P5 流水线CPU

5级流水线：

阶段	功能概述
取指 （F）	从IM中读取指令
译码 （D）	从GRF中读取源操作数并对指令译码得到信号
执行(E)	用ALU计算
存储（M）	读写DM
回写（W）	回写GRF

涉及的指令

add,sub,ori,lui,sw,lw,beq,jal,jr;

数据通路

与单周期CPU类似，通过添加寄存器存储中间值来实现流水线。

译码方式

采取集中式译码

冒险

冒险的种类

结构冒险，控制冒险和数据冒险，本质是需求与供给的不匹配。

结构冒险

指不同指令同时需要使用同一资源。

控制冒险

分支指令的判断结果会影响接下来指令的执行。

数据冒险

后面的指令使用结果尚未确定的数据。

解决冒险

阻塞

当发生数据依赖时只让前一条指令执行，而后一条指令被阻塞在流水线的某个阶段，等待前一条指令执行完成，在接触阻塞。发生冒险的指令阻塞在D流水级上。

提前分支判断

将分支判断提前到D级，来减少作废的指令。

延迟槽

转发

将后面的计算结果传送到前面的流水级的需求者来使用。（能处理部分数据冒险）

A模型（Address）

T模型

需求时间-供给时间模型。

定义

T_{use}为这条指令位于D级时再过多少个时钟周期就必须要使用相应的数据。（固定值，可以有两个）

T_{new} 某个流水级的某个指令，当它经过多少个时钟周期可以算出结果并且存储到流水级寄存器里。（动态值，最多有1个）

当 $T_{use} \geq T_{new}$ 时，说明可以通过转发实现。

反之，则必须使用阻塞流水线。

数据通路照旧，添加流水线寄存器。

控制部分分为原先的控制和新添加的ATCONTROL。

CONTROL需要计算D级指令的 T_{new} 和 T_{rt_use} ，T_{rs_use}

ATCONTROL需要计算阻塞信号，刷新信号，转发信号

阻塞信号的判定条件：

$T_{use} < T_{new}$ 且读和写寄存器相同且不为0，（M和W级都必须满足）

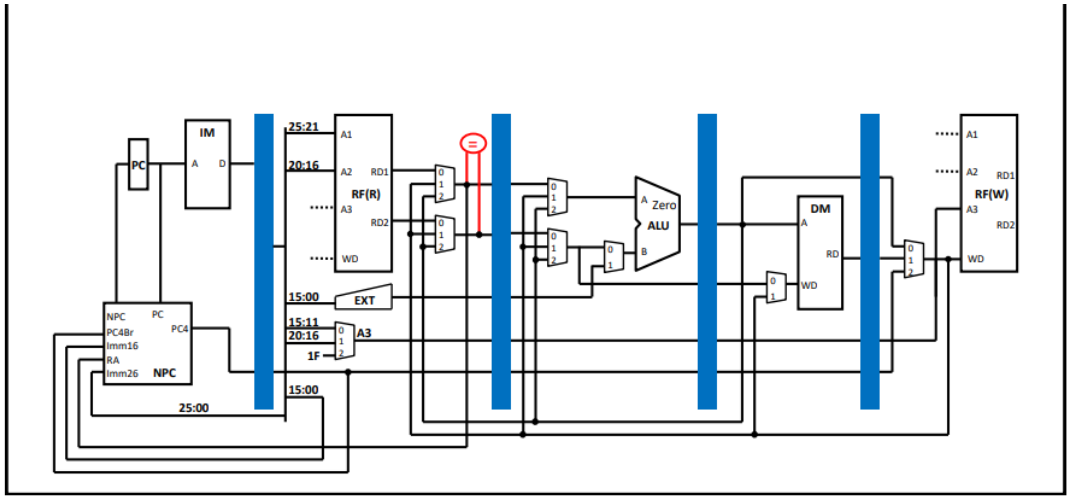
阻塞信号的功能：

1. 冻结PC
2. 冻结D级流水线寄存器的值
3. 将E级流水线寄存器清零

转发信号的判定条件:

$T_{use} \geq T_{new}$ 且读和写寄存器相同且不为0 （M和W级先后一个满足）

转发信号包括：发出者选择器信号，接收者选择器信号（发出仅限于M和W级）



指令	T_rsUse	T_rtUse		T_dnew	T_enew	T_mnew	T_wnew
add	1	1		2	1	0	0
sub	1	1		2	1	0	0
lw	1	-		3	2	1	0
sw	1	2		0	0	0	0
ori	1	-		2	1	0	0
lui	1	-		2	1	0	0
beq	0	0		0	0	0	0
jal	-	-		0	0	0	0
jr	0	-		0	0	0	0

指令	T_rsUse	T_rtUse		T_dnew	T_enew	T_mnew	T_wnew
add	1	1		2	1	0	0
sub	1	1		2	1	0	0
lw	1	-		3	2	1	0
sw	1	2		0	0	0	0
ori	1	-		2	1	0	0
lui	1	-		2	1	0	0
beq	0	0		0	0	0	0
jal	-	-		0	0	0	0
jr	0	-		0	0	0	0

	regE			regM		regW
Trs	0	1	2	0	1	0
0	F	S	S	F	S	F
1	F	F	S	F	F	F

	regE			regM		regW
Trt	0	1	2	0	1	0
0	F	S	S	F	S	F
1	F	F	S	F	F	F
2	F	F	F	F	F	F

思考题：

2.延迟槽会插入一条不相关的指令来提高效率，如果写回PC + 4,

会回到延迟槽的指令而不是真正的下一条指令

4.这样可以防止冲突，直接转发。

5.数据的供给者来自于M和W端，需求来自D,E,M端，

6.指令分为R I J 型指令，指令修改需要考虑转发和暂停,可能需要修改控制器和AT控制器，转发的多路选择器和相应的功能部件。

7.我的译码器采用集中式译码，这种译码器可以减少后续控制流水线寄存器的复杂度，但是会增加寄存器需要存储的值。

测试代码见code