RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHRSTUHL
INFORMATIK XI
SOFTWARE FÜR
EINGEBETTETE
SYSTEME

Professor Dr.-Ing. Stefan Kowalewski
Hilal Diab, M.Sc.
Kamal Barakat, M.Sc
Dipl.-Inform. Dominik Franke

Aachen, 08. Januar 2010
SWS: V4/Ü2, ECTS: 7

Einführung in die Technische Informatik

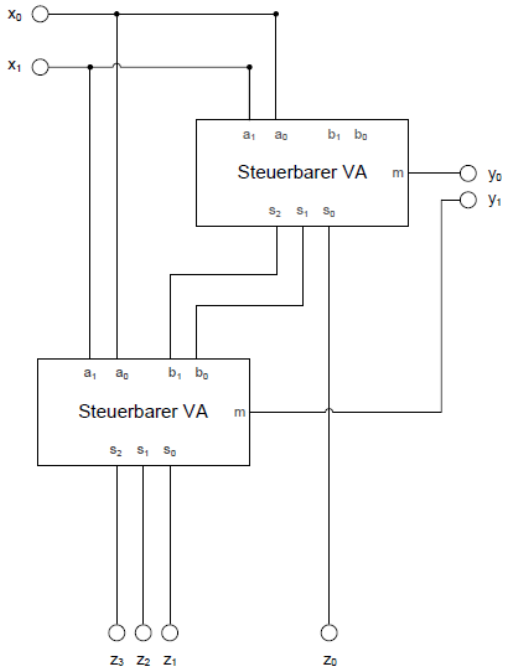
WS 2009/2010

Blatt 10: Addierwerke, Latches

Ihre Lösung zu den mit ($\star$) gekennzeichneten Übungen sollen Sie am **15.01.2010** in der Übung abgeben. Die Bearbeitung der Aufgaben in Lerngruppen von etwa drei oder vier Personen ist sinnvoll. Bitte geben Sie nur eine Lösung pro Lerngruppe ab.

Aufgabe 1: Digitale Schaltungen

Betrachten Sie die gegebene Schaltung



Hinweis: Die beiden steuerbaren Volladdierer besitzen zwei Eingänge (a_1, a_0) und (b_1, b_0), die entsprechend der Vorlesung addiert werden, falls der Steuereingang $m = 1$ ist. Eingang (a_1, a_0) wird abgeschaltet (d.h. $a_1 = a_0 = 0$), sobald der Steuereingang $m = 0$ ist. Das Ergebnis wird jeweils in (s_2, s_1, s_0) geschrieben.

a) Stellen Sie die Wertetabelle für die Schaltfunktion $F((x_1, x_0), (y_1, y_0)) = (z_3, z_2, z_1, z_0)$ auf.

b) Welche Funktion wird durch F realisiert?

c) Formulieren Sie mit eigenen Worten die Funktionsweise der Schaltung.

Aufgabe 2: Addierwerke

a) Demonstrieren Sie die Arbeitsweise eines Parallel-Addierwerks, eines Serien-Addierwerks und eines von-Neumann-Addierwerks (jeweils 4-Bit-Addierwerke) für die nacheinander ausgeführten Berechnungen 4+10, 13+5 und 10+10, indem Sie die Inhalte vorhandener Delays schrittweise in einer Tabelle protokollieren.[1]

S	U	$P_3P_2P_1P_0$	dezimal	$A_3A_2A_1A_0$	dezimal(inkl. U)
0	0	0000	0	0000	0
1	0	0100	4	1010	10
...	...	....	...	...	...

b) Nennen Sie für jedes der drei Addierwerke mindestens einen Vor- und einen Nachteil.

c) Beschreiben Sie die Funktionsweise eines Ripple-Carry-Adders und berechnen Sie die Schaltzeit. Gehen Sie hierfür von einem 12-Bit Addierwerk aus und nehmen Sie eine hypothetische Schaltzeit von 10 psec ($10^{-11}s$) pro Gatter an. Erklären Sie, wie mit Hilfe von Carry-Bypass-Addiernetzen die Dauer der Berechnung verkürzt werden kann und stellen Sie eine Beispielrechnung auf.

[1] Das Status-Delay S entfällt bei Parallel- und Serien-Addierwerk.

Aufgabe 3: Subtraktion zweier Boolescher Zahlen

In dieser Aufgabe sollen Sie Gatterschaltungen für einen Halbsubtrahierer und einen Vollsubtrahierer entwerfen. Ein Halbsubtrahierer subtrahiert zwei Dualziffern und erzeugt deren Differenz und ein Unterlauf-Bit. Wenn der Minuend kleiner ist als der Subtrahend, dann wird das Unterlauf-Bit auf 1 gesetzt, andernfalls auf 0. Ein Vollsubtrahierer besitzt drei Eingaben (Dualziffern) und zwei einstellige Ausgaben. Zwei der Eingaben sind die zu subtrahierenden Ziffern, die dritte Eingabe ist der Unterlauf der niederwertigen Bitposition. Die Ausgaben bestehen aus Differenz und neuem Unterlauf.

a) Stellen Sie Funktionstafeln für beide Bausteine auf. Hierzu ist es u.U. hilfreich, die schriftliche Subtraktion zweier Dezimalzahlen zu betrachten und dieses Verfahren auf die Subtraktion zweier Dualzahlen zu übertragen.

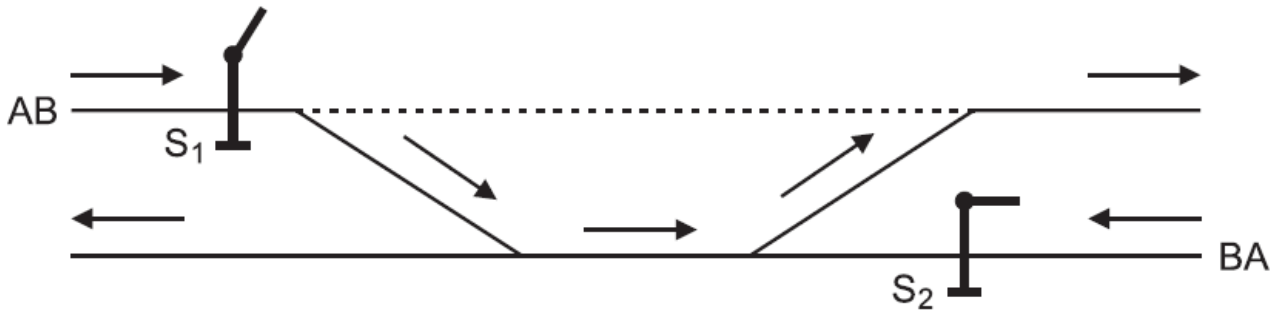
b) Leiten Sie für die Ausgaben beider Bausteine Boolesche Funktionen mit Hilfe von $\cdot, +, ^-$ her.

c) Vereinfachen Sie die Booleschen Funktionen für den Vollsubtrahierer mit Hilfe eines Karnaugh-Diagramms.

d) Zeichnen Sie Schaltnetze für Halb- und Vollsubtrahierer. Nutzen Sie hierbei nur AND-, OR- und NOT-Gatter.

Aufgabe 4: ($\star$)Schaltungsentwurf

Zwischen zwei Orten A und B gibt es eine Schienenverbindung mit zwei Gleiswegen AB und BA. Dabei nutzen von A nach B fahrende Züge den Gleisweg AB, während von B kommende Züge den Gleisweg BA befahren. Bauarbeiten auf einem Teilabschnitt von AB führen jedoch zu einer Ausnahmesituation, die in der folgenden Abbildung illustriert ist:



Danach müssen AB befahrende Züge für einen kurzen Teilabschnitt auf den Gleisweg BA ausweichen. Der Zugverkehr auf diesem Teilabschnitt soll durch zwei einfache Signalanlagen S1 und S2 geregelt werden, die entweder 'freie Fahrt' oder 'unbedingter Halt' signalisieren können. Weiterhin sollte die Regelung derart erfolgen, dass die beiden Signalanlagen in ihren Signalen als Paar (S_1, S_2) aufgefasst stets in einem Zyklus die Zustände $(0,0), (1,0), (0,0), (0,1)$ und wieder $(0,0)$ durchlaufen, wobei 'freie Fahrt' mit 1 und 'unbedingter Halt' mit 0 codiert ist.

a) Entwerfen Sie ein optimiertes (= minimiertes) (Steuer-)Schaltwerk, das die beiden Signalanlagen in der beschriebenen Weise und unter Verwendung der angegebenen Codierung steuert.

b) Demonstrieren Sie die Arbeitsweise Ihres Steuerwerks, indem Sie die Inhalte aller vorhandenen Delays vor und nach jedem Takt für einen kompletten Zyklus protokollieren.

c) Erweitern Sie Ihr Steuerwerk um eine Steuerleitung Z, deren Wert darüber bestimmt, ob der regelnde Betrieb stattfindet oder ob die beiden Signalanlagen ständig auf 'unbedingter Halt' gesetzt sind.

   Hinweis: Dies lässt sich mit zwei zusätzlichen Gattern realisieren lassen.

Aufgabe 5: ($\star$)Addierwerke

Demonstrieren Sie die Arbeitsweise eines Parallel-Addierwerks, eines Serien-Addierwerks und eines von-Neumann-Addierwerks (jeweils 3-Bit-Addierwerke) für die nacheinander ausgeführten Berechnungen 3+5, 7+2, 1+3, indem Sie die Inhalte vorhandener Delays schrittweise in einer Tabelle protokollieren.

S	U	$P_2P_1P_0$	dezimal	$A_2A_1A_0$	dezimal(inkl. U)
0	0	000	0	000	0
...	...	....	...	...	...

Aufgabe 6: ($\star$)Latches

Diese Aufgabe ist eine Erweiterung zu Aufgabe 6, Übungsblatt 9.

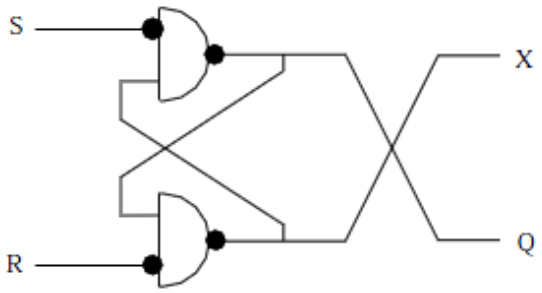
a) Zeichnen Sie Zustandsdiagramme (genau so wie auf Übungsblatt 9, Aufgabe 6) für ein

- AND-Latch
- NAND-Latch
- XOR-Latch

(d.h., ersetzen Sie beide Gatter im Latch entsprechend durch ANDs, NANDs, XORs).

b) Geben Sie für jedes in Aufgabenteil a) behandelte Latch eine entsprechende Beschreibung (ähnlich wie auf Übungsblatt 9, Aufgabe 6) und begründen Sie, ob diese Schaltung als 1-Bit-Speicherbaustein dienen könnte oder nicht.

c) Zeichnen Sie die Zustandsdiagramme für die Schaltung aus der nachfolgenden Abbildung und vergleichen Sie deren Funktionalität mit der Funktionalität des NOR-Latches.



Aufgabe 7: Matlab/Simulink

Benutzen Sie für diese Aufgabe die Datei 'uebung10.mdl', die im L2P unter Lernmaterialien->Matlab zur Verfügung gestellt wird.

a) Konstruieren Sie zwei Halbaddierer.

Hinweis: Konstruieren Sie zuerst einen Halbaddierer, ziehen Sie einen Rahmen um alle Bauteile und wählen Sie dann rechte Maustaste->Create Subsystem. Duplizieren Sie jetzt das entstandene Subsystem.

b) Konstruieren Sie einen Volladdierer mit Hilfe der zwei Halbaddierer aus Aufgabenteil a).

c) Erstellen Sie nun ein 4-Bit Paralleladdierwerk mit Hilfe des Volladdiererbausteins aus Aufgabenteil b).

d) Sie sollen jetzt einen 3-Bit up-Counter entwerfen, also einen Zähler, der immer wieder von 0 aufwärts zählt. Geben Sie zunächst ein Zustandsdiagramm für den Counter an.

e) Konstruieren Sie den Counter mit Hilfe von D-Flip-Flops.

Hinweis: Achten Sie unbedingt darauf, dass am !CLR Eingang des D-Flip-Flops der Signalpegel HIGH=1 anliegt, da das Flip-Flop seinen Inhalt sonst löschen wird. Benutzen Sie hierfür den Simulinkbaustein 'Constant', der in der Datei 'uebung10.mdl' zur Verfügung gestellt wird, und setzen Sie die Eigenschaft 'Constant value' des Bausteins auf 1.

f) Mit Flip-Flops können Speicherbausteine realisiert werden. Ein solcher Speicherbaustein, der in CPU's verwendet wird, ist das Register. Ein einfaches Register hat die Kommandos 'Laden' und 'Lesen'. Entwerfen Sie solch ein Register mit einer Speichergröße von vier Bits. Es soll vier Eingangsleitungen, vier Ausgangsleitungen und einen Eingang für das clock Signal besitzen. Weiterhin soll es zwei Steuerleitungen haben (Load, Read), um die zwei Kommandos auszuführen. Das Verhalten des Registers wird mit folgender Tabelle veranschaulicht:

LOAD	READ	Eingangsleitungen	Registerinhalt	Ausgangsleitungen
LOW	LOW	1011	0100	0000
LOW	HIGH	1011	0100	0100
HIGH	LOW	1011	1011	0000
HIGH	HIGH	1011	1011	1011

Für die Pegel gilt HIGH=1 und LOW=0.