Homework set 3

(These problems should be done individually, not with project partner)

Due – As indicated on Canvas

Instructions for submission

1) All answers are to be submitted in a **single** word/pdf file and as separate .vhd files for each entity required.

2) Include Quartus Prime, Modelsim/Altera vwf and observation screenshots whenever possible with appropriate annotations for better presentation in the word/pdf file.

Do not submit extra screenshots in the submission folder

3) Naming convention and files required

a) LastName_FirstName_HWSet3.docx/pdf - for Observations and Answers

b) Submit <Codefilename>.vhd/.v - separately for each question externally.

c) Follow the module (.v) or Entity name (.vhd) and port Name as per guidelines in each question. These modules will be auto graded and will not compile if the names do not match.

** Commenting of code, references and presentation of material also carries weight*

1. Design a FIFO memory in **Verilog or VHDL**. Make it 8-deep, 9 bits wide. When a read signal is asserted, the output of the FIFO should be enabled, otherwise it should be high impedance. When the write signal is asserted, write to one of the 9 bit registers. Use rdinc and wrinc as input signals to increment the pointers that indicate which register to read or write. Use RdPtrClr and WrPtrClr as input signals which reset the pointers to point to the first register in the array. **Do not implement full or empty signals.** Do an RTL simulation and capture the waveform in a screenshot.

Module/Entity Name: HW3Q1

Signal Names:

```
clk         -- Clock
RdPtrClr    --  Read Pointer Clear,  to reset the read pointer
WrPtrClr    --  Write Pointer Clear, to reset the write pointer
rdinc       --  Read pointer increment signal
wrinc       --  Write pointer increment signal
DataIn      --  Data input bus
DataOut     --  Data Output bus
rden        --  read (output) enable
wden        --  write (input) enable
```

2. The core of a soft processor is an ALU. Implement an example ALU in an Altera MAX10. Develop both a clocked (register the inputs and outputs) and unclocked version of the ALU. Compare the Fmax and #LUTs used for each. For the unclocked ALU you will need to determine the Fmax by measuring the delay from input to output, 1/delay = Fmax. **You do not need to load this on the board.**

Module/Entity Name: HW5Q2

Signal Names:

```
A       -- Input < as per Q> bit A to ALU
B       --  Input < as per Q> bit B to ALU
Op      --  Input ALU Opcodes
Y       --  <as per Q> bit output of ALU
```

Write a VHDL or Verilog model for a 32-bit, arithmetic logic unit (ALU). Verify correct operation with a simulation using the Altera CAD tools. A and B are 32-bit inputs to the ALU, and Y is the output. A shift operation follows the arithmetic and logical operation. The opcode controls ALU functions as follows:

Opcode	Operation	Function
000XX	ALU_OUT <= A	Pass A
001XX	ALU_OUT <= A + B	Add
010XX	ALU_OUT <= A-B	Subtract
011XX	ALU_OUT <= A AND B	Logical AND
100XX	ALU_OUT <= A OR B	Logical OR
101XX	ALU_OUT <= A + 1	Increment A
110XX	ALU_OUT <= A-1	Decrement A
111XX	ALU_OUT <= B	Pass B
XXX00	Y <= ALU_OUT	Pass ALU_OUT
XXX01	Y<= SHL(ALU_OUT)	Shift Left
XXX10	Y<=SHR(ALU_OUT)	Shift Right (unsigned-zero fill)
XXX11	Y<= 0	Pass 0's

3. Create a testbench for the Comparator2 problem from Homework 2 in VHDL or Verilog. Test all possible combinations of the inputs. Use the testbench template provided on Canvas as the basis of your code. Show a screen shot of the waveform output of your testbench.

4. Perform a gate-level simulation of problem 8 or 9 of Homework 2. Start the gate-level simulation from Quartus, and then instead of using the GUI interface, use the following vsim command in the tcl console of ModelSim:

```
vsim -t 1ps +transport_int_delays +transport_path_delays -sdftyp
/RAM128_32=RAM128_32_7_1200mv_125c_vhd_slow.sdo -L altera -L cycloneive -L gate_work -L work -
voptargs=\"+acc\" work.ram128_32
```

This assumes you are using the smallest Cyclone IV E part with –A7 temperature and speed range. It is specific to the part and design name. For another part, you would need to look in the simulation/modelsim directory to find the name of the .sdo file to include.

