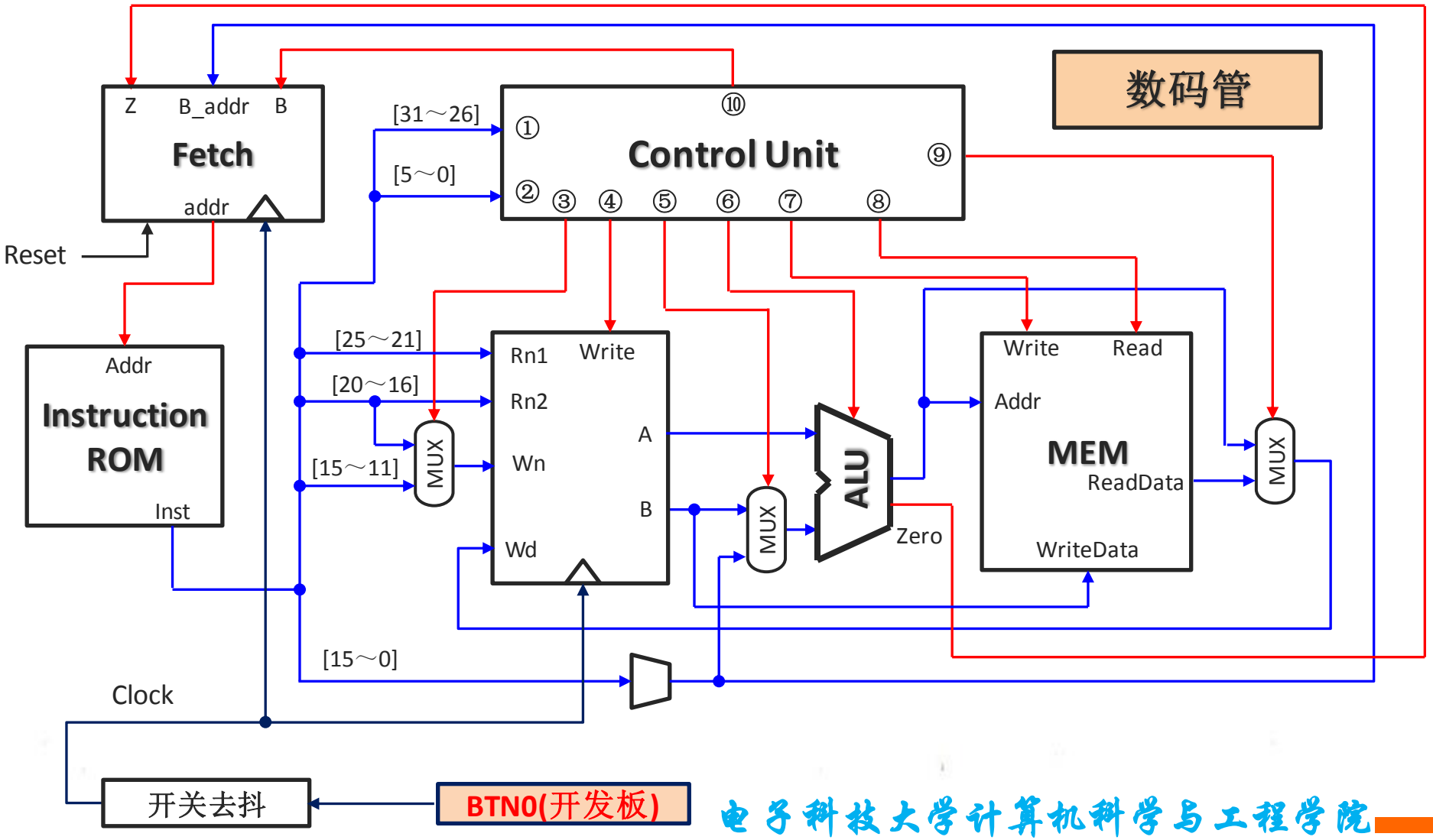
单周期CPU的设计

实　验

电子科技大学计算机科学与工程学院

实验总体要求：单周期CPU总体电路



实验总体要求：实现的指令

指令	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]	功能
add	000000	rs	rt	rd	00000	100000	寄存器加
sub	000000	rs	rt	rd	00000	100010	寄存器减
and	000000	rs	rt	rd	00000	100100	寄存器与
or	000000	rs	rt	rd	00000	100101	寄存器或
xor	000000	rs	rt	rd	00000	100110	寄存器异或

