Fachbereich Informatik

Lehrgebiet Technische Informatik I

Kurs 1707 „Technische Informatik I“

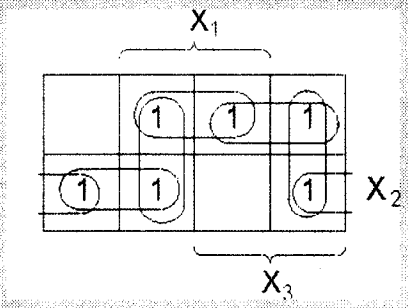
Lösungsvorschläge zu den Aufgaben

der Hauptklausur im SS 2002

(am 3. August 2002)

Aufgabe 1 (20 Punkte)

a) Geben Sie im Karnaugh-Diagramm eine Boolesche Funktion F von 3 Variablen an, die 6 Primimplikanten besitzt!



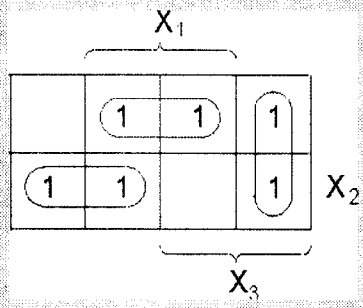
(5)

b) Wie viele derartige Funktionen gibt es ? – Begründen Sie Ihre Antwort !

Es gibt 4 derartige Funktionen, weil es genau 4 Paare von Eck-Kästchen und (per Rösselsprung erreichbaren) inneren Kästchen gibt bzw. umgekehrt.

(5)

c) Mit wie vielen Primimplikanten kann man eine derartige Funktion als möglichst kurze DNF darstellen ?



Wie man an den gezeigten Konturen sieht, kann man eine solche Funktion als DNF von 3 Konjunktionstermen darstellen.

(2)

d) Welcher besondere Typ von DNF entsteht bei c) ?

Es handelt sich hierbei um eine DDFN, d.h. eine DNF aus paarweise disjunkten Termen.

(2)

e) Bestätigen Sie die Ergebnisse aus dem Karnaugh-Diagramm zu d) durch eine algebraische Proberechnung !

Für das in Teilaufgabe 1a) abgebildete Karnaugh-Diagramm gilt beispielsweise

$$F = X_1\overline{X_2} \vee X_2\overline{X_3} \vee X_3\overline{X_1}$$

wobei alle Terme disjunkt sind, denn es gilt

$$X_1X_2\overline{X_2}X_3 = X_1X_2X_3\overline{X_1} = X_2X_3\overline{X_3}X_1 = 0$$

wegen $X_i\overline{X_i} = 0$

(3)

f) Weshalb ist eine KNF-Darstellung in diesem Fall günstiger ?

Eine mögliche KNF für das in Aufgabe 1a) gegebene Karnaugh-Diagramm, lautet

$$F = (X_1 \vee X_2 \vee X_3) \wedge (\overline{X_1} \vee \overline{X_2} \vee \overline{X_3})$$

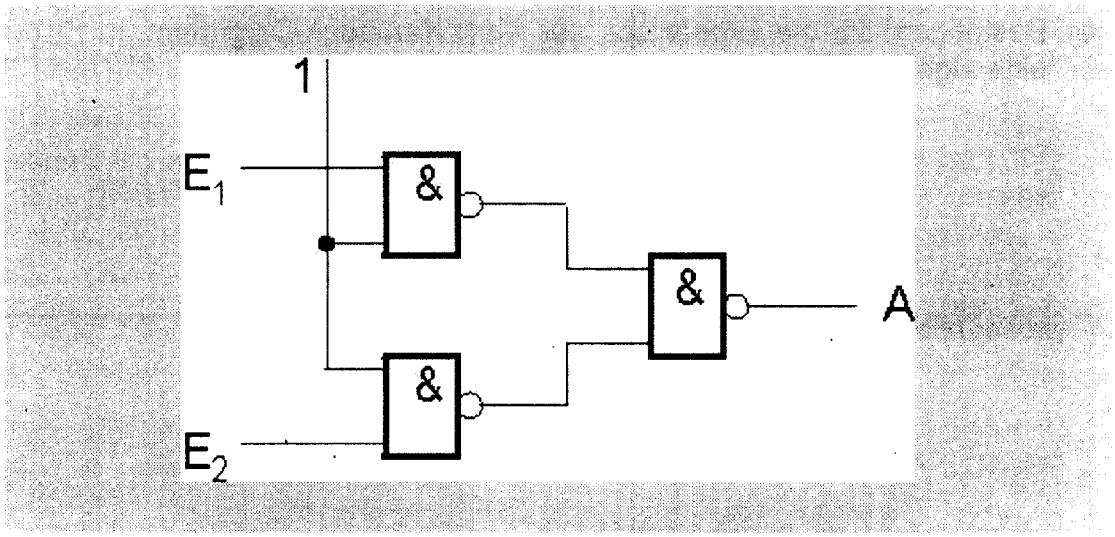
Offensichtlich werden bei der Verwendung der KNF weniger Gatter als bei einer Realisierung mittels DNF benötigt.

