lab2-2 实验报告

计算机组织与系统结构实习报告 Lab 2.2

学号：2200013093

姓名：张佳豪

大班教师：陆俊林

Part I：RISC-V 多周期模拟器 （50分）

基于实现的RISC-V ISA，给出指令各阶段的寄存器传输级描述 （10分）。每类指令举1-2个例子即可

R

add rd, rs1, rs2

阶段	操作描述
取指	`IR <- imem[PC]`，`NPC <- PC + 4`
译码	`opcode=0x33`，`func3=0x0`，`func7=0x00`；`raddr1=IR[19:15]`，`raddr2=IR[24:20]`，`waddr=IR[11:7]`
执行	`ALUOut <- rs1 + rs2`
访存	空
写回	`RegFile[rd] <- ALUOut`，`PC <- NPC`

I

ld rd, offset(rs1)

阶段	操作描述
取指	`IR <- imem[PC]`，`NPC <- PC + 4`
译码	`imm=SignExtend(IR[31:20])`，`raddr1=IR[19:15]`,`waddr=IR[11:7]`
执行	`ALUOut <- rs1 + imm`

阶段	操作描述
访存	`LMD <- dmem[ALUOut]_64`
写回	`RegFile[rd] <- LMD`, `PC <- NPC`

S

sd rs2, offset(rs1)

阶段	操作描述
取指	`IR <- imem[PC]`, `NPC <- PC + 4`
译码	`imm=SignExtend({IR[31:25], IR[11:7]})`, `raddr1=IR[19:15]` ,`raddr2=IR[24:20]`
执行	`ALUOut <- rs1 + imm`
访存	`dmem[ALUOut]_64 <- rs2`
写回	`PC <- NPC`

SB

beq rs1, rs2, offset

阶段	操作描述
取指	`IR <- imem[PC]`, `NPC <- PC + 4`
译码	`imm=SignExtend({IR[31], IR[7], IR[30:25], IR[11:8]}) << 1`, `raddr1=IR[19:15]` ,`raddr2=IR[24:20]`
执行	`ALUOut <- PC + imm`
访存	空
写回	`PC <- (rs1 == rs2) ? ALUOut : NPC`

U

auipc rd, offset

阶段	操作描述
取指	`IR <- imem[PC]`, `NPC <- PC + 4`
译码	`imm=IR[31:12] << 12`, `waddr=IR[11:7]`

