Lab Report No. 06:

Memory

By:

Jiaqing Wan(16), Jiaxin Zheng(19)

Instructor: Dr. Xi Zhou

Classnumber: ZM1501

NAU ID:

5358836, 5358840

November 16, 2017

I. INTRODUCTION

A. Objectives

At the completion of this lab, we will be able to implement a memory module on the Cyclone V GX FPGA, write initial data into the memory, write data to the memory using the input signals, and read data from the memory.

B. Background knowledge

Alternatively referred to as main memory, primary memory, or system memory, Random Access Memory (RAM) is a hardware device that allows information to be stored and retrieved on a computer. RAM is usually associated with DRAM, which is a type of memory module. Because information is accessed randomly instead of sequentially like it is on a CD or hard drive, the computer can access the data much faster. However, unlike ROM or the hard drive, RAM is a volatile memory and requires power to keep the data accessible. If the computer is turned off, all data contained in RAM is lost.

C. Simulation strategy

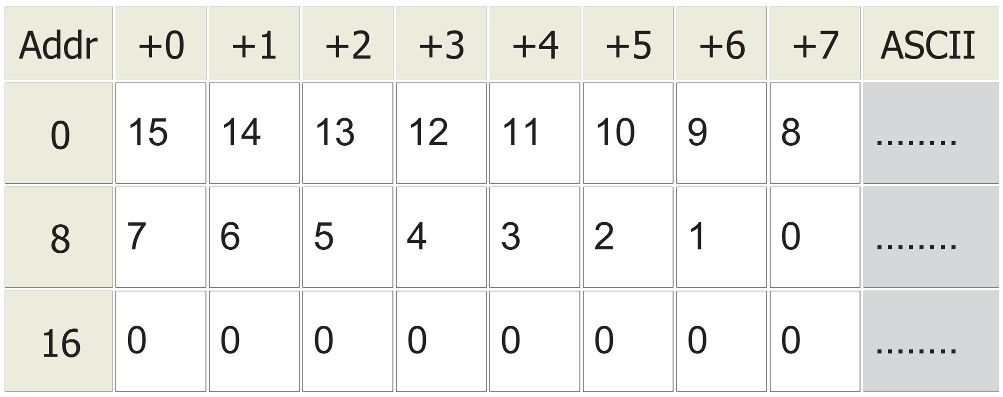
ramlpm:

- **Write Mode**: wren is 1; clock is rising edge; address is a constant; output q is follow by input data. data is write to memory.
- **Read Mode**: wren is 0; clock is rising edge; address is a constant; output q is read from current adress's data.
- **others**:not change.

ramplminit:

- **Read Mode**: wren is 0; clock is rising edge; address is a constant; output q is read from current adress's data. The data is initial in ".mif" file that show in follow figure.

- **others**: Not have any change.Hold the data.



Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	15	14	13	12	11	10	9	8	....
8	7	6	5	4	3	2	1	0	....
16	0	0	0	0	0	0	0	0	....

FIG. I.1 ramlpminit.mif

D. Demonstration strategy

Input

Load	Data_in[7..0]	clock	wren
KEY0	SW[7..0]	KEY[3]	SW[9]

Output

clock	wren	address[7..0]	data[7..0]	q[7..0]
LEDR9	LEDR8	HEX5,HEX4	HEX3,HEX2	HEX1,HEX0

E. Contribution of each team member

Jiaxin Zheng: Write code.

Jiaqing Wan: Do simulation and write report.

II. LAB DETAILS

A. Instructor questions in the lab instruction

1.The entity and architecture for the memory is in the file ramlpm.vhd. Open up the VHDL file (as TEXT) and take a look at what was just created for you. What are the input and output signals and how many bits are they?

Input		Output
address(8bit)	clock(1bit)	q(8bit)
data(8bit)	wren(1bit)	

2.What is the value of the generic parameter operation_mode?

A: SINGLE_PORT

3.What is the value of outdata_reg_a?

A: UNREGISTERED

4.What is the aspect ratio of this memory?

A: 256 x 8

B. VHDL code

ramlpm.vhd

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

LIBRARY altera_mf;
USE altera_mf.altera_mf_components.all;

ENTITY ramlpm IS
   PORT
   (
      address     : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
      clock    : IN STD_LOGIC  := '1';
      data     : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
      wren     : IN STD_LOGIC ;
      q     : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)
   );
END ramlpm;


ARCHITECTURE SYN OF ramlpm IS

   SIGNAL sub_wire0   : STD_LOGIC_VECTOR (7 DOWNTO 0);

