chap-05

Exceptions and Interrupt handling:-

* Exception handlers are heart of an embedded system which are responsible for handling errors, interrupts & other events generated by external system.
* Efficient handlers can dramatically improve system performance
* ARM processor has 7 exceptions that can halt the normal sequential execution of instrs:

1. Data Abort
2. Fast interrupt request
3. Interrupt request
4. Prefetch abort
5. Software interrupt
6. Reset
7. Undefined instr.

Exception handling:-

* Exception is any condition which needs to halt the normal sequential execution of instrs.

Ex:- - ARM core is reset,
- when an instr fetch or mem. access fails,
- when undefined instr is encountered,
- when external interrupt has been raised,

* Exception handling is a method of processing these exceptions; which have an associated software exception handler
  is a s/w routine which is executed when an exception occurs. Ex:- Data abort exception will have data abort handlers..

* The handler first determines the cause of exception & then services the exception.
  - within the handler
  - or
  - branches to a specific subroutine
* The following are the exception handling
  1. ARM processor mode & exceptions
  2. Vector table
  3. Exception priorities
  4. Link register offset.

(1) ARM processor mode & exceptions:

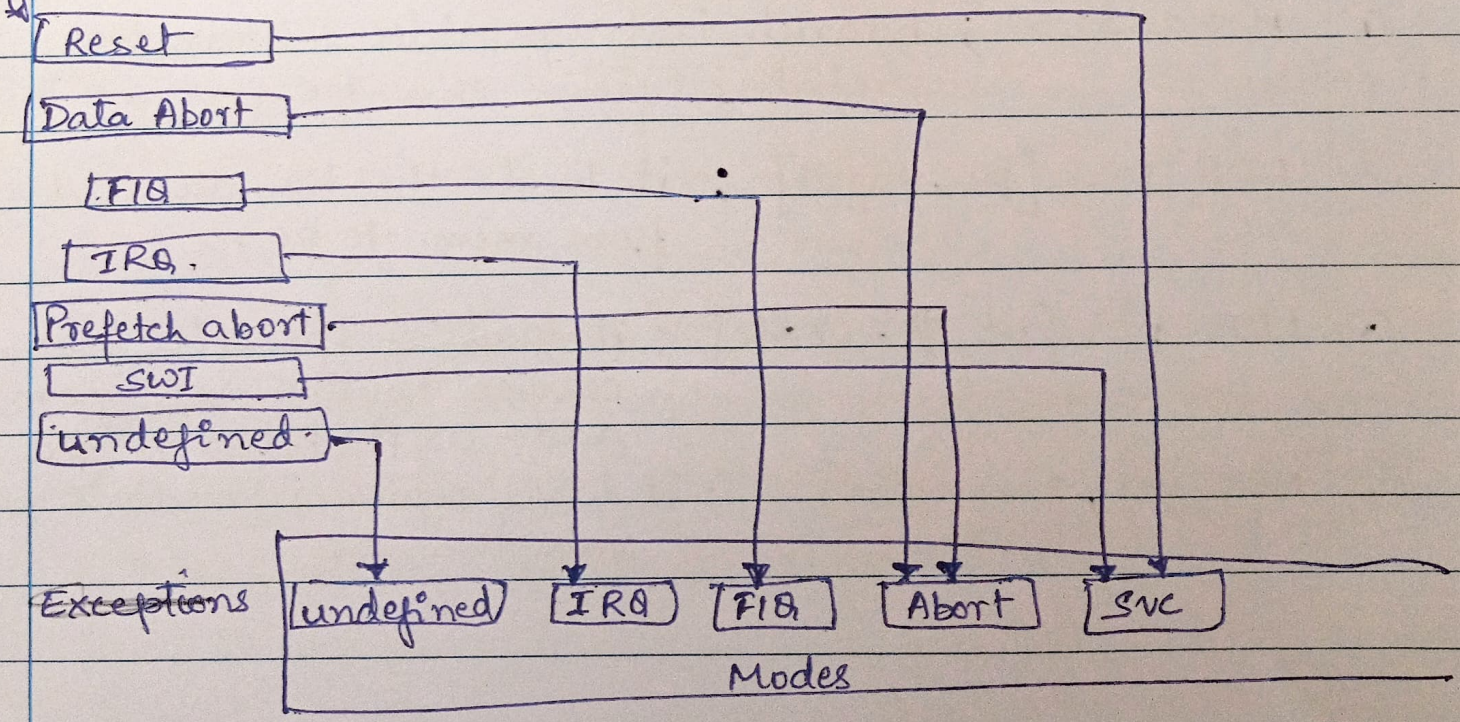
* Each exception causes the core to enter a specific mode or any of the ARM processor modes can be entered manually by changing the cpsr.
* User and system mode are the only two modes which are not entered by a corresponding exception
* when an exception causes a mode change, the core automatically
  1. saves the cpsr to spsr of the exception mode
  2. saves pc to Lr of the exception mode
  3. sets the cpsr to the exception mode
  4. sets pc to the addr of exception handler.

NOTE:- when exception occurs ARM processor switches to ARM state.

ARM processor exceptions & associated modes:

	Exception	Mode	Main purpose
①	Fast interrupt request	FIQ	Fast interrupt request handling.
②	Interrupt request	IRQ	Interrupt request handling.
③	SWI & Reset	SVC	protected mode for operating systems
④	Prefetch abort & Data abort.	Abort	Virtual mem. and/or mem. protection handling
⑤	Undefined instr.	undefined	software emulation of hardware coprocessors.

Exceptions & associated modes.



② **Vector table**: a table of addresses that the ARM core branches to when an exception is raised.

Vector table & processor modes.