5. Do problem 8.1 in section 8.13 of the alternate textbook, but skip the section about the video train simulation. Do complete a simulation (ModelSim), and use the State Diagram Editor/Viewer in Quartus to generate a state diagram **from your code.** Problem 8.1 is shown here:

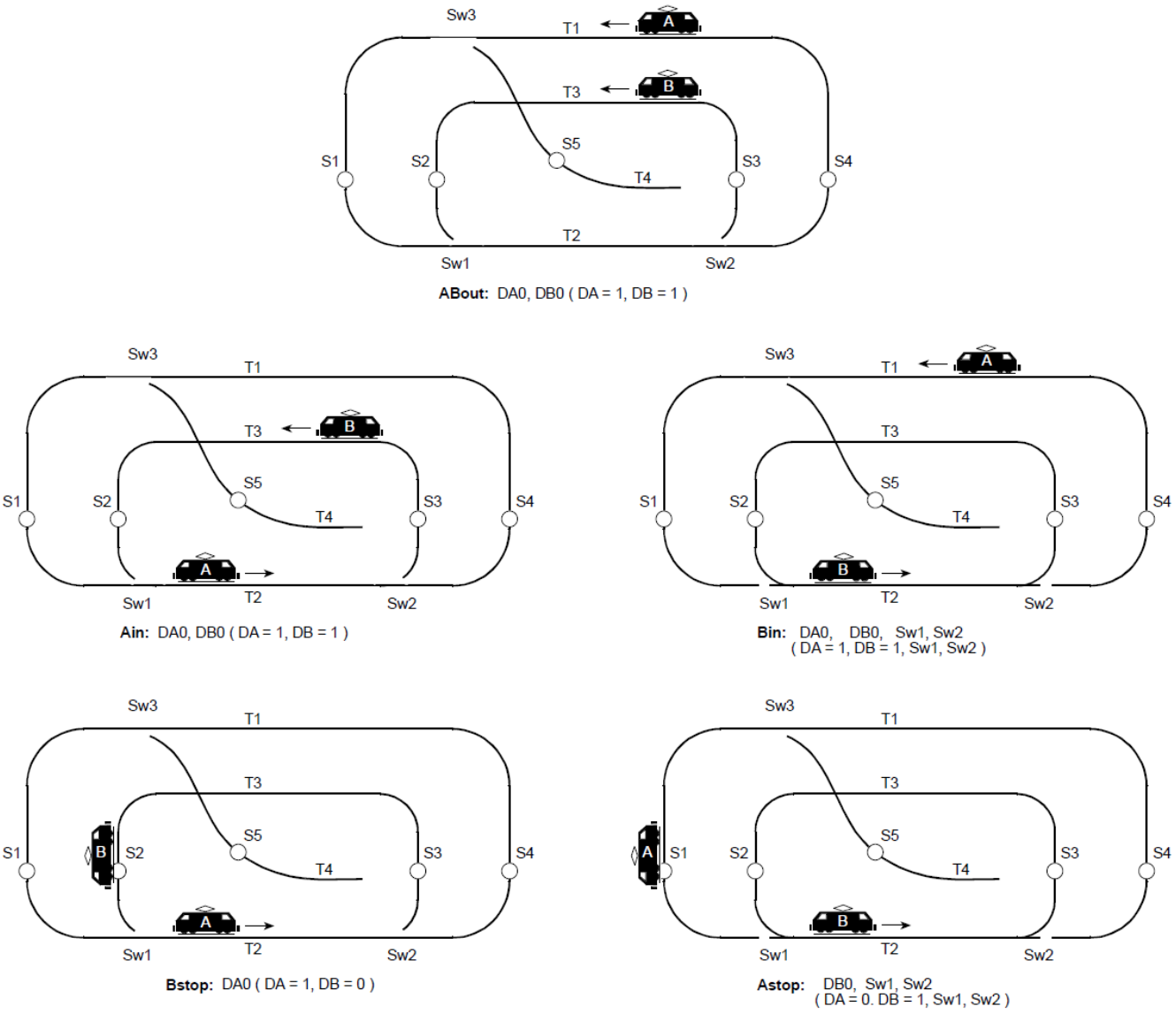


Figure 8.7 Working diagrams of train positions for each state.

