

Short Problems

Problem 1 (2.5 points, estimated time 15 minutes)

Draw a block diagram corresponding to the given below VHDL code:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_signed.all;

ENTITY proc IS
    PORT (Data          : IN          STD_LOGIC_VECTOR(7 DOWNTO 0);
          Reset, w      : IN          STD_LOGIC;
          Clock         : IN          STD_LOGIC;
          Done          : IN          STD_LOGIC;
          Func          : BUFFER      STD_LOGIC_VECTOR(1 TO 6);
          F, Rx, Ry     : IN          STD_LOGIC_VECTOR(1 DOWNTO 0);
          Sel           : IN          STD_LOGIC_VECTOR(5 DOWNTO 0);
          AddSub        : IN          STD_LOGIC;
          Sum           : OUT         STD_LOGIC_VECTOR(7 DOWNTO 0);
          BusWires      : INOUT      STD_LOGIC_VECTOR(7 DOWNTO 0));
END proc ;

ARCHITECTURE Behavior OF proc IS

    COMPONENT upcount IS
        PORT(Clear: IN STD_LOGIC;
             Clock: IN STD_LOGIC;
             Count: OUT STD_LOGIC_VECTOR(1 DOWNTO 0));
    END COMPONENT upcount;

    COMPONENT regn IS
        GENERIC ( N : INTEGER := 8 ) ;
        PORT (R      : IN          STD_LOGIC_VECTOR(N-1 DOWNTO 0) ;
              Rin, Clock : IN          STD_LOGIC ;
              Q      : OUT         STD_LOGIC_VECTOR(N-1 DOWNTO 0));
    END COMPONENT regn ;

    COMPONENT dec2to4 IS
        PORT(D_in:IN STD_LOGIC_VECTOR(1 DOWNTO 0);
             Enable:IN STD_LOGIC;
             D_out:OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
    END COMPONENT dec2to4;

    SIGNAL X, Y, Rin, Rout : STD_LOGIC_VECTOR(0 TO 3) ;
    SIGNAL Clear, High, FRin, Ain : STD_LOGIC ;
    SIGNAL Count, Zero, T, I : STD_LOGIC_VECTOR(1 DOWNTO 0) ;
    SIGNAL R0, R1, R2, R3 : STD_LOGIC_VECTOR(7 DOWNTO 0) ;
    SIGNAL A, G : STD_LOGIC_VECTOR(7 DOWNTO 0) ;
    SIGNAL FuncReg: STD_LOGIC_VECTOR(1 TO 6) ;

BEGIN
    Zero <= "00" ; High <= '1' ;
    Clear <= Reset OR Done OR (NOT w AND NOT T(1) AND NOT T(0)) ;
    counter: upcount PORT MAP (Clear, Clock, Count);
    T <= Count ;
    Func <= F & Rx & Ry ;
    FRin <= w AND NOT T(1) AND NOT T(0) ;
```

```

functionreg: regn GENERIC MAP ( N => 6 )
    PORT MAP ( Func, FRin, Clock, FuncReg );

I <= FuncReg(1 TO 2) ;

decX: dec2to4 PORT MAP ( FuncReg(3 TO 4), High, X );
decY: dec2to4 PORT MAP ( FuncReg(5 TO 6), High, Y );

reg0: regn PORT MAP ( BusWires, Rin(0), Clock, R0 );
reg1: regn PORT MAP ( BusWires, Rin(1), Clock, R1 );
reg2: regn PORT MAP ( BusWires, Rin(2), Clock, R2 );
reg3: regn PORT MAP ( BusWires, Rin(3), Clock, R3 );
regA: regn PORT MAP ( BusWires, Ain, Clock, A );

alu: WITH AddSub SELECT
    Sum <= A + BusWires WHEN '0',
        A - BusWires WHEN OTHERS;

WITH Sel SELECT
    BusWires <= R0    WHEN "100000",
        R1            WHEN "010000",
        R2            WHEN "001000",
        R3            WHEN "000100",
        G             WHEN "000010",
        Data          WHEN OTHERS;

