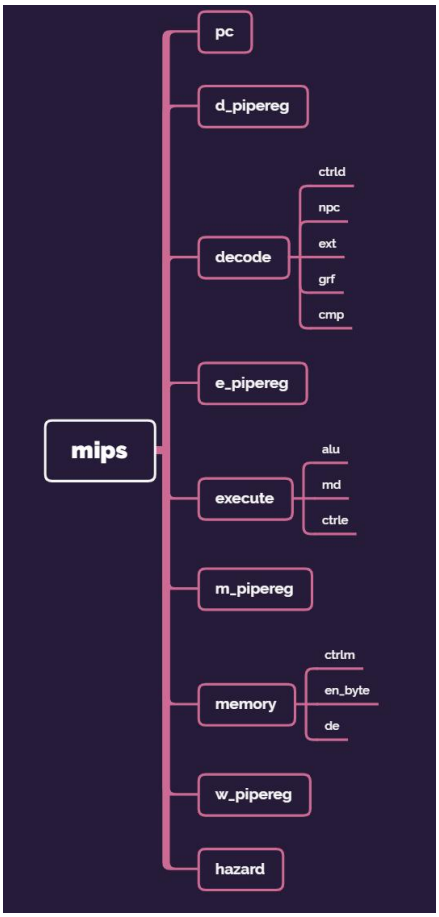
本 CPU 为 Verilog 实现的流水线 MIPS - CPU，支持的指令集包含{LB、LBU、LH、LHU、LW、SB、SH、SW、ADD、ADDU、 SUB、 SUBU、 MULT、 MULTU、DIV、 DIVU、 SLL、 SRL、 SRA、 SLLV、 SRLV、SRAV、AND、OR、XOR、NOR、ADDI、ADDIU、ANDI、ORI、 XORI、LUI、SLT、SLTI、SLTIU、SLTU、BEQ、BNE、BLEZ、BGTZ、 BLTZ、BGEZ、J、JAL、JALR、JR、MFHI、MFLO、MTHI、MTLO}。为了实现这些功能，CPU 主要包含了 IM、GRF 等模块，这些模块按照以下顶层设计逐级展开:



（二）关键模块定义

1.PC

（1） 端口说明

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	同步置零

3	NPC	I	下一条指令在 IM 的地址
4	en	I	阻塞信号
5	PC	O	当前指令在 IM 的地址

（2）功能定义

序号	功能	描述
1	存储指令地址	保存当前执行指令在 IM 中的地址
2	阻塞	阻塞为 0 时，PC 不工作

2. D_pipereg

（1）端口说明

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	同步复位信号
3	en	I	阻塞信号
4	F_PC	I	F 级的 PC
5	F_instr	I	F 级的指令
6	D_PC	O	D 级的 PC
7	D_instr	O	D 级的指令
8	F_BD	I	F 级产生的判断是否使用延迟槽
9	F_ExcCode	I	F 级异常码
10	D_BD	O	D 级产生的判断是否使用延迟槽
11	D_ExcCode	O	传到 D 级的异常码

（2）功能定义

序号	功能	描述
1	存储要流水的值	存储要流水的值
2	流水异常需要用的值	流水异常需要用的值

3. NPC

（1）端口说明

序号	信号名	方向	描述
1	PCSrc	I	分支信号
2	PCj	I	j 型跳转信号
3	PCjr	I	jr 和 jalr 跳转信号
4	D_PC[31:0]	I	F 级 PC
5	F_PC[31:0]	I	F 级 PC
6	Regjr[31:0]	I	jr 和 jalr 跳转的地址
7	instr_index[25:0]	I	D 级 26 位立即数
8	NPC[31:0]	O	下一个 PC 的值

（2）功能定义

序号	功能	描述
1	计算下一个 PC	计算下一个 PC

4. EXT

（1）端口说明

序号	信号名	方向	描述
1	imm	I	16 位立即数
2	sign	I	符号扩展选择信号 0：进行无符号扩展 1：进行符号扩展
3	SignImm	O	扩展后的 32 位数

