Fakultät für Mathematik und Informatik

Lehrgebiet Rechnerarchitektur

Kurs 1608 „Computersysteme I“

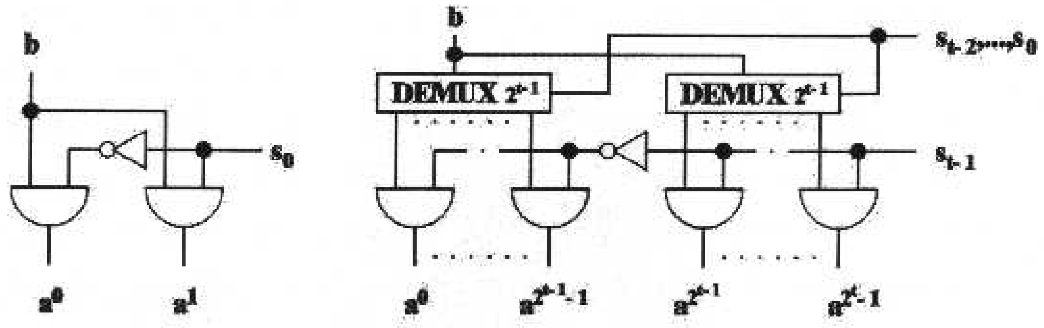
Lösungsvorschläge zu den Aufgaben

der Hauptklausur im SS 2006

(am 12. August 2006)

Aufgabe 1 — Schaltnetze — (25 Punkte)

Bestimmen Sie Kosten und Tiefe eines 2^t-Wege 1-Bit Demultiplexers gemäß der Konstruktion aus der folgenden Abbildung. Stellen Sie dazu die Differenzengleichungen auf und lösen Sie diese mit Lemma 1.28 nach Anpassung des Rekursionsendes, oder direkt.



(a) (b)

Hinweis:

Lemma 1.28 lautet:

Sei $f: \mathbb{N} \to \mathbb{N}$ eine Funktion mit $f(1) = c$ und $f(n) = a * f(n/b) + g(n)$ für alle Potenzen $n = b^k$ von b. Dann gilt

$$f(n) = a^{\log_b n} * c + \sum_{i=0}^{\log_b n - 1} a^i * g(n/b^i)$$

für alle Potenzen n von b.

Für die Kosten eines 2-Wege 1-Bit Demultiplexers gilt nach Abbildung (a)

$$C(DEMUX_2) = 3 .$$

Für die Kosten eines 2^t-Wege 1-Bit Demultiplexers gilt nach Abbildung (b)

$$C(DEMUX_{2^t}) = 2 * C(DEMUX_{2^{t-1}}) + 2^t + 1 ,$$

da er aus zwei 2^{t-1}-Wege Demultiplexern, einem UND-Gatter an jedem der 2^t Ausgänge, und einem Inverter besteht. Löst man direkt, so erhält man

$C(DEMUX_{2^t})$

$$= 2 * C(DEMUX_{2^{t-1}}) + 2^t + 1$$

$$= 4 * C(DEMUX_{2^{t-2}}) + 2 * (2^{t-1} + 1) + 2^t + 1$$

...

$$= 2^{t-1} * C(DEMUX_2t) + 2^{t-2} * (2^2 + 1) + \ldots + 2 * (2^{t-1} + 1) + 2^t + 1$$

$$= 3 * 2^{t-1} + (t-1) * 2^t + \sum_{i=0}^{t-2} 2^i$$

$$= (t+1) * 2^t - 1.$$

In Lemma 1.28 setzt man $n=2^t$, $a=2$, $b=2$, $c=3$, und $g(n)=g(2^t)=2^t+1=n+1$. Da das Rekursionsende hier nicht $f(1)$, sondern $f(2)=c$ ist, verändert sich die Lösungsformel zu

$$f(n) = a^{\log_b n - 1} * c + \sum_{i=0}^{\log_b n - 2} a^i * g(n / b^i).$$

Einsetzen in diese Formel und Lösen der Summe ergibt ebenfalls Kosten $(t+1) * 2^t - 1$.

Die Tiefe eines 2-Wege 1-Bit Demultiplexers beträgt

$$T(DEMUX_2) = 2.$$

Die Tiefe eines 2^t-Wege 1-Bit Demultiplexers beträgt

$$T(DEMUX_{2^t}) = T(DEMUX_{2^{t-1}}) + 1,$$

Sie ergibt sich aus der Tiefe des 2^{t-1}-Wege Multiplexers und den UND-Gatter, da die Tiefe eines Demultiplexers stets mindestens 1 beträgt, und somit nie kleiner als die eines Inverters ist. Die direkte Lösung ergibt

$$T(DEMUX_{2^t}) = T(DEMUX_{2^1}) + t - 1 = t + 1$$

In der veränderten Lösungsformel von Lemma 1.28 setzt man $n = 2^t$, $a = 1$, $b = 2$, $c = f(2) = 2$ und $g(n) = 1$. Man erhält

$$T(DEMUX_{2^t}) = 1^{t-1} * 2 + \sum_{i=0}^{t-2} 1^i * 1 = 2 + t - 1 = t + 1.$$

Damit gilt:

$$C(DEMUX_{2^t}) = (t+1) * 2^t - 1$$
$$T(DEMUX_{2^t}) = t + 1$$

(25)

Aufgabe 2 Zahlendarstellungen (5 Punkte)

Bilden Sie die 8-stellige Zweierkomplement-Darstellung für die Zahl $z=-64$. Welche Zahl wird durch die Zweierkomplement-Darstellung *00110000* dargestellt?

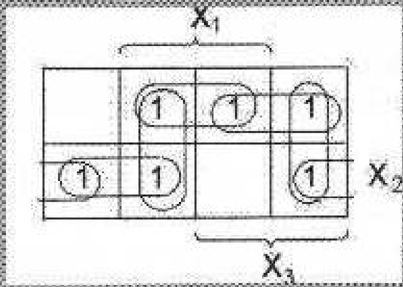
Es gilt $n=7$. Wegen $z<0$ ist $a_7=1$. Es gilt $z+2^n=64$.
Wegen $bin_7(64) = 1000000$ ist die Lösung *11000000*.

Da das oberste Bit den Wert *0* hat, ist die dargestellte Zahl nicht negativ. Die Bits a_4 und a_5 haben den Wert *1*, alle anderen den Wert *0*. Damit ist die dargestellte Zahl $z=2^4+2^5=48$.

(5)

Aufgabe 3 Karnaugh-Diagramme (20 Punkte)

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

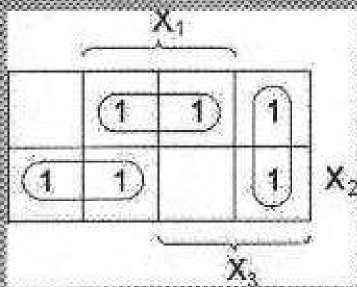


(5)

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 ?

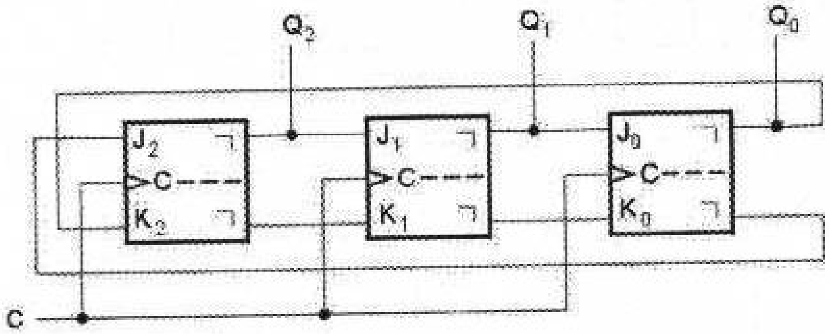
Eine mögliche KNF für das in Aufgabe 1a) gegebene Karnaugh-Diagramm lautet

$$F = (X_1 \vee X_2 \vee X_3) \wedge (\overline{X}_1 \vee \overline{X}_2 \vee \overline{X}_3)$$

Offensichtlich werden bei der Verwendung der KNF weniger Gatter als bei einer Realisierung mittels DNF benötigt. (3)

Aufgabe 4 Johnsonzähler (15 Punkte)

Analysieren Sie den sogenannten Johnsonzähler aus nachfolgend gezeigter Abbildung.

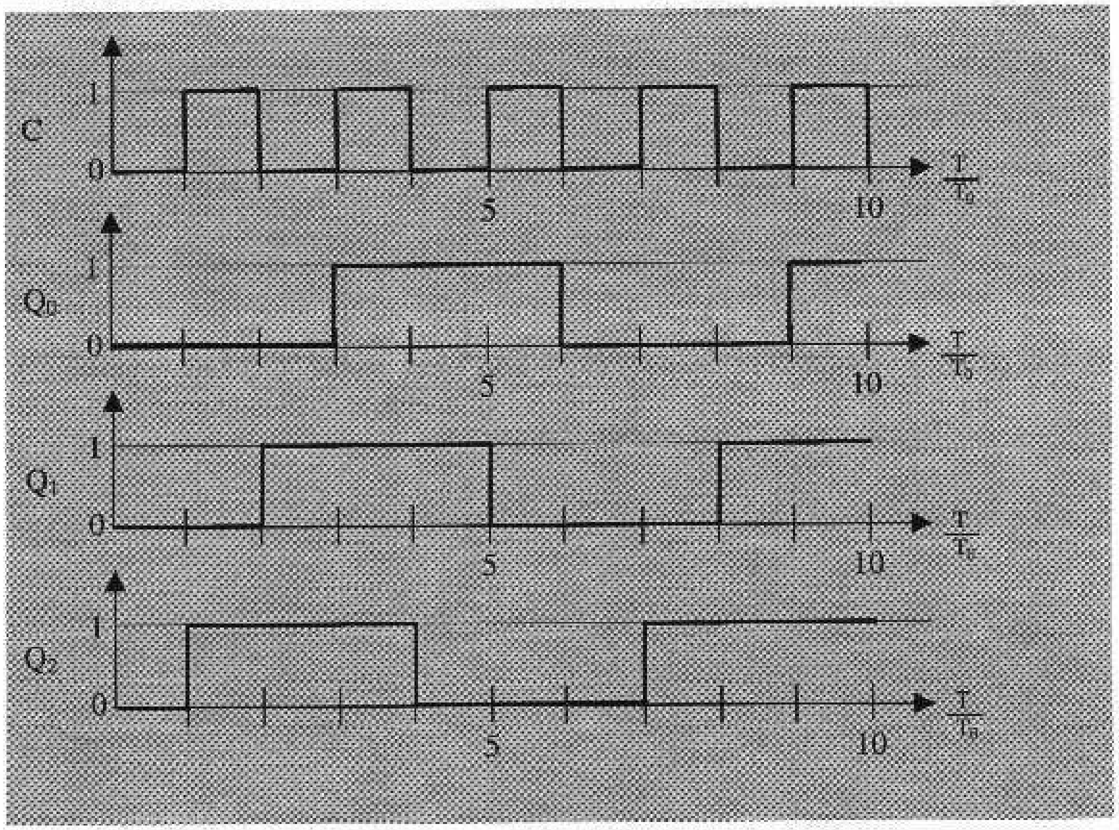


a) Bestimmen Sie ausgehend vom Startzustand (000) den Zählzyklus. Geben Sie eine Folgezustandstabelle an !

t_n									t_{n+1}		
Q_0	Q_0	Q_2	J_0	K_0	J_1	K_1	J_2	K_2	Q_0	Q_1	Q_2
0	0	0	0	1	0	1	1	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	1
0	1	1	1	0	1	0	1	0	1	1	1
1	1	1	1	0	1	0	0	1	1	1	0
1	1	0	1	0	0	1	0	1	1	0	0
1	0	0	0	1	0	1	0	1	0	0	0

Der Zählzyklus lautet : 0, 1, 3, 7, 6, 4, 0, ... (9)

b) Vervollständigen Sie das folgende Impulsdiagramm !



(6)

Aufgabe 5 Schaltwerkssynthese (20 Punkte)

Ein spezieller Zählcode ist der Gray-Code. Er ordnet die Zahlen so, daß bei jedem Zählschritt nur eine Stelle der Dualzahl verändert wird. Die folgende Tabelle zeigt einen Gray-Code für 3-stellige Dualzahlen. Nach der 100 folgt wieder die 000.

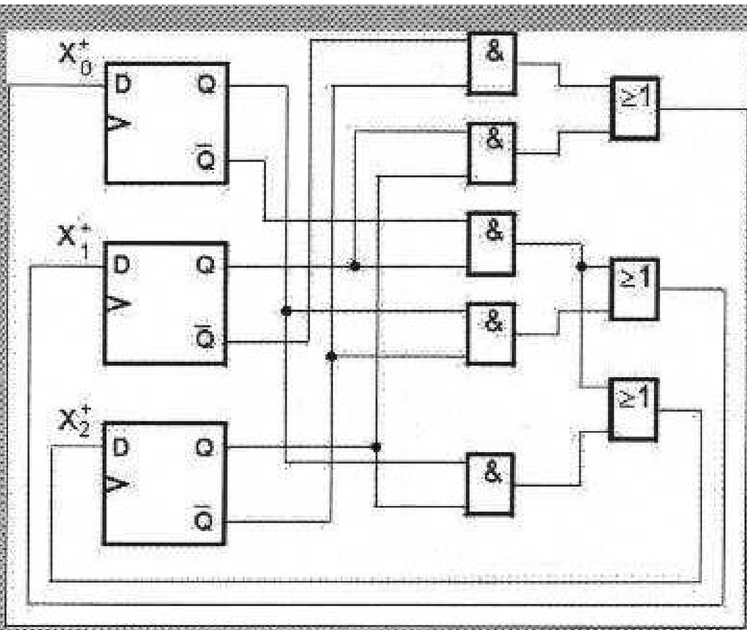
X_2	X_1	X_0
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Konstruieren Sie ein Schaltwerk für diesen Gray-Code, das nur aus 3 D-Flipflops sowie aus UND-Gattern, ODER-Gattern und Invertern besteht.

$$X_0^+ = \overline{X_0}\,\overline{X_1}\,\overline{X_2} \vee X_0\overline{X_1}\,\overline{X_2} \vee \overline{X_0}X_1X_2 \vee X_0X_1X_2 = \overline{X_1}\,\overline{X_2} \vee X_1X_2$$

$$X_1^+ = X_0\overline{X_1}\,\overline{X_2} \vee X_0X_1\overline{X_2} \vee \overline{X_0}X_1\overline{X_2} \vee \overline{X_0}X_1X_2 = X_0\overline{X_2} \vee \overline{X_0}X_1$$

$$X_2^+ = \overline{X_0}X_1\overline{X_2} \vee \overline{X_0}X_1X_2 \vee X_0\overline{X_1}X_2 \vee X_0X_1X_2 = \overline{X_0}X_1 \vee X_0X_2$$



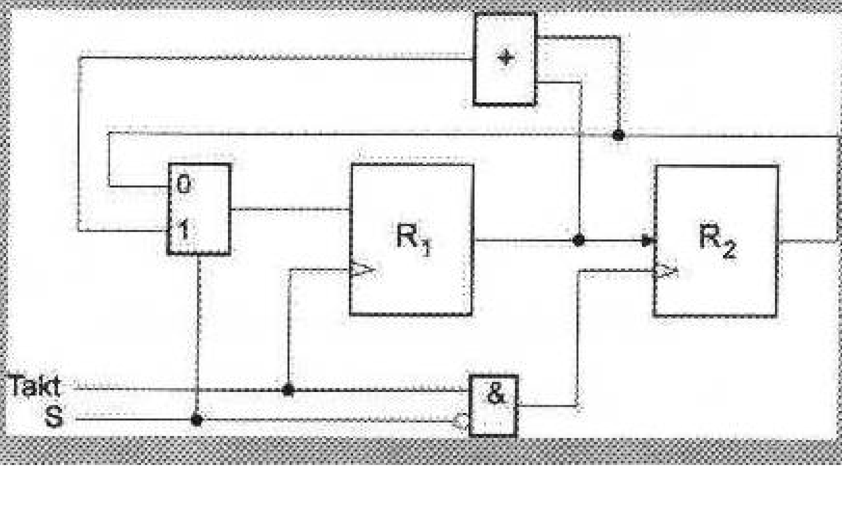
(20)

Aufgabe 6 — Operationswerk (10 Punkte)

Skizzieren Sie ein Operationswerk mit zwei Registern R_1 und R_2, die aus Master-Slave-D-Flipflops aufgebaut sind. Durch eine Steuervariable *S* soll es möglich sein, im darauffolgenden Taktzyklus eine der beiden folgenden Mikrooperationen auszuführen.

- *S*=0: Tausche die Registerinhalte von R_1 und R_2.
- *S*=1: Addiere die Registerinhalte von R_1 und R_2 und schreibe die Summe in das Register R_1. Das Register R_2 soll dabei unverändert bleiben.

Der Aufbau des Operationswerks ist unten dargestellt. Da das Register R_2 bei *S*=1 nicht verändert werden darf, muss der Takt durch ein UND-Schaltglied maskiert werden. Der Takt wird nur für *S*=0 weitergeleitet. In diesem Fall übernimmt das Register R_2 den Inhalt von Register R_1. Gleichzeitig wird über den Multiplexer vor R_1 das Register R_2 zum Einspeichern in R_1 ausgewählt. Für *S*=1 wird die Summe von R_1 und R_2 ausgewählt und von R_1 übernommen.



(10)

Aufgabe 7 **Befehlsholphase** **(5 Punkte)**

Ergänzen Sie den folgenden Lückentext:

Zu Beginn der Befehlsholphase steht die Speicheradresse des nächsten zu verarbeitenden Befehls im Programmzähler . Diese Adresse wird auf den Adressbus gelegt. Nachdem der Speicher den Opcode des adressierten Befehls auf dem Datenbus zur Verfügung gestellt hat, läd der Prozessor diesen in das Befehlsregister und erhöht den Wert des Programmzählers um 1. Das Steuerwerk setzt den eingelesenen Befehl in eine Folge von Steuerworten um, die die weitere Ausführung dieses Befehls steuern.