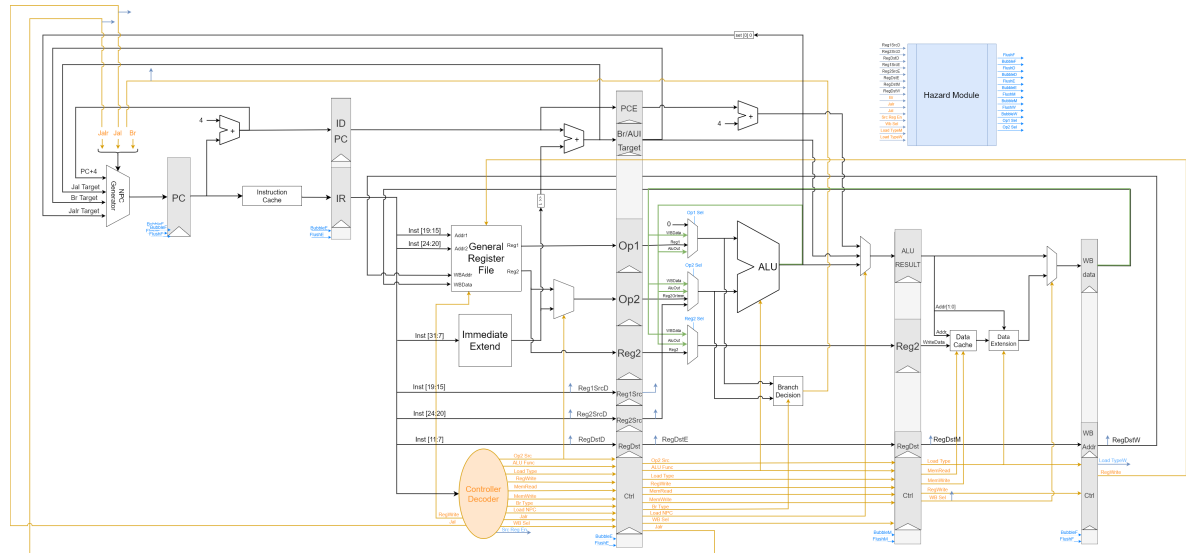


# 体系结构lab1

PB17111623

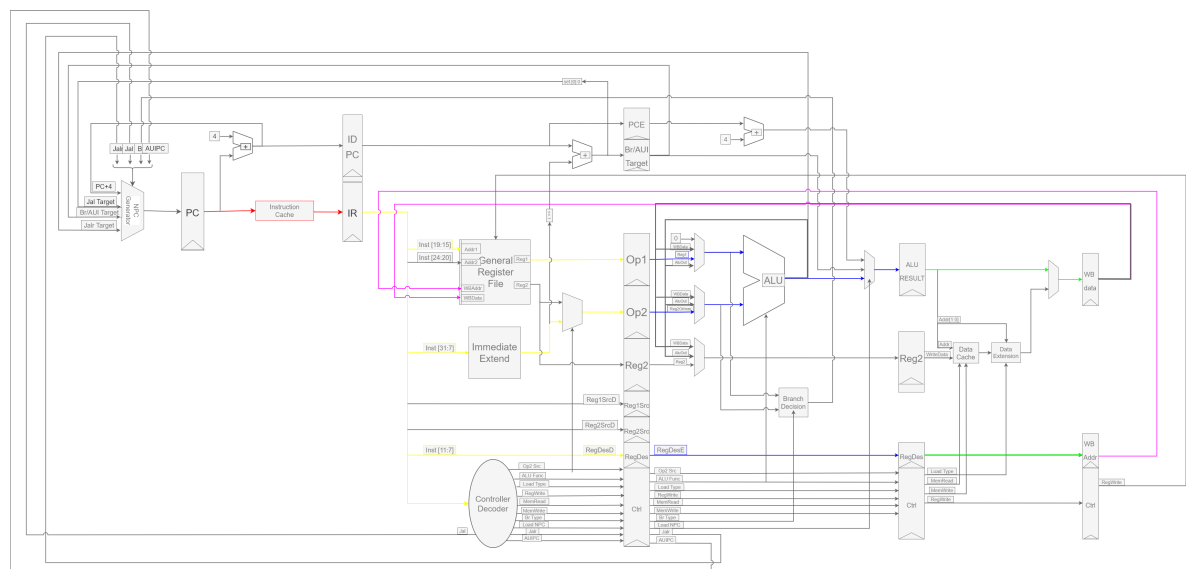
范睿

## Core数据通路



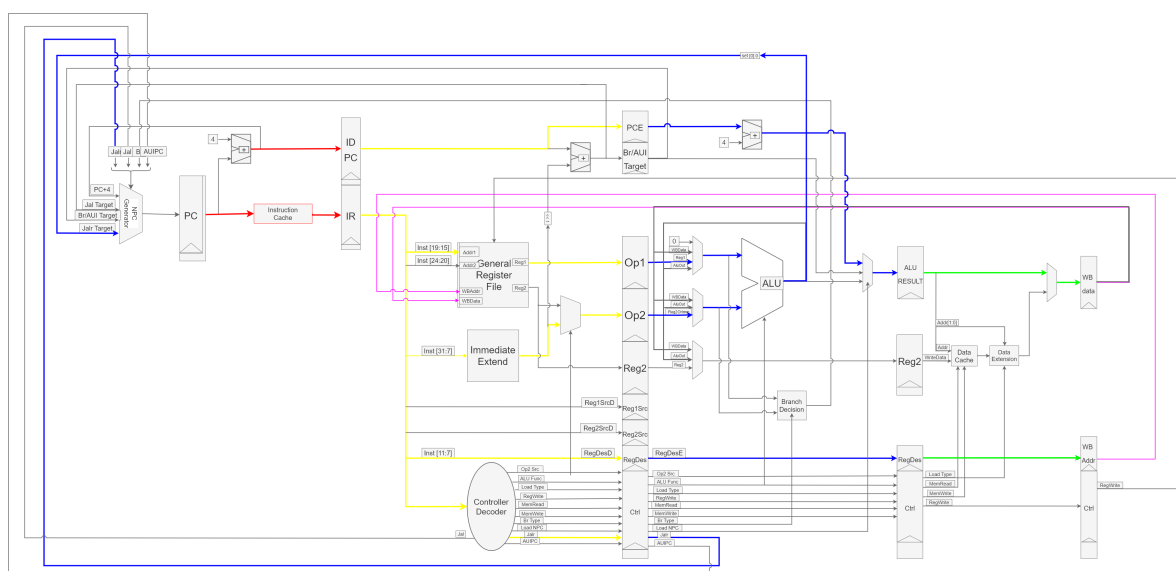
## 1.描述执行一条 ADDI 指令的过程

ADDI指令功能：ADDI将符号扩展的12位立即数加到寄存器rs1上。算术溢出被忽略，而结果就是运算结果的低XLEN位。ADDI rd,rs1,0用于实现MV rd,rs1汇编语言伪指令。



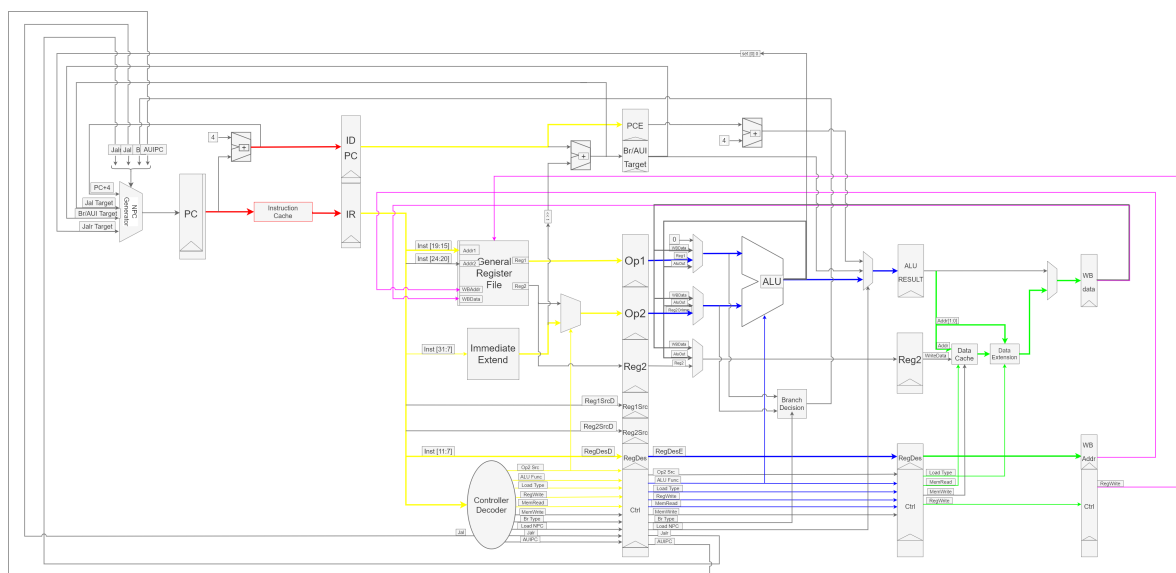
	取指 (红)	译码 (黄)	执行 (蓝)	访存 (绿)	写回 (紫)
数据通路	PC指向的本条ADDI指令进入IR PC=PC+4	[19:15]被送到Register File的Addr1 [31:7]被送入Immediate Extend, 其中[31:20]是真正有效的立即数 [11:7]被送入RegDst寄存器, WB时使用指令还进入control unit, 产生控制信号	ALU得到两个操作数执行相应运算 写回地址继续传递 Ctrl信号向前传递	ALU计算结果进入WBData 写回地址继续传递 Ctrl信号向前传递	Addr准备好写回地址, WBData准备好写回数据, 一起写回Register File
控制信号	Jar=0 Jalr=0 Br=0	Op2 Src=1 (选择立即数)	Op1 Sel=3 Op2 Sel=4 ALU Func=ADDI	Load NPC=0 (选择ALU Out)	RegWrite=1

