Derek Yang
63118832

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 ;
-- ----------------------
ARCHITECTURE lead_zero IS
BEGIN
      PROCESS(data)
      VARIABLE count: INTEGER range 0 to 8;
      BEGIN
            count := 0;
            FOR i IN data' range LOOP
                  CASE data(i)  IS
                        WHEN '0' => count := count +1;
                        WHEN others => EXIT;
                  END CASE;
            END LOOP;
            zeros <= count;
      END PROCESS;
END lead_zero;
```

2 Signals and variables (points 50)

2.1 Differences

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

A) Scope

Signals exist in a global scope, whereas variables only exist within a local scope, which means signals exists everywhere and variables only exist in the local code that it is defined in. An exception to variables being local is shared variables which can potentially also be global.

B) Update time

Signals update after the process is over, whereas the variables all update instantly, where they are assigned.

Derek Yang
63118832

C) Multiple assignments
Signals can only have one assignment at a time, but variables can have more than one.

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?
Architecture B.

B) Why?
Architecture A and C do not work because they have temp assigned as a bit signal but has multiple assignments to them. In architecture B, temp is also a signal, but it is a bit vector, which has more than one value.

C) Would there be any difference if we use variables instead of signals?
If a variable were to be used, there would be no problems with the other architectures.

```
-- ------------------------
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 ;
```

Derek Yang
63118832

```
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
block: BLOCK (clk= '1')
      BEGIN
            q <= GUARDED d;
      END BLOCK blk;
END ARCHITECTURE ;
-- ----------------------
```