

Aufgabenblatt 9

Ausgabe 14/12/2009, Abgabe bis 04/01/2010 12:00

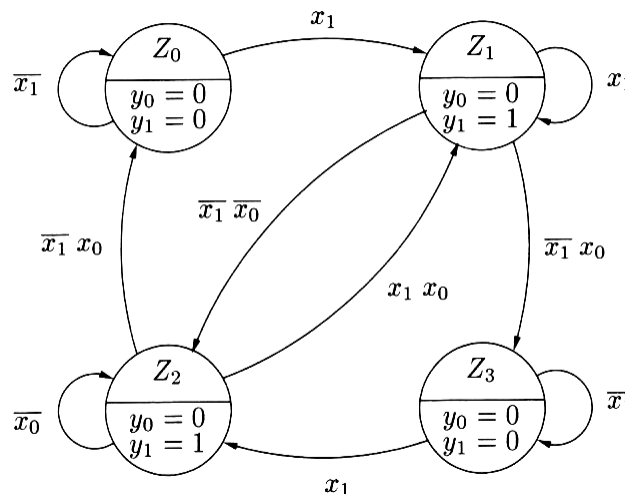
Name(n):

Matrikelnummer(n):

Übungsgruppe:

Aufgabe 9.1 Schaltwerk-Analyse (10+10+10 Punkte)

Wir betrachten das Zustandsdiagramm eines Moore-Schaltwerks mit Eingängen $X = (x_0, x_1)$ und Ausgaben $Y = (y_0, y_1)$ sowie vier Zuständen $\{Z_0, Z_1, Z_2, Z_3\}$. Wir codieren die Zustände binär mit zwei Bits, also $Z = (z_1, z_0)$ und damit $Z_0 = (0, 0)$, $Z_1 = (0, 1)$, $Z_2 = (1, 0)$, $Z_3 = (1, 1)$:



a) Ermitteln Sie aus dem Zustandsdiagramm die zugehörigen Gleichungen für die Übergangsfunktion δ zur Berechnung des Folgezustands Z^+ aus aktuellem Zustand Z und den Eingabewerten X .

Eine Lösungsmöglichkeit ist das Aufstellen der Flussstafel, alternativ das Aufstellen der Übergangs- und Ausgangstabellen und dann die Logikminimierung.

b) Ermitteln Sie die zugehörigen Gleichungen für die Ausgangsfunktion λ zur Berechnung des Ausgangswerts Y als Funktion des aktuellen Zustands Z .

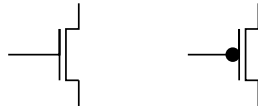
c) Überprüfen Sie den Automaten auf Vollständigkeit (in jedem Zustand ist für jede Eingangsbelegung mindestens ein Übergang aktiv) und Widerspruchsfreiheit (in jedem Zustand ist für jede Eingangsbelegung höchstens ein Übergang aktiv).

Aufgabe 9.2 CMOS-Complexgatter (10+10 Punkte)

Überlegen Sie sich die Struktur eines CMOS-Complexgatters zur Berechnung der folgenden Funktion:

$$y = \overline{(a \vee b) \wedge (c \vee d \vee e)}.$$

a) Beschreiben Sie den Aufbau und zeichnen Sie den Schaltplan für das Gatter (mit den vereinfachten Transistorsymbolen für n-Kanal und p-Kanal Transistoren).



b) Wie viele Transistoren benötigt das Komplexgatter? Wie viele Transistoren würde demgegenüber die herkömmliche Realisierung aus mehreren Gattern gemäß $y = \text{NAND}(\text{OR}(a,b), \text{OR}(c,d,e))$ benötigen?

Aufgabe 9.3 Installation und Test der GNU-Toolchain (10+40 Punkte)

Ziel dieser Aufgabe ist es, dass Sie selbst Zugang zu einem C-Compiler und den zugehörigen Tools haben. Wir empfehlen die *GNU Toolchain* mit dem `gcc` C-Compiler und Werkzeugen. Diese ist auf den meisten Linux-Systemen bereits vorinstalliert, so dass Sie die Befehle direkt ausführen können.

Für Windows-Systeme könnten Sie die sogenannte Cygwin-Umgebung von www.cygwin.com herunterladen und installieren. Im Setup von Cygwin dann bitte den `gcc`-Compiler und die Entwickler-Tools auswählen und mit installieren. Alternativ können Sie auch einen anderen C-Compiler verwenden, Sie müssen sich dann aber die benötigten Befehle und Optionen selbst herausuchen.

Für einen ersten Test tippen Sie bitte das folgenden Programm ab, oder laden Sie sich die Datei `template.c` von der Webseite zur Vorlesung herunter.

Die Syntax von C ist der Syntax von Java sehr ähnlich.

Ändern Sie dann die Datei, indem Sie ihre Matrikelnummer eintragen. Übersetzen Sie das Programm und schauen Sie sich den erzeugten Assembler- und Objektcode an.

```

/* template.c
 * Einfaches Programm zum Test des gcc-Compilers und der zugehörigen Tools.
 * Bitte setzen Sie in das Programm ihre Matrikelnummer ein und probieren
 * Sie alle der folgenden Operationen aus:
 *
 * C -> Assembler: gcc -O2 -S template.c (erzeugt template.s)
 * C -> Objektcode: gcc -O2 -c template.c (erzeugt template.o)
 * C -> Programm: gcc -O2 -o template.exe template.c (erzeugt template.exe)
 * Disassembler: objdump -d template.o
 * Ausführen: template.exe
 */

#include <stdio.h>

int main( int argc, char** argv ) {
    int matrikelnummer = 7654321;

    printf( "Meine Matrikelnummer ist %ld\n", matrikelnummer );
    return 0;
}

```

a) Machen Sie sich mit dem Compiler und den Tools vertraut.

b) Schicken Sie den Quellcode sowie den erzeugten Assemblercode und die Ausgabe des Befehls `objdump -d` (GNU Toolchain) ein. Bei Verwendung anderer Compiler und Tools bitte ebenfalls die entsprechenden Ausgabedateien generieren und einschicken.

Hinweis: der erzeugte Programmcode (`template.exe`) muss nicht mit abgegeben werden. Verschiedene Mailserver halten Mails mit angehängten ausführbaren Programmen wegen eventuell enthaltener Viren automatisch zurück.