



Aufgabenblatt 10

Ausgabe: 18.12., Abgabe: 08.01. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 10.1 (Punkte 5+10+10+15)

Entwurf eines Automaten: Wir betrachten einen 3-bit Zähler mit einem Steuereingang g der zyklisch 6 Werte (= Zustände) durchläuft. Im Fall $g = 0$ wird einfach die binäre Zählfolge, entsprechend $0, 1, 2, 3, 4, 5, 0, 1, \dots$, geschaltet. Bei $g = 1$ soll folgender Gray-Code zyklisch durchlaufen werden: $\{(0, 0, 0), (0, 0, 1), (0, 1, 1), (1, 1, 1), (1, 0, 1), (1, 0, 0)\}$. Die Zustände sind dabei als (z_2, z_1, z_0) codiert.

Zusätzlich hat die Schaltung einen Ausgang p , der eine ungerade Parität zu der Codierung erzeugt.

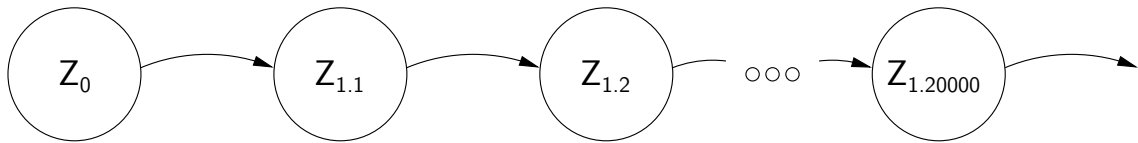
- (a) Handelt es sich um einen Mealy- oder um einen Moore-Automaten? Begründen Sie ihre Antwort kurz.
- (b) Zeichnen Sie das Zustandsdiagramm des Automaten.
- (c) In dem Zustandsdiagramm des Automaten (b) kann es wegen nicht genutzter Codierungen zu Problemen kommen. Beschreiben Sie diese.
Tipp: Achten Sie auf die Vollständigkeit.
- (d) Ergänzen Sie Ihre Automatenbeschreibung durch zusätzliche Zustandsübergänge nach $(0, 0, 0)$ in den zwei Problemfällen und entwerfen Sie den Automaten:
 1. Zustands- und Ausgangstabelle aufstellen, Codierung dabei: g, z_2, z_1, z_0
 2. Übertragen der Werte in KV-Diagramme
 3. Ablesen minimierter schaltalgebraischer Ausdrücke

Aufgabe 10.2 (Punkte 5+15+10+10)

Gekoppelte Automaten: In der Praxis müssen Automaten oft für eine bestimmte Zeit in Zuständen bleiben, beispielsweise für die einzelnen Phasen einer Ampelschaltung.

Wir nehmen jetzt an, dass in dem Automaten aus Aufgabe 10.1 der Zustand $Z_1 = (0, 0, 1)$ für genau 20 000 Takte anliegen soll.

- (a) Eine ad-hoc Lösung wäre natürlich, wie in der Grafik unten skizziert, entsprechend viele Zustände einzuführen, die nacheinander durchlaufen werden.



Dieser Ansatz hat jedoch mehrere Nachteile, weswegen man üblicherweise zwei (oder mehr) *gekoppelte Automaten* einsetzt: einen „Hauptautomaten“, der die eigentliche Funktionalität realisiert, und einen Zähler (trivialer Automat), der für die Wartezeiten sorgt. Zählen Sie kurz (jeweils ein Satz zur Begründung) auf, welche Nachteile dies sein könnten.

- (b) Für die Realisierung des Automaten mit Zähler, soll jetzt folgender, in VHDL codierter, vorderflankengesteuerter Zähler eingesetzt werden:

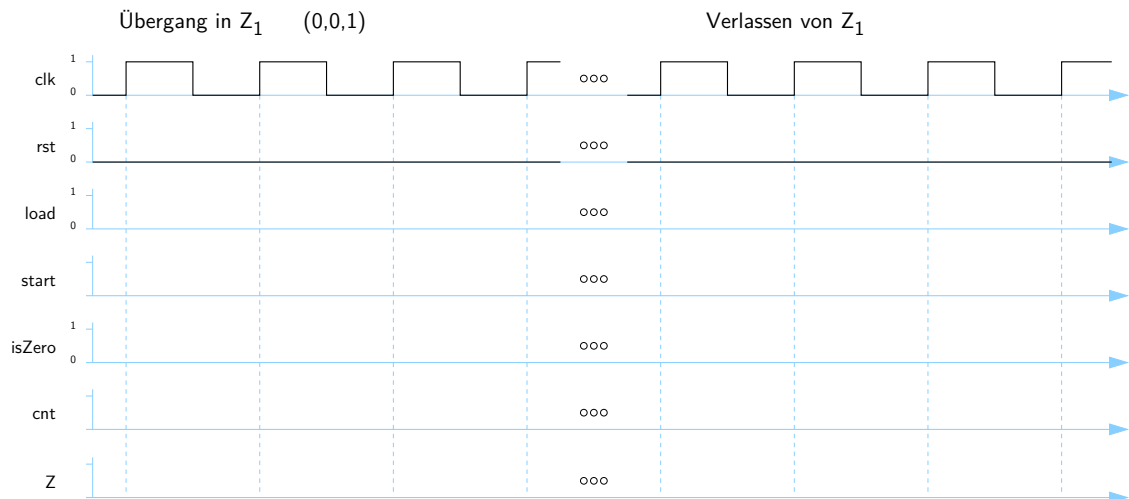
```

1  -- countDown.vhd
2  -----
3  library IEEE;
4  use IEEE.numeric_bit.all;
5
6  entity countDown is
7  generic(bitWd      : positive := 16);
8  port(  clk, rst, load : in bit;
9        start        : in integer range 0 to (2**bitWd)-1;
10       isZero       : out bit);
11 end entity countDown;
12
13 architecture behaviour of countDown is
14   signal cnt      : integer range 0 to (2**bitWD)-1; -- 'Zählvariable'
15 begin
16   cnt_P: process (rst, clk) is
17   begin
18     if rst = '1' then          cnt <= 0;           -- async. Reset: 0
19     elsif rising_edge(clk) then -- Taktvorderflanke
20       if load = '1' then      cnt <= start;       -- Startwert laden
21       elsif cnt > 0 then      cnt <= cnt-1;      -- Runterzählen
22       end if;
23     end if;
24   end process cnt_P;
25
26   isZero <= '1' when cnt = 0 else '0';           -- Ausgang: isZero?
27 end architecture behaviour;
  
```

Die Schaltung funktioniert folgendermaßen:

- Ein asynchrones Reset-Signal (= unabhängig vom Takt) rst setzt den Zähler auf 0.
- Bei aktivierter Steuerleitung load wird der Zähler mit dem Startwert start initialisiert. Ansonsten wird rückwärts gezählt, solange der Zählerstand > 0 ist.
- Als Ausgabe gibt die Leitung isZero an, ob der Zählerstand 0 erreicht wurde.

Der Automat (entsprechend Aufgabe 10.1) arbeitet auch mit vorderflankengesteuerten Flipflops im Zustandsregister Z. Überlegen Sie sich jetzt wann der Automat die Steuerleitungen für den Zähler setzen muss, und wie dabei die Automatenzustände, bzw. Zählerstände, aussehen. Beschreiben Sie den Ablauf von Zuständen und Steuerleitungen kurz als Text und ergänzen Sie das folgende Impulsdiagramm zur Veranschaulichung der Situation. Verzögerungszeiten sollen hier nicht berücksichtigt werden \Rightarrow Signalwechsel und Zustandsübergänge erfolgen direkt an den Taktflanken.



- (c) Welche Probleme ergeben sich, wenn bei allen Zuständen des Zählers zusätzliche Wartezeiten auftreten? Wie muss der ursprüngliche Automat, bzw. das System, geändert werden, wenn beispielsweise in $Z_0 = (0, 0, 0)$ 5 000 Takte gewartet werden muss?
- (d) Beschreiben Sie kurz (nur als Text), wie man vorgehen müsste, wenn der Zähler nicht von einem Startwert aus runterzählt, sondern nach einem Reset auf 0 raufzählt.