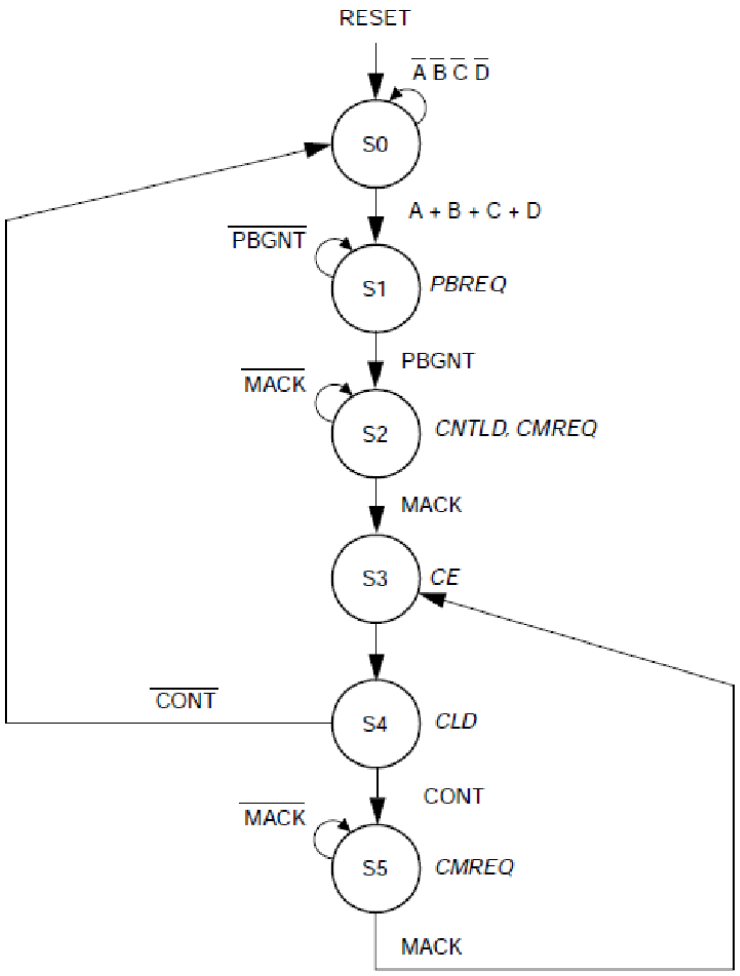
Assuming that train A now runs clockwise and B remains counterclockwise, draw a new state diagram and implement the new controller. Use VHDL to design the new controller, so you can modify the code presented in section 8.6.

******Please follow the entity name and signal name convention********

```
ENTITY Tcontrol IS
PORT( reset_n,                    --  asynchronous active low reset
      clock, sensor1, sensor2,
    sensor3, sensor4, sensor5 : IN STD_LOGIC;
    switch1, switch2, switch3 : OUT STD_LOGIC;
    -- dirA and dirB are 2-bit logic vectors(i.e. an array of 2 bits)
```

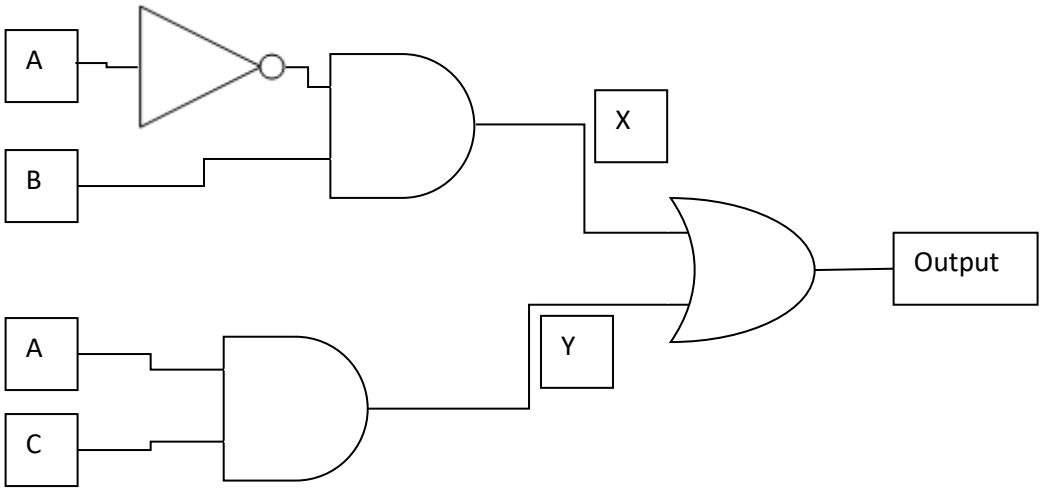
```
    dirA, dirB : OUT STD_LOGIC_VECTOR( 1 DOWNTO 0 ));
END Tcontrol;
```

6. Write HDL code to implement the state machine below, or use the Altera State Machine Editor (preferred) to automatically generate the HDL in either Verilog or VHDL.

