# Q1

1. Draw a block diagram of a digital circuit with the following specifications (20 Marks).

Interface:

Din – 16-bit data input

Dout – 16-bit data output

Addr – 3-bit address of the location (register) where input data is stored and output data is read

from

Read – control signal indicating read

Write – control signal indicating write

Clk – clock

Functionality:

• If Write = 1, then at the next rising edge of the clock, data from the input Din is stored

in the internal location given by the address Addr.

• If Read = 1, data from the location given by the address Addr is transferred to the output

Dout, and the contents of the internal memory (registers) does not change. If Read=0,

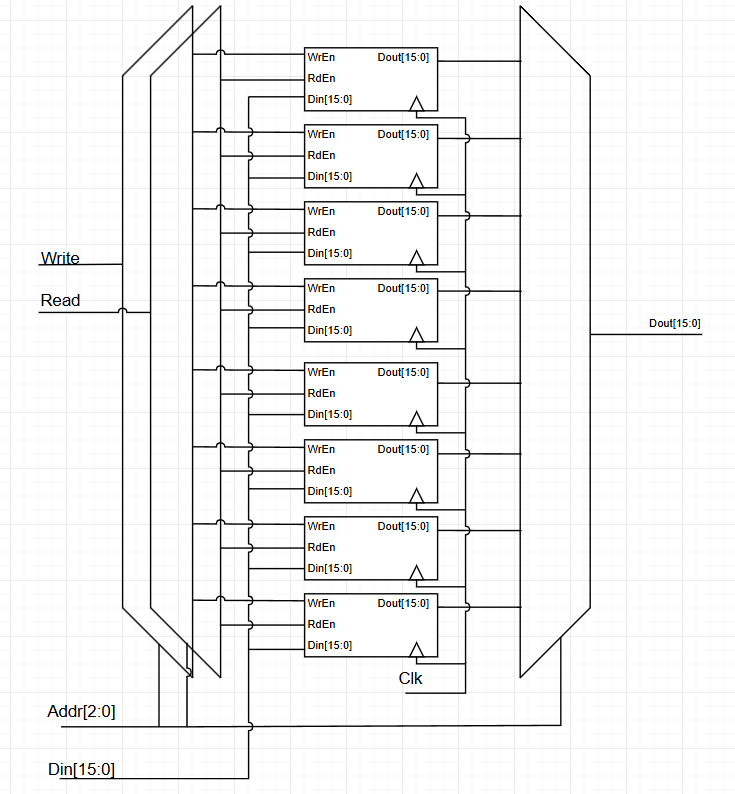
the output Dout should be set to the high impedance state. Assume that the internal

memory is implemented using registers. Use only medium scale components, such as

registers, multiplexers, encoders, decoders, buffers, etc.

You are not allowed to use RAM in your circuit.

# Answer Q1:



# Q2

The VHDL code given below is describing a block diagram of a circuit. Draw the

architecture RTL that is described by this code. (15 Marks)

LIBRARY ieee ;

USE ieee.std\_logic\_1164.all ;

ENTITY variable\_rotator is

PORT(

A : IN STD\_LOGIC\_VECTOR(15 downto 0);

B : IN STD\_LOGIC\_VECTOR(3 downto 0);

C : OUT STD\_LOGIC\_VECTOR(15 downto 0)

);

END variable\_rotator;

ARCHITECTURE structural OF variable\_rotator IS

TYPE array16 IS ARRAY (0 to 4) OF STD\_LOGIC\_VECTOR(15 DOWNTO 0);

SIGNAL Al : array16;

SIGNAL Ar : array16;

BEGIN

Al(0) <= A;

G:

FOR i IN 0 TO 3 GENERATE

Ar(i) <= Al(i)(15-2\*\*i downto 0) & Al(i)(15 downto 15-2\*\*i+1);

Al(i+1) <= Al(i) when B(i)=‘0’ else Ar(i);

END GENERATE;

C <= Al(4);

END dataflow;

# Answer Q2:

# Q3

Consider a cell-based 8-bit × 8-bit Cary-save array multiplier. (15 Marks)

a. Draw the architecture schematic of the RTL.

b. Estimate how many AND gates and Full adders it requires (Assume half adders

are implemented by full adders).