阶段	操作描述
执行	`ALUOut <- PC + imm`
访存	空
写回	`RegFile[rd] <- ALUOut`, `PC <- NPC`

UJ

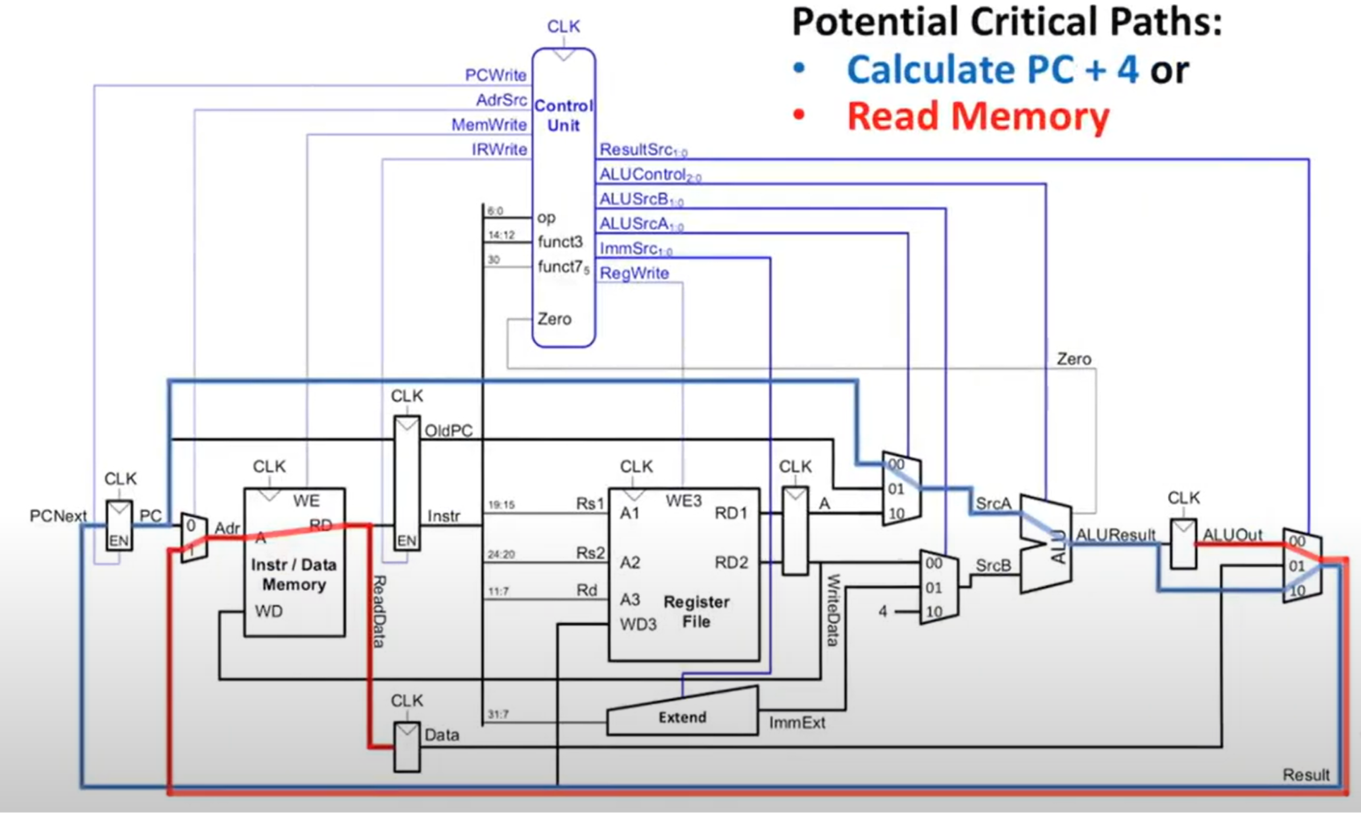
jal rd, offset

阶段	操作描述
取指	`IR <- imem[PC]`, `NPC <- PC + 4`
译码	`imm=SignExtend({IR[31], IR[19:12], IR[20], IR[30:21]}) << 1`, `waddr=IR[11:7]`
执行	`ALUOut <- PC + imm`
访存	空
写回	`RegFile[rd] <- NPC`, `PC <- ALUOut`