Aufgabe 10.3 (Punkte 10+5+5)

Installation und Test der GNU Toolchain: In Vorbereitung auf Kapitel 13 zum x86-Assembler und analog zu den Beispielen in Kapitel 2 ab Folie 90, sollen 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. Am einfachsten geht dies, indem Sie die Linux-, bzw. Dual-boot Rechner in den PC-Poolräumen nutzen.

Ansonsten soll Sie die Aufgabe aber auch motivieren vielleicht auf dem eigenen Rechner mit den Werkzeugen zu „spielen“. Also installieren Sie die entsprechenden Programme (distributionsabhängig).

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

Anmerkung: Keine Angst, die Aufgabe soll zeigen, wie Assemblercode aussieht und Ihnen helfen erste Einblicke zu gewinnen, wie Betriebssystem, (Programm-) Binär-Code und die Hardware zusammenspielen. **Es geht nicht darum Assemblerprogrammierung zu lernen!**

Für einen ersten Test tippen Sie bitte das folgenden Programm ab oder laden Sie sich die Datei `aufg10_3.c` von der Webseite herunter. Passen Sie die Datei an, indem Sie dort ihren Namen und die Matrikelnummer eintragen. Anschließend können Sie das Programm übersetzen und sich den erzeugten Assembler- und Objektcode anschauen.

```

1  /* aufg10_3.c
2  * Einfaches Programm zum Test des C-Compilers und der zugehörigen Tools.
3  * Bitte setzen Sie in das Programm ihren Namen und die Matrikelnummer ein
4  * und probieren Sie alle der folgenden Operationen aus:
5  *
6  * Funktion          Befehl                      erzeugt
7  * -----+-----+-----+
8  * C -> Assembler:  gcc -Og -S aufg10_3.c             -> aufg10_3.s
9  * C -> Objektcode: gcc -Og -c aufg10_3.c         -> aufg10_3.o
10 * C -> Programm:   gcc -Og -o aufg10_3.exe aufg10_3.c -> aufg10_3.exe
11 * Disassembler:   objdump -d aufg10_3.o
12 *                 objdump -d aufg10_3.exe
13 * Ausführen:     aufg10_3.exe
14 */
15
16 #include <stdio.h>
17
18 int main( int argc, char** argv )
19 { int matrikelNr    = 123456;
20   char vorname[32]  = "John";
21   char nachname[32] = "Doe";
22   //char *vorname   = "John";
23   //char *nachname  = "Doe";
24
25   printf("Name: %s %s - Matrikelnr.: %d\n", vorname, nachname, matrikelNr);
26   return 0;
27 }

```

- (a) 🧑¹ Machen Sie sich mit dem Compiler und den Tools vertraut. Probieren Sie die vorgeschlagenen Befehle aus und sehen Sie sich die Ausgaben an.
- (b) Erzeugen Sie eine Textdatei, die die Ausgabe des Programms und ein Listing des Disassemblers enthält. Dies geschieht am einfachsten mit den folgenden Befehlen:

```

./aufg10_3.exe          > loesung10_3.txt
echo "======" >> loesung10_3.txt
objdump -d aufg10_3.o >> loesung10_3.txt

```

Markieren Sie in der Datei an welcher Stelle des Codes: Vorname, Nachname und Matrikelnummer stehen (mit kurzer Begründung). Diese Datei ist als Lösung des Aufgabenteils abzugeben.

¹10 Punkte als Weihnachtsgeschenk ;-)

- (c) In dem Code `aufg10_3.c` sind die Zeilen 22 und 23 auskommentiert. Ändern Sie die Variablen für Vor- und Nachnamen in die zweite Version (Zeiger auf den String, statt char-Array).

Was ändert sich in dem Assembler-Code? Es genügt, die Änderungen (inhaltlich) zu beschreiben, es müssen keine Listings abgegeben werden.