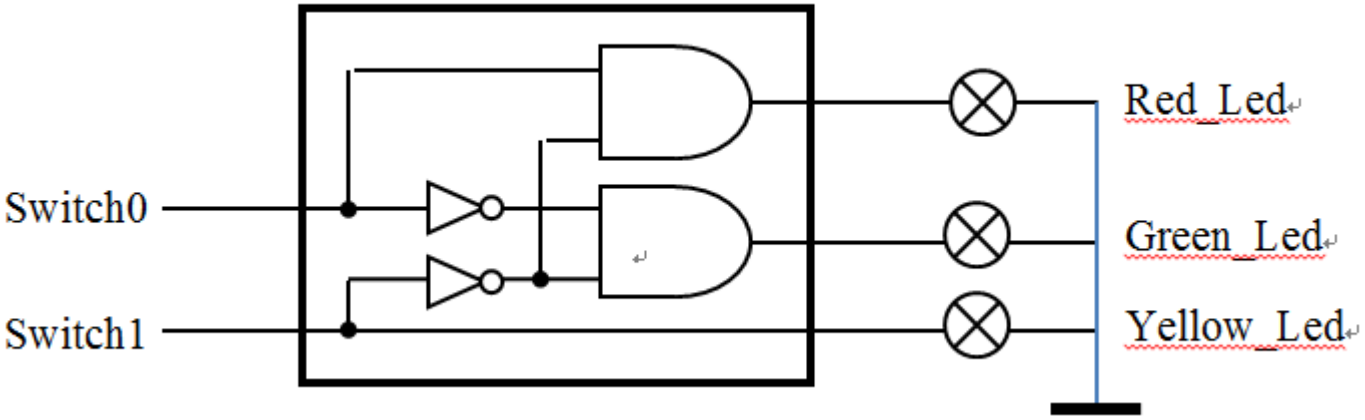
指令	[31:26]	[25:21]	[20:16]	[15:0]	功能
lw	100011	rs	rt	immediate	取字数据
sw	101011	rs	rt	immediate	存字数据
beq	000100	rs	rt	immediate	相等转移
lui	001111	00000	rt	immediate	设置高位

实验一：主要内容

1. 掌握用Verlog设计硬件电路的基本方法；
2. 开发板的基本使用；
3. 基本器件的设计；
   - 32位2选1多路选择器
   - 5位2选1多路选择器
   - 32位寄存器堆
   - ALU的设计
   - 符号扩展器的设计

Verilog设计硬件电路的基本方法

设计如下硬件电路：



Switch1	Switch0	Red_Led	Green_Led	Yellow_Led
0	0	灭	亮	灭
0	1	亮	灭	灭
1	X	灭	灭	亮

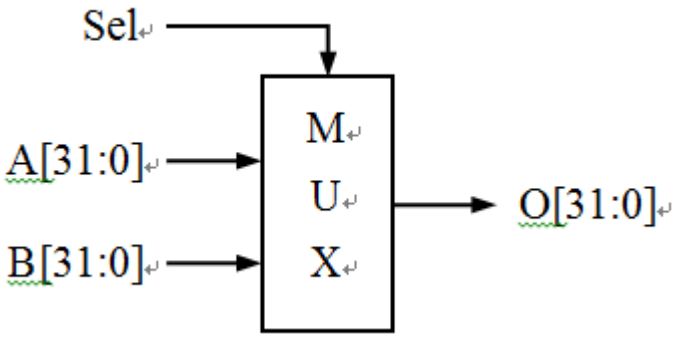
Verilog设计硬件电路的基本方法

1. 创建工程(Project)
2. 模块(Module)的创建
   - 输入代码
   - 语法检查（Synthesize – XST→Check Syntax）
   - 综合（Synthesize）
   - 仿真（Simulation）
3. 创建约束文件（UCF）