END Behavior;

```

Deliverable:

Hand-drawn block diagram.

Problem 2 (2.5 points, estimated time 15 minutes)

Using CAD tools available to you in the lab (including Constraint Editor) generate the UCF (User Constraint File) for the circuit described using VHDL code posted on the web at

http://ece.gmu.edu/courses/ECE449/exams_S04/section_Th.htm

Assume that the circuit is to be implemented using the XSA-100 board, and the inputs to the circuit are provided using the following sources:

- d - input D0 of the parallel port controlled using GXSPORT
- en – dip switch DIPSW1
- clk - Master Clock of the XSA-100 board
- rst - pushbutton of the XSA-100 board
- prst - input D7 of the parallel port controlled using GXSPORT.

Output q should be used to activate or deactivate the middle (central) segment of the seven segment display available on the board.

Deliverable:

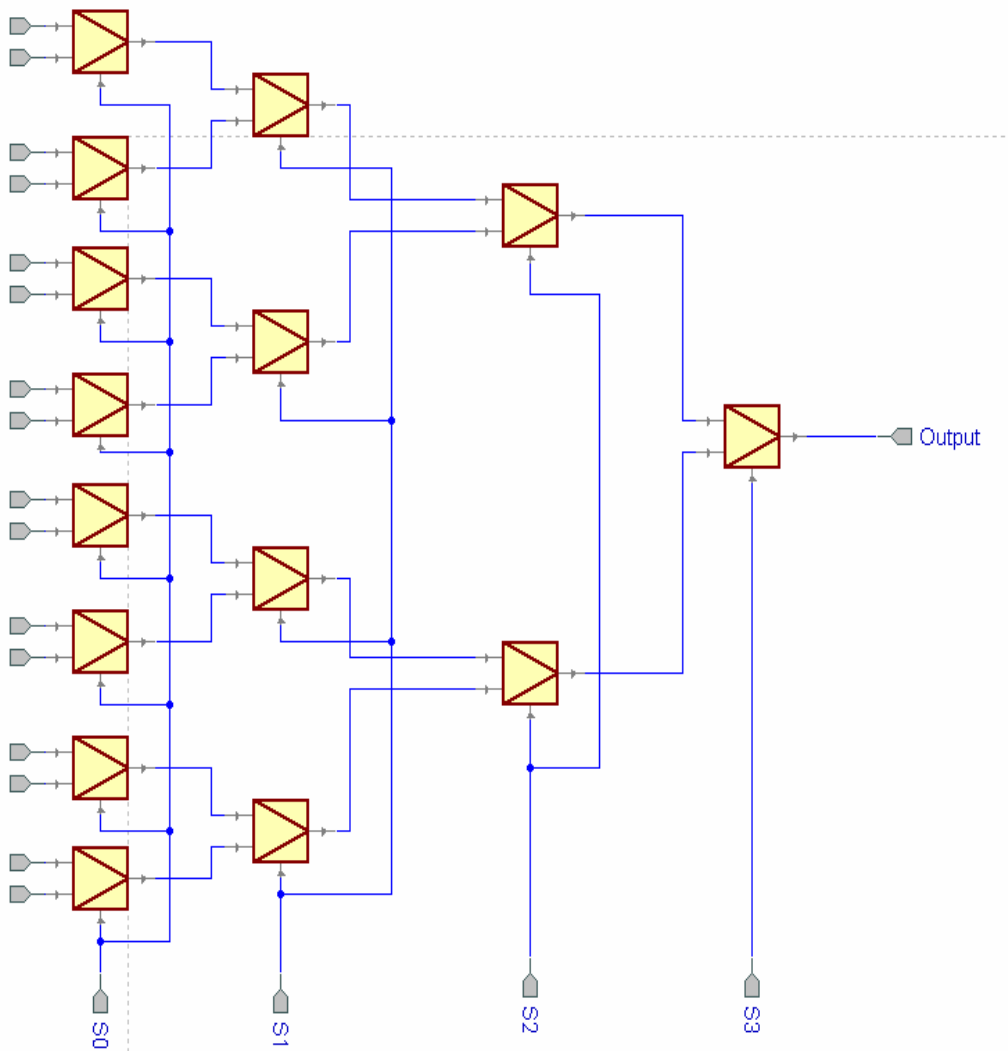
Electronic version of the UCF file, to be stored in the subdirectory Problem2.

Problem 3 (2.5 points, estimated time 15 minutes)

Describe the circuit given below in VHDL using “**for generate**” statements. Label the internal signals based on your code.

Deliverable:

Save an electronic version of the VHDL description in the subdirectory Problem 3.



Problem 4 (2.5 points, estimated time 15 minutes)

Draw the state diagram corresponding to the following VHDL code:

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity PROBLEM4 is
    port (
        CLK: in STD_LOGIC;
        d: in STD_LOGIC;
        q: out STD_LOGIC_VECTOR (1 downto 0));
end;

architecture PROBLEM4_arch of PROBLEM4 is

    type Sreg0_type is (S1, S2, S3, S4);
    signal Sreg0: Sreg0_type;

    attribute syn_state_machine: boolean;
    attribute syn_state_machine of Sreg0: signal is true;

    attribute syn_preserve: boolean;
    attribute syn_preserve of PROBLEM4_arch: architecture is false;

begin

    Sreg0_machine: process (CLK)
    begin
        if CLK'event and CLK = '1' then
            case Sreg0 is
                when S1 =>
                    if d='0' then
                        Sreg0 <= S1;
                    elsif d='1' then
                        Sreg0 <= S3;
                    end if;
                when S2 =>
                    if d='1' then
                        Sreg0 <= S3;
                    elsif d='0' then
                        Sreg0 <= S1;
                    end if;
                when S3 =>
                    if d='1' then
                        Sreg0 <= S3;
                    elsif d='0' then
                        Sreg0 <= S4;
                    end if;
                when S4 =>
                    if d='1' then
                        Sreg0 <= S1;
                    elsif d='0' then
                        Sreg0 <= S2;
                    end if;
                when others =>
            end case;
        end if;
    end process;
end;
```

```
                null;
            end case;
        end if;
    end process;

    q_assignment:
    q <= "01" when (Sreg0 = S1 and d='0') else
        "11" when (Sreg0 = S1 and d='1') else
        "10" when (Sreg0 = S2 and d='1') else
        "00" when (Sreg0 = S2 and d='0') else
        "01" when (Sreg0 = S3 and d='1') else
        "10" when (Sreg0 = S3 and d='0') else
        "10" when (Sreg0 = S4 and d='1') else
        "00" when (Sreg0 = S4 and d='0') else
        "00";

end PROBLEM4_arch;
```

