浙江大学

本科实验报告

课程名称：计算机组成

姓　　名：陈程

学　　院：计算机科学与技术学院

专　　业：计算机科学与技术

学　　号：3180103537

生活照：

指导教师：张明敏

2020 年 3 月 19 日

Experiment 3—SOC design of IP core integration

Name: 陈程 **Student ID**: 3180103537 **Major**: Computer Science and Technology

Course: Computer Organization **Group mate**:

Time: 2020-3-19 **Laboratory**: East4-509 **Instructor**: 洪奇军

I. Purposes And Requirements Of The Experiment

1.1 Preliminary understanding of GPIO interface and equipment

1.2 Understand the basic structure of computer system

1.3 Understand the relationship between computer components

1.4 Understand and master the use of IP core

1.5 Understand SoC system and implement simple SoC system with IP core

II. The Content And The Principle Of The Experiment

2.1 Experimental content

2.1.1 Analyze basic and interface IP core

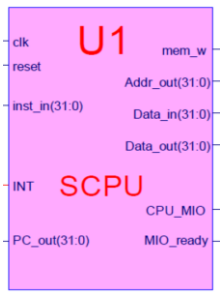
2.1.2 Design memory IP module

2.1.3 Practice how to use IP core

2.1.4 Select the third-party IP core to integrate with the existing modules to realize SOC (The top level of this experiment is realized by schematic design)

2.2 Experimental principles

2.2.1 U1-cpu module: SCPU



- MIPS framework
  - RISC architecture

- Three instruction types

- Implement basic instructions
  - The design implementation shall not be less than the following instructions:
    - R-Type： add, sub, and, or, xor, nor, slt, srl*, jr, jalr, eret*;
    - Type：addi, andi, ori, xori, lui, lw, sw, beq, bne, slti;
    - J-Type：J, Jal*;

- IP Core- U1 for this experiment
  - Core call module scpu.ngc
  - Nuclear interface signal module (empty document): scpu. V
  - Core module symbol document: scpu.sym

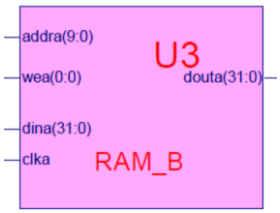
- SCPU.v

```
21  module SCPU(    input clk,          //
22                  input reset,
23                  input MIO_ready,
24
25                  input [31:0]inst_in,
26                  input [31:0]Data_in,
27
28                  output mem_w,
29                  output[31:0]PC_out,
30                  output[31:0]Addr_out,
31                  output[31:0]Data_out,
32                  output CPU_MIO,
33                  input INT
34              );
35
36
37  endmodule
```

2.2.2 U2 instruction code enclosure: ROM_B

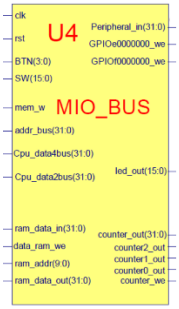
- ROM_D/B
  - Module designed by experiment one
  - FPGA internal memory
    - Block memory generator or distributed memory generator
  - Capacity
    - 1024 x 32bit
  - Core module symbol document: Rom? B.sym

2.2.3 U3 data storage module: RAM_B



- RAM_B
  - FPGA internal memory
    - Block Memory Generator
  - Capacity
    - 1024 x 32bit
  - Core module symbol document: ram_b.sym

