

103 上學期計算機組織實驗課:LAB4 參考解答

By: 許冠傑 2014/11/8

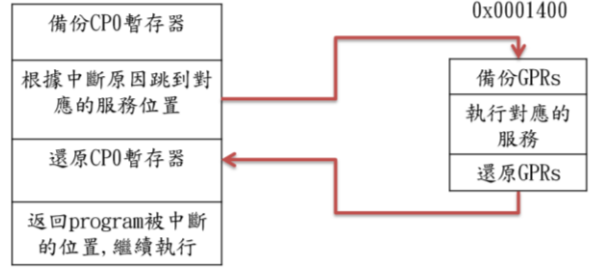
本次實驗要求: 解釋程式碼, 心得, 貼波形圖(optional)

inhandle.s

```

mfc0 $k0,$14 //將 coprocessor 14 號暫存器(EPC)搬到 GPR $k0
sw $k0,4($0) //將其存到記憶體空間 4($0)
mfc0 $k0,$13 //將 13 號 Cause 搬到 GPR $k0
sw $k0,8($0)
mfc0 $k0,$12 //將 12 號 Status 搬到 GPR $k0
sw $k0,12($0)
mfc0 $k0,$13
li $k1,4096 //載入立即值 0d4096
add $k1,$k0,$k1
jalr $k1

lw $k0,4($0)
mtc0 $k0,$14
lw $k0,8($0)
mtc0 $k0,$13
lw $k0,12($0)
mtc0 $k0,$12 //還原 CPO
li $k1,64513
mtc0 $k1,$12
mfc0 $k0,$14 //載入 EPC
jr $k0 //跳回原程式
    
```



Address	Memory
4(\$0)	EPC
8(\$0)	Cause
12(\$0)	Status

前六行指令執行後會使得記憶體區段出現以上結果。此為備份行為。

認識Coprocesor0部分暫存器

- > CPO Registers include status, cause, EPC ... etc
- The processor is running in Kernel Mode or Debug Mode.

Register Number	Sel ¹	Register Name	Function
12	0	Status	Processor status and control
13	0	Cause	Cause of last general exception
14	0	EPC	Program counter at last exception

jalr \$k1 行為: 將 PC+4 存到 \$ra (return address), 再把 \$k1 存到 PC 裡。程式會跳到指定 PC 值去執行。為呼叫其他函式的做法, 且有 return address 可以回到下一行 lw \$k0,4(\$0) 繼續做。

\$k1 : 固定位址 0d4096 再加上不同 exception 原因(cause's value), 那個位址的服務程式可以做相對應的處理。

0d 64513 (高位補 0)
 =0x FC01
 =0b 1111110000000001

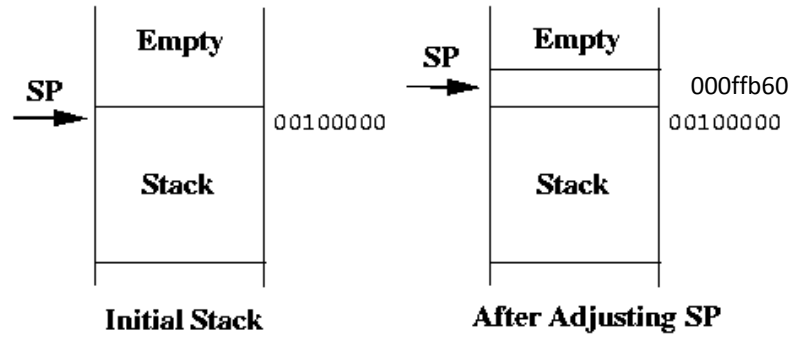
將 coprocessor 12 號 status 暫存器還原成預設值
 ([15:10]設成 enable for HW, [0]設成 enable for SW)

31	28	27	26	25	24	23	22	21	20	19	18	17	16	15	10	9	8	7	6	5	4	3	2	1	0
CU3..CU0	RP	FR	RE	MX	0	BEV	TS	SR	NMI	ASE	Impl	IM7..IM2	IM1..IM0	0	UM	R0	ERL	EXL	IE						
														IPL										KSU	