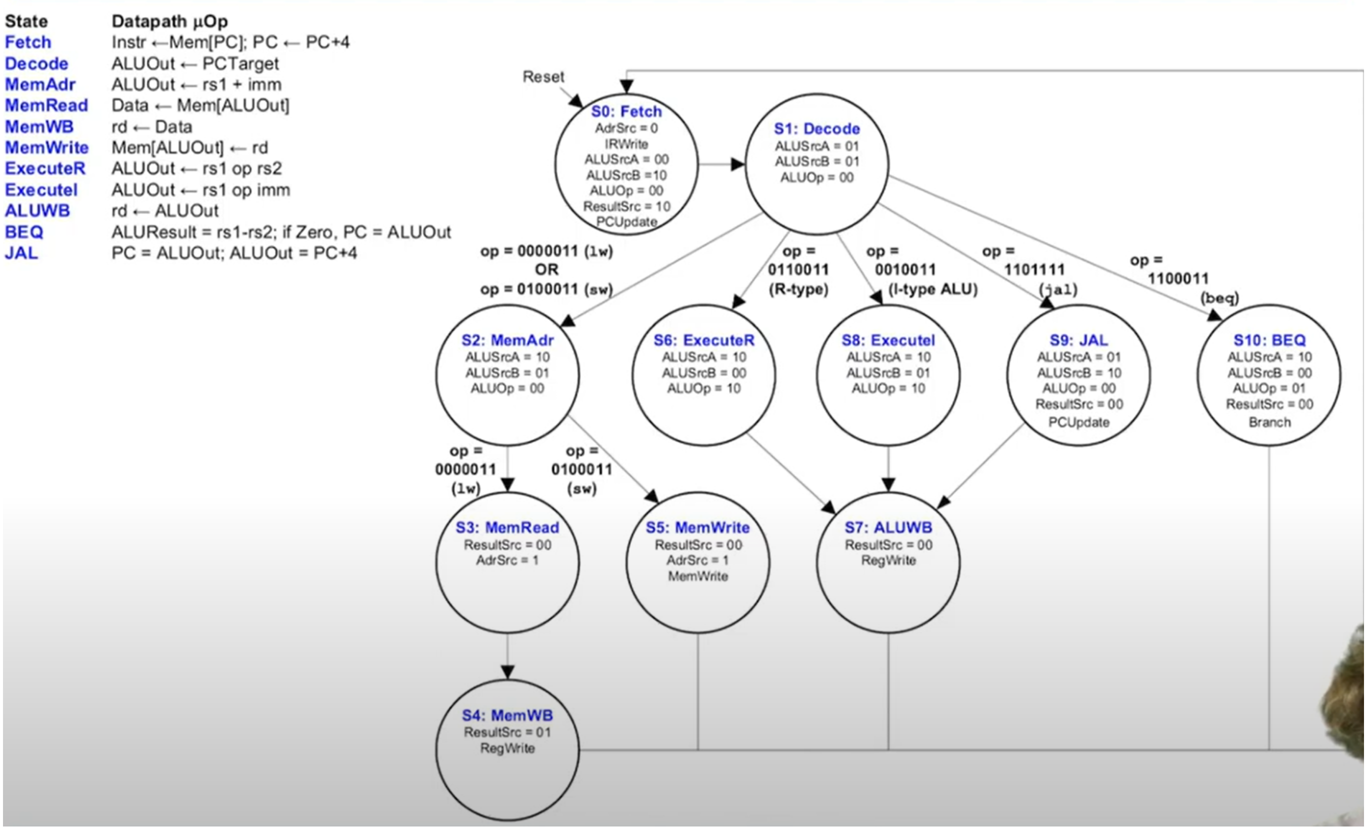
基于以上分析，给出多周期处理器的数据通路图和控制信号产生逻辑。不限形式，手绘也可。（10分）

> 为确保正确性，数据通路图和控制信号产生逻辑均使用了DDCA Ch7的设计 此外不一定和上一问保持一致，比如PC可在在S0一个phase更新，因为设计中有直接的数据通路。

数据通路:



控制信号:



State	Datapath μOp
Fetch	Instr ← Mem[PC]; PC ← PC+4
Decode	ALUOut ← PCTarget
MemAdr	ALUOut ← rs1 + imm
MemRead	Data ← Mem[ALUOut]
MemWB	rd ← Data
MemWrite	Mem[ALUOut] ← rd
ExecuteR	ALUOut ← rs1 op rs2
ExecuteI	ALUOut ← rs1 op imm
ALUWB	rd ← ALUOut
BEQ	ALUResult = rs1-rs2; if Zero, PC = ALUOut
JAL	PC = ALUOut; ALUOut = PC+4