2.2.4 U4 bus interface module: MIO_BUS



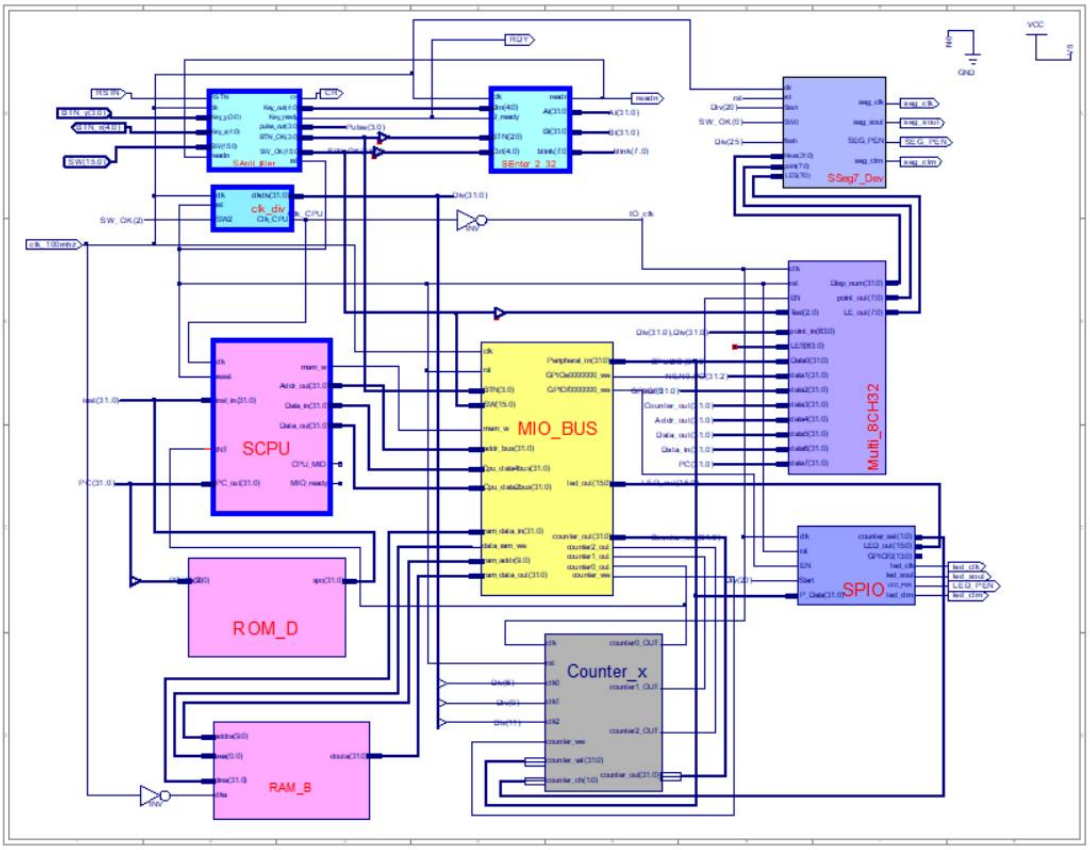
- MIO_BUS
  - CPU and external data exchange interface module
  - This course combines data exchange circuit into one module

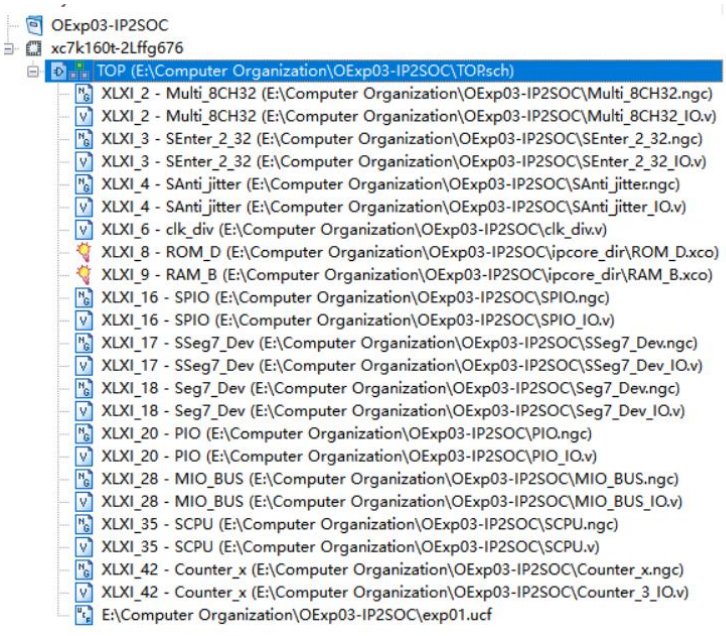
- Basic function
  - Data storage, 7-seg, SW, BTN and led interfaces
- IP soft core - U4 for this experiment
  - Core calling module MIO_BUS.ngc
  - Nuclear interface signal module (empty document): MIO_BUS_IO.v
  - Core module symbol document: MIO_BUS.sym