（2）功能定义

序号	功能	描述
1	无符号扩展	当 sign 为 0 时，进行无符号扩展
2	符号扩展	当 sign 为 1 时，进行有符号扩展

5. GRF

（1）端口说明

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	异步复位信号， 1：清零 0：保持
3	PC[31:0]	I	D 级 PC 的值，用于 display
4	A1[4:0]	I	5 位地址输入，从外界接收[25：21]，代表 32 个寄存器中的一个，并将其中的值输出到 RD1
5	A2[4:0]	I	5 位地址输入，从外界接收[20：16]，代表 32 个寄存器中的一个，并将其中的值输出到 RD2
6	A3[4:0]	I	5 位地址输入，选择 32 个寄存器中的一个，将 WD 输入的数据存储到其中
7	WD[31:0]	I	32 位写入数据
8	RD1[31:0]	O	输出 A1 代表的寄存器中的值
9	RD2[31:0]	O	输出 A2 代表的寄存器中的值

（2）功能定义

序号	功能	描述
1	异步复位	Reset 为 1 时，所有寄存器被清零
2	读数据	将 A1 和 A2 对应的寄存器中的值输出到 RD1 和 RD2
3	写数据	如果 WE 为 1，时钟上升沿时将 WD 中的数据写入到 A3 对应的寄存器中

6. CMP

（1）端口说明

序号	信号名	方向	描述
1	A	I	操作数 A
2	B	I	操作数 B
3	D_equal	O	相等信号
4	D_equal_0	O	是否等于 0
5	D_great_0	O	是否大于 0

（2）功能定义

序号	功能	描述
1	判断 A、B 是否相等	若 A=B，则输出 1
2	判断 A 是否等于 0	若 A=0，则输出 1
3	判断 A 是否大于 0	若 A>0，则输出 1

7. E_pipereg

（1）端口说明

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	同步复位信号
3	en	I	阻塞信号
4	flush	I	清洗信号
5	D_PC[31:0]	I	D 级 PC 值
6	D_instr[31:0]	I	D 级指令
7	D_signimm[31:0]	I	D 级扩展后 32 位数
8	D_RD1[31:0]	I	D 级 GRF 输出 1
9	D_RD2[31:0]	I	D 级 GRF 输出 2
10	D_A3[4:0]	I	D 级 A3 地址
11	D_WD[31:0]	I	D 级 GRF 写入值
12	E_PC[31:0]	O	E 级 PC 值

13	E_instr[31:0]	O	E 级指令
14	E_signimm[31:0]	O	E 级扩展后 32 位数
15	E_RD1[31:0]	O	E 级 RD1
16	E_RD2[31:0]	O	E 级 RD2
17	E_A3[4:0]	O	E 级 A3 地址
18	E_WD[31:0]	O	E 级 WD
19	D_BD	I	D 级 BD
20	D_ExcCode	I	D 级异常码
21	E_BD	O	E 级 BD
22	E_ExcCode	O	E 级异常码

（2）功能定义

序号	功能	描述
1	存储要流水的值	存储要流水的值
2	流水异常需要的变量	流水异常需要的变量

8. ALU

（1）端口说明

序号	信号名	方向	描述
1	op[2:0]	I	选择执行哪一个操作 000：加法 001：减法 010：与 011：或 100：左移 16 位
2	inA[31:0]	I	参与运算的第一个数
3	inB[31:0]	I	参与运算的第二个数
4	ALUResult[31:0]	O	输出 inA 和 inB 操作后的结果

（2）功能定义

序号	功能	描述

1	加法	将两个输入加起来输出到 ALUResult
2	减法	将两个输入相减输出到 ALUResult
3	与运算	对两个输入进行与操作输出结果到 ALUResult
4	或运算	对两个输入进行或操作输出结果到 ALUResult
5	左移 16 位	将 SrcB 左移 16 位输出到 ALUResult
6	或非	将两个输入的或非输出到 ALUResult
7	异或	将两个输入的异或输出到 ALUResult
8	逻辑左移	将两个输入的逻辑左移输出到 ALUResult
9	逻辑右移	将两个输入的逻辑右移输出到 ALUResult
10	算术右移	将两个输入的算术右移输出到 ALUResult
11	小于立即数置 1（有符号）	若 A 小于 B，输出置 1
12	小于立即数置 1（无符号）	若 A 小于 B，输出置 1

