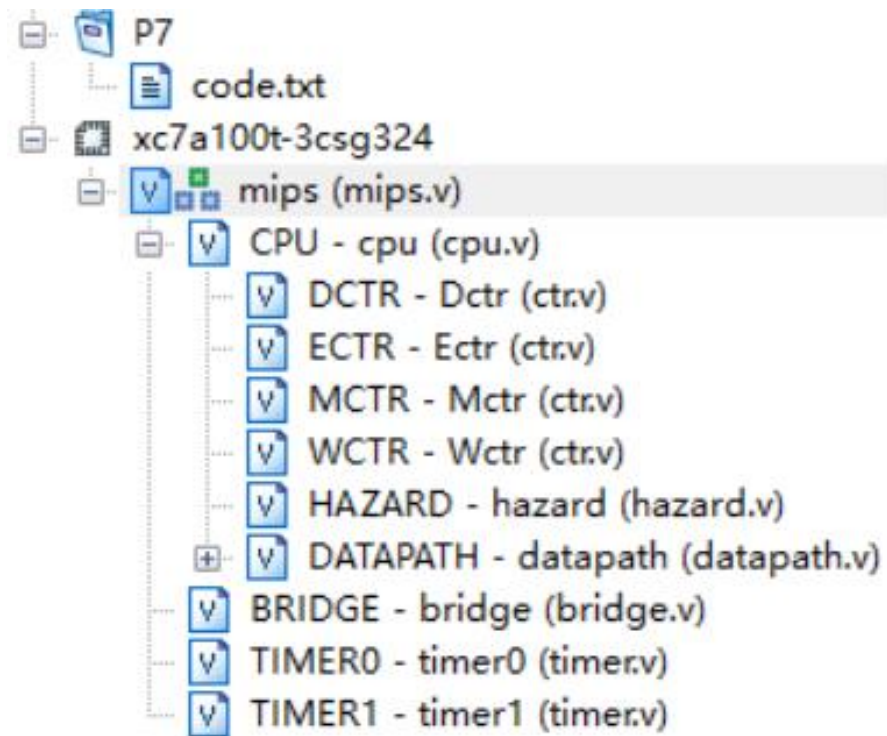


计算机组成原理实验报告

一、CPU 设计文档

(一) 总体设计



图表 1 模块设计

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
1			MUX	控制信号	0	1	2		lw	sw	addu/addsubu/subori		lui	beq	j	jal	jr	jalr	lb	lbu	lh	lhu	sb	sh	sll	
2	PC		M_nPC	PCSel	ADD4	NPC	MFPCF		ADD4	ADD4	ADD4	ADD4	ADD4	ADD4	ADD4/NPC	NPC	NPC	RF.V1	RF.V1	ADD4	ADD4	ADD4	ADD4	ADD4	ADD4	ADD4
3	ADD4				PC				PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC
4	IM				PC				PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC	PC
5	D级寄存器	R_D			IM				IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM
6		PC8_D			ADD4+4								sh+J2J3				ADD4+4		ADD4+4							
7		PC4_D			ADD4										ADD4			ADD4								
8	RF	A1			R_D[rs]				R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]			R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]	R_D[rs]
9		A2			R_D[rt]					R_D[rt]	R_D[rt]	R_D[rt]	R_D[rt]		R_D[rt]										R_D[rt]	R_D[rt]
10	EXT				R_D[16]				R_D[16]	R_D[16]				R_D[16]	R_D[16]					R_D[16]	R_D[16]	R_D[16]	R_D[16]	R_D[16]	R_D[16]	R_D[16]
11	CMP	D1			MFcmp1D										RF.V1											
12		D2			MFcmp2D										RF.V2											
13	NPC	I26			R_D[126]										R_D[116]	R_D[126]	R_D[126]									
14	E级寄存器	R_E			R_D				R_D	R_D	R_D	R_D	R_D	R_D			R_D		R_D	R_D	R_D	R_D	R_D	R_D	R_D	R_D
15		PC8_E			PC8_D												PC8_D		PC8_D							
16		PC4_E			PC4_D												PC4_D		PC4_D							
17		V1_E			D1				RF.V1	RF.V1	RF.V1	RF.V1	RF.V1						RF.V1	RF.V1	RF.V1	RF.V1	RF.V1	RF.V1	RF.V1	RF.V1
18		V2_E			D2				RF.V2	RF.V2	RF.V2	RF.V2													RF.V2	RF.V2
19		E32_E			EXT				EXT	EXT		EXT	EXT							EXT	EXT	EXT	EXT	EXT	EXT	EXT
20	ALU	A	M_ALUB	ASel	MFaluaE	R[10:6]	MFaluaE[4:0]		V1_E	V1_E	V1_E	V1_E	V1_E							V1_E	V1_E	V1_E	V1_E	V1_E	V1_E	R[10:6]
21		B	M_ALUB	BSel	MFalubE	E32_E			E32_E	E32_E	V2_E	V2_E	E32_E	E32_E						E32_E	E32_E	E32_E	E32_E	E32_E	E32_E	V2_E
22	M级寄存器	R_M							R_E	R_E	R_E	R_E	R_E	R_E			R_E		R_E	R_E	R_E	R_E	R_E	R_E	R_E	R_E
23		PC8_M			PC8_E												PC8_E		PC8_E							
24		PC4_M			PC4_E												PC4_E		PC4_E							
25		AO_M			ALU				ALU	ALU	ALU	ALU	ALU	ALU						ALU	ALU	ALU	ALU	ALU	ALU	ALU
26		V2_M			MFalubE					V2_E														V2_E	V2_E	
27	DM	A			AO_M				AO_M	AO_M										AO_M	AO_M	AO_M	AO_M	AO_M	AO_M	AO_M
28		WD			MFdmdataM					V2_M														V2_M	V2_M	
29	W级寄存器	R_W			R_M				R_M	R_M	R_M	R_M	R_M	R_M			R_M		R_M	R_M	R_M	R_M	R_M	R_M	R_M	R_M
30		PC8_W			PC8_M												PC8_M		PC8_M							
31		PC4_W			PC4_M												PC4_M		PC4_M							
32		AO_W			AO_M					AO_M	AO_M	AO_M	AO_M	AO_M												AO_M
33		DR_W			DM				DM											DM	DM	DM	DM			
34	DMEXT																									
35	RF	A3	M_WReg	WRSel	R_W[rt]	R_W[rd]	0x1f		R_W[rt]		R_W[rd]	R_W[rd]	R_W[rt]	R_W[rt]			0x1f		R_W[rd]	R_W[rt]	R_W[rt]	R_W[rt]	R_W[rt]	R_W[rt]		R_W[rt]
36		WD	M_Wdata	WDSel	AQ_W	DR_W	PC8_W		DR_W		AO_W	AO_W	AO_W	AO_W			PC8_W		PC8_W	DR_W	DR_W	DR_W	DR_W			AO_W

图表 2 数据通路设计

Tuse	rs	rt	指令	功能部件	E	M	W
addu	1	1	addu	ALU	1	0	0
subu	1	1	subu	ALU	1	0	0
ori	1		ori	ALU	1	0	0
lui			lui	ALU	1	0	0
lw	1		lw	DM	2	1	0
sw	1	2	sw				
beq	0	0	beq				
jr	0		jr				
jalr	0		jalr	PC	0	0	0
j			j				
jal			jal	PC	0	0	0

转发MUX	控制信号	输入0	输入1	输入2	输入3	输入4
MFcmp1D	Fcmp1D	RF.V1	M_Wdata	AO_M	PC8_M	PC8_E
MFcmp2D	Fcmp2D	RF.V2	M_Wdata	AO_M	PC8_M	PC8_E
MFaluaE	FaluaE	V1_E	M_Wdata	AO_M	PC8_M	
MFalubE	FalubE	V2_E	M_Wdata	AO_M	PC8_M	
MFdmdataM	FdmdataM	V2_M	M_Wdata			
MFPCF	FPCF	RF.V1	M_Wdata	AO_M	PC8_M	PC8_E

rs	E	M	W
ALU	DM	PC	ALU
DM	PC	ALU	DM
PC	ALU	DM	PC
ALU	DM	PC	ALU
DM	PC	ALU	DM
PC	ALU	DM	PC

Tuse	E	M	W
0	S	S	F
1	F	S	F

rt	E	M	W
ALU	DM	PC	ALU
DM	PC	ALU	DM
PC	ALU	DM	PC
ALU	DM	PC	ALU
DM	PC	ALU	DM
PC	ALU	DM	PC

Tuse	E	M	W
0	S	S	F
1	F	S	F
2	F	F	F

图表 3 转发暂停设计

图表 4 控制器设计

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	func	100001	100011	100000	100010	001000	001001								
2	op	000000	000000	000000	000000	000000	000000	001101	100011	101011	000100	001111	001110	101001	
3		addu	subu	add	sub	jr	jalr	ori	lw	sw	beq	lui	xori	sh	
4	RegDst	1	1	1	1	0	1	0	0	0	0	0	0	0	0 0rt 1rd 进rf2
5	Asel	00	00	00	00	00	00	00	00	00	00	00	00	00	0grfddata1 1lR[10:6] 2MfAluaE[4:0] 进aluA
6	Bsel	0	0	0	0	0	0	1	1	1	0	1	1	1	1 0grfddata2 1i32 进aluB
7	MemtoReg	0	0	0	0	0	0	0	1	0	0	0	0	0	
8	RegWrite	1	1	1	1	0	1	1	1	0	0	1	1	0	
9	MemWrite	0	0	0	0	0	0	0	0	1	0	0	0	1	
10	EXTop	00	00	00	00	00	00	10	10	11	01	00	10		0无符号 1低16位补0 2有符号 3有符号后左移两位
11	ALUctr	0000	0001	0000	0001	0000	0000	0010	0000	0000	0000	0011	0100	0000	0000加 0001减 0010或 0011B 0100异或 0101逻辑左移 0110逻辑右移 0111算数右移 1000与
12	dmextop	000	000	000	000	000	000	000	000	000	000	000	000	000	000不扩展 001无符号字节扩展 010有符号字节扩展 011无符号半字扩展 100有符号半字扩展
13															
14															
15	func	000000	000000	000000	000000	000000	000000	000011	000010	100000	100100	100001	100101	101000	
16	op	000000	000010	000011	000100	000110	000111	jal	j	lb	lbu	lh	lhu	sb	
17		sll	srl	sra	slv	slv	sra		0	0	0	0	0	0	0
18	RegDst	1	1	1	1	1	1	00	00	00	00	00	00	00	
19	Asel	01	01	01	10	10	10	0	0	1	1	1	1	1	
20	Bsel	0	0	0	0	0	0	0	0	1	1	1	1	0	
21	MemtoReg	0	0	0	0	0	0	1	0	1	1	1	1	0	
22	RegWrite	1	1	1	1	1	1	0	0	0	0	0	0	1	
23	MemWrite	0	0	0	0	0	0	00	00	10	10	10	10	10	
24	EXTop	00	00	00	00	00	00	0000	0000	0000	0000	0000	0000	0000	
25	ALUctr	0101	0110	0111	0101	0110	0111	000	000	010	001	100	011	000	
26	dmextop	000	000	000	000	000	000								
27															
28															
29															
30															
31															
32															

（二）数据通路设计

1. datapath（数据通路）

1) 端口说明

表格 1datapath 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1：复位 0：无效
PCsel[1:0]	I	D 控制器发来选择 nPC 信号 10：转发器 MFPCF 结果 01：NPC 00：ADD4
i16[15:0]	I	D 控制器发来 16 位立即数
i26[25:0]	I	D 控制器发来 26 位立即数
RegWrite	I	W 控制器发来写寄存器信号 1：写寄存器 0：无效
MemWrite	I	M 控制器发来写内存信号 1：写内存 0：无效
EXTop[2:0]	I	D 控制器发来扩展器信号 000：无符号扩展 001：低 16 位补 0 010：有符号扩展 011：有符号扩展后逻辑左移两位
ALUctr[2:0]	I	E 控制器发来 ALU 控制信号 000：加运算 001：减运算 010：或运算 011：输出写入数据 2 100：异或运算
WRsel[1:0]	I	D 控制器发来选择寄存器 A3 端口信号 10:31 号寄存器 01：IR_W[rd] 00：IR_W[rt]

WDsel[1:0]	I	W 控制器发来选择寄存器输入信号 00: AO_W01: DR_W10: PC8_W
Fcmp1D[2:0]	I	冒险单元发来选择 MFcmp1D 信号 000: RF.V1 001: M_Wdata 010: AO_M 011: PC8_M 100: PC8_E
Fcmp2D[2:0]	I	冒险单元发来选择 MFcmp2D 信号 000: RF.V2 001: M_Wdata 010: AO_M 011: PC8_M 100: PC8_E
FdmdataM	I	冒险单元发来选择 MFdmdataM 信号 0: V2_M 1: M_Wdata
FaluaE[1:0]	I	冒险单元发来选择 MFaluaE 信号 00: V1_E 01: M_Wdata 10: AO_M 11: PC8_M
FalubE[1:0]	I	冒险单元发来选择 MFalubE 信号 00: V2_E 01: M_Wdata 10: AO_M 11: PC8_M
FPCF[2:0]	I	冒险单元发来选择 MFPCF 信号 000: RF.V1 001: M_Wdata 010: AO_M 011: PC8_M 100: PC8_E
Bsel	I	E 控制器发来 ALUb 选择信号 0: MFalubE 选择结果 1: E32_E
stall	I	冒险单元发来选择暂停信号 1: 暂停 0: 无效
beq	I	D 控制器发来 beq 识别信号 1: beq 0: 无效
IRF[31:0]	0	输出到 D 控制器的 F 级指令
IRD[31:0]	0	输出到 E 控制器的 D 级指令
IRE[31:0]	0	输出到 M 控制器的 E 级指令
IRM[31:0]	0	输出到 W 控制器的 M 级指令
RESE[1:0]	0	输出到冒险单元的 E 级 Tnew 状态 00: NW 不写 01: 写 ALU 10: 写 DM 11: 写 PC
RESM[1:0]	0	输出到冒险单元的 M 级 Tnew 状态 00: NW 不写 01: 写 ALU 10: 写 DM 11: 写 PC
RESW[1:0]	0	输出到冒险单元的 W 级 Tnew 状态 00: NW 不写 01: 写 ALU 10: 写 DM 11: 写 PC
RFA3E[4:0]	0	输出到冒险单元的 E 级指令 A3 寄存器
RFA3M[4:0]	0	输出到冒险单元的 M 级指令 A3 寄存器
RFA3W[4:0]	0	输出到冒险单元的 W 级指令 A3 寄存器

2) 功能定义

表格 2datapath 功能定义

序号	功能名称	功能描述
1	连接基本模块	通过 datapath，以声明中间变量和实例化引用的方式连接各基础模块

2. ifu（取指令单元）

1) 端口说明

表格 3ifu 端口说明

信号名	方向	描述
reset	I	复位信号 1：复位 0：无效
clk	I	时钟信号
PCsel[1:0]	I	D 控制器发来选择 nPC 信号 10：转发器 MFPCF 结果 01：NPC 00：ADD4
C0	I	cmp 发来比较信号 1：alu 两输入相等 0：alu 两输入不等
i16 [15:0]	I	D 控制器发来 16 位立即数
i26[25:0]	I	D 控制器发来 26 位立即数
PCtempD[31:0]	I	MFPCF 转发多选器的结果
beq	I	D 控制器发来 beq 识别信号 1：beq 0：无效
stall	I	冒险单元发来选择暂停信号 1：暂停 0：无效
instruction[31:0]	O	输出的指令
PC8[31:0]	O	输出的当前 PC+8

2) 功能定义

表格 4 ifu 功能定义

序号	功能名称	功能描述
1	复位	复位信号有效时，PC 被置为 0x00000000
2	取指令	根据 PC 从 IM 中取出指令
3	计算下条指令地址	$PC \leftarrow PC+4 \quad \quad PC \leftarrow reg1data \quad \quad PC \leftarrow PC + 4 + immed32 \quad $ $PC \leftarrow \{PC[31:28], immed26, 2'b0\}$
4	暂停	Stall 信号有效时，冻结 PC 寄存器

3. rf（寄存器堆）

1) 端口说明

表格 5grf 端口说明

信号名	方向	描述
reset	I	复位信号 1：复位 0：无效
clk	I	时钟信号
reg1 [4:0]	I	读寄存器号 1 编号
Reg2 [4:0]	I	读寄存器号 2 编号
writereg[4:0]	I	写寄存器编号
regwrite	I	写控制信号 1：写入 0：无效
writedata[31:0]	I	写入的 32 位数据
data1[31:0]	O	32 位寄存器 1 输出
data2[31:0]	O	32 位寄存器 2 输出

2) 功能定义

表格 6grf 功能定义

序号	功能名称	功能描述
1	复位	复位信号有效时，32 个寄存器被置为 0x00000000
2	写寄存器	写寄存器控制信号有效时，把 32 位数据写入寄存器
3	读寄存器	根据输入的地址读出两个寄存器中的值

4. alu（算术逻辑单元）

1) 端口说明

表格 7alu 端口说明

信号名	方向	描述
A[31:0]	I	32 位写入数据 1
B[31:0]	I	32 位写入数据 2
ALUOp[2:0]	I	控制信号 000：加运算 001：减运算 010：或运算 011：输出写入数据 2 100：

		异或运算
A0[31:0]	0	32 位输出数据

2) 功能定义

表格 8alu 功能定义

序号	功能名称	功能描述
1	加运算	$A+B$
2	减运算	$A-B$
3	或运算	$A B$
4	输出写入数据 2	B
5	异或运算	A^B

5. dm（数据存储器）

1) 端口说明

表格 9dm 端口说明

信号名	方向	描述
D1[31:0]	I	32 位输入数据 1
D2[31:0]	I	32 位输入数据 2
CO	0	比较结果 0：不相等 1：相等

2) 功能定义

表格 10dm 功能定义

序号	功能名称	功能描述
1	比较	比较两输入数据大小，相等输出 1，否则输出 0

6. cmp（比较器）

1) 端口说明

表格 11dm 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1：复位 0：无效
ADDR [31:0]	I	32 位写入内存地址
din[31:0]	I	32 位写入数据
PC8[31:0]	I	当前 PC+8
MemWrite	I	写内存控制信号 1：写入 0：无效
dout[31:0]	O	32 位输出数据

2) 功能定义

表格 12dm 功能定义

序号	功能名称	功能描述
1	复位	复位信号有效时，内存和读出内存的寄存器被置为 0x00000000
2	写内存	写内存控制信号有效时，根据输入的地址写入 32 位数据
3	读内存	根据输入的地址读出内存数据

7. ext（位扩展器）

1) 端口说明

表格 13ext 端口说明

信号名	方向	描述
immed16[15:0]	I	16 位写入立即数
EXTop[2:0]	I	扩展控制信号 000：无符号扩展 001：低 16 位补 0 010：有符号扩展 011：有符号扩展后逻辑左移两位
E0[31:0]	O	32 位输出立即数

2) 功能定义

表格 14ext 功能定义

序号	功能名称	功能描述
1	无符号扩展	高 16 位补 0
2	低 16 位补 0	低 16 位补 0
3	有符号扩展	Immed[15]为 1 时高 16 位补 1，为 0 时高 16 位补 0
4	有符号扩展后 逻辑左移两位	Immed[15]为 1 时高 16 位补 1，为 0 时高 16 位补 0，再左移两位，溢出舍去，低 2 位补 0

8. mux（多路选择器）

1) 端口说明

表格 15mux 端口说明

信号名	方向	描述
A[4:0]	I	5 位输入 A
B[4:0]	I	5 位输入 B
C[4:0]	I	5 位输入 C
Op[1:0]	I	选择控制信号 10：输出 C 01：输出 B 00：输出 A
O [4:0]	O	5 位输出 0
A[31:0]	I	32 位输入 A
B[31:0]	I	32 位输入 B
C[31:0]	I	32 位输入 C
D[31:0]	I	32 位输入 D
E[31:0]	I	32 位输入 E
op[2:0]	I	选择控制信号 100：输出 E 011：输出 D 010：输出 C 001：输出 B 000：输出 A
O[31:0]	O	32 位输出 0

2) 功能定义

表格 16mux 功能定义

序号	功能名称	功能描述
1	5 位输入 3 选 1	option 为 10 输出 C，为 01 输出 B，为 00 输出 A

2	32 位输入 2 选 1	option 为 1 输出 B, 为 0 输出 A
3	32 位输入 3 选 1	option 为 10 输出 C, 为 01 输出 B, 为 00 输出 A
4	32 位输入 4 选 1	option 为 11 输出 D, 为 10 输出 C, 为 01 输出 B, 为 00 输出 A
5	32 位输入 5 选 1	option100 输出 E, 011 输出 D, 010 输出 C, 001 输出 B, 000 输出 A

9. Dregs (D 级流水线寄存器)

1) 端口说明

表格 17 Dregs 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1: 复位 0: 无效
IR[31:0]	I	D 级部件将使用的指令
PC8[31:0]	I	D 级部件对应指令的 PC+8
Stall	I	冒险单元输入的暂停信号
PCsel[1:0]	I	将传回 F 级部件的 PC 选择信号
I16[15:0]	I	将传回 F 级部件的 16 位立即数
I26[25:0]	I	将传回 F 级部件的 26 位立即数
Beq	I	将传回 F 级部件的 beq 识别信号
EXTop[2:0]	I	扩展控制信号 000: 无符号扩展 001: 低 16 位补 0 010: 有符号扩展 011: 有符号扩展后逻辑左移两位
WRsel[1:0]	I	选择寄存器 A3 端口信号 10:31 号寄存器 01: IR_W[rd] 00: IR_W[rt]
IR_D [31:0]	O	输出到 E 级寄存器的指令
PC8_D[31:0]	O	输出到 E 级寄存器的 PC+8
PCsel_D[1:0]	O	输出到 F 级寄存器的 PC 选择信号
i16_D	O	将传回 F 级部件的 16 位立即数
I26_D	O	将传回 F 级部件的 26 位立即数
Beq_D	O	将传回 F 级部件的 beq 识别信号
EXTop_D[2:0]	O	扩展控制信号 000: 无符号扩展 001: 低 16 位补 0 010: 有符号扩展 011:

		有符号扩展后逻辑左移两位
WRsel_D[1:0]	0	选择寄存器 A3 端口信号 10:31 号寄存器 01: IR_W[rd] 00: IR_W[rt]

2) 功能定义

表格 18 Dregs 功能定义

序号	功能名称	功能描述
1	存储结果	存储 F 级部件结果，发送到 D 级部件或 F 级部件

10. Eregs (E 级流水线寄存器)

1) 端口说明

表格 19 Eregs 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1: 复位 0: 无效
IR[31:0]	I	E 级部件将使用的指令
PC8[31:0]	I	E 级部件对应指令的 PC+8
Stall	I	冒险单元输入的暂停信号
V1[31:0]	I	MFcmp1D 转发而来的结果
V2[31:0]	I	MFcmp2D 转发而来的结果
E32[31:0]	I	EXT 结果
RFA3[4:0]	I	E 级指令要写入的寄存器编号
Bsel	I	Alub 多选器的选择信号
ALUctr [2:0]	I	控制信号 000: 加运算 001: 减运算 010: 或运算 011: 输出写入数据 2 100: 异或运算
IR_E[31:0]	0	输出到 E 级寄存器的指令
PC8_E[31:0]	0	输出到 E 级寄存器的 PC+8
RFA3E[4:0]	0	输出到 E 级的要写入的寄存器编号
V1[31:0]	0	MFcmp1D 转发而来的结果

V2[31:0]	0	MFcmp2D 转发而来的结果
Bsel_E	0	Alub 多选器的选择信号
ALUctr_E[2:0]	0	alu 控制信号 000: 加运算 001: 减运算 010: 或运算 011: 输出写入数据 2 100: 异或运算
Res_E [1:0]	0	E 级指令对部件的产生结果位置 ALU: 在 alu 产生结果 DM: 在 dm 产生结果 PC: 产生 PC 结果 NW: nowrite, 不产生结果

2) 功能定义

表格 20Eregs 功能定义

序号	功能名称	功能描述
1	存储结果	存储 D 级部件结果, 发送到 E 级部件
2	产生控制转发信号	计算 E 级指令对部件的产生结果位置

11. Mregs (M 级流水线寄存器)

1) 端口说明

表格 21Mregs 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1: 复位 0: 无效
IR[31:0]	I	M 级部件将使用的指令
PC8[31:0]	I	M 级部件对应指令的 PC+8
A0[31:0]	I	ALU 的结果
V2[31:0]	I	MFcmp2D 转发而来的结果
RFA3[4:0]	I	M 级指令要写入的寄存器编号
MemWrite	I	Dm 写入控制信号 1: 写入 0: 无效
IR_M[31:0]	O	输出到 W 级寄存器的指令
PC8_M[31:0]	O	输出到 W 级寄存器的 PC+8
RFA3M[4:0]	O	输出到 M 级的要写入的寄存器编号

A0_M[31:0]	0	输出到 W 级的 alu 结果
V2[31:0]	0	输出到 W 级的 MFcmp2D 转发而来的结果
MemWrite	0	输出到 M 级部件的 Dm 写入控制信号 1: 写入 0: 无效
Res_M[1:0]	0	M 级指令对部件的产生结果位置 ALU: 在 alu 产生结果 DM: 在 dm 产生结果 PC: 产生 PC 结果 NW: nowrite, 不产生结果

2) 功能定义

表格 22Mregs 功能定义

序号	功能名称	功能描述
1	存储结果	存储 E 级部件结果, 发送到 M 级部件
2	产生控制转发信号	计算 M 级指令对部件的产生结果位置

12. Wregs (W 级流水线寄存器)

1) 端口说明

表格 23Wregs 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1: 复位 0: 无效
IR[31:0]	I	W 级部件将使用的指令
PC8[31:0]	I	W 级部件对应指令的 PC+8
A0[31:0]	I	ALU 的结果
DR[31:0]	I	DM 的结果
RFA3[4:0]	I	W 级指令要写入的寄存器编号
RegWrite	I	寄存器堆写入控制信号 1: 写入 0: 无效
WDsel[1:0]	I	寄存器堆写入数据选择信号 10: PC8_W 01: DR_W 00: A0_W
PC8_W[31:0]	0	输出到 W 级部件的 PC+8
RFA3W[4:0]	0	输出到 W 级的要写入的寄存器编号
A0_W[31:0]	0	输出到 W 级部件的 alu 结果

DR_W[31:0]	0	输出到 W 级部件 DM 的结果
RegWrite	0	输出到 W 级部件的寄存器写入控制信号 1：写入 0：无效
WDsel_W[1:0]	0	寄存器堆写入数据选择信号 10：PC8_W 01：DR_W 00：AO_W
Res_W[1:0]	0	W 级指令对部件的产生结果位置 ALU：在 alu 产生结果 DM：在 dm 产生结果 PC：产生 PC 结果 NW：nowrite，不产生结果

2) 功能定义

表格 24Wregs 功能定义

序号	功能名称	功能描述
1	存储结果	存储 M 级部件结果，发送到 W 级部件
2	产生控制转发信号	计算 W 级指令对部件的产生结果位置

13. Cp0（协处理器）

1) 端口说明

表格 25cp0 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1：复位 0：无效
A1[4:0]	I	读 cp0 寄存器号
A2[4:0]	I	写 cp0 寄存器号
DIn[31:0]	I	写入数据
PC[31:0]	I	异常 PC
BD	I	延迟槽指示信号 1：在延迟槽中 0：无效
ExcCode [6:2]	I	中断异常类型 0：中断 4：取指取数错误 5：存数地址错误 10：无效指令 12：自陷式指令溢出
HWInt[5:0]	I	6 个中断设备
We	I	写使能信号 1：写入 0：无效
EXLClr	I	EXL 清零信号 1：清零 0：无效
AdEL	I	AdEL 指示信号

AdES	I	AdES 指示信号
IntReq	0	中断指示 1：中断 0：无效
EPC[31:2]	0	异常 PC
D0ut[31:0]	0	Cp0 输出数据

2) 功能定义

表格 26cp0 功能定义

序号	功能名称	功能描述
1	复位	复位信号有效时，HI 和 LO 寄存器和信号被置为 0
2	写入数据	向 cp0 中寄存器写入数据 din
3	读取数据	读出 cp0 中寄存器值
4	输出中断信号	输出中断信号
5	存储中断异常类型	在寄存器中存储中断异常类型

14. Muldiv（乘除单元）

1) 端口说明

表格 27Muldiv 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1：复位 0：无效
A[31:0]	I	乘除单元输入数据 1
B[31:0]	I	乘除单元输入数据 2
op[2:0]	I	乘除单元控制信号 000：无效 001：有符号乘 010：无符号乘 011：有符号除 100：无符号除 101：mthi 110：mtlo
HI[31:0]	O	HI 寄存器
LO[31:0]	O	LO 寄存器
busy	O	乘除运算进行信号 1：正在运算 0：无效
start	O	Busy 信号产生信号 1：下一个时钟上升沿 busy 有效 0：无效

2) 功能定义

表格 28Muldiv 功能定义

序号	功能名称	功能描述
1	复位	复位信号有效时，HI 和 L0 寄存器和信号被置为 0
2	有符号乘	有符号乘法，其中结果高 32 位保存在 HI，低 32 位保存在 L0，时长 5 周期
3	无符号乘	无符号乘法，其中结果高 32 位保存在 HI，低 32 位保存在 L0，时长 5 周期
4	有符号除	有符号除法，其中结果余数保存在 HI，商保存在 L0，时长 10 周期
5	无符号除	无符号除法，其中结果余数保存在 HI，商保存在 L0，时长 10 周期
6	修改 HI	将 rs 的值存入 HI 寄存器
7	修改 L0	将 rs 的值存入 L0 寄存器
8	读取 HI	读取 HI 寄存器的值，写入 rd
9	读取 L0	读取 L0 寄存器的值，写入 rd

(三) 冒险单元

1. 端口说明

表格 29hazard 端口说明

信号名	方向	描述
IR[31:0]	I	D 级部件将使用的指令
Res_E[1:0]	I	E 级传来的控制信号
Res_M[1:0]	I	M 级传来的控制信号
Res_W[1:0]	I	W 级传来的控制信号
A3_E[4:0]	I	E 级指令要写入的寄存器编号
A3_M[4:0]	I	M 级指令要写入的寄存器编号
A3_W[4:0]	I	W 级指令要写入的寄存器编号
A1_D[4:0]	I	D 级指令要读入的寄存器编号 1
A2_D[4:0]	I	D 级指令要读入的寄存器编号 2
A1_E[4:0]	I	E 级指令要读入的寄存器编号 1

A2_E[4:0]	I	E 级指令要读入的寄存器编号 2
A2_M[4:0]	I	M 级指令要读入的寄存器编号
Busy	I	乘除运算进行信号 1: 正在运算 0: 无效
start	I	Busy 信号产生信号 1: 下一个时钟上升沿 busy 有效 0: 无效
stall	0	暂停信号 1: 暂停 0: 无效
Fcmp1D[2:0]	0	输出到 D 级的 cmp 编号 1 转发信号 000: RF.V1 001: M_Wdata 010: AO_M 011: PC8_M 100: PC8_E
Fcmp2D[2:0]	0	输出到 D 级的 cmp 编号 2 转发信号 000: RF.V2 001: M_Wdata 010: AO_M 011: PC8_M 100: PC8_E
FaluaE[1:0]	0	输出到 E 级 ALUa 的转发信号 00: V1_E 01: M_Wdata 10: AO_M 11: PC8_M
FalubE[1:0]	0	输出到 E 级 ALUb 的转发信号 00: V2_E 01: M_Wdata 10: AO_M 11: PC8_M
FPCF [2:0]	0	输出到 F 级 PC 的转发信号 000: RF.V1 001: M_Wdata 010: AO_M 011: PC8_M 100: PC8_E
FdmdataM	0	输出到 M 级 DMin 的转发信号 0: V2_M 1: M_Wdata

2. 功能定义

表格 30hazard 功能定义

序号	功能名称	功能描述
1	产生暂停信号	根据 Tuse 和 Tnew 产生暂停信号
2	产生转发信号	根据 Tuse 和 Tnew 产生转发信号

(四) Bridge 和 timer

1. 端口说明

表格 31bridge 和 timer 端口说明

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号 1: 复位 0: 无效
ADD_I [1:0]	I	写入寄存器信号 0: ctrl 1: preset 2: count

WE_I	I	写入控制信号 1：写入 0：无效
DAT_I [31:0]	I	写入数据
DAT_O [31:0]	O	输出数据
IRQ	O	Timer 中断信号 1：中断 0：无效
PrAddr [31:0]	I	识别写入 timer0 或 timer1
timer0we	O	Timer0 指示信号 1：对 timer0 处理 0：无效
Timer1we	O	Timer1 指示信号 1：对 timer1 处理 0：无效

2. 功能定义

表格 32bridge 和 timer 功能定义

序号	功能名称	功能描述
1	复位	复位信号有效时，HI 和 LO 寄存器和信号被置为 0
2	产生中断信号	倒数至 0 时产生中断信号
3	读取数据	读取 timer 中数据
4	写入数据	向 timer 中写入数据

（五） 控制器设计

1. 端口说明

表格 33ctr 端口说明

信号名	方向	描述
instruction[31:0]	I	32 位指令
WR_sel	O	grf 写寄存器决定信号 0：rd 1：rt 2：31 号寄存器
A_sel	O	alu 输入数据 A 决定信号 0：MFAluaE 1：IR[10:6] 2：MFAluaE[4:0]
B_sel	O	alu 输入数据 B 决定信号 1：32 位立即数 0：GRF 寄存器 2 输出值
RegWrite	O	grf 写寄存器信号 1：写寄存器 0：无效
MemWrite	O	写内存 dm 信号 1：写入内存 0：无效
PC_sel	O	PC 跳转信号`ADD4：跳转 PC+4 `NPC：跳转 npc `RFV1：跳转寄存器堆 1
EXTop[1:0]	O	扩展控制信号 00：无符号扩展 01：低 16 位补 0 10：有符号扩展 11：

		有符号扩展后逻辑左移两位
ALUctr[3:0]	0	alu 控制信号 0000 加 0001 减 0010 或 0011B 0100 异或 0101 逻辑左移 0110 逻辑右移 0111 算数右移 1000 与 1001 或非 1010 有符号小于置 1 1011 无符号小于置 1
beq	0	Beq 指示信号 0: 无效 1: beq
bne	0	bne 指示信号 0: 无效 1: bne
Bgez	0	Bgez 指示信号 0: 无效 1: Bgez
Blez	0	Blez 指示信号 0: 无效 1: Blez
Bgtz	0	Bgtz 指示信号 0: 无效 1: Bgtz
blez	0	blez 指示信号 0: 无效 1: blez
Muldivop[2:0]	0	乘除模块控制信号 0: 无效 1: mult 2: multu 3: div 4: divu 5: mthi 6: mtlo
WDsel[1:0]	0	写入寄存器堆数据选择信号 0: DR_W 1: PC8_W 2: A0_W
Dmexto[2:0]	0	Dm 扩展器控制信号 0: 无效 1: lbu 2: lb 3: lhu 4: lh

2. 真值表

func	100001	100011	100000	100010	001000	001001							
op	000000	000000	000000	000000	000000	000000	001101	100011	101011	000100	001111	001110	101001
	addu	subu	add	sub	jr	jalr	ori	lw	sw	beq	lui	xori	sh
RegDst	1	1	1	1	0	1	0	0	0	0	0	0	0
Asel	00	00	00	00	00	00	00	00	00	00	00	00	00
Bsel	0	0	0	0	0	0	1	1	1	0	1	1	1
MemtoReg	0	0	0	0	0	0	0	1	0	0	0	0	0
RegWrite	1	1	1	1	0	1	1	1	1	0	0	1	1
MemWrite	0	0	0	0	0	0	0	0	1	0	0	0	1
EXTop	00	00	00	00	00	00	00	10	10	11	01	00	10
ALUctr	0000	0001	0000	0001	0000	0000	0010	0000	0000	0000	0011	0100	0000
dmextop	000	000	000	000	000	000	000	000	000	000	000	000	000
func	000000	000000	000000	000000	000000	000000							
op	000000	000010	000011	000100	000110	000111	000011	000010	100000	100100	100001	100101	101000
	sll	srl	sra	sllv	srlv	srav	jal	j	lb	lbu	lh	lhu	sb
RegDst	1	1	1	1	1	1	0	0	0	0	0	0	0
Asel	01	01	01	10	10	10	00	00	00	00	00	00	00
Bsel	0	0	0	0	0	0	0	0	1	1	1	1	1
MemtoReg	0	0	0	0	0	0	0	0	1	1	1	1	0
RegWrite	1	1	1	1	1	1	1	0	1	1	1	1	0
MemWrite	0	0	0	0	0	0	0	0	0	0	0	0	1
EXTop	00	00	00	00	00	00	00	00	10	10	10	10	10
ALUctr	0101	0110	0111	0101	0110	0111	0000	0000	0000	0000	0000	0000	0000
dmextop	000	000	000	000	000	000	000	000	010	001	100	011	000

表格 34ctr 真值表

（六） 测试

（一） 测试程序

功能测试：

#功能测试	add \$a2,\$t1,\$t1 #++	lui \$t3,0xf39f
ori \$t0,\$0,520	subu \$a0,\$t0,\$t1 #-+	or \$a1,\$t0,\$t3 #--
ori \$t1,\$t0,233	subu \$a1,\$t1,\$t0 #+-	xor \$a2,\$t3,\$t0
	subu \$a2,\$t1,\$t2 #++	nor \$a0,\$t0,\$t1 #-+
lui \$t0,520	subu \$a3,\$t2,\$t1 #++	nor \$a1,\$t1,\$t0 #+-
lui \$t1,0xffff	lui \$t3,0xf39f	nor \$a2,\$t1,\$t2 #++
ori \$t1,\$t1,0xffff	subu \$a1,\$t0,\$t3 #--	
	subu \$a2,\$t3,\$t0	#mult
#addu, subu		lui \$t0,0xfeaf
addu \$t2,\$t0,\$t0#++	#and, or, xor, nor	ori \$t0,\$t0,0x5254
addu \$t3,\$t0,\$t1#+-	ori \$s0,\$0,1	lui \$t1,0x0243
addu \$t4,\$t1,\$t1#--	ori \$s1,\$0,4	ori \$t1,\$t1,0x323f
subu \$t5,\$t0,\$t2#++	ori \$s2,\$0,15	mult \$t0,\$t0 #--
subu \$t6,\$t0,\$t1#+-	lui \$t0,0xfda3	mfhi \$a0
subu \$t7,\$t1,\$t4#--	ori \$t0,\$t0,0x34f5 #-	mflo \$a1
	ori \$t1,\$0,0x234 #+	mult \$t1,\$t0 #+-
#add, sub	lui \$t2,0x424	mfhi \$a0
ori \$s0,\$0,1	ori \$t2,\$t2,32853	mflo \$a1
ori \$s1,\$0,4	and \$a0,\$t0,\$t1 #-+	mult \$t1,\$t1 #-+
ori \$s2,\$0,15	and \$a1,\$t0,\$t0 #--	mfhi \$a1
lui \$t0,0xfda3	and \$a2,\$t1,\$t1 #++	mflo \$a0
ori \$t0,\$t0,0x34f5 #-	or \$a0,\$t0,\$t1 #-+	
ori \$t1,\$0,0x234 #+	or \$a1,\$t1,\$t0 #+-	#multu
lui \$t2,0x424	or \$a2,\$t1,\$t2 #++	lui \$t0,0xfeaf
ori \$t2,\$t2,32853	xor \$a3,\$t2,\$t1 #++	ori \$t0,\$t0,0x5254
add \$a0,\$t0,\$t1 #-+	xor \$a0,\$t0,\$t1 #-+	lui \$t1,0x0243
add \$a1,\$t0,\$t0 #--	xor \$a1,\$t1,\$t0 #+-	ori \$t1,\$t1,0x323f

mflo \$a1	mflo \$a1	mfhi \$a0
multu \$t1, \$t0	div \$t1, \$t3 #++	mflo \$a1
mfhi \$a0	mfhi \$a0	mult \$t1, \$t0
mflo \$a1	mflo \$a1	mthi \$t1
multu \$t1, \$t1	div \$t0, \$t1 #--+	mfhi \$a0
mfhi \$a0	mfhi \$a0	mult \$t1, \$t1
mflo \$a1	mflo \$a1	mtlo \$t0
	div \$t1, \$t0	mfhi \$a0
#div, divu	mfhi \$a0	mflo \$a1
lui \$t0, 0xfeaf	mflo \$a1	
ori \$t0, \$t0, 0x5254	div \$t0, \$t2 #--	#sll, srl, sra
lui \$t1, 0x0243	mfhi \$a0	ori \$t0, \$t0, 123
ori \$t1, \$t1, 0x323f	mflo \$a1	sll \$t0, \$t0, 4
lui \$t2, 0xe120	div \$t2, \$t0	srl \$t0, \$t0, 4
ori \$t2, \$t2, 0x300	mfhi \$a0	ori \$t1, \$t1, 0xffff
ori \$t3, \$0, 0x99	mflo \$a1	sll \$t1, \$t1, 2
div \$t0, \$t1 #--+	div \$t1, \$t3 #++	srl \$t1, \$t1, 2
mfhi \$a0	mfhi \$a0	sra \$t0, \$t0, 3 #+
mflo \$a1	mflo \$a1	lui \$t0, 0xf234
div \$t1, \$t0		sra \$t1, \$t0, 3 #-
mfhi \$a0	#mthi, mtlo	lui \$t0, 0x23
mflo \$a1	lui \$t0, 0xfeaf	sra \$t0, \$t0, 4 #+
div \$t0, \$t2 #--	ori \$t0, \$t0, 0x5254	
mfhi \$a0	lui \$t1, 0x0243	#sllv, srlv, srav
mflo \$a1	ori \$t1, \$t1, 0x323f	ori \$s0, \$0, 1
div \$t2, \$t0	mult \$t0, \$t0	ori \$s1, \$0, 4
mfhi \$a0	mthi \$t0	ori \$s2, \$0, 15

```

lui $t0, 0xfda3
ori $t0, $t0, 0x34f5
ori $t1, $0, 0x234
lui $t2, 0x424
ori $t2, $t2, 32853
sllv $a0, $t0, $s0 #-
sllv $a1, $t1, $s1 #+
sllv $a3, $t2, $s2 #+
srlv $a0, $t0, $s0
srlv $a1, $t1, $s1
srlv $a3, $t2, $s2
srav $a0, $t0, $s0
srav $a0, $t0, $s1
srav $a1, $t1, $s1
srav $a3, $t2, $s2

#slt, sltu
ori $t0, $0, 2
ori $t1, $0, 1
ori $t2, $0, 2
lui $t3, 0xffff
slt $a0, $t2, $t1#++
slt $a1, $t3, $t1#--
ori $t4, $t3, 0x1234
slt $a2, $t3, $t4 #--
sltu $a0, $t2, $t1 #++
sltu $a1, $t3, $t1 #-+

```

```

sltu $a2, $t3, $t4
sltu $a3, $t4, $t3

#jalr
ori $t0, $0, 1
jal jalr_loop
ori $t0, $ra, 0
ori $t2, $0, 2
addu $a1, $a1, $a0
j jalr_end
ori $t6, $t6, 6
jalr_loop:
addu $a0, $0, $ra
jalr $a0, $ra
ori $t4, $a0, 0
ori $t5, $t5, 5
jalr_end:

#addi, addiu, andi, xori
ori $s0, $0, 1
ori $s1, $0, 4
ori $s2, $0, 15
lui $t0, 0xfda3
ori $t0, $t0, 0x34f5 #-
ori $t1, $0, 0x234 #+
lui $t2, 0x424
ori $t2, $t2, 32853

```

```

addi $a0, $t0, 0xea4
addi $a1, $t0, -134
addi $a2, $t1, 0xf53f
addi $a3, $t1, 533
addiu $a0, $t0, 0xea4
addiu $a1, $t0, -134
addiu $a2, $t1, 0xf53f
addiu $a3, $t1, 533
andi $a0, $t0, 0xea4
andi $a1, $t0, -134
andi $a2, $t1, 0xf53f
andi $a3, $t1, 533
xori $a0, $t0, 0xea4
xori $a1, $t0, -134
xori $a2, $t1, 0xf53f
xori $a3, $t1, 533

#slti, sltiu
ori $t0, $0, 2
ori $t1, $0, 1
lui $t3, 0xffff
slti $a0, $t1, 2 #+
slti $a1, $t0, 1
slti $a2, $t0, -100
slti $a3, $t3, 2390
sltiu $a0, $t1, 2 #+
sltiu $a1, $t0, 1

```

sltiu \$a2, \$t0, -100 #-	lbu \$2, 3(\$0)	bgtz \$a2, bgtz_label3
sltiu \$a3, \$t3, 2390		ori \$t0, \$t0, 1
	#bne	ori \$t1, \$t1, 2
#sw, sh, sb, lh, lhu, lb, lbu	ori \$t0, \$0, 1	bgtz \$a0, bgtz_label3
ori \$20, \$0, 1	bne \$t0, \$t0, bne_label1	ori \$t2, \$t2, 2
lui \$21, 0xffff	ori \$t1, \$0, 1	ori \$t3, \$t3, 3
ori \$21, \$21, 0xffff	ori \$t2, \$0, 2	bgtz \$a1, bgtz_label3
lui \$3, 0xf3f4	bne \$t0, \$t2, bne_label1	ori \$t5, \$t5, 5
ori \$3, \$3, 0x71f2	ori \$t3, \$0, 3	ori \$t6, \$t6, 6
sw \$3, 0(\$0)	ori \$t4, \$0, 4	bgtz_label3:ori \$t4, \$t4, 4
sh \$3, 8(\$0)	bne_label1:ori \$t3, \$t3, 3	
sh \$3, 10(\$0)		#bltz, bgez
sb \$3, 4(\$0)	#blez, bgtz	ori \$a1, \$0, 1
sb \$3, 5(\$0)	ori \$a1, \$0, 1	lui \$a2, 0xf433
sb \$3, 6(\$0)	lui \$a2, 0xf433	bgez \$a2, bgez_label1
sb \$3, 7(\$0)	blez \$a1, blez_label1	ori \$t0, \$t0, 1
lh \$2, 0(\$0)	ori \$t0, \$t0, 1	ori \$t1, \$t1, 2
lh \$2, 2(\$0)	ori \$t1, \$t1, 2	ori \$a0, \$0, 0
lhu \$2, 0(\$0)	ori \$a0, \$0, 0	bgez \$a0, bgez_label1
lhu \$2, 2(\$0)	blez \$a0, blez_label1	ori \$t2, \$t2, 2
lb \$2, 0(\$0)	ori \$t2, \$t2, 2	ori \$t3, \$t3, 3
lb \$2, 1(\$0)	ori \$t3, \$t3, 3	bgez_label1:ori \$t4, \$t4, 4
lb \$2, 2(\$0)	blez_label1:ori \$t4, \$t4, 4	bgez \$a1, bgez_label2
lb \$2, 3(\$0)	blez \$a2, bgez_label2	ori \$t5, \$t5, 5
lbu \$2, 0(\$0)	ori \$t5, \$t5, 5	ori \$t6, \$t6, 6
lbu \$2, 1(\$0)	ori \$t6, \$t6, 6	bgez_label2:ori \$t7, \$t7, 7
lbu \$2, 2(\$0)	blez_label2:ori \$t7, \$t7, 7	bltz \$a1, bgez_label3


```
ori $t0, $t0, 1
ori $t1, $t1, 2
bltz $a0, bgez_label3
ori $t2, $t2, 2
ori $t3, $t3, 3
bltz $a2, bgez_label3
ori $t5, $t5, 5
ori $t6, $t6, 6
bgez_label3:ori $t4, $t4, 4

ori $t0, $0, 8
sw $t1, -8($t0)
sw $t2, 0($t0)
sw $t3, 8($t0)
lw $t4, -8($t0)
lw $t5, 0($t0)
lw $t6, 8($t0)
```

异常测试:

```

1  li $28, 0
2  li $29, 0
3  mtc0 $0, $13
4  mtc0 $0, $14
5  li $0, 0x45762dbc
6  li $1, 0x1402ca0f
7  li $2, 0x203c737d
8  li $3, 0x25df6e42
9  li $4, 0x7d998177
10 li $5, 0x1c55e393
11 sub $5, $1, $3
12 sub $2, $5, $2
13 andi $1, $2, 0x47c0
14 lh $3, 8($0)
15 sltiu $4, $1, 0x6fb6
16 mtc0 $3, $14
17 mtc0 $0, $14
18 and $5, $4, $4
19 divu $4, $0
20 mthi $5
21 nor $0, $2, $0
22 sw $4, 22($0)
23 mfc0 $2, $13
24 sub $2, $5, $3
25 addu $0, $1, $3
26 add $2, $2, $3
27 li $5, 0xdc874dc4
28 sub $4, $5, $3
29 nor $3, $3, $5
30 mfc0 $1, $13
31 slti $3, $5, 0x5393
32 xori $3, $1, 0x94b
33 addi $4, $0, 0x1a02
34 addiu $5, $2, 0x70a
35 multu $0, $0
36 lw $3, 34($0)
37 slti $1, $3, 0x12d3
38 nor $4, $5, $2
39 addi $5, $4, 0x5b64
40 mtc0 $5, $14
41 lb $4, 27($0)
42 add $2, $2, $1
43 divu $3, $1
44 multu $1, $5
45 addi $1, $5, 0x5d5d
46 addi $3, $2, 0x2b9d
47 addi $3, $2, 0x2b9d
48 sub $5, $0, $5
49 andi $4, $3, 0x25c4
50 addi $1, $3, 0x4ffb
51 mflo $2
52 mtc0 $5, $14
53 lw $1, 1($0)
54 sub $5, $5, $5
55 sltu $1, $2, $4
56 or $4, $2, $2
57 addi $2, $2, 0x48da
58 addi $0, $5, 0x76a3
59 xori $2, $0, 0xb0c
60 mtlo $3
61 add $2, $2, $0
62 srav $0, $2, $0
63 xor $1, $5, $1
64 slt $4, $3, $2
65 mfc0 $3, $13
66 mflo $3
67 addi $5, $2, 0x4540
68 lui $0, 0x4c44
69 multu $4, $0
70 mtc0 $2, $14
71 srlv $5, $0, $4
72 sub $0, $1, $3
73 sub $0, $3, $3
74 sltu $0, $0, $1
75 mflo $3
76 mfc0 $2, $14
77 or $1, $2, $4
78 add $1, $3, $0
79 sub $5, $4, $1
80 addi $2, $3, 0x34e6
81 xor $4, $2, $5
82 ori $3, $4, 0x6f34
83 mfc0 $2, $13
84 and $3, $1, $1
85 or $5, $0, $2
86 mfc0 $1, $13
87 and $5, $2, $3
88 lw $1, 3($0)
89 lui $2, 0x8eb
90 subu $0, $4, $0
91 sub $1, $3, $0
92 mflo $4

```

92	sub \$2, \$2, \$1	137	add \$2, \$2, \$1
93	mthi \$4	138	add \$1, \$3, \$2
94	mfc0 \$1, \$14	139	addi \$2, \$4, 0x3fa6
95	add \$4, \$4, \$3	140	sub \$0, \$1, \$5
96	nor \$0, \$4, \$4	141	mfhi \$4
97	mtc0 \$3, \$13	142	div \$1, \$4
98	add \$3, \$3, \$0	143	slti \$4, \$4, 0x7bfb
99	multu \$1, \$1	144	sub \$0, \$0, \$3
100	nor \$3, \$5, \$4	145	li \$2, 0x8eb3782a
101	addi \$1, \$5, 0x38e5	146	addi \$4, \$0, 0x6b37
102	mthi \$3	147	subu \$5, \$1, \$1
103	addi \$0, \$1, 0x46a	148	addu \$1, \$1, \$2
104	addi \$3, \$4, 0x372f	149	li \$0, 0xef7fcf4f
105	addi \$1, \$1, 0x5d91	150	mtc0 \$1, \$13
106	addi \$5, \$1, 0x9c4	151	addi \$2, \$4, 0x1bf0
107	addi \$4, \$2, 0x78a1	152	addi \$4, \$4, 0xd2a
108	addi \$0, \$4, 0x168a	153	add \$1, \$0, \$4
109	add \$1, \$0, \$2	154	sub \$1, \$2, \$0
110	sltu \$1, \$4, \$0	155	mtc0 \$3, \$13
111	lh \$3, 44(\$0)	156	add \$2, \$4, \$0
112	mfhi \$5	157	sltiu \$1, \$3, 0x2efb
113	mtc0 \$1, \$14	158	sub \$1, \$5, \$2
114	slt \$1, \$1, \$5	159	add \$5, \$3, \$3
115	mtc0 \$2, \$13	160	li \$3, 0xab904171
116	subu \$4, \$1, \$0	161	addi \$5, \$2, 0x5d55
117	add \$2, \$4, \$2	162	slt \$3, \$2, \$1
118	mflo \$4	163	sub \$5, \$2, \$0
119	lh \$2, 23(\$0)	164	xor \$1, \$5, \$1
120	div \$1, \$4	165	nor \$1, \$4, \$4
121	lhu \$0, 51(\$0)	166	mfc0 \$1, \$13
122	or \$1, \$2, \$4	167	lui \$0, 0x35fa
123	sub \$1, \$3, \$4	168	addi \$1, \$2, 0x1731
124	slt \$5, \$2, \$4	169	divu \$3, \$3
125	lui \$1, 0x317c	170	sw \$4, 48(\$0)
126	and \$0, \$5, \$3	171	lbu \$3, 18(\$0)
127	and \$4, \$5, \$4	172	sltu \$2, \$1, \$5
128	and \$3, \$0, \$1	173	sb \$5, 20(\$0)
129	addi \$3, \$1, 0x4c72	174	mthi \$0
130	sub \$4, \$0, \$4	175	sw \$3, 22(\$0)
131	mflo \$4	176	sltiu \$4, \$5, 0x3755
132	or \$3, \$0, \$1	177	addiu \$4, \$4, 0x2c82
133	nor \$0, \$0, \$3	178	sub \$2, \$3, \$5
134	andi \$3, \$2, 0x148	179	add \$0, \$4, \$3
135	sub \$5, \$2, \$2	180	sub \$1, \$4, \$5
136	subu \$3, \$2, \$4	181	lhu \$4, 11(\$0)
137	add \$2, \$4, \$3	182	add \$1, \$2, \$0

```

183    sltiu $2, $3, 0x56d2
184    xor $2, $0, $4
185    add $0, $0, $4
186    lb $1, 36($0)
187    mtc0 $5, $14
188    lhu $4, 0($0)
189    lhu $4, 51($0)
190    sw $4, 12($0)
191    sh $1, 13($0)
192    sb $4, 46($0)
193    addi $0, $3, 0x2615
194    addi $3, $0, 0x2f38
195    nor $3, $0, $0
196    sw $1, 8($0)
197    lhu $0, 51($0)
198    addiu $4, $4, 0xe58
199    mtlo $1
200    addi $2, $3, 0x14d6
201    xor $5, $2, $4
202    sub $5, $2, $2
203    mfc0 $0, $13
204    lh $0, 28($0)
205    andi $3, $2, 0x3a10
206    addiu $3, $3, 0x6447
207    addi $2, $2, 0x6efc
208    mflo $4
209    addi $4, $0, 0x332f
210    slt $1, $0, $3
211    mult $5, $0
212    srlv $5, $1, $0
213    addiu $1, $3, 0x108a
214    addi $1, $1, 0x21a8
215    sltiu $5, $2, 0x12af
216    slt $1, $0, $1
217    subu $1, $0, $1
218    sb $4, 45($0)
219    sub $3, $0, $1
220    mtlo $0
221    add $4, $1, $0
222    sb $4, 10($0)
223    add $2, $4, $3
224    sub $2, $2, $3
225    add $2, $3, $3
226    addi $1, $2, 0xc9a
227    addi $5, $1, 0x4de1
228    sub $0, $4, $2

```

```

228    sub $0, $4, $2
229    add $4, $4, $0
230    mtc0 $3, $14
231    sub $4, $4, $2
232    sub $0, $0, $0
233    sub $3, $3, $0
234    add $0, $0, $5
235    lhu $2, 41($0)
236    sh $2, 12($0)
237    sub $5, $2, $3
238    ori $3, $5, 0x721f
239    nor $2, $0, $4
240    addi $0, $1, 0x2e54
241    subu $4, $3, $0
242    addi $0, $1, 0x7c05
243    lh $0, 25($0)
244    xori $2, $5, 0x2235
245    addi $2, $4, 0x2006
246    mflo $3
247    addi $0, $4, 0x6bc5
248    sub $0, $1, $4
249    sub $2, $4, $2
250    srav $3, $3, $0
251    sub $4, $0, $3
252    nor $5, $5, $2
253    sw $5, 14($0)
254    andi $3, $3, 0x4240
255    mthi $1
256    sub $1, $3, $4
257    or $5, $1, $0
258    mfhi $2
259    sub $1, $4, $4
260    multu $2, $5
261    addi $2, $4, 0xcd7
262    slti $1, $0, 0x711c
263    lhu $3, 35($0)
264    xori $1, $0, 0x2ad9
265    mult $1, $4
266    or $2, $3, $5
267    mflo $5
268    li $4, 0x7b81228b
269    addu $0, $0, $5
270    sub $0, $2, $1
271    add $1, $0, $1
272    sub $4, $2, $4
273    nor $1, $4, $0

```

```

274 lw $1, 9($0)
275 mfhi $1
276 addi $2, $1, 0x2b41
277 addiu $4, $5, 0x3aad
278 addi $4, $1, 0x73e4
279 lh $2, 42($0)
280 sltu $0, $0, $2
281 sub $3, $1, $2
282 addi $3, $1, 0x66c6
283 li $2, 0x607b981a
284 mtc0 $3, $14
285 addiu $3, $4, 0x2a02
286 mtlo $1
287 sltiu $4, $1, 0x2238
288 add $1, $3, $3
289 addi $0, $1, 0x5141
290 sltiu $3, $2, 0xfd0
291 addu $3, $3, $2
292 add $1, $4, $3
293 sub $1, $5, $4
294 mfc0 $3, $13
295 addi $5, $5, 0x7f52
296 slti $5, $1, 0x3c8d
297 addi $1, $4, 0x4848
298 mfc0 $4, $14
299 sub $5, $2, $2
300 add $2, $1, $1
301 mflo $0
302 addiu $3, $4, 0x4584
303 sw $5, 30($0)
304 xor $4, $0, $5
305 subu $1, $0, $4
306 sub $3, $2, $0
307 add $4, $1, $3
308 mfc0 $4, $14
309 lhu $3, 35($0)
310 mtc0 $2, $14
311 sb $4, 9($0)
312 lh $1, 0($0)
313 addi $3, $2, 0x2f4f
314 srlv $2, $0, $2
315 addi $0, $3, 0x6a22
316 div $5, $4
317 andi $5, $5, 0x5c4d
318 add $4, $0, $4
319 lbu $3, 5($0)

```

```

319 lbu $3, 5($0)
320 sltiu $4, $0, 0x722d
321 lhu $0, 25($0)
322 addi $4, $2, 0x4bdf
323 addi $1, $1, 0x77bb
324 add $4, $5, $0
325 lui $5, 0x1f7f
326 lui $0, 0x59b1
327 mthi $0
328 slti $0, $5, 0x20ce
329 lbu $0, 0($0)
330 mfc0 $0, $14
331 addi $2, $1, 0x10d8
332 addi $0, $0, 0x3e87
333 mtc0 $2, $13
334 sltiu $1, $2, 0x3a84
335 addi $1, $1, 0x206c
336 slti $3, $4, 0x5108
337 addi $1, $4, 0x694e
338 sub $5, $0, $5
339 sub $1, $1, $2
340 slti $1, $2, 0x5353
341 sub $5, $2, $4
342 add $0, $5, $3
343 sltu $1, $0, $2
344 lhu $1, 22($0)
345 mfhi $4
346 mfc0 $3, $14
347 sub $3, $0, $2
348 addi $4, $1, 0x4211
349 sub $4, $2, $4
350 or $1, $1, $0
351 sltiu $0, $4, 0x7968
352 sb $3, 39($0)
353 sub $0, $2, $3
354 addi $3, $3, 0xa69
355 addiu $2, $1, 0x5635
356 lui $4, 0x63ac
357 addi $3, $5, 0x7e2d
358 mfc0 $5, $14
359 div $5, $5
360 add $2, $5, $3
361 lhu $5, 37($0)
362 add $1, $3, $4
363 slt $3, $1, $2
364 add $2, $3, $2
365 lb $3, $3, $0

```



```

365 add $3, $3, $2
366 add $3, $0, $0
367 mfc0 $5, $13
368 mfhi $5
369 srlv $2, $2, $5
370 mtlo $2
371 sltu $4, $0, $5
372 add $0, $4, $4
373 mtc0 $3, $13
374 mtc0 $0, $13
375 addi $2, $1, 0x4ceb
376 slti $4, $4, 0x11ac
377 mfc0 $4, $14
378 div $3, $3
379 lw $2, 39($0)
380 lb $3, 10($0)
381 addi $4, $4, 0x63b2
382 andi $5, $2, 0x16a
383 add $3, $0, $3
384 sltiu $5, $3, 0x4b45
385 sub $1, $3, $3
386 andi $3, $2, 0x1624
387 ori $0, $0, 0x5729
388 ori $4, $1, 0x1c01
389 add $5, $2, $0
390 add $0, $5, $3
391 sub $0, $5, $5
392 subu $4, $4, $3
393 ori $5, $3, 0x6c18
394 addu $5, $1, $3
395 lbu $2, 4($0)
396 sltiu $1, $1, 0x4bd3
397 mthi $2
398 mfc0 $5, $14
399 and $3, $0, $2
400 addi $3, $5, 0x258e
401 lh $3, 9($0)
402 divu $0, $4
403 srav $3, $1, $3
404 lb $4, 32($0)
405 sub $1, $2, $2
406 add $1, $4, $0
407 andi $4, $4, 0x6943
408 lhu $3, 8($0)
409 add $4, $4, $5
410 add $1, $4, $5

```

```

410 addu $1, $4, $5
411 sub $3, $4, $0
412 lui $4, 0x24b5
413 lbu $1, 33($0)
414 add $0, $1, $0
415 add $0, $1, $4
416 add $4, $5, $1
417 nor $2, $5, $5
418 srlv $2, $1, $3
419 subu $1, $1, $0
420 nor $5, $2, $2
421 sltu $3, $4, $3
422 mtc0 $3, $13
423 slt $2, $0, $4
424 addi $3, $4, 0x3a34
425 xor $3, $5, $3
426 lhu $2, 49($0)
427 add $3, $2, $3
428 addu $2, $0, $2
429 mfc0 $0, $14
430 add $0, $3, $5
431 andi $4, $3, 0x7451
432 lui $0, 0x78d3
433 sub $1, $0, $3
434 addi $0, $5, 0x48d9
435 andi $2, $1, 0x6c88
436 addi $3, $1, 0x2d4d
437 li $1, 0x90b0c80e
438 lbu $1, 39($0)
439 sub $0, $1, $1
440 mfc0 $2, $14
441 addi $4, $0, 0xf09
442 multu $5, $5
443 mtc0 $3, $14
444 mtc0 $3, $13
445 sub $2, $0, $0
446 or $2, $2, $3
447 subu $3, $1, $1
448 lbu $4, 47($0)
449 mtc0 $1, $14
450 mthi $2
451 srav $2, $2, $4
452 sub $3, $4, $4
453 mfc0 $5, $14
454 lw $1, 36($0)
455 sub $4, $4, $3

```

456	add \$2, \$5, \$5	501	sub \$3, \$1, \$4
457	add \$4, \$1, \$1	502	sw \$1, 12(\$0)
458	sub \$4, \$5, \$2	503	xori \$2, \$4, 0x14c2
459	divu \$3, \$0	504	srlv \$1, \$3, \$3
460	add \$1, \$4, \$3	505	sub \$3, \$1, \$1
461	sb \$3, 46(\$0)	506	add \$5, \$1, \$2
462	addi \$3, \$0, 0x50a1	507	lw \$2, 28(\$0)
463	sllv \$0, \$0, \$0	508	mtc0 \$3, \$13
464	xor \$1, \$1, \$4	509	add \$4, \$5, \$0
465	sw \$4, 35(\$0)	510	lb \$0, 17(\$0)
466	lbu \$2, 3(\$0)	511	addi \$5, \$5, 0x1a05
467	sub \$2, \$4, \$2	512	sub \$5, \$0, \$2
468	xori \$3, \$5, 0x603a	513	lw \$4, 22(\$0)
469	nor \$3, \$5, \$3	514	sub \$0, \$5, \$2
470	div \$5, \$2	515	addi \$0, \$4, 0x7384
471	or \$0, \$2, \$3	516	add \$0, \$0, \$5
472	srav \$5, \$5, \$4	517	and \$4, \$0, \$5
473	lh \$1, 46(\$0)	518	mtc0 \$4, \$14
474	mtc0 \$5, \$13	519	nor \$3, \$1, \$5
475	mtlo \$0	520	add \$1, \$5, \$5
476	nor \$0, \$3, \$4	521	sb \$1, 51(\$0)
477	sltiu \$1, \$5, 0x1557	522	add \$0, \$1, \$3
478	mfhi \$1	523	sub \$0, \$4, \$1
479	nor \$5, \$1, \$3	524	add \$5, \$4, \$4
480	addi \$0, \$1, 0x6d93	525	mtc0 \$0, \$14
481	div \$5, \$0	526	addi \$0, \$3, 0x7be8
482	lui \$3, 0x2224	527	addi \$4, \$2, 0xe05
483	multu \$2, \$0	528	xor \$2, \$3, \$5
484	add \$3, \$1, \$3	529	divu \$5, \$5
485	mfc0 \$4, \$13	530	sltiu \$3, \$5, 0x7491
486	sub \$0, \$4, \$4	531	lb \$0, 8(\$0)
487	ori \$0, \$1, 0x475c	532	add \$2, \$4, \$2
488	sub \$1, \$1, \$2	533	add \$4, \$4, \$4
489	sub \$2, \$5, \$2	534	addi \$0, \$4, 0xcb6
490	addu \$3, \$1, \$2	535	li \$5, 0x44eeba22
491	add \$1, \$4, \$1	536	add \$3, \$4, \$4
492	nor \$5, \$0, \$3	537	addi \$2, \$4, 0x738c
493	mfc0 \$3, \$13	538	addi \$4, \$2, 0x6afd
494	sub \$1, \$1, \$1	539	add \$5, \$4, \$5
495	or \$0, \$3, \$1	540	multu \$4, \$1
496	add \$4, \$0, \$4	541	add \$5, \$1, \$3
497	addi \$2, \$0, 0x697a	542	mtc0 \$4, \$13
498	addi \$1, \$1, 0x1d37	543	addi \$1, \$4, 0x3f30
499	sh \$1, 30(\$0)	544	mthi \$1
500	li \$2, 0xd4cd0200	545	mtlo \$3
501	sub \$3, \$1, \$4	546	mult \$4, \$3

```
543      addu $1, $4, 0x3130
544      mthi $1
545      mtlo $3
546      mult $4, $3
547      addi $5, $0, 0x3ded
548      add $2, $1, $1
549      add $1, $1, $1
550      mfc0 $0, $14
551      mult $2, $4
552      lhu $5, 38($0)
553      lb $4, 2($0)
554      xori $0, $4, 0x7152
555      mfc0 $3, $13
556      mtc0 $0, $14
557      xori $1, $4, 0x68e
558      mtc0 $3, $14
559      sllv $5, $5, $2
560      srav $4, $1, $5
561      subu $1, $0, $2
562      sw $0, 34($0)
563      sltiu $1, $0, 0x5c40
564      mfhi $4
565      sub $1, $4, $4
566      sllv $4, $4, $1
567      sub $3, $2, $5
568      lbu $4, 49($0)
569      add $4, $0, $1
570      nor $0, $0, $2
571      addiu $5, $4, 0x4f5e
572      lbu $3, 24($0)
573      lhu $0, 33($0)
574      addu $5, $4, $0
575      lb $1, 46($0)
576      lhu $4, 26($0)
577      sub $0, $2, $0
578      sltiu $1, $5, 0x809
579      add $3, $3, $5
580      mtc0 $5, $13
581      srav $1, $1, $5
582      lw $1, 18($0)
583      lhu $1, 30($0)
584      slti $1, $2, 0x3404
585      sh $2, 19($0)
586      xori $1, $0, 0x1411
587      addi $4, $3, 0x127
588      sub $0, $0, $1
```


(二) 结论

期望输出与实际输出相同。

二、 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”？

硬件/软件接口：在接口一端是中断处理程序和用于不同设备的设备驱动程序，另一端是各种硬件设备的控制器，如键盘控制器、打印机控制器。由于设备种类繁多，故该接口相当复杂。

(二) 在我们设计的流水线中，DM 处于 CPU 内部，请你考虑现代计算机中它的位置应该在何处。

计算机中的存储器按用途存储器可分为主存储器（内存）和辅助存储器（外存）。外存通常是磁性介质或光盘等，能长期保存信息，容量巨大。内存（即主存）指主板上的存储部件，用来存放当前正在执行的数据和程序，容量大、速度较慢。

无论是 IM 指令存储器还是 DM 数据存储器，以及所有设备，都应该位于 CPU 外部，因为 DM 相当于现代计算机中的外存，现代计算机通过总线来传输所有的数据和信号，达到沟通的目的。

(三) BE 部件对所有的外设都是必要的吗？

并不是对所有外设都需要。比如本实验用的两个 timer 只需要支持 sw，而不需要 sb、sh，而对于需要支持按半字和字节写入的外设，比如 dm 来说，BE 是必须的。BE 是否使用，取决于外设的需求。

(四) 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。

定时器在主程序中被初始化为模式

定时器倒计时至 0 产生中断

handler 设置使能 Enable 为 1 从而再次启动定时器的计数。

2 及 3 被无限重复。设定初值寄存器的为某个值，如 100 或 1000。（注意，主程序可能需要涉及对 CP0.SR 的编程，推荐

阅读过后文再进行。

```
ori $t4, $0, 64513
mtc0 $t4, $12
ori $t1, $0, 0
ori $t1, $t1, 0x7f00
ori $t2, $0, 0
ori $t2, $t2, 1000#preset
sw $t2, 4($t1) #preset存入相关寄存器
ori $t2, $0, 0
ori $t2, $t2, 9
sw $t2, ($t1)#启动倒数
ori $t3, $0, 1
ori $t3, $0, 2
ori $t3, $0, 3
ori $t3, $0, 4
ori $t3, $0, 5
ori $t3, $0, 6
ori $t3, $0, 7
ori $t3, $0, 8
ori $t3, $0, 9
ori $t3, $0, 10
ori $t3, $0, 1
ori $t3, $0, 2
ori $t3, $0, 3
ori $t3, $0, 4
ori $t3, $0, 5
ori $t3, $0, 6
ori $t3, $0, 7
ori $t3, $0, 8
ori $t3, $0, 9
ori $t3, $0, 10
```

(五) 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？

键盘、鼠标这类的低速设备是通过中断请求的方式进行 I/O 操作的。即当键盘上按下一个按键的时候，键盘会发出一个中断信号，中断信号经过中断控制器传到 CPU，然后 CPU 根据不同的中断号执行不同的中断响应程序，然后进行相应的 I/O 操作，把按下的按键编码读到寄存器（或者鼠标的操作），最后放入内存中。