Overview of 8086 microprocessor and ARM Processor

Department of Electronics Communication Engg.
ITER, S'O'A Deemed to be University, Odisha

On completion of the lecture the students will be able to -

- Understand the over view of 8086 and ARM processors, their architecture and addressing modes.

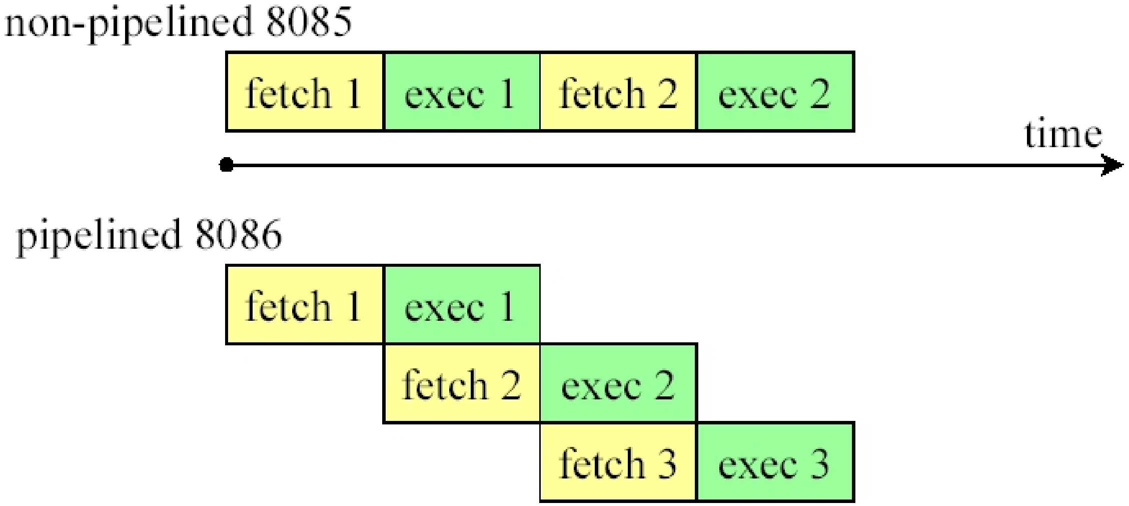
