CPU 设计文档

一、 数据通路设计

1、 IF 级

1） PC

端口	方向	功能
Npc[31:0]	I	下一条指令地址
Clk	I	时钟端
Reset	I	同步复位
Enable	I	使能端
Pc[31:0]	O	指令地址

时钟上升沿到来时若使能端有效 Pc<=Npc。

Reset 同步复位将 Pc 初始化为 0x00003000

2） ROM

端口	方向	功能
Pc[31:0]	I	指令地址
Instr[31:0]	O	指令码

Rom 规格为 32bit*4096，截取 Pc 的 14 到 2 位作为地址信号。ROM 范围从 0x0c000 到 0x1bff。

3） MUX_PC

端口	方向	功能
Pc_add_4[31:0]	I	PC+4
B_pc[31:0]	I	B 型指令的下一条地址
J_pc[31:0]	I	J 型指令的下一条地址
RD1[31:0]	I	JR 型指令的下一条地址
MUX_PCsel[1:0]	I	选择信号： 00：Pc_add_4 01:B_pc 10:J_pc 11:RD1
Npc[31:0]	O	输出下一条指令地址

该模块虽然放在 IF 级，但是从时间上来讲属于 ID 级或 EX 级

2、 ID 级

1） IF_ID_REG

端口	方向	功能

Pc[31:0]	I	IF 级的 PC
Instr[31:0]	I	IF 级的指令码
Reset	I	同步复位
Clk	I	时钟端
Enable	I	使能端
Pc_ID[31:0]	O	ID 级 PC
Instr_ID[31:0]	O	ID 级指令码

流水线寄存器，时钟上升沿到来时若使能端有效，分别将上一级数据读入本级对应寄存器，同步复位 PC 初始化为 0x00003000

2） GRF（ID）

端口	方向	功能
RA1[4:0]	I	读地址一
RA2[4:0]	I	读地址二
Rdata1[31:0]	O	读数据一
Rdata2[31:0]	O	读数据二

模块为 GRF 的 ID 级部分，即仅包括读数据功能

3） EXT

端口名	方向	功能
In[15:0]	I	十六位输入信号
EXTsel[1:0]	I	功能选择信号 00：无符号拓展 01：符号拓展 10：加载到高位，低 16 位补零
EXTOut[31:0]	O	三十二位输出信号

4） COMP

端口	方向	功能
RD1[31:0]	I	比较数 1
RD2[31:0]	I	比较数 2
bcp[2:0]	I	选择比较功能： 3’b000: Zero<=(RD1==RD2) 3’b001: Zero<=(RD1!=RD2)

		3’b010: Zero<=(signed(RD1)<=0) 3’b011: Zero<=(signed(RD1)>0) 3’b100: Zero<=(signed(RD1)<0) 3’b100: Zero<=(signed(RD1)>=0)
Zero	0	比较结果判定： 0：不成立 1：成立

B 型指令比较模块，跳转条件成立输出 1，不成立输出 0

5） B_PC

端口	方向	功能
Pc_add_4[31:0]	I	PC+4
EXToff[31:0]	I	立即数的符号拓展结果
Zero	I	跳转条件判定信号 1:Pc_add_4+(EXToff<<2) 0:PC+8
B_pc[31:0]	0	下一条指令地址：

判定条件信号为一时跳转，否则输出 PC+8（延迟槽）

6） J_PC

端口	方向	功能
Pc[31:0]	I	PC 值
Instr_index[25:0]	I	J 型指令低 26 位
J_pc[31:0]	0	PC[31:28]|Instr_index|00

输出 J 型指令地址

7） MFRSD

端口	方向	功能
RD1[31:0]	I	GRF 的 Rdata1
PC_EX_8[31:0]	I	EX 级 PC+8
AO_MEM[31:0]	I	MEM 级 ALUOut
PC_MEM_8[31:0]	I	MEM 级 PC+8

MUX_WD[31:0]	I	WB 级 MUX_WD 输出
PC_WB_8[31:0]	I	WB 级 PC+8
HI_MEM[31:0]	I	MEM 级 HI 寄存器
LO_MEM[31:0]	I	MEM 级 LO 寄存器
HI_WB[31:0]	I	WB 级 HI 寄存器
LO_WB[31:0]	I	WB 级 LO 寄存器
MFRSDsel[3:0]	I	选择信号: 0000:RD1 0001:PC_EX_8 0010:AO_MEM 0011:PC_MEM_8 0100:MUX_WD 0101:PC_WB_8 0110:HI_MEM 0111:LO_MEM 1000:HI_WB 1001:LO_WB
MFRSDout[31:0]	O	输出结果

ID 级 RS 转发器多选器，用以代替原本数据通路中的 Rdata1

8) MFRTD

端口	方向	功能
RD2[31:0]	I	GRF 的 Rdata2
PC_EX_8[31:0]	I	EX 级 PC+8
AO_MEM[31:0]	I	MEM 级 ALUOut
PC_MEM_8[31:0]	I	MEM 级 PC+8
MUX_WD[31:0]	I	WB 级 MUX_WD 输出
PC_WB_8[31:0]	I	WB 级 PC+8
HI_MEM[31:0]	I	MEM 级 HI 寄存器
LO_MEM[31:0]	I	MEM 级 LO 寄存器

HI_WB[31:0]	I	WB 级 HI 寄存器
LO_WB[31:0]	I	WB 级 LO 寄存器
MFRTDsel[2:0]	I	选择信号: 0000:RD1 0001:PC_EX_8 0010:AO_MEM 0011:PC_MEM_8 0100:MUX_WD 0101:PC_WB_8 0110:HI_MEM 0111:LO_MEM 1000:HI_WB 1001:LO_WB
MFRTDout[31:0]	O	输出结果

ID 级 RT 转发器多选器，用以代替原本数据通路中的 Rdata2

3、 EX 级

1） ID_EX_REG

端口	方向	功能
MFRSD[31:0]	I	MFRSD 转发器结果
MFRTD[31:0]	I	MFRTD 转发器结果
EXT[31:0]	I	立即数拓展结果
Pc_ID[31:0]	I	ID 级 PC
Instr_ID[31:0]	I	ID 级指令码
Reset	I	同步复位
Clk	I	时钟端
RS_EX[31:0]	O	EX 级 RS 数据，对应 MFRSD
RT_EX[31:0]	O	EX 级 RT 数据，对应 MFRTD
EXT_EX[31:0]	O	EX 级立即数拓展结果
Pc_EX[31:0]	O	EX 级 PC

Instr_EX[31:0]	O	EX 级指令码

EX 级流水线寄存器，时钟上升沿到来时更新对应数据，同步复位将 PC 初始化为 0x00003000

2） ALU

端口	方向	功能
A[31:0]	I	第一个操作数
B[31:0]	I	第二个操作数
Op[3:0]	I	操作码： 0000:A&B 0001:A|B 0010:A+B 0011:A-B 0100:A^B\ 0101:~(A|B) 0110:B<<s 0111:B>>s 1000:signed(B)>>>s 1001:signed(A)<signed(B) 1010:A<B
C[31:0]	O	运算结果

3） MUX_ALU_B

端口	方向	功能
RT[31:0]	I	来自 MFRTE 的结果
EXT[31:0]	I	来自 EXT@EX
MUX_ALU_Bsel	I	选择信号： 0:RT 1:EXT
B[31:0]	O	输出结果

4） MUX_ALU_A

端口	方向	功能
MFRSE[31:0]	I	来自 MFRSE 的结果
S[4:0]	I	当前 EX 级指令的 10 到 6 位
MUX_ALU_Asel	I	选择信号 0:MFRSE 1:zero_ext(s)
ALU_A[31:0]	O	输出

5) M_D

端口	方向	功能
start	I	Start 信号
D1[31:0]	I	计算数 1
D2[31:0]	I	计算数 2
sel[2:0]	I	功能选择信号： 000:mult 001:multu 010:div 011:divu 100:mthi 101:mtlo
clk	I	时钟信号
reset	I	复位信号
Busy	O	繁忙信号
HI[31:0]	O	HI 寄存器
LO[31:0]	O	LO 寄存器

M_D 乘除单元模拟真实乘除器，乘法需要 5 个周期，除法需要 10 个，乘除单元运行时只有非乘除类指令才能继续进行

6) MFRSE

端口	方向	功能

RS_EX[31:0]	I	来自 RS@EX
AO_MEM[31:0]	I	MEM 级 ALUOut
PC_MEM_8[31:0]	I	MEM 级 PC+8
MUX_WD[31:0]	I	WB 级 MUX_WD 输出
PC_WB_8[31:0]	I	WB 级 PC+8
HI_MEM[31:0]	I	MEM 级 HI 寄存器
LO_MEM[31:0]	I	MEM 级 LO 寄存器
HI_WB[31:0]	I	WB 级 HI 寄存器
LO_WB[31:0]	I	WB 级 LO 寄存器
MFRSEsel[3:0]	I	选择信号: 0000:RS_EX 0001:AO_MEM 0010:PC_MEM_8 0011:MUX_WD 0100:PC_WB_8 0101:HI_MEM 0110:LO_MEM 0111:HJ_MEM 1000:LO_MEM
MFRSEout[31:0]	O	输出结果

MFRSE 转发器，用以代替 RS@EX

7) MFRTE

端口	方向	功能
RT_EX[31:0]	I	来自 RT@EX
AO_MEM[31:0]	I	MEM 级 ALUOut
PC_MEM_8[31:0]	I	MEM 级 PC+8
MUX_WD[31:0]	I	WB 级 MUX_WD 输出
PC_WB_8[31:0]	I	WB 级 PC+8
HI_MEM[31:0]	I	MEM 级 HI 寄存器

LO_MEM[31:0]	I	MEM 级 LO 寄存器
HI_WB[31:0]	I	WB 级 HI 寄存器
LO_WB[31:0]	I	WB 级 LO 寄存器
MFRTEsel[3:0]	I	选择信号: 0000:RT_EX 0001:AO_MEM 0010:PC_MEM_8 0011:MUX_WD 0100:PC_WB_8 0101:HI_MEM 0110:LO_MEM 0111:HI_MEM 1000:LO_MEM
MFRTEout[31:0]	O	输出结果

MFRTE 转发器，用以代替 RT@EX

4、 MEM 级

1）EX_MEM_REG

端口	方向	功能
ALUout[31:0]	I	EX 级的 ALU 计算结果
MFRTE[31:0]	I	EX 级 MFRTE 的输出
Instr_EX[31:0]	I	EX 级指令码
Pc_EX[31:0]	I	EX 级 PC
hi[31:0]	I	来自 EX 的 HI 寄存器
lo[31:0]	I	来自 EX 的 LO 寄存器
Clk	I	时钟端
Reset	I	同步复位
RT_MEM [31:0]	O	MEM 级 RT 数据，对应 MFRTE
AO_MEM[31:0]	O	MEM 级 ALU 结果
Pc_MEM[31:0]	O	MEM 级 PC

Instr_MEM[31:0]	O	MEM 级指令码
HI_MEM[31:0]	O	MEM 级 HI 寄存器
LO_MEM[31:0]	O	MEM 级 LO 寄存器

MEM 级流水线寄存器，时钟上升沿到来时更新数据，同步复位将 PC 初始化为 0x00003000

2）Wdata_produce

端口	方向	功能
MFRTMout[31:0]	I	来自 MFRTM 的结果
Address[31:0]	I	写入地址信号
Sel[1:0]	I	功能选择信号： 00:sw 01:sh 10:sb
Wdata[31:0]	O	输出写入内存数据

输出信号根据 sel 和地址信号低两位将所要写入的数据放在对应位上，其他位为 0

3）BEproduce

端口	方向	功能
Address[31:0]	I	地址信号
BEsel[1:0]	I	选择信号 00: BE=1111 01: BE=0011/1100 10: BE=0001/0010/0100/1000
BE[3:0]	O	输出 BE 信号，BE[i]=1 表示对应 8 位可写入

用以产生标志信号表示哪些位置可以写入

4）DM

端口名	方向	功能
Address[11:2]	I	选择内存位置
Wdata[31:0]	I	写数据

DMld	I	选择读写操作信号 0：写操作 1：读操作
Reset	I	异步复位
Clk	I	时钟信号
BE[3:0]	I	写入位使能
Pc[31:0]	I	MEM 级 PC 用来 display 输出
Rdata[31:0]	O	读数据

本模块规格为 32bit*1024，截取地址信号的 13 到 2 位作为选择地址

5）MFRTM

端口	方向	功能
RT_MEM[31:0]	I	来自 RT@MEM
MUX_WD[31:0]	I	WB 级 MUX_WD 的结果
PC_WB_8[31:0]	I	WB 级 PC+8
MFRTMsel[1:0]	I	MFRTM 选择信号： 00：RT_MEM 01：MUX_WD 10：PC_WB_8
MFRTMout[31:0]	O	MFRTM 的结果

MEM 级的 MFRTM 转发多选器，用以代替 RT@MEM

5、 WB 级

1）MEM_WB_REG

端口名	方向	功能
RD_MEM[31:0]	I	MEM 级读内存结果
AO_MEM[31:0]	I	MEM 级的 ALU 结果
Insre_MEM[31:0]	I	MEM 的指令码
Pc_MEM[31:0]	I	MEM 级 PC
HI_MEM[31:0]	I	来自 MEM 级的 HI 数据
LO_MEM[31:0]	I	来自 MEM 级的 LO 数据

Clk	I	时钟端
Reset	I	同步复位清零
RD_WB[31:0]	O	WB 级读内存结果
AO_WB[31:0]	O	WB 级 ALU 结果
Instr_WB[31:0]	O	WB 级指令码
Pc_WB[31:0]	O	WB 级 PC
HI_WB[31:0]	O	WB 级 HI 寄存器
LO_WB[31:0]	O	WB 级 LO 寄存器

WB 级流水线寄存器，时钟上升沿到来时更新数据，同步复位将 PC 初始化为 0x00003000

2）Load_ext

端口	方向	功能
address[31:0]	I	来自 WB 级的 AO
Rdata[31:0]	I	来自 WB 级的 RD
loadsel[2:0]	I	选择信号： 000:lw 001:lh 010:lhu 011:lb 100:lbu
out[31:0]	O	Load 拓展后的输出

这个部件主要用于将从内存读出的数据进行截取和拓展

3）MUX_WA

端口名	方向	功能
Instr_WB[31:0]	I	WB 级指令码
MUX_WAsel[1:0]	I	选择信号： 00：rd 01：rt 10：31

MUX_WAout[4:0]	O	输出结果

4）MUX_WD

端口名	方向	功能
AO_WB[31:0]	I	WB 级 AlU 计算结果
RD_WB[31:0]	I	WB 级内存读取数
Instr_WB[31:0]	I	WB 级指令码
Pc_WB[31：0]	I	WB 级 PC
hi[31:0]	I	WB 级的 HI 寄存器
lo[31:0]	I	WB 级的 LO 寄存器
MUX_WDsel[2:0]	I	选择信号 000：AO_WB 001：PC_WB+8 010：RD_WB 011:hi 100:lo
MUX_WDout[31:0]	O	输出结果

5）GRF(WB)

端口名	方向	功能
WA[4:0]	I	写数据地址,来自 MUX_WA
Wdata[31:0]	I	写入数据，来自 MUX_WD
WE	I	写使能
Reset	I	同步复位清零
Clk	I	时钟端
Pc[31:0]	I	WB 级 PC

WB 级的 GRF 部分，时钟上升沿到来时若写使能有效，写入数据，冲突通过 GRF 内部转发实现

二、 控制器设计

端口名	方向	功能
Op[5:0]	I	各流水级指令操作码

Func[5:0]	I	各流水级 R 型指令功能码
EXTsel@ID[1:0]	O	ID 级 EXT 选择信号： 00：无符号 01：有符号 10：加载到高位
MUX_PCsel@ID[1:0]	O	PC 选择信号： 00：PC+8 01：B 型指令地址 10：J 型指令地址 11：jr 跳转地址
bcp@ID[2:0]	O	B 型指令功能选择： 000: beq 001: bne 010: blez 011: bgtz 100: bltz 101: bgez
M_Dsel@EX[2:0]	O	乘除单元功能选择信号： 000: mult 001: multu 010: div 011: divu 100: mthi 101: mtlo
MUX_ALU_Asel@EX	O	EX 级 ALUA 操作数选择信号： 0:MFRSE 1:zero_ext(s)
MUX_ALU_Bsel@EX	O	EX 级 ALUB 操作数选择信号 0：RT

		1：imi_EXT
ALUop@EX[3:0]	0	ALU 运算选择信号： 0000：与 0001：或 0010：加 0011：减 0100: 异或 0101：或非 0110：逻辑左移 0111：逻辑右移 1000：算术右移 1001：有符号小于 1010：无符号小于
DMld@MEM	0	MEM 级读使能 0：写内存 1：读内存
BEsel@MEM[1:0]	0	BE 信号的选择 00:1111 01:0011/1100 10:0001/0010/0100/1000
Wdata_producesel@MEM[1:0]	0	Wdata 扩展选择信号： 00: sw 01: sh 10: sb
Load_extsel@WB[2:0]	0	从内存读取的数据的拓展信号： 000:lw 001:lh 010:lhu 011:lb

		100:lbu
MUX_WAsel@WB[1:0]	O	WB 级 WA 选择信号 00：rd 01：rt 10：31
MUX_WDsel@WB[2:0]	O	WB 级 WD 选择信号 000：AO 001：PC+8 010：RD 011：hi 100：lo
WE@WB	O	WB 级写使能 1：可以写入 0：不可以写入

控制器采用分布译码方式，具体逻辑见附表

三、 暂停器设计

端口名	方向	功能
Instr_ID[31:0]	I	ID 级指令码
Instr_EX[31:0]	I	EX 级指令码
Instr_MEM[31:0]	I	MEM 级指令码
Busy_start	I	乘除单元占用信号
Enable_PC	O	PC 使能信号
Enable_IF_ID	O	ID 流水寄存器使能
Reset_ID_EX	O	EX 流水寄存器复位信号

暂停器逻辑设计具体方式见附表

四、 转发器

端口名	方向	功能

Instr_ID[31:0]	I	ID 级指令码
Instr_EX[31:0]	I	EX 级指令码
Instr_MEM[31:0]	I	MEM 级指令码
Instr_WB[31:0]	I	WB 级指令码
MFRSDsel[3:0]	O	MFRSD 选择信号
MFRTDsel[3:0]	O	MFRTD 选择信号
MFRSEsel[3:0]	O	MFRSE 选择信号
MFRTEsel[3:0]	O	MFRTE 选择信号
MFRTMsel[2:0]	O	MFRTM 选择信号

转发器逻辑设计具体方式见附表

五、 测试代码

```
.text
ori $0, $0, 869
lui $1, 0xCA65
ori $3, $0, 8
ori $2, $1, 61000
addi $1, $1, 138
sub $4, $3, $1
subu $5, $1, $2
add $6, $3, $4
sw $1, -4($3)
sw $2, 4($3)
sh $5, 0($3)
sh $6, 2($3)
jal label4
addu $ra, $ra, $3
nop
ori $7, $0, 4
lb $16, -1($3)
```

```
lw $9, 4($3)
beq $16, $9, label1
lbu $16, -4($3)
xori $7, $7, 0xFF00
label1: beq $16, $16, label2
or $7, $7, $3
ori $7, $7, 2
label2: sll $7, $7, 1
sb $7, 5($3)
j label3
ori $2, $0, 0x8C09
label4: sb $ra, 6($3)
jr $ra
nop
label3:
sra $10, $7, 5
srav $5, $7, $16
sub $9, $0, $7
srlv $11, $9, $16
srl $17, $9, 5
sllv $11, $9, $16
srav $11, $9, $16
sra $17, $9, 5
ori $9, $0, 0xF3F2
nor $2, $2, $9
andi $2, $2, 0x000F
div $5, $2
mflo $5
mfhi $6
xor $5, $5, $6
```

```
mtlo $6
addiu $7, $5, -127
mthi $5
mflo $5
mfhi $6
div $17, $9
mflo $5
mfhi $6
divu $17, $9
mflo $5
mfhi $6
mult $17, $9
mtlo $6
mflo $5
mfhi $6
multu $17, $9
mflo $5
mfhi $6
sub $18, $0, $3
sw $18, 8($3)
lhu $18, 8($3)
blez $18, label5
nop
jal label6
addi $20, $ra, -20
label6: jalr $21, $20
sh $0, 8($3)
label5: lh $18, 10($3)
bne $2, $2, label7
slt $19, $11, $0
```

```
ori $7, $0, 1
label7: bne $2, $18, label8
sltu $19, $11, $0
ori $7, $7, 2
label8: bgez $18, label9
slti $19, $9, -0x8000
ori $7, $7, 4
label9: bgez $0, label10
sltiu $19, $9, -0x8000
ori $7, $7, 8
label10: lui $22, 0x4000
beq $0, $0, label11
add $22, $22, $22
ori $23, $0, 1
label11: lui $23, 0x2000
addu $23, $23, $22
addu $23, $23, $22
sub $24, $23, $22
srl $25, $24, 30
sw $24, 0($24)
lw $25, -8($25)
lw $25, -8($25)
lw $25, -8($25)
loop: beq $0, $0, loop
nop
.ktext 0x4180
mfc0 $29, $12
mfc0 $30, $13
andi $30, $30, 0x7C
ori $28, $0, 48
```

```
bne $30, $28, eov
nop
srl $22, $22, 1
eov: ori $28, $0, 20
bne $30, $28, eas
nop
sll $24, $0, 0
eas: ori $28, $0, 16
bne $30, $28, eal
nop
sll $25, $25, 1
mfc0 $27, $14
addi $27, $27, 4
sw $27, 0x2FFC($0)
lw $28, 0x2FFC($0)
mtc0 $28, $14
eal: eret
ori $1, $0, 237
```

六、 思考题

1、 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”？

操作系统内核和系统调用层合起来就是操作系统，正是操作系统连接了软件和硬件，成为了软件和硬件的接口

2、 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。

应该属于外设部分，通过系统桥连接

3、 BE 部件对所有的外设都是必要的吗？

不是，仅仅对于 DM 是需要的对于 Timer 不需要

4、 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：

```
.text
mfc0 $1 $12
ori $1 $1 0xfc01
mtc0 $1 $12
li $5 1
li $6 0x00007f00
li $7 9
li $8 2
sw $7 0($6)
sw $8 4($6)
label:
addi $5 $5 1
addi $5 $5 1
addi $5 $5 1
jal label
addi $5 $5 1
.ktext 0x4180
mfc0 $26 $13
mfc0 $27 $14
```

```
nop
nop
sw $7 0($6)
eret
```

5、 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？

键盘和鼠标动作时产生一个中断信号，cpu 进入处理程序