ECE 448
Midterm Exam – Solutions for Group 2
Monday, March 2, 2020

Problem 1 (15 points)

Perform the following tasks for

a combinational Variable Logical Shifter Right,

with the block diagram shown in Fig. 1.

A. Fill in the blanks in the code of this component given on the next page.
B. Draw a symbol of this component with the correct
   - number
   - direction, and
   - widths

   of ALL inputs and outputs.
   Name each input and output, matching the VHDL code on the next page.

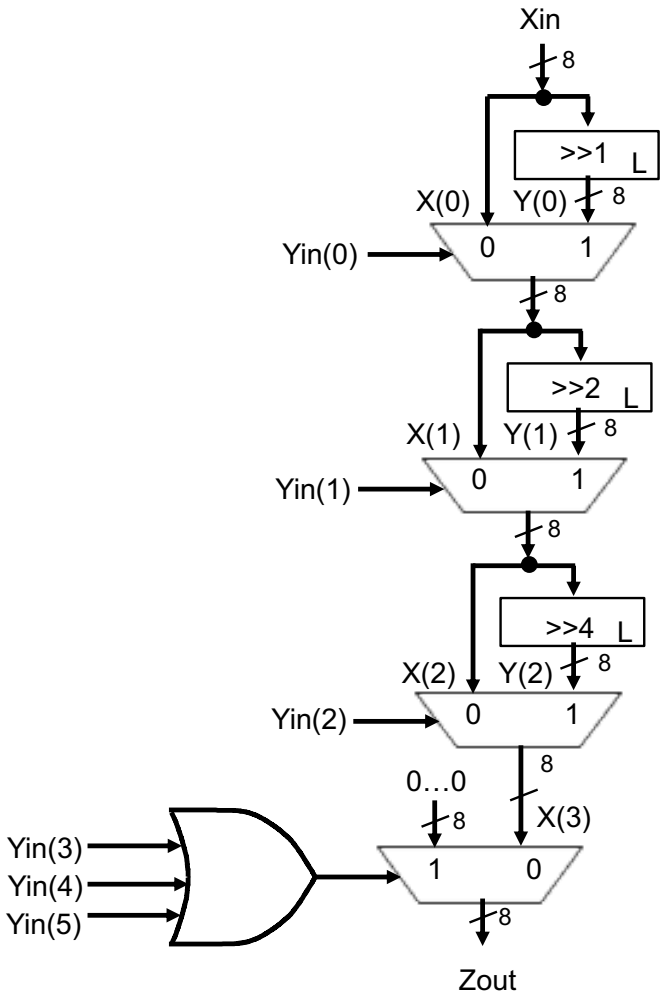


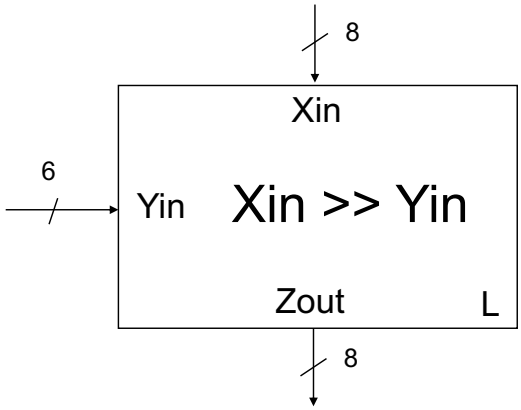
Fig. 1: Block diagram of a combinational Variable Logical Shifter Right

Answer A:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

ENTITY fixed_shifter_right IS
       GENERIC(L : INTEGER := 1);
       PORT(
            a : IN STD_LOGIC_VECTOR(7 downto 0);
            y : OUT STD_LOGIC_VECTOR(7 downto 0)
           );

END fixed_shifter_right;

ARCHITECTURE dataflow OF fixed_shifter_right IS
BEGIN
     y(7-L downto 0)   <= a(7 downto L);
     y(7 downto 7-L+1)<= OTHERS => '0';
END dataflow;

library IEEE;
use IEEE.STD_LOGIC_1164.all;

