David Elliott

Evan Yeh

EE125 Homework #3

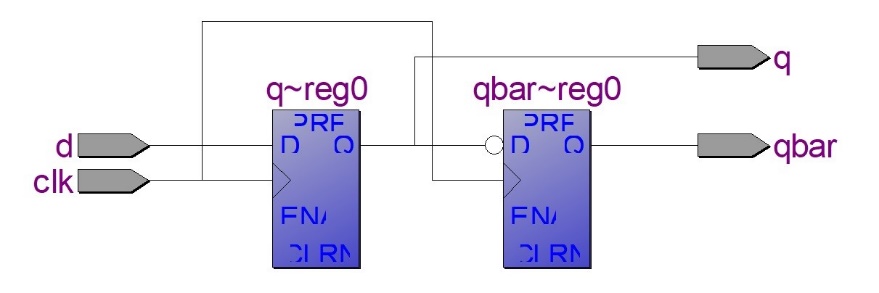
Exercise 1:

b.) arch1, arch2, arch3 are compiled and RTL views are shown below:

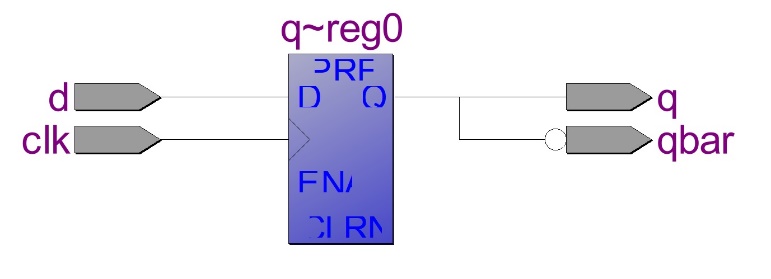
**arch1:**

**<insert>**

**arch2:**

****

**arch3:**

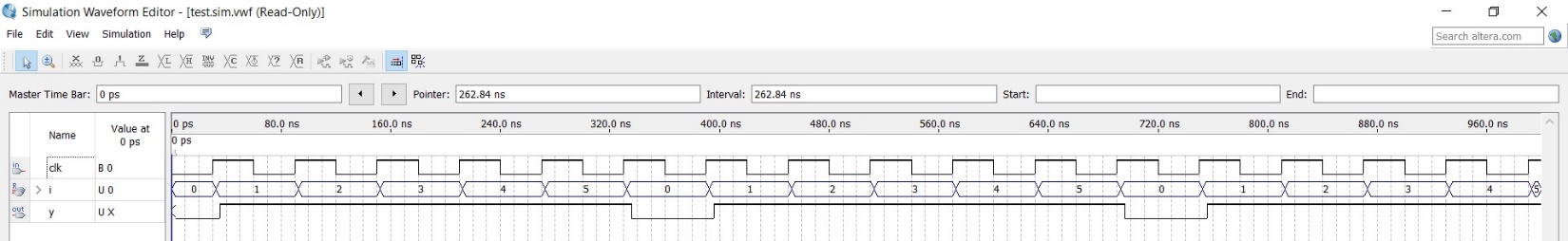
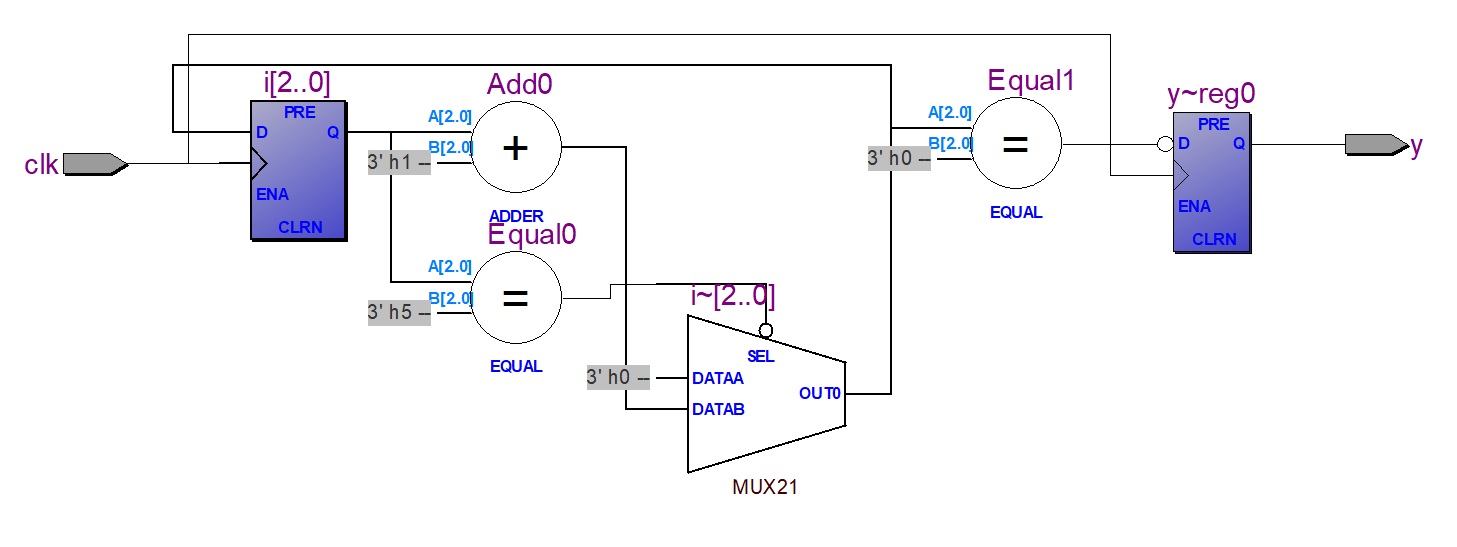


