

Microcontrollers and Embedded Systems.

18CS44.

Module - I

Microcontroller:- A microcontroller abbreviated (MCU or uC) is a computer system on a chip (soc) that does a job. It contains an integrated processor, memory and programmable I/O peripherals.

→ Difference between Microcontroller and Microprocessor.

Microcontroller

* Heart of Embedded System

* Has external places - or along with memory, I/O peripheral

* Can be used in compact systems

* Cost of Entire system is low.

Microprocessor

* Heart of Computer System

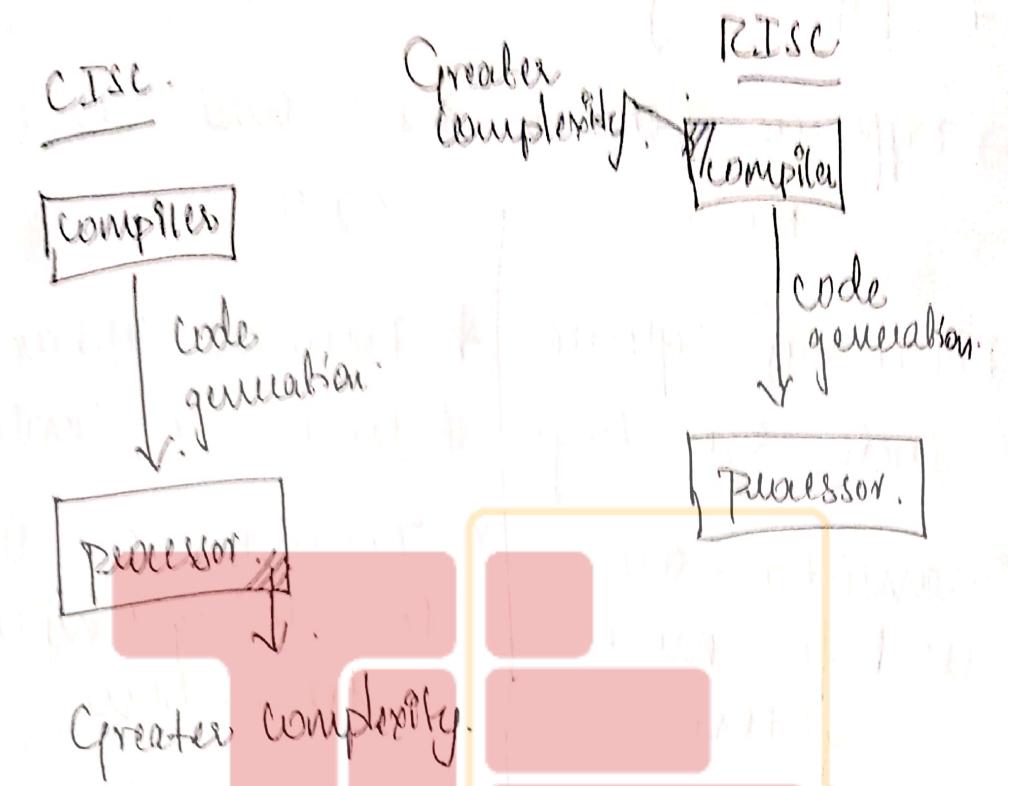
* It is just a processor. Other components are connected separately

* Cannot be used in compact system

* High cost

Microprocessor uses - personal computer
Microcontroller used can be used in MP3
players

→ RISC Design philosophy.



RISC:- Reduced Instruction Set Computer, is a type of Microprocessor Architecture that utilizes a small highly optimized set of Instructions.

Characteristics of RISC.

- Simple Instruction, hence simple instruction decoding
- Instruction take single clock cycle to be executed.

- * More number of general purpose registers
- * Simple Addressing modes
- * Less data types
- * Pipelining

→ Difference between RISC and CISC.

RISC	CISC
* Focus on software	* Focus on software
* Code size - Large	* Code size - Small
* Transistors are used for more registers	* Transistors are used for storing complex instructions
* An Instruction can fit in one word.	* Instruction is larger than size of one word.

→ ARM Design Philosophy.

There are number of physical features that have driven the ARM processor design.

