

# Lab 6-7 Quiz Solution Guide

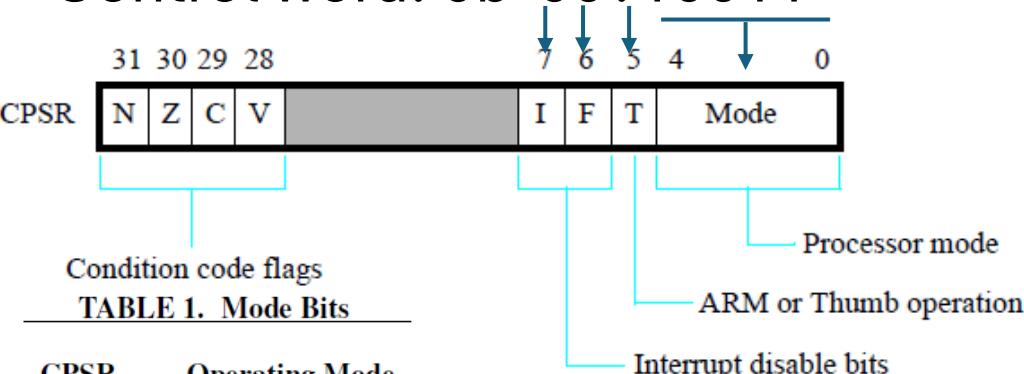
Interrupt, KEY, and Timer

# Mode selection

This is a special Register: **CPSR**

Use MOV to write on general purpose registers, but use **MSR** (move status register) to change CPSR.

Control word: 0b 00?10011



CPSR<sub>4-0</sub> Operating Mode

10000	User
10001	FIQ
10010	IRQ
10011	Supervisor

```
MOV R0, #0b00010011  
MSR CPSR_c, R0
```

# Stack Pointers

Use different stacks in supervisor and IRQ modes, this way you can use the stack in both modes safely.

SP is also a register, but now we try to write a 32-bit number there.

LDR SP, =a\_word\_address

# Enable interrupts

Processor



Interrupt Controller



Device



1. Disable interrupts, because you don't want interruption while you are configuring the devices.
2. Tell the device to send you interrupt requests.  
See device interface for the address and data.
3. Tell the GIC to let you see the requests that device. Enable GIC
4. Enable Interrupts

# Device

Example: KEY

1. Device requests interrupt.

# Interrupt Requested!

## Interrupt Controller



2. GIC checks its priority etc. and then requests interrupt.

## Processor



3. Processor goes to IRQ mode. Checks who requested interrupt and service with a subroutine.

LDR	R5, [R4]	// Read interrupt ID from ICCIAR into R5
CMP	R5, #73	// Compare with KEY interrupt ID
BLEQ	KEY_ISR	// If equal, branch to KEY_ISR
CMP	R5, #29	// Compare with Timer interrupt ID
HALT:	BNE HALT	// Unexpected ID → halt program
	BLEQ TIMER_ISR	// If equal, branch to TIMER_ISR