1. Problem/Objective

State the problem statement and/or objective of the lab. This must be a complete paragraph (i.e., at least 5 sentences).

The objective of this lab was to use PicoBlaze assembly to write a program that included a barrel shifter. We coded the barrel shifter in PicoBlaze and simulated it, then we modified the code to work on the Digilent S3 board. There were four types of shifts that we needed to code, shift left, shift right, rotate left, and rotate right. We needed to shift a certain amount of bits within one clock cycle using either type.

2. Methodology

You are to search on the web (i.e., CSUF library, Google Scholar, etc.) to find a scholarly paper (i.e., 1 paper) on an application of a barrel shifter. Briefly describe (about 2 complete paragraphs) what the paper is about.

The paper chosen was "Design of reversible bidirectional logarithmic barrel shifter" by Mrinal Goswami, Aron Narzary, and Govind Raj. In this study they talk about bit rotation and shifting as very significant operations in digital logic applications. They state that reversible logic has been on the climb because of its loss-less information processing and low power dissipation within logic synthesis. The paper talks about the reversible bidirectional barrel shifter which is able to rotate bits in both directions.

They talk about the test feature of this barrel shifter which is analyzed through the integer linear program method. This method has reported 7 test vectors of size 6 as a minimal test set for single and multiple stuck-at-faults. They go into detail first about how a simple barrel shifter works using logic schematics of the designs, then they talk about how the reversible barrel shifter will implement the Fredkin gate. This gate is used for multiplexing in their design.

3. Question(s)

a. What is a common usage for a barrel shifter? Why?

A common usage for barrel shifter is within the hardware implementation of floating-point arithmetic. The coefficients of two numbers while adding or subtracting floating-point, must be aligned. You need to shift the smaller number to the right to increase the exponent to match the larger exponent number. This is done when subtracting the exponents, also using a barrel shift to the right by the difference within one clock cycle.

b. What operation of the barrel shifter (i.e., left shift, right shift, left rotate, or right rotate) is being shown in Figure 1?

Below the operation of barrel shifter being used is a left shift of multiple places.