1. Small to reduced power consumption and extend battery operation

2. High code density.
3. price sensitive and use slow and low cost memory devices
4. Reduce the area of die taken by the embedded processor.
5. Hardware debug technology
6. ARM core is not a pure RISC architecture.

→ Instruction set for Embedded Systems

The ARM instruction set differs from the pure RISC definition in several ways that make ARM instruction set suitable for Embedded Application.

* Variable cycle execution for certain instruction

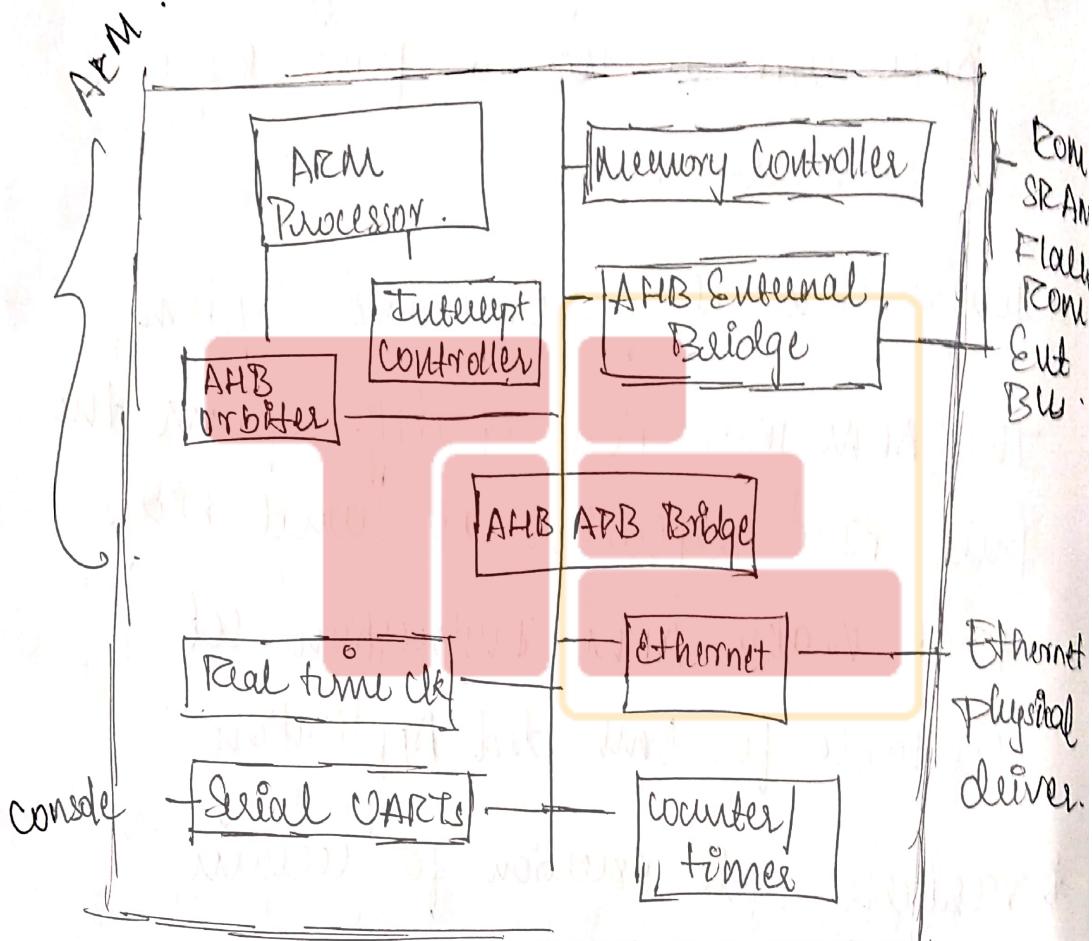
* Inline branch Shifter leading to more complex instruction

* Thumb 16 bit Instruction set: Can execute 16 or 32 bit Instruction

* Conditional Execution.

* Enhanced Instruction :- The enhanced Digital Signal processor (DSP) Instructions were added to support fast 16×16 Multiplier operations and Saturations.

→ Embedded System Hardware



Four main hardware components

* The ARM Processor :- It controls the embedded device. An ARM processor comprises a core plus the surrounding components that interface it with bus. These components include cache and

Memory management.

- * **Controllers**:- They coordinate important functional blocks of the system. Two commonly found controllers are interrupt controller and memory controller.
- * **Peripherals**:- They provide Input or Output capabilities.
- * **Bus**:- Communication b/w different parts

→ ARM Bus Technology.

Embedded systems use different bus technology but the most common PC bus technology is the Peripheral Component Interconnect.

Embedded systems use one single bus to connect internal parts and peripherals.

→ Two different classes of devices attached to bus.

↓
Bus Master (Arm processor core)

↓
Bus Slave (Peripherals)

1 Initiates data transfer with another device in same bus

2 Responds to master's request

→ Architecture levels:

Physical

- covers electrical characteristics
- covers Bus width.

Second level (Protocol)

- Rules that govern the communication b/w peripheral & processor.

→ AMBA Bus Protocol.

Advanced Microcontroller Bus Architecture was introduced in 1996 and has been widely adopted as the on-chip bus.

- The first AMBA buses introduced were ARM system bus (ASB) and ARM peripheral bus (APB).
- Later ARM introduced another bus design, called ARM High performance bus.

→ Characteristics of AHB

- * High data throughput than APB
- * High performance peripherals compared to APB's slow peripheral
- * External Bus acquires a specialized bus bridge to connect with AHB bus.

Memory

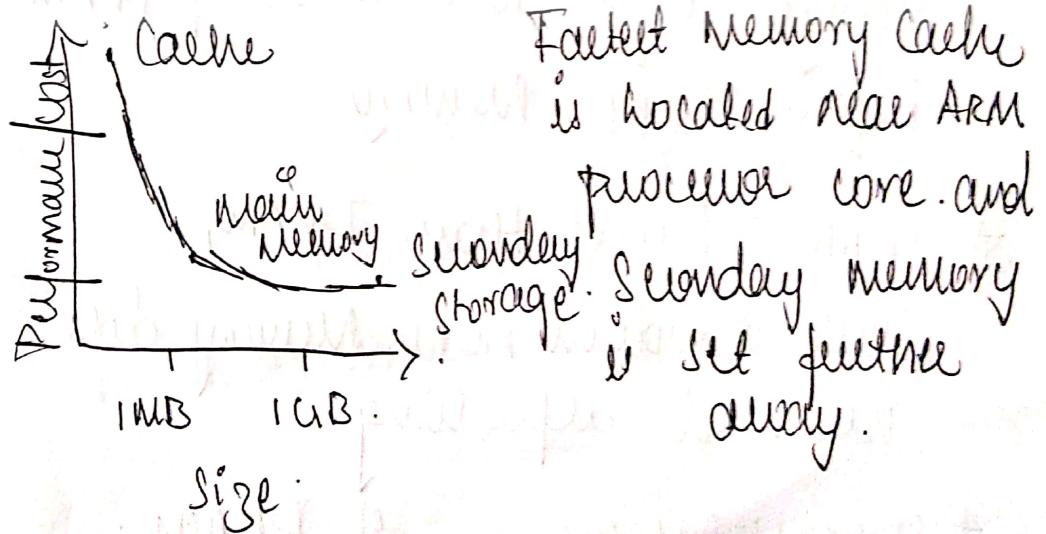
Embedded System has to have some form of Memory to store and execute code. It has some specific memory characteristics such as Hierarchy, width and type-

(i) Hierarchy

- * All computer system have memory arranged in some form of hierarchy.

Cache

Fastest Memory Cache is located near ARM processor core and



- * Closer Memory - High cost, Smaller

Capacity

- * Cache:- placed b/wn ARM core, & Main Memory speeds up transfer b/wn Processor and Main memory.
- * Main Memory:- Range [256kb to 8GB]
- * Secondary Storage:- Largest and slowest form of Memory. [600mb to 10GB]

(ii) Types

- * ROM:- Stores long term data that is to preserved even after power is off.
- * Flash ROM:- Read-write, but is slow to write. Hence not used for holding Dynamic data.
- * DRAM:- Low cost, need to refresh storage cells. Need to setup DRAM before using memory.
- * SRAM:- Faster than DRAM, Static Random Access Memory does not need refreshing.
- * SDRAM:- Sub category of DRAM.