运行测试程序，给出动态执行的指令数。（共5个定点程序，每个2分）

运行测试程序，给出多周期处理器的执行周期数，并计算平均CPI。（共5个定点程序，每个4分）

> 由于在lab2-1中修改了起始地址为0x08000000, 因此使用源代码编译测试，而不直接使用二进制程序

这一部分的模拟器实现只需要预先保存一个表格记录每个指令所需的周期，然后执行过程中累加即可。 特殊处理为div和rem连续的情况，测试程序保存上一条指令进行特判。

运行测试程序的命令为：

```
./driver.sh --perf multicycle -E "lab2"
```

结果

```
Checking Dependencies...
Build Simulator...
Simulator building finished.
Processing: lab2-2-1
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 74
  Dynamic cycles: 305
  CPI: 4.12
```

```
Processing: lab2-2-10
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 64
  Dynamic cycles: 303
  CPI: 4.73
Processing: lab2-2-2
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 74
  Dynamic cycles: 305
  CPI: 4.12
Processing: lab2-2-3
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 149
  Dynamic cycles: 620
  CPI: 4.16
Processing: lab2-2-4
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 209
  Dynamic cycles: 875
  CPI: 4.19
Processing: lab2-2-5
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 150
  Dynamic cycles: 819
  CPI: 5.46
Processing: lab2-2-6
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 110
  Dynamic cycles: 649
  CPI: 5.90
Processing: lab2-2-7
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 179
  Dynamic cycles: 745
  CPI: 4.16
Processing: lab2-2-8
Success
Performance Metrics:
  Performance Profiler: Multicycle
  Dynamic instructions: 106
```

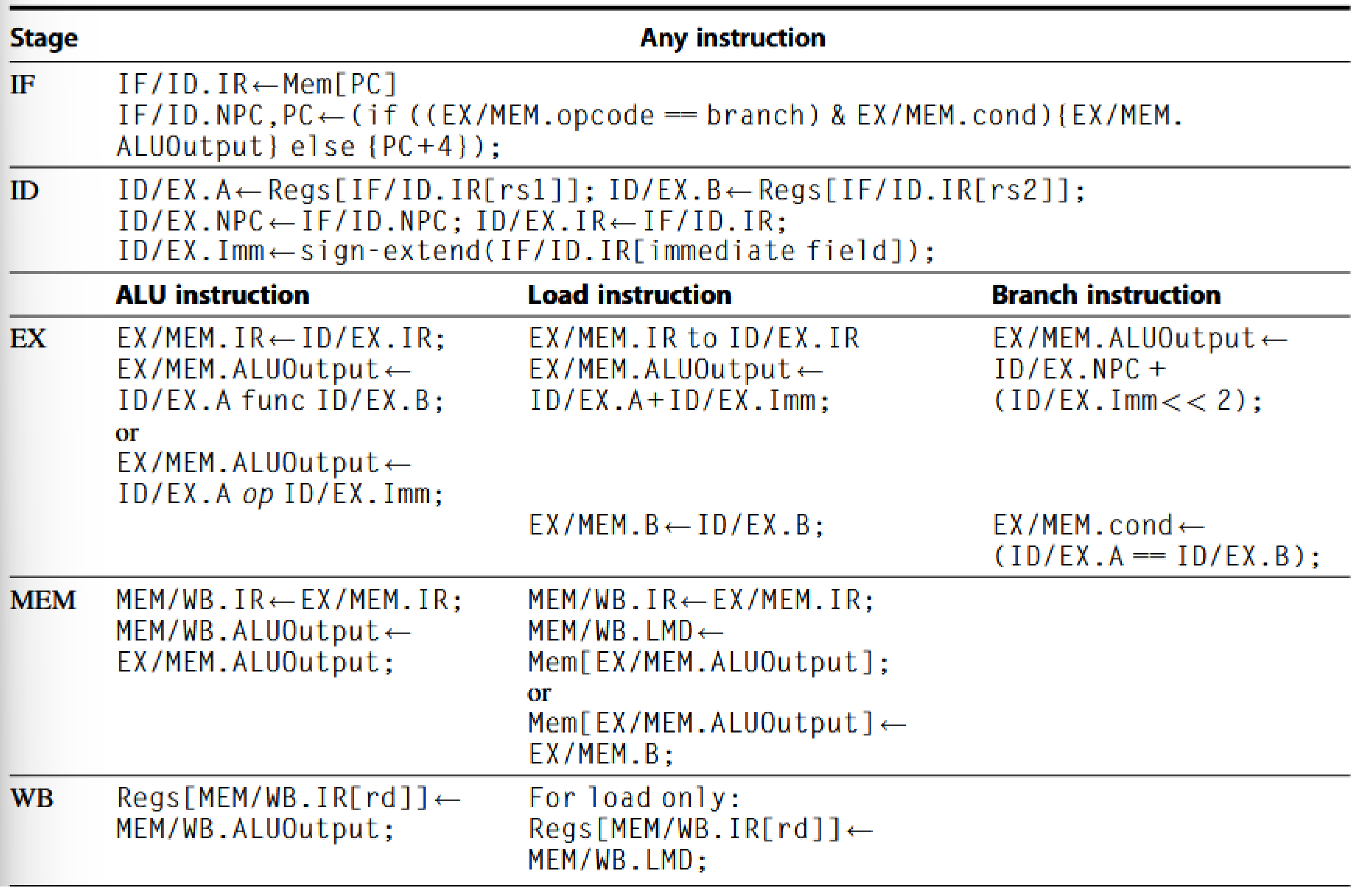
```
    Dynamic cycles: 434
    CPI: 4.09
  Processing: lab2-2-9
  Success
  Performance Metrics:
    Performance Profiler: Multicycle
    Dynamic instructions: 74
    Dynamic cycles: 344
    CPI: 4.65

  Score: 10/10 (filtered: 'lab2')
```

Part II：RISC-V 流水线模拟器（50分）

基于实现的RISC-V ISA，给定流水线处理器的阶段划分，并简单介绍各阶段的工作。 （5分）

> Part II的设计参考《Computer Architecture A Quantitative Approach》五级划分IF, ID, EX, MEM, WB.

1. IF 从imem中读取IR，更新PC
2. ID 解码，读取寄存器文件，符号拓展立即数
3. EX ALU选择输入执行运算
4. MEM 读/写内存
5. WB 写寄存器文件



Stage	**Any instruction**		

IF	IF/ID.IR←Mem[PC]
	IF/ID.NPC,PC←(if ((EX/MEM.opcode == branch) & EX/MEM.cond){EX/MEM.ALUOutput} else {PC+4});		

ID	ID/EX.A←Regs[IF/ID.IR[rs1]]; ID/EX.B←Regs[IF/ID.IR[rs2]];
	ID/EX.NPC←IF/ID.NPC; ID/EX.IR←IF/ID.IR;
	ID/EX.Imm←sign-extend(IF/ID.IR[immediate field]);		

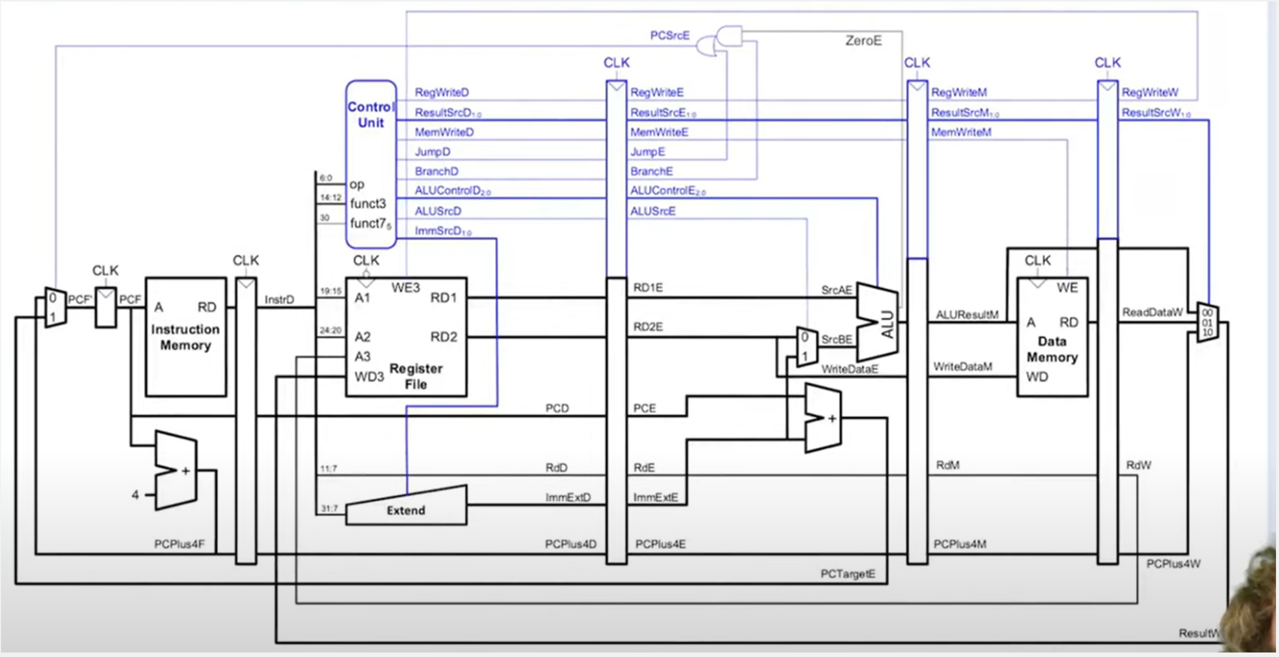
	ALU instruction	**Load instruction**	**Branch instruction**

