CPEN 311 - Digital Systems Design

Midterm Exam Review Package (October 2015)

UNDERGRADUATE
STUDENT SOCIETY

http://ubcecess.com/

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

Short Answer

Midterm - Summer 2015 - Question 1

a. Explain the meaning of the VHDL synthesis tool warning “latches were inferred”:

b. Explain the meaning of the VHDL synthesis tool warning “multiple drivers were found”.

c. Explain the purpose of a VHDL “sensitivity list”:

d. Consider the following snippet of VHDL:

```
...
signal f, g, clk : std_logic;
...
process (CLK)
  variable h : std_logic;
begin
  if (rising_edge(clk)) then
    h := g;
    g <= h or f;
    h := f and g;
    g <= not f;
  end if;
end process
...
```

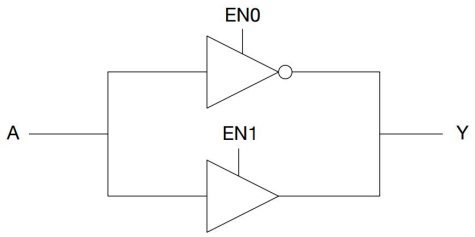
At some point during simulation, a rising edge occurs on “clk” causing the process to be evaluated. When this event occurs, f=’1’, g=’0’, and h=’1’. What values do “g” and “h” have when the end of the process is reached and all updates have occurred?

Midterm - Spring 2015 - Question 1

a. Explain when a “for” loop in a VHDL process can be synthesized:

b. Explain when you might want to use a VHDL “generate” statement:

c. Consider the following circuit that consists of a tri-state inverter and a tri-state buffer. Write a truth table for the output “Y” (which can have values of ‘0’, ‘1’, ‘X’, and ‘Z’) for the inputs “A”, “EN0”, and “EN1” (which can have values of ‘0’ and ‘1’).



Midterm - Spring 2009 - Question 1

a. List the three types of VHDL processes that are always synthesizable:

b. Explain the difference between the behaviour of an edge-triggered flip-flop and the behaviour of a level-sensitive latch:

c. What is the “critical path” of a digital circuit?

d. Draw a schematic for a 3-bit shift register without a parallel load capability:

Midterm - Spring 2008 - Question 2

a. List at least one reason why you might use VHDL to specify your design instead of using other design methods:

b. How is a signal different from a variable in VHDL?

Short Answer (Solutions)

Midterm - Summer 2015 - Question 1

e. Explain the meaning of the VHDL synthesis tool warning "latches were inferred": **When describing a combinational process, if an output of the process is not assigned a value for some combination of inputs then the output must retain its previous value. This implies the need for storage so the process would not actually be combinational logic.**

f. Explain the meaning of the VHDL synthesis tool warning "multiple drivers were found". **This warning is issued when a signal is driven by more than one source. A common mistake on this question is having two sequential statements in a process driving a signal. In a process, only the last assignment to a signal takes effect.**

g. Explain the purpose of a VHDL "sensitivity list": **A sensitivity list is a list of inputs to a process that may trigger a change in the process output. In other words, if a sensitivity list signal changes, a process output may change due to this input event.**

h. Consider the following snippet of VHDL:

```
...

signal f, g, clk : std_logic;

...

process (CLK)

  variable h : std_logic;

begin

  if (rising_edge(clk)) then

    h := g; -- h updated to 0 immediately
```

```
    g <= h or f; -- 0 or 1, g will get updated to 1 at end of process
    h := f and g; -- 1 and 0, h updated immediately to 0
    g <= not f; -- not 1, g gets updated to 0 at end of process
  end if;
end process
...
```

At some point during simulation, a rising edge occurs on "clk" causing the process to be evaluated. When this event occurs, f='1', g='0', and h='1'. What values do "g" and "h" have when the end of the process is reached and all updates have occurred?