8086

Microprocessor

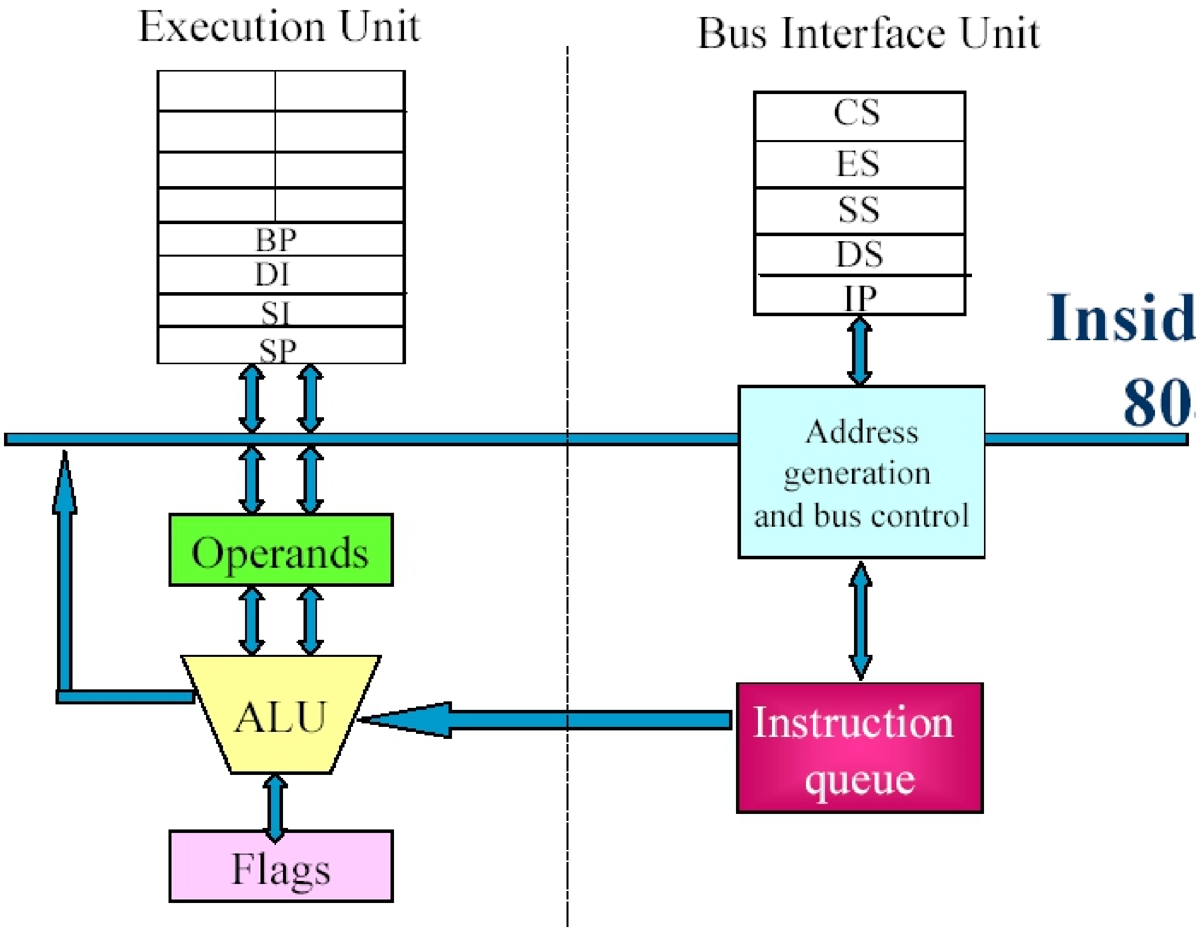
(Already covered, just go thro')

8086 Microprocessor

- 16 bit processor, 20 address bus, 5 MHz, 29000 transistors
- Developed in 1978, capable of addressing 1 M bytes of memory
- Pipelining in 8086 is to allow CPU to fetch and execute at the same time

