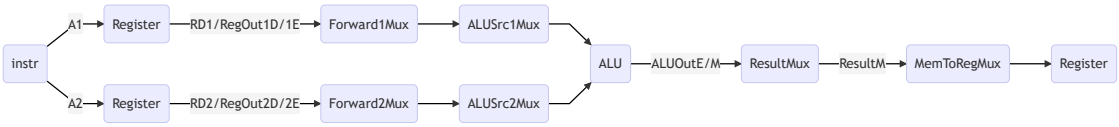
1. 描述执行一条 XOR 指令的过程（数据通路、控制信号等）。

   假设没有出现Forward、Hazard。

   数据通路：



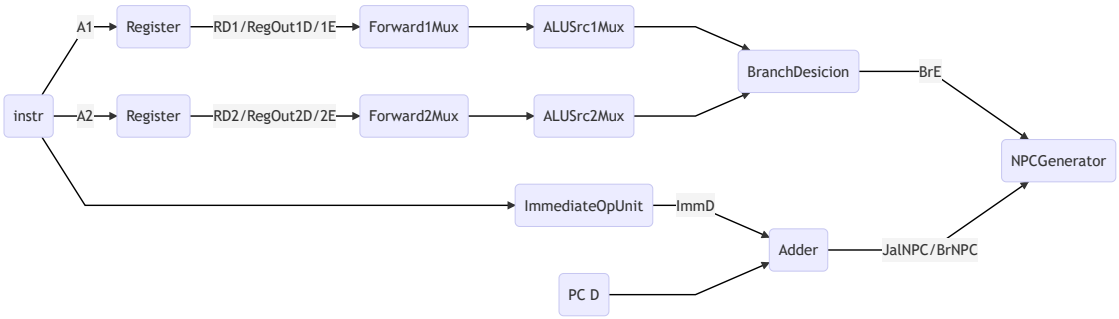
控制信号（Mux按图从上向下进行端口编码）：

控制信号名	控制信号含义	值	控制信号名	控制信号含义	值
Forward1E	Mux-FromRegOut1E	2'b00	ALU Control	XOR	3'b100
Forward2E	Mux-FromRegOut2E	2'b00	Reg Write	LW	3‘d3
ALUSrc1	Mux-FromForward1E	1'b1	MemToReg	From ALU	1'b1
ALUSrc2	Mux-FromForward2E	2'b00	MemWrite	-	4'b0
ImmType	-	-	BranchType	No Branch	3'b000
LoadNpc	-	1'b0	JalD	-	1'b0
			JalrD	-	1'b0

2. 描述执行一条 BEQ 指令的过程·（数据通路、控制信号等）。

   假设没有出现Forward、Hazard。

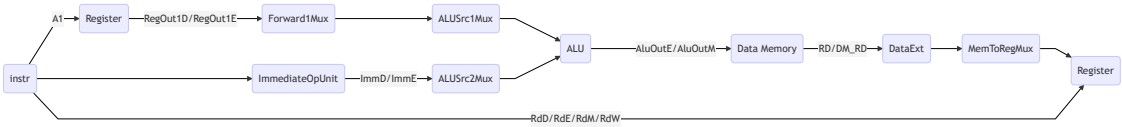
   数据通路：



控制信号（Mux按图从上向下进行端口编码）：

控制信号名	控制信号含义	值	控制信号名	控制信号含义	值
Forward1E	Mux-FromRegOut1E	2'b00	ALU Control	-	-
Forward2E	Mux-FromRegOut2E	2'b00	Reg Write	-	3'b0
ALUSrc1	Mux-FromForward1E	1'b1	MemToReg	-	-
ALUSrc2	Mux-FromForward2E	2'b00	MemWrite	-	4'b0
ImmType	BTYPE	3'b011	BranchType	BEQ	3'b001
LoadNpc	-	1'b0	JalD	-	1'b0
			JalrD	-	1'b0