All circuits match with our predicted ones.

Exercise 2:

e.) The RTL views and simulation results of both Code 1 and Code are shown below:

**Code 1:**



**Code 2:**

<insert>

Exercise 3:

a.)

01 -------------------------------------------------------------------------

02 **library** ieee**;**

03 **use** ieee**.**std\_logic\_1164**.all;**

04 **use** ieee**.**numeric\_std**.all;**

05 -------------------------------------------------------------------------

06 **entity** leading\_ones\_sequential **is**

07 **generic** **(**N**:** integer **:=** 8 **);** --number of input bits

08 **port** **(**x**:** **in** std\_logic\_vector**(**N**-**1 **downto** 0**);**

09 clk**:** **in** std\_logic**;**

10 ssd**:** **out** std\_logic\_vector**(**6 **downto** 0**);**

11 cnt**:** **buffer** integer**);**

12 **end** **entity;**

13 -------------------------------------------------------------------------

14 **architecture** leading\_ones\_sequential **of** leading\_ones\_sequential **is**

15 **begin**

16 **process(**x**,**clk**)**

17 **variable** count**:** integer **:=** 0**;**

18 **begin**

19 **if** **rising\_edge(**clk**)** **then**

20 count **:=** 0**;**

21 **for** i **in** N**-**1 **downto** 0 **loop**

22 **if** **(**x**(**i**)** **=** '1'**)** **then**

23 count **:=** count **+** 1**;**

24 **else**

25 **exit;**

26 **end** **if;**

27 **end** **loop;**

28

29 cnt **<=** count**;**

30

31 --encode into ssd bits

32

33 **case** count **is**

34 **when** 0 **=>** ssd **<=** "0000001"**;**

35 **when** 1 **=>** ssd **<=** "1001111"**;**

36 **when** 2 **=>** ssd **<=** "0010010"**;**

37 **when** 3 **=>** ssd **<=** "0000110"**;**

38 **when** 4 **=>** ssd **<=** "1001100"**;**

39 **when** 5 **=>** ssd **<=** "0100100"**;**

40 **when** 6 **=>** ssd **<=** "0100000"**;**

41 **when** 7 **=>** ssd **<=** "0000001"**;**

42 **when** 8 **=>** ssd **<=** "0000000"**;**

43 **when** 9 **=>** ssd **<=** "0000100"**;**

44 **when** **others** **=>** ssd **<=** "0110000"**;**

45 **end** **case;**

46 **end** **if;**

47 **end** **process;**

48 **end** **architecture;**

49 -------------------------------------------------------------------------

b.) Sequential code is easier to write. It is more intuitive and requires fewer signals.

c.) Using the same test cases as last week:

