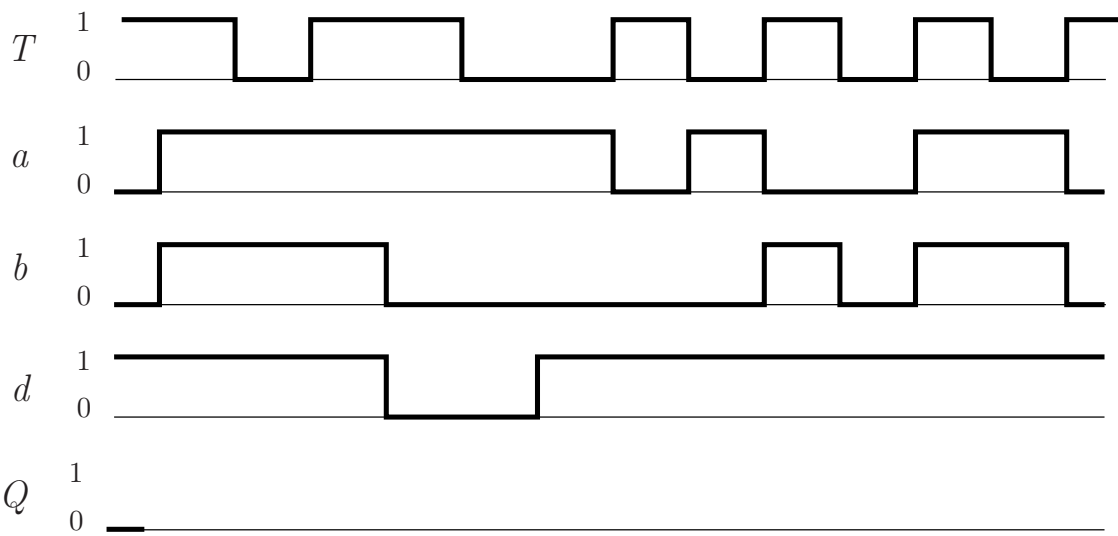
T_0^t :



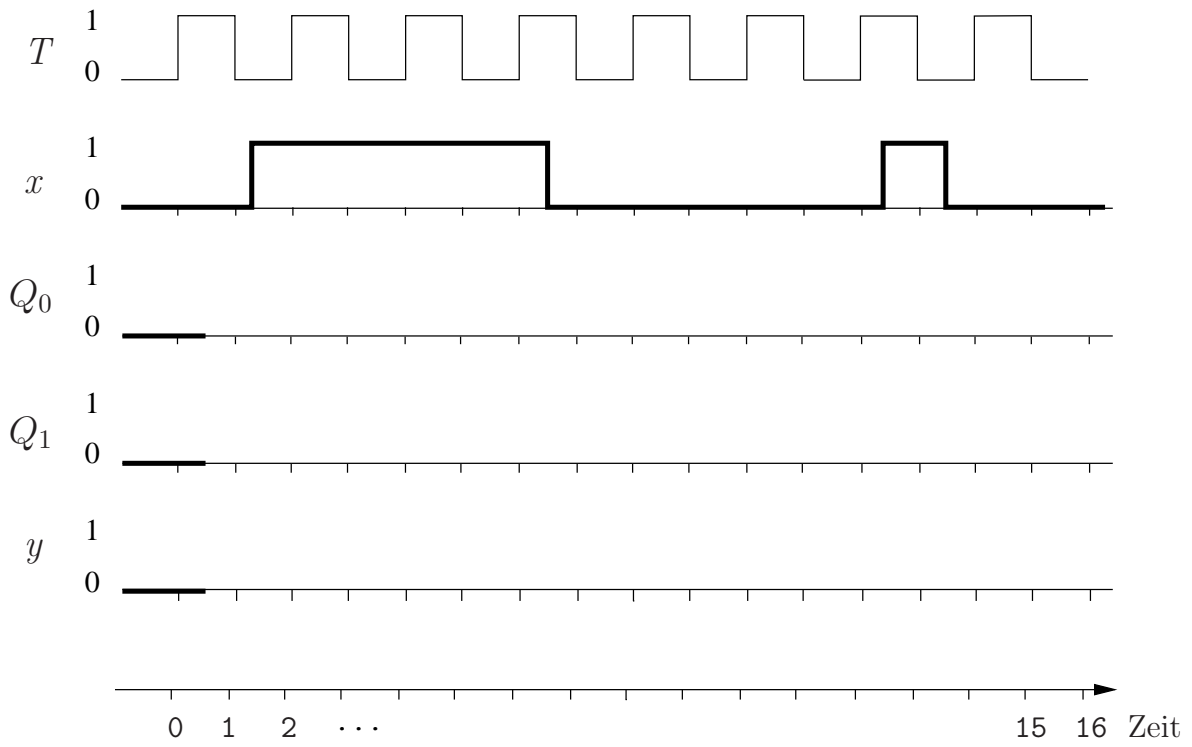
$T_1^t =$

$T_0^t =$

3. Verlauf von Q :



4. Verläufe von Q_0, Q_1 und y :



5. Funktion des Schaltwerkes:

Aufgabe 5

1. Anzahl der Prüfbits:

2. *Carry-Ripple*-Addierer und *Carry-Lookahead*-Addierer:

3. Mindest-Bitanzahl für die Darstellung von -70 als Zweierkomplementzahl:

- -70 mit minimaler Bitanzahl als Zweierkomplementzahl:

- -70 als 16-Bit-Zweierkomplementzahl:

Name:

Vorname:

Matr.-Nr.:

11

4. 1000 0011 0101 1000 0000 0000 0000 0101

(a) BCD:

(b) Vorzeichenlose Dualzahl:

(c) Gleitkommazahl im IEEE-754-Standard in einfacher Genauigkeit:

Aufgabe 6

1. MIPS-Assembler:

(a)

(b)

(c)

2. Register- und Speicherinhalte nach der Ausführung:

Registersatz		Hauptspeicher	
Register	Inhalt	Adresse	Inhalt
\$t0		\$0x20	
\$t1		\$0x24	
\$t2		\$0x28	
\$t3		\$0x2C	
\$t4		\$0x30	

3. Anzahl Bits pro Befehlswort:

4. Unterschied Maschinensprache zu Assemblersprache:

Aufgabe 7

1. Wert der Speicherstelle 0x10010000 nach sequentieller Ausführung:

2. Datenabhängigkeiten:

- Echte Abhängigkeiten:

- Gegenabhängigkeiten:

- Ausgabeabhängigkeiten:

Aufgabe 9

1. Unterteilung der virtuellen Adresse:

2. Physikalische Adressen:

Virtuelle Adresse	Physikalische Adresse
1024	
2047	
2048	
2102	
4095	
4096	
8192	
8202	

3. Beschleunigung durch TLB:

4. Breite des *Tags*:

Aufgabe 10

1. Y-Diagramm:

2. Nachteil eines Befehlsformats mit variabler Länge:

3. „ALU“ steht für:

4. „RISC“ steht für: