1701 HK 99

Aufgabe 1: Boolesche Algebra (20 Punkte)

Prüfen Sie algebraisch (d.h. ohne Verwendung einer Wertetabelle) unter ausschließlicher Verwendung der Operatoren Konjunktion, Disjunktion und Negation, ob die unten aufgeführten Beziehungen wahr sind. Das Symbol $\oplus$ steht dabei für die Antivalenz.

1. $A \oplus B = B \oplus A$ (2 Punkte)
2. $A \oplus (B \oplus C) = (A \oplus B) \oplus C$ (3 Punkte)
3. $A \wedge (B \oplus C) = (A \wedge B) \oplus (A \wedge C)$ (5 Punkte)
4. $A \oplus (B \wedge C) = (A \oplus B) \wedge (A \oplus C)$ (5 Punkte)
5. $(A \oplus B) \oplus AB = A \vee B$ (5 Punkte)

Aufgabe 2: Normalformen (20 Punkte)

Gegeben sei die folgende Funktion:

$$\begin{aligned} f : \{0,1\}^3 &\rightarrow \{0,1\} \\ (x_2, x_1, x_0) &\mapsto (x_0 x_1 \oplus x_0 x_2) \oplus x_1 x_2 \end{aligned}$$

1. Geben Sie die (kanonische) DNF (disjunktive Normalform) zu f an (4 Punkte).
2. Geben Sie die (kanonische) KNF (konjunktive Normalform) zu f an (8 Punkte).
3. Geben Sie eine kurze DNF mit paarweise disjunkten Paaren zu f an. Benutzen Sie dazu den Shannon'schen Entwicklungssatz (8 Punkte).

Aufgabe 3: Optimierung (20 Punkte)

Gegeben sei die folgende Funktion:

$$\begin{aligned} f : \{0,1\}^4 &\rightarrow \{0,1\} \\ (x_3, x_2, x_1, x_0) &\mapsto \begin{cases} 1 & \text{falls } \langle x_3, x_2, x_1, x_0 \rangle \text{ prim} \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

Dabei soll $\langle x_3, x_2, x_1, x_0 \rangle$ die durch die Bitfolge (x_3, x_2, x_1, x_0) dargestellte Dualzahl kennzeichnen.

1. Geben Sie alle Minterme der Funktion f an (2 Punkte).
2. Geben Sie alle Primterme der Funktion f an, indem Sie das Verfahren von Quine-McCluskey verwenden (10 Punkte).
3. Geben Sie alle Primimplikanten an, indem Sie eine entsprechende Tabelle erstellen (4 Punkte).
4. Geben Sie eine kürzeste DNF (disjunktive Normalform) zu f an. Zeichnen Sie dazu die einzelnen Primterme in einem Karnaugh-Diagramm ein (4 Punkte).

Aufgabe 4: Schaltkreise (30 Punkte)

In modernen Mikroprozessoren (z.B. Intel Pentium MMX) werden spezielle Befehle zur Beschleunigung von Multimedia-Anwendungen eingesetzt. Ein Beispiel eines solchen Befehls ist die parallele Addition, wie sie in Abbildung 1 dargestellt ist. Im Falle eines 32-Bit Addierers werden in Abhängigkeit von Steuersignalen entweder zwei 32-Bit, vier 16-Bit, oder acht 8-Bit Dualzahlen addiert. Die Addition der einzelnen Dualzahlen erfolgt dabei unabhängig voneinander, d.h. die Überträge propagieren nicht über die Grenzen zwischen zwei Operanden hinweg.

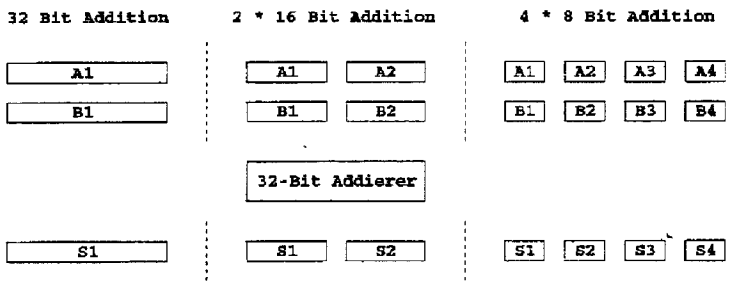

Abbildung 1: parallele Addition

Die parallele Addition wird oft mit der Möglichkeit der 'erschöpfenden' (saturation) Addition kombiniert:

$$
\begin{aligned}
f : \{0,1\}^{2n} &\rightarrow \{0,1\}^{n+1} \\
(a,b) &\mapsto \begin{cases} \mathrm{bin}_{n+1}(a+b) & \text{falls } \langle a\rangle + \langle b\rangle \leq 2^n - 1 \\ (1, \underbrace{1\ldots1}_{n}) & \text{falls } \langle a\rangle + \langle b\rangle > 2^n - 1 \end{cases}
\end{aligned}
$$

Dabei soll $\langle a\rangle$ die durch die Bitfolge a dargestellte Dualzahl $\mathrm{bin}_{n+1}(a)$ die Darstellung von a als $n+1$-stellige Dualzahl kennzeichnen, z. B. ist $\mathrm{bin}_4(7) = 0111$, $\langle 0111\rangle = 7$.

