Danny Pan

Mackenzie Collins

**7.2**

A. Inputs as wires, Outputs as Reg

**7.3**

Named association

**7.4**

10 ns clock periods

**7.5**

forever

begin

Clk = 1;

#10;

Clk = 0;

#20;

end

**7.6**

C. Neither

**7.7**

Code A and Code C work.

Code B doesn’t work because the wait doesn’t add a delay to the code. The while code will execute all the way down because clk is high and it’ll trigger all the wait statements.

Code A’s @ blocks will wait a full clock it effectively spaces waits the specified clocks

Code C’s wait statements do the same by waiting for a 2 period sequence of clk

**7.8**

reg first\_done;

always(@ posedge clk)

begin

if (DONE\_S1 == 1 && DONE\_S2 == 1)

begin

#1;

Ack = 1;

@(posedge clk)

Ack = 0;

first\_done <= 0;

end

else if (first\_done == 1 && ( DONE\_S1 == 1 || DONE\_S2 == 1) )

begin

#1;

Ack = 1;

@(posedge clk)

Ack = 0;

first\_done <= 0;

end

else if ( DONE\_S1 == 1 || DONE\_S2 == 1)

first\_done <= 1;

end