**Description of the VPortYieldISR function**

1. **Registers Description**

**k0:k1** : Interrupt/Trap handler registers: Reserved for use by interrupt/trap handler.

**s0:s7** : Subroutine register variables : A subroutine that writes one of these must save the old value and restore it before it exists, so the calling routine sees the values preserved.

**s8** : Ninth subroutine register variable: Can either be used as above or as the frame pointer for functions that require one.

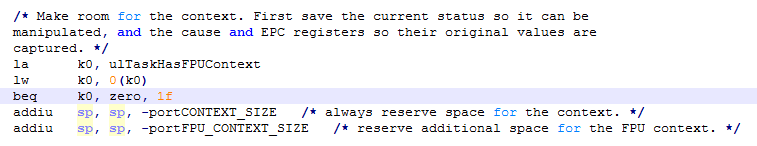
**t0:t9** : Subroutine register temporaries: Registers that subroutines can use without saving.

**a0:a3** : Argument registers: First few parameters for a subroutine.

**v0:v1** : Return value registers: Value returned by subroutine

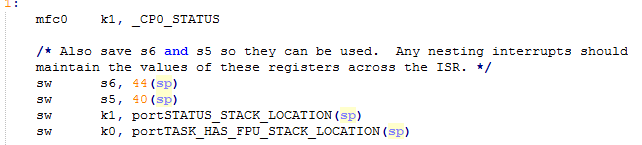
**zero** : Zero register: Always returns 0

**sp** : Stack pointer register: Stack pointer

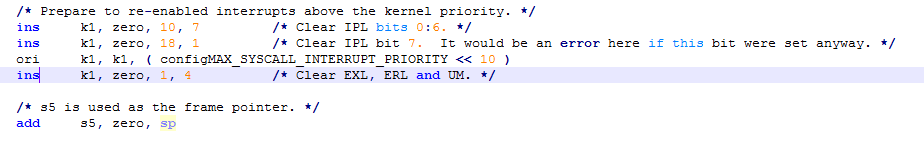


Load k0 at his original value  
If ko is equal to 0,

Move the Stack Pointer to give some space for the context

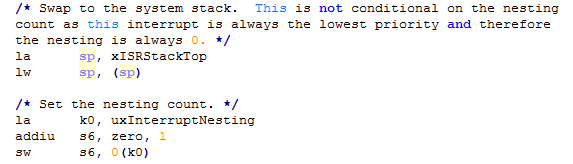


S6 is saved at sp +44 , s5 at sp +40  
k1 and k0 are saved too



“ins” = Insert Bit field. Here, it consist to replace some bits by 0.

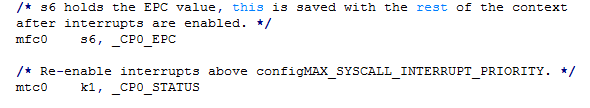
“ori” = Or immediate. Logical bitwise OR between k1 and “configMAX\_ … “. Result placed in k1.



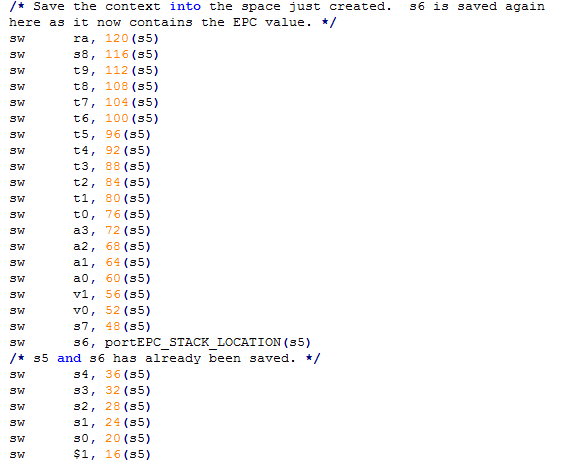
“la” : load address

“lw” : load word

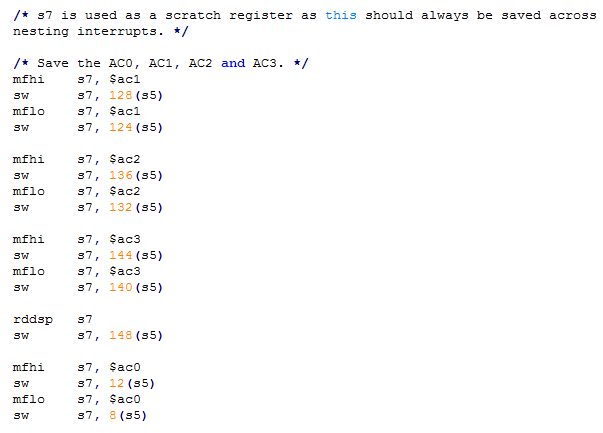
Save “uxInterruptNesting” in “s6”.