EX	EX/MEM.IR←ID/EX.IR;
	EX/MEM.ALUOutput←
	ID/EX.A func ID/EX.B;
	or
	EX/MEM.ALUOutput←
	ID/EX.A *op* ID/EX.Imm;
		EX/MEM.IR to ID/EX.IR
		EX/MEM.ALUOutput←
		ID/EX.A+ID/EX.Imm;
		EX/MEM.B←ID/EX.B;
			EX/MEM.ALUOutput←
			ID/EX.NPC +
			(ID/EX.Imm<< 2);
			EX/MEM.cond←
			(ID/EX.A == ID/EX.B);

MEM	MEM/WB.IR←EX/MEM.IR;
	MEM/WB.ALUOutput←
	EX/MEM.ALUOutput;
		MEM/WB.IR←EX/MEM.IR;
		MEM/WB.LMD←
		Mem[EX/MEM.ALUOutput];
		or
		Mem[EX/MEM.ALUOutput]←
		EX/MEM.B;	

WB	Regs[MEM/WB.IR[rd]]← MEM/WB.ALUOutput;	For load only: Regs[MEM/WB.IR[rd]]← MEM/WB.LMD;	

给出流水线处理器的数据通路图和控制信号产生逻辑。不限形式，手绘也可。 （5分）

> 为确保正确性，数据通路图和控制信号产生逻辑均使用了DDCA Ch7的设计
>
> 
>
>
> 控制信号产生具体的布尔逻辑由op，func3,func7共同决定。

请简要描述该流水线中会产生的各种冒险，每类均需举例说明。 （10分）

数据冒险

1. ALU类

```
add x1, x2, x3
sub x4, x1, x5
```

stall两个周期 2. load/use类

```
ld x1, 0(x2)
add x3, x1, x4
```

stall两个周期 3. jalr PC寄存器数据冒险

```
jalr x1, 0(x2)
add x3, x4, x5
```

jalr指令需要stall两个周期，等待ALUOut的结果前递到IF/ID流水寄存器(前递数据通路未在上一问画出)

控制冒险

```
beq x1, x2, target
add x3, x4, x5
```

总是不跳转预测错误需要清空错误预取的2条指令(同时也是数据冒险，因为上一问架构中跳转地址计算在EX阶段完成并前递)

其他stall

```
div x1, x2, x3
add x4, x5, x6
```

ALU进行除法操作时不可流水造成的39个周期的stall.(因为是顺序处理器)

> 当div，rem可共用结果时，rem等同于一般流水指令

```
mul x1, x2, x3
add x4, x1, x5
```

ALU进行64位乘法操作时造成的1个周期的stall.(因为是顺序处理器)

运行测试程序，给出流水线处理器的执行周期数，并计算平均CPI。（共5个测试程序，每个4分）

请对该流水线处理器中因不同类型的冒险而发生的停顿进行统计，并打印数据和分析。（共5个测试程序，每个2分）

采用模拟流水线的方法.