(3)

Aufgabe 2 (20 Punkte)

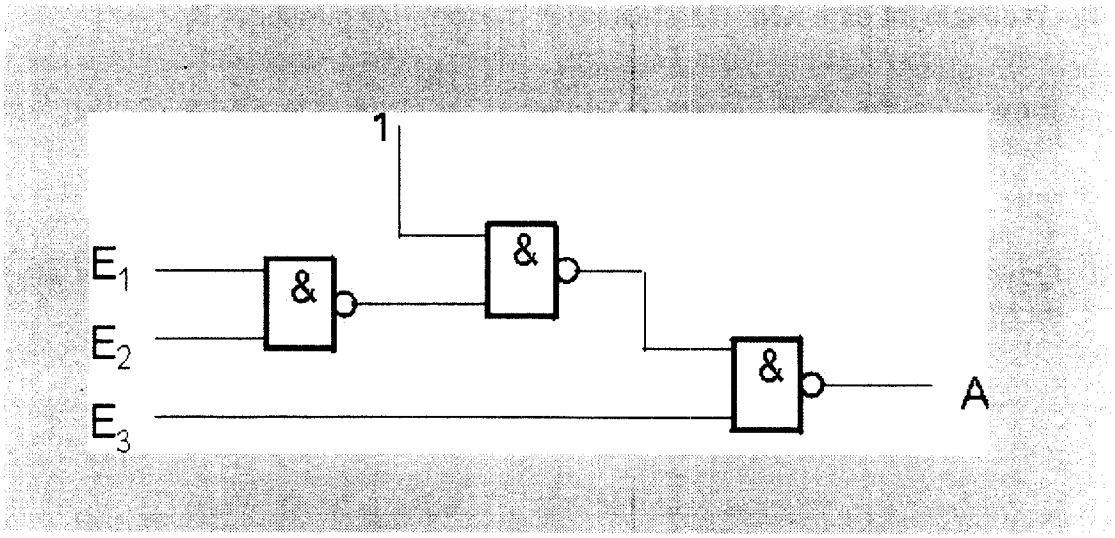
Neben dem UND-ODER-NICHT- gibt es noch weitere vollständige Operatorensysteme, so zum Beispiel das NAND- oder das NOR-System. Da diese Systeme mit nur einer einzigen Bausteinart auskommen, erfreuen sie sich in der Digitaltechnik einer großen Beliebtheit.

a) Wie lässt sich die ODER-Operation durch die Verwendung von NAND-Gattern nachbilden ? Zeichnen Sie ein entsprechendes Schaltbild.



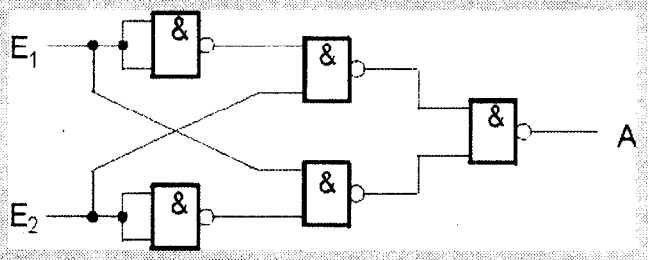
(4)

b) Wie kann ein NAND-Gatter mit 3 Eingängen unter alleiniger Verwendung von NAND-Gattern mit 2 Eingängen realisiert werden ? Zeichnen Sie das Schaltbild.



