第五次實驗報告

題目：

Substitute dual-port memory by single-port memory

姓名：黃子軒

學號：**310581003**

繳交日期：**111** 年 **11** 月 **21** 日

1. 實驗目的

透過本次實驗，除了熟悉如何撰寫 verilog 並實現於 FPGA 板，並首次學習如何使用 memory 的 module 功能。

2. 實驗程式碼

(1) 主要程式

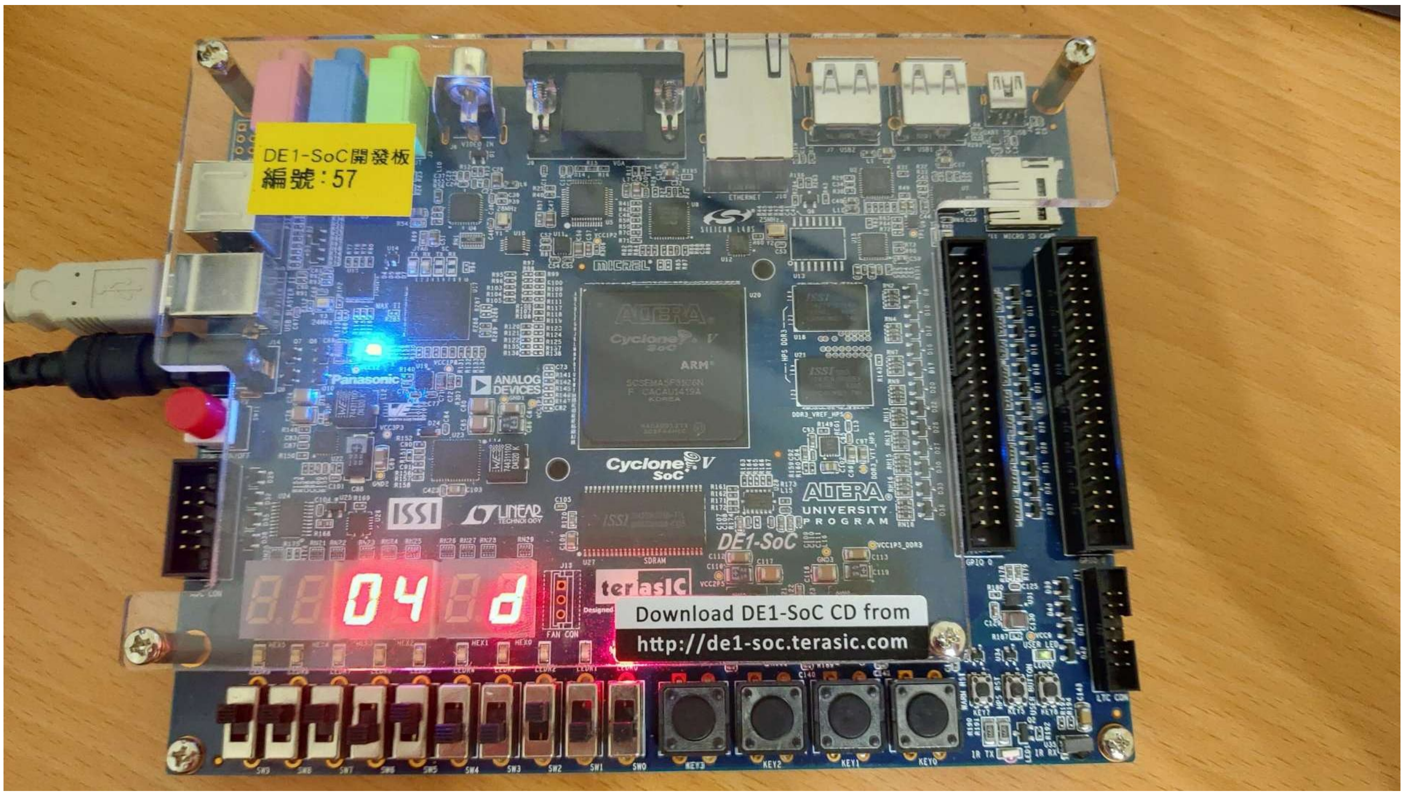
```
module Lab5_part5(SW, CLOCK_50, LEDR, HEX0, HEX2, HEX3);
  input [9:0] SW;
  input CLOCK_50;
  output [9:0] LEDR;
  output [6:0] HEX0, HEX2, HEX3;
  wire [3:0] readout;



  ramlpm    rm(
  .address(SW[4:0]),
  .clock(CLOCK_50),
  .data(SW[8:5]),
  .wren(SW[9]),
  .q(readout));

  assign LEDR[0] = SW[9];
  seven_segment s1(SW[4],HEX3);
  seven_segment s2(SW[3:0],HEX2);
  seven_segment s3(readout,HEX0);

endmodule

// Seven-segment desplay //
```