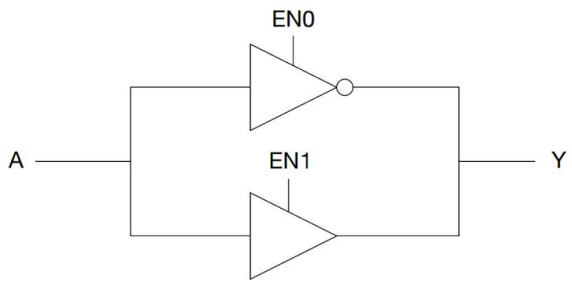
At the end of the process, g='0' and h='0'.

Midterm - Spring 2015 - Question 1

d. Explain when a "for" loop in a VHDL process can be synthesized: **It is only synthesizable when the number of loop iterations is constant at synthesis time. This is because the synthesis tool needs to "unroll" the loop.**

e. Explain when you might want to use a VHDL "generate" statement: **It is useful when you want to succinctly describe a series of component instantiations that follow a regular/uniform instantiation and port mapping pattern.**

f. Consider the following circuit that consists of a tri-state inverter and a tri-state buffer. Write a truth table for the output "Y" (which can have values of '0', '1', 'X', and 'Z') for the inputs "A", "EN0", and "EN1" (which can have values of '0' and '1').



When Tristate gate is disabled, output is high-impedance 'Z' (the gate makes no contributions to the output wire). When both are on in this case, different values are driven onto wire 'Y', and so there is fighting 'X'.

A	EN0	EN1	Y
0	0	0	Z
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	Z
1	0	1	1
1	1	0	0
1	1	1	X

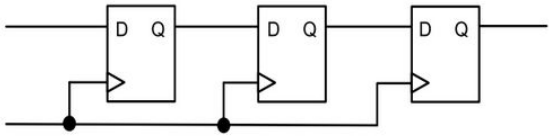
Midterm - Spring 2009 - Question 1

e. List the three types of VHDL processes that are always synthesizable:
   i. **Purely combinational**
   ii. **Purely synchronous (purely sequential)**
   iii. **Synchronous with an asynchronous reset**

f. Explain the difference between the behaviour of an edge-triggered flip-flop and the behaviour of a level-sensitive latch: **The D output of a positive edge triggered flip-flop is updated to match the Q input on the rising edge of the clock whereas the Q output of a level sensitive latch follows the D input as long as the clock/enable input is high.**

g. What is the "critical path" of a digital circuit? **The longest path from the output of any flip-flop to the input of any flip-flop.**

h. Draw a schematic for a 3-bit shift register <u>without</u> a parallel load capability:



Midterm - Spring 2008 - Question 2

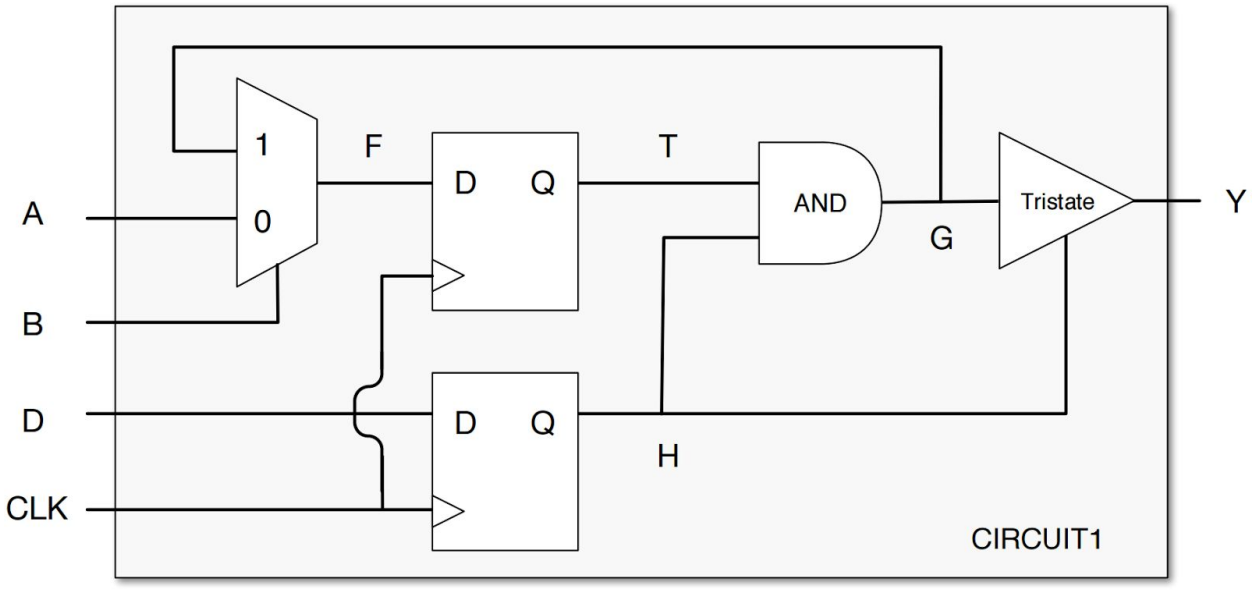
c. List at least one reason why you might use VHDL to specify your design instead of using other design methods:
   i. **Provides a higher level of abstraction than a schematic**
   ii. **Easy to debug**
   iii. **Can easily employ hierarchy and modularization to make designs more understandable**
   iv. **Portability (the same VHDL specification can be understood by several design tools)**

