

Government of Karnataka
Department of Collegiate and Technical Education
Bapuji Polytechnic Shabanur ,Davangere(335)

Study Material/Notes

ON

ARM CONTROLLER-15EC52T

Unit-1: ARM Embedded Systems & ARM Processor Fundamentals

For 5th Semester Diploma in Electronics & Communication Engineering

Prepared By;

Sujatha N
SGL,ECE Dept
Bapuji Polytechnic Shabanur
Davanrege

For the Academic Year 2020-21

Unit-1: ARM EMBEDDED SYSTEMS AND ARM PROCESSOR FUNDAMENTALS

Topics:

- RISC design philosophy
- ARM design philosophy
- Embedded system hardware
- AMBA bus protocol
- Embedded system software
- Applications
- ARM core data flow model
- Registers
- CPSR
- Processor modes
- Banked registers
- Pipeline characteristics

Introduction:

- ARM stands for Advanced RISC (Reduced Instruction Set Computer) Machine.
- It is a 32-bit processor used for higher end applications.
- It has a control over both ALU (Arithmetic and Logic Unit) and shifter in data processing instructions.
- It is an engine within a system that fetches ARM instructions from memory and executes them.
- It is designed to be small to reduce power consumption.
- It has low code size, less price, low power consumption, less silicon area (size) and high performance.
- It has hardware debug technology.
- These are used for many applications in mobile, imaging, automotive, robotics, video games & consumer devices.

RISC Design Philosophy:

- 1) **Instructions:** Each instruction is executed in a single cycle.
- 2) **Pipelines:** To speed up execution instructions are broken into smaller units and executed in parallel.
- 3) **Registers:** Registers can store data or address and function as faster access.
- 4) **Load –store architecture:** Separate instructions to transfer data between the register and external memory.

ARM Design Philosophy:

The physical features that are considered for ARM processor design are.

- 1) **Battery Power:** It is designed small to reduce power consumption and extended battery operation for applications like mobile phones and PDAs.
- 2) **High Code Density:** Embedded systems have limited memory due to cost and physical size

High code density is useful for applications such as mobile phones and mass storage devices.

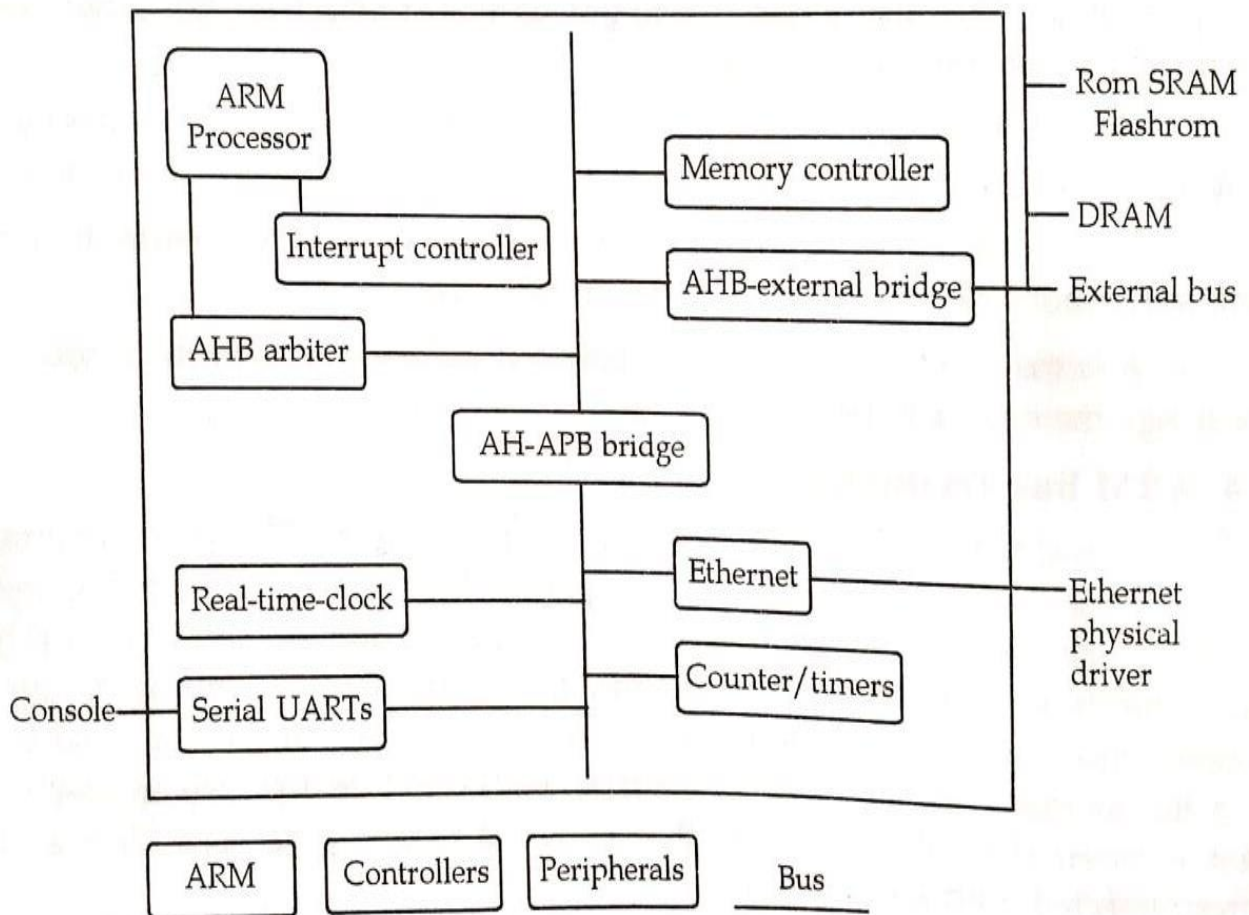
3) **Price:** Embedded systems are price sensitive, use slow and low-cost memory devices.

