Lab 6

ENGR 2323

Digital Design Lab

Class Outline

- Sequential circuit design using VHDL.
- State memory.
- Design of 2-bit Up and Down Counter.
- Lab 6 deliverables.

Objectives

- Be able to design sequential circuits using a hardware description language such as VHDL.
- Be able to verify sequential circuit design operation using simulations and testing on a FPGA development board.
- Be able to verify sequential circuit design operation using a mixed signal oscilloscope/logic analyzer.

Important Terms/Concepts

- Complex sequential circuit designs can be prototyped rapidly using behavioral architectures. The behavioral design can be done directly from a state diagram, state transition table, or register transfer language description.
- Important terms/concepts include:
  - sequential circuit (state machine)
  - behavioral architecture
  - state transitions, output equations

Sequential Circuit Design

- A behavioral description of a sequential circuit can be based off the state diagram or state transition table. The next state and output equations are not necessary.
- The state transitions are described conditionally inside a process statement and state assignment and outputs are described using concurrent statements in parallel with the process describing the state transitions.
- A single file VHDL design for a sequential circuit has the same outline as for a combinational circuit:
  - Specify the libraries and packages
  - Entity specifies the device name and the interface (inputs and outputs).
  - Architecture describes the device operation.

State Memory

- State memory (flip-flops and registers) can be inferred in VHDL by using a conditional statement that does not specify all the possible cases (if statement with no else portion). The value of the signal when not specified in this case is interpreted as a hold of the last value.

```
ARCHITECTURE behavior of dflipflop IS
BEGIN
   PROCESS (CLK, CLRN)
   BEGIN
        IF (CLRN = '0') THEN
               Q <= '0';
        ELSIF (RISING_EDGE(CLK)) THEN
               Q <= D;
        END IF;
   END PROCESS;
END behavior;
```

Note: you will not need to instantiate D flip-flops for lab 6, this is an example illustrating the inferred hold.

No ELSE block specifying what to do on other portions of clock signal.

State Memory

```
PROCESS(CLOCK, RESETN)
BEGIN
   IF RESETN = '0' THEN
      STATE <= STATE0;
   ELSIF RISING_EDGE(CLOCK) THEN
      CASE STATE IS
         WHEN STATE0 =>
            IF X = '0' THEN
               STATE <= STATE0;
            ELSE
             STATE <= STATE1;
            END IF;
        WHEN STATE1 =>
          IF X = '0' THEN
             STATE <= STATE0;
          ELSE
             STATE <= STATE1;
          END IF;
        END CASE;
   END IF;
END PROCESS;
```

No ELSE block for the IF ELSEIF checking the reset and rising edge of clock.

The behavior of the circuit is not specified for cases where RESETN is high and CLOCK is not a rising edge. VHDL compiler infers a hold (STATE <= STATE).

2-bit Up and Down Counter

- The counter counts up through the 2-bit sequence when CE = 1 and UDN = 1 and counts down when CE = 1 and UDN = 0. When not enabled, the counter holds the current count.

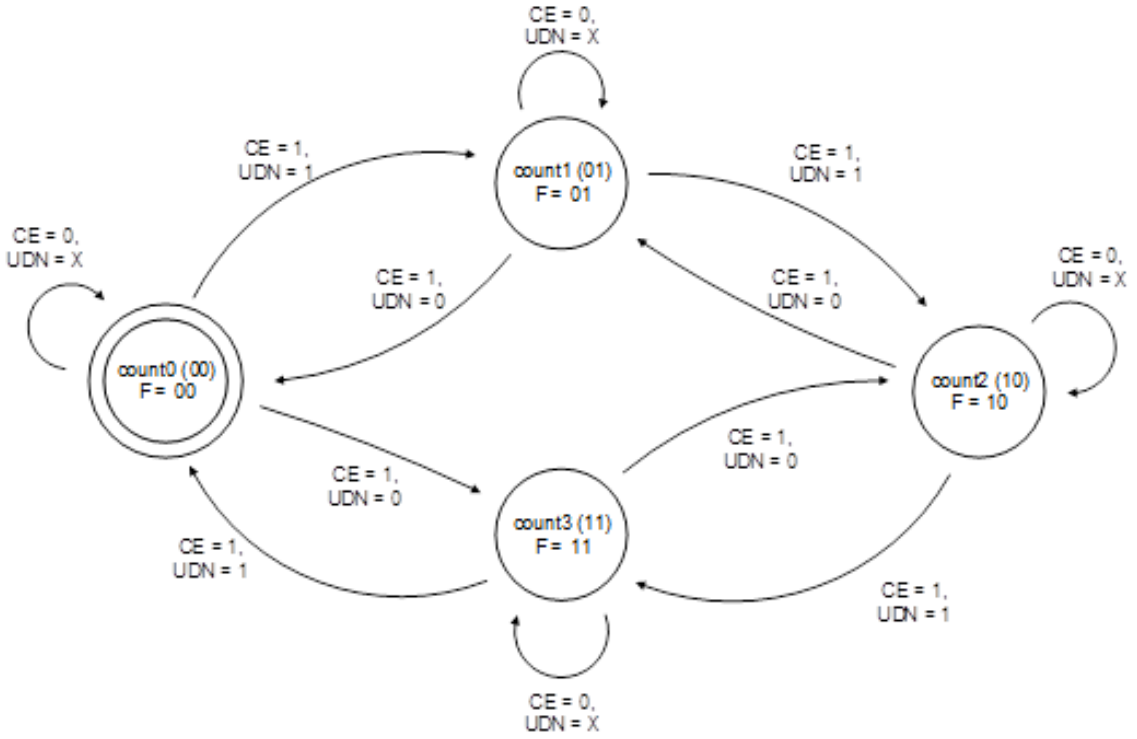


Figure 1. State Diagram of 2-bit Up and Down Counter

2-bit Up and Down Counter

Count	Q1 Q0 CE UDN	F1 F0	Q1+ Q0+
count0	0 0 0 0	0 0	0 0
	0 0 0 1	0 0	0 0
	0 0 1 0	0 0	1 1
	0 0 1 1	0 0	0 1
count1	0 1 0 0	0 1	0 1
	0 1 0 1	0 1	0 1
	0 1 1 0	0 1	0 0
	0 1 1 1	0 1	1 0
count2	1 0 0 0	1 0	1 0
	1 0 0 1	1 0	1 0
	1 0 1 0	1 0	0 1
	1 0 1 1	1 0	1 1
count3	1 1 0 0	1 1	1 1
	1 1 0 1	1 1	1 1
	1 1 1 0	1 1	1 0
	1 1 1 1	1 1	0 0

State names and codings:
count0 Q = 00
count1 Q = 01
count2 Q = 10
count3 Q = 11
Output is the count, F = Q.

Figure 2. State transition Table of 2-bit Up and Down Counter

2-bit Up and Down Counter

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY counterupdown IS
PORT(CLOCK, RESETN : IN STD_LOGIC;
   CE, UDN  : IN STD_LOGIC;
   Q : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
        F : OUT STD_LOGIC_VECTOR(1 DOWNTO 0));
END counterupdown;
```

Entity, Q is an output for testing

Libraries and what portion of library to use

Port map

2-bit Up and Down Counter

```
LIBRARY
ARCHITECTURE behavior of counterupdown IS
TYPE STATETYPE IS (COUNT0, COUNT1, COUNT2, COUNT3);
SIGNAL STATE : STATETYPE;
SIGNAL CE_UDN : STD_LOGIC_VECTOR(1 DOWNTO 0);

BEGIN
   -- concatenate the CE and UDN signals
   CE_UDN <= CE & UDN;
```

Architecture, state type definition, input signal concatenation.

Type definition, these correspond to the states

Signal declarations, used to hold current state and concatenated input

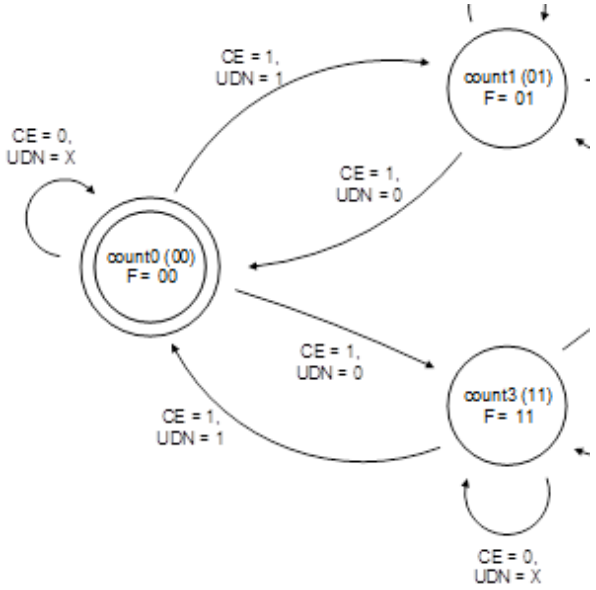
2-bit Up and Down Counter

```
-- next state logic
PROCESS(CLOCK, RESETN)
     BEGIN
        IF RESETN = '0' THEN
           STATE <= COUNT0;
        ELSIF RISING_EDGE(CLOCK) THEN
           CASE STATE IS
              WHEN COUNT0 =>
                 CASE CE_UDN IS
                    WHEN "00" => STATE <= COUNT0;
                    WHEN "01" => STATE <= COUNT0;
                    WHEN "10" => STATE <= COUNT3;
                    WHEN "11" => STATE <= COUNT1;
                 END CASE;
```

Architecture, description of a state transition.

Sensitivity list, process executed when event occurs on one of these signals

Description of one state transition



2-bit Up and Down Counter

```
-- output logic
   F  <= "00" WHEN STATE = COUNT0 ELSE
         "01" WHEN STATE = COUNT1 ELSE
         "10" WHEN STATE = COUNT2 ELSE
         "11" WHEN state = COUNT3 ELSE
         "00";

   -- state assignments
   Q  <= "00" WHEN STATE = COUNT0 ELSE
         "01" WHEN STATE = COUNT1 ELSE
         "10" WHEN STATE = COUNT2 ELSE
         "11" WHEN state = COUNT3 ELSE
         "00";
END behavior;
```

Architecture, output and state assignment

The statements are concurrent with the process for the state transitions

Lab 6 Deliverables

- Lab 6 Prelab
  - State transition table and state diagram for the state machine.
  - Behavioral VHDL design for state machine.
  - Functional simulation results for the design.
- Lab 6 Work
  - Program the DE10-standard FPLD with your state machine design and verify the design operation.
  - Logic analyzer TBD.

Lab 6 Deliverables

- Lab 6 results
  - State transition table and state diagram for the state machine.
  - Behavioral VHDL design for state machine.
  - Functional simulation results for the design.
  - Oscilloscope screen capture showing the state machine operation.
  - Explanation of how the design operation was verified on the DE10-Standard board and using the logic analyzer.