> 其它stall不统计(但是产生效果) 总周期以最后一条指令出流水线为记

运行测试程序的命令为：

```
./driver.sh --perf pipeline -E "lab2"
```

结果

```
Checking Dependencies...
Build Simulator...
Simulator building finished.
Processing: lab2-2-1
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 74
  Dynamic cycles: 156
  CPI: 2.11
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 78

Processing: lab2-2-10
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 64
  Dynamic cycles: 171
  CPI: 2.67
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 64

Processing: lab2-2-2
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 74
  Dynamic cycles: 156
  CPI: 2.11
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 78

Processing: lab2-2-3
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 149
  Dynamic cycles: 322
  CPI: 2.16
  Control Hazard Stall Cycles: 10
  Data Hazard Stall Cycles: 159

Processing: lab2-2-4
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 209
  Dynamic cycles: 462
  CPI: 2.21
  Control Hazard Stall Cycles: 10
  Data Hazard Stall Cycles: 239

Processing: lab2-2-5
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 150
  Dynamic cycles: 520
  CPI: 3.47
  Control Hazard Stall Cycles: 10
  Data Hazard Stall Cycles: 161

Processing: lab2-2-6
Success
```

```
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 110
  Dynamic cycles: 432
  CPI: 3.93
  Control Hazard Stall Cycles: 10
  Data Hazard Stall Cycles: 113

Processing: lab2-2-7
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 179
  Dynamic cycles: 392
  CPI: 2.19
  Control Hazard Stall Cycles: 10
  Data Hazard Stall Cycles: 199

Processing: lab2-2-8
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 106
  Dynamic cycles: 242
  CPI: 2.28
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 132

Processing: lab2-2-9
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 74
  Dynamic cycles: 185
  CPI: 2.50
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 68


Score: 10/10 (filtered: 'lab2')
```

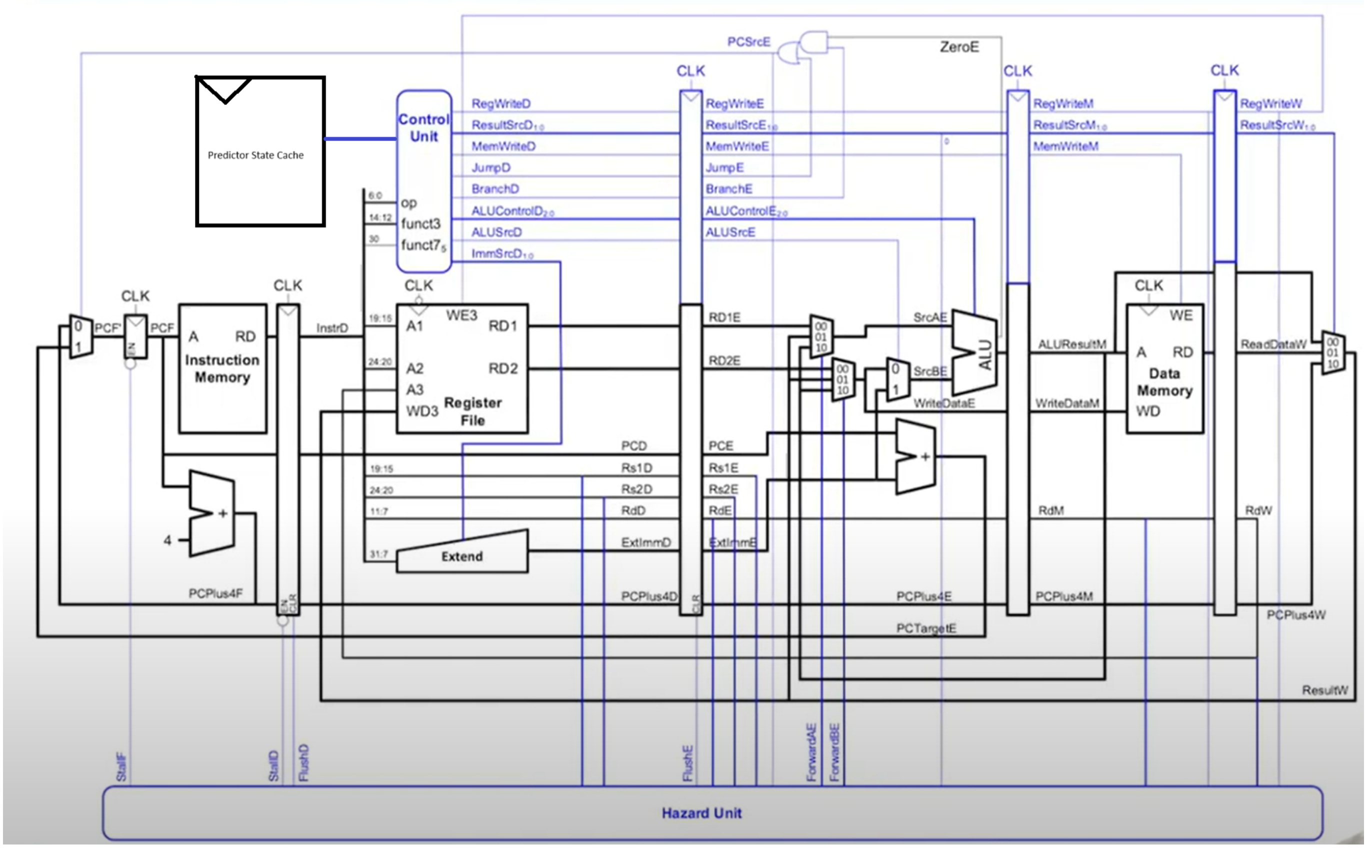
分析: 结果CPI距离理想CPI 1有不少距离。 结果主要由测试程序的性质决定，control stalls基本等于循环数*2, 但是测试程序普遍有较强数据依赖，所以data hazard的stall数目较多。

