CPU 设计文档

1.CPU 模块和数据通路

1.1 CP0 端口定义及其功能说明

CP0 端口定义

信号	方向	描述
reset	I	同步复位信号
clk	I	时钟信号
add	I	五位内部寄存器地址
wd	I	32 位写入数据数据
rd	O	32 位置读出数据
we	I	CP0 写使能信号
I_EPC	I	下一条传入 CP0 指令的 PC
O_EPC	O	CP0 内的 EPC 值
O_SR	O	CP0 内 SR 寄存器的数值
we1	I	中断/异常有效信号
bd	I	下一条指令的 bd 位置
ex	I	下一条指令的 excode
hwint	I	6 位当前中断信号
we2	I	eret 使能信号

CP0 功能定义

序号	功能名称	功能描述
1	复位	当时钟上升沿时若 reset 信号有效是 PC 置为 0
2	写数据	时钟上升沿时若 we 信号有效则写入数据
3	响应中断/异常	当时钟上升沿到来时若异常中断信号有效，则存入 I_EPC 和 ex 且 exl 位置为 1
4	响应 eret	若 we2 信号有效则响应 eret，exl 位置为 0

说明：信号优先级：reset>(we1=we2)>we

1.1 BRIDGE 端口定义及其功能说明

BRIDGE 端口定义

信号	方向	描述
add	I	32 位设备地址
wd	I	32 位写入数据
rd0	I	第一个外设读入的 32 位数据
rd1	I	第二个外设读入的 32 位数据
rd2	I	第三个外设读入的 32 位数据
hit0	O	第一个设备的地址范围命中信号
hit1	O	第二个设备的地址范围命中信号
hit2	O	第三个设备的地址范围命中信号
dev_rd	O	32 位设备读出数据

BRIDGE 功能定义

序号	功能名称	功能描述
1	设备命中选择	根据传入地址判断命中的设备
2	设备读出数据选择	根据命中情况将从设备中传出数据有选择地传入 CPU，若都为命中，则传入 0

2. 测试

测试代码:

1：

```
block0:
bne $2 $0 block1
nop
ori $2 1
ori $31 $0 0x3000
addi $31 $31 -8
jr $31
nop


block1:
ori $2 $0 0x7f10 #ctrl
ori $3 $0 14
sw  $3 ($2)
addi $2 $2 4 #preset
sw $3 ($2)
addi $2 $2 4 #count
sw $3 ($2)
addi $2 $2 -4
lw $4 ($2)


block2:
ori $2 $0 3
lw $5 ($2)
lh $5 ($2)
lhu $5 ($2)
lb $5 ($2)
lbu $5 ($2)
sw $5 ($2)
sh $5 ($2)
sb $5 ($2)
```

```
ori $2 $0 0
lui $2 0x8000
sw $2 -1($2)

block3:
lui $2  0x8000
lui $3 0x7fff
ori $3 0xffff
sub $2 $2 $3
add $3 $3 $3

block4:
nop




.text 0x4180
mfc0 $10 $14
addi $10 $10 4
mtc0 $10 $14
eret
addu $5 $5 $5
```