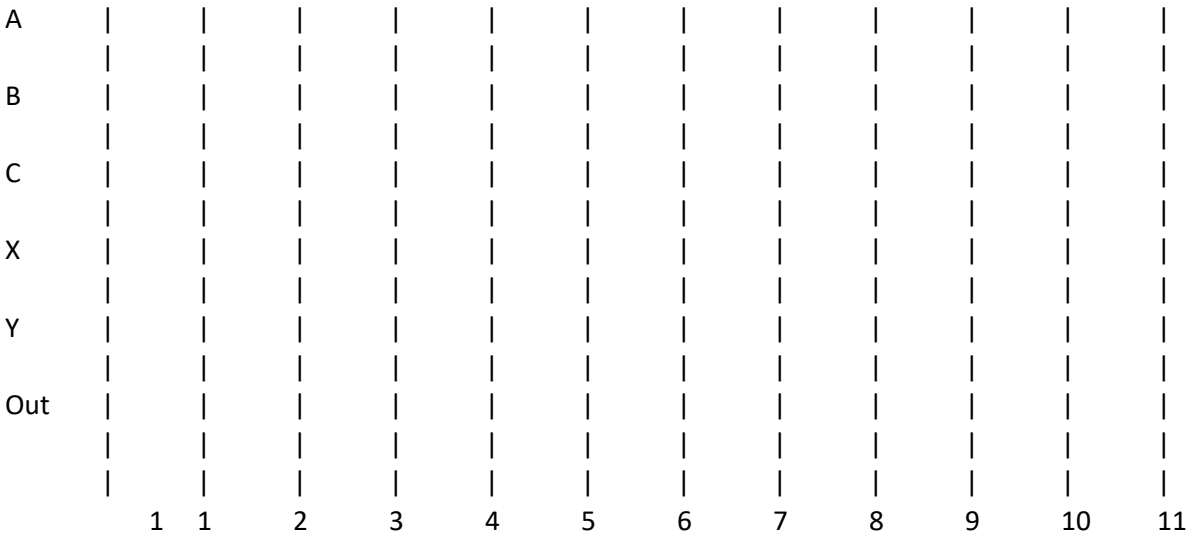


The inputs are A, B, C, D, PBGNT, MACK, CON, and RST. The outputs are PBREQ, CMREQ, CE, CNTLD, and CLD which are associated with particular states. Verify this using simulation.

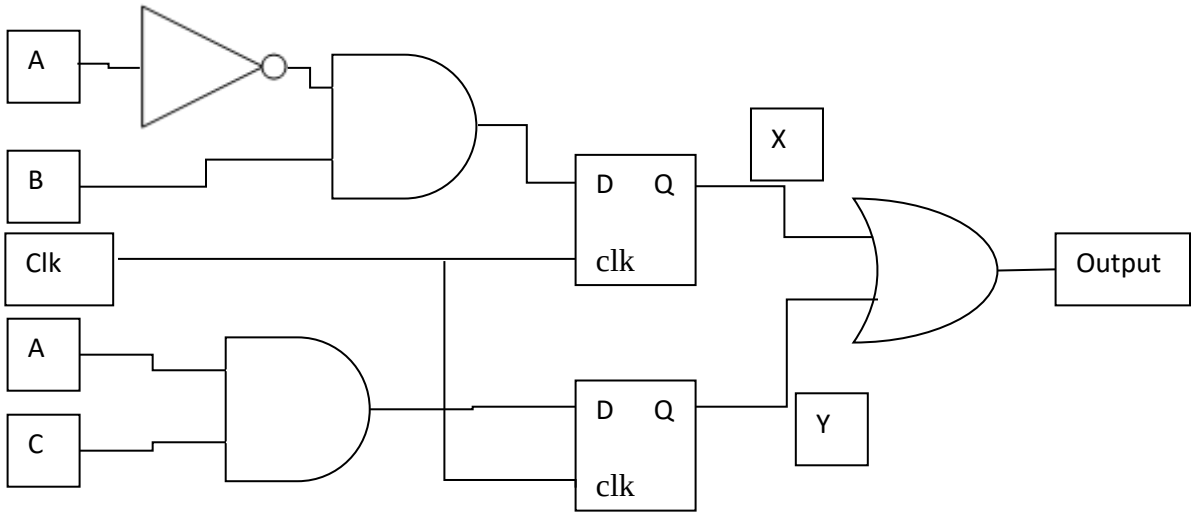
7. Assuming initial input values are high, Draw a timing diagram to scale by hand of the following circuit:



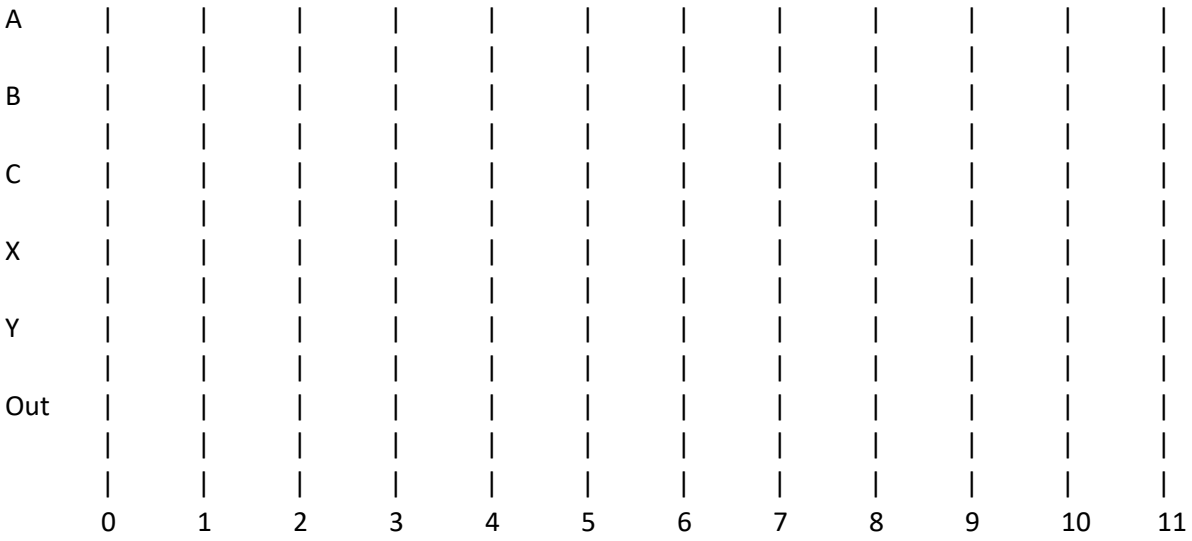
Assume propagation delays of 1 ns minimum, 3 ns maximum for each gate. Show the uncertainty in the switching time. Assume A goes low at 2 ns until 5 ns, and then goes high again. The B and C inputs are always high.



8. Assuming initial inputs are high, Draw a timing diagram <u>to scale</u> by hand of the following circuit:



Assume propagation delays of 1 ns minimum, 3 ns maximum for each gate. Show the uncertainty in the switching time. Assume A goes low at 2 ns until 5 ns, and then goes high again. The B and C inputs are always high. Assume the D FF is active on the rising edge of the clock at 2 ns, with an 8 ns clock period. Assume setup time of D FF is 2 ns, and hold time is 0 ns. Do we have a violation here?



Grading Rubric

1) - [6 points]
    ----Auto Graded
    [3 point] Verilog/VHDL Code.
    [2 point] RTL Simulation using Modelsim, Screenshots of all scenarios.
    [1 point] Implementation and working.
2) - [5 points]
    [2 point]Verilog/ VHDL Code
    [2 point] RTL Simulation -All ALU Test Scenarios, using Modelsim
    [1 point] Comparison
3) - [5 points]
    ----Auto Graded
    [2 point]Coding.
    [3 points]Implementation and RTL Simulation using Modelsim
4) - [4 points] - Gate-level simulation
5) - [6 points]
    [2 points] **Designed State Diagram**
    [2 points] VHDL Code.
    [2 points] Simulation for all cases using Modelsim
6) - [4 points]
    [3 point] Verilog/VHDL Code.
    [1 point] Explanation
7) - [5 points]
    Complete timing diagram
8) - [5 points]
    Complete timing diagram

Guides for Modelsim and simulation

1) **ModelSim GUI -Quartus Guide**
a) On Canvas, ModelSim_TUI_Introduction.pdf
b) http://staff.cs.upt.ro/~opritoiu/modelsim/simex1/en/index.html
2) **Modelsim post synthesis simulation guide**

http://cseweb.ucsd.edu/classes/fa10/cse140L/lab2/Lab2_tutorial_timinig_simulation.pdf