4) **Size:** Embedded systems should have reduced size so that small area is used by the processor.

5) **Hardware Debug Technology:** Software engineer can view what is happening while processor is executing code. Reduces time to market and overall development costs.

6) **Processor Core:** ARM core is enhanced RISC architecture.

Embedded System Hardware:



The typical ARM core embedded device consists of 4 main hardware components.

1) **ARM processor:** It controls the embedded device and is interfaced with memory management unit and caches with a bus.

2) **Controllers:** It co-ordinates important functional blocks of the embedded system.

a) Memory controller: They connect memories like SRAM, DRAM and flash ROM etc to processor bus. On power-up it is configured in hardware to allow certain memory devices to be active.

b) Interrupt Controllers: Provides communication of a device, which needs service from the processor. There are 2 types of interrupt controller

i) Standard Interrupt Controller.

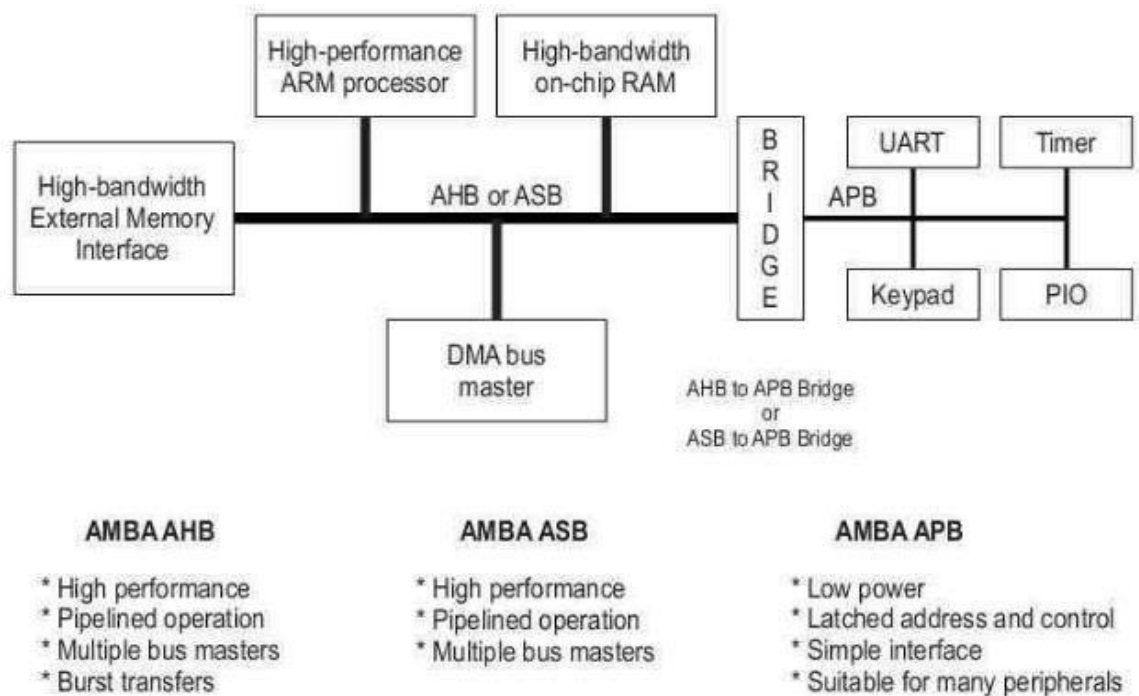
ii) Vector Interrupt Controller.