(ii) Width of memory width is number of bits the memory stores on each access typically 8, 16, 32 or 64 bits.

(iv) Peripherals:

- * Outside World Interfacing of Embedded system
- * performs I/P and O/P by connecting to other device or sensors that are off chip.

- * Specialized peripherals are called controllers implement higher level of functionality

Memory
controllers

Connects different
types of memory
to processor bus

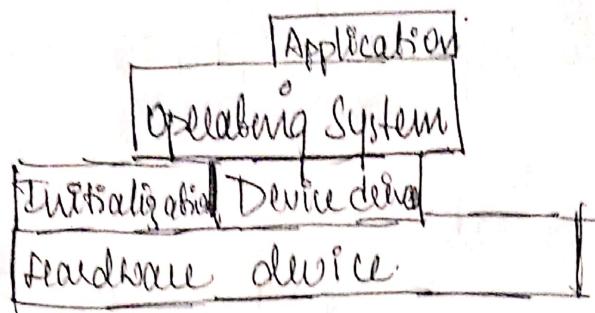
Interrupt
controllers

allow peripheral
requires attention if
issues an interrupt
to processor

Standard
Interrupt
controller

vector
interrupt
controller

Embedded System Software



An Embedded System Obviously needs a Software to drive every Application.

Initialization (Boot) code:

Select state to a user state where operating system can run.

It configures

→ Memory controller

→ Processor caches

→ Initializes some devices

We can group into three phases

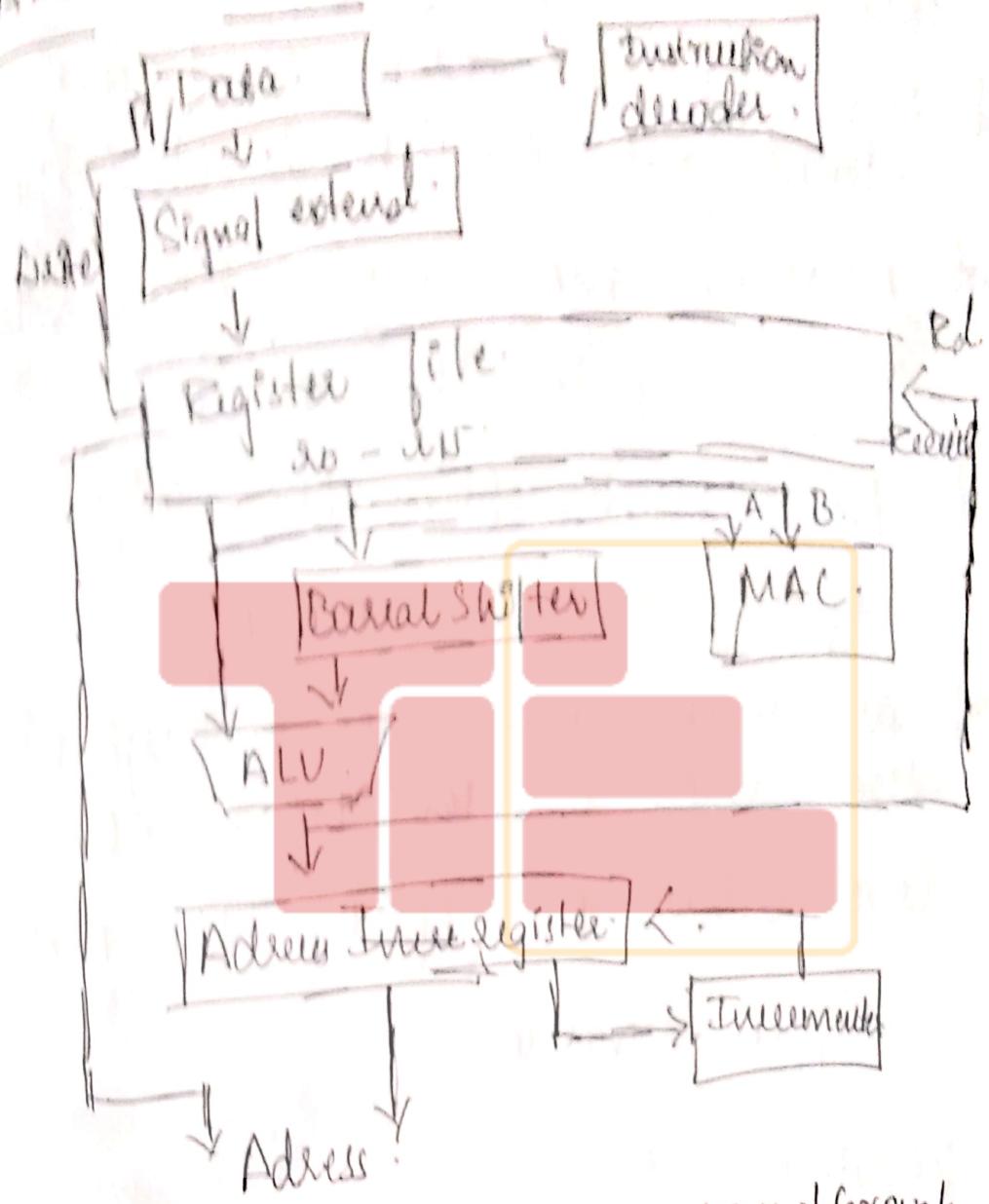
→ Initial hardware configuration:

sets up target platform to boot an image.

→ Diagnostic code is fault Identification and Isolation.

→ pointing to where loading an image and handing control over to that image.

ARM core



* Data enters the processor core through data bus

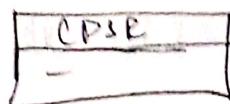
* Instruction decoder translates instruction before they are executed.

* Signal Extend covers converts signed 8 bit and 16 bit numbers to 32 bit

- values, as they are read from memory.
- * Register file consists of 16 registers in which data is available for the user to access and store.
 - * Left shift and right shift are carried out through barrel shifter.
 - * ALU or MAC takes register values and computes the result.
 - * Address register stores the address of any particular address.
 - * Incrementer increases the address register before the core reads or writes the next register value.

Registers

r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13 SP

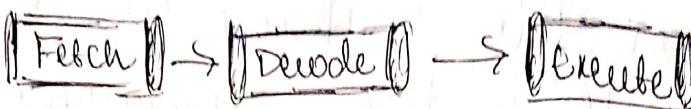


- * General purpose registers either hold data or address
- * Identified by 'r' as prefix
- * Register file contains all registers available to the programmer.
- * Available registers depends upon the mode of the processor.
- * All registers are 32 bit in size.
- * Data registers visible to programmer are r0 to r15
- * Sp → Stack pointer, stores starting address of stack in current processor mode.
- * lr → Link register, saves the core plus the return address whenever it calls a subroutine.
- * PC → Program counter holds the address of next instruction.
- * Program status register → CPSR
SPSR.

Pipeline

This is a mechanism RISC uses to handle instructions.

Next instruction is fetched while other instructions are being loaded and decoded.

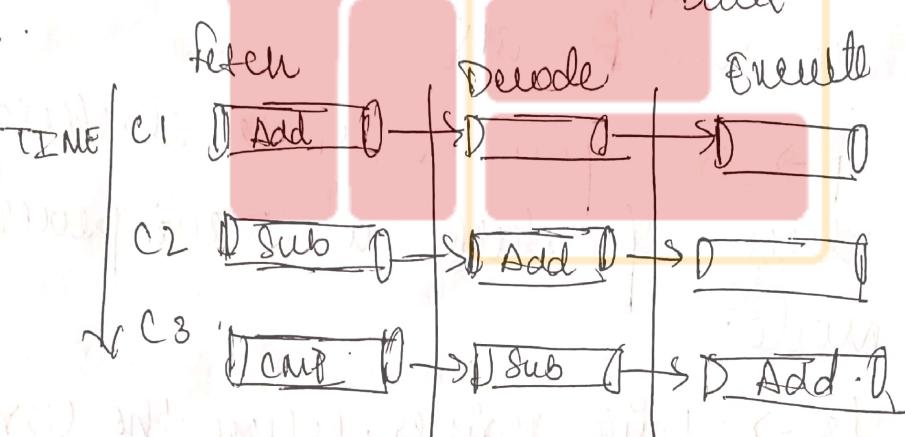


↓
Loads all
Instructions
from Memory.