Deliverable:

Hand-drawn block diagram.

Hands-on Design Problem (25 points, estimated time 1 hour)

Circuit description

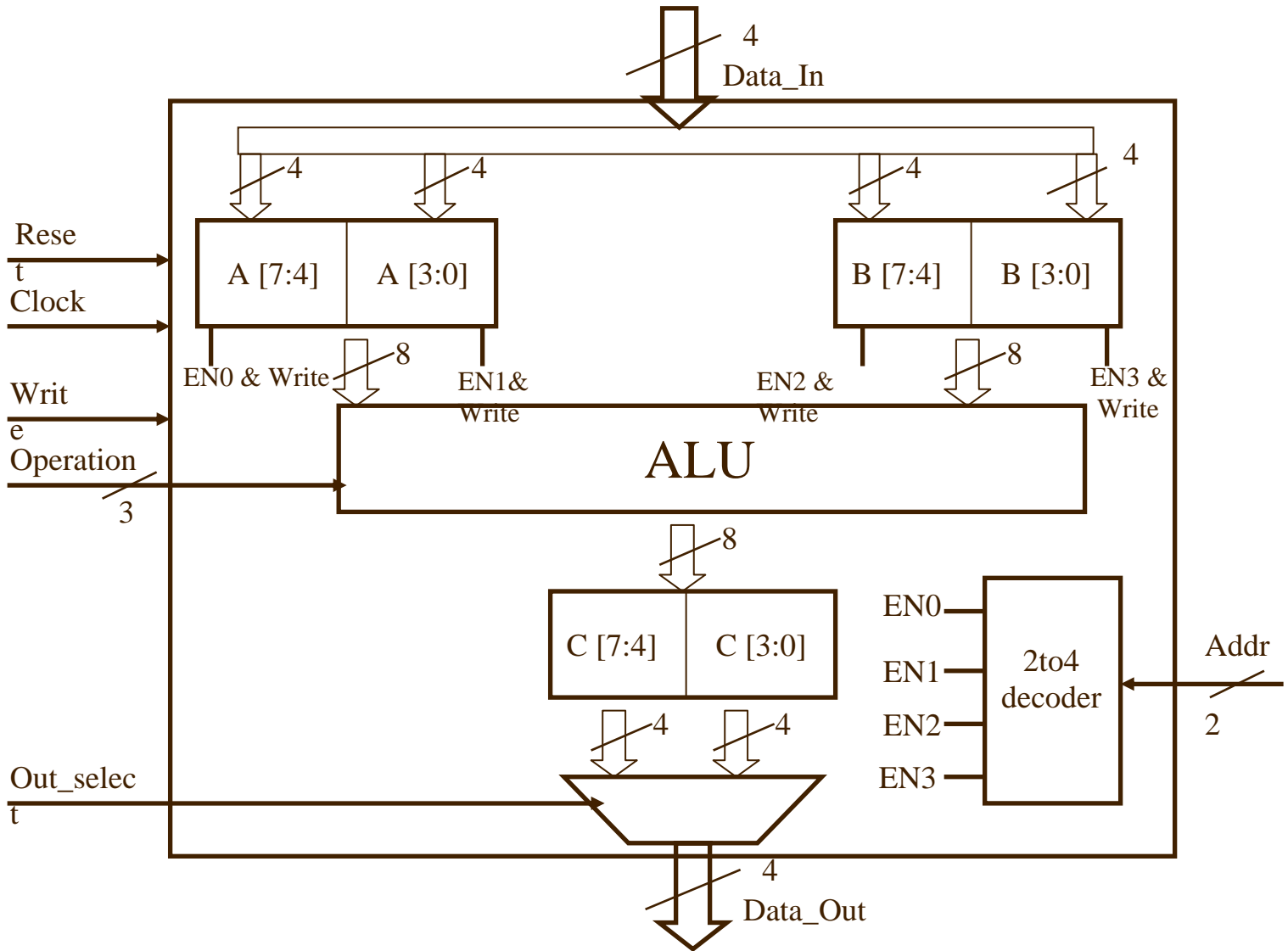
Design and write an RTL VHDL code for an 8 bit Arithmetic-Logic Unit (ALU) capable of performing eight operations specified in the table below.

