

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

Laborprüfung 'Digitaltechnik' WS 2003/2004, 12.01.2004

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	<input type="checkbox"/>	_____
Funktion der Schaltung	<input type="checkbox"/>	_____

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 Fußgängerzone einer Stadt wird von einer wenig befahrenen Straße gekreuzt. Die dort installierte Ampelsteuerung soll mit Hilfe einer Fahrzeugsensorschleife auf Fußgänger-vorrang 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 grün und die Verkehrsampel rot (*fgruen* und *vrot* aktiv).

Wenn ein Fahrzeug auf die Sensorschleife fährt, soll die Fußgängerampel auf rot und die Verkehrsampel über rot-gelb auf grün schalten. Nach einigen Grünphasen der Verkehrs-Ampel schaltet diese über gelb wieder auf rot und die Fußgängerampel auf grün (siehe Tabelle 2-1). Insgesamt werden bei diesem Ablauf 12 Phasen durchlaufen.

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

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

Das Ausgangssignal der Sensorschleife 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*, *reset_shs* *clock_sensor* und *reset* (globaler Reset). Als Ausgangssignal ist das Signal *start* vorhanden.

Eine einmal erkannte Anforderung (Eingangssignal *sensor* = '1') wird mit dem Ausgangssignal *start* = '1' signalisiert, das auch bestehen bleibt, wenn *sensor* wieder auf '0' geht (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.

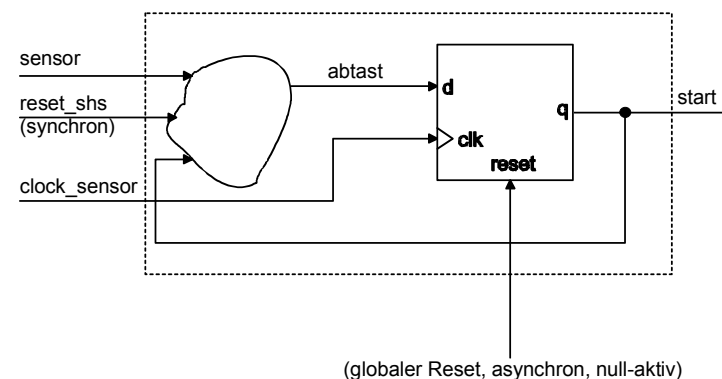


Abbildung 1-1 Selbsthalteschaltung

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 vier Zustandsbits des Modulo-12-Zählers (*q3*, *q2*, *q1*, *q0*), der Schaltwerkstakt *clock_ampel* und die Signale *start* und *start_sync* sollen über sieben 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 gemäß der obigen Beschreibung.
 Wenn die Signale *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	gruen	rot
1	rot	rot
2	rot	rot + gelb
3	rot	gruen
4	rot	gruen
5	rot	gruen
6	rot	gruen
7	rot	gelb
8	rot	rot
9	rot	rot
10	gruen	rot
11	gruen	rot

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

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 *dtit*, und kopieren Sie die VHDL-Textvorlage *ampel.vor* nach *c:\users\lampel.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. Die Simulationsdauer soll 10000 ns betragen. 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
q_out0	11
q_out1	12
q_out2	14
q_out3	16
fgruen	20
frot	21
vgruen	24
vgelb	25
vrot	26
start_sync_out	27
start_out	28
clock_ampel	43
clock_sensor	2
sensor	40
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 vier Zustandsbits des Modulo-12-Zählers (*q3*, *q2*, *q1*, *q0*), der Schaltwerkstakt *clock_ampel* und die Signale *start* und *start_sync* sollen über sieben LEDs der LED-Kette angezeigt werden. (Die Reihenfolge ist wie angegeben, mit der roten LED 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;
        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 start, reset_shs, start_sync, abtast: std_logic;

begin

  -- Modulo-12-Zaehler 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;

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

-- Ueberfuehrungs-Schaltnetz Modulo-12-Zaehler
process (_____, _____) begin
  q_ns <= unsigned(q) + 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 Selbsthaltungeschaltung
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

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;
```