CPEN 311 - Digital Systems Design

Final Exam Review Package (December 2015)

http://ubcecess.com/

https://www.facebook.com/groups/ubcecess/

True or False

1	In VHDL, a “case” statement can only be used within a process.	
2	In VHDL, a process is “executed” when any of the signals in the sensitivity list changes from 0 to 1 or from 1 to 0.	
3	A 16-bit multiplier can be described in a single synthesizable VHDL process.	
4	In VHDL, a “generate” statement can only be used within a process.	
5	In VHDL, variables are normally used to communicate from one process to another.	
6	A Moore state machine can be specified in a single synthesizable VHDL process (reminder: in a Moore machine, the outputs depend on the current state).	
7	In VHDL, variables within a process are assigned sequentially.	
8	Inside a process in VHDL, a signal is updated immediately and a variable is only updated at the end of a process.	
9	In VHDL, more than one process can update its outputs at exactly the same time.	
10	In VHDL, signals and variables within a process are assigned concurrently	
12	In VHDL, the “process” construct allows describing both combinational and sequential digital circuits	
13	To read a signal inside a VHDL process, the signal must appear in the sensitivity list.	
14	In VHDL, the following statement is synthesizable: “a <= b and c after 5 ns;”	
15	A Mealy state-machine cannot be specified in a single synthesizable VHDL process. (reminder: in a Mealy machine, the outputs depends on the current inputs and the current state)	
16	FPGAs are dedicated circuits that execute programs written in VHDL.	

17	In Quartus II (which we have been using in the lab), you can use a .mif file to store timing constraints that the compiler would use when compiling your VHDL.	
18	When you “compile” your code in the lab, Quartus II converts your VHDL code to digital logic (gates) and then maps these gates to lookup-tables.	
19	The maximum clock frequency achievable on a FPGA depends on both the FPGA and the digital circuit being implemented on the FPGA.	
20	In an FPGA, a look-up table (LUT) is configured by shifting in a bitstream of ‘1’ and ‘0’ logic values into the memory elements of the LUT.	
21	Hold time violations (failures due to not meeting a flip-flop’s hold time) can usually be eliminated by increasing the clock period.	
22	Set-up time violations (failures due to not meeting a flip-flop’s set-up time) can usually be eliminated by increasing the clock period.	
23	Clock-to-Q delay of flip-flops does not influence the maximum clock frequency of a digital circuit.	
24	Asynchronous state machines can be affected by set-up and hold time violations.	
25	Metastability can occur only in combinational circuits.	
26	An 8-bit BCD counter has exactly 100 states.	
27	An n-bit Linear-Feedback Shift Register (LFSR) cycles through a total of 2^(n-1) unique states before repeating.	
28	A decoder asserts one of 2^N output signals depending on the value on the N input signals.	
29	A priority encoder asserts one of 2^N output signals depending on the value of N input signals.	

True or False (Solutions)

1	In VHDL, a “case” statement can only be used within a process.	True
2	In VHDL, a process is “executed” when any of the signals in the sensitivity list changes from 0 to 1 or from 1 to 0.	True
3	A 16-bit multiplier can be described in a single synthesizable VHDL process.	True
4	In VHDL, a “generate” statement can only be used within a process.	False
5	In VHDL, variables are normally used to communicate from one process to another.	False
6	A Moore state machine can be specified in a single synthesizable VHDL process (reminder: in a Moore machine, the outputs depend on the current state).	True
7	In VHDL, variables within a process are assigned sequentially.	True
8	Inside a process in VHDL, a signal is updated immediately and a variable is only updated at the end of a process.	False
9	In VHDL, more than one process can update its outputs at exactly the same time.	True
10	In VHDL, signals and variables within a process are assigned concurrently	False
12	In VHDL, the “process” construct allows describing both combinational and sequential digital circuits	True
13	To read a signal inside a VHDL process, the signal must appear in the sensitivity list.	False
14	In VHDL, the following statement is synthesizable: “a <= b and c after 5 ns;”	False
15	A Mealy state-machine cannot be specified in a single synthesizable VHDL process. (reminder: in a Mealy machine, the outputs depends on the current inputs and the current state)	True
16	FPGAs are dedicated circuits that execute programs written in VHDL.	False

