

Lab04-0

集成CPU核

-IP核设计CPU/IP2CPU

Course Outline

- 一、实验目的
- 二、实验环境
- 三、实验目标及任务

实验目的

1. 复习寄存器传输控制技术
2. 掌握CPU的核心组成：数据通路与控制器
3. 设计数据通路的功能部件
4. 进一步了解计算机系统的基本结构
5. 熟练掌握IP核的使用方法

实验环境

□ 实验设备

1. 计算机（Intel Core i5以上，4GB内存以上）系统
2. NEXYS A7开发板
3. Xilinx VIVADO2017.4及以上开发工具

□ 材料

无

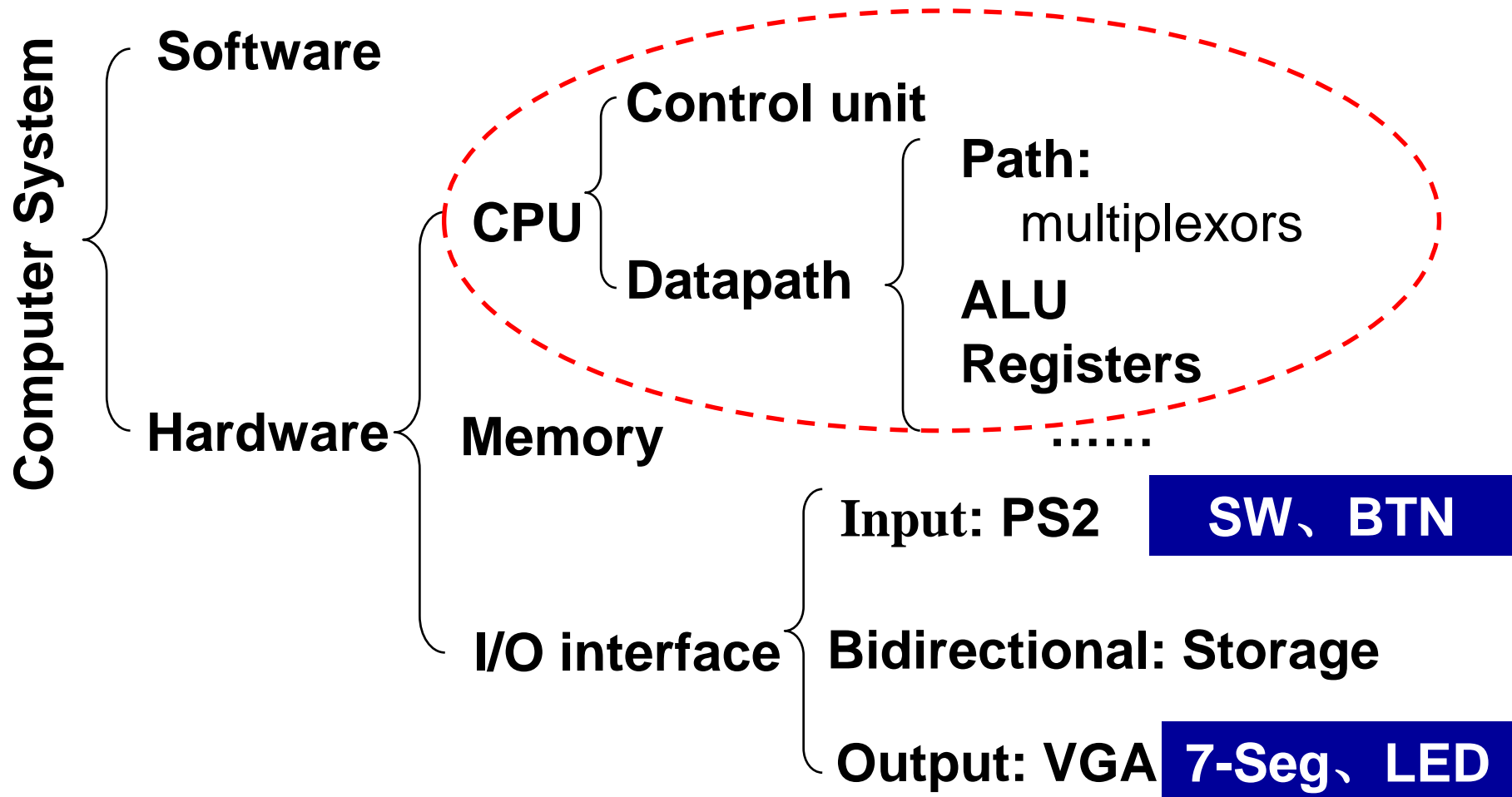
实验目标及任务

- **目标**：熟悉SOC系统的原理，掌握IP核集成设计CPU的方法
- **任务**：利用数据通路和控制器两个IP核集成设计CPU（根据原理图采用RTL代码方式）