↓
Identifies the
Instruction to
be executed

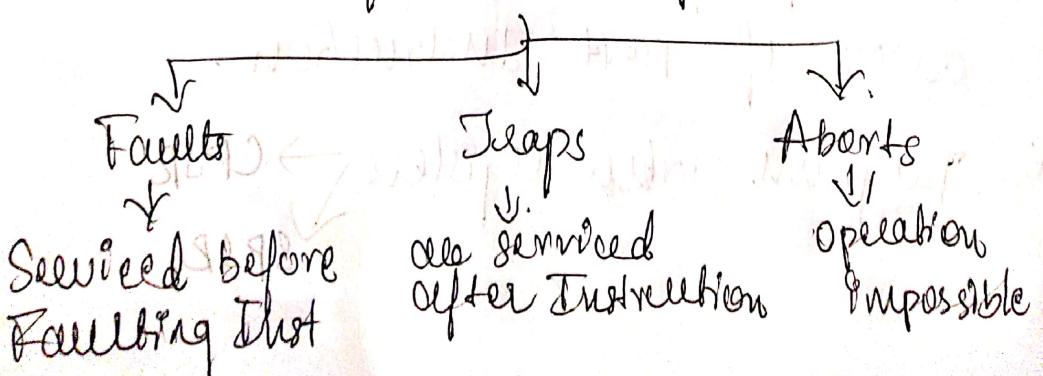
↓
Processes the
Instruction
and writes
the result
back

Ex:



→ Interrupts, Exceptions and Vector Table

Exception:- Exceptions are used to handle Instruction faults [Div by zero]



Interrupts:- Interrupts are used to handle external events. [Keyboard, Serial port]

S/W

H/W

Called by
the software
of the system.

H/W

S/W

Called by
Hardware
of the system.

Vector Table:- It contains the reset value of the start pointer, and the start addresses are also called as Exception Vector.

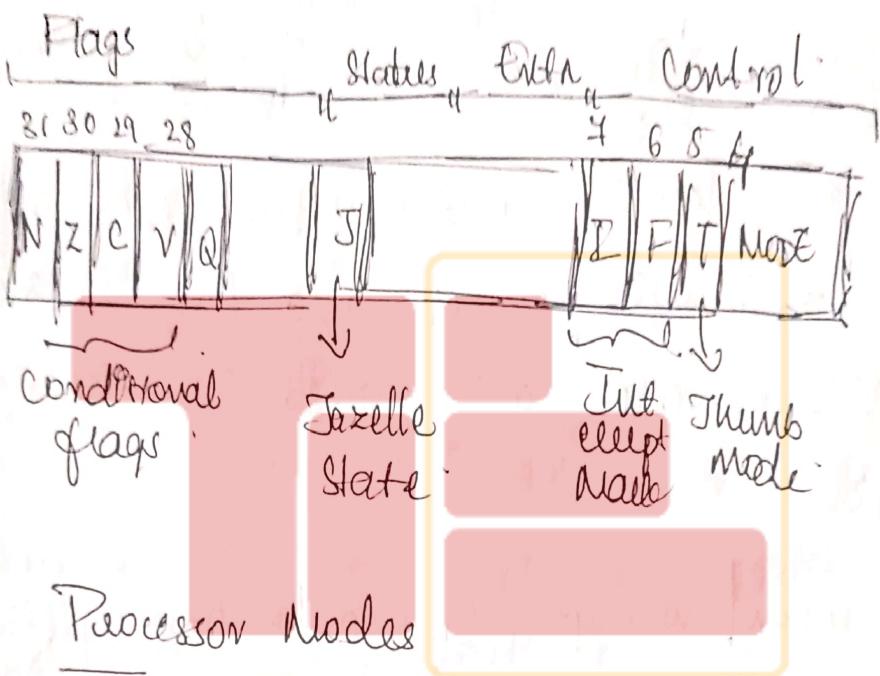
Exception/Interrupt	Short hand	Address	High Address
Reset	RESET	0x00000000	0xffffffff0000
Undefined Interrupt	UNDEF	0x00000004	0xffffffff0004
Software Interrupt	SWI	0x00000008	0xffffffff0008
Prefetch Interrupt	PABT	0x0000000C	0xffffffff000C
Data abort	DABT	0x00000010	0xffffffff0010
Reserved.	-	0x00000014	0xffffffff0014
Trap - required	TRQ	0x00000018	0xffffffff0018

Current Program Status Register

ARM uses CPSR to monitor and control internal operations. CPSR is a dedicated 32 bit register.