9. M_pipereg

（1）端口说明

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	同步复位信号
3	flush	I	清洗信号
4	E_PC[31:0]	I	E 级 PC
5	E_WD[31:0]	I	E 级 WD
6	E_A3[4:0]	I	E 级 A3
7	E_instr[31:0]	I	E 级 instr
8	E_ALUResult[31:0]	I	E 级 ALUResult
9	E_RD2[31:0]	I	E 级 RD2
10	M_PC[31:0]	O	M 级 pc
11	M_WD[31:0]	O	M 级 WD

12	M_A3[4:0]	O	M 级 A3
13	M_instr[31:0]	O	M 级 instr
14	M_ALUResult[31:0]	O	M 级 ALUResult
15	M_RD2[31:0]	O	M 级 RD2
16	E_BD	I	E 级 BD
17	E_ExcCode	I	E 级异常码
18	M_BD	O	M 级 BD
19	M_ExcCode	O	M 级异常码

（2）功能定义

序号	功能	描述
1	存储要流水的值	存储要流水的值
2	流水异常需要的变量	流水异常需要的变量

10. en_byte

（1）端口说明

序号	信号名	方向	描述
1	reset	I	同步复位信号
2	Din	I	带扩展的数据
3	WE	I	写入信号
4	DMaddr	I	DM 的读取地址
5	Dout	O	扩展后的数据
6	width	I	是半字还说字节存取
7	m_data_byteen	O	要求输出的信号

（2）功能定义

序号	功能	描述
1	整字	整个完整输出
2	半字	按半字规则输出
3	字节	按字节读取规则输出

11. data_extend

（1）端口说明

序号	信号名	方向	描述
1	DMaddr	I	ALU 计算出来的地址
2	dataop	I	数据扩展控制码 000：无扩展 001：无符号字节数据扩展 010：符号字节数据扩展 011：无符号半字数据扩展 1000：符号半字数据扩展
3	Din	I	输入 32 位数据
4	Dout	O	扩展后的 32 位数据

（2）功能定义

序号	功能	描述
1	无扩展	无扩展
2	无符号字节数据扩展	无符号字节数据扩展
3	符号字节数据扩展	符号字节数据扩展
4	无符号半字数据扩展	无符号半字数据扩展
5	符号半字数据扩展	符号半字数据扩展

12. W_pipereg

（1）端口说明

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	同步复位信号
3	M_PC[31:0]	I	M 级 PC
4	M_instr[31:0]	I	M 级 instr
5	M_A3[4:0]	I	M 级 A3
6	M_WD[31:0]	I	M 级 WD

7	W_PC[31:0]	O	W 级 PC
8	W_instr[31:0]	O	W 级 instr
9	W_A3[4:0]	O	W 级 A3
10	W_WD[31:0]	O	W 级 WD

（2）功能定义

序号	功能	描述
1	存储要流水的值	存储要流水的值

13. CP0

（1）端口说明

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	同步复位信号
3	WE	I	CP0 寄存器写使能信号
4	[4:0]A1	I	读 CP0 寄存器编号
5	[4:0]A2	I	写 CP0 寄存器编号
6	bd	I	看是否用延迟槽
7	[31:0]Din	I	CP0 寄存器的写入数据
8	[31:0]PC	I	中断/异常时的 PC
9	[6:2]ExcCode	I	中断/异常的类型
10	[5:0]HWint	I	6 个设备中断
11	EXLclr	I	置 0 SR 的 EXL 位
12	IntReq	O	中断请求
13	[31:0] EPC	O	EPC 寄存器输出至 NPC
14	[31:0] Dout	O	CP0 寄存器的输出数据
15	tbReq	O	tb 中断