-
- 任务： 利用数据通路和控制器两个IP核集成设计CPU

Computer Organization

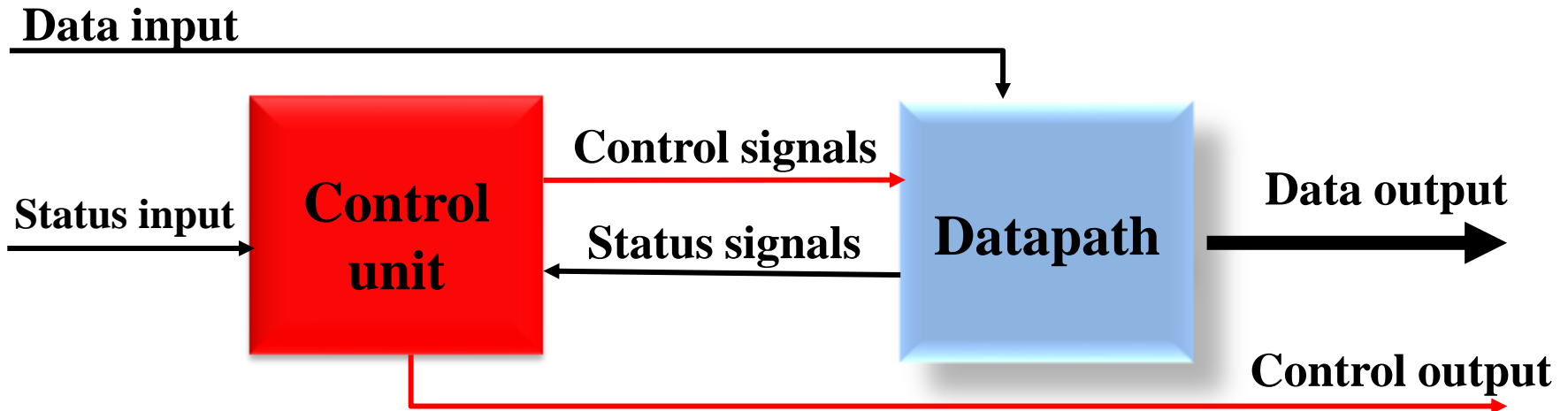
□ Decomposability of computer systems



Digital circuits vs CPU organization

□ Digital circuit

- General circuits that controls logical event with logical gates -
-Hardware



□ Computer organization

- Special circuits that processes logical action with instructions
-Software

CPU部件之1-数据通路: **DataPath**

□ Data_path

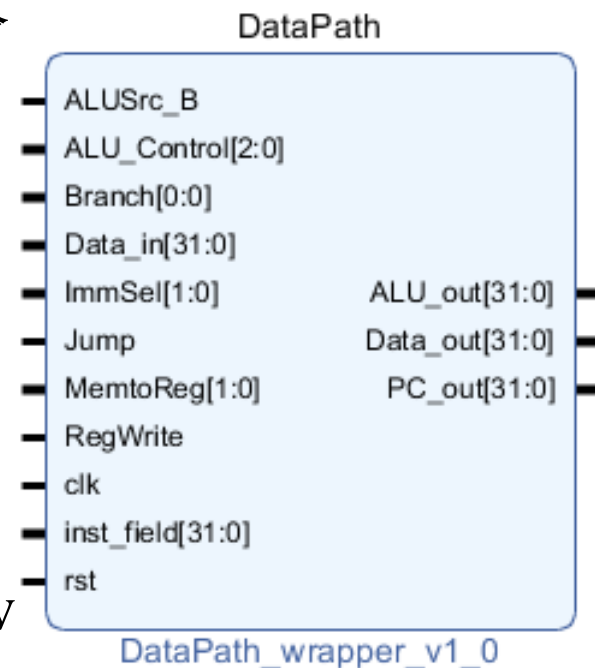
- CPU主要部件之一
- 寄存器传输控制对象: 通用数据通路

□ 基本功能

- 具有通用计算功能的算术逻辑部件
- 具有通用目的寄存器
- 具有通用计数所需的尽可能的路径

□ 本实验用IP 软核- **DataPath**

- 核调用模块DataPath.v
- 核接口信号模块(空文档): DataPath.v



数据通路空模块- **DataPath.v**

```
module      DataPath( input clk,                //时钟
                      input rst,                //复位
                      input[31:0]inst_field,    //指令数据域[31:7]
                      input ALUSrc_B,
                      input [1:0]MemtoReg,
                      input [1:0] ImmSel,
                      input Jump,
                      input Branch,
                      input RegWrite,
                      input[31:0]Data_in,
                      input[2:0]ALU_Control,

                      output[31:0]ALU_out,
                      output[31:0]Data_out,
                      output[31:0]PC_out
                      );

endmodule
```

CPU部件之2-控制器： **SCPU_ctrl**

□ SCPU_ctrl

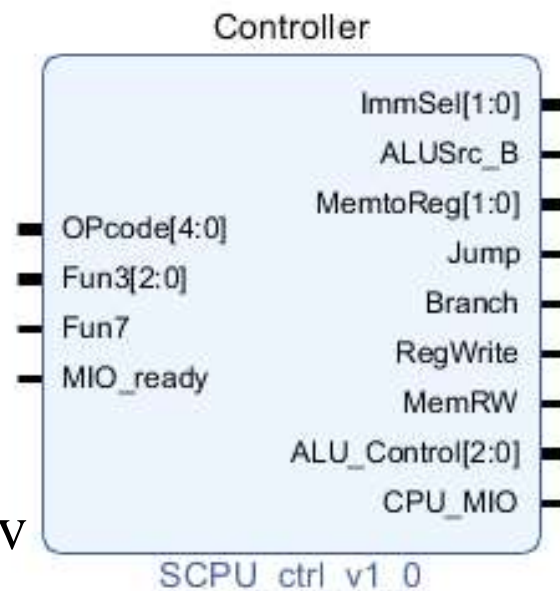
- CPU主要部件之一
- 寄存器传输控制技术中的运算和通路控制器：

□ 基本功能

- 指令译码
- 产生操作控制信号：ALU运算控制
- 产生指令所需的路径选择

□ 本实验用IP 软核- **SCPU_ctrl**

- 核调用模块SCPU_ctrl.v
- 核接口信号模块(空文档)：SCPU_ctrl.v



控制器接口文档- **SCPU_ctrl.v**

```
module      SCPU_ctrl(  
    input[4:0]OPcode,        //Opcode—inst[6:2]  
    input[2:0]Fun3,          //Function-inst[14:12]  
    input      Fun7,          //Function-inst[30]  
    input MIO_ready,          //CPU Wait  
    output reg [1:0] ImmSel    //立即数选择  
    output reg ALUSrc_B,  
    output reg [1:0] MemtoReg,  
    output reg Jump,  
    output reg Branch,  
    output reg RegWrite,  
    output reg MemRW,  
    output reg [2:0]ALU_Control,  
    output reg CPU_MIO  
);  
  
endmodule
```