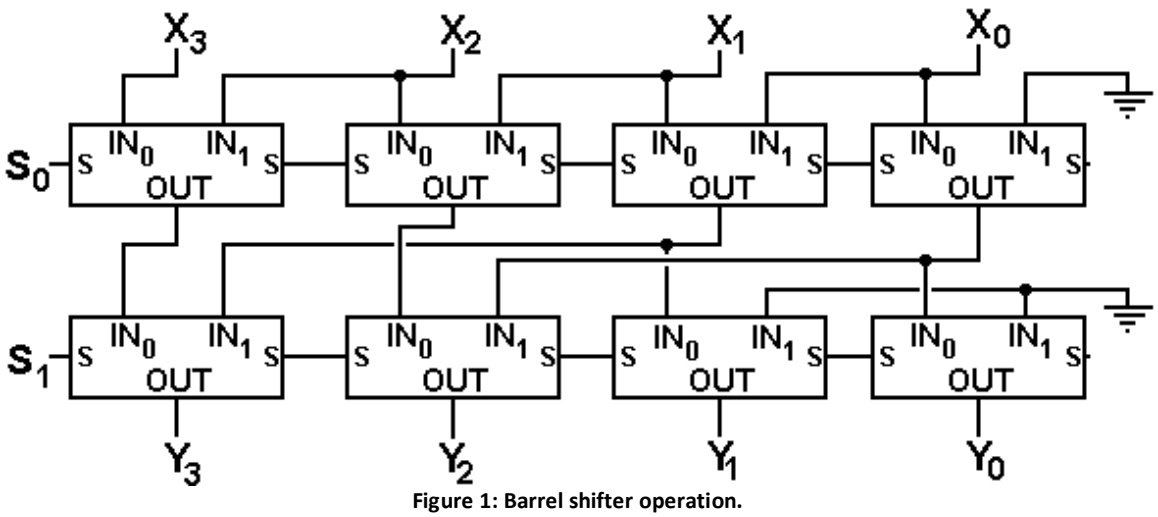
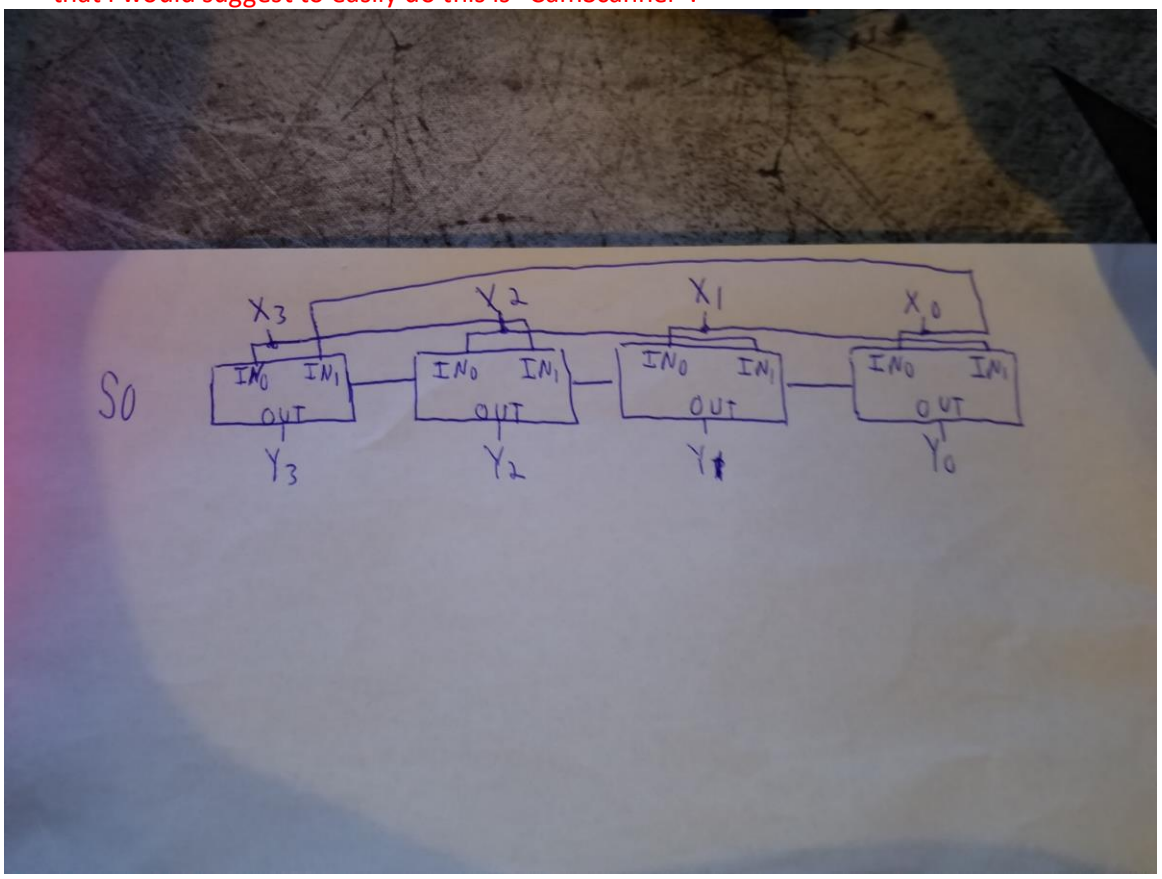


Figure 1: Barrel shifter operation.

c. Now, in a similar fashion to part b, draw the logic schematic for the barrel shifter rotate right operation. You do the drawing by hand, take a picture, and paste/import the image here. One app that I would suggest to easily do this is "CamScanner".



4. Program Code

Copy your code here. Please provide comments in your code. This will help me analyze your code and remove any ambiguity. **Provide your code as text, not as a screenshot/image**.