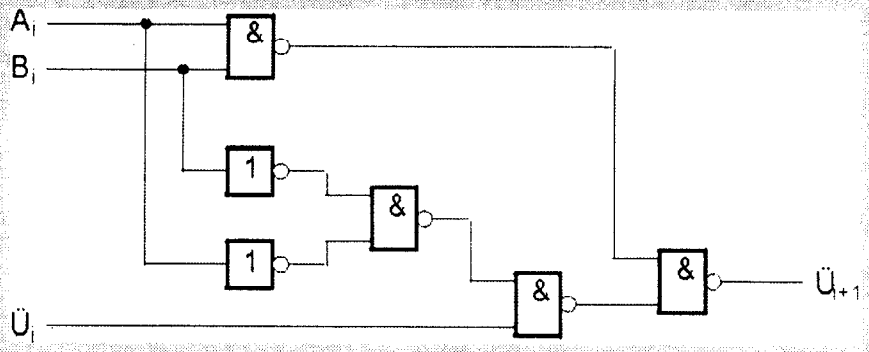
(4)

c) Beschreiben Sie mittels eines Schaltbildes, wie ein XOR-Gatter aus NAND-Gattern aufgebaut werden kann.



(6)

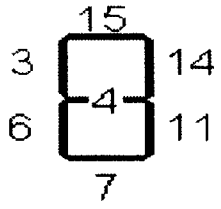
d) Realisieren Sie die Übertragsfunktion eines Volladdierers unter ausschließlicher Verwendung von NAND-Gattern und Invertern.



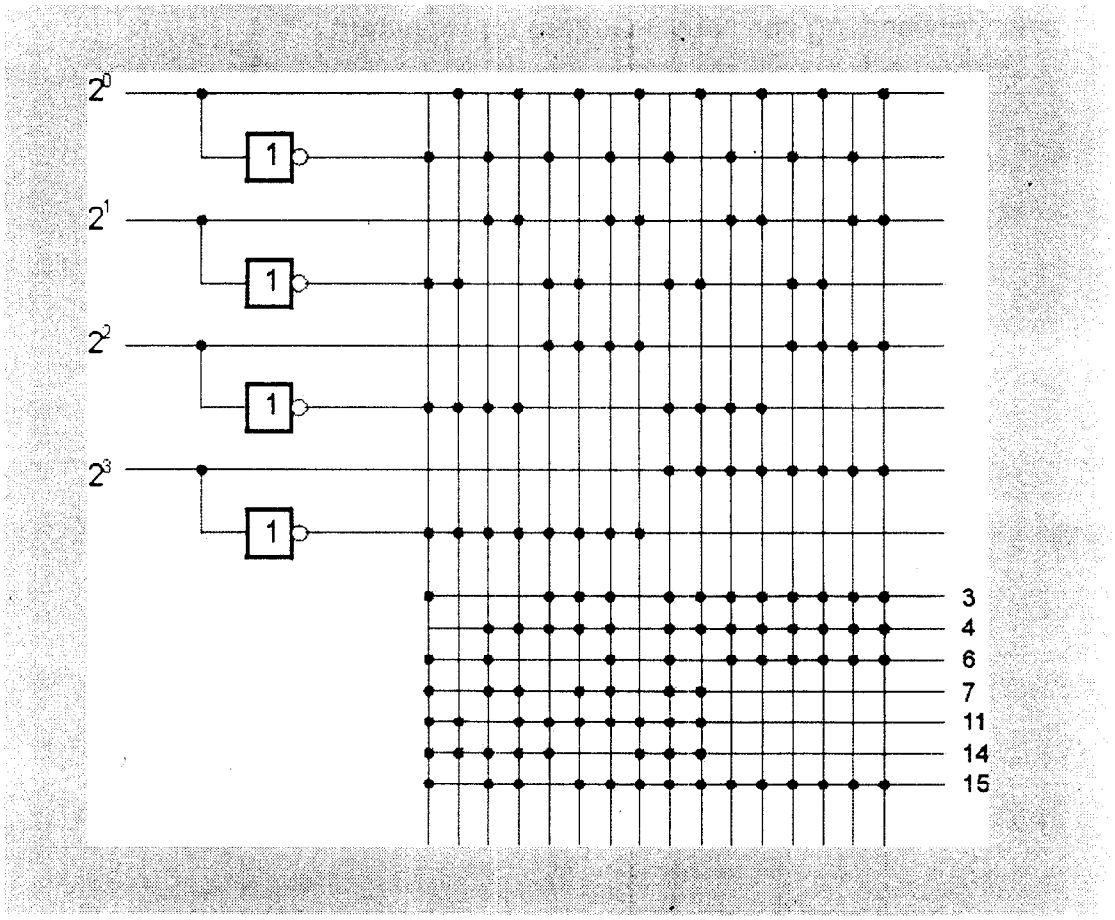
(6)