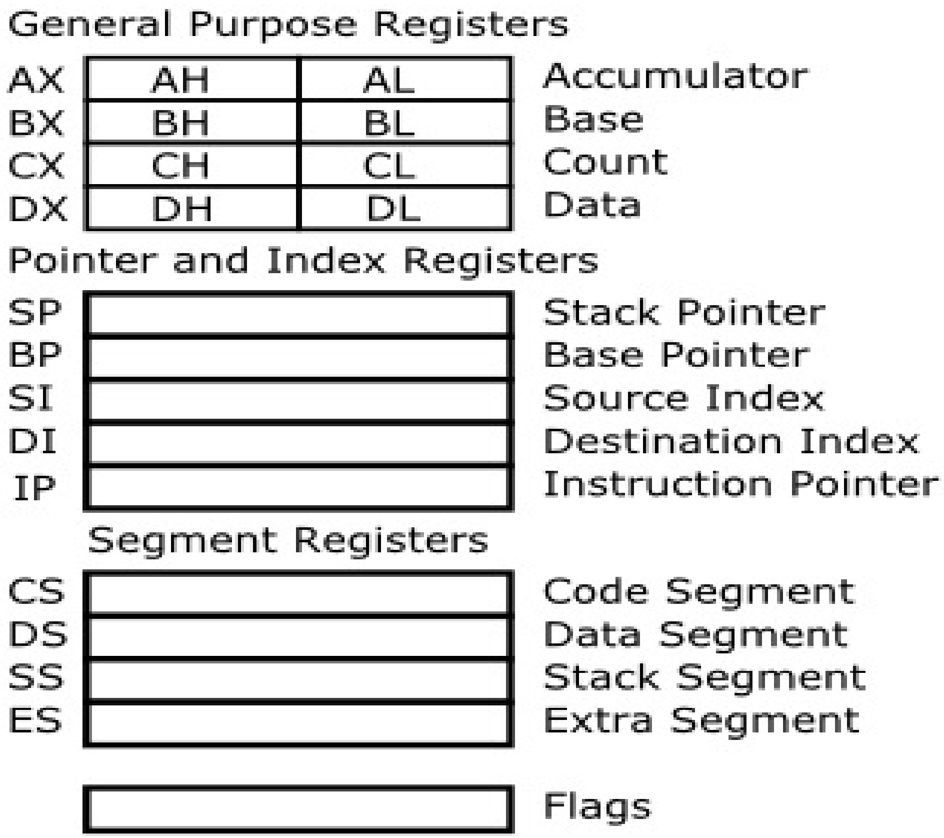


Internal Architecture of 8086



Inside the 8086

Registers in 8086 Microprocessor



General Purpose Registers

AX	AH	AL	Accumulator
BX	BH	BL	Base
CX	CH	CL	Count
DX	DH	DL	Data

Pointer and Index Registers

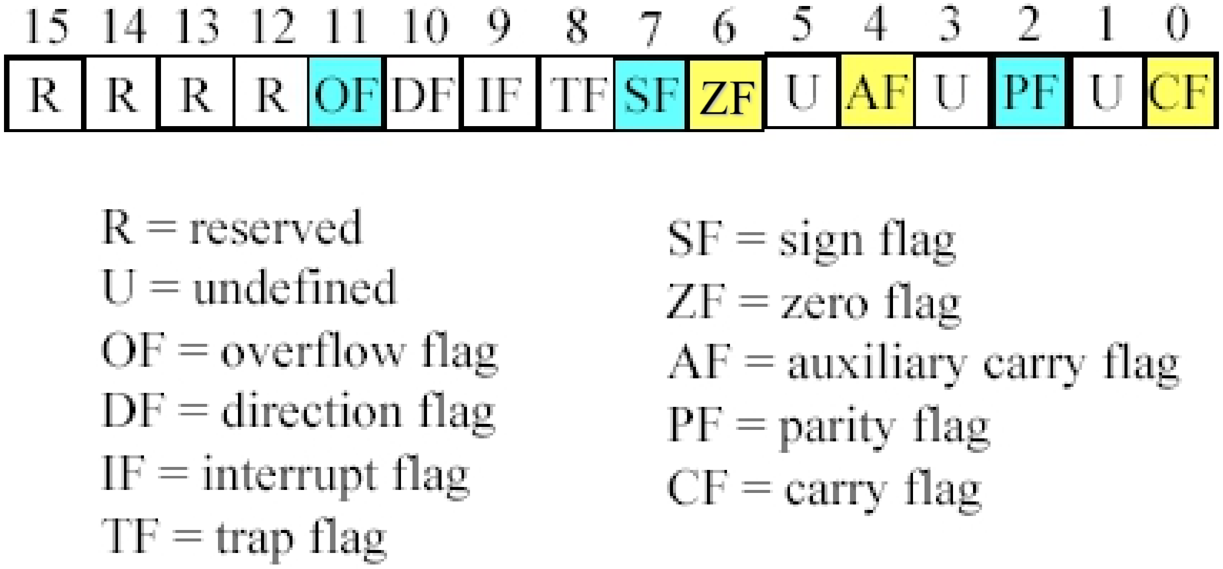
SP		Stack Pointer
BP		Base Pointer
SI		Source Index
DI		Destination Index
IP		Instruction Pointer

Segment Registers

CS		Code Segment
DS		Data Segment
SS		Stack Segment
ES		Extra Segment
		Flags

Flag Register

- Flag Register (status register)
  - 16-bit register
  - Conditional flags: CF, PF, AF, ZF, SF, OF
  - Control flags: TF, IF, DF



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R	R	R	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF

R = reserved
U = undefined
OF = overflow flag
DF = direction flag
IF = interrupt flag
TF = trap flag

SF = sign flag
ZF = zero flag
AF = auxiliary carry flag
PF = parity flag
CF = carry flag