PicoBlaze Assembly:

```
; PBlazeIDE Template

;====================================================
; generate vhdl file
;====================================================
vhdl          "rom_form.vhd", "lab_6.vhd", "lab_6"

;====================================================
; data constants
;====================================================
amount_mask                 equ  $03
type_mask           equ      $30

;====================================================
; register aliases
;====================================================
data_in             equ      s0             ;data in register
data                equ      s1             ;temp data register
type_in             equ      s2             ;type of shift
amount_in           equ      s3             ;amount of manipulations
sw_in               equ      sF

;====================================================
; port aliases
;====================================================
; input ports
sw_port             dsin     $00

; output ports
led_port            dsout    $80

;====================================================
; main program
;====================================================
       call init
main_loop:
                eint
                out data,led_port
                jump main_loop

;====================================================
; subroutines
;====================================================
left_shift:
               comp amount_in,$00
               jump z,main_loop
               SL0 data
```

```
                sub amount_in,$01
                jump left_shift

right_shift:
        comp amount_in,$00
                jump z,main_loop
                SR0 data
                sub amount_in,$01
                jump right_shift

left_rotate:
                comp amount_in,$00
                jump z,main_loop
                rl data
                sub amount_in,$01
                jump left_rotate

right_rotate:
                comp amount_in,$00
                jump z,main_loop
                rr data
                sub amount_in,$01
                jump right_rotate

shift_type:
                comp type_in,$00
                call z,left_shift
                comp type_in,$01
                call z,right_shift
                comp type_in,$02
                call z,left_rotate
                comp type_in,$03
                call z,right_rotate
                ret

init:
        load data_in,$3C
        load data,$3C
        load type_in,$00
        load amount_in,$00
        ret
;=====================================================
; time critical segment
;=====================================================
critical_timing:
                                dint
                                eint
                                ret
```

```
;=====================================================
; interrupt service routine (isr)
;=====================================================
isr:
    comp data,$00               ;compare data if empty to enable another iteration
    jump z,reset_data
    jump dont_reset_data
reset_data:
    load data,$3C               ;reset data to default "00111100"
dont_reset_data:
    in type_in,sw_in
    and type_in,type_mask       ;mask all bits but type bits 5,4
    SR0 type_in                 ;shift bits to be 2 LSBs
    SR0 type_in
    SR0 type_in
    SR0 type_in
    in amount_in,sw_in
    and amount_in,amount_mask   ;mask all bits but 2 LSBs for amount
    call shift_type             ;call to determine shift type
    reti enable

;=====================================================
; interrupt vector
;=====================================================
org $3ff
jump isr
```

Top Level:

```
library ieee;
use ieee.std_logic_1164.all;

entity top_level is
    port (
        int_butt:in std_logic;
        switches : in std_logic_vector(7 downto 0);
        clk : in std_logic;
        leds : out std_logic_vector(7 downto 0)
    );
end top_level;

architecture behavioral of top_level is
    -- declaration of kcpsm3 (always use this declaration to call
    -- up picoblaze core)
    component kcpsm3
        port (
            address : out std_logic_vector(9 downto 0);
            instruction : in std_logic_vector(17 downto 0);
```

```
                port_id : out std_logic_vector(7 downto 0);
                write_strobe : out std_logic;
                out_port : out std_logic_vector(7 downto 0);
                read_strobe : out std_logic;
                in_port : in std_logic_vector(7 downto 0);
                interrupt : in std_logic;
                interrupt_ack : out std_logic;
                reset : in std_logic;
                clk : in std_logic
        );
    end component;
    ----------------------------------------------------------------------
    -- declaration of program memory (here you will specify the entity name
    -- as your .psm prefix name)
    component lab_6
        port (
                address : in std_logic_vector(9 downto 0);
                instruction : out std_logic_vector(17 downto 0);
                clk : in std_logic
        );
    end component;
    ----------------------------------------------------------------------
    -- signals used to connect picoblaze core to program memory and i/o logic
    signal address : std_logic_vector(9 downto 0);
    signal instruction : std_logic_vector(17 downto 0);
    signal port_id : std_logic_vector(7 downto 0);
    signal out_port : std_logic_vector(7 downto 0);
    signal in_port : std_logic_vector(7 downto 0);
    signal write_strobe : std_logic;
    signal read_strobe : std_logic;
    signal interrupt_ack : std_logic;
    signal reset : std_logic;
    -- the following input is assigned an inactive value since it is
    -- unused in this example
    signal interrupt : std_logic := '0';
    signal bounce: std_logic := '0';
    ----------------------------------------------------------------------
    -- start of circuit description
begin
    -- instantiating the picoblaze core
    processor : kcpsm3
    port map(
        address => address,
        instruction => instruction,
        port_id => port_id,
        write_strobe => write_strobe,
        out_port => out_port,
        read_strobe => read_strobe,
```

```
            in_port => in_port,
            interrupt => interrupt,
            interrupt_ack => interrupt_ack,
            reset => reset,
            clk => clk
    );

    -- instantiating the program memory
    program : lab_6
    port map(
            address => address,
            instruction => instruction,
            clk => clk
    );




    -- connect i/o of picoblaze
    ----------------------------------------------------------------------
    ----- TO DO: kcpsm3 define input ports
    ----------------------------------------------------------------------
    input_ports : process(clk)
    begin
            if (rising_edge(clk)) then
                    case port_id(1 downto 0) is
                    -- read simple toggle switches and buttons at address 00 hex
                            when "00" =>
                                    in_port <= switches;
                                    -- don't care used for all other addresses to ensure minimum
                                    -- logic implementation
                            when others =>
                                    in_port <= "XXXXXXXX";
                    end case;
            end if; end
process input_ports;
    ----------------------------------------------------------------------
    -- TO DO: kcpsm3 define output ports
    ----------------------------------------------------------------------
    -- adding the output registers to the processor at address 80 hex
    output_ports : process(clk)
    begin
            if (rising_edge(clk)) then
                    if port_id(7) = '1' then
                            leds <= out_port;
                    end if;
            end if;
    end process output_ports;
```

```
    ----------------------------------------------------------------------
    -- TO DO: kcpsm3 define interrupt control
    ----------------------------------------------------------------------
    int_control : process(clk)
    begin
            if (reset = '1') then
                    interrupt <= '0';
            elsif (rising_edge(clk)) then
                    if (interrupt_ack = '1') then
                            interrupt <= '0';
                    elsif(int_butt = '1' and bounce = '0') then
                            interrupt <= '1';
                            bounce <= '1';
                    elsif(int_butt = '0') then
                            bounce <= '0';
                    end if;
            end if;
    end process int_control;

end behavioral;
```