Verilog设计硬件电路的基本方法

4. 创建流代码（生成bit文件）
5. 下载到开发板进行板级验证

基本器件的设计：32位2选1多路选择器



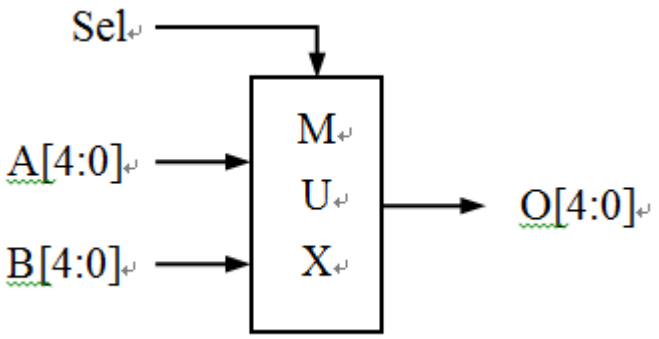
Function:

Sel = 0： O = A

Sel = 1： O = B

```
module MUX32_2_1(
    input [31:0] A , B ,
    input Sel ,
    output [31:0] O
    );

    assign O = Sel ? B : A;

endmodule
```

基本器件的设计：5位2选1多路选择器



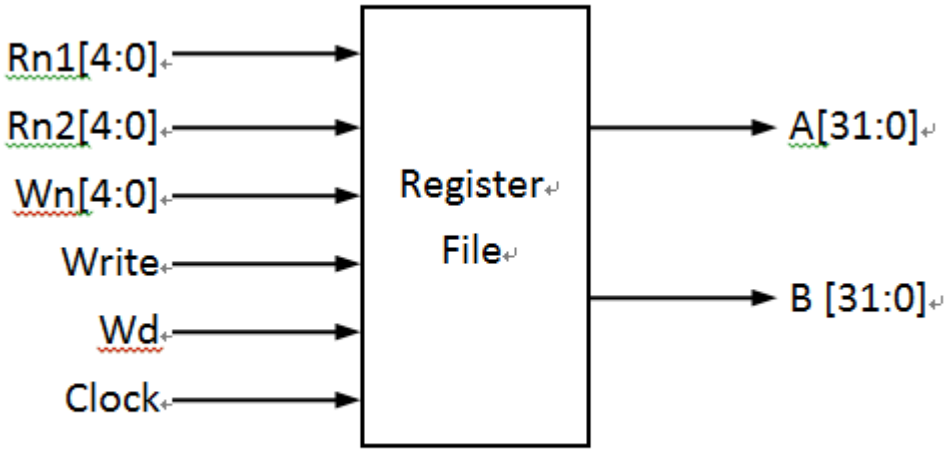
Function:

Sel = 0： O = A

Sel = 1： O = B

```
module MUX5_2_1(
    input [4:0] A , B ,
    input Sel ,
    output [4:0] O
    );

    assign O = Sel ? B : A;

endmodule
```