ALU Functionality		
Operation	Outputs	Opcode
Clear	"00000000"	"000"
ROT3	$A \lll 3$	"001"
SUB	$A - B$	"010"
ADD	$A + B$	"011"
XNOR	$A \text{ XNOR } B$	"100"
NOR	$A \text{ NOR } B$	"101"
NAND	$A \text{ NAND } B$	"110"
Preset	"11111111"	"111"

The block diagram of the circuit is shown in the figure on the next page, and its inputs/outputs are specified in the table below.

Port	Width	Mode	Function
Data_In	4	IN	4 bits of one of the operands
Clock	1	IN	Main Clock
Reset	1	IN	Resets all Registers
Write	1	IN	High during clock periods when operand registers can be written to
Addr	2	IN	Specifies which register sector to enable for writing
Operation	3	IN	Selects between one of eight ALU functions
Out_select	1	IN	High/Low will select the MSB/LSB sector of register C
Data_Out	4	OUT	4 bits of Final Output

8-bit operands are loaded to the registers A and B in 4-bit chunks using the data input Data_In. The two-bit address Addr is used to enable only one 4-bit sector of one of the input registers at a time. The data is stored in an appropriate sector of one of the input registers at the rising edge of Clock, only when the control input Write is high, and the appropriate internal Enable line EN_x is high. The operation to be performed by ALU is determined by the three-bit control input Operation. The result can be read from the register C in 4-bit data chunks using the output Data_Out. The control signal Out_select is used to select one of the two 4-bit sectors of the output register C to be written to Data_Out.



Design Requirements

The combinational portion of the circuit should be described using the dataflow VHDL code, and the sequential portion of the circuit should be described using the synthesizable behavioral code. Your code should infer a circuit that requires a minimum amount of FPGA resources. The target clock frequency should be 50 MHz.

Tasks

Perform the following tasks:

1. Write and debug a VHDL code of your main circuit.
2. Write a testbench verifying the operation of your main circuit.
3. Perform functional simulation of your circuit and use it to debug your VHDL code.
4. Synthesize your main circuit using Synplify Pro in the GUI mode. Save the RTL Netlist diagram.
5. Implement your main circuit using Xilinx ISE.
6. Perform post-synthesis and timing simulations of your circuit using Active-HDL. Check if post-synthesis simulation matches functional simulation. Based on the

circuit block diagram and timing simulation, determine the most critical path in your circuit and its length.

Deliverables (electronic files to be written to the zip disk)

1. VHDL code of your main circuit fulfilling the requirements specified in the *Design Requirements* section above - 10 pts
2. VHDL code of your testbench – 5 points
3. RTL Netlist of your main circuit (extension .srs) – 2 points
4. Simulation waveforms from the post-synthesis simulation proving the correct operation of your circuit – 2 points
5. Simulation waveforms from the timing simulation demonstrating the delay of its critical path – 2 points
6. FPGA resource utilization – 2 points
7. Minimum clock period of your circuit – 2 points.