（2）功能定义

序号	功能	描述

1	实现对异常、中断的控制	实现对异常、中断的控制

14. Hazard

序号	信号名	方向	描述
1	clk	I	时钟信号
2	reset	I	同步复位信号
3	D_A1[4:0]	I	D 级读取 rs 的序号
4	D_A2[4:0]	I	D 级读取 rt 的序号
5	D_RD1[31:0]	I	D 级读取 rs 的值
6	D_RD2[31:0]	I	D 级读取 rt 的值
7	D_rs_Tuse	I	D 级是否在使用 rs
8	D_rt_Tuse	I	D 级是否在使用 rt
9	E_A1[4:0]	I	E 级读取 rs 的序号
10	E_A2[4:0]	I	E 级读取 rt 的序号
11	E_RD1[31:0]	I	E 级读取 rs 的值
12	E_RD2[31:0]	I	E 级读取 rt 的值
13	E_A3[4:0]	I	E 级写入寄存器的序号
14	E_WD[31:0]	I	E 级写入寄存器的值
15	E_rs_Tuse	I	E 级是否使用 rs
16	E_rt_Tuse	I	E 级是否使用 rt
17	M_A2[4:0]	I	M 级读取 rt 的序号
18	M_RD2[31:0]	I	M 级读取到寄存器的值
19	M_A3[4:0]	I	M 级写入寄存器的序号
20	M_WD[31:0]	I	M 级写入寄存器的值
21	W_A3[4:0]	I	W 级写入寄存器的序号
22	W_WD[31:0]	I	W 级写入寄存器的值
23	D_Forward1[31:0]	O	转发给 D 级的值 1
24	D_Forward2[31:0]	O	转发给 D 级的值 2

25	E_Forward1[31:0]	O	转发给 E 级的值 1
26	E_Forward2[31:0]	O	转发给 E 级的值 2
27	M_Forward2[31:0]	O	转发给 M 级的值 2
28	F_en	O	PC 的使能信号
29	D_pipereg_en	O	D 级流水线寄存器的使能信号
30	E_pipereg_en	O	E 级流水线寄存器的使能信号
31	E_pipereg_flush	O	E 级流水线寄存器的清洗信号
32	M_pipereg_flush	O	M 级流水线寄存器的清洗信号

（2）功能定义

序号	功能	描述
1	产生各转发值	产生各转发值
2	产生各阻塞信号	产生各阻塞信号
3	产生各清洗信号	产生各清洗信号

15. Control（分布式译码）

（1）端口说明

序号	信号名	方向	描述
1	instr[31:0]	I	每一级的指令
2	D_equal	I	D 级 RD1 和 RD2 是否相等
3	sign	O	ext 是否进行有符号扩展
4	PCSrc	O	是否进行分支
5	PCj	O	是否为 J 型指令
6	PCjr	O	是否为 jr 或 jalr
7	D_WDsel	O	D 级中写入 GRF 值的选择信号
8	D_A3[4:0]	O	D 级中写入 GRF 地址的选择信号
9	ALUControl[3:0]	O	ALU 选择哪种计算方式
10	ALUASel	O	ALU 里面 inA 选择哪个作为输入
11	ALUBSel	O	ALU 里面 inB 选择哪个作为输入

12	E_WDSel[1:0]	O	E 级中写入 GRF 值的选择信号
13	MemWrite	O	E 级中写入 GRF 地址的选择信号
14	width[1:0]	O	写入 DM 的位宽
15	sign_1	O	写入 DM 的值是否进行有符号扩展
16	M_WDSel	O	M 级中写入 GRF 值的选择信号
17	D_rs_Tuse	O	D 级是否读取 rs
18	D_rt_Tuse	O	D 级是否读取 rt
19	E_rs_Tuse	O	E 级是否读取 rs
20	E_rt_Tuse	O	E 级是否读取 rt
21	MD_yes	O	是否使用乘除模块
22	BD	O	是不是 branch 类指令
23	excRI	O	是不是已有的指令
24	start	O	乘除法要开始信号
25	[2:0] MDop	O	是乘法还是除法
26	HIwrite	O	是 mthi 吗
27	LOwrite	O	是 mtlo 吗
28	ALU_arith_overflow	O	是否为计算类指令
29	ALU_lw_overflow	O	load 类指令
30	ALU_sw_overflow	O	save 类指令
31	eret	O	返回现场
32	CP0WE	O	CP0 是否写入

（2）真值表

端口	addu	subu	ori	lw	sw	beq	lui
func	100001	100011	n/a	n/a	n/a	n/a	n/a
op	000000	000000	001101	100011	101011	000100	001111
sign	x	x	0	1	1	x	x
Branch	0	0	0	0	0	1	0
MemWrite	0	0	0	0	1	0	0
RegWrite	1	1	1	1	0	0	1

MemtoReg	0	0	0	1	x	x	0
ALUSrc	0	0	1	1	1	0	1
RegDst	1	1	0	0	x	x	0
ALUControl	000	001	011	000	000	001	100

（三）重要机制实现方法

1. 跳转

CMP 模块在 D 级生成跳转信号

2. 流水线延迟槽

保证在 b 指令和 j 指令来临时，一定会执行后一条指令，根据判断结果，再进行跳转

3. 转发

在 hazard 统一生成，再传到各个流水线级中

4. 进入中断处理程序

用 en 和 flush 进行终端和清洗

二、测试方案

（一）测试代码

见附件

（二）测试方案

1. 检验增加指令后 p6 部分是否仍正确

2. 检验新增的 eret、mtc0、mfc0 行为是否正确，以及它们的阻塞转发是否正确

3. 对所有可能的异常进行测试，看行为是否正确

4. 对所有可能的中断进行测试，看行为是否正确

5. 对 IO 操作进行测试，对 timer 进行读写操作

6. 一些特殊情况，如 eret 前使用乘除模块、使用乘除模块前中断

三、思考题

（一）我们计组课程一本参考书目标题中有“硬件/软件接口”接口字样，那么到底什么是“硬件/软件接口”？(Tips：什么是接口？和我们到现在为止所学的有什么联系？)

中断异常和软件运行在硬件和软件中的约定，起到连接硬件和软件的作用

（二）BE 部件对所有的外设都是必要的吗？

不是，比如 timer 模块需要整字存取，异常中就专门有 sb、sh、lb、lh 对 timer 操作会报异常的。

（三）请开发一个主程序以及定时器的 exception handler。整个系统完成如下功能：

（1）定时器在主程序中被初始化为模式 0；

（2）定时器倒计数至 0 产生中断；

（3）handler 设置使能 Enable 为 1 从而再次启动定时器的计数器。(2) 及 (3) 被无限重复。

（4）主程序在初始化时将定时器初始化为模式 0，设定初值寄存器的初值为某个值，如 100 或 1000。(注意，主程序可能需要涉及对 CP0.SR 的编程，推荐阅读过后文后再进行。)

```
.ktext 0x4180
## $30 记录异常次数
## $29 记录中断次数
## $28 调整EPC
## $27 取Cause的BD判断
## $26 取Cause的ExcCode
## $25 取SR
## 规定延迟槽异常为$10导致的异常（如add $10,$10,$10 溢出），db_hanlder将$10清0后返回，$30加1
## 规定非延迟槽指令异常，则exc_handler将EPC对齐后EPC+4跳过该指令，$30加1
## 规定中断处理 EPC不改变 返回原指令，$29加1
mfc0 $28,$14
mfc0 $27,$13
mfc0 $26,$13
mfc0 $25,$12
sll $26,$26,25
srl $26,$26,27 ## ExcCode（通过移位消除IP和BD的影响）
srl $27,$27,31 ## BD
beq $26,$0,interrupt_handler ## ExcCode=0 中断处理
nop
bgtz $27,db_handler ## BD=1
nop
## 延迟槽指令异常同时被外部中断，优先处理外部中断
exc_handler:
andi $28,$28,0xfffffffc ## 字对齐
addi $28,$28,4 ## EPC+4
mtc0 $28,$14
addi $30,$30,1 ## 记录异常次数
```

```
1  ## 测试各类指令异常
2  li $2,0x3011 ## test1的PC：0x3010
3  jr $2        ## PC字未对齐(非受害指令)
```

（四）请查阅相关资料，说明鼠标和键盘的输入信号是如何被 CPU 知晓的？

鼠标点击、位置移动或按下键盘时，将产生中断信号，从而使 CPU 进入相应的中断处理程序。