	Exception	Mode	vector table offset
①	Reset	SVC	+0x00
②	undefined instr	UND	+0x04
③	Software Interrupt (SWI)	SVC	+0x08
④	Prefetch abort	ABT	+0x0C
⑤	Data Abort	ABT	+0x10
⑥	Not assigned	—	+0x14
⑦	IRQ	IRQ	+0x18
⑧	FIQ	FIQ	+0x1C

* The following branch instrs are used

① B <addr> ; branch instr provides a branch relative from the PC

② LDR pc, [pc, #offset] ; it loads the handler addr from mem. to pc.

③ LDR pc, [pc, #-0xff0] ; it loads a specific interrupt service routine addr from addr 0xffffff030 to PC.

④ MOV pc, #immediate ; it loads/copies immediate value into the PC.

③ Exception handlers :-

* Exceptions can occur simultaneously, so the processor has to adopt a priority mechanism.

Exceptions priority levels

	Exceptions	Priority	I-bit	F-bit
①	Reset	1 (Highest)	1	1
②	Data Abort	2	1	–
③	Fast interrupt request	3	1	1
④	Interrupt request	4	1	–
⑤	Prefetch Abort	5	1	–
⑥	Software Interrupt	6	1	–
⑦	Undefined Instr	6 (Lowest)	1	–

① Reset exception : (Highest priority) is always taken whenever it is signaled.

- the reset handler initializes the system, including setting up mem. & caches ; also set up stack pointers for all processor modes.
- During the first few instrs of the handler, it is assumed that no exceptions or interrupts will occur.
- the code is designed to avoid SWI's, undefined instrs & mem. access which may abort.

② Data Abort: occur when mem. controller or MMU indicates that an invalid mem. address has been accessed or current code attempts to read or write to mem. without the correct access permissions.

Data abort handler
↓ within (inside
FIQ exception
is raised since FIQ cannot be disabled.
end.

③ Fast Interrupt Request (FIQ): occurs when an external peripheral sets the FIQ pin to nFIQ.
- the core disables both IRQ & FIQ exceptions on entry into FIQ handler.
- Thus no external source can interrupt the processor unless IRQ & FIQ exceptions are reenabled by software.

④ Interrupt Request (IRQ): occurs when an external peripheral sets the IRQ pin to nIRQ.
- IRQ handler will be entered if neither an FIQ exception nor Data Abort exception occurs.
- On entry to IRQ handler, the IRQ exceptions are disabled & should remain disabled until the current interrupt source has been cleared.

Nested Interrupt handler :-

(5) Prefetch Abort: occurs when an attempt to fetch an instr results in mem. fault.
- It is raised when the instr is in the execute stage of pipeline & if none of the higher exceptions are raised.
- On entry to the handler, IRQ exceptions will be disabled, but FIQ exceptions will remain unchanged
- If FIQ is enabled & if FIQ exception occurs, then it will be taken first & then Prefetch abort exception is serviced.

(6) Software Interrupt (SWI): occurs when SWI instr is executed & none of the other higher priority exceptions have been flagged.
- On entry to handler, cpsr will be set to supervisor mode.
- If the system uses nested SWI calls, the link register R14 & spsr must be stored away before branching to nested SWI to avoid possible corruption of link register & spsr.

(7) Undefined instr : occurs when an instr not in the ARM or THUMB instr set reaches the execute stage of the pipeline & none of the other exceptions have been flagged.

- The ARM processor "asks" the coprocessors if they can handle this coprocessor instr.
  since coprocessors follow the pipeline, instr identification can take place in execute stage of the core.
- If none of the coprocessors claims the instr, Undefined instr exception is raised.

(4) Link Register offsets :

* when an exception occurs, the link register is set to a specific address based on the current pc.
Ex: when IRQ exception is raised, the link reg. Lr points to last executed instr + 8.
* Care has to be taken to make sure the exception handler does not corrupt Lr because Lr is used to return from exception handler.

Useful link-register-based addresses :

	Exception	Addr	Use.
(1)	Reset	—	Lr is not defined on Reset
(2)	Data Abort	Lr-8	points to instr that caused Data abort exception
(3)	FIQ	Lr-4	return addr from FIQ handler
(4)	IRQ	Lr-4	return addr from IRQ handler
(5)	Prefetch abort	Lr-4	points to instr that caused Prefetch abort exception
(6)	SWI	Lr	points to next instr after SWI instr
(7)	undefined Instr	Lr	points to next instr after undefined instr

Ex: Different methods of returning from IRQ or FIQ exception handler

Ex① using SUBS

```
handler
        < handler code >
        ---
        SUBS  pc, R14, #4     ; pc = R14 - 4
```

Because there is an S at the end of SUB instr & pc is the destination reg., the cpsr is automatically restored from spsr reg.

Ex② subtracts the offset from link reg. R14 at the beginning of the handler

```
handler
        SUB  R14, R14, #4.        ; R14 = R14 - 4
        ---
        < handler code >
        ---
        MOVS  pc, R14             ; return
```

After servicing is complete, return to normal execution occurs by moving link reg. R14 into pc & restoring cpsr from spsr.

Ex:③ use of interrupt stack to store the link reg.
- subtracts an offset from link reg. & then stores it onto interrupt stack.

```
handler
        SUB R14, R14, #4          ; R14=R14-4
        STMFD R13!, {R0-R3, R14}  ; store context
        ---
        <handler code>
        ---
        LDMFD R13!, {R0-R3, PC}^  ; return
```

^ forces cpsr to be restored from spsr.

Interrupts : 2 types on ARM processor
① causes exception raised by external peripheral IRQ & FIQ.
② causes an exception — SWI instr

Both types suspend the normal flow of a program

Assigning interrupts :-
* A system designer can decide which hardware peripheral can produce which interrupt request.
- This decision can be implemented in h/w or s/w (or both) & depends upon the embedded system being used.