基本器件的设计：32寄存器堆



Register File 逻辑结构

各信号引脚的功能：

- Rn1：A输出端口地址，5位，输入；
- Rn2：B输出端口地址，5位，输入；
- Wn：数据写入的寄存器号，5位，输入；

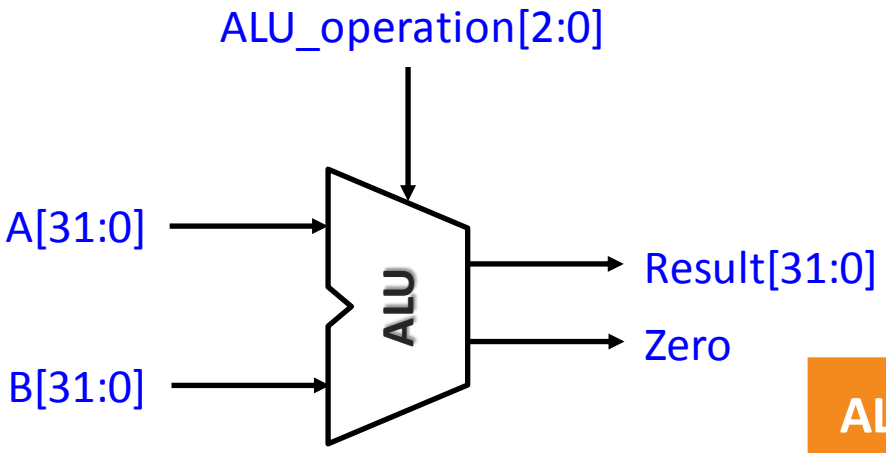
基本器件的设计：32寄存器堆

- Write：写信号，1位，输入；
- Wd：写入数据，32位，输入；
- A_：A端口的输出数据，32位，输出；
- B：B端口的输出数据，32位，输出；
- Clock：时钟信号，输入。

基本器件的设计：32寄存器堆

```
module RegFile(
          input  [4:0]  Rn1,Rn2,Wn,
          input         Write,
          input  [31:0] Wd,
          output [31:0] A,B,
          input         Clock
    );

    reg [31:0] Register[1:31]; //定义31个32位的寄存器

    //Read data
    assign A = (Rn1 == 0)? 0 : Register[Rn1];
    assign B = (Rn2 == 0)? 0 : Register[Rn2];

    //Write data
    always @ ( posedge Clock) begin
        if (( Write ) && ( Wn != 0)) Register[Wn] <= Wd;
    end

endmodule
```

基本器件的设计：ALU的设计

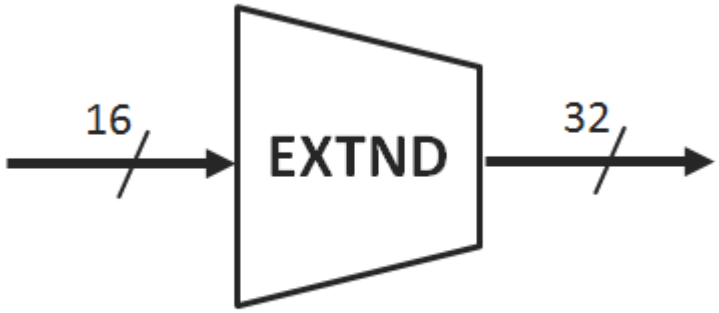


ALU_Operation	功能描述
000	Add（加）
100	Sub（减）
001	And（与）
101	Or（或）
010	Xor（异或）
110	设置高位

基本器件的设计：ALU的设计

```
module ALU(
    input  [31:0] A,B,
    input  [ 2:0] ALU_operation,
    output [31:0] Result,
    output        Zero
    );

    assign  Result = (ALU_operation == 3'b000) ? A + B :
                     (ALU_operation == 3'b100) ? A - B :
                     (ALU_operation == 3'b001) ? A & B :
                     (ALU_operation == 3'b101) ? A | B :
                     (ALU_operation == 3'b010) ? A ^ B :
                     (ALU_operation == 3'b110) ? {B[15:0],16'h0}
                     32'hxxxxxxxx;

    assign Zero = ~|Result;

endmodule
```

基本器件的设计：符号扩展器的设计



Sign extender

```
module Sign_Extender(
    input [15:0]  a ,
    output [31:0] b
    );

    assign b = $signed(a) ? { 16'hffff , a} :    { 16'h0 , a} ;

endmodule
```

POWERPOINT 2010