3) Peripherals: They provide input/output capabilities of embedded system outside the chip.

4) Bus: It is used to communicate between different parts of the device.

AMBA Bus Protocol:

ARM Company has given AMBA Bus Protocol for communication between processor and peripheral. AMBA stands for Advanced Microcontroller Bus Architecture. It was introduced in 1996. The first AMBA buses introduced were ARM System Bus (ASB) and ARM Peripheral Bus (APB). Later ARM introduced another bus called ARM High Performance Bus (AHB).



ASB and APB are bidirectional bus designs. AHB has a centralized multiplexed bus scheme hence provides higher data throughput.

AHB has 2 variants

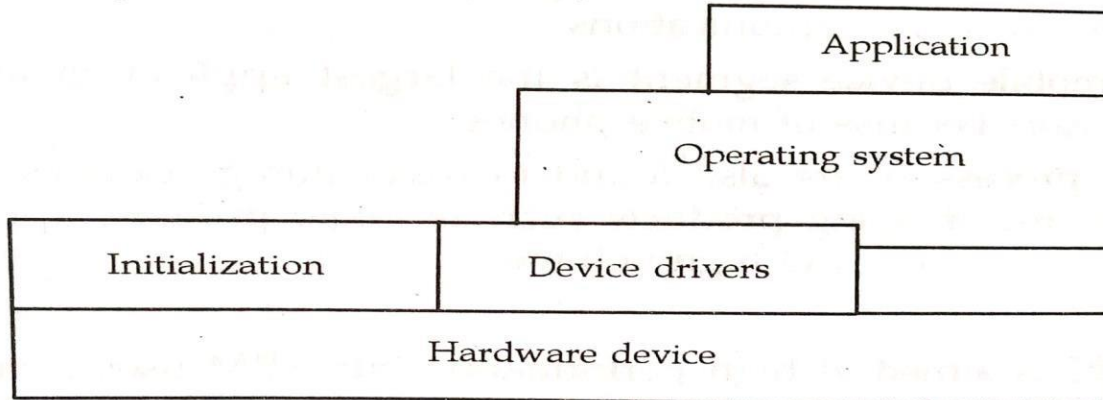
(a) Multilayer AHB: It allows multiple bus masters to be active on the bus any time.

(b) AHB Lite: It has single bus master. It is a subset of the AHB bus with fewer of standard AHB bus.

Embedded System Software:

It has 4 typical software components required to control embedded device. Each software components are separated from each other and hardware device. But each component communicates with each other.

