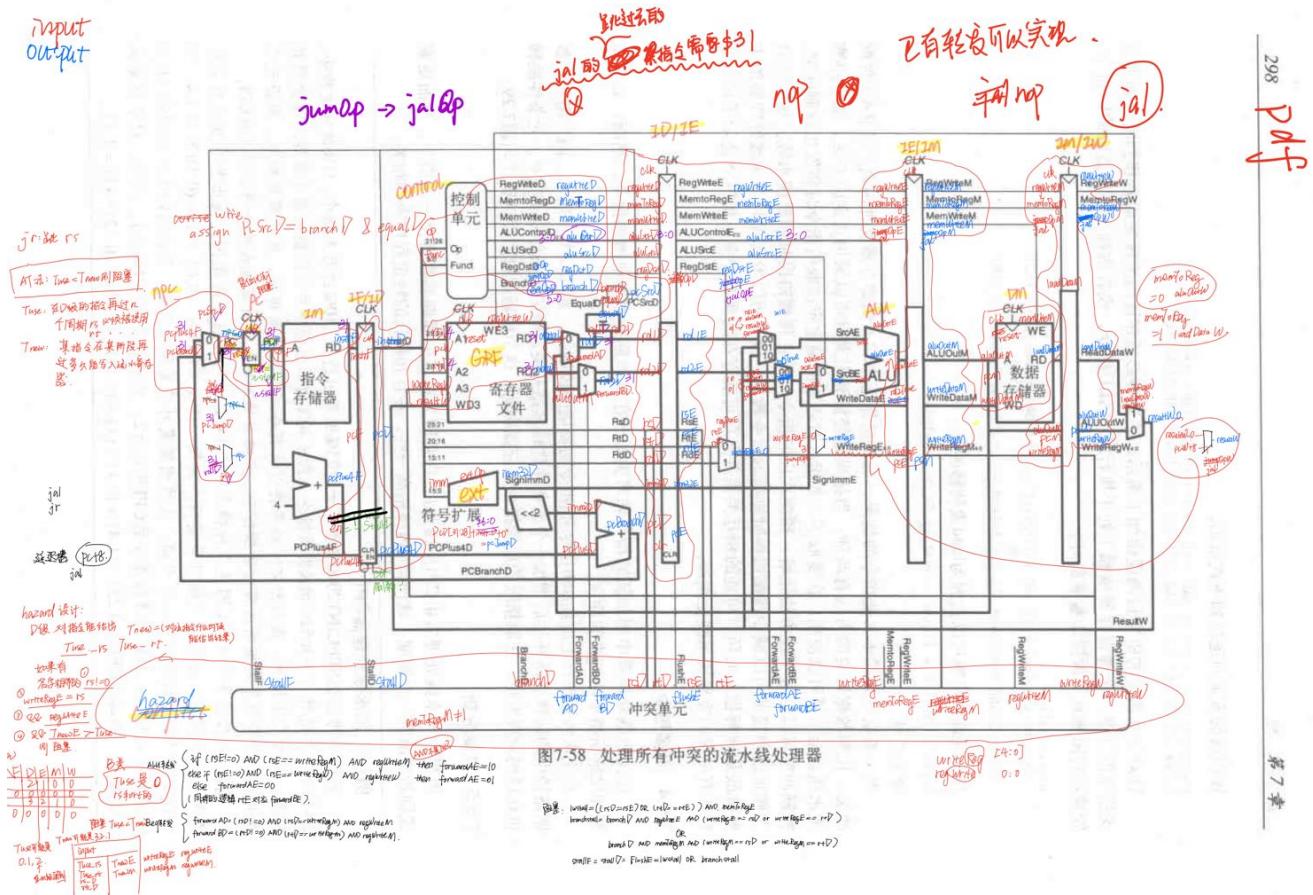


一.设计图



二.流水线重点

1.阻塞

<1>Tuse 【1: 0】

当这条指令处于D级的时候，该寄存器的值还有多久会被需要

	add	sub	ori	lw	sw	beq	lui	jal	jr	nop
Tuse_rsD	1	1	1	1	1	0	3	3	0	3
Tuse_rtD	1	1	3	3	2	0	3	3	3	3

当某寄存器的值不被需要的时候置为可能的最大值3，这样就不会被stall

<2>Tnew 【1: 0】

当这条指令处于某个级的时候，再经过多少个时钟周期结果可以被存进流水寄存器

	add	sub	ori	lw	sw	beq	lui	jal	jr	nop
Tnew_D	2	2	2	3	0	0	2	2	0	0
Tnew_E	1	1	1	2	0	0	1	1	0	0
Tnew_M	0	0	0	1	0	0	0	0	0	0

