VerilogHDL 开发 MIPS 微系统

一、模块定义

1. pc

（1） 基本描述

PC 主要功能是完成输出当前指令地址并保存下一条指令地址。复位后，PC 指向 0x0000_3000，此处为第一条指令的地址。

（2）模块接口

表 1- 1 pc 模块接口

信号名	方向	描述
Addr_I [31:2]	I	所输入的 30 位 NPC。
clk_I	I	时钟信号。
rst_I	I	复位信号。 1：复位 0：无效
PC_en_I	I	PC 的使能信号。 1：有效 0：无效
PC_O[31:2]	O	将输入的 NPC 输出。

（3）功能定义

表 1- 2 pc 功能定义

序号	功能名称	功能描述
1	复位	当复位信号有效时，PC 被设置为 0x0000_3000。
2	保存 NPC 并输出	在每个 clk 的上升沿，若使能信号为高电平时，保存 NPC 并输出。

2. npc

（1） 基本描述

NPC 负责计算下一条 PC。

（2） 模块接口

表 2- 1 npc 模块接口

信号名	方向	描述
pc_I [31:2]	I	目前执行的 PC。
PCSrc_I [1:0]	I	控制 NPC 的信号。
RD1_I [31:0]	I	GPR 的 A 寄存器所保存的数据。
Shift_I [31:0]	I	分支指令的目标偏移。
Jumpadd_I [25:0]	I	J 型指令中的跳转地址。
EPC_I [31:2]	I	CP0 中 EPC 所存储的地址。
Zero_I	I	标志 ALU 计算结果是否为零，分支指令所用。
npc_O [31:2]	O	输出下一条指令 npc。

（3） 功能定义

表 2- 2 npc 功能定义

序号	功能名称	功能描述
1	计算下一条指令	根据控制信号 PCSrc_I 计算 NPC 并输出。 2'b0：执行正常的 PC 递增 2'b1：jr 指令对应的 GPR 的 A 寄存器中的数据赋给 NPC 2'b10：bne 以外的分支指令对应的根据是否 Zero_I 为高电平进行跳转 ； 2'b11：j 和 jal 指令对应的直接跳转； 2’b100：bne 指令对应的根据是否 Zero_I 为低电

		平进行跳转 ； 2’b101: 进入 ISR； 2’b110: eret 指令对应的恢复 PC。

3. gpr

（1） 基本描述

gpr 是由 32 个具有写使能的 32 位寄存器组成的寄存器堆，主要功能是储存数据通路中的参数、变量、数据、指针等。

（2） 模块接口

表 3- 1 gpr 模块接口

信号名	方向	描述
A1_I [31:0]	I	输入 GPR 的寄存器编号 1。
A2_I [31:0]	I	输入 GPR 的寄存器编号 2。
clk_I	I	时钟信号
clr_I	I	复位信号。 1：复位 0：无效
WD_I [31:0]	I	写入 GPR 的数据。
Wreg_I [4:0]	I	要写入的寄存器的编号。
RegWrite_I	I	控制信号。 1：允许数据写入 GPR 0：不允许数据写入 GPR
RD1_O [31:0]	O	对应 A1 寄存器编号的寄存器中的数据。
RD2_O [31:0]	O	对应 A2 寄存器编号的寄存器中的数据。

（3） 功能定义

表 3- 2 gpr 功能定义

序号	功能名称	功能描述
1	复位	当复位信号有效时，32 个寄存器被设置为 0x00000000。
2	写入数据	通过 3 个信号（clk、RegWrite、Wreg）控制下的 Wreg 所指的寄存器会在时钟下降沿将数据 WD 写入。
3	同时读取 2 个数据	根据 A1、A2 可以分别同时输出他们对应的寄存器中的数据。

4. alu

（1） 基本描述

ALU 由加法器、减法器、移位器、比较器以及逻辑电路组成，功能是对输入的两个数根据控制信号传递的信号进行运算并输出。

（2） 模块接口

表 4- 1 alu 模块接口

信号名	方向	描述
A_I [31:0]	I	输入 GPR 的寄存器编号 1。
B_I [31:0]	I	输入 GPR 的寄存器编号 2。
ALUOperation_I [3:0]	I	根据控制信号 ALUOp[3:0]产生，控制运算的种类。
Zero_O	O	ALU 计算结果为 0 标志。 1：计算结果为 0 0：计算结果非 0
Result_O [31:0]	O	输出运算结果。

（3） 功能定义

表 4- 2 alu 功能定义

序号	功能名称	功能描述
1	运算	根据 ALUoperation [3:0]，分别支持 16 种运算。
2	结果判零	根据结果是否为 0 进行分支指令的控制。

5. ext

（1） 基本描述

EXT 为扩展单元，可以把 16 位立即数扩展为 32 位参与运算。

（2） 模块接口

表 5- 1 ext 模块接口

信号名	方向	描述
imm16_I [15:0]	I	输入需要被扩展的 16 位立即数。
op_I [31:26]	I	由指令种类决定符号扩展还是零扩展。slti、addi、addiu、sltiu 以及分支指令目前进行符号扩展，其余进行零扩展。
Signext_O [31:0]	O	将 16 位 imm16_I 扩展为 32 位并输出。

（3） 功能定义

表 5- 2 ext 功能定义

序号	功能名称	功能描述
1	16-32 位符号扩展	把 16 位立即数符号扩展为 32 位参与运算。
2	16-32 位零扩展	把 16 位立即数零扩展为 32 位参与运算。

6. dm_12k

（1） 基本描述

DM 即 Data Memory，主存储器，本质为一个 3072X32 位 RAM。

（2） 模块接口

表 6- 1 dm_4k 模块接口

信号名	方向	描述
addr [16:2]	I	输入需要操作的 DM 中的地址。
din [31:0]	I	输入需要写入的数据。
we	I	控制信号。为高电平时，DM 执行写操作。
be	I	控制信号，受不同 save 指令影响。
clk	I	时钟信号。
clr	I	复位信号。 1：复位 0：无效
dout [31:0]	O	输出根据 addr [16:2]所读的 DM 中的数据。

（3） 功能定义

表 6- 2 dm_4k 功能定义

序号	功能名称	功能描述
1	读操作	当 re 信号为 1 时，在时钟上升沿可以根据输入的地址读出 DM 中相应的数据。
2	写操作	当 we 信号为 1 时，在时钟上升沿可以根据输入的地址把输入的 din [31:0]写入相应的位置。

7. im_8k

（1） 基本描述

IM 负责存储指令。

（2） 模块接口

表 7- 1 im_4k 模块接口

信号名	方向	描述
addr [21:2]	I	输入需要读取的 IM 中的地址。

dout [31:0]	O	输出根据 addr [21:2]所读的指令。

（3） 功能定义

表 7- 2 im_4k 功能定义

序号	功能名称	功能描述
1	加载指令	加载 code.txt 中的指令到 im 中。
2	读指令	读出 addr [21:2]对应位置的指令。

8. mux

（1） 基本描述

4 个多路选择器。

（2） 模块接口

表 8- 1 MUX_RegDst 模块接口

信号名	方向	描述
RegDst	I	控制信号。
Ir [31:0]	I	指令寄存器所传入的指令。
Wreg [4:0]	O	控制写入 GPR 的地址来源。

表 8- 2 MUX_MemtoReg 模块接口

信号名	方向	描述
MemtoReg	I	控制信号。
Aluout [31:0]	I	ALUout 寄存器所存的运算结果。
Dr [31:0]	I	数据寄存器所存的来自于 DM 的数据。
High [31:0]	I	乘除模块 HI 寄存器所存的运算结果。
Low [31:0]	I	乘除模块 LO 寄存器所存的运算结果。
PrDIn	I	Bridge 向 mips 处理器的输出。

[31:0]		
CP0 [31:0]	I	CP0 的输出。
Wdata [31:0]	O	控制写入 GPR 的数据来源。

表 8- 3 MUX_ALUSrcA 模块接口

信号名	方向	描述
ALUSrcA [1:0]	I	控制信号。
GPR_A [31:0]	I	当前 GPR_A 寄存器存数据。
Pc [31:2]	I	当前 PC。
Ir [31:0]	I	指令寄存器中的指令。
A [31:0]	O	控制 ALU 的 A 端数据来源。

表 8- 4 MUX_ALUSrcB 模块接口

信号名	方向	描述
ALUSrcB [1:0]	I	控制信号。
EXT [31:0]	I	16 位立即数经过 32 位扩展后的结果。
GPR_B [31:0]	I	当前 GPR_B 寄存器存数据。
B [31:0]	O	控制 ALU 的 B 端数据来源。

（3） 功能定义

表 8- 5 mux 功能定义

序号	功能名称	功能描述
1	控制 Wreg	RegDst

		2’b0：Rt 2’b1：Rd 2’b10：31
2	控制 Wdata	MemtoReg 3’b0：ALU 运算结果 3’b1：DR 中的数据 3’b10: HI 中的结果 3’b11: LO 中的结果 3’b100: Bridge 的输出 3’b101: CP0 的输出
3	控制 ALU 的 A 端	ALUSrcA 2’b0：GPR_A 2’b1：PC 2’b1：指令中 10 到 6 位，即“sa”
4	控制 ALU 的 B 端	ALUSrcB 2’b0：32 位扩展的立即数 2’b1：GPR_B 2’b10：0 2’b11：32 位扩展的立即数左移 2 位

9. dmout_ext

（1）基本描述

此模块的作用是将 DM 所读出的数据根据指令所要求的读取类型对数据进行改动，结果将存入 DR 中。

（2）模块接口

表 9- 1 dmout_ext 模块接口

信号名	方向	描述
op [5:0]	I	指令的操作码。
dout [31:0]	I	DM 所读出的数据。

ALUout [1:0]	I	ALU 运算所得地址的低 2 位，用来控制存储对应的半字或字节。
ext_dout [31:0]	O	输出给 DR 的数据。

（3）功能定义

表 9- 2 dmout_ext 功能定义

序号	功能名称	功能描述
1	数据符号扩展	从 DM 读出数据中取出特定半字或字节进行扩展。

10. ir、gpr_a、gpr_b、aluout、dr

（1） 基本描述

多周期处理器所增加的 5 个寄存器，用于在各状态转换时锁存相应的数据。其中 ir 有写入使能，收到 IRWrite 信号控制。这 5 个寄存器都是在时钟上升沿到来时锁存输入的数据。

11. mult_div_unit

（1） 基本描述

乘除法模块。使用了时序逻辑。

（2） 模块接口

表 11- 1 mult_div_unit 的模块接口

信号名	方向	描述
clk_I	I	时钟信号。
MDCtrl_I	I	高电平时此模块才允许工作。
A_I [31:0]	I	被乘（除）数。
B_I [31:0]	I	乘（除）数。
op_I [2:0]	I	决定 4 种运算的种类。
MT_I [1:0]	I	决定 move to 的位置是 HI 还是 LO。
memtoreg [2:0]	I	控制信号。
Wdata_I	I	写入的数据。

[31:0]		
HI_O [31:0]	O	HI 寄存器的结果。
LO_O [31:0]	O	LO 寄存器的结果。
Ready_O	O	运算完毕后会产生高电平，决定状态机的运转。

（3） 功能定义

表 11- 2mult_div_unit 的功能定义

序号	功能名称	功能描述
1	有/无符号乘除法	通过迭代法进行乘除法的计算。

12. Control

（1） 基本描述

Control 由两部分组成，其一是 Control Unit，其二是专门负责 ALU 信号的 ALU Control。通过输入的 32 位 Instruction 中 6 位 Op 以及 R-Instruction 的 6 位 Function，可以生成各指令需要产生的信号。

（2） 模块接口

表 12- 1 control 模块接口

模块	信号名	方向	描述
control_unit	op_I [31:26]	I	指令的操作码 Op。
	Func_I [5:0]	I	指令的 Function 码。
	clk_I	I	时钟信号。
	rst_I	I	复位信号。 1：复位 0：无效
	ALUout_I [1:0]	I	ALUout 寄存器保存的结果，用来控制 be 信号。
	Zero_I	I	ALU 结果是否为零将影响分支指令的进行。

	Ir_20_16_I [4:0]	I	指令的 20 至 16 位。
	Ir_25_21_I [4:0]	I	指令的 25 至 21 位。
	IntReq_I	I	Timer 产生的中断信号。
	Ready	I	乘除模块产生的信号。
	RegDst_O [1:0]	O	控制写入 GPR 数据的地址。
	PCSrc_O [2:0]	O	控制 NPC 的信号。
	MemtoReg_O [2:0]	O	控制写入 GPR 的数据来源。
	RegWrite_O	O	GPR 的写使能。高电平为允许写入。
	be [3:0]	O	控制存入 DM 数据的半字/字节。
	MemWrite_O	O	DM 的写使能。高电平为允许写入。
	ALUSrcA_O [1:0]	O	ALU 的 A 端来源。
	ALUSrcB_O [1:0]	O	ALU 的 B 端来源。
	PCWrite_O	O	PC 的写使能。高电平时允许写入。
	IRWrite_O	O	IR 的写使能。高电平时允许写入。
	MDOp_O [2:0]	O	给乘除模块的运算种类识别信号。
	MT_O [1:0]	O	给乘除模块的表明写入 HO 还是 LO 寄存器的信号。
	MDCTRL_O	O	乘除模块启动信号。
	TimerWrite_O	O	Timer 的写使能。
	EPC_Wen_O	O	EPC 的写使能。
	EXLSet_O	O	exl 位置 1。
	EXLClr_O	O	exl 位清 0。
	ALUOp_O [3:0]	O	仅在涉及 ALU 计算的状态发生改变，控制 ALU 的运算类型。
ALU_ctrl	ALUOp_I [3:0]	I	control_unit 模块所输出的控制信号。

	Func_I [5:0]	I	R 型指令中的 Function 码。
	ALUoperation_O [3:0]	O	输出 4 位 ALUoperation 到 ALU 指导其进行 16 种类运算。

（3） 功能定义

表 12- 2 control 功能定义

序号	功能名称	功能描述
1	FSM 的运作	通过 FSM 实现多周期处理指令。
2	产生各组件控制信号	对应 FSM 的每个状态将产生不同的信号。
3	产生 ALU 信号	涉及 ALU 运算时，可以控制运算的类型。

13. CP0

（1） 基本描述

0 号协处理器。可以处理最低中断，这里仅涉及 12——15 号寄存器的部分位。其中 15 号寄存器里所存储的是 0x19940819，作为特殊标记。

（2） 模块接口

表 13- 1 CP0 模块接口

信号名	方向	描述
clk	I	时钟信号。
rst	I	复位信号。
PC [31:2]	I	当前的 PC，用于写入 EPC。
DIn [31:0]	I	写入 CP0 的数据。
HWInt [5:0]	I	6 个中断位，这里仅 HWInt[2]有效。
Sel [5:0]	I	决定写入寄存器的信号。
Wen	I	CP0 写使能。
EXLSet	I	EXL 置 1 信号。
EXLClr	I	EXL 清 0 信号。
IntReq	O	输出给控制器的中断信号，进入 ISR 用。

EPC [31:2]	O	输出 EPC，eret 还原用。
DOut [31:0]	O	输出 CP0 的数据。

（3） 功能定义

表 13- 2 CP0 功能定义

序号	功能名称	功能描述
1	控制进入 ISR	根据 Timer 的中断信号进入中断处理程序。

14. Bridge

（1） 基本描述

位于 MIPS 处理器与 Timer 设备之间的桥。

（2） 模块接口

表 14- 1 Bridge 模块接口

信号名	方向	描述
PrAddr_I [31:2]	I	处理器传入的写入地址。
PrWD_I [31:0]	I	处理器传入的写入数据。
DEV_RD_I [31:0]	I	设备传入的读出数据。
BE_I [3:0]	I	控制器给的 be 信号。
PrRD_O [31:0]	O	输出给处理器的读出数据。
DEV_Addr_O [1:0]	O	输出给设备的写入地址。
DEV_WD_O [31:0]	O	输出给设备的写入数据。

（3） 功能定义

表 14- 2 Bridge 功能定义

序号	功能名称	功能描述
1	数据桥	在 MIPS 处理器与设备之间进行数据的传输。

15. Timer

完全基于《COCO 定时器设计规范》设计。

二、测试

1. 测试代码

```
#详见 Project8.asm。
ori $t0, 0x7F00
ori $t1, 9
ori $t2, 1000
ori $t3, 0xABCD1234
sw $t2, 4($t0)
sw $t1, 0($t0)
sw $t3, 0($zero)
li $t0, 0x401
mtc0 $t0, $12
Loop:
beq $zero, 0, Loop

.ktext 0x00004180
add $sp, $sp, -16
sw $t0, 0($sp)
sw $t1, 4($sp)
sw $t2, 8($sp)
```

```
sw $t3, 12($sp)
li $t3, 0x7F00
li $t1, 800
sw $t1, 4($t3)
lw $t0, 0($sp)
lw $t1, 4($sp)
lw $t2, 8($sp)
lw $t3, 12($sp)
eret
```

2. 测试结果

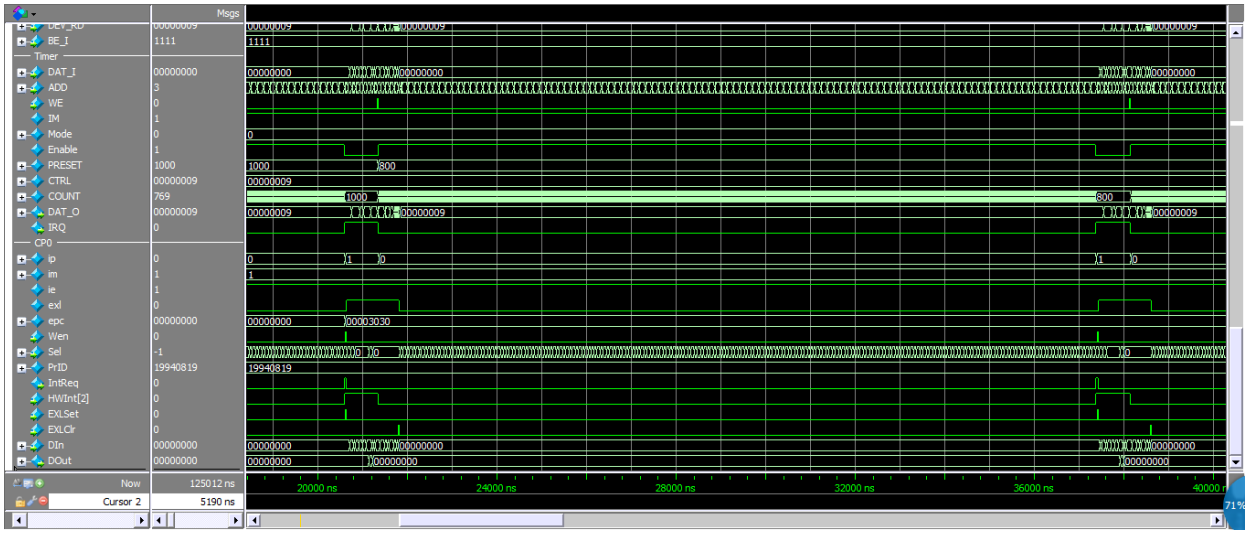

图 1 测试结果

3. 测试原理

开始时为计时器赋初值 1000 以及 control 值（保证 0 模式、中断有效）。之后为 CP0 赋初值（保证 im 最低位为 1，支持最低中断、中断有效位置 1）。然后开始死循环等待计时器到 0，产生中断。

终端产生后进入 ISR，ISR 中先通过进栈保存现场，再将新的初值 800 写入 Timer 后启动 Timer，之后跳出 ISR 回到死循环，等待下一次 800 计时结束的中断产生，以此无限循环下去。

已附上 wave.do。testbench_mips 为主测试模块，请模拟此模块并加载 wave.do。

testbench_timer 和对应的 do 文件只是为测试 Timer 所用。

三、 问答

第 23 项定义的初值寄存器初值不能低于多少？为什么？

答：初值不能低于主程序进入死循环之前所做准备工作指令的总时间，否则未设置好 CP0 的初值时迎来中断信号，将不能正常进入 ISR。