Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

Laborprüfung 'Digitaltechnik' WS 2003/2004, 13.01.2004

Name: _____ Matrikel-Nr.: _____

Erlaubte Hilfsmittel: Erlaubt sind alle stoffrelevanten Hilfsmittel außer Datenträger.

Prüfungsdauer: max. 3 h (1,5 h Vorbereitung, 1,5 h im Labor)

Prüfungswertung: Das Prüfungsergebnis kann „bestanden" oder „nicht bestanden" lauten, Noten werden keine vergeben (Schein).

Testate: Simulation der Schaltung ☐ _____
Funktion der Schaltung ☐ _____

Am Ende der Laborprüfung sind abzugeben:

- Die Aufgabenblätter und die Lösungsblätter, **alle** mit Namen versehen.
- Ein Ausdruck der Datei *ampel.vhd*, sowie ein Ausdruck der Simulation.
- Das Testatblatt mit den Testaten zu allen Versuchen des Digitaltechnik-Labors

1. Beschreibung der Aufgabe

Die Hauptverkehrsstraße eines Dorfes wird von einem Fußgängerweg gekreuzt. Die dort installierte Ampelsteuerung soll mit Hilfe eines Tasters (Ampel-Taster), für Fußgänger auf grün umgestellt werden. Um schnelle Autofahrer von Zeit zu Zeit auszubremsen, soll die Ampelsteuerung in regelmäßigen Zeitabständen auch automatisch für Fußgänger auf grün umgestellt werden.

Die Fußgängerampel zeigt die Signale rot und grün, die Verkehrsampel die Signale rot, gelb und grün. Die Signale zur Ansteuerung dieser fünf Ampellichter werden im folgenden mit *frot, fgruen, vrot, vgelb* und *vgruen* bezeichnet. Im Grundzustand zeigt die Fußgängerampel rot und die Verkehrsampel grün (*frot* und *vgruen* aktiv).

Wenn ein Fußgänger den Taster drückt, soll die Verkehrsampel von grün über gelb auf rot schalten und die Fußgängerampel von rot auf grün. Nach einigen Grünphasen der Fußgängerampel schaltet diese auf rot und die Verkehrsampel auf grün (siehe Tabelle 2-1). Insgesamt werden bei diesem Ablauf 12 Phasen durchlaufen. Wird kein Ampel-Taster betätigt, so soll sich dieses Schaltverhalten nach einer bestimmten Zeit automatisch wiederholen (Automatik-Steuerung).

Die Ampelsteuerung soll mit Hilfe eines Modulo-12-Zählers realisiert werden, der mit einem entsprechend langsamen Takt (ca. 0.2 - 0.5 Hz, NE555, Signalname: *clock_ampel*) arbeitet. Ein nachgeschalteter Decoder setzt die Zählerschritte in die fünf Ampelsignale um.

FACHHOCHSCHULE PFORZHEIM
GESTALTUNG HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT

Digitaltechnik-Labor WS 2003/2004 FB2 Studiengänge EIT/TI

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

Die Automatik-Steuerung der Ampel besteht aus einem 4 Bit Zähler. Bei jedem Überlauf soll der Zähler das Signal *automatik = 1* generieren. Durch das Signal *reset_shs = '1'* soll der Zähler neu gestartet werden. Der Zähler arbeitet mit dem gleichen Takt wie die Ampel-Steuerung (*clock_ampel*).

Das Ausgangssignal des Tasters (*sensor*) bzw. das Ausgangssignal der Automatik-Steuerung (*automatik*) wird einem synchronen Schaltwerk zugeführt. Bekanntlich besteht ein solches Schaltwerk aus einem kombinatorischen und sequenziellen Teil. Dieses Schaltwerk soll eine Selbsthalteschaltung (SHS) realisieren. Die Selbsthalteschaltung besitzt die Eingänge *sensor, automatik*, *reset_shs, clock_sensor* und *reset* (globaler Reset). Als Ausgangssignal ist das Signal *start* vorhanden.