ENTITY variable_shifter_right IS
       PORT(
           Xin : IN STD_LOGIC_VECTOR(7 downto 0);
           Yin : IN STD_LOGIC_VECTOR(5 downto 0);
           Zout : OUT STD_LOGIC_VECTOR(7 downto 0)
 );
END variable_shifter_right;

ARCHITECTURE mixed OF variable_shifter_right IS

TYPE array8 IS ARRAY (0 to 3) OF STD_LOGIC_VECTOR(7 DOWNTO 0);

SIGNAL X : array8;
SIGNAL Y : array8;


BEGIN
     X(0) <= Xin;
     G: FOR i IN 0 TO 2 GENERATE

           SHIFT_I: ENTITY work.fixed_shifter_right(dataflow)
                GENERIC MAP (L => 2**i)
                PORT MAP ( a => X(i),
                           y => Y(i));

           X(i+1) <=  Y(i) WHEN Yin(1)='1'  ELSE X(i);

     END GENERATE;
```

```
Zout <= OTHERS => '0'  WHEN (Yin(5) OR Yin(4) OR Yin(3)) = '1'
        ELSE X(3);

END mixed;
```

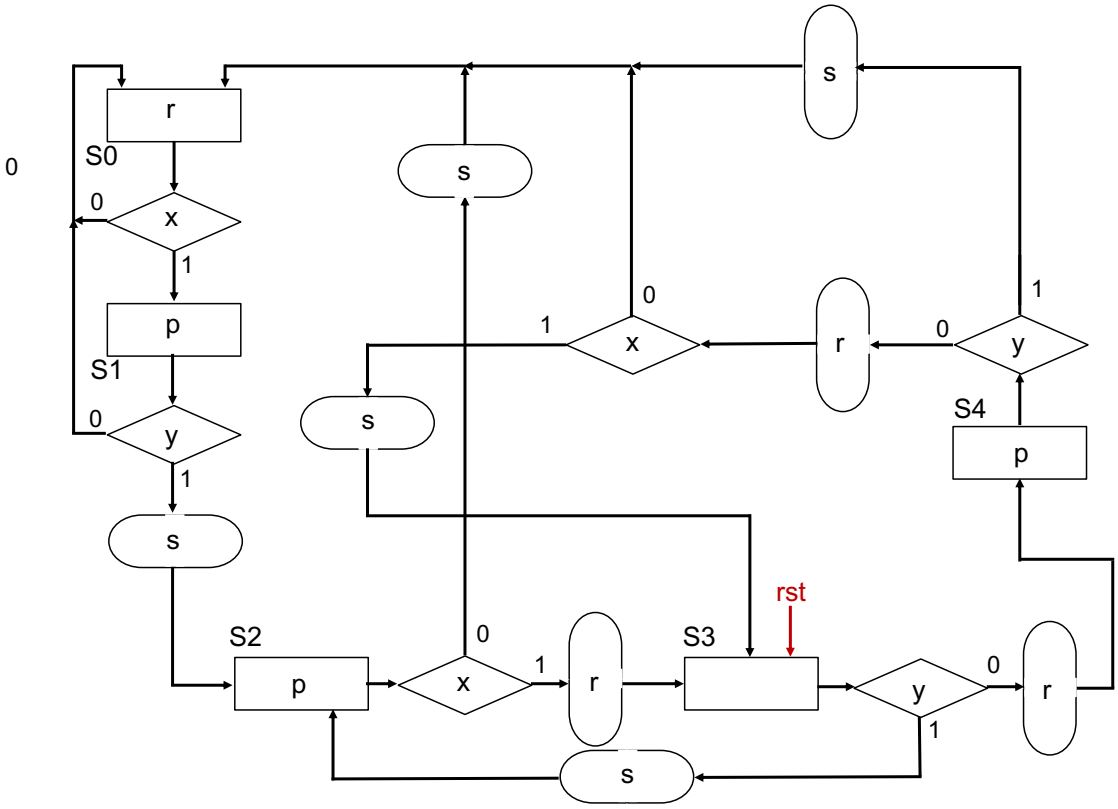
Answer B:



Problem 2