Lab_6.vhd:

```
--
-- Definition of a single port ROM for KCPSM3 program defined by lab_6.psm
--
-- Generated by KCPSM3 Assembler {timestamp}.
--
-- Standard IEEE libraries
--
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
-- The Unisim Library is used to define Xilinx primitives. It is also used during
-- simulation. The source can be viewed at %XILINX%\vhdl\src\unisims\unisim_VCOMP.vhd
--
library unisim;
use unisim.vcomponents.all;
--
--
entity lab_6 is
   Port (      address : in std_logic_vector(9 downto 0);
         instruction : out std_logic_vector(17 downto 0);
               clk : in std_logic);
   end lab_6;
--
```

```
architecture low_level_definition of lab_6 is
--
-- Attributes to define ROM contents during implementation synthesis.
-- The information is repeated in the generic map for functional simulation
--
attribute INIT_00 : string;
attribute INIT_01 : string;
attribute INIT_02 : string;
attribute INIT_03 : string;
attribute INIT_04 : string;
attribute INIT_05 : string;
attribute INIT_06 : string;
attribute INIT_07 : string;
attribute INIT_08 : string;
attribute INIT_09 : string;
attribute INIT_0A : string;
attribute INIT_0B : string;
attribute INIT_0C : string;
attribute INIT_0D : string;
attribute INIT_0E : string;
attribute INIT_0F : string;
attribute INIT_10 : string;
attribute INIT_11 : string;
attribute INIT_12 : string;
attribute INIT_13 : string;
attribute INIT_14 : string;
attribute INIT_15 : string;
attribute INIT_16 : string;
attribute INIT_17 : string;
attribute INIT_18 : string;
attribute INIT_19 : string;
attribute INIT_1A : string;
attribute INIT_1B : string;
attribute INIT_1C : string;
attribute INIT_1D : string;
attribute INIT_1E : string;
attribute INIT_1F : string;
attribute INIT_20 : string;
attribute INIT_21 : string;
attribute INIT_22 : string;
attribute INIT_23 : string;
attribute INIT_24 : string;
attribute INIT_25 : string;
attribute INIT_26 : string;
attribute INIT_27 : string;
attribute INIT_28 : string;
attribute INIT_29 : string;
attribute INIT_2A : string;
```

```
    attribute INIT_2B : string;
    attribute INIT_2C : string;
    attribute INIT_2D : string;
    attribute INIT_2E : string;
    attribute INIT_2F : string;
    attribute INIT_30 : string;
    attribute INIT_31 : string;
    attribute INIT_32 : string;
    attribute INIT_33 : string;
    attribute INIT_34 : string;
    attribute INIT_35 : string;
    attribute INIT_36 : string;
    attribute INIT_37 : string;
    attribute INIT_38 : string;
    attribute INIT_39 : string;
    attribute INIT_3A : string;
    attribute INIT_3B : string;
    attribute INIT_3C : string;
    attribute INIT_3D : string;
    attribute INIT_3E : string;
    attribute INIT_3F : string;
    attribute INITP_00 : string;
    attribute INITP_01 : string;
    attribute INITP_02 : string;
    attribute INITP_03 : string;
    attribute INITP_04 : string;
    attribute INITP_05 : string;
    attribute INITP_06 : string;
    attribute INITP_07 : string;
    --
    -- Attributes to define ROM contents during implementation synthesis.
    --
    attribute INIT_00 of ram_1024_x_18 : label is
"500143004009C301010E500143004004C3010106500143004001C180C0010021";
    attribute INIT_01 of ram_1024_x_18 : label is
"10134203100E4202100942011004420040l3C301010C50014300400EC3010102";
    attribute INIT_02 of ram_1024_x_18 : label is
"020EA23052F0013C402D502C4100A000C001C000A00003000200013C003CA000";
    attribute INIT_03 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000080010018A30353F0020E020E020E";
    attribute INIT_04 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_05 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_06 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_07 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
```

```
    attribute INIT_08 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_09 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_0A of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_0B of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_0C of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_0D of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_0E of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_0F of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_10 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_11 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_12 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_13 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_14 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_15 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_16 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_17 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_18 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_19 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_1A of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_1B of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_1C of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_1D of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_1E of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_1F of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
```

```
    attribute INIT_20 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_21 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_22 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_23 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_24 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_25 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_26 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_27 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_28 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_29 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_2A of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_2B of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_2C of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_2D of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_2E of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_2F of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_30 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_31 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_32 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_33 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_34 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_35 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_36 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_37 of ram_1024_x_18 : label is
"000000000000000000000000000000000000000000000000000000000000000000000000";
```

```
    attribute INIT_38 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_39 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_3A of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_3B of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_3C of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_3D of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_3E of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INIT_3F of ram_1024_x_18 : label is
"4029000000000000000000000000000000000000000000000000000000000000";
    attribute INITP_00 of ram_1024_x_18 : label is
"0000000000000000000000000000003C2A80F6F802DDDDDB76DDB76DEF";