Aufgabe 3 (15 Punkte)

a) Gegeben sei eine 7-Segmentanzeige, wie sie in der folgenden Abbildung dargestellt ist. Die neben den einzelnen Segmenten dargestellten Nummern kennzeichnen jeweils den Bausteinanschluß, der für die Aktivierung dieses Segments auf logisch 1 gesetzt werden muß.



Zeichnen Sie einen PLA-Baustein, der die an seinen Eingängen anliegenden Signale als BCD-Ziffern auffasst und für die Darstellung mit dem gezeigten Display vorbereitet. Die Ausgänge des Bausteins sollen hierbei jeweils mit der Nummer des von ihnen gesteuerten Anzeigen-Segments versehen werden. An den Eingängen angelegte Pseudotetraden sollen mit einem „F" im Display angezigt werden. Die Ziffer Eins ist rechtbündig darzustellen. Bei den Ziffern 9 und 6 sollen jeweils alle horizontalen Segmente aktiviert werden. Die 7 ist ohne horizontalen Mittelstrich darzustellen.

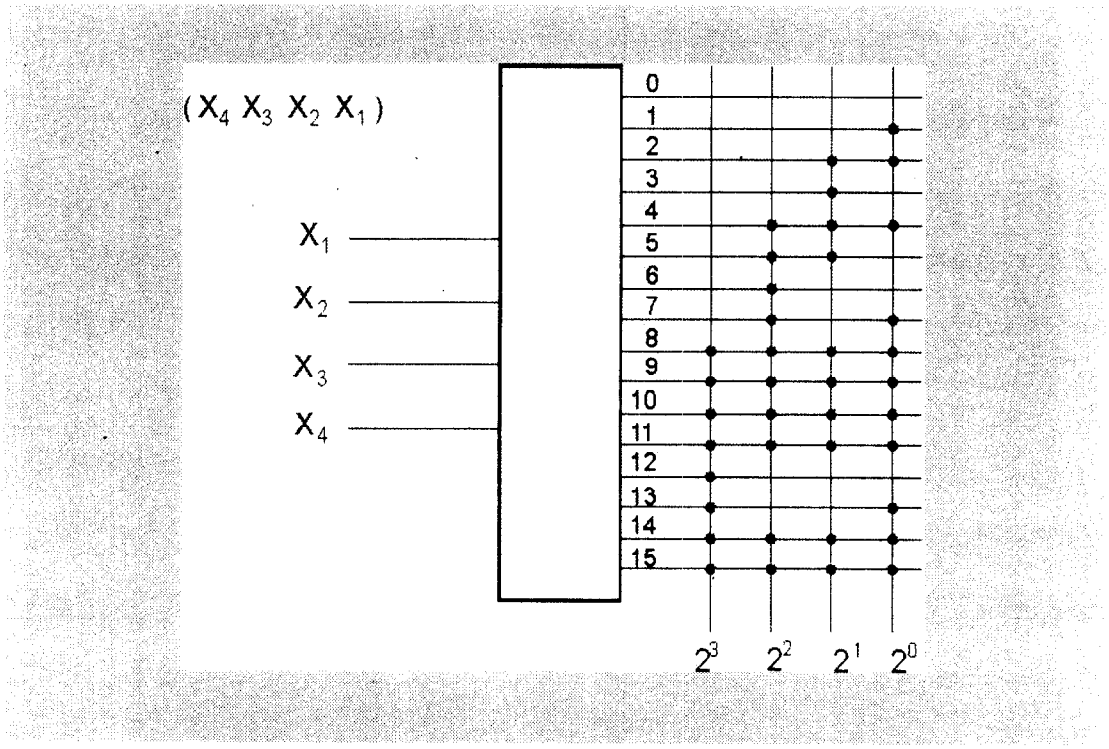


(7)

b) Neben der einfachen binären Darstellung und der BCD-Codierung gibt es noch zahlreiche andere Codes, mit denen die Ziffern des Dezimalsystems binär dargestellt werden könne. Einer dieser Codes, der sog. Gray-Code, zeichnet sich dadurch aus, dass sich zwischen zwei benachbarten Ziffern jeweils nur ein einziges Bit ändert. Für die Lösung dieser Aufgabe verwenden Sie bitte die folgende Code-Tabelle