17	In Quartus II (which we have been using in the lab), you can use a .mif file to store timing constraints that the compiler would use when compiling your VHDL.	False
18	When you “compile” your code in the lab, Quartus II converts your VHDL code to digital logic (gates) and then maps these gates to lookup-tables.	True
19	The maximum clock frequency achievable on a FPGA depends on both the FPGA and the digital circuit being implemented on the FPGA.	True
20	In an FPGA, a look-up table (LUT) is configured by shifting in a bitstream of ‘1’ and ‘0’ logic values into the memory elements of the LUT.	True
21	Hold time violations (failures due to not meeting a flip-flop’s hold time) can usually be eliminated by increasing the clock period.	False
22	Set-up time violations (failures due to not meeting a flip-flop’s set-up time) can usually be eliminated by increasing the clock period.	True
23	Clock-to-Q delay of flip-flops does not influence the maximum clock frequency of a digital circuit.	False
24	Asynchronous state machines can be affected by set-up and hold time violations.	False
25	Metastability can occur only in combinational circuits.	False
26	An 8-bit BCD counter has exactly 100 states.	True
27	An n-bit Linear-Feedback Shift Register (LFSR) cycles through a total of 2^(n-1) unique states before repeating.	False
28	A decoder asserts one of 2^N output signals depending on the value on the N input signals.	True
29	A priority encoder asserts one of 2^N output signals depending on the value of N input signals.	False

Short Answer

Final - Spring 2014 - Question 2

a. List three disadvantages of using an FPGA compared to a custom chip when implementing a digital circuit.

b. In class, we saw four types of adders: ripple adder (carry-propagate adder), carry-select adder, carry-save adder, and carry look-ahead adder.
   i. Suppose we wish to add two 32-bit unsigned numbers. Of these adders, which is the fastest?
   ii. Suppose we wish to add two 32-bit signed numbers. Of these adders, which will use the least hardware?
   iii. Which of these adders would be most appropriate if we wish to add eight 32-bit unsigned numbers together?

c. In a large digital design written in VHDL, explain the purpose of a "testbench".

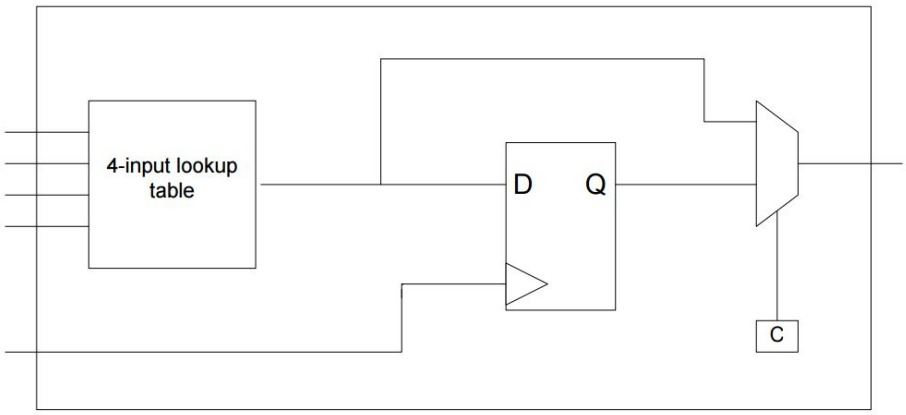
d. Timing analysis in all FPGA CAD tools is "conservative"; that is, these tools typically report a clock frequency that is lower than the maximum clock frequency of the actual circuit. Why?

e. Consider the following VHDL code:

```
entity test3 is
   port ( clk: in std_logic;
          A, B, C, D, E : in std_logic;
          outsig : out std_logic);
end test3;

architecture behavioural of test3 is
signal Q1, Q2 : std_logic;
begin
  process(clk)
  begin
       if (clk'event and clk= '1') then
           Q1 <= A and B and C and D and Q2;
           Q2 <= A or B or C or D or Q1;
       end if;
   end process;
   outsig<=Q1 or E;
end behavioural;
```

Suppose the above code is compiled to an FPGA that contains logic blocks that look like this:



What is the minimum number of logic blocks that would be required to implement the circuit?