    attribute INITP_01 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INITP_02 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INITP_03 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INITP_04 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INITP_05 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INITP_06 of ram_1024_x_18 : label is
"0000000000000000000000000000000000000000000000000000000000000000";
    attribute INITP_07 of ram_1024_x_18 : label is
"C000000000000000000000000000000000000000000000000000000000000000";
    --
    begin
    --
     --Instantiate the Xilinx primitive for a block RAM
     ram_1024_x_18: RAMB16_S18
     --synthesis translate_off
     --INIT values repeated to define contents for functional simulation
     generic map ( INIT_00 =>
X"500143004009C301010E500143004004C3010106500143004001C180C0010021",
            INIT_01 =>
X"10134203100E4202100942011004420040l3C301010C50014300400EC3010102",
            INIT_02 =>
X"020EA23052F0013C402D502C4100A000C001C000A00003000200013C003CA000",
            INIT_03 =>
X"0000000000000000000000000000000080010018A30353F0020E020E020E",
```

```
            INIT_04                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_05                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_06                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_07                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_08                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_09                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_0A                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_0B                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_0C                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_0D                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_0E                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_0F                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_10                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_11                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_12                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_13                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_14                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_15                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_16                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_17                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_18                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_19                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_1A                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_1B                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
            INIT_1C                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_1D                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_1E                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_1F                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_20                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_21                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_22                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_23                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_24                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_25                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_26                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_27                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_28                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_29                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_2A                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_2B                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_2C                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_2D                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_2E                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_2F                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_30                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_31                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_32                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
            INIT_33                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
                INIT_34                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_35                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_36                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_37                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_38                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_39                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_3A                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_3B                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_3C                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_3D                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_3E                                                                 =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INIT_3F                                                                 =>
X"4029000000000000000000000000000000000000000000000000000000000000",
                INITP_00                                                                =>
X"0000000000000000000000000000000003C2A80F6F802DDDDDB76DDB76DEF",
                INITP_01                                                                =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INITP_02                                                                =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INITP_03                                                                =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INITP_04                                                                =>
X"00000000000000000000000000000000000000000000000000000000000000000",
                INITP_05                                                                =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INITP_06                                                                =>
X"0000000000000000000000000000000000000000000000000000000000000000",
                INITP_07                                                                =>
X"C000000000000000000000000000000000000000000000000000000000000000")
        --synthesis translate_on
        port map(    DI => "0000000000000000",
                    DIP => "00",
                     EN => '1',
                     WE => '0',
                    SSR => '0',
                    CLK => clk,
                   ADDR => address,
```

```
            DO => instruction(15 downto 0),
           DOP => instruction(17 downto 16));
--
end low_level_definition;
--
------------------------------------------------------------------------------
--
-- END OF FILE lab_6.vhd
--
------------------------------------------------------------------------------
```