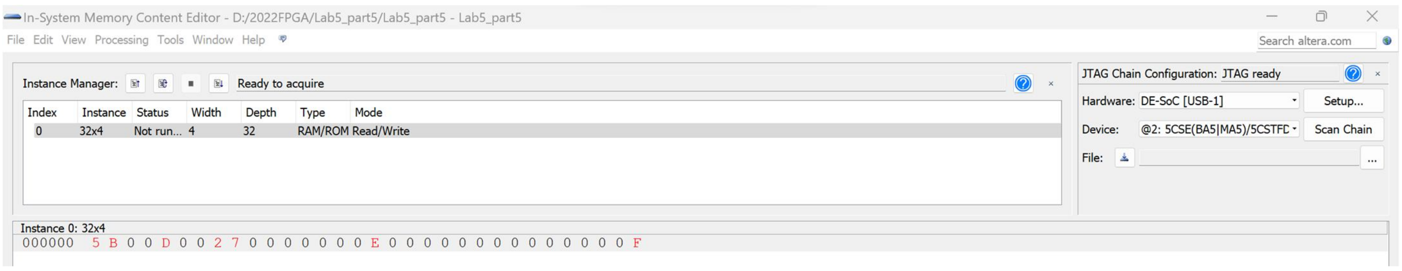
```
module seven_segment(in,HEX);

  input [3:0] in;

  output wire [6:0] HEX;



  assign HEX[0] =
~in[3]&in[2]&~in[0]|~in[3]&~in[2]&~in[1]&in[0]|in[3]&in[2]&~in[1]&in[0]|in[3]&~in[2]&in[1]&in[0]
;

  assign HEX[1] =
~in[3]&in[2]&~in[1]&in[0]|in[3]&in[2]&in[1]|in[3]&~in[0]&in[2]|in[2]&in[1]&~in[0]|in[3]&in[1]&in[
0];

  assign HEX[2] = in[3]&in[2]&~in[0]|in[3]&in[2]&in[1]|~in[3]&~in[2]&in[1]&~in[0];

  assign HEX[3] =
in[3]&~in[2]&in[1]&~in[0]|~in[3]&in[2]&~in[1]&~in[0]|~in[2]&~in[1]&in[0]|in[2]&in[1]&in[0];

  assign HEX[4] = ~in[2]&~in[1]&in[0]|~in[3]&in[0]|~in[3]&in[2]&~in[1];

  assign HEX[5] =
~in[3]&~in[2]&in[0]|~in[3]&~in[2]&in[1]|~in[3]&in[1]&in[0]|in[3]&in[2]&~in[1]&in[0];

  assign HEX[6] = ~in[3]&in[2]&in[1]&in[0]|in[3]&in[2]&~in[1]&~in[0]|~in[3]&~in[2]&~in[1];

endmodule
```

