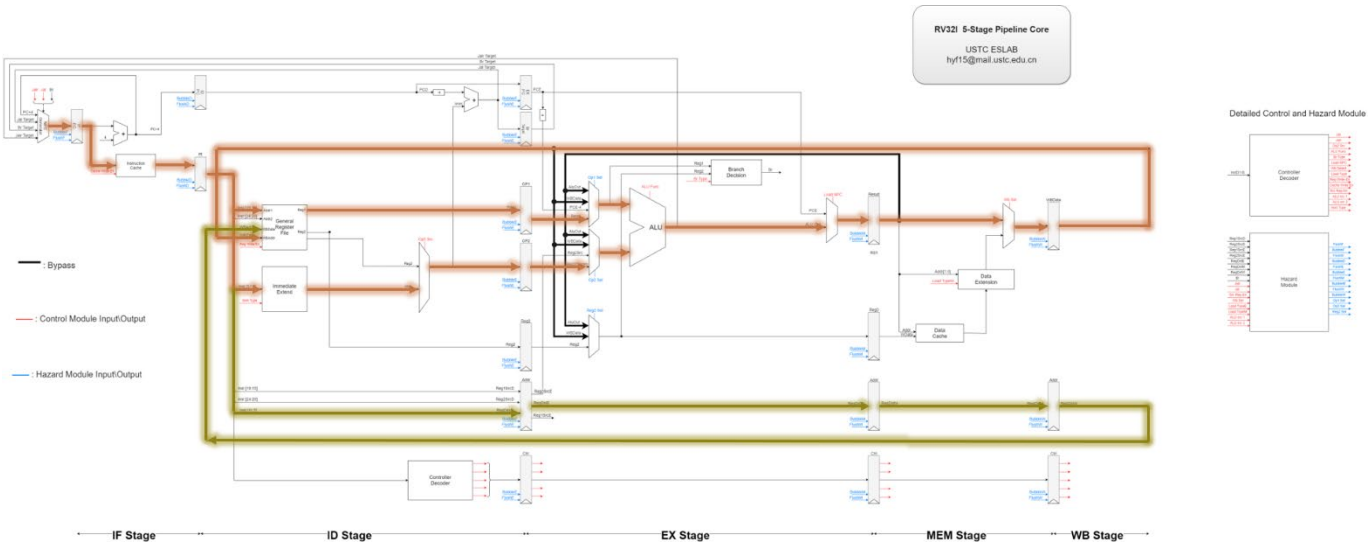
计算机体系结构 Lab1

廖洲洲 PB17081504

报告内容：

1. 描述执行一条ADDI 指令的过程（数据通路、控制信号等）。



1) IF：从Instruction Cache 中取出指令，PC加4。Jalr、Jal、Br=0,Cache_Write_En=0 。

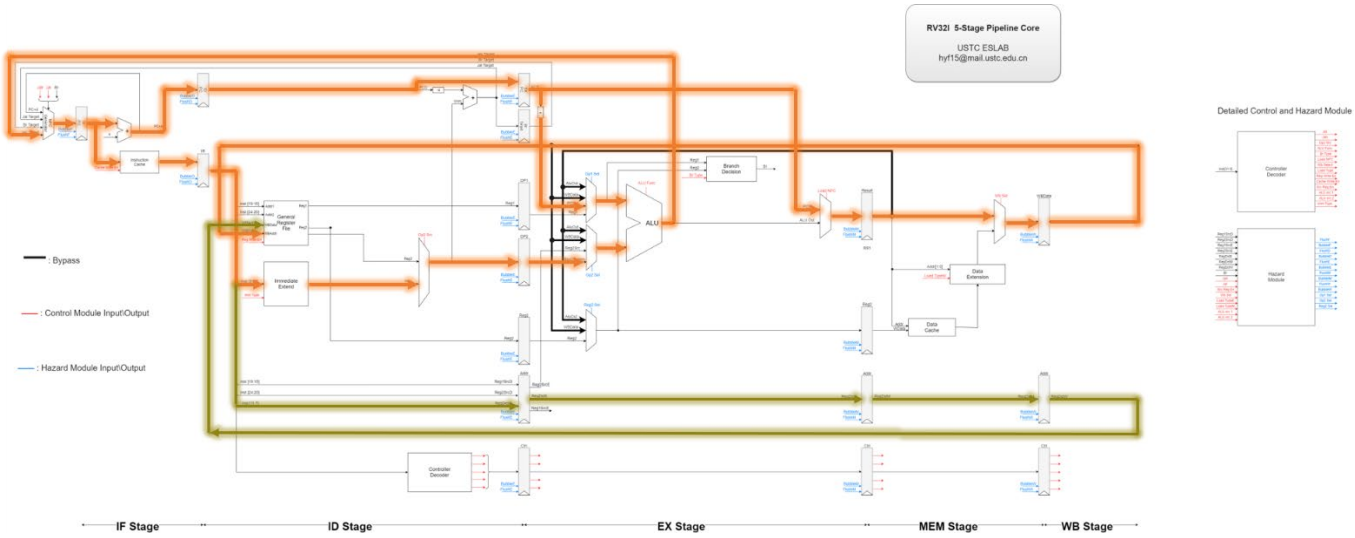
2) ID：从寄存器堆读出寄存器rs1的值，将12位立即数符号扩展。Reg_Write_En=0,Imm_Type=有符号数，Op2_Src=1(立即数)。