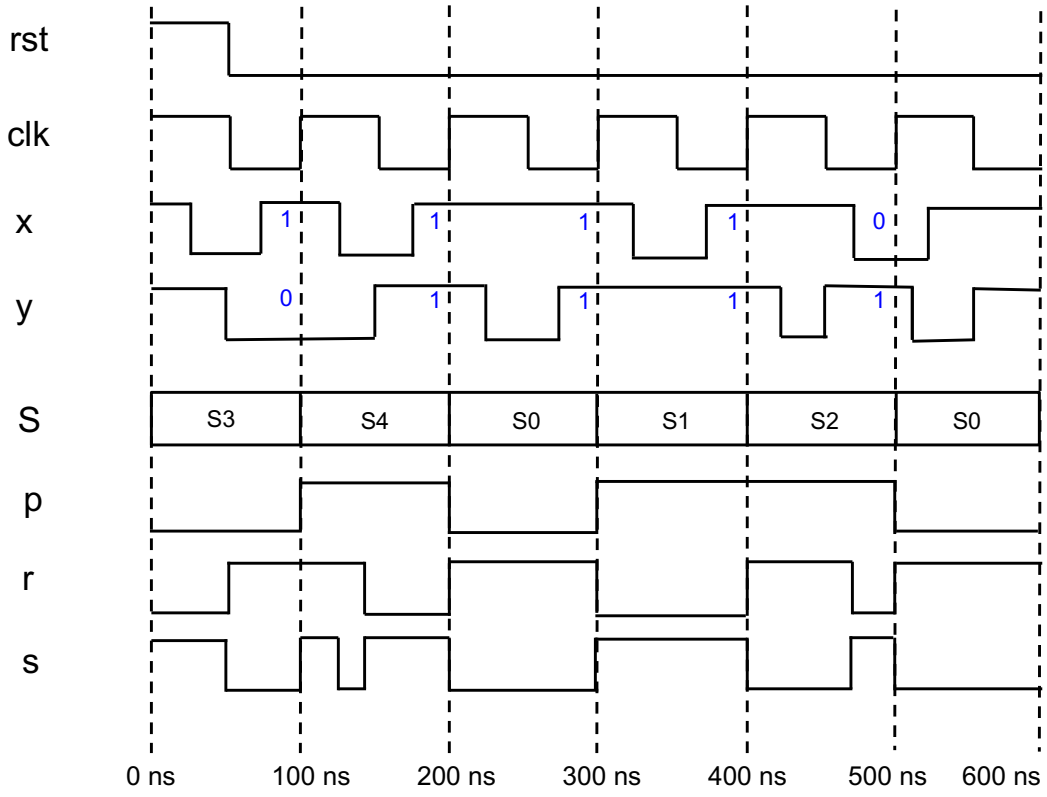
Assuming the controller described using the given below ASM chart:

A. Supplement timing waveforms given on the next page with the values of the **state S**, and the values of the **outputs p**, **r**, and **s.**
B. Fill in the blanks in the fragment of the code of this component given on the next page.



Answer A:

Be careful regarding the initial value of the state S after reset!



Answer B:

```
ARCHITECTURE behavioral of Controller IS
TYPE state IS (S0, S1, S2, S3, S4);
SIGNAL state_reg, state_next: state;

BEGIN
P1: PROCESS(clk, rst)
BEGIN
      IF(rst = '1') THEN
            state_reg <= S3;
      ELSIF rising_edge(clk) THEN
            state_reg <= state_next;
        END IF;
END PROCESS;

Next_State_Output:
PROCESS (state_reg, x, y)
BEGIN
      state_next <= state_reg;
      p <= '0';
      r <= '0';
      s <= '0';