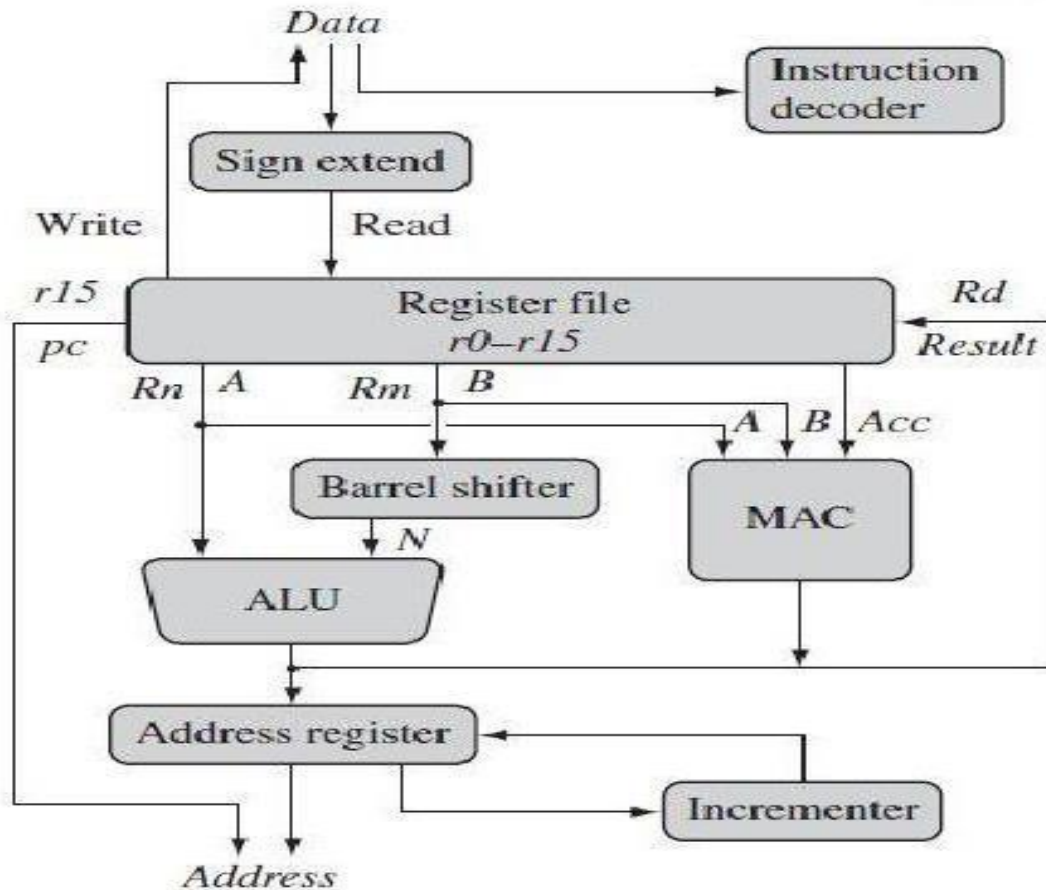
Software abstraction layers executing on hardware:



- 1) **Hardware Device:** It is a board on which processor exists.
- 2) **Initialization:** It is the first code executed on the board. Initialization code does the tasks into 3 phases.
 - i) **Initial hardware Configuration:** It sets-up the target platform so that it can boot an image.
 - ii) **Diagnostics:** It tests the system to check if the target is working.
 - iii) **Bootting:** It involves loading an image and handing control to that image.
- 3) **Device Drivers:** These provide a consistent software interface to the peripherals on the hardware.
- 4) **Operating System:** It organizes and manages hardware system resources like peripherals, memory and processing time. ARM supports 50 operating systems.
- 5) **Application:** It performs the task specific duties of an embedded system.

Applications of ARM Processors:

- 1) Networking applications like home gateways, DSL modems and wireless communication.
- 2) Mobile devices li phones.
- 3) Robotics applications.
- 4) Mass storage devices like hard drives.
- 5) Imaging products such as inkjet printers.
- 6) Video games.
- 7) It is used in cost sensitive and high-volume applications.

Data flow model of ARM core:

It shows Von Neumann implementation of ARM i.e. data items and instruction share same bus.

1) Instruction decoder: This translates instructions before they are executed.

2) Register file: It is a storage bank of 32-bit register, which can hold signed or unsigned 32-bit value.

3) Sign extended: The hardware converts signed 8-bit and 16-bit numbers to 32-bit values.

4) ALU or MAC (Multiply –Accumulate unit): MAC takes the register value Rn and Rm from **A** and **B** buses and computes a result. Load and store instructions use the ALU to generate an address to be held in the address register and sent on address bus.

5) Barrel Shifter: Barrel shifter along with ALU can calculate a wide range of expressions and addresses.

6) Address Register: The incrementer updates the address register before the core reads or write the next register values.

Register File of ARM:

There are 16 data registers and 2 processor status registers. Total 18 active registers.

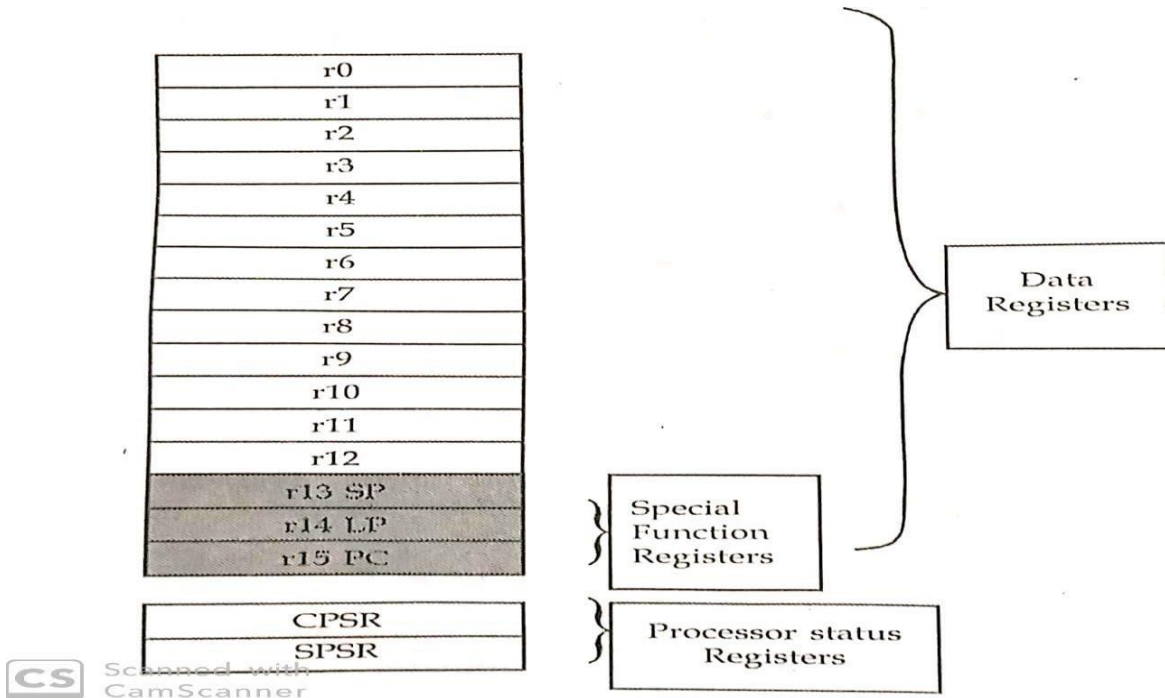
Register R13 is Stack Pointer (sp) stores top of the stack in the current processor mode.

Register R14 is Link Register (lr) which holds the return address when a subroutine call is executed.