FInal - Fall 2007 - Question 2

a. List two advantages of using an FPGA compared to using a custom chip when implementing a digital circuit:

b. Explain the difference between an “asynchronous reset” and a “synchronous reset” for a state machine.

Final - Fall 2006 - Question 2

a. Explain the difference between an “encoder” and a “priority encoder”.

Short Answer (Solutions)

Final - Spring 2014 - Question 2

f. List three disadvantages of using an FPGA compared to a custom chip when implementing a digital circuit.
   i. **Slower**
   ii. **More expensive in large volumes**
   iii. **Higher power consumption**

g. In class, we saw four types of adders: ripple adder (carry-propagate adder), carry-select adder, carry-save adder, and carry look-ahead adder.
   i. Suppose we wish to add two 32-bit unsigned numbers. Of these adders, which is the fastest? **carry look-ahead**
   ii. Suppose we wish to add two 32-bit signed numbers. Of these adders, which will use the least hardware? **ripple (carry-propagate)**
   iii. Which of these adders would be most appropriate if we wish to add eight 32-bit unsigned numbers together? **carry-save**

h. In a large digital design written in VHDL, explain the purpose of a "testbench". **A testbench is used to provide stimulus for inputs of a design under test during simulation. It can also be used to monitor output signals.**

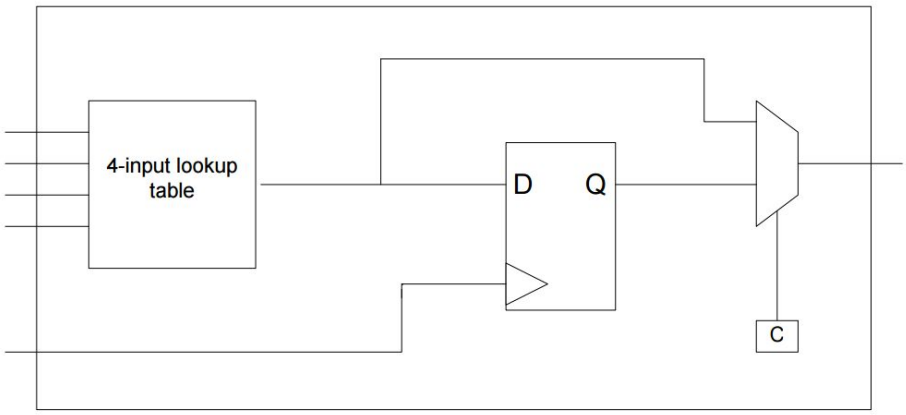
i. Timing analysis in all FPGA CAD tools is "conservative"; that is, these tools typically report a clock frequency that is lower than the maximum clock frequency of the actual circuit. Why? **All chips have slightly different speeds due to manufacturing differences. The tools needs to present a "worst case" so your circuit will work with any of these chips.**

j. Consider the following VHDL code:

```
entity test3 is
   port ( clk: in std_logic;
          A, B, C, D, E : in std_logic;
          outsig : out std_logic);
end test3;

architecture behavioural of test3 is
signal Q1, Q2 : std_logic;
begin
  process(clk)
  begin
       if (clk'event and clk= '1') then
           Q1 <= A and B and C and D and Q2;
           Q2 <= A or B or C or D or Q1;
       end if;
   end process;
   outsig<=Q1 or E;
end behavioural;
```

Suppose the above code is compiled to an FPGA that contains logic blocks that look like this:



What is the minimum number of logic blocks that would be required to implement the circuit? **The minimum number of logic blocks needed is 5**

FInal - Fall 2007 - Question 2

c. List two advantages of using an FPGA compared to using a custom chip when implementing a digital circuit:
   i. **Faster time-to-market**
   ii. **Less expensive in small volumes**

d. Explain the difference between an "asynchronous reset" and a "synchronous reset" for a state machine.
   i. **Asynchronous reset: the state of the circuit is updated immediately when the reset signal is asserted.**
   ii. **Synchronous reset: the state of the circuit is updated on the next active edge of the clock after the reset signal is asserted.**

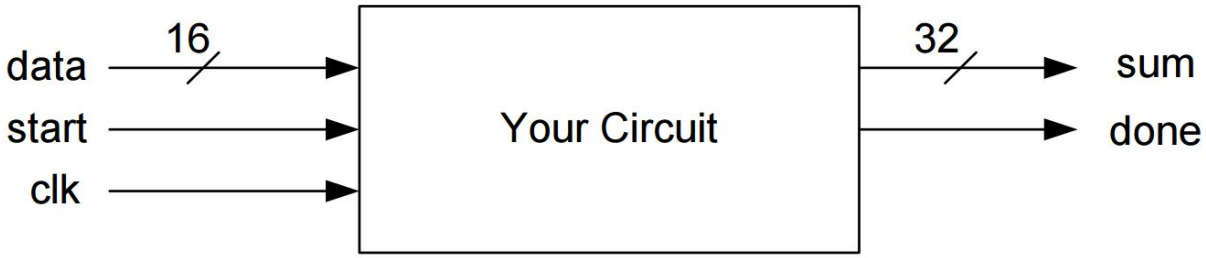
Final - Fall 2006 - Question 2

b. Explain the difference between an "encoder" and a "priority encoder". **A priority encoder is a special case of an encoder. The difference is how they handle the case when more than one input is a 1. In a priority encoder, the inputs are each given a priority, and the ID of the highest priority input that is a 1 is sent to the output. In a regular encoder, the ID of any one of the inputs that are 1 could be sent to the output.**

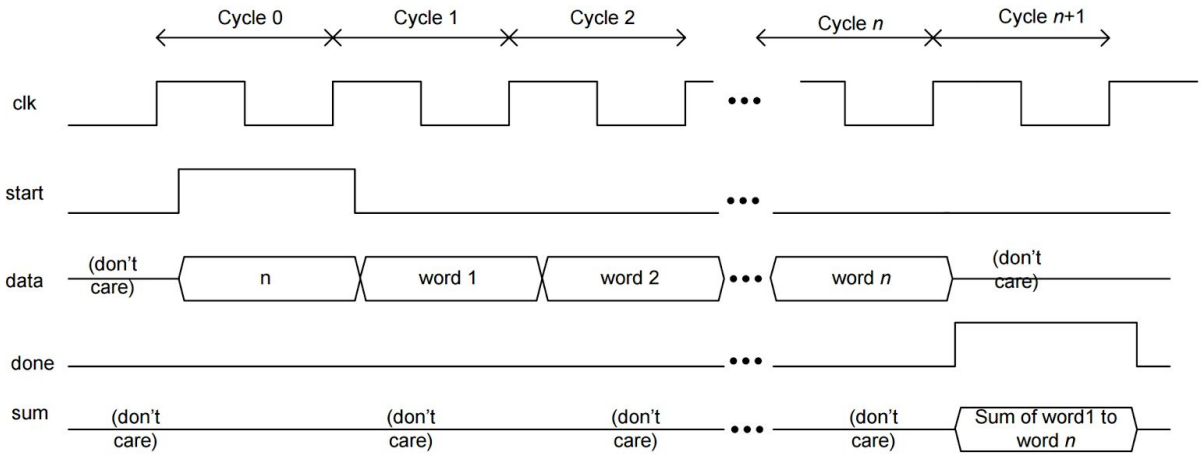
VHDL

Final - Summer 2014 - Question 5

In this question, you are to design an adder that adds an arbitrary number of 16-bit words. The inputs and outputs of your circuit are as follows:



During the first cycle (which we will call cycle 0), the user asserts 'start', and sends the number of words to be added on input 'data'. Suppose *n* words are to be added. On each of the next *n* cycles (which we will call cycle 1 through cycle *n*), the user sends one word on input 'data'. In cycle *n*+1, your circuit will assert output 'done', and send the sum of the n words on output 'sum'. This is shown graphically below. Note that, *except* in cycle *n*+1, we don't care what value is driven on 'sum'.



You can assume that n will not be zero, but your circuit should work for any other value of n that can be specified using 16 bits.

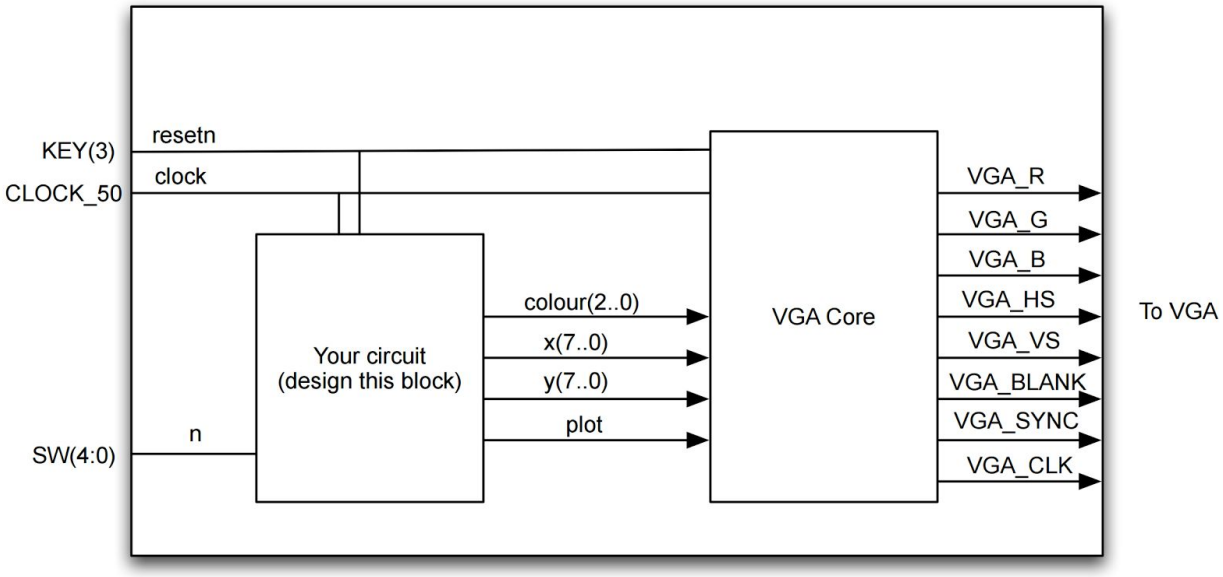
You are to write synthesizable VHDL code to specify the behaviour of this circuit.

Design stub & testbench: http://www.edaplayground.com/x/FmC

Design solution & testbench: http://www.edaplayground.com/x/Rae

Final - Spring 2014 - Question 5

You are to design a circuit to interface to a VGA core as shown in this diagram.



This VGA core has inputs 'x', 'y', 'plot', 'colour', and 'clk'. To turn on a pixel, you drive the 'x' input with the x position of the pixel, drive the 'y' input with the y position of the pixel, and 'colour' with the colour that you wish to use. You then raise input 'plot'. At the next rising clock edge, the pixel turns on. It is important to note that, at most, one pixel can be turned on each cycle. Thus, if you want to turn on *m* pixels, you need *m* cycles. The screen has 160 pixels in the x direction and 120 pixels in the y direction.

