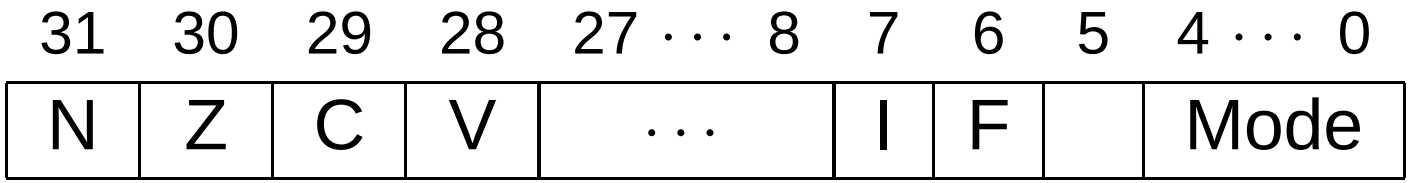
Interrupts and Exceptions

Interrupts

- Asynchronous event
- External event

- Nonmaskable Interrupts (NMI)
  (FIQ: Fast Interrupt)
- Prioritised Interrupts
- Vectored Interrupts

Status Register



I Disable Interrupt Requests

F Disable Fast Interrupt Requests

Mode Processor Mode

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

Interrupts or Exceptions

- Interrupt
  Externally Generated
  External device requesting attention

- Exception
  Internally generated interrupt

  $\Rightarrow$ CPU based exceptions:
  - ◇ Prefetch Abort
  - ◇ Data Abort

  $\Rightarrow$ Programmer Initiated:
  - ◇ Undefined Instructions
  - ◇ Software Interrupt (SWI)

Processing

- On accepting an interrupt CPU will:

  1 Complete current instruction
  2 Save *return* address in LR
  3 Save CPSR in SPSR
  (Saved Process Status Register)
  4 Switch to interrupt mode
  5 Set PC to address of handler

- On return from interrupt CPU will:

  1 Recover CPSR from SPSR
  2 Recover PC from Link Register
  (Switch to original CPU mode)
  3 Continue interrupted program

Processing

- On accepting an interrupt CPU will:

  1 Complete current instruction
  2 Save *return* address in LR
  3 Save CPSR in SPSR
  (Saved Process Status Register)
  4 Switch to interrupt mode
  5 Set PC to address of handler

  LR_*mode* ← PC
  SPSR_*mode* ← CPSR

  CPSR(Mode) ← *mode*
  MAR ← *handler*
  MBR ← M(MAR)
  PC ← MBR

- On return from interrupt CPU will:

  1 Recover CPSR from SPSR
  2 Recover PC from Link Register
  (Switch to original CPU mode)
  3 Continue interrupted program

Processing

- On accepting an interrupt CPU will:

1	Complete current instruction	
2	Save *return* address in LR	LR_*mode* ← PC
3	Save CPSR in SPSR (Saved Process Status Register)	SPSR_*mode* ← CPSR
4	Switch to interrupt mode	CPSR(Mode) ← *mode*
5	Set PC to address of handler	MAR ← *handler*
		MBR ← M(MAR)
		PC ← MBR

- On return from interrupt CPU will:

1	Recover CPSR from SPSR	CPSR ← SPSR_*mode*
2	Recover PC from Link Register (Switch to original CPU mode)	PC ← LR_*mode*
3	Continue interrupted program	

Exception Processing Modes

Handler	Mode	Function	Priority
0x00	Supervisor	System Reset	1
0x04	Undefined	Undefined Instruction	6
0x08	Supervisor	Software Interrupt (SWI)	6
0x0C	Abort	Prefetch Abort	5
0x10	Abort	Data Abort	2
0x14	—	Reserved	—
0x18	IRQ	Interrupt Request	4
0x1C	FIQ	Fast Interrupt Request	3

Interrupt: System Reset

Called when Reset pin activated, normally at Power-Up or system reset.

Reset signal also used to reset hardware.

This is where you would place the start up or initialisation code.

This is different to other exceptions in that you are not expected to return from an reset exception, so the LR and SPSR register are not initialised.

Exceptions: Undefined / SWI

- Undefined Instructions

  Occurs when an attempt is made to execute an instruction that has not been defined.

  This can be used by a system to provide it's own special purpose instructions.

- Software Interrupt

  The Software Interrupt instruction (`SWI`) is used to request a service of the operating-system.

  Two methods of indicating which service is required:

  ⇒ In the `SWI` instruction (i.e., `SWI &11`)

  ⇒ In a designated register (i.e., `R0`)

Exceptions: Abort

- Prefetch Abort

  If the processor tries to executed at an invalid address — all instructions must be word aligned.

  The Breakpoint (BKPT) instruction will also force a Prefetch Abort.

- Data Abort

  If an attempt is made to access a word or halfword on a non-aligned address.

  $\Rightarrow$ Words must be word aligned (4 byte)

  $\Rightarrow$ Halfwords must be halfword aligned (2 byte)

Interrupt Request

Called when the Interrupt Request (IRQ) pin is activated.

An external device has requested attention. The handler should pole each device in an attempt to discover which device generated the interrupt.

Disables further IRQs from occurring until the current request has been handled.

As the interrupt request (IRQ) has a lower priority than the fast interrupt request (FIQ), interrupt request's are disabled during a FIQ request.

Fast Interrupt Request

Called when the Fast Interrupt Request (FIQ) pin is activated.

An external device has requested a Fast Interrupt.

This is designed to support a high-speed data transfer process. It has sufficient private registers to remove the need for register saving, thus minimizing the memory access overhead.

A fast interrupt request (FIQ) will disable further FIQs and IRQs until the current request has been completed.