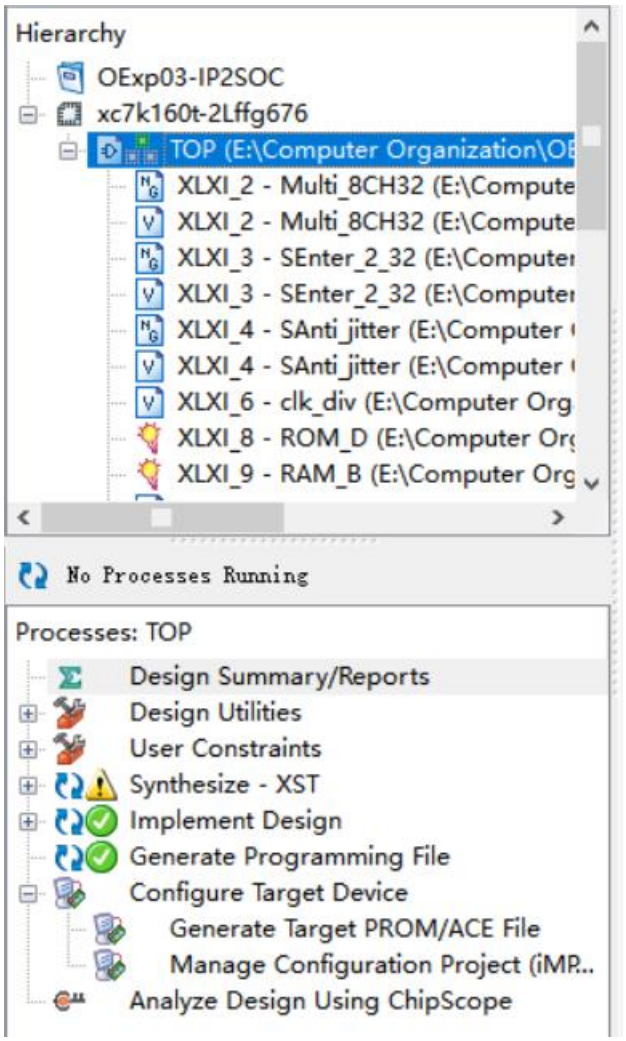
- MIO_BUS.v

```
21  module MIO_BUS(input clk,
22                 input rst,
23                 input[3:0]BTN,
24                 input[15:0]SW,
25                 input mem_w,
26                 input[31:0]Cpu_data2bus,            //data from CPU
27                 input[31:0]addr_bus,
28                 input[31:0]ram_data_out,
29                 input[15:0]led_out,
30                 input[31:0]counter_out,
31                 input counter0_out,
32                 input counter1_out,
33                 input counter2_out,
34
35                 output reg[31:0]Cpu_data4bus,          //write to CPU
36                 output reg[31:0]ram_data_in,           //from CPU write to Memory
37                 output reg[9:0]ram_addr,                  //Memory Address signals
38                 output reg data_ram_we,
39                 output reg GPIOf0000000_we,
40                 output reg GPIOe0000000_we,
41                 output reg counter_we,
42                 output reg[31:0]Peripheral_in
43                 );
44
45  endmodule
```