Addressing Modes of 8086

1.Sequential Control Flow instructions

2. Control Transfer instructions

Addressing Modes of sequential control flow instructions

1. Immediate Addressing Mode:

   Example: MOV AX,0005H

2. Direct Addressing Mode:

   Example: MOV AX, [5000H]

   Effective address: 10H*DS+5000H

3. Register Addressing Mode:

   Example: MOV AX,BX

Addressing Modes of sequential control flow instructions

4. Register Indirect addressing mode:

Example: MOV AX,[BX]

Effective Address: 10H*DS+[BX]

5.Indexed addressing mode:

Example: MOV AX,[SI]

Effective Address: 10H*DS+[SI]

6.Register Relative addressing mode:

Example: MOV AX,50H[BX]

Addressing Modes of sequential control flow instructions

7. Based Index addressing mode:

Example: MOV AX,[BX],[SI]

Effective address: 10H*[DS]+[BX]+[SI]

8. Relative Based Index addressing mode:

Example: MOV AX,50H[BX],[SI]

Effective address: 10H*[DS]+[BX]+[SI]+50H

ARM

Processors

(Already covered just go thro')

What Is ARM ?

- Advanced RISC Machine
- First processor used for commercial purpose
- Developed at Acorn Computers Limited between 1983-85
- It has architectural Simplicity which results low power consumption
- Used in video game, modems, mobile phones, handy cams etc

ARM Features

RISC:

Large Uniform Register File

Load and Store Architecture

Simple Addressing Mode

Uniform fixed length instruction field

ENHANCED FEATURES:

Each Instruction controls the ALU and shifter

Auto Increment and auto decrement addressing mode

Multiple load/store

Conditional Execution

ARM Architecture

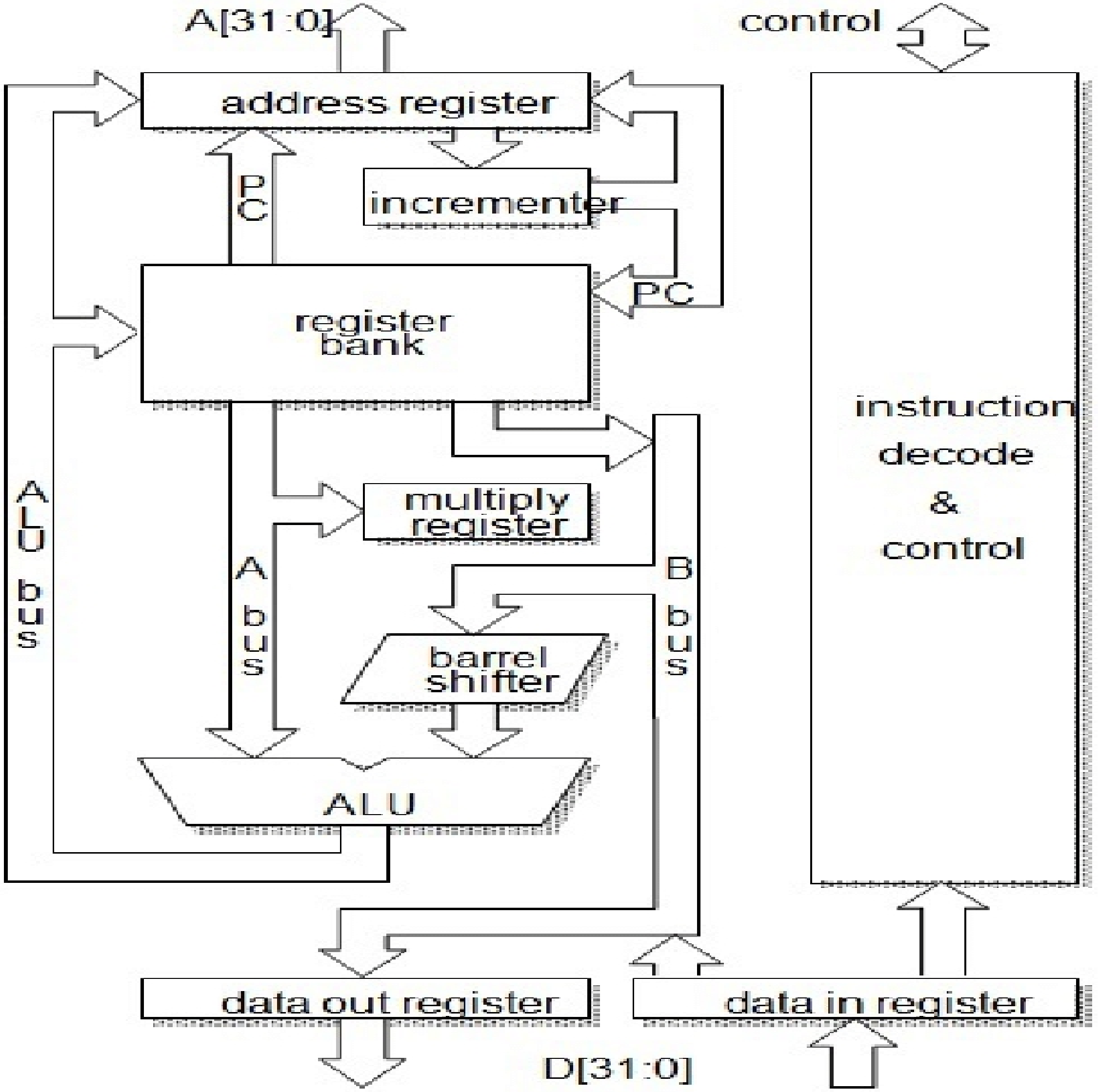
Based on Berkeley RISC Machine ARM architecture has:

- fixed length instruction
- Pipelines
- load/store architecture
- 32 bit architecture
- In relation to ARM word length means 4 Bytes

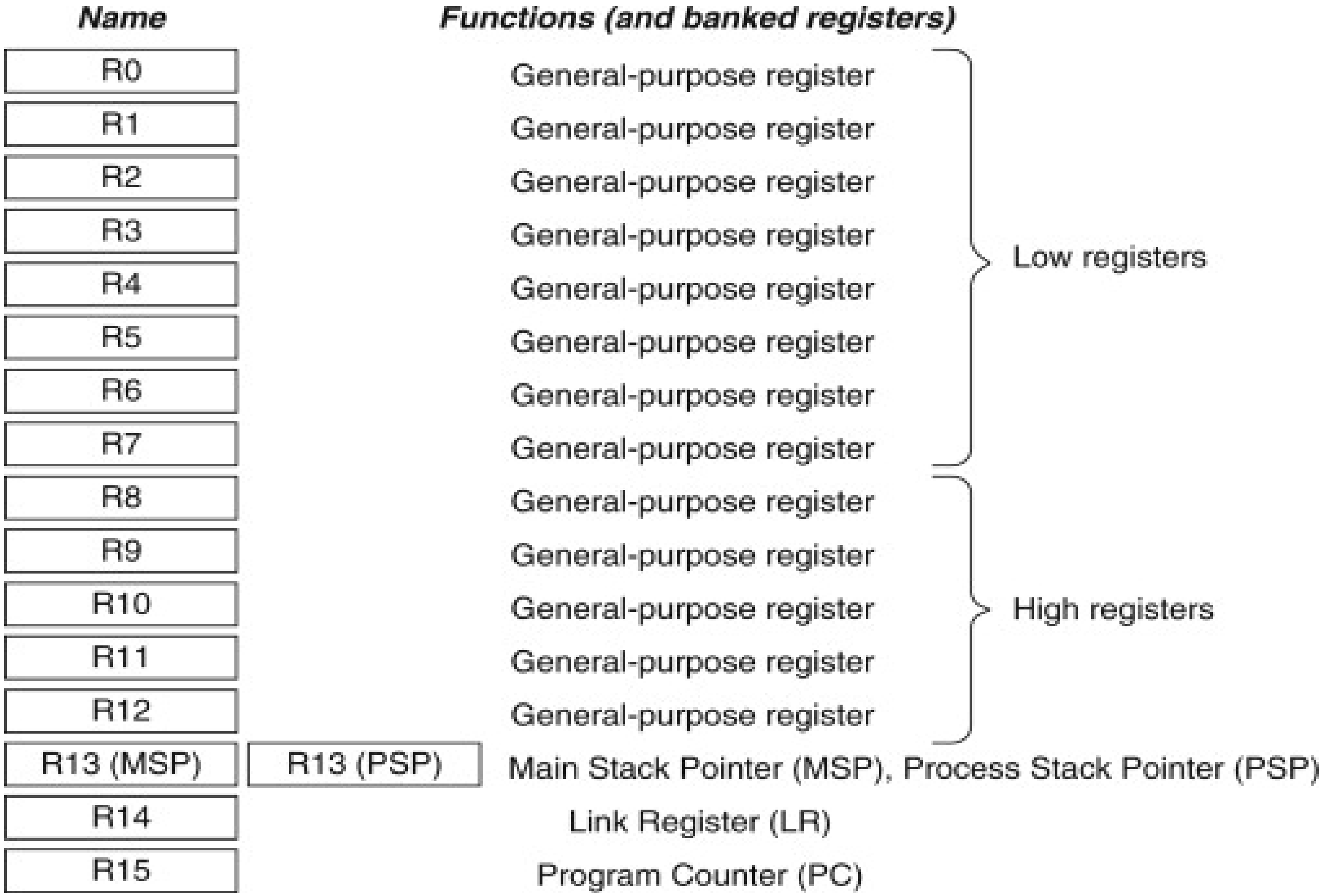
Most ARM implements 3 (2+1) types of instruction set