## 2.描述执行一条 JALR 指令的过程



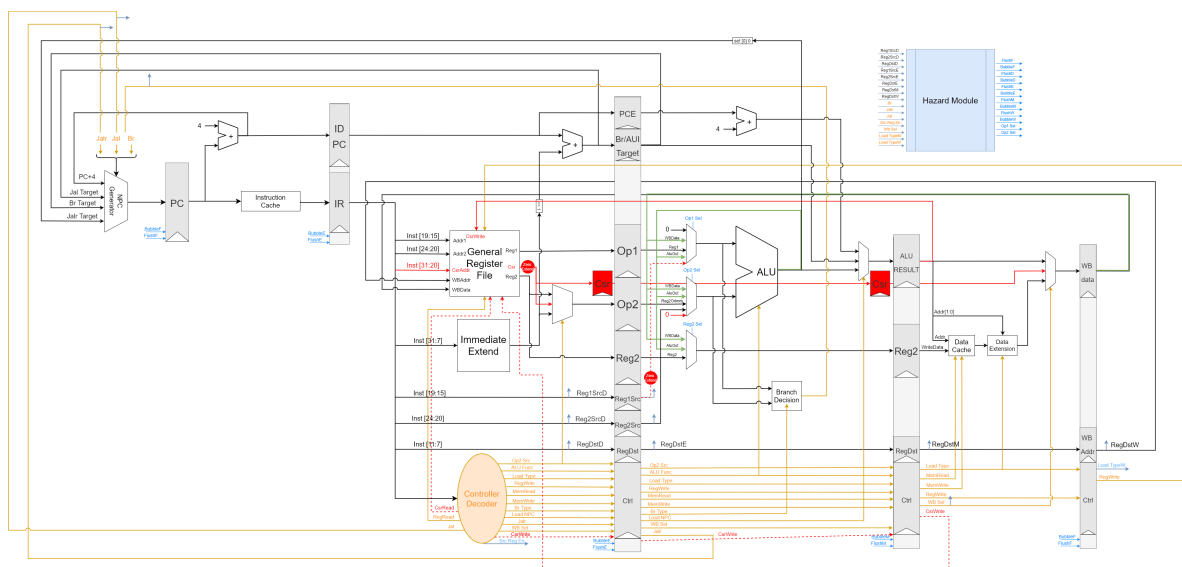
	取指 (红)	译码 (黄)	执行 (蓝)	访存 (绿)	写回 (紫)
数据通路	PC指向的本条JALR指令进入IR PC=PC+4	[19:15]被送入Addr1, 读取rs1的值 [31:7]被送入Immediate Extend, 真正有效的立即数是[31:20] [11:7]被送入RegDst, 保存rd ID PC被送入PCE	ALU执行加法, 得到Jalr Target送入给PC RegDst继续传递PCE的值+4获得下一条指令的位置, 送入ALU Result	ALU Result传递给WB data RegDst继续传递	将WB Data写回给WB Addr指向的地址
控制信号	Jalr=0 Jalr=0 Br=0	Jalr=1, 进入ID/EX段的Ctrl Op Src2=1, 选择Imm	Jalr=1 Load NPC=1, 选择PCE	Wb Sel=0, 选择ALU Result	RegWrite=1