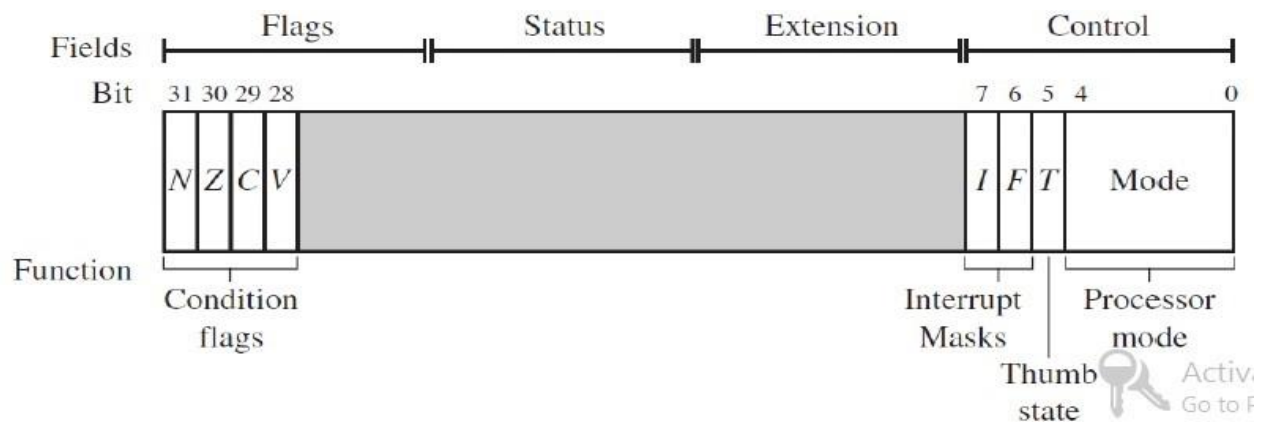
Register R15 is Program Counter (pc) which contains the address of the next instructions to be fetched by the processor.

There are 2 program status registers cpsr and spsr (current and saved program status register respectively)

User mode register:



Structure of CPSR (Current Program Status Register):



CPSR is divided into 4 fields each 8-bit wide: Flags (bit 24 to 31), Status (bit 16 to 23), Extension (bit 8 to 15) and Control bit 0 to 7. The extension and status fields are reserved for future use.

Condition Flags

N- Negative Flag Bit (31): It is set to 1 if result of arithmetic or logical operation is negative otherwise reset to 0.

Z- Zero Flag Bit (30): It is set to 1 if result of arithmetic or logical operation is zero otherwise reset to 0.

C- Carry Flag Bit (29): It is set to 1 if result of 32-bit arithmetic operation generates carry, otherwise reset to 0.

V- Overflow Flag Bit (28): It is set to 1 if result of 32-bit signed integer arithmetic operation generates carry, otherwise reset to 0.

Control Flags

IRQ (Interrupt request) Bit 7: When set to 1, IRQ interrupts are disabled, when 0 it is enabled.

FIQ (Fast Interrupt request) Bit 6: When set to 1, FIQ interrupts are disabled, when 0 it is enabled.

T (Thumb Instruction) Bit 5: When set to 1, processor switches to thumb state. When reset to 0, the processor is in ARM state.

Processor Mode Bits: Bits 4 to 0 select the different processor modes

Bits [4:0]	Mode
10000	User mode
10001	FIQ mode
10010	IRQ mode
10011	Supervisor mode
10111	Abort mode
11011	Undefined mode
11111	System mode

Processor Modes:

There are 7 processor modes, 6 privileged modes (abort FIQ, IRQ, supervisor, system and undefined) and one non-privileged mode (user).

Abort mode: Processor enters into this mode when memory access fails.

FIQ and IRQ: Two interrupt level mode.