d. How is a signal different from a variable in VHDL?
   i. **A signal represents a wire in the circuit but a variable is more abstract and might not represent a wire.**
   ii. **Signals can be used to communicate between processes while a variable exists only within one process.**
   iii. **A signal is updated at the end of a process whereas a variable is updated immediately.**
   iv. **Signals and variables use different assignment operators.**

VHDL

Midterm - Spring 2015 - Question 2

Write synthesizable VHDL code with the same behaviour as the following circuit. Your code should consist of one entity and no more than three processes.



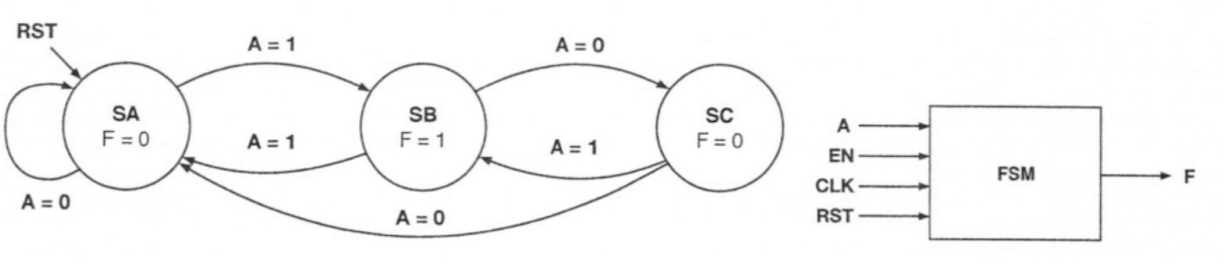
Design stub & testbench: http://www.edaplayground.com/x/BaE

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

Alternate design solution & testbench: http://www.edaplayground.com/x/8Vn

Midterm - Summer 2015 - Question 2

Write synthesizable VHDL code to describe the following finite-state machine.



This is a Moore FSM with one input 'A' and one output 'F'. There are three states named 'SA', 'SB', and 'SC'. The FSM has an asynchronous reset 'RST' and an enable signal 'EN'. Specifically, when 'EN' is 0, the FSM does not attempt to make any state transitions. Your code consist of one entity and no more than three processes.

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

Design solution & testbench: http://www.edaplayground.com/x/D6Z

Alternate design solution & testbench: http://www.edaplayground.com/x/6wg

Midterm - Summer 2014 - Question 1

Write synthesizable VHDL code that creates a 4-bit BCD (Binary Coded Decimal) output but uses an internal one-hot counter implementation. Remember that BCD counts (in binary) from 0 to 9 (each clock cycle, the count increments by 1) and then rolls over to 0, the output of your circuit should do this. The internals will use a one-hot counter; therefore you must translate between your internal count and the output. Your circuit should have a synchronous reset signal and should consist of one entity containing as few processes as possible.