Eine einmal erkannte Anforderung (Eingangssignal *sensor* ='1', bzw. automatik ='1') wird mit dem Ausgangssignal *start* ='1'signalisiert, das auch bestehen bleibt, wenn *sensor* und *automatik* wieder auf '0' gehen (Selbsthaltung). Erst über das Eingangssignal *reset_shs* ='1' (abgeleitet vom vorletzten Zählerschritt des Modulo-12-Zählers) wird die Selbsthalteschaltung wieder in den Grundzustand (*start* ='0') zurückgesetzt. Bevor das Signal *start* dem Modulo-12-Zähler zugeführt werden kann, muss es noch auf den Takt des Modulo-12-Zählers synchronisiert werden. Das synchronisierte Signal *start* wird dann mit *start_sync* bezeichnet.

Für die Selbsthalteschaltung (synchrones Schaltwerk) soll ein Takt von ca. 2,5 MHz (Quarz-Oszillator, Signalname: *clock_sensor*) verwendet werden.

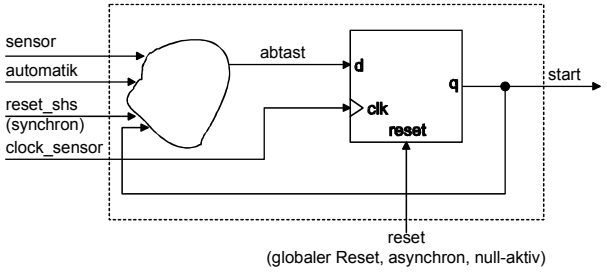


Abbildung 1-1 Selbsthalteschaltung

Hinweise zur Realisierung:

Die Ampelsignale *vrot, vgelb, vgruen, frot, fgruen* sollen mit den bereits ampelförmig angeordneten LEDs auf dem Laborsystem angezeigt werden. Die Ansteuerung dieser LEDs erfolgt 0-aktiv (470 Ω Vorwiderstände sind auf dem Laborsystem bereits vorhanden).

Die Sensorschleife (Signal *sensor*) wird auf dem Laborsystem durch einen der Taster realisiert. Der Schaltwerkstakt *clock_ampel* wird vom Taktgenerator (NE555) zur Verfügung gestellt.

Die gesamte Ampelsteuerung wird im CPLD EPM7064SLC44-10 realisiert. Die Pin-Belegung ist aus Tabelle 3-1 ersichtlich.

Die Zustandsbits des Modulo-12-Zählers (*q3, q2, q1, q0*), der Schaltwerkstakt *clock_ampel* und die Signale *automatik, start* und *start_sync* sollen über acht LEDs der LED-Kette angezeigt werden.

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

2. Vorbereitung auf dem Lösungsblatt

a) Ergänzen Sie auf dem Lösungsblatt die VHDL-Beschreibung der Prozesse für die Selbsthalteschaltung und die Automatik-Steuerung gemäß der obigen Beschreibung.
Wenn die Signale *automatik, sensor* und *reset_shs* gleichzeitig aktiv sind, soll *reset_shs* die höhere Priorität haben.

b) Ergänzen Sie auf dem Lösungsblatt die VHDL-Beschreibung der Prozesse für den Modulo-12-Zähler.

c) Ergänzen Sie auf dem Lösungsblatt die VHDL-Beschreibung des Ampel-Decoders. Dieses Schaltnetz generiert aus dem Zählerstand (*q3*, *q2*, *q1*, *q0)* des Modulo-12-Zählers die Signale *vrot*, *vgelb*, *vgruen*, *frot* und *fgruen*. Dabei wird der folgende Phasenablauf gefordert:

Zählerstand (Phase)	Fußgänger-ampel	Verkehrs-ampel
0	rot	gruen
1	rot	gelb
2	rot	rot
3	rot	rot
4	gruen	rot
5	gruen	rot
6	gruen	rot
7	rot	rot
8	rot	rot
9	rot	rot + gelb
10	rot	gruen
11	rot	gruen

Tabelle 2-1 Phasenablauf

d) Zeichnen Sie den SW-Graphen des Modulo-12-Zählers. Beachten Sie die Vorgaben auf dem Lösungsblatt. Der Modulo-12-Zähler besitzt die Zählerstände 0 bis 11. Im Zählerstand 0 soll der Modulo-12-Zähler anhalten, wenn das Eingangssignal *start_sync* den Wert `'0'` besitzt.

e) Ergänzen Sie auf dem Lösungsblatt die VHDL-Beschreibung des Stimulations-Teils für die Testbench. Das Eingangssignal *sensor* soll folgenden zeitlichen Verlauf besitzen:

Zeit / ns	sensor
0	0
545	1
555	0
3555	1
6555	0

Tabelle 2-2 Testbench-Vorgabe

FACHHOCHSCHULE PFORZHEIM
GESTALTUNG HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT
Digitaltechnik-Labor WS 2003/2004 FB2 Studiengänge EIT/TI

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

3. Durchführung am Laborarbeitsplatz

a) Öffnen Sie im Startmenü den Ordner *Vorlagen*, wechseln Sie in das Unterverzeichnis *dtlt*, und kopieren Sie die VHDL-Textvorlage *ampel.vor* nach *c:\users\ampel.vhd*. Erstellen Sie in *c:\users* ein neues OrCAD-Projekt, und binden Sie die Datei *ampel.vhd* in das Projekt ein.

b) Übernehmen Sie den VHDL-Code von Ihrem Lösungsblatt in die Datei *ampel.vhd.*

c) Erstellen Sie mit Hilfe von OrCAD-Express eine Testbench. Übernehmen Sie die VHDL-Beschreibung des Stimulations-Teils von Ihrem Lösungsblatt in die Testbench.
Überprüfen Sie die Funktionalität Ihres Entwurfs mit dem Simulator von OrCAD-Express.
Insgesamt soll die Simulation 15000 ns dauern, damit auch die Funktion des Signals *automatik* = `'1'` zu sehen ist.
Erstellen Sie einen Ausdruck Ihrer Simulation. Die korrekte Simulation wird testiert.

d) Erstellen Sie mit Hilfe von OrCAD-Express und MAX+plus II die Programmierdatei für den Baustein EPM7064SLC44-10.
Am CPLD soll folgende Pin-Belegung benutzt werden:

Signalname	Pin
fgruen	20
frot	21
vgruen	24
vgelb	25
vrot	26
clock_ampel	43
clock_sensor	2
sensor	40

Signalname	Pin
start_sync_out	27
start_out	28
automatik_out	18
q_out0	11
q_out1	12
q_out2	14
q_out3	16
reset	1

Tabelle 3-1 Pin-Belegung

e) Bauen Sie die Schaltung unter Berücksichtigung der folgenden Vorgaben auf:

Das Sensorausgangssignal *sensor* soll über einen Taster eins-aktiv bedient werden. Für den null-aktiven, globalen Reset (*reset*) benötigen Sie ebenfalls einen Taster. Alle dafür notwendigen Widerstände sollen als Pull-Up-Widerstände verschaltet sein.
Das Taktsignal für die Ampel wird dem Taktgenerator (NE555) auf dem Laborsystem entnommen. Das Taktsignal für die Selbsthalteschaltung soll dem Quarzgenerator auf dem Laborsystem entnommen werden.
Die Ampelsignale *vrot, vgelb, vgruen, frot, fgruen* sollen mit den bereits ampelförmig angeordneten LEDs auf dem Laborsystem angezeigt werden.
Die Zustandsbits der Zähler (*q3, q2, q1, q0*), der Schaltwerkstakt *clock_ampel* und die Signale *start*, *start_sync* und *automatik* sollen über acht LEDs der LED-Kette angezeigt werden. (Die Reihenfolge ist wie angegeben, von links beginnend, einzuhalten.)

f) Konfigurieren (programmieren) Sie den EPM7064SLC44-10 und überprüfen Sie die Funktionalität Ihrer Schaltung. Die korrekte Funktion der Schaltung wird testiert.

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

LÖSUNGSBLATT

a) VHDL-Textvorlage *ampel.vor*

```
-- ampel.vor / ampel.vhd - VHDL-Textvorlage, FH Pforzheim

-- Alle mit _____ gekennzeichneten Felder muessen von Ihnen
-- durch sinnvolle Eintraege ersetzt oder geloescht werden!!!

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

entity ampel is
  port (__________,__________,__________, reset : in std_logic;
        start_out, start_sync_out : out std_logic;
        automatik_out : out std_logic;
        vrot, vgelb, vgruen, frot, fgruen : out std_logic;
        q_out  : out std_logic_vector(__________ downto 0));
end;

architecture behavior of __________ is

  signal q, q_ns: std_logic_vector(__________ downto 0);
  signal a, a_ns: std_logic_vector(__________ downto 0);
  signal start, reset_shs, start_sync, abtast: std_logic;
  signal automatik: std_logic;

begin
```