- 32 bit ARM instruction set
- 16 bit Thumb instruction set
- Jazelle instruction set (implemented using JAVA)

ARM Architecture

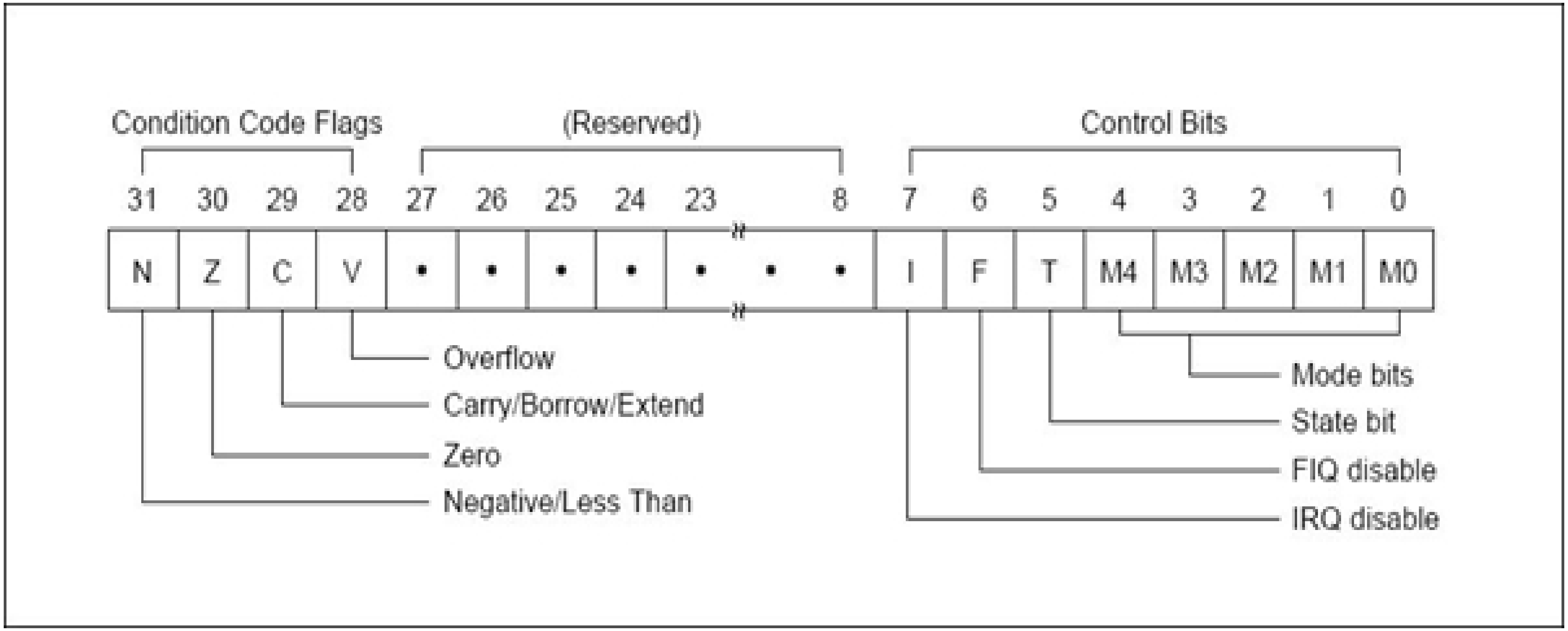


ARM Register Organization



Name		Functions (and banked registers)	
R0		General-purpose register	Low registers
R1		General-purpose register	
R2		General-purpose register	
R3		General-purpose register	
R4		General-purpose register	
R5		General-purpose register	
R6		General-purpose register	
R7		General-purpose register	
R8		General-purpose register	High registers
R9		General-purpose register	
R10		General-purpose register	
R11		General-purpose register	
R12		General-purpose register	
R13 (MSP)	R13 (PSP)	Main Stack Pointer (MSP), Process Stack Pointer (PSP)	
R14		Link Register (LR)	
R15		Program Counter (PC)	

CPSR (Current Program Status Register)



Addressing Modes:

- The general syntax is

```
<opcode>{condition} {S} <Rd>,<Rn>,<shifter operand>
```

```
Example:  MOV R0,R1
          MOVCS R0,R1
          MOV R0,#0
          MOVS R0,#0
```

Data Processing Instructions:

- Immediate addressing Mode

  syntax: #<immediate>

  Examples: MOV R0,#0
  ADD,R3,R3,#1

- Register addressing Mode

  syntax: <Rm>

  Examples: MOV R2,R0
  ADD R4,R3,R2

Data Processing Instructions:

- Shifted register operand addressing mode

Syntax: <Rm>, shift/rotate #<immediate>