1. Erweitern Sie einen konventionellen (Paralleladdierwerk mit Serienübertrag) 8-Bit Addierer um 'erschöpfende' Addition in Abhängigkeit von einem Steuersignal S (S = 1 : 'erschöpfende' Addition, S = 0: normale Addition) (10 Punkte).

2. Geben Sie den Schaltplan eines 32-Bit Addierers mit Serienübertrag an, der die oben beschriebene parallele Addition in Kombination mit 'erschöpfender' Addition in Abhängigkeit von 3 Steuersignalen (s_2, s_1, s_0) unterstützt. Greifen Sie dabei auf den im Teil 1) entwickelten 8-Bit Addierer zurück. Die Bedeutung der einzelnen Steuersignale ist in der untenstehenden Tabelle angegeben. Das Übertragssignal des Addierers soll genau dann 1 sein, wenn in einem oder mehreren der Teilergebnisse ein Übertrag entsteht bzw. entstehen würde (20 Punkte).

s_2	s_1	s_0	Befehl
0	0	0	32 Bit, normale Addition
0	0	1	2 * 16 Bit, normale Addition
0	1	0	unbenutzt
0	1	1	4 * 8 Bit, normale Addition
1	0	0	32 Bit, 'erschöpfende' Addition
1	0	1	2 * Bit, 'erschöpfende' Addition
1	1	0	unbenutzt
1	1	1	4 * 8 Bit, 'erschöpfende' Addition

Aufgabe 5: Schaltwerke (30 Punkte)

Im Kurstext wurde das Gewicht eines Bitmusters (Anzahl der Einsen) eingeführt. Neben dem Gewicht ist auch die Anzahl der führenden Nullen von Bedeutung, der sog. *leading zero count*. Die Anzahl der führenden Nullen für das Bitmuster 0010 ist z.B. 2, falls das höchstwertigste Bit links steht.

1. Wieviele Bits werden benötigt, um die Anzahl der führenden Nullen eines n-Bit langen Bitmusters als Dualzahl darzustellen ? Begründen Sie Ihre Antwort (5 Punkte).

2. Entwickeln Sie ein Schaltwerk, welches die Anzahl der führenden Nullen des in einem n-Bit Register gespeicherten Bitmusters berechnet und das Ergebnis in einem Register speichert. Das Ende der Berechnung soll zum frühestmöglichen Zeitpunkt erfolgen und durch die Aktivierung eines zusätzlichen Ausgangs O angezeigt werden. Für die Konstruktion stehen Ihnen Register, Flip-Flops, Addierer sowie einzelne Gatter zur Verfügung. (20 Punkte).

3. Anstelle eines Schaltwerks kann die Anzahl der führenden Nullen auch mit einer Schaltfunktion berechnet werden. Stellen Sie die Wertetabelle dieser Funktion für den Fall $n = 4$ auf. Welche Vor- und Nachteile hat die Verwendung dieser Schaltfunktion gegenüber dem oben beschriebenen Schaltwerk (5 Punkte) ?

Aufgabe 6: Zustandsautomaten (30 Punkte)

Betrachten Sie das durch den Übergangsgraphen in Abbildung 2 gegebene synchrone Schaltwerk. Das Schaltwerk besitzt die Zustände Z_i ($i = 0, 1, 2, 3, 6, 7$) und verfügt sowohl über einen Eingang X als auch einen Ausgang Y. Die Beschriftung der einzelnen Kanten ist in der Form X/Y angegeben, wobei 'd' für 'don't care steht. Der Zustand 3 ist der Startzustand des Schaltwerks. Realisieren Sie das oben beschriebene Schaltwerk mit Hilfe von JK-Flip-Flops. Die Kodierung der einzelnen Zustände soll dabei der Darstellung des Zustands als duale Zahl entsprechen, d.h. der Zustand 3 soll als 011 kodiert werden. Nutzen Sie die Tatsache aus, das das Schaltwerk immer im Zustand 3 startet.

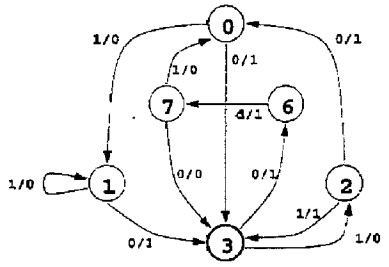

Abbildung 2: Übergangsgraph Aufgabe 6

1. Erstellen Sie anhand des Übergangsgraphen die Wertetabelle für die Steuersignale der JK-Flip-Flops. Kodieren Sie dazu die einzelnen Zustände als Dualzahlen, d.h. sie benötigen insgesamt 3 JK-Flip-Flops. Nutzen Sie bei der Kodierung der Steuersignale die Möglichkeit zur Optimierung aus ! (10 Punkte).

2. Geben Sie für die Steuersignale jeweils eine kürzeste DNF an. Die Optimierung soll dabei für jedes Steuersignal einzeln erfolgen und die Unvollständigkeit der Wertetabelle ausnutzen (20 Punkte).