In this question, you are to design a circuit to accept a 5-bit input from switches. This quantity will indicate an integer between 0 and 31, which we will denote by *n*. Your circuit has an asynchronous active-low 'resetn'. After being reset, the circuit uses the VGA core to draw a line connecting (0,0) to (*n*,*n*) and from (*n*,*n*) to (0,2*n*) where *n* was defined above. Note that the lines will be at exactly 45 degrees. Your lines should be white (colour code = "111").

After the lines are both done, the circuit holds the screen constant until the reset line goes low, and then the process repeats. You can assume the user does not change the input n while the lines are being drawn. You do not need to worry about clearing the screen.

You are to write synthesizable VHDL code to specify the behaviour of this circuit.

Design stub & testbench: http://www.edaplayground.com/x/Mw5

Design solution & testbench: http://www.edaplayground.com/x/CkG

Final - Summer 2014 - Question 4

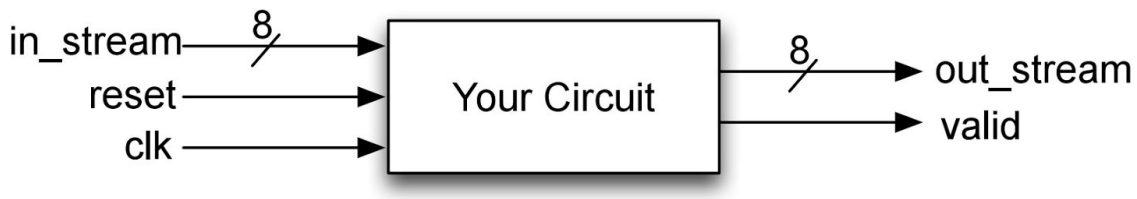
Consider the following circuit, which is common in communications-related applications to filter out high frequency noise. The circuit receives a series of data elements, one per clock cycle. The circuit produces a series of output data elements, one per clock cycle. Each output data element is the *average* of the four most-recently received input data elements. In other words,

$$out_i = (in_i + in_{i-1} + in_{i-2} + in_{i-3}) / 4$$

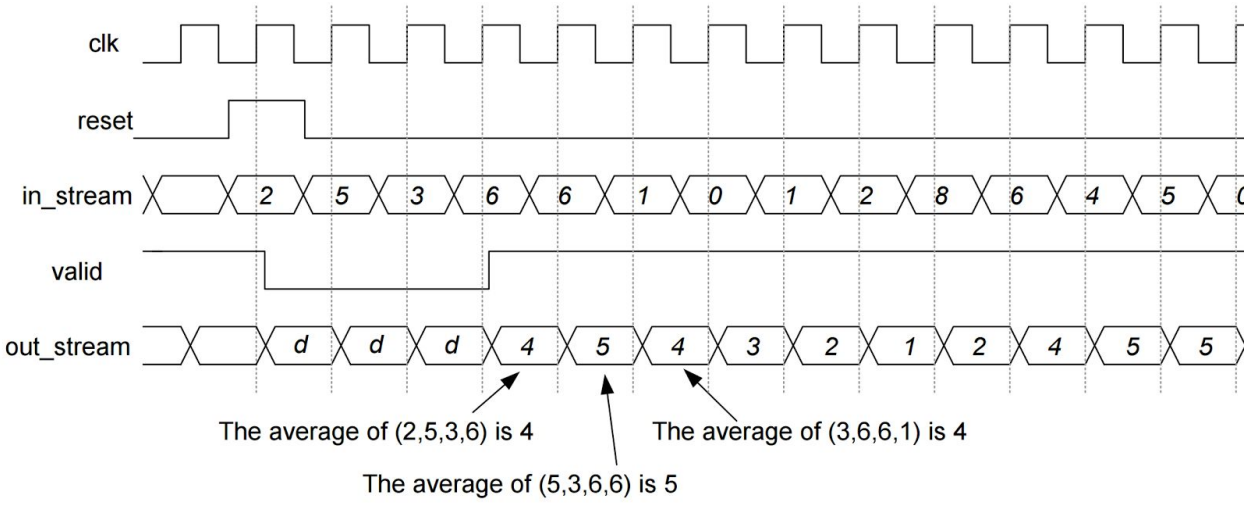
where:

- out_i is the output data element produced during cycle i
- in_i is the input data element received during cycle i
- in_{i-1} is the input data element received during cycle i-1
- in_{i-2} is the input data element received during cycle i-2
- in_{i-3} is the input data element received during cycle i-3

If the result is not an integer, it is rounded down (so if the average is 3.6, it is rounded down to 3).



Each cycle one 8-bit input data element appears on 'in_stream', and each cycle, one 8-bit result is produced on 'out_stream'. There is a synchronous reset. The 'valid' output is low during the first three cycles after the reset, and goes high after the first three cycles (the output is not valid for the first three cycles because there are fewer than four data elements to average).



In the example timing diagram, during the first three cycles after the reset, the ‘valid’ output is low, and the ‘out_stream’ can be anything (denoted *d* on the diagram). During the fourth cycle, the ‘valid’ output goes high, and the number 4 appears on ‘out_stream’ (this is the average of 2,5,3, and 6). For all remaining cycles, the value appearing on ‘out_stream’ is the average of the inputs received during the current cycle and the previous three cycles. All outputs change aligned to the positive clock edge.

Write synthesizable VHDL code to specify the behaviour of this circuit. You can assume that the division operator (“/”) is synthesizable, and that if the result of a divide is fractional, it is rounded down.

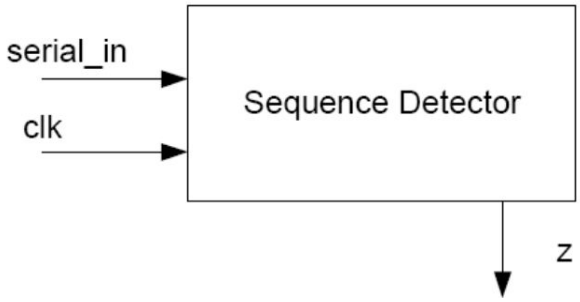
Design stub & testbench: http://www.edaplayground.com/x/4UL

Design solution & testbench: http://www.edaplayground.com/x/5TH

Alternate design solution & testbench: http://www.edaplayground.com/x/5YY

Final - Fall 2007 - Question 4

Consider the following synchronous sequence detector. Each clock cycle, one bit is received on "serial_in". When the three most-recently received bits are "010" or "101", the sequence detector should output a 1 at the "z" output. Otherwise, it should output a 0. The circuit is sensitive to the rising edge of the clock. Write a single synthesizable VHDL process to describe this circuit (hint: think "shift register").