BEGIN
   q    <= sub_wire0(7 DOWNTO 0);

   altsyncram_component : altsyncram
   GENERIC MAP (
      clock_enable_input_a => "BYPASS",
      clock_enable_output_a => "BYPASS",
      intended_device_family => "Cyclone_V",
      lpm_hint => "ENABLE_RUNTIME_MOD=NO",
      lpm_type => "altsyncram",
      numwords_a => 256,
      operation_mode => "SINGLE_PORT",
      outdata_aclr_a => "NONE",
      outdata_reg_a => "UNREGISTERED",
      power_up_uninitialized => "FALSE",
      ram_block_type => "M10K",
      read_during_write_mode_port_a => "NEW_DATA_NO_NBE_READ",
      widthad_a => 8,
      width_a => 8,
      width_byteena_a => 1
   )
   PORT MAP (
      address_a => address,
      clock0 => clock,
      data_a => data,
      wren_a => wren,
      q_a => sub_wire0
   );



END SYN;
```

ramlpminit

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

LIBRARY altera_mf;
USE altera_mf.altera_mf_components.all;

ENTITY ramlpminit IS
    PORT
    (
        address     : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
        data     : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
        clock     : IN STD_LOGIC  := '1';
        wren     : IN STD_LOGIC ;
        q     : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)
    );
END ramlpminit;


ARCHITECTURE SYN OF ramlpminit IS

    SIGNAL sub_wire0    : STD_LOGIC_VECTOR (7 DOWNTO 0);

BEGIN
    q    <= sub_wire0(7 DOWNTO 0);

    altsyncram_component : altsyncram
    GENERIC MAP (
        clock_enable_input_a => "BYPASS",
        clock_enable_output_a => "BYPASS",
        init_file => "ramlpminit.mif",
        intended_device_family => "Cyclone_V",
        lpm_hint => "ENABLE_RUNTIME_MOD=NO",
        lpm_type => "altsyncram",
        numwords_a => 256,
        operation_mode => "SINGLE_PORT",
        outdata_aclr_a => "NONE",
        outdata_reg_a => "UNREGISTERED",
        power_up_uninitialized => "FALSE",
        ram_block_type => "M10K",
        read_during_write_mode_port_a => "NEW_DATA_NO_NBE_READ",
        widthad_a => 8,
        width_a => 8,
        width_byteena_a => 1
    )
    PORT MAP (
        address_a => address,
        clock0 => clock,
        data_a => data,
        wren_a => wren,
        q_a => sub_wire0
    );



END SYN;
```