intprocess.s

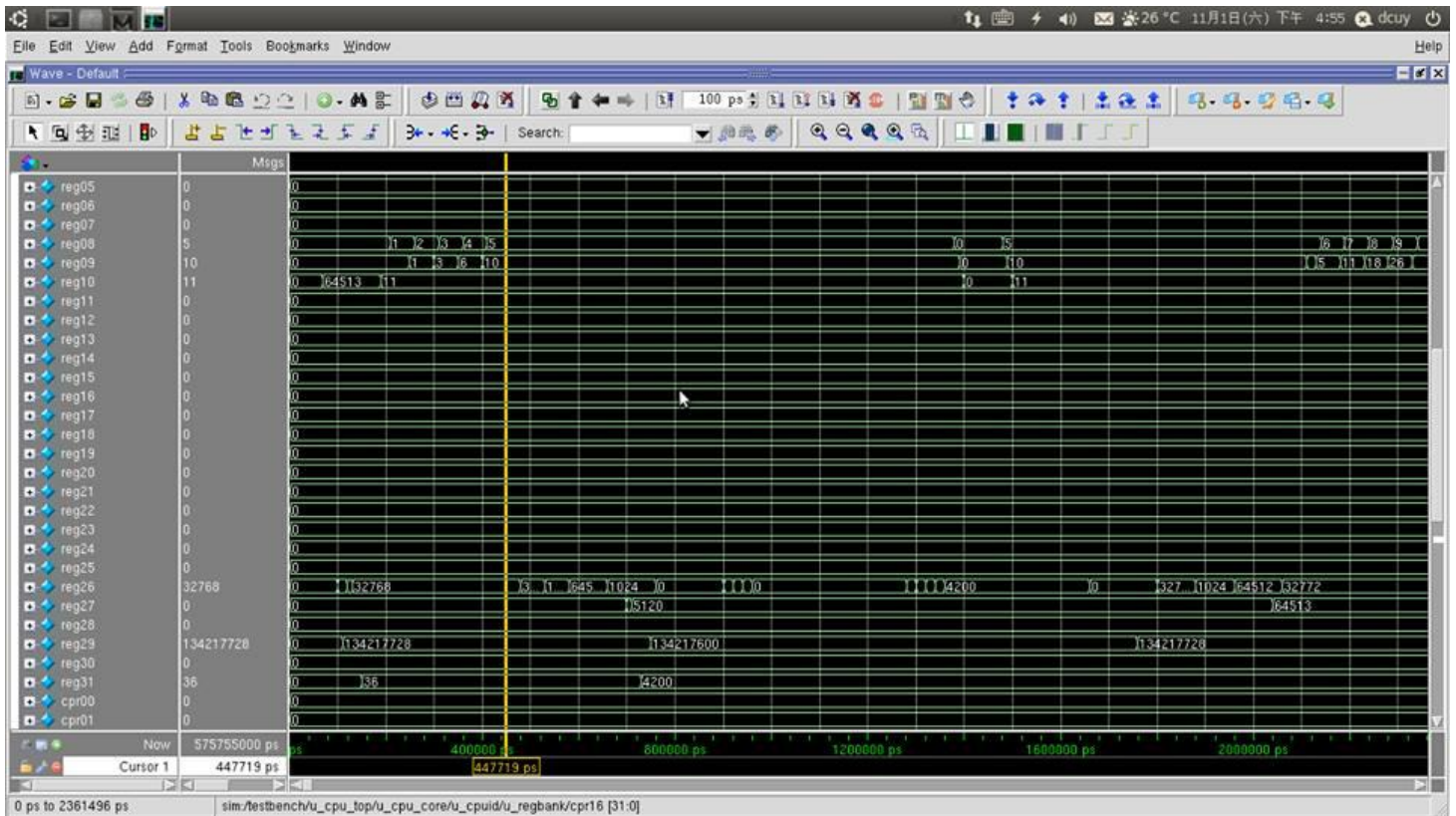
```

addiu $sp,$sp,-128 //獲取記憶體空間128 bytes
move $k0,$1
sw $k0,0($sp) //依序儲存GPRs到此記憶體空間
...
move $k0,$31
sw $k0,120($sp) //備份GPRs完成
move $8,$0
move $9,$0
move $10,$0
lw $1,0($sp) //還原GPRs
...
lw $31,120($sp) //還原GPRs完成
addiu $sp,$sp,128 //釋放記憶體空間
jr $31 //跳回inhandle.s
    
```



服務程式的內容: 我們僅預設為清空三個暫存器以方便驗證。實際上為中斷處理需要實現的部分。

執行結果波型圖(第一組同學的圖):



請參考 LAB4 投影片的詳細解說會更加了解本次實驗內容。有任何問題請再告訴助教。