c. What is the critical-path delay of this multiplier, assuming an AND gate delay is

t\_a, and adder delay is t\_add

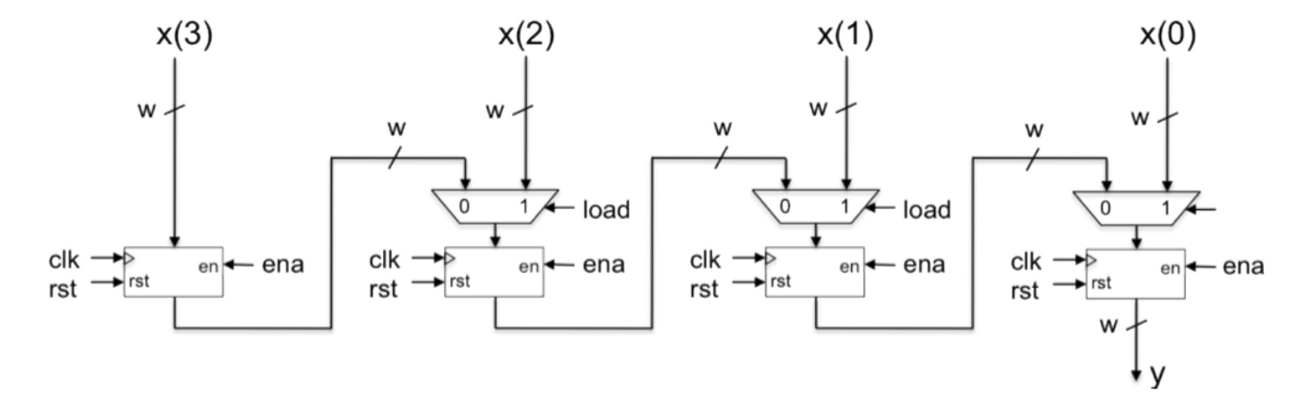
# Answer Q3:

# Q4

This circuit is called parallel-in-serial-out. Fill in the blanks (dashed lines) in the given

VHDL codes which is for entity declaration and architecture, with the size of the output

bus, w, treated as a generic with the default value equal to 8. (20 Marks)



|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;    entity piso is  generic (  w : integer := …………………);  port (  clk : in std\_logic;  ena : in std\_logic;  load : in std\_logic;  rst : in std\_logic;  x : in std\_logic\_vector(……………………. downto 0);  y : out std\_logic\_vector(w-1 downto 0));  end piso;    architecture behavioral of piso is  type bus\_array is array (3 downto 0) of std\_logic\_vector(w-1  downto 0);  signal reg, mux : bus\_array;  begin  mux(3) <= x(4\*w-1 downto ……………………);  mux\_gen : for i in 2 downto 0 generate  mux(i) <= x(………………………downto i\*w) when load = '1' else  reg(i+1);  end generate;    regx\_gen : for i in 3 downto 0 generate  regx : process ( clk )  begin  if ……………………………………. then  if rst = '1' then  reg(i) <= (……………………………);  elsif ena = '1' then  reg(i) <= mux(i);  end if;  end if;  end process;  end generate;    y <= reg(0);  end behavioral; |

# Answer Q4

|  |
| --- |
| library ieee;  use ieee.std\_logic\_1164.all;    entity piso is  generic (  w : integer := **4**);  port (  clk : in std\_logic;  ena : in std\_logic;  load : in std\_logic;  rst : in std\_logic;  x : in std\_logic\_vector(**w-1** downto 0);  y : out std\_logic\_vector(w-1 downto 0));  end piso;    architecture behavioral of piso is  type bus\_array is array (3 downto 0) of std\_logic\_vector(w-1  downto 0);  signal reg, mux : bus\_array;  begin  mux(3) <= x(4\*w-1 downto ……………………);  mux\_gen : for i in 2 downto 0 generate  mux(i) <= x(………………………downto i\*w) when load = '1' else  reg(i+1);  end generate;    regx\_gen : for i in 3 downto 0 generate  regx : process ( clk )  begin  if ……………………………………. then  if rst = '1' then  reg(i) <= (……………………………);  elsif ena = '1' then  reg(i) <= mux(i);  end if;  end if;  end process;  end generate;    y <= reg(0);  end behavioral; |