2：

```
    .ktext 0x4180
_entry:
 mfc0   $k0, $14
 mfc0   $k1, $13
 ori $k0, $0, 0x1000
 sw  $sp, -4($k0)

 addiu  $k0, $k0, -256
 move   $sp, $k0

 j   _save_context
 nop

_main_handler:
 mfc0   $k0, $13
 ori    $k1, $0, 0x007c
 and $k0, $k1, $k0
 beq    $0, $k0, _restore_context
 nop
 mfc0   $k0, $14
 addu   $k0, $k0, 4
 mtc0   $k0, $14
 j   _restore_context
 nop

_restore:
 eret

_save_context:
 sw     $1, 4($sp)
    sw     $2, 8($sp)
    sw     $3, 12($sp)
    sw     $4, 16($sp)
    sw     $5, 20($sp)
    sw     $6, 24($sp)
    sw     $7, 28($sp)
    sw     $8, 32($sp)
    sw     $9, 36($sp)
    sw     $10, 40($sp)
    sw     $11, 44($sp)
    sw     $12, 48($sp)
    sw     $13, 52($sp)
    sw     $14, 56($sp)
    sw     $15, 60($sp)
    sw     $16, 64($sp)
    sw     $17, 68($sp)
    sw     $18, 72($sp)
    sw     $19, 76($sp)
    sw     $20, 80($sp)
    sw     $21, 84($sp)
    sw     $22, 88($sp)
    sw     $23, 92($sp)
    sw     $24, 96($sp)
    sw     $25, 100($sp)
    sw     $26, 104($sp)
    sw     $27, 108($sp)
    sw     $28, 112($sp)
    sw     $29, 116($sp)
    sw     $30, 120($sp)
    sw     $31, 124($sp)
 mfhi   $k0
 sw $k0, 128($sp)
 mflo   $k0
 sw  $k0, 132($sp)
 j   _main_handler
 nop



_restore_context:
 lw  $1, 4($sp)
    lw     $2, 8($sp)
    lw     $3, 12($sp)
    lw     $4, 16($sp)
    lw     $5, 20($sp)
    lw     $6, 24($sp)
    lw     $7, 28($sp)
    lw     $8, 32($sp)
    lw     $9, 36($sp)
    lw     $10, 40($sp)
    lw     $11, 44($sp)
    lw     $12, 48($sp)
    lw     $13, 52($sp)
    lw     $14, 56($sp)
    lw     $15, 60($sp)
    lw     $16, 64($sp)
    lw     $17, 68($sp)
    lw     $18, 72($sp)
    lw     $19, 76($sp)
    lw     $20, 80($sp)
    lw     $21, 84($sp)
    lw     $22, 88($sp)
    lw     $23, 92($sp)
    lw     $24, 96($sp)
    lw     $25, 100($sp)
    lw     $26, 104($sp)
    lw     $27, 108($sp)
    lw     $28, 112($sp)
    lw     $29, 116($sp)
    lw     $30, 120($sp)
    lw     $31, 124($sp)
 lw  $k0, 128($sp)
 mthi   $k0
 lw  $k0, 132($sp)
 mtlo   $k0
    j   _restore
 nop
```

```
.text
 ori    $2, $0, 0x1001
    mtc0   $2, $12
 ori $28, $0, 0x0000
 ori $29, $0, 0x0000
 lui $8, 0x7fff
 lui $9, 0x7fff
 lw $1,0($0)
 beq $0,$1,end
 add $10,$8,$9
 nop
 nop

end:
 beq $0, $0, end
 nop
```

实际结果：

1：

```
 46@00003008: $ 2 <= 00000001
 50@0000300c: $31 <= 00003000
 54@00003010: $31 <= 00002ff8
 86@00004180: $10 <= 00002ff8
 94@00004184: $10 <= 00002ffc
134@00004180: $10 <= 00002ffc
142@00004184: $10 <= 00003000
174@0000301c: $ 2 <= 00007f10
178@00003020: $ 3 <= 0000000e
186@00003028: $ 2 <= 00007f14
194@00003030: $ 2 <= 00007f18
214@00004180: $10 <= 00003034
222@00004184: $10 <= 00003038
246@00003038: $ 2 <= 00007f14
250@0000303c: $ 4 <= 0000000e
254@00003040: $ 2 <= 00000003
274@00004180: $10 <= 00003044
282@00004184: $10 <= 00003048
322@00004180: $10 <= 00003048
```

```
330@00004184: $10 <= 0000304c
370@00004180: $10 <= 0000304c
378@00004184: $10 <= 00003050
402@00003050: $ 5 <= 00000000
406@00003054: $ 5 <= 00000000
426@00004180: $10 <= 00003058
434@00004184: $10 <= 0000305c
474@00004180: $10 <= 0000305c
482@00004184: $10 <= 00003060
502@00003060: *00000000 <= 00000000
510@00003064: $ 2 <= 00000000
514@00003068: $ 2 <= 80000000
534@00004180: $10 <= 0000306c
542@00004184: $10 <= 00003070
566@00003070: $ 2 <= 80000000
570@00003074: $ 3 <= 7fff0000
574@00003078: $ 3 <= 7fffffff
594@00004180: $10 <= 0000307c
602@00004184: $10 <= 00003080
642@00004180: $10 <= 00003080
650@00004184: $10 <= 00003084
```

2：

```
 38@00003000: $ 2 <= 00001001
 46@00003008: $28 <= 00000000
 50@0000300c: $29 <= 00000000
 54@00003010: $ 8 <= 7fff0000
 58@00003014: $ 9 <= 7fff0000
 62@00003018: $ 1 <= 00000000
 82@00004180: $26 <= 0000301c
 86@00004184: $27 <= 00001000
 90@00004188: $26 <= 00001000
 90@0000418c: *00000ffc <= 00000000
 98@00004190: $26 <= 00000f00
102@00004194: $29 <= 00000f00
110@000041d4: *00000f04 <= 00000000
114@000041d8: *00000f08 <= 00001001
118@000041dc: *00000f0c <= 00000000
122@000041e0: *00000f10 <= 00000000
126@000041e4: *00000f14 <= 00000000
130@000041e8: *00000f18 <= 00000000
134@000041ec: *00000f1c <= 00000000
138@000041f0: *00000f20 <= 7fff0000
142@000041f4: *00000f24 <= 7fff0000
146@000041f8: *00000f28 <= 00000000
```

```
150@000041fc: *00000f2c <= 00000000
154@00004200: *00000f30 <= 00000000
158@00004204: *00000f34 <= 00000000
162@00004208: *00000f38 <= 00000000
166@0000420c: *00000f3c <= 00000000
170@00004210: *00000f40 <= 00000000
174@00004214: *00000f44 <= 00000000
178@00004218: *00000f48 <= 00000000
182@0000421c: *00000f4c <= 00000000
186@00004220: *00000f50 <= 00000000
190@00004224: *00000f54 <= 00000000
194@00004228: *00000f58 <= 00000000
198@0000422c: *00000f5c <= 00000000
202@00004230: *00000f60 <= 00000000
206@00004234: *00000f64 <= 00000000
210@00004238: *00000f68 <= 00000f00
214@0000423c: *00000f6c <= 00001000
218@00004240: *00000f70 <= 00000000
222@00004244: *00000f74 <= 00000f00
226@00004248: *00000f78 <= 00000000
230@0000424c: *00000f7c <= 00000000
238@00004250: $26 <= 00000000
238@00004254: *00000f80 <= 00000000
246@00004258: $26 <= 00000000
246@0000425c: *00000f84 <= 00000000
262@000041a0: $26 <= 00001000
266@000041a4: $27 <= 0000007c
270@000041a8: $26 <= 00000000
286@00004268: $ 1 <= 00000000
290@0000426c: $ 2 <= 00001001
294@00004270: $ 3 <= 00000000
298@00004274: $ 4 <= 00000000
302@00004278: $ 5 <= 00000000
306@0000427c: $ 6 <= 00000000
310@00004280: $ 7 <= 00000000
314@00004284: $ 8 <= 7fff0000
318@00004288: $ 9 <= 7fff0000
322@0000428c: $10 <= 00000000
326@00004290: $11 <= 00000000
330@00004294: $12 <= 00000000
334@00004298: $13 <= 00000000
338@0000429c: $14 <= 00000000
342@000042a0: $15 <= 00000000
346@000042a4: $16 <= 00000000
```

```
350@000042a8: $17 <= 00000000
354@000042ac: $18 <= 00000000
358@000042b0: $19 <= 00000000
362@000042b4: $20 <= 00000000
366@000042b8: $21 <= 00000000
370@000042bc: $22 <= 00000000
374@000042c0: $23 <= 00000000
378@000042c4: $24 <= 00000000
382@000042c8: $25 <= 00000000
386@000042cc: $26 <= 00000f00
390@000042d0: $27 <= 00001000
394@000042d4: $28 <= 00000000
398@000042d8: $29 <= 00000f00
406@000042dc: $30 <= 00000000
410@000042e0: $31 <= 00000000
414@000042e4: $26 <= 00000000
430@000042ec: $26 <= 00000000