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

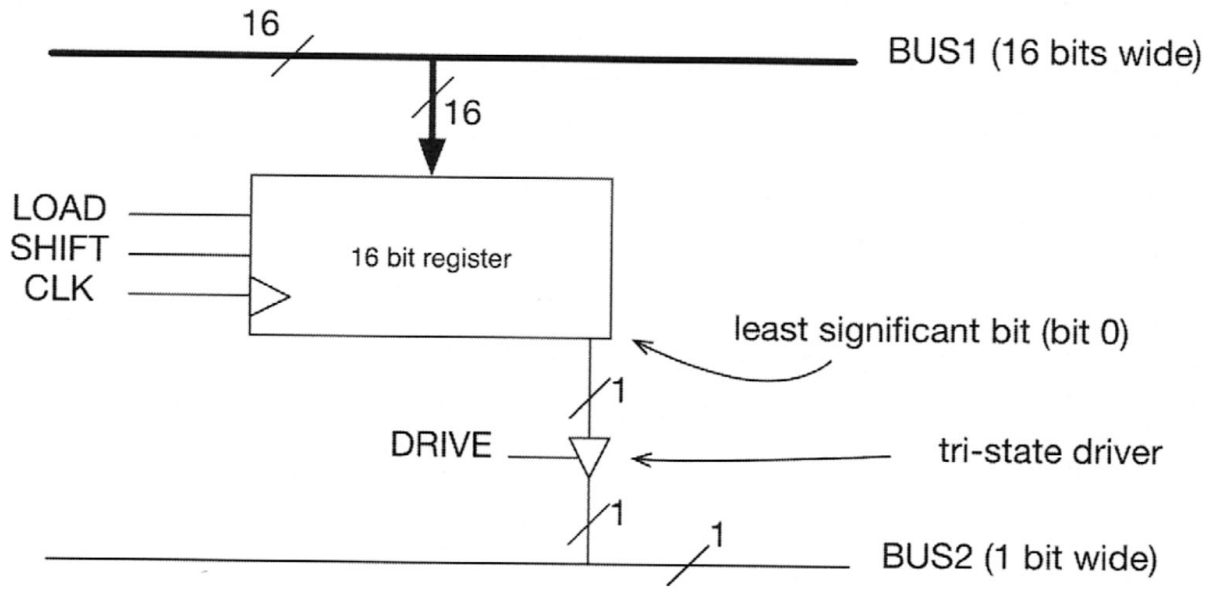
Design solution & testbench: http://www.edaplayground.com/x/4W6

Midterm - Summer 2015 - Question 4

Consider the following "bridge" circuit which is used to capture data from a 16-bit bus and *serialize* it (transfer one bit at a time) onto a single bit bus.

On a rising clock edge, if LOAD is high, the value on BUS1 is loaded into the register. If LOAD is not high, the register ignores the value on BUS1. On the rising clock edge, if SHIFT is held high, the value in the register is shifted to the right (i.e. such that the value in bit position 1 is now held in position 0, and so on for all the other bits). If both LOAD and SHIFT are asserted at the same time, SHIFT is ignored and a new value is loaded.

Finally, at any time, when DRIVE is high, the value of bit position 0 in the register is driven onto BUS2 (independent of the clock). If DRIVE is low, BUS2 is not driven and remains in a high-impedance state.