RTL集成设计实现CPU

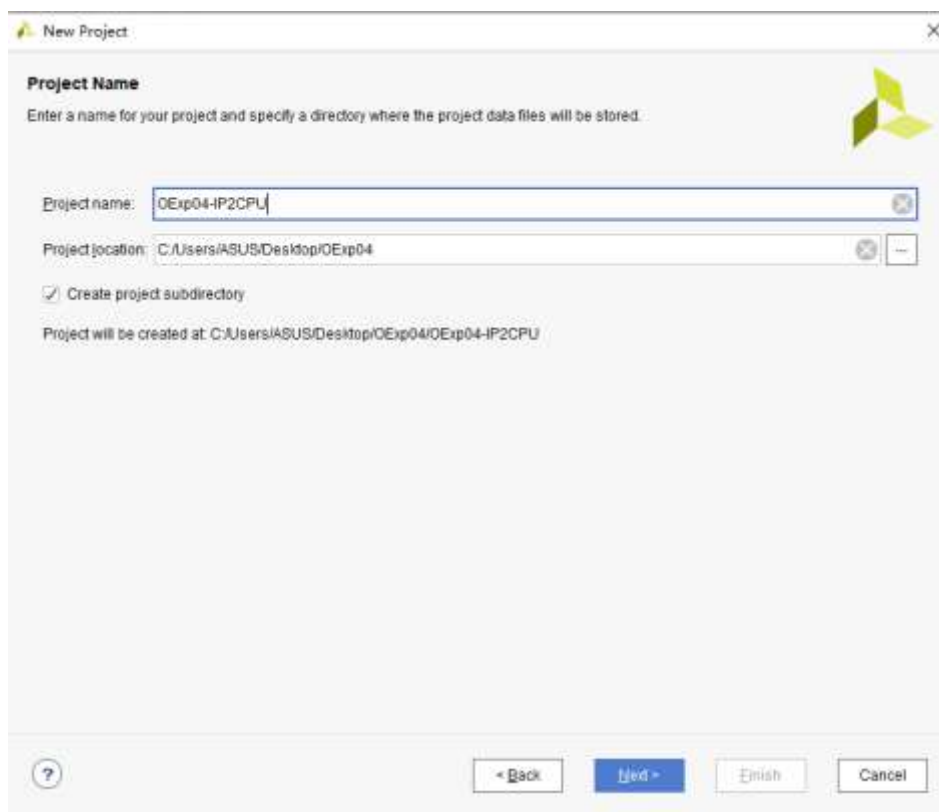
设计工程：OExp04-IP2CPU

◎ 分解CPU为二个IP核

④ 用二个IP核构建CPU


④ 顶层工程名称沿用OExp04-IP2CPU


◎ 顶层模块名：SCPU.v





设计工程：OExp04-IP2CPU

1. 将提供的IP目录放置在新创建的工程目录下
2. 在工程中直接导入IP目录下的.v和.edf文件

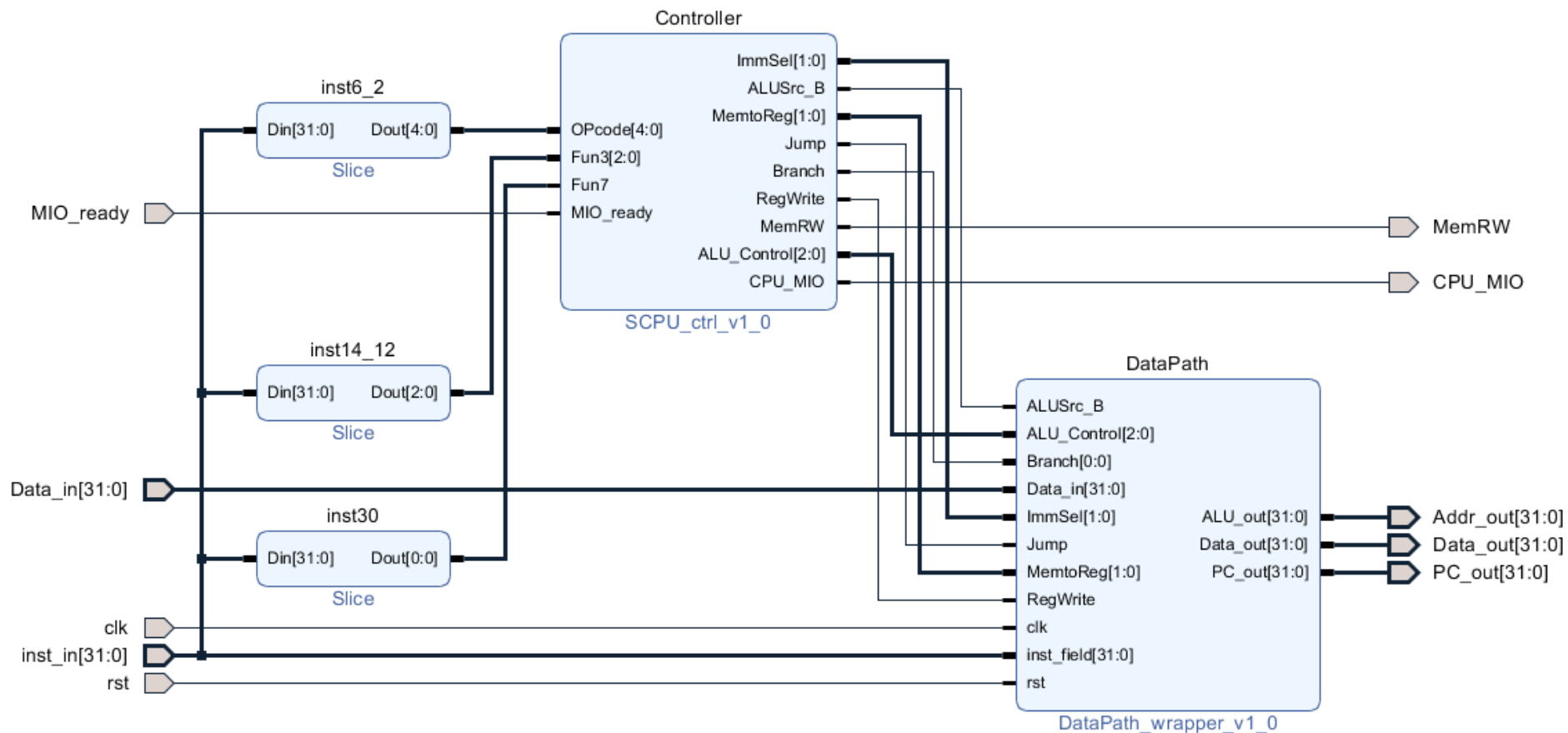
 DataPath.edf

 DataPath.v

 SCPU_ctrl.edf

 SCPU_ctrl.v

逻辑原理图



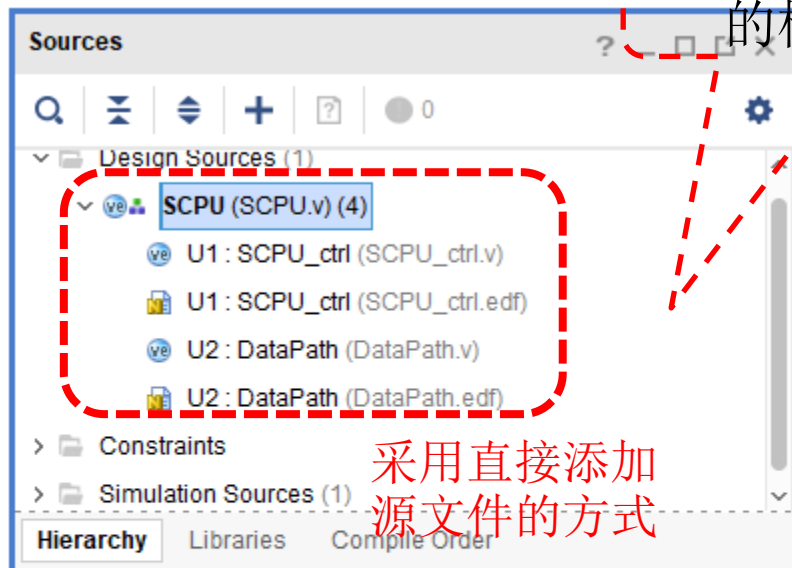
用RTL代码描述CPU设计 SCPU.V

- 利用Verilog描述，在SCPU.v顶层进行模块调用和连接

```
.....  
.....  
module SCPU(  
    input wire clk,  
    input wire rst,  
    input wire MIO_ready,  
    input wire[31:0] inst_in,  
    input wire[31:0] Data_in,  
  
    output wire CPU_MIO,  
    output wire MemRW,  
    output wire[31:0] PC_out,  
    output wire[31:0] Data_out,  
    output wire[31:0] Addr_out  
);  
  
    .....  
    .....  
  
    SCPU_ctrl1 U1(  
        .OPcode      (inst_in[6:2] ),  
        .Fun3        (inst_in[14:12]),  
        .Fun7        (inst_in[30] ),  
        .MIO_ready   (MIO_ready ),  
        .ImmSel      (ImmSel      ),  
        .ALUSrc_B    (ALUSrc_B    ),  
        .MemtoReg    (MemtoReg    ),  
        .Jump        (Jump        ),  
        .Branch      (Branch      ),  
        .RegWrite    (RegWrite    ),  
        .MemRW       (MemRW       ),  
        .ALU_Control (ALU_Control ),  
        .CPU_MIO     (CPU_MIO     )  
    )  
  
    .....  
    .....
```

CPU集成设计后的层次关系

Exp04完成CPU设计后的
模块调用关系



```
C:/Users/ASUS/Desktop/teacher/OExp04-IP2CPU/OExp04-IP2CPU.srscs/sources_1

40  wire      Branch;
41  wire      RegWrite;
42  wire [2:0] ALU_Control;
43
44  SCPU_ctrl U1(
45      .OPcode      (inst_in[6:2] ),
46      .Fun3         (inst_in[14:12]),
47      .Fun7         (inst_in[30] ),
48      .MIO_ready    (MIO_ready ),
49      .ImmSel1      (ImmSel1 ),
50      .ALUSrc_B     (ALUSrc_B ),
51      .MemtoReg     (MemtoReg ),
```

 **END**