482@00004180: $26 <= 0000301c
486@00004184: $27 <= 80000030
490@00004188: $26 <= 00001000
490@0000418c: *00000ffc <= 00000f00
498@00004190: $26 <= 00000f00
502@00004194: $29 <= 00000f00
510@000041d4: *00000f04 <= 00000000
514@000041d8: *00000f08 <= 00001001
518@000041dc: *00000f0c <= 00000000
522@000041e0: *00000f10 <= 00000000
526@000041e4: *00000f14 <= 00000000
530@000041e8: *00000f18 <= 00000000
534@000041ec: *00000f1c <= 00000000
538@000041f0: *00000f20 <= 7fff0000
542@000041f4: *00000f24 <= 7fff0000
546@000041f8: *00000f28 <= 00000000
550@000041fc: *00000f2c <= 00000000
554@00004200: *00000f30 <= 00000000
558@00004204: *00000f34 <= 00000000
562@00004208: *00000f38 <= 00000000
566@0000420c: *00000f3c <= 00000000
570@00004210: *00000f40 <= 00000000
574@00004214: *00000f44 <= 00000000
578@00004218: *00000f48 <= 00000000
582@0000421c: *00000f4c <= 00000000
586@00004220: *00000f50 <= 00000000
590@00004224: *00000f54 <= 00000000
```

```
594@00004228: *00000f58 <= 00000000
598@0000422c: *00000f5c <= 00000000
602@00004230: *00000f60 <= 00000000
606@00004234: *00000f64 <= 00000000
610@00004238: *00000f68 <= 00000f00
614@0000423c: *00000f6c <= 80000030
618@00004240: *00000f70 <= 00000000
622@00004244: *00000f74 <= 00000f00
626@00004248: *00000f78 <= 00000000
630@0000424c: *00000f7c <= 00000000
638@00004250: $26 <= 00000000
638@00004254: *00000f80 <= 00000000
646@00004258: $26 <= 00000000
646@0000425c: *00000f84 <= 00000000
662@000041a0: $26 <= 80000030
666@000041a4: $27 <= 0000007c
670@000041a8: $26 <= 00000030
686@000041b4: $26 <= 0000301c
690@000041b8: $ 1 <= 00000000
694@000041bc: $ 1 <= 00000004
698@000041c0: $26 <= 00003020
714@00004268: $ 1 <= 00000000
718@0000426c: $ 2 <= 00001001
722@00004270: $ 3 <= 00000000
726@00004274: $ 4 <= 00000000
730@00004278: $ 5 <= 00000000
734@0000427c: $ 6 <= 00000000
738@00004280: $ 7 <= 00000000
742@00004284: $ 8 <= 7fff0000
746@00004288: $ 9 <= 7fff0000
750@0000428c: $10 <= 00000000
754@00004290: $11 <= 00000000
758@00004294: $12 <= 00000000
762@00004298: $13 <= 00000000
766@0000429c: $14 <= 00000000
770@000042a0: $15 <= 00000000
774@000042a4: $16 <= 00000000
778@000042a8: $17 <= 00000000
782@000042ac: $18 <= 00000000
786@000042b0: $19 <= 00000000
790@000042b4: $20 <= 00000000
794@000042b8: $21 <= 00000000
798@000042bc: $22 <= 00000000
802@000042c0: $23 <= 00000000
```

```
 806@000042c4: $24 <= 00000000
 810@000042c8: $25 <= 00000000
 814@000042cc: $26 <= 00000f00
 818@000042d0: $27 <= 80000030
 822@000042d4: $28 <= 00000000
 826@000042d8: $29 <= 00000f00
 834@000042dc: $30 <= 00000000
 838@000042e0: $31 <= 00000000
 842@000042e4: $26 <= 00000000
 858@000042ec: $26 <= 00000000
 906@00004180: $26 <= 00003020
 910@00004184: $27 <= 00000030