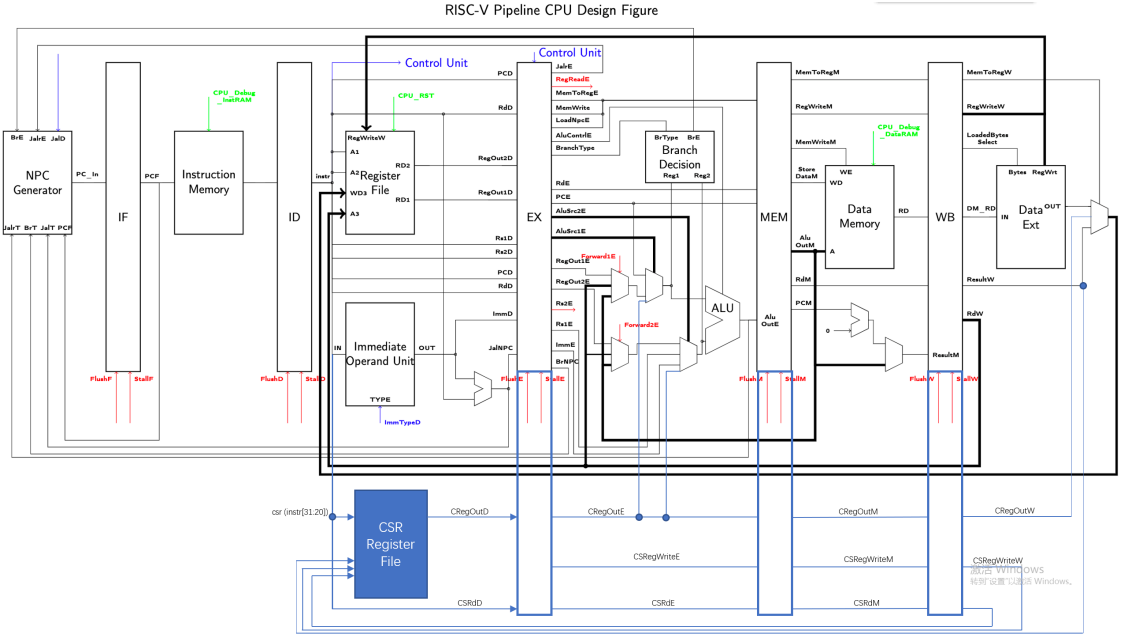
3. 描述执行一条 LHU 指令的过程（数据通路、控制信号等）。



控制信号名	控制信号含义	值	控制信号名	控制信号含义	值
Forward1E	Mux-FromRegOut1E	2'b00	ALU Control	ADD	4'd3
Forward2E	-	-	Reg Write	LH	3'd2
ALUSrc1	Mux-FromForward1E	1'b1	MemToReg	From Mem	1'b0
ALUSrc2	Mux-FromImmE	2'b10	MemWrite	-	4'b0
ImmType	ITYPE	3'b001	BranchType	No Branch	3'b0
LoadNpc	-	1'b0	JalD	-	1'b0
			JalrD	-	1'b0

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。

   如图所示，添加csr到ALU，csr到MemToReg MUX与ALU到csr的数据通路



5. Verilog 如何实现立即数的扩展？

   利用Verilog对wire的拼接实现立即数的扩展。代码如下：

```
always@(*)
    begin
        case(Type)
            `ITYPE: Out <= { {21{In[31]}}, In[30:20] };
            `RTYPE: Out <= 32'b0;
            `STYPE: Out <= { {31{In[31]}}, In[30:25], In[11:7] };
            `BTYPE: Out <= { {20{In[31]}}, In[7], In[30:25], In[11:8], 1'b0 };
            `UTYPE: Out <= { In[31:12], 12'b0 };
            `JTYPE: Out <= { {12{In[31]}}, In[19:12], In[20], In[30:21], 1'b0 };
            default:Out<=32'hxxxxxxxx;
        endcase
    end
```

6. 如何实现 Data Cache 的非字对齐的 Load 和 Store?

   根据非对齐地址的最后两位与将要存入的数据的有效位，进行对应的移位操作，使其依照小端存储规则存储到相应位置，具体实现代码如下（仅展示了Store）：

```
    // 根据DataRam模块描述，wea传入值与写入满足以下关系：
    // |  WE  | 写入地址 |
    // | ---- | ------- |
    // | 0001 | [7:0]   | 小端
    // | 0010 | [15:8]  |
    // | 0100 | [23:16] |
    // | 1000 | [31:24] | 大端
    // 若写入类型为SB，则CU给出WE=0001
    // 目标写入地址为 xxxx00，则写入到xxxx00的[7:0]
    // 目标写入地址为 xxxx01，则写入到xxxx00的[15:8]
    // 由小端的特性决定我们只需要将WE进行一次左移操作即可实现写入到目标位置，而由于字大小为8，故WD数据需要偏移的量为A[1:0] * 8 即左移3位
    DataRam DataRamInst (
        .clk    ( clk                   ),
        .wea    ( WE << A[1:0]          ),
        .addra  ( A[31:2]               ),
        .dina   ( WD << (A[1:0] << 3)   ),
        .douta  ( RD_raw                ),
        .web    ( WE2                   ),
        .addrb  ( A2[31:2]              ),
        .dinb   ( WD2                   ),
        .doutb  ( RD2                   )
    );
```

   需要注意的是，以当前的设计模型，若数据跨越了一个地址块，则CPU无法在一条指令中完成读取，因为其Hazard Unit没有针对多周期读取与拼接的Stall检测。

7. ALU 模块中，默认 wire 变量是有符号数还是无符号数?

   默认为无符号数

8. 简述BranchE信号的作用。

   BranchE信号有效时说明分支决策器判断代码需要执行分支指令，NPC Generator接收到该信号时将内部的PC寄存器值改为BrNPC数据线中的值。

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

   有优先级。Branch、Jalr的跳转优先级高于Jal跳转。

10. Harzard 模块中，有哪几类冲突需要插入气泡，分别使流水线停顿几个周期?
    - Load指令后接任意需要用到Load指令的目标寄存器的值的指令：停顿一个周期
    - 分支预测错误：停顿2个周期
11. Harzard 模块中采用静态分支预测器，即默认不跳转，遇到 branch 指令时，如何控制 flush 和 stall 信 号?

    IF、ID段寄存器Flush信号置为有效，其余Flush与Stall信号保持无效

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

    会产生影响。若指令1向0号寄存器写入数据，而指令2从0号寄存器中读取。若不做特殊处理，转发控制器会进行一次转发，导致指令2从0号寄存器获得的值不一定是0，从而导致结果出错，故需要特殊处理这种情况。