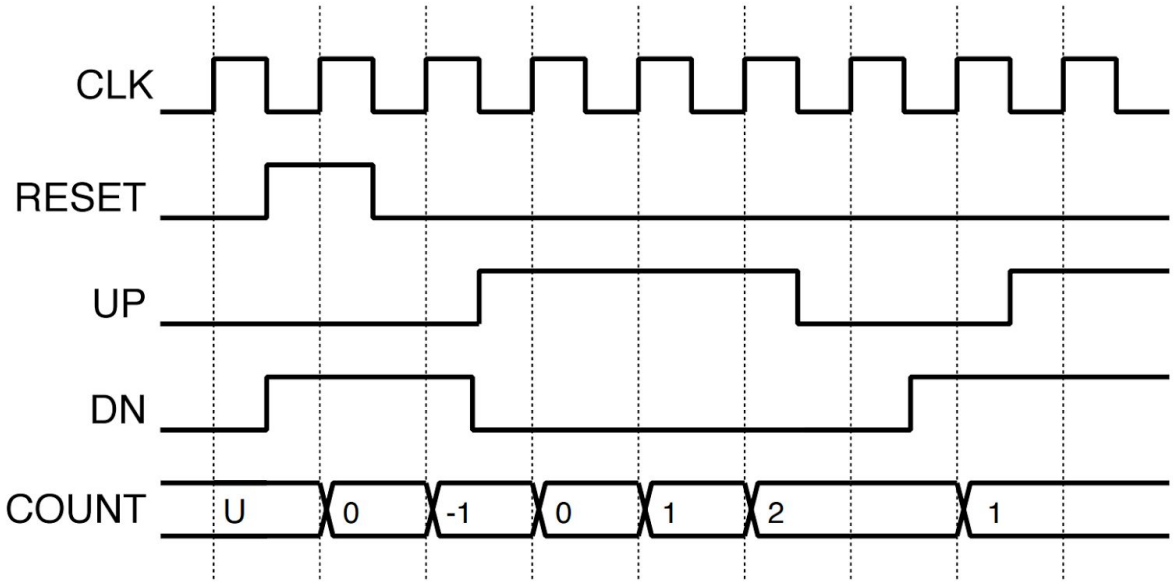
Write synthesizable VHDL that describes this circuit using as few processes as possible.

Design stub & testbench: http://www.edaplayground.com/x/7u5

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

Midterm 1 - Fall 2012 - Question 3

Consider an up/down counter circuit that operates as follows. The inputs are CLK, RESET, UP, and DN (each one bit wide). The output is an 8-bit value. When using this circuit, setting the input UP=1 will cause the counter to increment by 1 each clock cycle. Similarly, setting the input DN=1 will cause the counter to decrement by 1 each clock cycle. When both UP=1 and DN=1, the counter must not change. Likewise, it must not change if both UP=0 and DN=0. The circuit has a synchronous reset.



In the last clock cycle of this timing example, notice that both UP=1 and DN=1 and the count remains unchanged at 1.

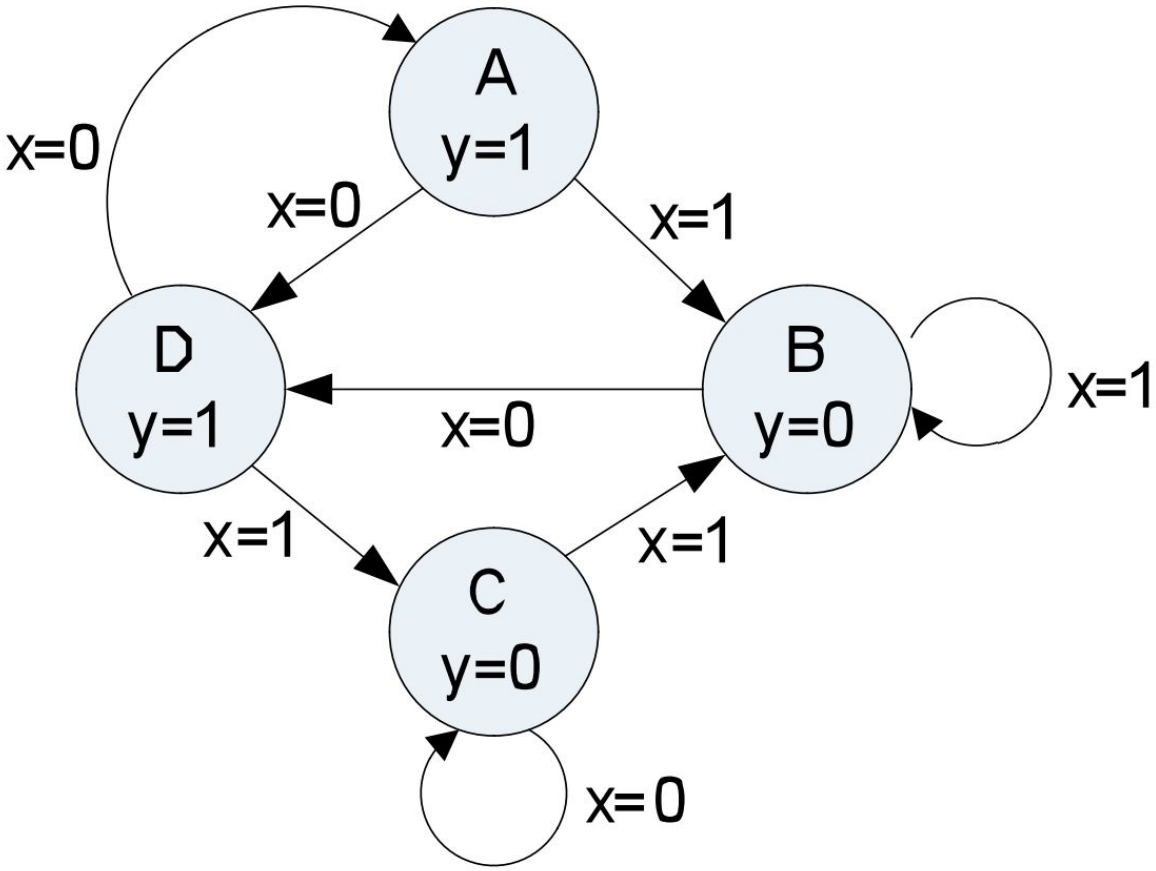
Write synthesizable VHDL code to specify the behaviour of this circuit using as few processes as possible.