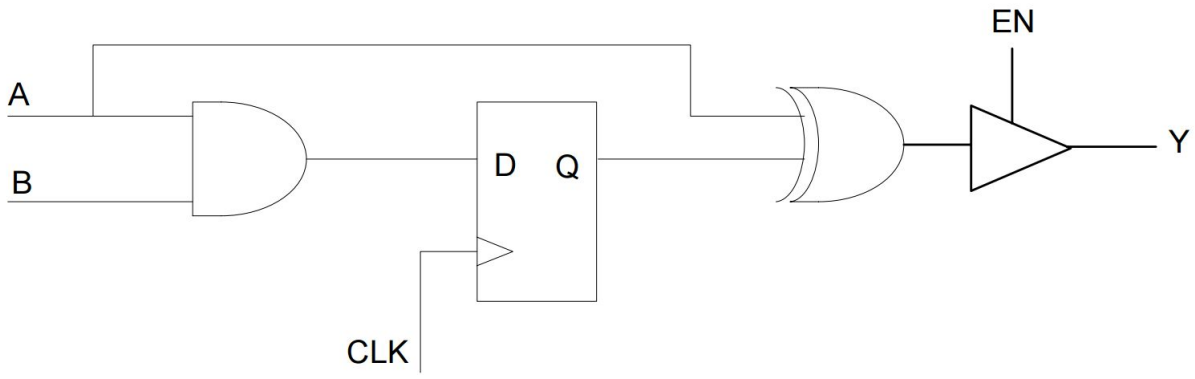


Design stub & testbench: http://www.edaplayground.com/x/SRD

Design solution & testbench: http://www.edaplayground.com/x/PLm

Final - Fall 2007 - Question 3

Using no more than two processes, write synthesizable VHDL code that has the same behaviour as the following circuit.

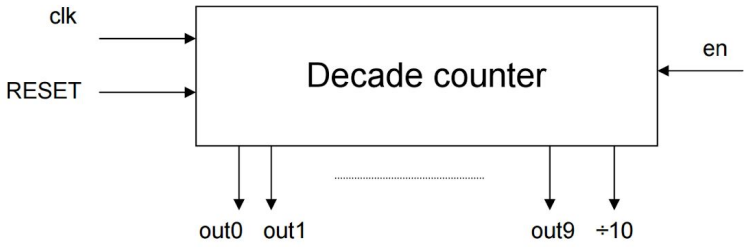


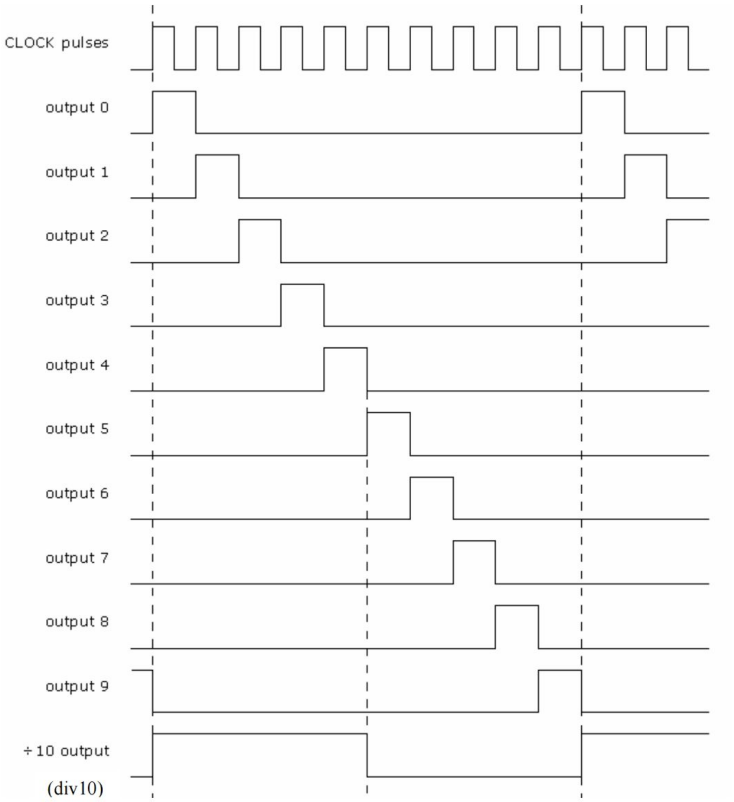
Design stub & testbench: http://www.edaplayground.com/x/UYG

Design solution & testbench: http://www.edaplayground.com/x/GEW

Final - Fall 2007 - Question 6

Using as few processes as possible, write synthesizable VHDL code that describes a 10-bit decade counter. The counter counts from 0 to 9 (each clock cycle, the count increments by 1). The counter has a 10-bit output, plus an additional output “div10”, which is high for counts 0 to 4 and low for counts 5 to 9. The circuit only counts when “en” is high, otherwise it keeps the previous value of the inputs. Your circuit should have an asynchronous reset.





Design stub & testbench: http://www.edaplayground.com/x/PeK

Design solution & testbench: http://www.edaplayground.com/x/GaF