Tnew被流水的策略是，在不为0的时候会减1，为0的时候会传递0

一些不需要结果的置为0，这样就不会被stall

<3>比较Tnew (rs, rt) 和Tuse (E级, M级) 从而stall

```
stall = 0;
if((rsD != 0)&&(rsD == writeRegE)&&(regWriteE === 1)&&(Tuse_rsD < TnewE)) stall = 1;
else if((rsD != 0)&&(rsD == writeRegM)&&(regWriteM === 1)&&(Tuse_rsD < TnewM))stall = 1;
if((rtD != 0)&&(rtD == writeRegE)&&(regWriteE === 1)&&(Tuse_rtD < TnewE)) stall = 1;
else if((rtD != 0)&&(rtD == writeRegM)&&(regWriteM === 1)&&(Tuse_rtD < TnewM))stall = 1;
```

以rs和E级比较为例 (rs和M比较, rt和E级比较, rt和M级比较同理)

1. rs != 0
2. rsD == writeRegE
3. regWriteE === 1
4. Tuse_rsD < TnewE

2.转发

<1>GRF内部转发

```
always@(*)begin
    if(regWriteW === 1 && writeRegW == rsD && rsD != 0) data1 = resultW;
    else data1 = regs[rsD];
    if(regWriteW === 1 && writeRegW == rtD && rtD != 0) data2 = resultW;
    else data2 = regs[rtD];
end
```

注意所写的寄存器不为0以及写信号为1

<2>D级读出转发

```
forwardAD = (rsD != 0)&&(rsD == writeRegM)&&(regWriteM);
forwardBD = (rtD != 0)&&(rtD == writeRegM)&&(regWriteM);
```

只转发M级 (因为W级的会通过GRF内部转发)

<3>E级转发改变read1和read2的值

```

if((rsE != 0)&&(rsE == writeRegM)&&(regWriteM)&&(memToRegM != 1)) forwardAE = 2'b10; //forward m first
else if ((rsE != 0)&&(rsE == writeRegW)&&(regWriteW)) forwardAE = 2'b01;
else forwardAE = 0;

if((rtE != 0)&&(rtE == writeRegM)&&(regWriteM)&&(memToRegM != 1)) forwardBE = 2'b10;
else if ((rtE != 0)&&(rtE == writeRegW)&&(regWriteW)) forwardBE = 2'b01;
else forwardBE = 0;

```

先转发M级，如果M级不能转发再转发W级

<4> M级转发

```

assign forwardM = (rtM != 0)&(rtM == writeRegW)&(regWriteW);
assign writeM = (forwardM) ? resultW : writeDataM;

```

只转发W级的值，改变写入DM的值（这一部分是在mips主程序里面加入的）

思考题

1. 我们使用提前分支判断的方法尽早产生结果来减少因不确定而带来的开销，但实际上这种方法并非总能提高效率，请从流水线冒险的角度思考其原因并给出一个指令序列的例子。

提前到D级判断，可能数据还没有写入寄存器，因此需要阻塞，在这种情况下会降低效率。

2. 因为延迟槽的存在，对于 jal 等需要将指令地址写入寄存器的指令，要写回 PC + 8，请思考为什么这样设计？

因为下一条指令（pc+4）由于延迟槽的存在一定会执行，所以如果要跳回的话应该执行再下一条指令（pc+8）

3. 我们要求所有转发数据都来源于流水寄存器而不能是功能部件（如 DM、ALU），请思考为什么？

因为要使得数据可以稳定地达到正确，所以要确定某一时钟周期（最小时钟周期由各部件的最大延迟决定）

4. 我们为什么要使用 GPR 内部转发？该如何实现？

为了应对W写的是D读的数据的情况，因为有这种可能，所以必须要转发来读出正确数据，实现的方式参见上面转发部分的图

5. 我们转发时数据的需求者和供给者可能来源于哪些位置？共有哪些转发数据通路？

见上面转发部分

6. 在课上测试时，我们需要你现场实现新的指令，对于这些新的指令，你可能需要在原有的数据通路上做哪些扩展或修改？提示：你可以对指令进行分类，思考每一类指令可能修改或扩展哪些位置。

可能修改ALU加入新的计算，可能修改存取指令（在存数部分和取数部分通过加入多路选择器来实现load和save指令的变形）

7. 简要描述你的译码器架构，并思考该架构的优势以及不足。

集中式译码，优势是只要译码一次节省控制器资源，不足是需要一直流水控制信号

8. [P5 选做] 请详细描述你的测试方案及测试数据构造策略。

9. [P5、P6 选做] 请评估我们给出的覆盖率分析模型的合理性，如有更好的方案，可一并提出