(2) RAM

```
// megafunction wizard: %RAM: 1-PORT%

// GENERATION: STANDARD

// VERSION: WM1.0

// MODULE: altsyncram



// ============================================================

// File Name: ramlpm.v

// Megafunction Name(s):

//                      altsyncram

//

// Simulation Library Files(s):

//                      altera_mf
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 13.1.0 Build 162 10/23/2013 SJ Full Version
// ************************************************************


//Copyright (C) 1991-2013 Altera Corporation
//Your use of Altera Corporation's design tools, logic functions
//and other software and tools, and its AMPP partner logic
//functions, and any output files from any of the foregoing
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Altera Program License
//Subscription Agreement, Altera MegaCore Function License
//Agreement, or other applicable license agreement, including,
//without limitation, that your use is for the sole purpose of
//programming logic devices manufactured by Altera and sold by
//Altera or its authorized distributors.  Please refer to the
//applicable agreement for further details.


// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module ramlpm (
  address,
  clock,
  data,
  wren,
  q);

  input   [4:0]  address;
  input          clock;
  input   [3:0]  data;
  input          wren;
  output  [3:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
  tri1    clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

  wire [3:0] sub_wire0;
  wire [3:0] q = sub_wire0[3:0];

  altsyncram    altsyncram_component (
                .address_a (address),
                .clock0 (clock),
                .data_a (data),
                .wren_a (wren),
                .q_a (sub_wire0),
                .aclr0 (1'b0),
                .aclr1 (1'b0),

                .address_b (1'b1),

                .addressstall_a (1'b0),

                .addressstall_b (1'b0),

                .byteena_a (1'b1),

                .byteena_b (1'b1),

                .clock1 (1'b1),

                .clocken0 (1'b1),

                .clocken1 (1'b1),

                .clocken2 (1'b1),

                .clocken3 (1'b1),

                .data_b (1'b1),

                .eccstatus (),

                .q_b (),

                .rden_a (1'b1),

                .rden_b (1'b1),

                .wren_b (1'b0));

  defparam

        altsyncram_component.clock_enable_input_a = "BYPASS",

        altsyncram_component.clock_enable_output_a = "BYPASS",

        altsyncram_component.init_file = "ramlpm.mif",

        altsyncram_component.intended_device_family = "Cyclone V",

        altsyncram_component.lpm_hint =
"ENABLE_RUNTIME_MOD=YES,INSTANCE_NAME=32x4",

        altsyncram_component.lpm_type = "altsyncram",

        altsyncram_component.numwords_a = 32,

        altsyncram_component.operation_mode = "SINGLE_PORT",

        altsyncram_component.outdata_aclr_a = "NONE",
        altsyncram_component.outdata_reg_a = "UNREGISTERED",
        altsyncram_component.power_up_uninitialized = "FALSE",
        altsyncram_component.ram_block_type = "M10K",
        altsyncram_component.read_during_write_mode_port_a = "NEW_DATA_NO_NBE_READ",
        altsyncram_component.widthad_a = 5,
        altsyncram_component.width_a = 4,
        altsyncram_component.width_byteena_a = 1;


endmodule

// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrData NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: DataBusSeparated NUMERIC "1"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"

// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"

// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "1"

// Retrieval info: PRIVATE: JTAG_ID STRING "32x4"

// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"

// Retrieval info: PRIVATE: MIFfilename STRING "ramlpm.mif"

// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "32"

// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "2"

// Retrieval info: PRIVATE: READ_DURING_WRITE_MODE_PORT_A NUMERIC "3"

// Retrieval info: PRIVATE: RegAddr NUMERIC "1"

// Retrieval info: PRIVATE: RegData NUMERIC "1"

// Retrieval info: PRIVATE: RegOutput NUMERIC "0"

// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"

// Retrieval info: PRIVATE: SingleClock NUMERIC "1"

// Retrieval info: PRIVATE: UseDQRAM NUMERIC "1"

// Retrieval info: PRIVATE: WRCONTROL_ACLR_A NUMERIC "0"

// Retrieval info: PRIVATE: WidthAddr NUMERIC "5"

// Retrieval info: PRIVATE: WidthData NUMERIC "4"

// Retrieval info: PRIVATE: rden NUMERIC "0"

// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all

// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"

// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"

// Retrieval info: CONSTANT: INIT_FILE STRING "ramlpm.mif"

// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"

// Retrieval info: CONSTANT: LPM_HINT STRING
"ENABLE_RUNTIME_MOD=YES,INSTANCE_NAME=32x4"

// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"

// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "32"
// Retrieval info: CONSTANT: OPERATION_MODE STRING "SINGLE_PORT"

// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"

// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"

// Retrieval info: CONSTANT: POWER_UP_UNINITIALIZED STRING "FALSE"

// Retrieval info: CONSTANT: RAM_BLOCK_TYPE STRING "M10K"

// Retrieval info: CONSTANT: READ_DURING_WRITE_MODE_PORT_A STRING
"NEW_DATA_NO_NBE_READ"

// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "5"

// Retrieval info: CONSTANT: WIDTH_A NUMERIC "4"

// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"

// Retrieval info: USED_PORT: address 0 0 5 0 INPUT NODEFVAL "address[4..0]"

// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"

// Retrieval info: USED_PORT: data 0 0 4 0 INPUT NODEFVAL "data[3..0]"

// Retrieval info: USED_PORT: q 0 0 4 0 OUTPUT NODEFVAL "q[3..0]"

// Retrieval info: USED_PORT: wren 0 0 0 0 INPUT NODEFVAL "wren"

// Retrieval info: CONNECT: @address_a 0 0 5 0 address 0 0 5 0

// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0

// Retrieval info: CONNECT: @data_a 0 0 4 0 data 0 0 4 0

// Retrieval info: CONNECT: @wren_a 0 0 0 0 wren 0 0 0 0

// Retrieval info: CONNECT: q 0 0 4 0 @q_a 0 0 4 0

// Retrieval info: GEN_FILE: TYPE_NORMAL ramlpm.v TRUE

// Retrieval info: GEN_FILE: TYPE_NORMAL ramlpm.inc FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL ramlpm.cmp FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL ramlpm.bsf FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL ramlpm_inst.v FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL ramlpm_bb.v TRUE

// Retrieval info: LIB_FILE: altera_mf
```