FACHHOCHSCHULE PFORZHEIM
GESTALTUNG HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT

Digitaltechnik-Labor WS 2003/2004 FB2 Studiengänge EIT/TI

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

```
 -- Modulo-12-Zaehler sequenziell und
 -- Synchronisation von start auf den Ampel-Takt
process (__________,__________) begin
 if (reset = '0') then
      q <= __________;
      start_sync <= __________;
 elsif (__________ and __________ = '1') then
      q <= __________;
      start_sync <= start;
   end if;
end process;



 -- Automatik sequenziell
process (__________, reset) begin
 if (reset = '0') then
      a <= x"0";
  elsif (__________ and __________ = '1') then
      a <= __________;
  end if;
end process;



 -- Selbsthaltung sequenziell
process (__________,__________) begin
 if (__________ = '0') then
      start <= __________;
  elsif (__________ and __________ = '1') then
      start <= __________;
  end if;
end process;
```

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

```
 -- Ueberfuehrungs-Schaltnetz und Decoder der Automatik-Steuerung
process (___________,___________) begin
 a_ns <= unsigned(a) + 1; -- default
 automatik <= '0'; -- default
 if (reset_shs = '1') then
   a_ns <= x"0";
 elsif (a = ___________) then
   automatik <= ___________;
  end if;
end process;



-- Ueberfuehrungs-Schaltnetz Sensor-Selbsthaltung
process (___________,___________,___________,___________) begin
 abtast <= ___________;
 if (___________ = '1') then
   abtast <= ___________;
 elsif (___________ or ___________ or ___________) then
   abtast <= '1';
  end if;
end process;



-- Ueberfuehrungs-Schaltnetz Modulo-12-Zaehler
process (___________, ___________) begin
  q_ns <= ___________ + 1; -- default
  if (___________ = x"0") then
      if (___________ = '1') then
        ___________ <= unsigned(q) + 1;
      else
        ___________ <= q;
      end if;
  end if;
  if (q = ___________) then
    q_ns <= "0000";
  end if;
end  process;



-- Resetsignal fuer die Selbsthalteschaltung
process (q) begin
 reset_shs <= '0';
  if (q = x"A") then
   reset_shs <= '1';
  end if;
end process;
```

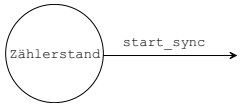

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

```
 -- LED-SN
process (__) begin
  case __ is
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when "_____" => vrot<='__'; vgelb<='__'; vgruen<='__';
                    frot<='__'; fgruen<='__';
    when others => vrot<='0';  vgelb<='1'; vgruen<='1';
                    frot<='0';  fgruen<='1';  --Fehler
  end case;
end process;



 -- Anschluss der Outputs
q_out <= q;
start_out <= ___________;
start_sync_out <= ___________;
automatik_out <= ___________;


end behavior;
```

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

b) SW-Graph des Modulo-12-Zählers:

Legende:



FACHHOCHSCHULE PFORZHEIM
GESTALTUNG
HOCHSCHULE FÜR TECHNIK UND
WIRTSCHAFT

Digitaltechnik-Labor WS 2003/2004 FB2 Studiengänge EIT/TI

Prof. Dr.-Ing. Rainer Dietz
Dipl.-Ing. (FH) Holger Dihlmann
Dipl.-Ing. (FH) Manuel Gaiser

c) VHDL-Testbench-Textvorlage (Stimulus-Teil)

```
-- Alle mit ____ gekennzeichneten Felder muessen von Ihnen
-- durch sinnvolle Eintraege ersetzt oder geloescht werden!!!


-- Place stimulus and analysis statements here

-- Ampel-Takt
process begin
   clock_ampel <= '0'; wait for 100 ns;
   clock_ampel <= '1'; wait for 100 ns;
end process;

-- Sensor-Takt
process begin
   clock_sensor <= '0'; wait for 1 ns;
   clock_sensor <= '1'; wait for 1 ns;
end process;

process begin
   reset <= '__';
   wait for 200ns;
   reset <= '__';
   wait;
end process;

process begin
   sensor <= _____________________________;
   sensor <= _____________________________;
   sensor <= _____________________________;
   sensor <= _____________________________;
   sensor <= _____________________________;
end process;
```