Gray-Code	Dezimalzahl
0 0 0 0	0
0 0 0 1	1
0 0 1 1	2
0 0 1 0	3
0 1 1 0	4
0 1 1 1	5
0 1 0 1	6
0 1 0 0	7
1 1 0 0	8
1 1 0 1	9

Realisieren Sie mit einem Festwertspeicher ein Schaltnetz, das den oben gegebenen Gray-Code in BCD-Code umwandelt, wobei ungültige Eingabewerte zu der Ausgabe 1 1 1 1 führen sollen.



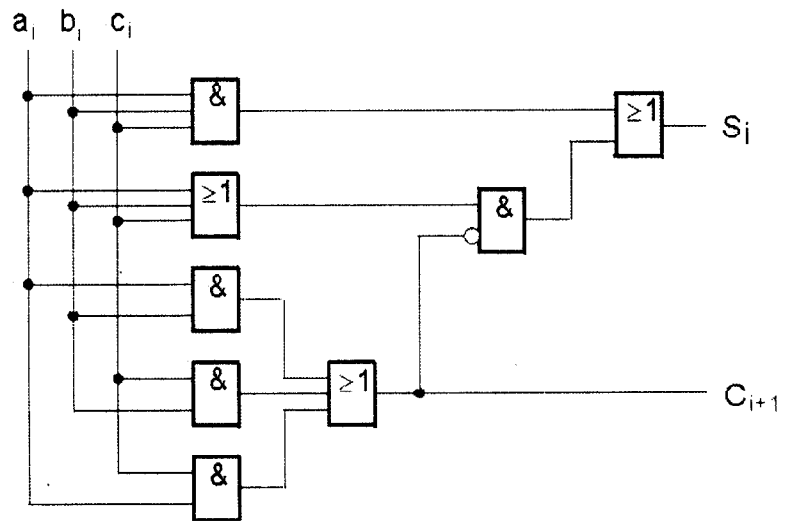
(6)

c) Beschreiben Sie mit einem Satz, welche Veränderung an dem PLA-Baustein aus Aufgabenteil a) durchgeführt werden müßte, damit dieser Gray-Code als Eingabe akzeptiert und ungültige Eingaben mit „F" quittiert !

Bei dem gegebenen PLA-Baustein müssten lediglich die Spalten der ODER-Matrix untereinander getauscht werden. (2)

Aufgabe 4 (15 Punkte)

Gegeben sei das folgende Schaltnetz:



a) Bestimmen Sie aus dem Schaltbild die Schaltfunktion für den Ausgang S_i.

$$S_i = a_i b_i c_i \vee [(a_i \vee b_i \vee c_i) \wedge \overline{(a_i b_i \vee b_i c_i \vee a_i c_i)}]$$

(3)

b) Überführen Sie diese Schaltfunktion durch Boole'sche Umformungen in die DNF für S_i! Lösungen mit Wahrheitstabellen werden dabei nicht anerkannt !

$$
\begin{aligned}
S_i &= a_ib_ic_i \vee [(a_i \vee b_i \vee c_i) \wedge (\overline{a_ib_i \vee b_ic_i \vee a_ic_i})] \\
&= a_ib_ic_i \vee [(a_i \vee b_i \vee c_i) \wedge ((\overline{a_ib_i}) \wedge (\overline{b_ic_i}) \wedge (\overline{a_ic_i}))] \\
&= a_ib_ic_i \vee [(a_i \vee b_i \vee c_i) \wedge ((\bar{a}_i \vee \bar{b}_i) \wedge (\bar{b}_i \vee \bar{c}_i) \wedge (\bar{a}_i \vee \bar{c}_i))] \\
&= a_ib_ic_i \vee [(a_i \vee b_i \vee c_i) \wedge ((\bar{a}_i\bar{b}_i \vee \bar{b}_i \vee \bar{a}_i\bar{c}_i \vee \bar{b}_i\bar{c}_i) \wedge (\bar{a}_i \vee \bar{c}_i))] \\
&= a_ib_ic_i \vee [(a_i \vee b_i \vee c_i) \wedge (\bar{a}_i\bar{b}_i \vee \bar{a}_i\bar{c}_i \vee \bar{b}_i\bar{c}_i)] \\
&= a_ib_ic_i \vee [a_i\bar{b}_i\bar{c}_i \vee \bar{a}_ib_i\bar{c}_i \vee \bar{a}_i\bar{b}_ic_i] \\
&= a_ib_ic_i \vee a_i\bar{b}_i\bar{c}_i \vee \bar{a}_ib_i\bar{c}_i \vee \bar{a}_i\bar{b}_ic_i
\end{aligned}
$$