A shifter register operand value is the value of a register shifted(or rotated) before it is used as the data processing operand.

Examples: MOV R2,R0,LSL #2
ADD R9,R5,R5,LSL #3
MOV R12,R4,ROR R3

ASR: Arithmetic Shift Right
LSL: Logical Shift Left
LSR: Logical Shift Right
ROR: Rotate Right
RRX: Rotate Right with extend

Addressing Modes: Load Store

- Immediate offset:

  Example: LDR R4,[R2,#5]

- Register Offset

  Example: LDR R4, [R2,R3]

- Scaled Register Offset

  Example: STR R0, [R1,R2,LSL #2]

- Immediate pre indexed:

  Example: STR R0,[R1,#2] !

Addressing Modes: Load Store

- Register pre indexed:

  Example: LDR R4,[R2,R3] !

- Scale Register Pre indexed

  Example: LDR R4, [R2,R3,LSL #2]!

ASSIGNMENTS

8086 BASED PROGRAMMING

Write a program

1. To study the different addressing modes of 8086 microprocessor
2. To find sum of two BCD numbers
3. To calculate the average of n 16 bit numbers
4. To calculate the largest and smallest number in an array
5. Arrange the data elements in an array in ascending and descending order

Contd.

ARM BASED PROGRAMMING

Write a program

1. Move the content of one 16 bit value to another 16 bit variable.
2. Perform 16 bit addition between two numbers
3. Find 1’s and 2’s complement of a number.
4. Disassemble a byte into its high and low order nibbles
5. Find the larger of two numbers.

Thank

You