 914@00004188: $26 <= 00001000
 914@0000418c: *00000ffc <= 00000f00
 922@00004190: $26 <= 00000f00
 926@00004194: $29 <= 00000f00
 934@000041d4: *00000f04 <= 00000000
 938@000041d8: *00000f08 <= 00001001
 942@000041dc: *00000f0c <= 00000000
 946@000041e0: *00000f10 <= 00000000
 950@000041e4: *00000f14 <= 00000000
 954@000041e8: *00000f18 <= 00000000
 958@000041ec: *00000f1c <= 00000000
 962@000041f0: *00000f20 <= 7fff0000
 966@000041f4: *00000f24 <= 7fff0000
 970@000041f8: *00000f28 <= 00000000
 974@000041fc: *00000f2c <= 00000000
 978@00004200: *00000f30 <= 00000000
 982@00004204: *00000f34 <= 00000000
 986@00004208: *00000f38 <= 00000000
 990@0000420c: *00000f3c <= 00000000
 994@00004210: *00000f40 <= 00000000
 998@00004214: *00000f44 <= 00000000
1002@00004218: *00000f48 <= 00000000
1006@0000421c: *00000f4c <= 00000000
1010@00004220: *00000f50 <= 00000000
1014@00004224: *00000f54 <= 00000000
1018@00004228: *00000f58 <= 00000000
1022@0000422c: *00000f5c <= 00000000
1026@00004230: *00000f60 <= 00000000
1030@00004234: *00000f64 <= 00000000
1034@00004238: *00000f68 <= 00000f00
1038@0000423c: *00000f6c <= 00000030
1042@00004240: *00000f70 <= 00000000
```

```
1046@00004244: *00000f74 <= 00000f00
1050@00004248: *00000f78 <= 00000000
1054@0000424c: *00000f7c <= 00000000
1062@00004250: $26 <= 00000000
1062@00004254: *00000f80 <= 00000000
1070@00004258: $26 <= 00000000
1070@0000425c: *00000f84 <= 00000000
1086@000041a0: $26 <= 00000030
1090@000041a4: $27 <= 0000007c
1094@000041a8: $26 <= 00000030
1110@000041b4: $26 <= 00003020
1114@000041b8: $ 1 <= 00000000
1118@000041bc: $ 1 <= 00000004
1122@000041c0: $26 <= 00003024
1138@00004268: $ 1 <= 00000000
1142@0000426c: $ 2 <= 00001001
1146@00004270: $ 3 <= 00000000
1150@00004274: $ 4 <= 00000000
1154@00004278: $ 5 <= 00000000
1158@0000427c: $ 6 <= 00000000
1162@00004280: $ 7 <= 00000000
1166@00004284: $ 8 <= 7fff0000
1170@00004288: $ 9 <= 7fff0000
1174@0000428c: $10 <= 00000000
1178@00004290: $11 <= 00000000
1182@00004294: $12 <= 00000000
1186@00004298: $13 <= 00000000
1190@0000429c: $14 <= 00000000
1194@000042a0: $15 <= 00000000
1198@000042a4: $16 <= 00000000
1202@000042a8: $17 <= 00000000
1206@000042ac: $18 <= 00000000
1210@000042b0: $19 <= 00000000
1214@000042b4: $20 <= 00000000
1218@000042b8: $21 <= 00000000
1222@000042bc: $22 <= 00000000
1226@000042c0: $23 <= 00000000
1230@000042c4: $24 <= 00000000
1234@000042c8: $25 <= 00000000
1238@000042cc: $26 <= 00000f00
1242@000042d0: $27 <= 00000030
1246@000042d4: $28 <= 00000000
1250@000042d8: $29 <= 00000f00
1258@000042dc: $30 <= 00000000
```

```
                1262@000042e0: $31 <= 00000000
                1266@000042e4: $26 <= 00000000
                1282@000042ec: $26 <= 00000000
```

