Introduction to Digital Logic
EECS/CSE 31L

Homework 2
Sequential logic modeling

course instructor: Pooria M.Yaghini
Henry Samueli School of Engineering
University of California, Irvine

Nov 15, 2014

Due on Saturday 11/22/2014 , 11:00 pm

1 Sequential Code (points 25)

Design a circuit with sequential coding to count the number of leading zeros in a binary vector, starting from its left end(MSB). (leading zeros are the number of consecutive zeros before interrupted by a '1'. Example 00001101 has 4. Example 00000001 has 7). Write the ARCHITECTURE of this design.

```
-------------------------
LIBRARY ieee;
USE ire.std_logic_1164.all;
-------------------------
ENTITY leading_zeros IS
 PORT ( data : IN STD_LOGIC_VECTOR ( 7 DOWNTO 0);
             zeros : OUT INTEGER range 0 to 8);
END ENTITY;
-------------------------
```

2 Signals and variables (points 50)

2.1 Differences

What are the differences between Signal and variable in following issues.

A) Scope
B) Update time
C) Multiple assignments

2.2 Signal and Variable properties

In this part, we try to implement a parity detector. Three architectures below try to implement the same functionality. Answer to the following questions?

A) Which code works correctly?
B) Why?
C) Would there be any difference if we use variables instead of signals?

```
--------------------------
ENTITY parity_det IS
 GENERIC ( N : POSITIVE := 8);
 PORT ( x: IN BIT_VECTOR (N -1 DOWNTO );
      y: OUT BIT);
END ENTITY;
--------------------------
```

```
------ARCHITECTURE (A)-----
ARCHITECTURE A OF parity_det IS
 SIGNAL temp: BIT;
BEGIN
 temp <= x(0);
 gen: FOR i IN 1 TO N-1 GENERATE
  temp <= temp XOR x(i);
 END GENERATE;
 y <= temp;
END ARCHITECTURE;
```

```
------ARCHITECTURE (B)-----
ARCHITECTURE B of parity_det IS
 SIGNAL temp: BIT_VECTOR (N-1 DOWNTO 0);
BEGIN
 TEMP(0) <= x(0);
 gen: FOR i IN 1 TO N-1 GENERATE
  temp(i) <= temp(i-1) XOR x(i);
 END GENERATE;
 y <= temp(N-1);
END ARCHITECTURE;
```

```
------ARCHITECTURE (C)-----
ARCHITECTURE C OF parity_det IS
 SIGNAL temp: BIT;
BEGIN
 PROCESS (x)
 BEGIN
  temp <= x(0);
  FOR i IN 1 to N-1 LOOP
   temp <= temp XOR x(i);
  END LOOP;
  y <= temp;
 END PROCESS;
END ARCHITECTURE;
```

3 Guarded Blocks (points 25)

In this question, we practice use of guarded block. Complete the architecture below using a guarded block to implement a D-type latch.

```

--------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.all;
--------------------------
ENTITY latch IS
 PORT (d, clk: IN STD_LOGIC;
          q: OUT STD_LOGIC);
END ENTITY;
--------------------------
ARCHITECTURE block_latch OF latch IS
BEGIN
?
?
?
?
END ARCHITECTURE;
--------------------------
```