Part III：其他加分

实现2bit 4096 entry 动态分支预测。

数据通路部分需要加一个预测状态表，由control unit进行读写，control unit增添和状态表关联的对PcSrcE的控制逻辑

> 预测状态表采取PC的13:2位作为索引，直接映射无组相联 没有命中时默认是weak的向低处跳转 同时增加数据的前递逻辑，消除ALU类指令的数据冒险，将load/use类指令的stall数目减少到1个周期,同时将jalr数据前递到IF，减少stall数目到1



运行测试程序的命令为：

```
./driver.sh --perf pipeline_pro -E "lab2"
```

结果

```
Checking Dependencies...
Build Simulator...
Simulator building finished.
Processing: lab2-2-1
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 74
  Dynamic cycles: 87
  CPI: 1.18
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 9

Processing: lab2-2-10
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 64
  Dynamic cycles: 114
  CPI: 1.78
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 7

Processing: lab2-2-2
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 74
  Dynamic cycles: 87
  CPI: 1.18
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 9

Processing: lab2-2-3
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 149
  Dynamic cycles: 184
  CPI: 1.23
  Control Hazard Stall Cycles: 2
  Data Hazard Stall Cycles: 29

Processing: lab2-2-4
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 209
  Dynamic cycles: 254
  CPI: 1.22
  Control Hazard Stall Cycles: 2
  Data Hazard Stall Cycles: 39

Processing: lab2-2-5
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 150
  Dynamic cycles: 380
  CPI: 2.53
  Control Hazard Stall Cycles: 2
  Data Hazard Stall Cycles: 29

Processing: lab2-2-6
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 110
  Dynamic cycles: 330
  CPI: 3.00
  Control Hazard Stall Cycles: 2
  Data Hazard Stall Cycles: 19

Processing: lab2-2-7
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 179
  Dynamic cycles: 219
  CPI: 1.22
  Control Hazard Stall Cycles: 2
  Data Hazard Stall Cycles: 34

Processing: lab2-2-8
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 106
  Dynamic cycles: 122
  CPI: 1.15
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 12

Processing: lab2-2-9
Success
Performance Metrics:
  Performance Profiler: Pipeline
  Dynamic instructions: 74
  Dynamic cycles: 125
  CPI: 1.69
  Control Hazard Stall Cycles: 0
  Data Hazard Stall Cycles: 8


Score: 10/10 (filtered: 'lab2')
```

分析: 数据前递和分支预测的实现使得CPI大幅降低