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

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

Midterm - Spring 2009 - Question 2

Consider the following Moore state machine with states A, B, C, D, input “x” and output “y”.



Write synthesizable VHDL for this circuit. Use your own encoding or declare your own VHDL “type” to represent the states.

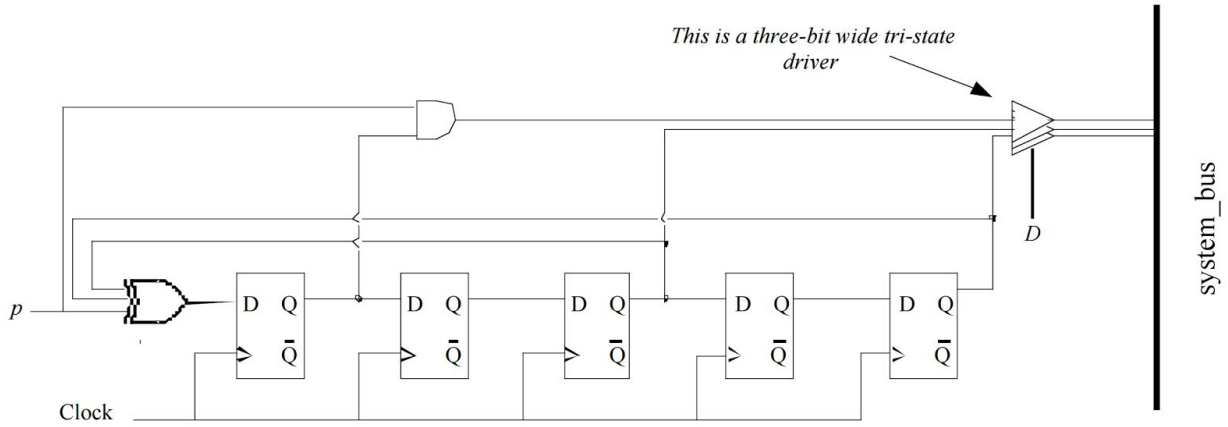
Design stub & testbench: http://www.edaplayground.com/x/GXS

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

Alternate solution & testbench: http://www.edaplayground.com/x/WfK

Midterm - Spring 2009 - Question 4

Consider the following signature analysis circuit. The circuit constructs a signature of the data arriving on input line “p” (one bit per clock cycle). When input “D” goes high, this signature is driven onto a system bus (when “D” is low, the system bus is not driven). Write a synthesizable VHDL description of this block using as few processes as possible.