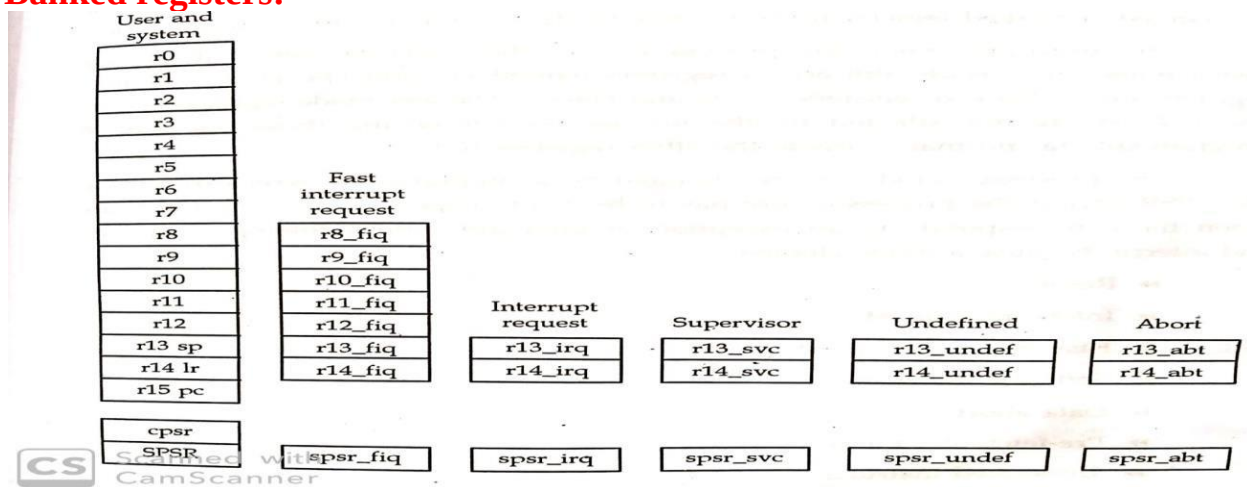
Supervisor mode: Processor enters into this mode after reset and OS kernel operates in this mode.

System mode: Special version of user mode that allows full read-write access to CPSR.

Undefined mode: Processor enters into this mode when undefined instruction is encountered.

User mode: Used for programs and applications.

Banked registers:



There are 37 registers in the register file of ARM processor. Out of 37, 20 registers are hidden from a program at different times. These registers are called banked registers and are identified by the shading in diagram. These registers are available in a particular mode. Banked registers of a particular mode are denoted by an underline and mode name.

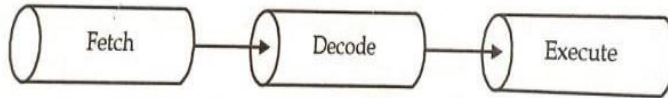
Every processor mode except user mode can be changed by writing directly to the mode bits of the CPSR.

For ex when processor is in IRQ mode the instructions executed will access the banked registers r13_irq and r14_irq. The user mode registers r13_usr and r14_usr are not affected by the instruction. a program still has normal access to register r0 to r12.

Pipeline:

It is a method of executing instructions in parallel. There are 3 stages of pipelining Fetch, Decode and Execute.

Three stage pipeline:

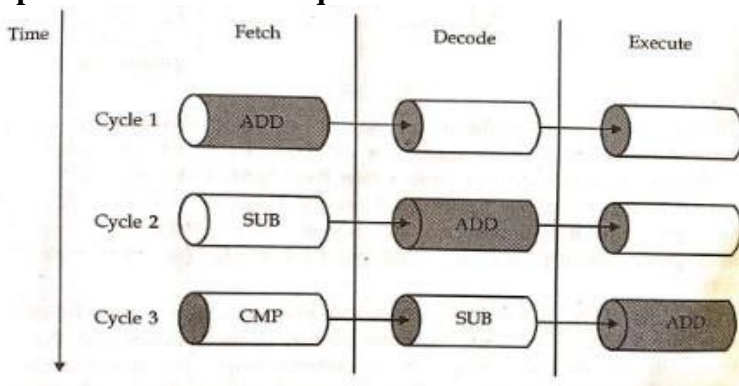


Fetch loads an instruction from memory.

Decode identifies the instruction to be executed.

Execute executes instruction and write the result back.

Pipelined instruction sequence:



In 1st cycle processor fetches ADD instruction from memory. In 2nd cycle processors fetches SUB instruction and decode ADD instruction. In 3rd cycle processors fetches CMP instruction, decode SUB instruction and executes ADD instruction.