      CASE state_reg IS

                WHEN S2 =>
                  p <='1';
                  if x='1' then
                        r <='1';
                        state_next <= S3;
                  else
                        s <='1';
                        state_next <= S0;
                  end if;

                WHEN S3 =>
                  if y='1' then
                        s <='1';
                        state_next <= S2;
                  else
                        r <='1';
                        state_next <= S4;
                  end if;
```

Please provide the code only for the cases when state_reg = S2 and state_reg = S3.

Problem 3

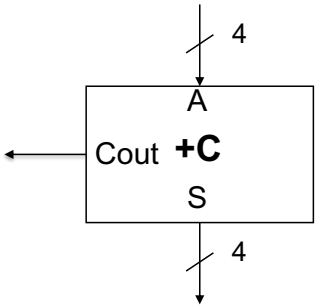
Perform the following tasks for a component

taking a 4-bit input A of the type std_logic_vector,
adding to it a constant, given by the generic C of the type natural (in the range 0 to 15), and returning the sum S of the type std_logic_vector and the carry out Cout of the type std_logic.

A. Draw a symbol of this component with the correct
   - number
   - direction, and
   - widths

   of ALL inputs and outputs.
   Name each input and output.
B. Write the full synthesizable code of this design entity, including entity declaration and architecture body, using the package numeric_std.
C. **(bonus)** Draw an internal block diagram of this component, composed of medium logic components and/or basic logic gates for the case of C=6.

Answer A:



Answer B:

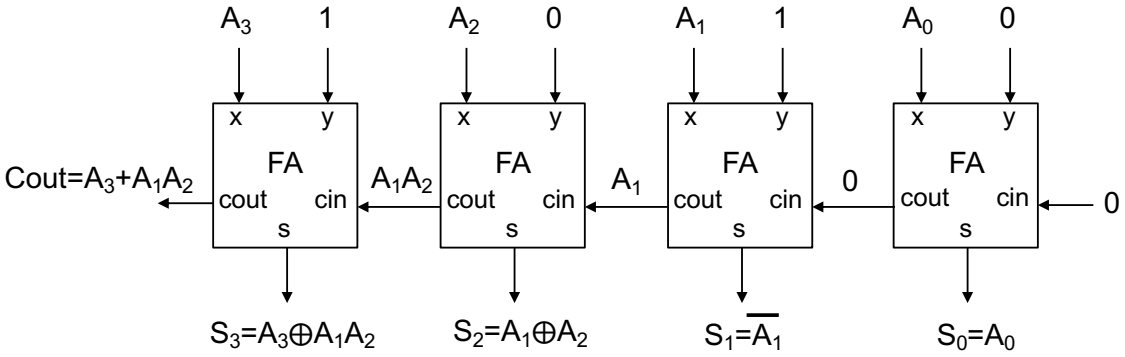
You can use the following table of overloaded operators in the IEEE numeric_std package

overloaded operator	description	data type of operand a	data type of operand b	data type of result
abs a - a	absolute value negation	signed		signed
a * b a / b a mod b a rem b a + b a - b	arithmetic operation	unsigned unsigned, natural signed signed, integer	unsigned, natural unsigned signed, integer signed	unsigned unsigned signed signed
a = b a /= b a < b a <= b a > b a >= b	relational operation	unsigned unsigned, natural signed signed, integer	unsigned, natural unsigned signed, integer signed	boolean boolean boolean boolean

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

ENTITY plusC IS
   GENERIC ( C : NATURAL RANGE 0 TO 15 := 6 ) ;
   PORT ( A : IN STD_LOGIC_VECTOR(3 downto 0);
          S : OUT STD_LOGIC_VECTOR(3 downto 0);
          Cout : OUT STD_LOGIC);
END plusC;

ARCHITECTURE dataflow OF plusC IS
  SIGNAL Sum : UNSIGNED(4 downto 0);
BEGIN
  Sum <= unsigned('0' & A) + C;
  S <= std_logic_vector(Sum(3 downto 0));
  Cout <= Sum(4);
END dataflow;
```

Answer C:

Derivation:



Final Optimized Circuit: