Technische Informatik I: 7. Übungsblatt

20. Unter Verwendung von vier JK-Flipflops ist ein synchroner 4-Bit-Binärzähler zu entwerfen, der über einen Takteingang C und vier Ausgänge $Q4$, $Q3$, $Q2$ und $Q1$ verfügt. Mit der ansteigenden Taktflanke zählt der Zähler die an $Q4 \dots Q1$ ausgegebene Binärzahl um eins hoch. Hat er den Wert 9 erreicht, so stellt er sich bei der nächsten Taktflanke auf 0 zurück. Verwendung finden solche Zähler überall in der Digitaltechnik; eine verbreitete Anwendung sind Digitaluhren.

    a. Vervollständigen Sie die folgende Zustandstabelle der Zählers (die Indizes beziehen sich bei allen Variablen auf das i-te Flipflop; ein hochgestelltes „+“ gibt den Folgezustand zum Zeitpunkt $k+1$ an). Tragen Sie überall dort ein „X“ ein, wo der Wert des entsprechenden Flipflop-Eingangs keine Rolle spielt.

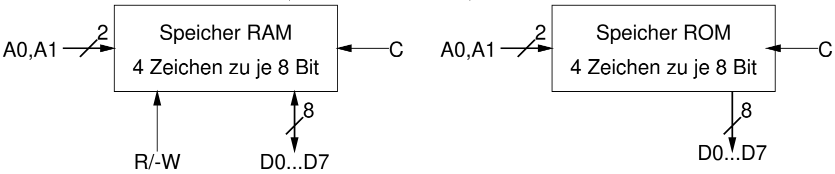
$Q4$	$Q3$	$Q2$	$Q1$	$J4$	$K4$	$J3$	$K3$	$J2$	$K2$	$J1$	$K1$	$Q4^+$	$Q3^+$	$Q2^+$	$Q1^+$
L	L	L	L									L	L	L	H
L	L	L	H									L	L	H	L
L	L	H	L									L	L	H	H
L	L	H	H									L	H	L	L
L	H	L	L									L	H	L	H
L	H	L	H									L	H	H	L
L	H	H	L									L	H	H	H
L	H	H	H									H	L	L	L
H	L	L	L									H	L	L	H
H	L	L	H									L	L	L	L
H	L	H	L									X	X	X	X
H	L	H	H									X	X	X	X
H	H	L	L									X	X	X	X
H	H	L	H									X	X	X	X
H	H	H	L									X	X	X	X
H	H	H	H									X	X	X	X

    b. Geben Sie die Schaltfunktionen für die vier Flipflops an, die zu den in der Tabelle angegebenen Zustandswechseln führen.

    c. Vereinfachen Sie diese mit Hilfe eines Karnaugh-Diagramms oder Quine-McClusky.

    d. Zeichnen Sie die Schaltung auf.

21. Es stehe ein Schreib-/Lesespeicher RAM (random access memory) zur Verfügung, der vier 8-Bit Wörter aufnehmen kann (siehe Bild links).



Seine Funktion ist einfach zu erklären: Über die beiden Adressierungsleitungen $A0$ und $A1$ wird eine aus vier Speicherzellen ausgewählt, in welche über die acht Datenleitungen $D0 \dots D7$ mit der ansteigenden Flanke von C ein 8-bit-Wort eingeschrieben oder ausgelesen werden kann. Die „Schreibrichtung" wird dabei von der Steuerleitung $R/\neg W$ (Read/¬Write) vorgegeben.

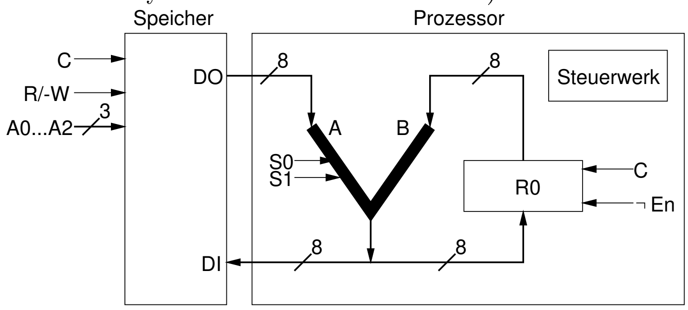
Es sei ferner ein Nur-Lesespeicher ROM (read-only memory) gegeben (Bild rechts), der wie das RAM über vier adressierbare Zellen verfügt, die aber nur gelesen werden können.

Entwerfen Sie mit den Ihnen bekannten Bausteinen (Gatter, Speicher, Zähler, Tristate etc.) und unter Verwendung des seriellen Addierwerks aus Aufgabe 12 ein sequentielles Steuerwerk, das alle Steuer- und Datenleitungen der abgebildeten Speicher und des Addierwerks bedient. Dieses Steuerwerk soll dafür sorgen, daß die an RAM-Adresse 0 und ROM-Adresse 0 stehenden Worte addiert und das Ergebnis an RAM-Adresse 1 gespeichert wird (ohne Übertrag). Danach soll sich der Ablauf für zwei Operanden an den Adressen 2 RAM und 2 ROM wiederholen und das Ergebnis an RAM-Adresse 3 abgelegt werden.

22. Entwerfen Sie als Grundlage eines sogenannten Registers einen Baustein, der über zwei Steuereingänge C (Clock) und ¬En (enable), einen Dateneingang DI (Data In) für ein Bit und einen Datenausgang DO (Data Out) für ein Bit verfügt. Seine Funktion ist einfach anzugeben: Solange an den Steuereingang ¬En der Pegel H angelegt wird, bleibt der Pegel von DO erhalten, d.h. der Baustein speichert diesen Pegel über beliebig viele Taktzyklen, unabhängig vom Zustand des Eingangs DI. Sobald ¬En jedoch L führt, wird der Pegel von DI mit der aufsteigenden Flanke von C in den Baustein übernommen und bleibt von diesem Einschreibvorgang bis zur nächsten Änderung (durch Anlegen der entsprechenden Pegel an ¬En und DI) beibehalten.

Verwenden Sie als Grundlage des Bausteins ein RS-Flipflop, dessen Eingänge sie so beschalten, daß sein Verhalten dem des aus der Vorlesung bekannten Verzögerungsgliedes entspricht. Führen Sie eine Rückkopplung ein, die dafür sorgt, daß bei ¬En $= H$ der Ausgabewert wieder in das Verzögerungsglied eingekoppelt wird.

23. Die Inhalte der Speicherzellen des RAM nach Aufgabe 21 sollen nunmehr mit einem Prozessor verbunden werden, dessen Verknüpfungsglied die ALU nach Aufgabe 15 ist (siehe Bild; hier wurde eine vereinfachte Symbolik für die ALU benutzt).



Da aus dem Speicher mit 8 Datenleitungen nicht gleichzeitig zwei Speicherplätze ausgelesen werden können, die ALU aber zwei Operanden verknüpfen soll, wird wie folgt verfahren: Zuerst wird der erste Operand (an Adresse Adr0) aus dem Speicher über den Eingang A der ALU in ein 8-bit-Register R0 übertragen. Dieses Register wird als Akkumulator bezeichnet und besteht aus acht parallelgeschalteten Bausteinen nach Aufgabe 22.; es ist mit seinen Ausgängen mit dem Eingang B der ALU verbunden. Danach wird ein zweiter Operand an Adresse Adr1 aus dem Speicher ausgelesen, d.h. der Inhalt der Zelle an Adr1 wird an den Eingang A der ALU gelegt. Die Verknüpfung in der ALU findet asynchron statt, beim nächsten Takt wird das Ergebnis entweder in den Speicher oder in das Register R0 geschrieben. Der Speicher hat daher im Gegensatz zu dem aus Aufgabe 21 einen vom Ausgang getrennten Eingang, weil der Ausgang des Speichers niemals direkt mit dem Ausgang der ALU verbunden werden darf. Ansonsten entspricht die Funktion dieses Speichers exakt der desjenigen aus Aufgabe 21.
Es ist ein Steuerwerk für alle Steuer- und Adreßleitungen zu entwerfen, das Operationen auf Speicherinhalten in der folgenden Weise zuläßt:

i. Hole Operand O1 von Adr0 und speicher in R0
ii. Verknüpfe den Operanden O2 aus Adr1 mit R0
iii. Speichere Ergebnis an Adr2 ab (alternativ: in R0)

In gebräuchlicher Notation würde dies wie folgt aufgeschrieben:
<Oper>[Adr0], [Adr1], [Adr2]
Dabei ist für <Oper> eine arithmetische oder logische Operation, wie z.B. AND einzusetzen; die eckigen Klammern bedeuten, daß der Inhalt der Speicherzelle an der spezifizierten Adresse als Operand zu benutzen ist, nicht der Zahlenwert der Adresse selber.

a. Welche sehr einfache Operation muß die ALU hier durchführen können, die in Aufgabe 14 nicht vorgesehen ist? Definieren Sie einen Operationscode dafür.
b. Zeichnen Sie den Automatengraphen des Steuerwerks mit allen Ausgaben für ¬En, S0, S1, A0, A1, R/¬W.
c. Geben Sie die Schaltung des Steuerwerks an, das mit einer Operation Ihrer Wahl zunächst die Inhalte von Adr0 und Adr1 verknüpft, in R0 speichert und danach R0 mit dem Inhalt von Adresse Adr2 verknüpft, um das Ergebnis an Adr3 zu speichern.