3) EX：对ALU的两个操作数进行加操作，结果暂存寄存器。ALU_Func=111(ADDI)，Op1_set=3(Reg1),Op2_set=3(Reg2OrImm),Load_NPC=1(ALU_Out)。

4) MEM：Wb_Set=0(ALU_Out)。

5) WB:选择目标寄存器，将ALU的结果写入寄存器。Reg_Write_En=1。

2. 描述执行一条JALR 指令的过程·（数据通路、控制信号等）。



1) IF：从Instruction Cache 中取出指令，PC加4。Jalr、Jal、Br=0,Cache_Write_En=0 。

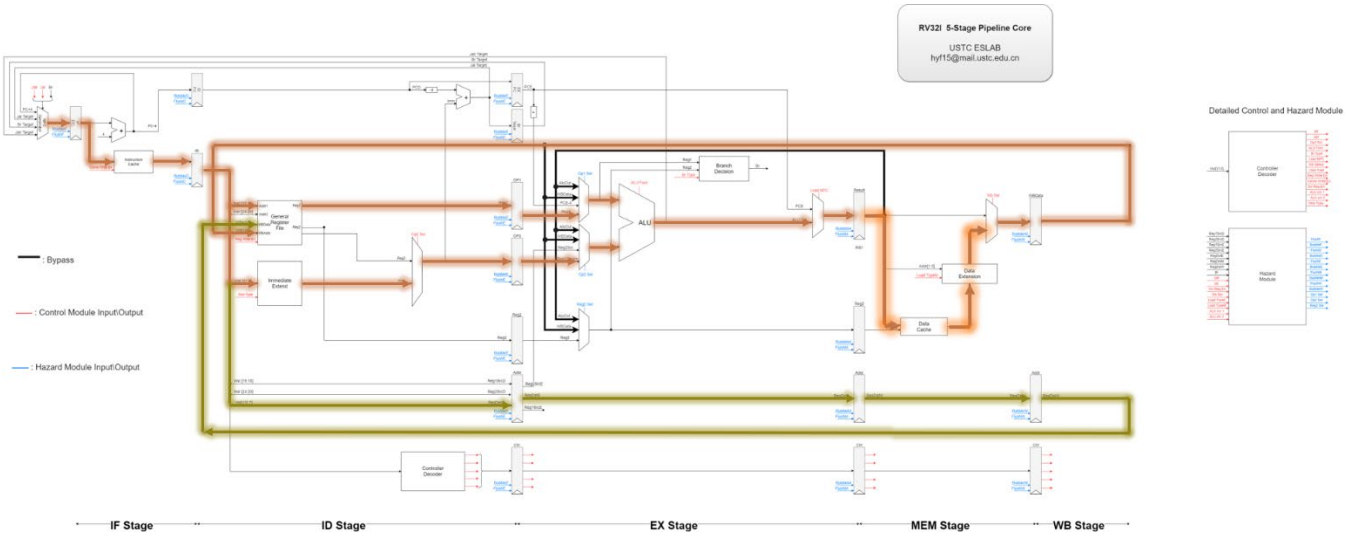
2) ID：从指令读出12位有符号立即数，将12位立即数符号扩展。Reg_Write_En=0,Imm_Type=有符号数，Op2_Src=1(立即数)。

3) EX：对 ALU 的两个操作数进行加操作，结果送入 NPC Generator，NPC Generator 将目标地址的最低位设置为 0。ALU_Func=111(ADDI)，Op1_set=2(PCE-4),Op2_set=3(Reg2OrImm),Load_NPC=1(PCE)，Jalr=1，Jal=0，Br=0。

4) MEM：Wb_Set=0(PCE)。

5) WB:选择目标寄存器，将pc+4写入寄存器。Reg_Write_En=1。

3. 描述执行一条LW 指令的过程（数据通路、控制信号等）。



1) IF：从Instruction Cache 中取出指令，PC加4。Jalr、Jal、Br=0,Cache_Write_En=0 。

2) ID：从寄存器堆读出寄存器rs1的值，将12位立即数符号扩展。Reg_Write_En=0,Imm_Type=有符号数，Op2_Src=1(立即数)。

3) EX：对ALU的两个操作数进行加操作，结果暂存寄存器。ALU_Func=111(ADDI)，Op1_set=3(Reg1),Op2_set=3(Reg2OrImm),Load_NPC=1(ALU_Out)。

4) MEM：将ALU的结果作为数据Cache的地址，Wb_Set=1(Mem data),Load_TypeM=000~101。

5) WB:选择目标寄存器，将ALU的结果写入寄存器。Reg_Write_En=1。

4. 如果要实现CSR 指令（csrrw，csrrs，csrrc，csrrwi，csrrsi，csrrci），设计图中还需要增加什么部件和数据通路？给出详细说明。