### 3.描述执行一条 LW 指令的过程



	取指 (红)	译码 (黄)	执行 (蓝)	访存 (绿)	写回 (紫)
数据通路	PC指向的本条JALR指令进入IR PC=PC+4	[19:15]被送入Addr1,读取rs1的值 [31:7]被送入Immediate Extend, 真正有效的立即数是[31:20] [11:7]被送入RegDst, 保存rd	ALU执行加法操作, 传递给ALU Result RegDst继续传递	根据ALU Result进行访存操作, 取出对应位置的值 Data Extension先根据Addr[1:0]看看地址有无对齐, 再根据Load Type判断区多长的数据	将WB Data写回给WB Addr指向的地址
控制信号	Jar=0 Jalr=0 Br=0	Op Src2=1, 选择Imm	Load NPC=0, 选择ALU的结果	MemRead=1 Load Type=指令中func3对应的宽度对应的数字	RegWrite=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。**



红色为新添加的数据通路。红色实线为数据通路，虚线为控制指令。

CSR类的指令的功能可以概括为：

- 取CSR中的数据，扩展运算后写入rd
- 取rs1中的数据（CSRRW）或CSR中的数据按照rs1或zimm运算后（其他）写入CSR

### 更新RD

- 取指阶段不变
- 译码阶段：将CSR数据取出来进入CSR寄存器（红色的寄存器），rd同时进入RegDst
- 执行阶段：CSR寄存器进行传递，RegDst同时传递
- 写回阶段：WB Sel选择CSR作为WB data，正常写回

### 更新CSR

- 取指阶段不变
- 译码：（分成三个部分讨论）
  - CSRRW: Inst[19:15]进入Register File，读取rs1进入Op1
  - CSRRS, CSRRC: Inst[19:15]进入Register File，读取rs1进入Op1；CSR进入Op2
  - CSRWI, CSRSI, CSRCI: Inst[19:15]进入Reg1Src
- 执行：
  - CSRRW: Op1和0作为ALU的两个操作数，执行加法后进入ALU RESULT
  - CSRWI: Reg1Src和0作为ALU的两个操作数，执行加法后进入ALU RESULT
  - CSRRS, CSRRC, CSRSI, CSRCI: Op1（rs1的值或zimm扩展后的值）和Op2（此时是CSR的值）作为ALU的两操作数，根据规则进行运算后，结果进入ALU RESULT
- 访存：CSRWrite为1，将ALU RESULT写回到CSR
- 写回：更新CSR的过程写回阶段不做事

## 5.哪些指令分别采用了五类立即数？ Verilog 如何将这些立即数拓展成 32 位的？

- I-Type: OP-IMM类指令，LOAD指令，JALR

```
1 | assign Imm = {{21{Inst[31]}}, Inst[30:20]};
```

- S-Type: STORE

```
1 | assign Imm = {{21{Inst[31]}}, Inst[30:25], Inst[11:7]};
```

- B-Type: BRANCH类指令，比如BEQ

```
1 | assign Imm = {{20{Inst[31]}}, Inst[7], Inst[30:25], Inst[11:8], 1'b0}
```

- U-Type: LUI, AUIPC

```
1 | assign Imm = {Inst[31:12], 12'b0}
```

- J-Type: JAL

```
1 | assign Imm = {{12{Inst[31]}}, Inst[19:12], Inst[20], Inst[30:21], 1'b0}
```

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

**Load:** 多次读取加以拼接

**Store:** 先将数据覆盖的对齐数据多次取出来，依次修改后写入

## 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和Jal会使Load NPC为1。

在我的设计中，AUIPC会使Load NPC为2

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

Jalr和Br的跳转地址在EX段被计算出来，Jal的地址在ID被计算出来。

当流水线的EX段是Jalr或Br，且ID段是Jal时，会先响应Jalr或Br，不再响应Jal。因为这相当于代码中Jalr或Br指令在Jal之前。

因此，优先级Jalr=Br>Jal

## 10. Harzard 模块中，有哪几类冲突需要插入气泡?

数据相关

- Load指令和ALU指令的RAW相关需要在ALU指令IF段插入一个气泡

## 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?

- 跳转指令进入ID阶段时，CPU发现这是一个跳转指令
- EX阶段时，IR正常更新，跳转指令不跳转的下一条指令进入译码阶段
- 若跳转结果为不跳转，则无flush
- 若跳转结果为跳转，则flush IF和ID段（此时是跳转指令不跳转的下一条指令），PC更新为新的指令地址

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

有可能会产生影响。

若用户向0号寄存器写入非零值，且此非0值被转发，那么得到值的指令本该得到0，却得到了一个非0数，产生错误。