It is divided into:

- Flags
- Control
- Status
- Extension



Processor Modes

Determines which registers are affine
Privileged: full read-write
Non Privileged: only read, read-write: conditional flags.

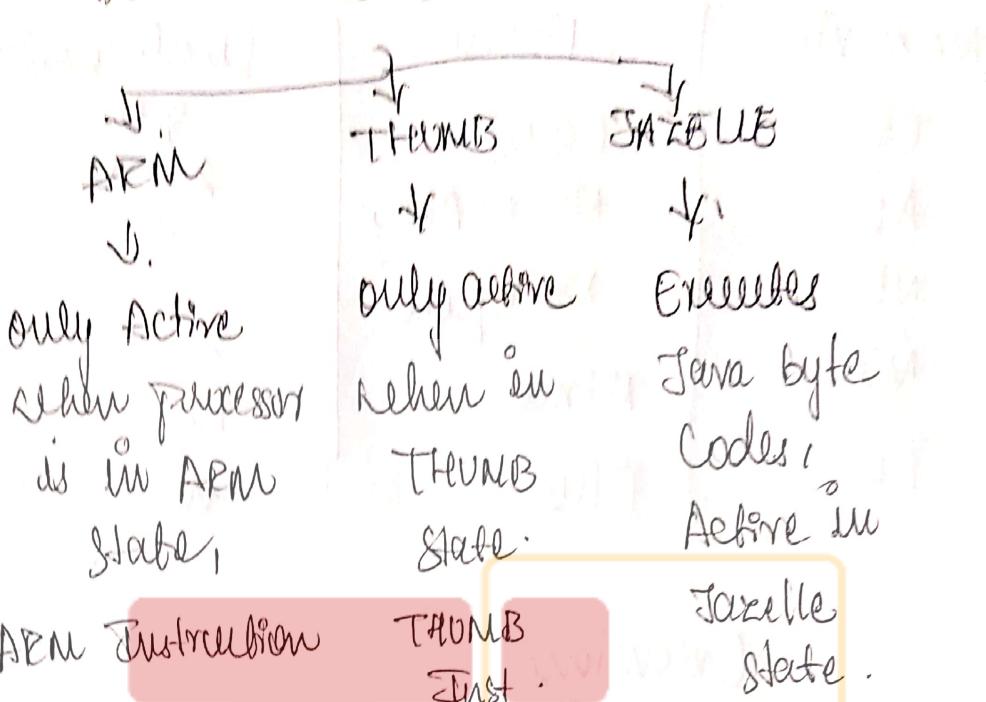
Privileged Modes

- Abort
- Fault interrupt request
- Interrupt request
- Supervisor
- System
- Undefined

User mode, for programs & Applications.

Processor States and Instruction sets

State of core determines which instruction set is executed.



Interrupt Mail :
Used to Stop a Specific Instruction

conditional flags -

updated by comparisons / results of ALU.

Flag	Flag Name	Set when
Q	Saturation	Result carries overflow
V	Overflow	Carries a signed overflow
C	Carry	Carries a unsigned carry
Z	Zero	Result is zero.
N	Negative	Result is negative.

Conditional Branch

Controls whether or not the core will execute an instruction.

Mnemonic	Name	Conditional flags
EQ	Equal	Z
AL	Not Always	-
NE	Not equal	-
MI	Minus	Z
PL	Plus	N n

Extensions

- * Standard Components Next to ALU core
- * Improve Performance / Memory Management, Resource Management etc.

Type

- Cache & Lightly coupled
- Memory Management
- Coprocessor, Interface

Coprocessor → Extends the features of core.