补充：

```
@java -jar D:\mar_s\Mars4_4.jar nc me a  mc CompactDataAtZero
^
dump 0x3000-0x3ffc HexText ^
E:\university\computer_struct\P0-P\P7\CPU\code.txt ^
mips1.asm


@java -jar D:\mar_s\Mars4_4.jar nc me a mc CompactDataAtZero
^
dump 0x4180-0x4fffc  HexText ^
E:\university\computer_struct\P0-P\P7\CPU\code_handler.txt ^
mips1.asm
```

测试时自动 assembled

3. 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”？

软件

在接口之上是中断处理程序和用于不同设备的设备驱动程序，在驱动之下为硬盘等硬件。而“硬件/软件接口”是指的是一套体系和结构，它使得硬件和软件只需要依照指令集体系结构设计即可。

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。

应该在 CPU 的外部。

3. BE 部件对所有的外设都是必要的吗？

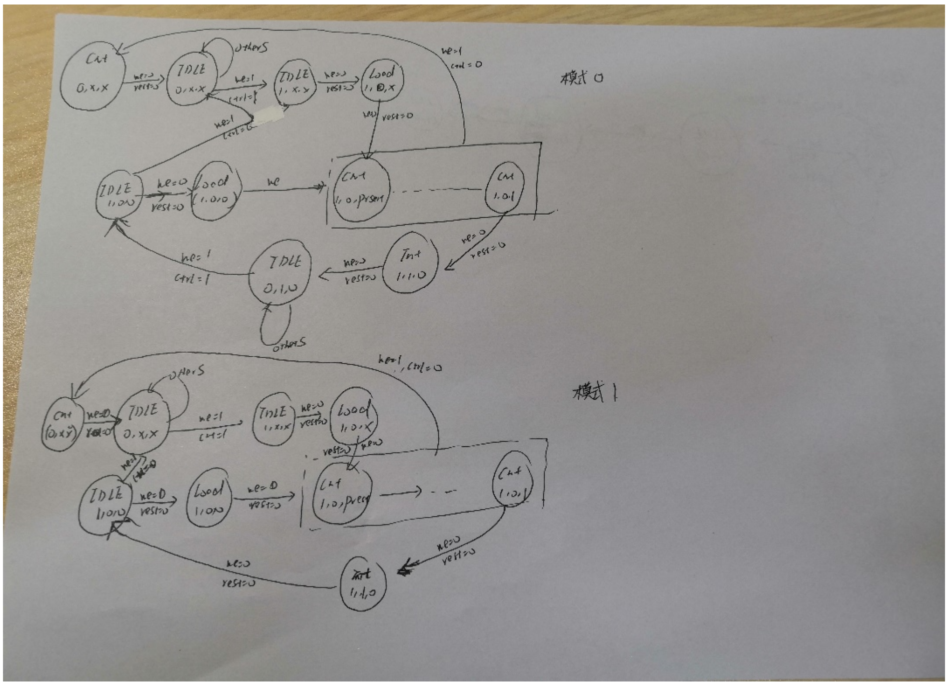
不是必要的，对于按照字读取的外设是没有必要的。

4. 阅读官方提供的定时器源代码，阐述两种中断模式的异同，并分别针对每一种模式绘制状态转移图。

同：都是自减型记时间，且到达 0 时产生中断信号

异：0 模式下产生中断后会把 ctrl 信号自动置为 0，需要外部写入 ctrl 信号来

重新 Load,1 模式下则会自动装载。



两种模式状态转移

5.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

a)定时器在主程序中被初始化为模式 0;

b)定时器倒计数至 0 产生中断;

c)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。

d)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。)

代码如下

```
.text
ori $3 0x7f00 #ctrl
ori $2 0x9
sw $2 0($3)

addi $3 $3 4 #present
ori $4 $0 5
sw $4 0($3)

ori $4 $0 0xfc02 #SR
mtc0 $4 $12

loop:
j loop
```

```
nop


.ktext 0x4180
ori $3 $0 0x7f00
sw $2 0($3)
eret
```

6.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。

通过产生相应的中断信号，然后CPU来进行处理