Characteristics of pipeline:

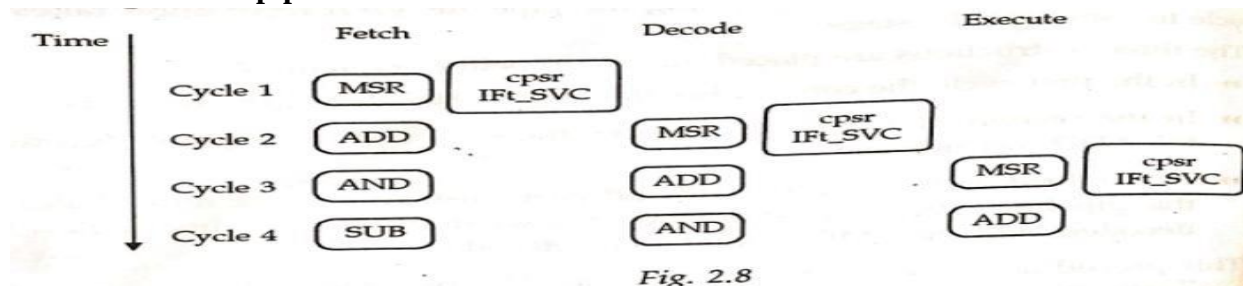
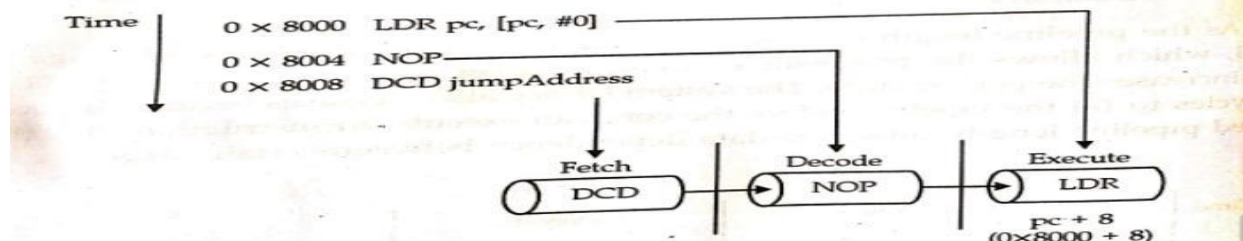


Fig. 2.8



MSR instruction completes execute stage of the pipeline. It clears I-bit in CPSR to enable IRQ interrupts. Once the add instruction enters execute stage IRQ interrupts are enabled. PC always points to the address of instruction +8 bytes.

- 1) The ARM pipeline will not process an instruction till it passes through execute stage.
- 2) The PC always point to the address of instruction in execute stage plus 8-bytes in ARM.PC always points to address of instruction plus 4-bytes in THUMB.
- 3) The execution of a branch instruction causes the ARM core to store (flush) the pipeline.
- 4) ARM 10 uses branch prediction this reduces pipeline storing.
- 5) Execution of instruction will be completed even though an interrupt is raised.

Likelihood questions:

5 Marks

Remember

1. List advantages & drawbacks of RISC.
2. List the features of ARM instructions suitable for embedded applications.
3. List the special features of ARM processor design.
4. Define pipelining & high code density.
5. Define processor modes. List all different processor modes.
6. Define banked registers. Explain.
7. Describe the functions of flags of CPSR register.
8. Describe the functions of IFT bits of CPSR register.

Understand

1. Explain AMBA bus protocol.
2. Explain the active registers used in user mode.
3. Explain 3-stage pipe line.
4. Explain processor modes.
5. Explain the role of software components in an embedded system.

Application

1. List the applications of ARM processor.
2. Sketch a neat ARM core data flow model.
3. Write the advantages & disadvantages of pipelining.
4. Discuss the process of filling the pipeline with a neat sketch.

Analyse

1. justify how ARM is suited to perform DSP type function.
2. justify how ARM is suitable for mobile applications.
3. justify the features that improves code density.

10 Marks**Understand**

1. a) Explain AMBA bus protocol. (7)
b) Discuss mode bits of CPSR register. (3)
2. a) Explain the two architectural levels of Bus. (3)
b) Explain 3-stage pipelining of ARM-7 with example. (7)
3. a) Explain register file of ARM processor with a neat sketch. (7)
c) Explain the function of special function registers of ARM. (3)
4. a) Explain different processor modes. (7)
d) Explain the AMBA bus variants. (3)
5. Explain banked registers with a neat diagram.
6. Explain the bit structure of CPSR.
7. Explain ARM core data flow model.
8. Explain the block diagram of ARM based embedded device.