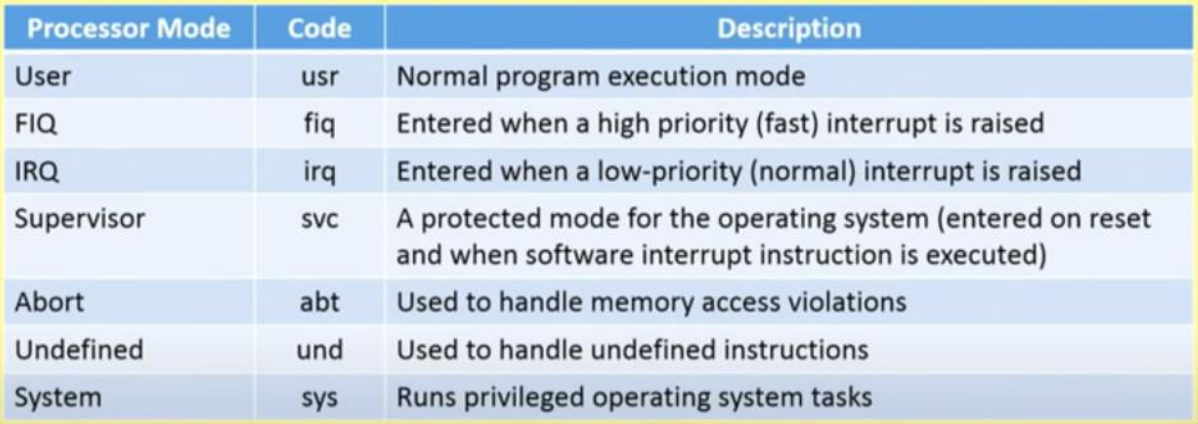
In this section

- ARM processor modes and registers
- Special registers and exception handling
- ARM and Thumb modes of execution

Processor Modes



Processor Mode	Code	Description
User	usr	Normal program execution mode
FIQ	fiq	Entered when a high priority (fast) interrupt is raised
IRQ	irq	Entered when a low-priority (normal) interrupt is raised
Supervisor	svc	A protected mode for the operating system (entered on reset and when software interrupt instruction is executed)
Abort	abt	Used to handle memory access violations
Undefined	und	Used to handle undefined instructions
System	sys	Runs privileged operating system tasks

Registers

- ARM has 37 registers all of which
- These registers are:
  - ↳ 1 dedicated program counter (PC)
  - ↳ 1 dedicated current program status registers (CPSR)
  - ↳ 5 dedicated saved program status registers (SPSR)
  - ↳ 30 general-purpose registers (GPR)

- The current processor mode governs which of several register sets is accessible

- Only 16 registers are visible to a specific mode of operation

- Each mode can access

→ a particular set of registers r0 - r12

→ r13 (SP, stack pointer)

→ r14 (LR, link register)

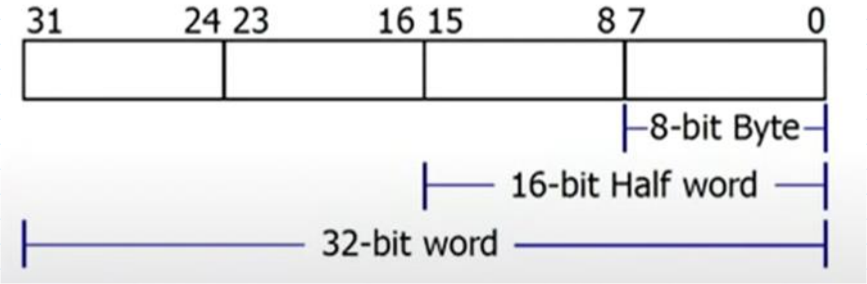
→ r15 (PC, program counter)

→ Current program status register (CPSR)

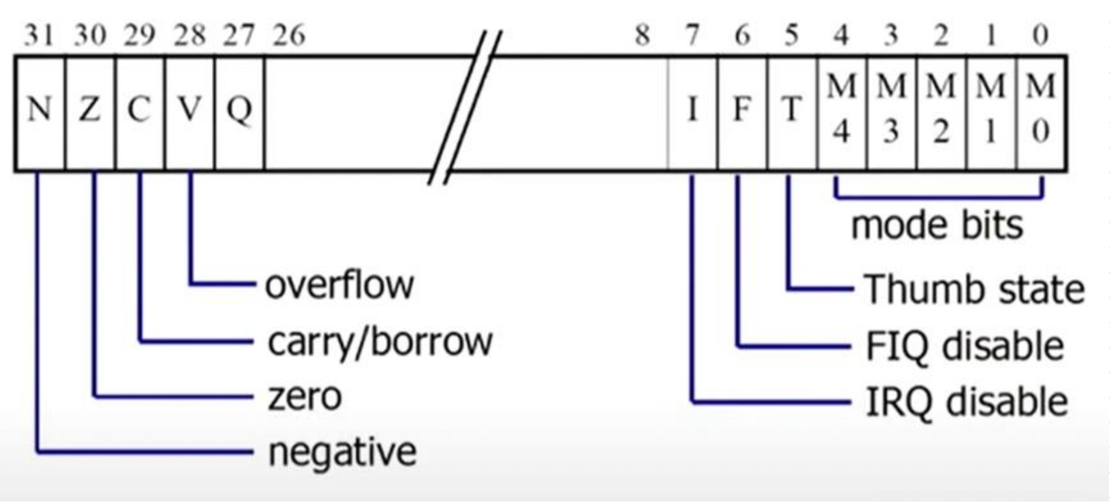
- Privileged modes (except system) can also access a particular SPSR

General Purpose Registers

- 6 data types are supported (signed/unsigned)
  - ↳ 8-bit byte, 16-bit half word, 32-bit word
- All ARM operations are 32-bit
  - ↳ shorter data types are only supported by data transfer



Current Program Status Register



Special Registers

- PC (r15)
- LR (r14)
- SP (r13)
- CPSR
- SPSR

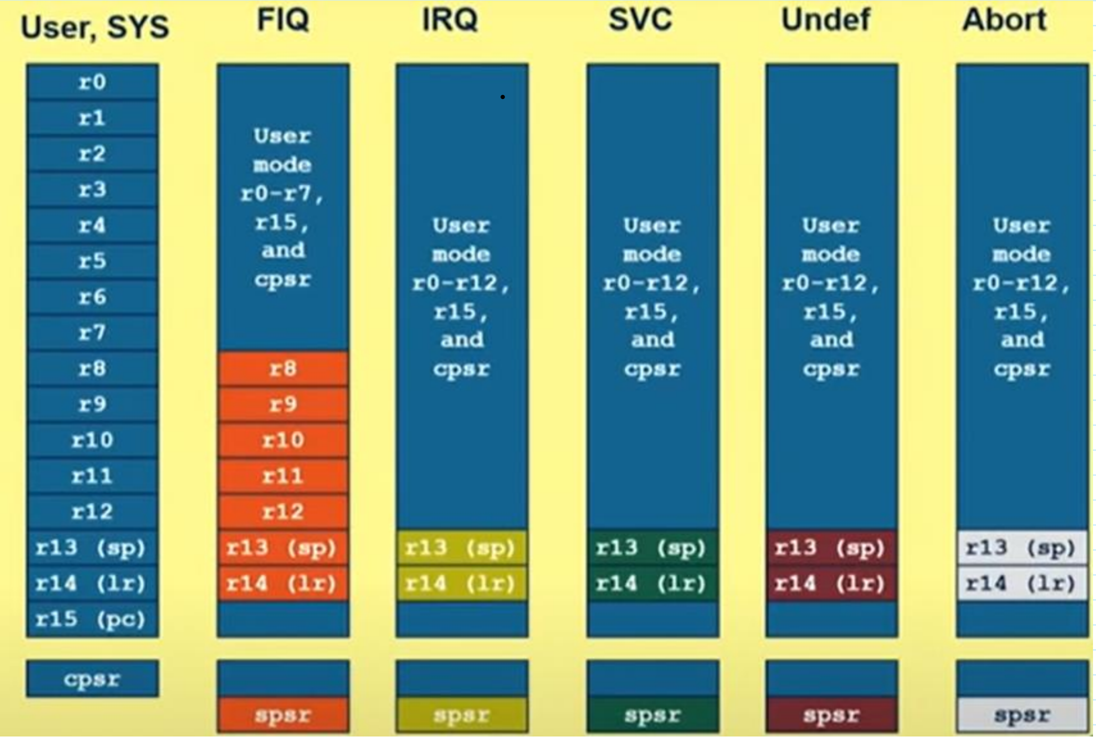
Program Counter

- When the processor is executing in ARM mode
  - ↳ All instructions are 32-bits wide and must be word aligned
  - ↳ The last two bits of PC are zero (ie not used)
  - ↳ Due to pipelining, PC point 8 bytes ahead of the current

instruction or 12 bytes ahead if the current instruction includes a register specific shift

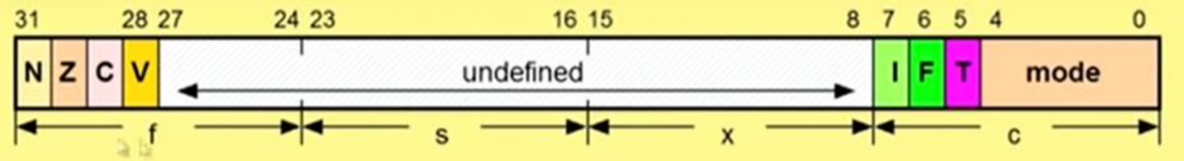
- When the processor is executing in Thumb mode
  - all instructions are 16-bits wide and are half word aligned
  - the last word of the PC is zero (ie not used)

Register Organization Summary



User, SYS	FIQ	IRQ	SVC	Undef	Abort
r0	User mode r0-r7, r15, and cpsr	User mode r0-r12, r15, and cpsr	User mode r0-r12, r15, and cpsr	User mode r0-r12, r15, and cpsr	User mode r0-r12, r15, and cpsr
r1					
r2					
r3					
r4					
r5					
r6					
r7					
r8	r8				
r9	r9				
r10	r10				
r11	r11				
r12	r12				
r13 (sp)	r13 (sp)	r13 (sp)	r13 (sp)	r13 (sp)	r13 (sp)
r14 (lr)	r14 (lr)	r14 (lr)	r14 (lr)	r14 (lr)	r14 (lr)
r15 (pc)					
cpsr					
	spsr	spsr	spsr	spsr	spsr

Program Status Register



- C ⇒ ALU operation Carried out
- V → ALU operation oVerflowed

T-bit

T = 0 ⇒ ARM state

T = 1 ⇒ Thumb State

Mode bits :

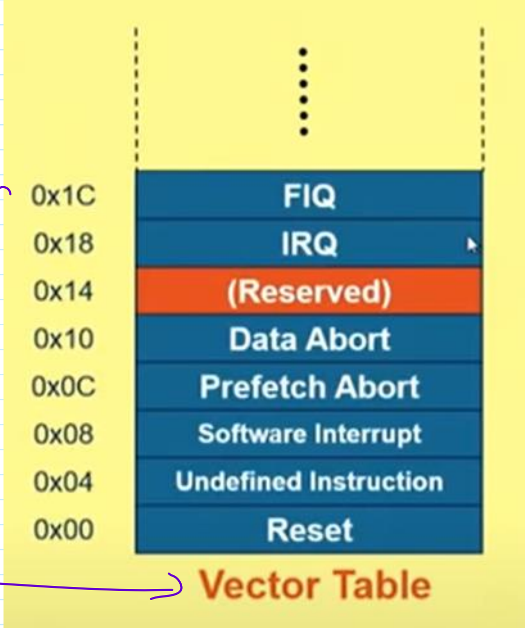
10000	User	10010	IRQ	10111	Abort	11111	System
10001	FIQ	10011	Supervisor	11011	Undefined		

Exception Handling

When an exception occurs, the processor :-

- Copies CPSR into SPSR_<mode>
- Set the appropriate bits in CPSR
  - ↳ changes to ARM state
  - ↳ Change to related mode
  - ↳ Disable IRQ, FIQ.
- Store return address to LR_<mode>
  - ↳ PC gets vector table address
- To return, the exception handler needs to :-

FINISH This!!



0x1C	FIQ
0x18	IRQ
0x14	(Reserved)
0x10	Data Abort
0x0C	Prefetch Abort
0x08	Software Interrupt
0x04	Undefined Instruction
0x00	Reset

Vector Table

4-bytes each (32-bit address)

These are addresses