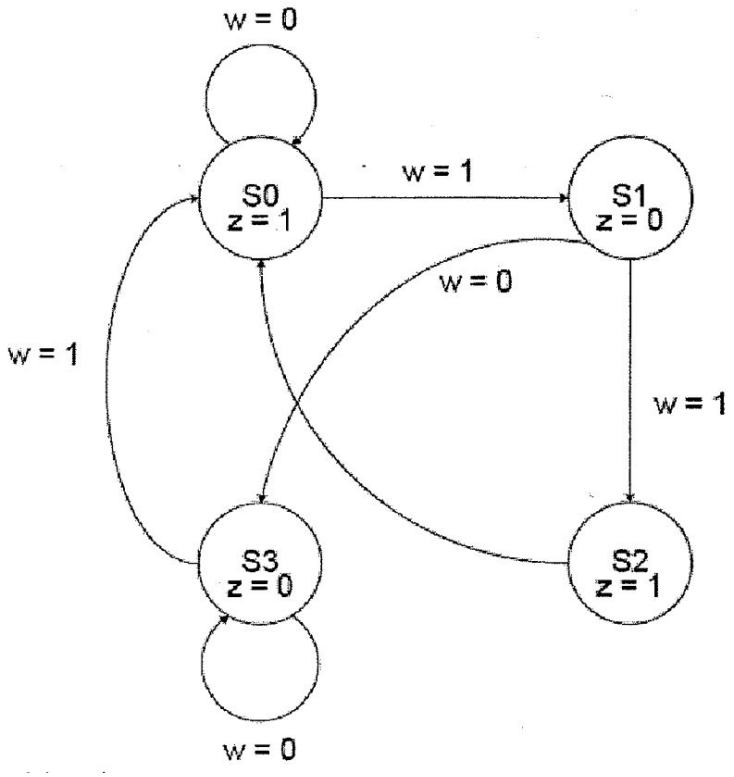


Design stub & testbench: http://www.edaplayground.com/x/N46

Design solution & testbench: http://www.edaplayground.com/x/Jf7

Midterm - Summer 2008 - Question 4

Write synthesizable VHDL code using as few processes as possible to describe the state machine represented in the figure below. Use one-hot encoding to encode the states. The state machine has an active high asynchronous reset (not shown in the bubble diagram). When the reset button is pushed, the state machine resets to state S0.

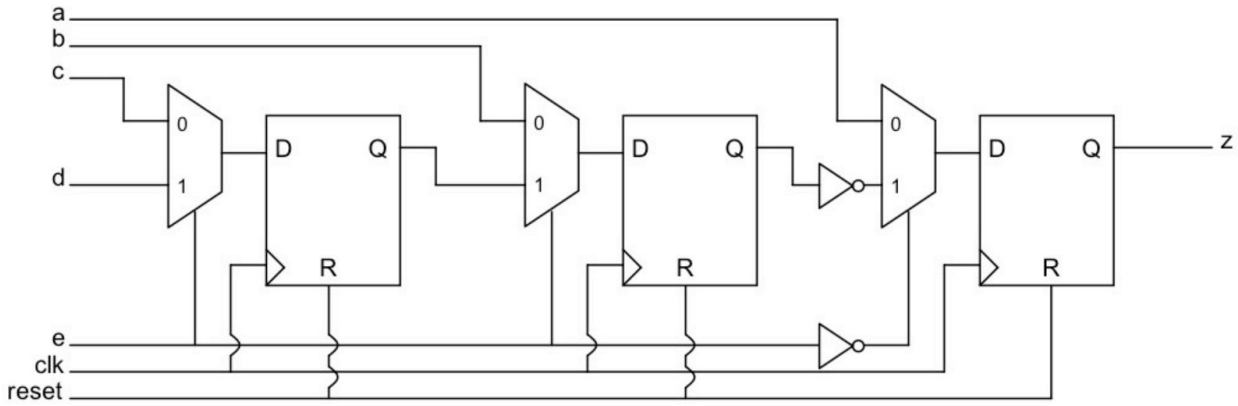


Design stub & testbench: http://www.edaplayground.com/x/6P7

Design solution & testbench: http://www.edaplayground.com/x/Wh5

Midterm 1 - Spring 2008 - Question 3

Using only one process, write synthesizable VHDL that has the same behaviour as the circuit below. Each flip-flop has a synchronous reset.



Design stub & testbench: http://www.edaplayground.com/x/6ge

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