2.2.5 Top module







2.2.6 UCF

```
NET "clk_100mhz"              LOC = AC18      | IOSTANDARD = LVCMOS18 ;
NET "RSTN"                    LOC = W13       | IOSTANDARD = LVCMOS18 ;
NET "clk_100mhz"              TNM_NET = TM_CLK ;
TIMESPEC TS_CLK_100M       = PERIOD "TM_CLK"       10 ns HIGH 50%;

NET "led_clk"           LOC = N26       | IOSTANDARD = LVCMOS33 ;
NET "led_clrn"          LOC = N24       | IOSTANDARD = LVCMOS33 ;
NET "led_sout"          LOC = M26       | IOSTANDARD = LVCMOS33 ;
NET "LED_PEN"           LOC = P18       | IOSTANDARD = LVCMOS33 ;

NET "RDY"         LOC = U21       | IOSTANDARD = LVCMOS33 ;
NET "readn"       LOC = U22       | IOSTANDARD = LVCMOS33 ;
NET "CR"          LOC = V22       | IOSTANDARD = LVCMOS33 ;


NET "BTN_x[0]"          LOC = V17       | IOSTANDARD = LVCMOS18 ;
NET "BTN_x[1]"          LOC = W18       | IOSTANDARD = LVCMOS18 ;
NET "BTN_x[2]"          LOC = W19       | IOSTANDARD = LVCMOS18 ;
NET "BTN_x[3]"          LOC = W15       | IOSTANDARD = LVCMOS18 ;
NET "BTN_x[4]"          LOC = W16       | IOSTANDARD = LVCMOS18 ;



NET "BTN_y[0]"          LOC = V18       | IOSTANDARD = LVCMOS18 ;
NET "BTN_y[1]"          LOC = V19       | IOSTANDARD = LVCMOS18 ;
NET "BTN_y[2]"          LOC = V14       | IOSTANDARD = LVCMOS18 ;
NET "BTN_y[3]"          LOC = W14       | IOSTANDARD = LVCMOS18 ;
```

```
28
29  NET "SW[0]"         LOC = AA10      | IOSTANDARD = LVCMOS15 ;
30  NET "SW[1]"         LOC = AB10      | IOSTANDARD = LVCMOS15 ;
31  NET "SW[2]"         LOC = AA13      | IOSTANDARD = LVCMOS15 ;
32  NET "SW[3]"         LOC = AA12      | IOSTANDARD = LVCMOS15 ;
33  NET "SW[4]"         LOC = Y13       | IOSTANDARD = LVCMOS15 ;
34  NET "SW[5]"         LOC = Y12       | IOSTANDARD = LVCMOS15 ;
35  NET "SW[6]"         LOC = AD11      | IOSTANDARD = LVCMOS15 ;
36  NET "SW[7]"         LOC = AD10      | IOSTANDARD = LVCMOS15 ;
37  NET "SW[8]"         LOC = AE10      | IOSTANDARD = LVCMOS15 ;
38  NET "SW[9]"         LOC = AE12      | IOSTANDARD = LVCMOS15 ;
39  NET "SW[10]"        LOC = AF12      | IOSTANDARD = LVCMOS15 ;
40  NET "SW[11]"        LOC = AE8       | IOSTANDARD = LVCMOS15 ;
41  NET "SW[12]"        LOC = AF8       | IOSTANDARD = LVCMOS15 ;
42  NET "SW[13]"        LOC = AE13      | IOSTANDARD = LVCMOS15 ;
43  NET "SW[14]"        LOC = AF13      | IOSTANDARD = LVCMOS15 ;
44  NET "SW[15]"        LOC = AF10      | IOSTANDARD = LVCMOS15 ;
45
46
```

III. Main instrument and equipment

■Computer system (Intel Core I3 or above, 1GB memory or above) 1

■Sword development board 1

■Xilinx ise12.4 and above development tools 1

IV. Experimental Results and Analysis

V. Discussion and Experience

In this experiment, I basically realized the established goals of preliminary understanding of GPIO interface and equipment, understanding the basic structure of computer system, understanding the relationship between various components of computer, understanding and mastering the use method of IP core, understanding SOC system and using IP core to realize simple SOC system.

At the same time, I further understand that SOC design is designed around the bus. In understanding the bus, the core point is to understand the master-slave communication mechanism. As the host, the processor is the party that initiates the communication. As a slave, peripheral and on-chip memory

respond to the communication initiated by the host. On this basis, we need to understand the timing of the bus and the whole process from initiating communication on the bus to the end of communication. In this way, we can understand how to access a specific address space through software.