增加CSR File，使得可以读写CSR FILE，输入端口有inst[20:31]，写使能,写数据，输出对应CSR中的旧值；增加零扩展部件，使得可以将CSR的旧值零扩展到32位；增加CSR旧值暂存寄存器，以便WB阶段将CSR旧值写入rd；给ALU增加按位掩码功能，以实现CSRRS、CSRRC、CSRRSI、CSRRCI的按位掩码功能，其输入是rs1中的值或零扩展到32位的5位立即数（zimm[4:0]）与CSR中的旧值，根据func3判断将对应位置0或置1；增加Op1_src部件，以选择Op1数据来源，增加Op1数据来源零扩展到32位的5位立即数（zimm[4:0]）；增加Op2_src的数据来源-CSR中的旧值。

1) IF：不变

2) ID：从CSR FILE读出CSR的旧值，并0扩展到32位；从寄存器堆读出寄存器rs1的值并直接从指令读出立即数的值，将立即数0扩展。增加的控制信号CSR_Write_En=0，Op1_src=rs1 or 扩展的zimm[4:0]，Op2_src=CSR旧值。

3) EX：对ALU的两个操作数进行按位掩码操作，结果暂存寄存器。ALU_Func=掩码为0或1，Op1_set=3(Reg1Orzimm[4:0]),Op2_set=3(Reg2OrImmOrCsr),Load_NPC=1(ALU_Out)。

4) MEM：不变。

5) WB: 选择CSR寄存器，将ALU的结果写入CSR寄存器；选择rd寄存器，将CSR旧值写入rd寄存器。Reg_Write_En=1，CSR_Write_En=1。

5. 哪些指令分别采用了五类立即数（I-type，S-type，B-type，U-type，J-type 至少各举一例）？Verilog 如何将这些立即数拓展成32 位的?

I-type:ADDI,ANDI,ORI,XORI,SLTI,SLTIU等;

S-type:store操作-SB,SH,SW;

B-type:BEQ,BNE,BGE,BGEU等;

U-type:LUI,AUIPC等;

J-type:JAL等。

设计一个立即数拓展模块，根据控制信号Imm-Type可以将inst[31:7]中的立即数分解出来，然后对它们按位重组，由于每种立即数的位数是确定的，然后根据有、无符号数对高位部分进行拓展即可。

代码:

```
module Immediate_Extend(
input [2:0]Imm_Type,
input [31:7]Inst,
output reg [31:0]out
  );
always @(*)
    begin
    case(Imm_Type)
    3'b000:   //I
    begin
        out<={ {21{Inst[31]}} , Inst[30:20]};
    end
    3'b001:   //S
    begin
        out<={ {21{Inst[31]}}, Inst[30:25], Inst[11:8], Inst[7]};
    end
    3'b010:   //B
    begin
        out<={ {20{Inst[31]}}, Inst[7],Inst[30:25],Inst[11:8],1'b0};
    end
    3'b011:   //U
    begin
        out<={ Inst[31:12],12'b000000000000};
    end
    3'b100:   //J
    begin
        out<={ {12{Inst[31]}}, Inst[19:12], Inst[20], Inst[30:25], Inst[24:21],1'b0};
    end
    default:out<=32'hxxxxxxxx;
```

```
            endcase
        end
endmodule
```

6. 如何实现Data Cache 的非字对齐的Load 和Store?

   Load:将计算所得地址低两位保存下来，然后将地址的低两位清零，从Data Cache中取出4字节数据，然后根据地址低两位从Data Extension中取出数据,如果所取数据范围超过了这4个字节，则另取出下一块4字节数据，再从中选择。

   Store：将计算所得地址低两位保存下来，然后将地址的低两位清零，从Data Cache中选中4字节数据，然后根据后两位地址选中要写的字节块，如果要写的数据范围超过了这4个字节，则另取下一块4字节地址，再从中选择。

7. ALU 模块中，默认wire 变量是有符号数还是无符号数?

   无符号数

8. 哪条指令执行过程中会使得Load Npc == 1?

   JALR指令 and JAL指令

9. NPC Generator 中对于不同跳转target 的选择有没有优先级?

   有，比如Branch指令是否跳转需要再EX阶段才能出结果，而jal指令是无条件跳转，在ID阶段就能计算出目标地址，这样当Branch指令后跟一条Jal指令时，会发现它们的跳转使能信号在同一周期有效，故需要划分优先级。

10. Harzard 模块中，有哪几类冲突需要插入气泡?

    RAW冲突中：lw后跟需要读lw指令中被写寄存器的值。

11. Harzard 模块中采用默认不跳转的策略，遇到branch 指令时，如何控制flush 和stall 信号?

    遇到Branch指令时，继续执行顺序的指令流，如果在EX阶段发现分支发生，则将IF/ID、ID/EX 段寄存器的flush置1，stall置0。

12. 0 号寄存器值始终为 0，是否会对 forward 的处理产生影响?

    会，假如有人尝试向 0 号寄存器写值，然后下一条指令对 0 号寄存器读，这时会有写后读相关，如果不特殊出路，forward 会将要写到 R0 的值进行转发，这样使得读到的 R0 值不为 0，产生错误。