3. 實驗結果照片 (optional)




In-System Memory Content Editor - D:/2022FPGA/Lab5_part5/Lab5_part5 - Lab5_part5

Instance Manager: Ready to acquire

Index	Instance	Status	Width	Depth	Type	Mode
0	32x4	Not run...	4	32	RAM/ROM	Read/Write

JTAG Chain Configuration: JTAG ready

Hardware: DE-SoC [USB-1]

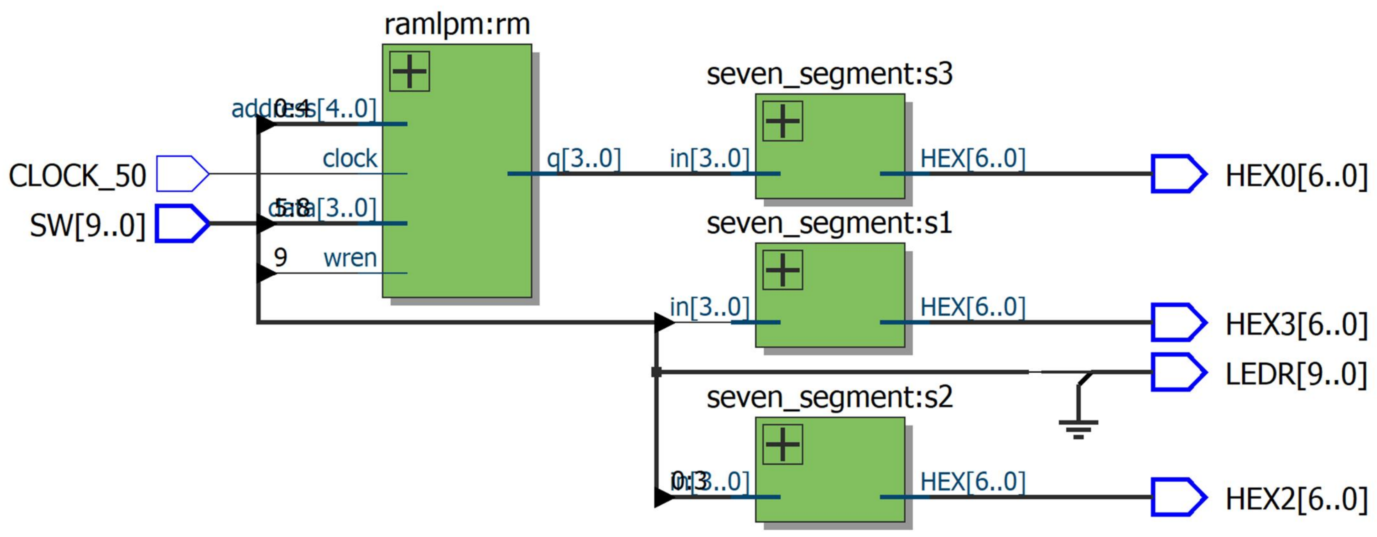
Device: @2: 5CSE(BA5|MA5)/5CSTFD

File:

Instance 0: 32x4

```
000000  5 B 0 0 D 0 0 2 7 0 0 0 0 0 0 E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 F
```

4. RTL 佈局(optional)



5.問題與討論

經過本次實驗，對 memory 在 FPGA 板上的使用方法更加熟悉，亦對 memory 的內部運作有更深的了解！