(9)

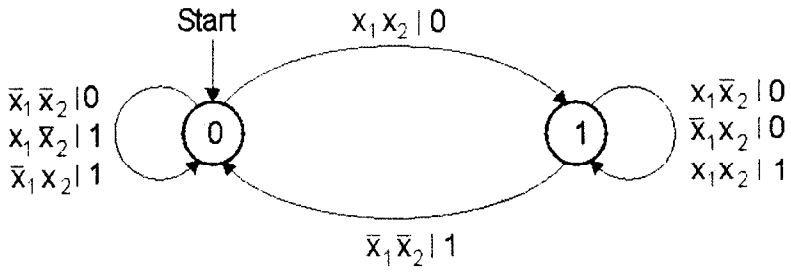
c) Realisiert die o.g. Schaltung einen Volladdierer ? Geben Sie eine kurze Begründung !

Diese Schaltung ist ein Volladdierer, da S_i wie der Summen-Ausgang des Volladdierers genau dann den Wert 1 annimmt, wenn die Anzahl der an den Eingängen anliegenden Einsen ungerade ist. Der Ausgang Ci+1 wird genau dann 1, wenn an mdst. 2 Eingängen eine 1 anliegt. Er funktioniert also genau wie der Übertrags-Ausgang des Volladdierers.

(3)

Aufgabe 5 (15 Punkte)

Die folgende Abbildung zeigt den Zustandsgraphen eines Serienaddierers, d.h. der Serienaddierer wird als Schaltwerk betrachtet.



Analysieren Sie das Schaltwerk und lösen Sie dabei die folgenden Aufgaben:

a) Um welchen Automatentyp handelt es sich ? - Begründen Sie Ihre Antwort !

Es handelt sich um einen Mealy-Automaten. Aus dem Zustandsgraphen erkennt man, dass der Automat übergangsorientiert ist, da der Ausgangsvektor an den Kanten neben dem Eingangsverktor angegeben ist.

(2)

b) Erstellen Sie die Zustandsfolgetabelle !

$Z(t_n)$	X_2	X_1	$Z(t_{n+1})$	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(3)

c) Geben Sie die Übergangs- und Ausgangsfunktion in minimaler DNF an.

Das Karnaugh-Diagramm der Ausgangsfunktion wird in der folgenden Abbildung dargestellt:

	X_1			
0	1	0	1	
1	0	1	0	$Z(t_n)$
		X_2		

Es ist keine Minimierung möglich. Die DNF lautet:

$$Y = \overline{Z(t_n)}\,\overline{X_2}X_1 \vee \overline{Z(t_n)}\,X_2\overline{X_1} \vee Z(t_n)\,\overline{X_2}\,\overline{X_1} \vee Z(t_n)\,X_2X_1$$

Das Karnaugh-Diagramm der Übergangsfunktion ist in der folgenden Abbildung dargestellt.

	X_1			
0	0	1	0	
0	1	1	1	$Z(t_n)$
		X_2		

Die minimierte DNF lautet:

$$Z(t_{n+1}) = X_2X_1 \vee Z(t_n)\,X_2 \vee Z(t_n)\,X_1$$

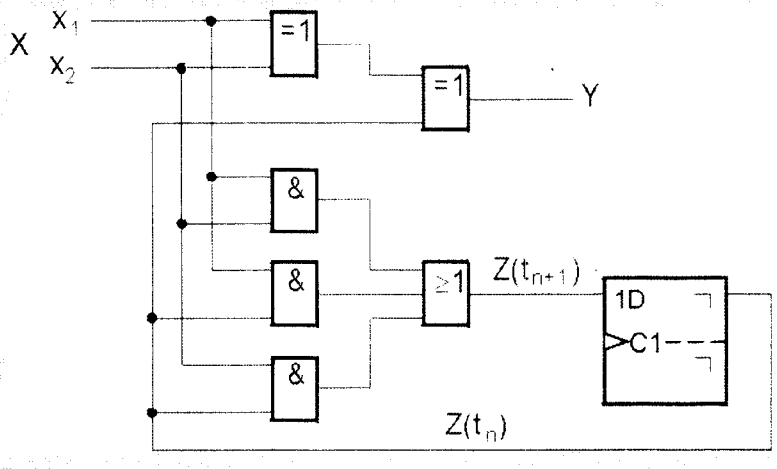
(5)

d) Zeichnen Sie das Schaltwerk und benutzen Sie im Schaltnetz Antivalenzglieder. Hinweis: Der Anfangszustand des Speichergliedes soll als 0 angenommen werden !

Um das Schaltwerk zu zeichnen und im Schaltnetz Antivalenzglieder zu benutzen, wird die Ausgangsfunktion Y umgeformt:

$$
\begin{aligned}
Y &= \overline{Z(t_n)}\,\overline{X_2}X_1 \vee \overline{Z(t_n)}\,X_2\overline{X_1} \vee Z(t_n)\overline{X_2}\,\overline{X_1} \vee Z(t_n)\,X_2X_1 \\
&= \overline{Z(t_n)}\,(\overline{X_2}X_1 \vee X_2\overline{X_1}) \vee Z(t_n)\,(\overline{X_2}\,\overline{X_1} \vee X_2X_1) \\
&= \overline{Z(t_n)}\,(X_2 \not\equiv X_1) \vee Z(t_n)\,(\overline{X_2 \not\equiv X_1}) \\
&= Z(t_n) \not\equiv (X_2 \not\equiv X_1)
\end{aligned}
$$

Damit ergibt sich das folgende Schaltwerk:



(5)

Aufgabe 6 (10 Punkte)

Welche der folgenden Aussagen sind richtig ? (3 + 1)[1]

	ja	nein
a) Bei einem **mikroprogrammierten Mikroprozessor** kann der Anwender den Befehlssatz selbst ändern und so seinen Anforderungen anpassen. *Dies ist nur bei den mikroprogrammier**baren** Prozessoren möglich; mikroprogrammiert haben einen vom Hersteller festgelegten Befehlssatz.*	☐	X
b) Die **Arbeitsgeschwindigkeit** moderner Prozessoren wird nur durch die Schaltgeschwindigkeit der verwendeten Transistoren begrenzt. *Bei den modernen Prozessoren wird die Arbeitsgeschwindigkeit wesentlich durch die Laufzeit der Signale auf den Verbindungsleitungen zwischen den Transistoren beschränkt.*	☐	X
c) Eine Speicherverwaltungseinheit wird auch als **MAC** *(Memory Access Control)* bezeichnet. *MAC steht für Multiplier/Accumulator; die Speicherverwaltungseinheit heißt MMU (Memory Management Unit).*	☐	X

d) Welche Funktionen besitzen die beiden folgenden Register eines Mikroprozessors:

Befehlsregister:

Das Befehlsregister enthält stets den gerade in Ausführung befindlichen Befehl.

(1)

Befehlszähler:

Der Befehlszähler zeigt stets auf den nächsten zu ladenden Befehl bzw. Befehlsteil.

(1)

[1] Die Aufgabenteile a bis c werden jeweils mit einem Punkt gewertet. Bei korrekter Beantwortung aller drei Teilaufgaben wird ein zusätzlicher Punkt vergeben.

e) *Traps* (im engeren Sinne) und *Faults* gehören zu den internen / externen *(Nichtzutreffendes streichen!)* Unterbrechungsanforderungen. Sie unterscheiden sich dadurch, daß

Traps...

i.d.R. zum Abbruch des Befehls führen,

(1)

und Faults...

durch eine Wiederholung der Operation behoben werden können.

(1)

f) Die Verwaltung eines großen Registerspeichers in disjunkten Registerbänken hat insbesondere die folgenden Vorteile gegenüber seiner Verwaltung als homogenen Registersatz:

1.:

Es können im Befehl kurze relative Adressen verwendet werden.

2.:

Zur Parameterübergabe zwischen Programmen ist kein physikalsicher Transport von Registerinhalten nötig.

(2)

Aufgabe 7 (15 Punkte)

a) Das Basisadress-Register einer Ausnahmevektor-Tabelle habe den Wert $A000 8C00. Die Tabelle enthalte 256 Einträge. Jeder Eintrag sei 4 Byte lang. Geben Sie für die Interrupt-Vektornummer IVN=22 die Speicheradressen, unter denen die Startadresse der zugehörigen Ausnahme-Behandlungsroutine in der Tabelle abgelegt ist, sowie den Adreßbereich an, der von der Tabelle belegt wird.

Anfangsadresse des Vektors: $A000 8C00 + 4*22 = $A000 8C00 + $58

Größe der Tabelle: 256 * 4 = 1024 = $0400 ⇒

Startadresse: $A000 8C58 – $A000 8C5B

Vektortabelle: $A000 8C00 – $A000 8FFF (2)

b) Ein 16-Bit-Indexregister verfüge über die Möglichkeiten der automatischen Modifikation durch Inkrementierung bzw. Dekrementierung um einen wählbaren Offset n sowie der Skalierung mit einem Faktor m. Das Register besitze den momentanen (Hexadezimal-) Wert $0A80. Vom Befehl werde eine skalierte Adressierung mit Postdekrement mit dem Faktor m=4 und dem Offset n= – 4 ausgeführt. Geben Sie an, welche Speicheradresse durch den Befehl ausgegeben wird und welcher Wert danach im Indexregister steht. (Rechnung muß angegeben werden!)

Rechnung: Adresse = $0A80*4 = $2A00, Postdekrement:$0A80 - 4 = $0A7C

Angesprochene Speicheradresse: $2A00

Indexregister nach Befehlsausführung: $0A7C (3)

c) Die 8-bit-Register R0, R1, R2 eines Mikroprozessors werden mit den folgenden Werten initialisiert: R0 := $E5, R1 := $00, R2 := $00.
Der Prozessor führe danach **8mal** hintereinander das folgende Operationspaar aus:

R0 := <<R0	*logisches Linksschieben um 1 bit*
R1 := R1 + R2 + CF	***Addition mit Carry-Flag***

Welche Werte stehen danach in R0 und R1 ?

R0: $00, da von rechts Nullen nachgezogen werden;

R1: $05 (2)

Welche Zahl gibt der Wert in R1 allgemein an ?

Es wird die Anzahl der ‚1'-Bits im Register R0 berechnet.

(1)

Zum Abschluß der genannten Operationsfolge werde nun zusätzlich die folgende Operation ausgeführt wird:

R1 := R1 ^ #$01 *Und-Verknüpfung mit der Konstanten $01.*

Welcher Wert steht danach in R1 ?

R1: $01 (Maskierung des LSBs vom Inhalt, also $05 = 0000 0101) (1)

Welche Funktion berechnet die so ergänzte Befehlsfolge ?

Die Befehlsfolge berechnet die (gerade) Parität über den Inhalt des Registers R0.

(1)

c) Die 64-Bit-MMX-Register R0, R1 enthalten die folgenden hexadezimalen Werte:

R0 = (7FAD 84AE 59CA FCFF), R1 = (AD5F 01DD F6FA 6C00).

Geben Sie die Werte der Register nach der Ausführung der folgenden MMX-Operationen an. Dabei werde durch den Buchstaben B, W, D, Q angegeben, ob die Registerinhalte als gepackte Bytes, Wörter, Doppelwörter oder Quadwords aufgefaßt werden sollen.

i) Logisches Linksschieben (D):

R0 = (FF5B095C B395F9FE) (1)

ii) Arithmetisches Rechtsschieben (W):

R1 = (D6AF 00EE FB7D 3600) (1)

iii) Vorzeichenbehaftete Addition mit Sättigung (W):

R0 + R1 = (2D02 868B 50C4 68FF) (1)

iv) Vorzeichenlose Addition mit Sättigung (B):

R0 + R1 = (FF FF 85 FF FF FF FF FF) FF: Sättigung (1)

v) Vorzeichenloser Vergleich auf *‚größer als'* (W):

R0 > R1 = (0000 FFFF 0000 FFFF) (1)