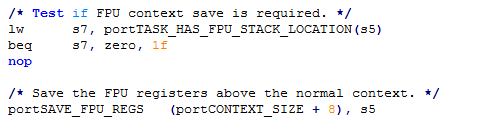
Mfc0 permit to import value in a coprocessor register to a work register. Mtc0 is the opposite.



Those registers are saved in order to use them for work (in case of modification)

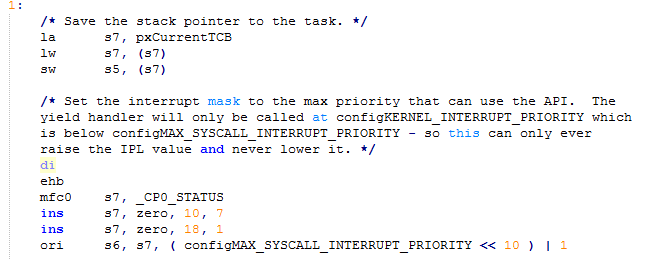


“mfhi/mhlo” : Move from HI/LO register



The beq instruction is used to pass the FPU context saving if he is not necessary.

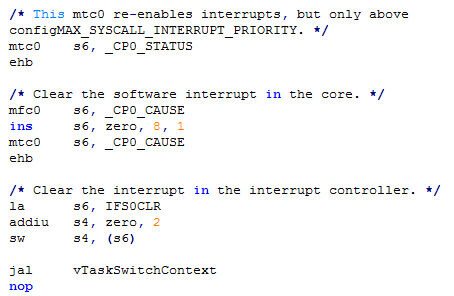
If it is needed, the the following instructions are executed and the necessary information about the FPU context are saved.



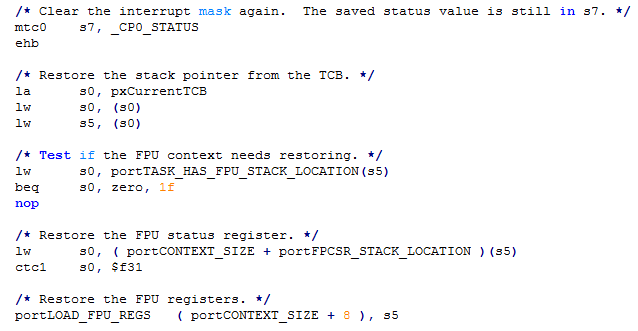
“di” : disable interrupts

“ehb” : Execution Hazard Barrier = stop instruction execution

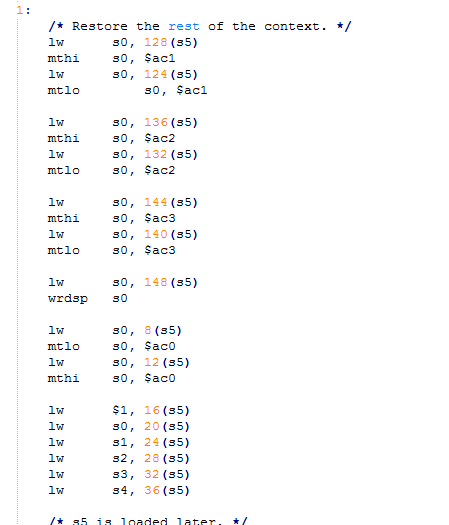
“ins” is used to clear the same bit as earlier.



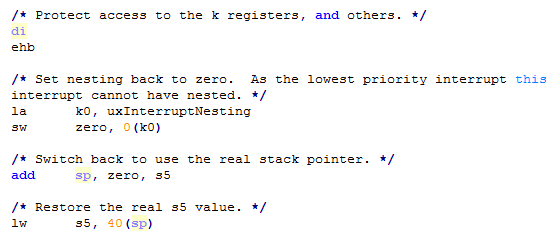
“jal” = jump and link. So the procedure “vTaskSwitchContext” is executed and then we come back at the next instructions.



If the FPU context doesn’ t need restoring, the 2 following restoring are not executed.



The context saved precedently is now restored.



End of the restoring.