C. Block diagram

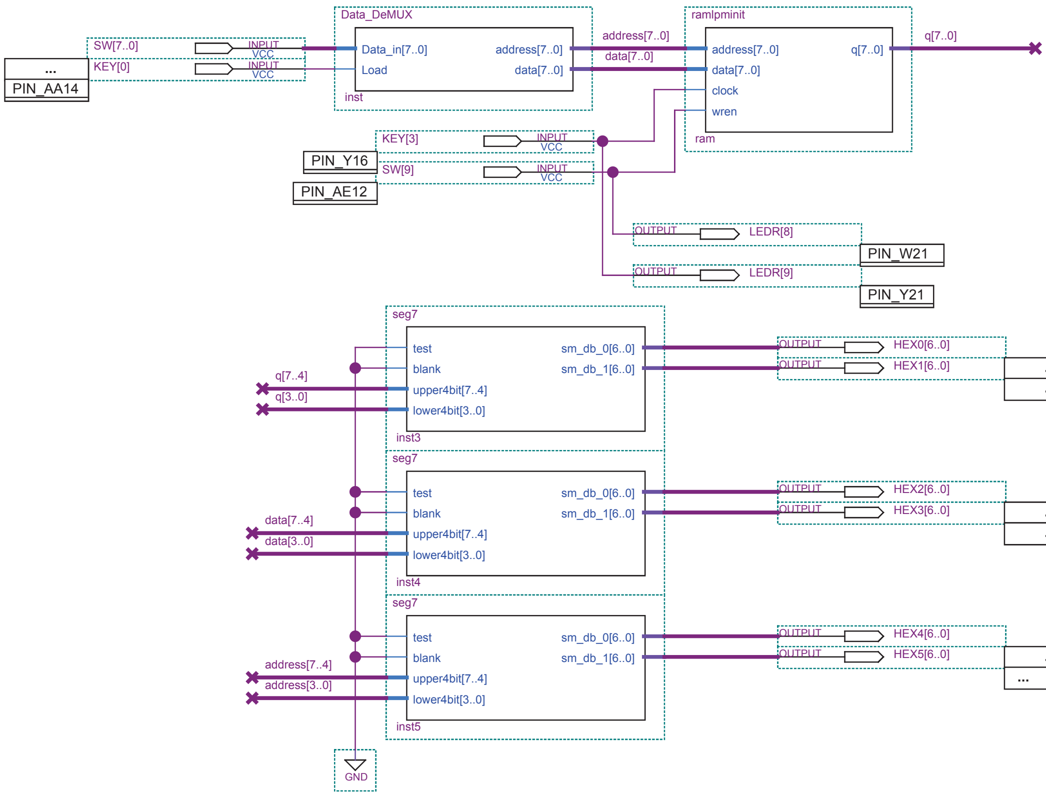


FIG. II.1 ram_tb Block diagram

D. ModelSim force file

ramlpm.txt

```
force clock 0 0ns,1 50ns -r 100ns
#write data to address
force wren 1
force data x"0F"
force address x"00"
run 200ns
force wren 1
force data x"0E"
force address x"01"
run 200ns
force wren 1
force data x"0D"
force address x"02"
run 200ns
force wren 1
force data  x"0C"
force address  x"03"
run 200ns
force wren  1
force data  x"0B"
force address  x"04"
run 200ns
force wren  1
force data  x"0A"
force address  x"05"
run 200ns
force wren  1
force data  x"09"
force address  x"06"
run 200ns
force wren  1
force data  x"08"
force address  x"07"
run 200ns
force wren  1
force data  x"07"
force address  x"08"
run 200ns
force wren  1
force data  x"06"
force address  x"09"
run 200ns
force wren  1
force data  x"05"
force address  x"0A"
run 200ns
force wren  1
force data  x"04"
force address  x"0B"
run 200ns
force wren  1
force data  x"03"
force address  x"0C"
run 200ns
force wren  1
force data  x"02"
force address  x"0D"
run 200ns
force wren  1
force data  x"01"
force address  x"0E"
run 200ns
force wren  1
force data  x"00"
force address  x"0F"
run 200ns


#read from address
force wren  0
force address  x"00"
run 200ns
force wren  0
force address  x"01"
```

```
run 200ns
force wren  0
force address  x"02"
run 200ns
force wren  0
force address  x"03"
run 200ns
force wren  0
force address  x"04"
run 200ns
force wren  0
force address  x"05"
run 200ns
force wren  0
force address  x"06"
run 200ns
force wren  0
force address  x"07"
run 200ns
force wren  0
force address  x"08"
run 200ns
force wren  0
force address  x"09"
run 200ns
force wren  0
force address  x"0A"
run 200ns
force wren  0
force address  x"0B"
run 200ns
force wren  0
force address  x"0C"
run 200ns
force wren  0
force address  x"0D"
run 200ns
force wren  0
force address  x"0E"
run 200ns
force wren  0
force address  x"0F"
run 200ns
```

ramlpminit.txt

```
#read data from the initial memory
force clock 0 50ns, 1 100ns -r 100ns
force wren  0

force address  x"00"
run 200ns
force address  x"01"
run 200ns
force address  x"02"
run 200ns
force address  x"03"
run 200ns
```

```
force address  x"04"
run 200ns
force address  x"05"
run 200ns
force address  x"06"
run 200ns
force address  x"07"
run 200ns
force address  x"08"
run 200ns
force address  x"09"
run 200ns
force address  x"0A"
run 200ns
force address  x"0B"
run 200ns
force address  x"0C"
run 200ns
force address  x"0D"
run 200ns
force address  x"0E"
run 200ns
force address  x"0F"
run 200ns
```

E. Simulation results and detailed explanation

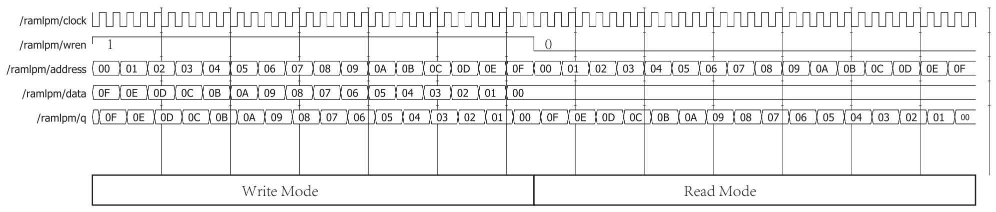


FIG. II.2 Wave

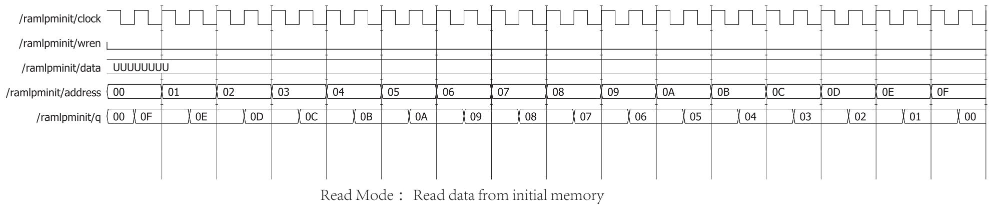


FIG. II.3 Wave

F. Board demonstration result

The board demonstration result is following as the table at section ”Simulation strategy”.
In others case the result is the same logic we need.

III. CONCLUSION

The lab is successful.
I realize all function we should achieve. The most difficulty things is the Instructor questions.
I refer to manual and ask classmates that I solved it.