

2/11/2020

## ASSIGNMENT 1

List and differentiate b/w μP and μC with neat diagrams.

μP

- ① Heart of computer s/m.
- ② Only a processor, so mly and I/O components need to connect externally.
- ③ circuit large.
- ④ cost is high.
- ⑤ External component, so power consumption is high. So not ideal to use battery.
- ⑥ Most have no power saving features.
- ⑦ less register, so more mly based operation.
- ⑧ CPU on single silicon chip.
- ⑨ based on von neumann.
- ⑩ no RAM, ROM, I/O device, timer.
- ⑪ use external bus.
- ⑫ μP based s/m can run at high speed because of technology used.
- ⑬ complex and expensive with large number of instructions.
- ⑭ Few popular μP and features.

8080 - world's first with 8 bit machine  
8 bit data path.

μC

- ① heart of ES.
- ② μC has processor along with mly and I/O components.
- ③ compact s/m.
- ④ less cost.
- ⑤ power consumption is less.
- ⑥ has power saving features.
- ⑦ More register, easier to write.
- ⑧ CPU + other peripheral device.
- ⑨ Harvard architecture.
- ⑩ CPU + RAM + ROM + timer + I/O devices embedded.
- ⑪ internal control bus.
- ⑫ can be used 200 MHz.
- ⑬ simple & inexpensive w/r to number of instructions.

Speed

Pentium III	- 450-600 MHz	} level 2 cache with 512 KB size and external type.
Athlon model 1	- 550-700 MHz	
Athlon model 2	- 750-850 MHz	
Pentium pro	- 1500-2000 MHz, 256 KB-1 MB	full core type external
K6 -3	- 350-450 MHz, 256 KB	External full core type.
Duron	- 550-700 MHz, 64 KB	On die On die full core type.
Celeron	- 300-800 MHz, 128 KB	On die "
Pentium II Xeon	- 400-450 MHz, 512-2 MB	External "
Athlon	- 650-1000+ MHz, 256 KB	On die "
Pentium III	- 500-1000+ MHz, 256 KB	On die "

INTEL CORE - 2 - 1.0 GHz to 3 GHz.

64 bit μP.

INTEL i7	- 66 GHz - 3.33 GHz	} 64 bit.
INTEL i5	- 2.4 GHz - 3.6 GHz	
INTEL i3	- 2.93 GHz to 3.33 GHz.	

Q. List popular microcontrollers and its features?

① AVR microcontroller -

- modified Harvard architecture 8 bit  
RISC single chip mcu. first mc to use  
on chip flash memory for program.

② PIC mcu - made by microchip

technology, derived from the PIC 1650  
originally developed by general.

③ ESP 8266 - full TCP/IP stack &

mcu capability, produced by Espressif  
systems in Shanghai, China.

④ STM 32 - family of 32  
bit mc integrated circuit  
by STMicroelectronics.

⑤ INTEL MCS - 51 (8051)

single chip mcu developed  
by intel for embedded systems  
CISC and separate memory  
space for program instructions  
and data.

8 bit, binary compatible,  
some integrate digital  
signal processing.

⑥ ESP 32 - low cost with integrated WiFi and dual mode Blue tooth. Includes antenna switches, RF balun, power amplifiers, low noise receiver amplifiers, filter and power management modules.

employs Tensilica Xensa LX6 microprocessor in both dual core and single core variation

⑦ Infineon T210 Core - 32 bit MC from Infineon. Unles RISC processor, a microcontroller and a DSP in 1 chip package.

⑧ Arduino - CPU - Atmel AVR (8bit) ARM cortex-M0+ - 32 bit from cortex M3 - 32 bit Intel Quark (x86) 32 bit memory - SRAM, storage - flash, EEPROM

① which is the first μP and its features?

Intel 4004 is the 4 bit CPU released by Intel corporation in 1971. first commercial produced microprocessors.

Performance

max. CPU clock rate - 740-750 KHz

Data width - 4 bit

address width - 12 (multiplexed)

physical specification

transistors - 2250

packages - 16 pin DIP

sockets - DIP 16

Architecture and classification

Application Buscom calculator  
arithmetic manipulation.

History

successor - Intel 4040.

min feature size 100 μm.

inst set 4 bit BCD oriented

② which is the first PIC and its features?

TMS 1802 NC  
TMS 1000 - commercially available in 1971.

features:

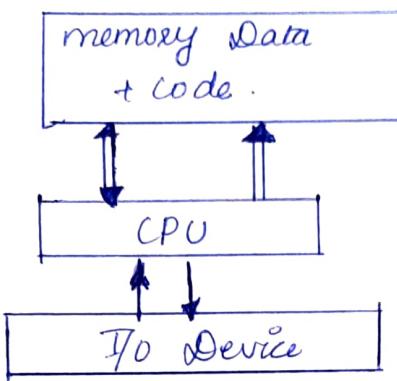
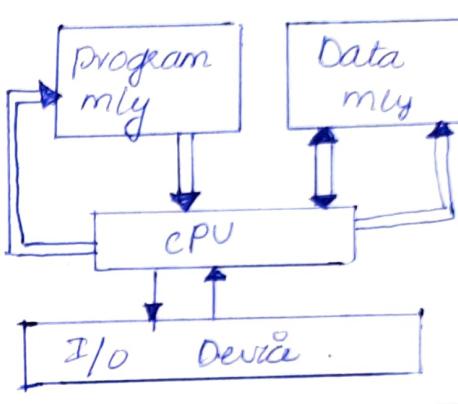
(i) 5000 transistors

(ii) 3000 bits of program memory

(iii) 128 bit of access memory

(iv) possible to program it to perform a range of functions

⑥ With neat diagram explain the difference between Von neumann & Harvard architecture.

	Harvard architecture	Von neumann architecture
Arrangement	<p>The CPU is connected with both the data memory and program memory repeatedly.</p> 	<p>no separate data &amp; program memory. single memory is connected.</p> 
Hardware requirement.	<p>more hardware required.</p>	<p>less hardware.</p>
space	<p>Requires more space</p>	<p>less space.</p>
speed of execution	<p>faster because processor fetches data &amp; inst. simultaneously.</p>	<p>execution is slower since cannot fetch data &amp; inst at a time.</p>
space usage.	<p>wastage of space since if the space of data left, the inst memory cannot use the space in data memory.</p>	<p>Space is not wasted because, data memory can be used for inst. mly and vice versa.</p>
controlling	<p>complex controlling because of simultaneous fetching of data &amp; instruction.</p>	<p>Controlling is simple.</p>

3/11/2020

Q) With neat diagram, explain briefly the architecture of ARM7.

32 bit RISC processor core

37 pieces of 32 bit register registers

pipelined - 3 stages (instn fetch, decode and execute).

Cached

Von Neumann type bus structure

8/16/32 bit data types

7 modes of operation (usr, fiq, svc, abt, sys, und)

Simple structure, reasonable good speed ratio power cons.

ARM7TDMI - is the core processor module embedded in many ARM 7.

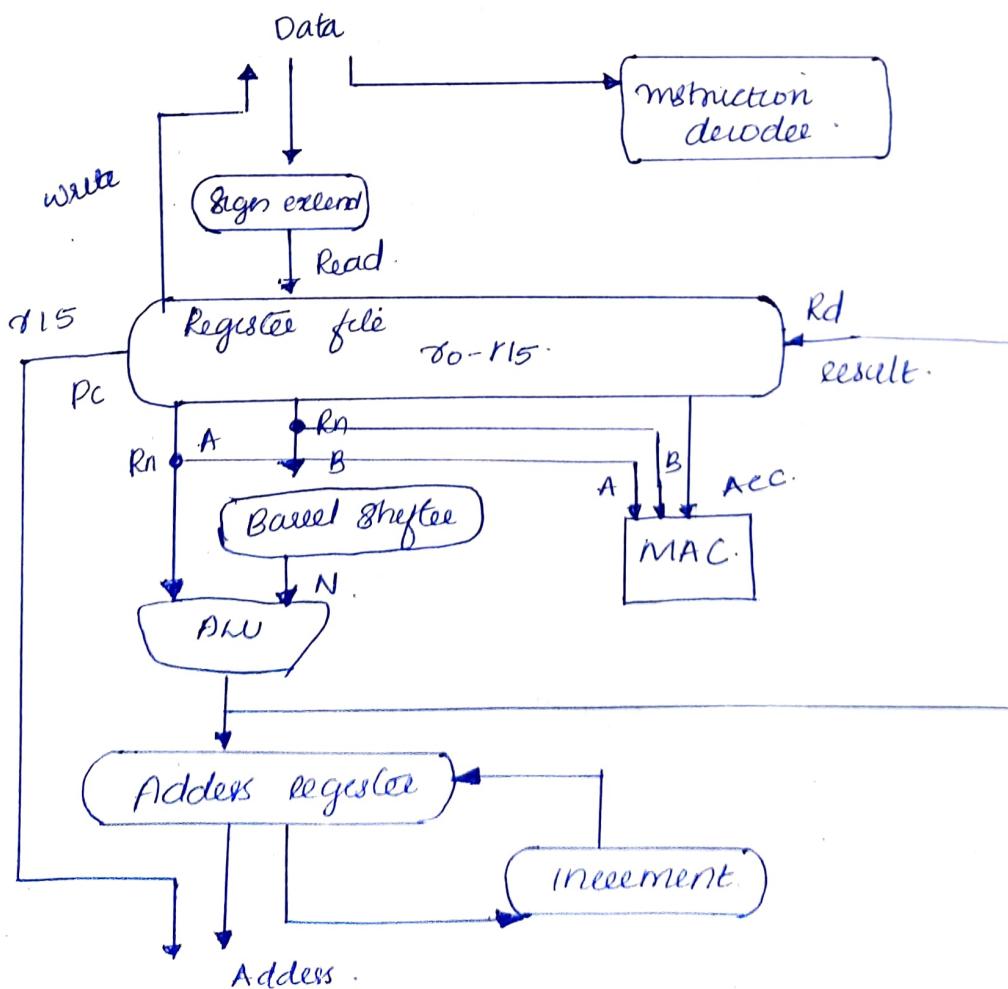
T - Thumb instruction

D - features with IEEE Std. 1149.1 JTAG boundary scan debugging interface

M - features with multiplier and accumulate unit

I → support of embedded In circuit emulator

16 thumb sets, high density core.



① (contd.)

R<sub>x-R15</sub> - 32 bit address lines,  $2^{32}$  - 4 GB memory address space.

PC-(R15) - 32 bit address of instruction to be fetched.

It will periodically increment after fetching address.

Instruction decoder  $\rightarrow$  decodes instruction & generates control signal for execution.

ALU - 32 bit. Obtained the operands from R<sub>0-R15</sub> registers. The result after operation store back in register & flags sets.

Barrel shifter  $\rightarrow$  preshifts operands before giving to the ALU. which means multiple shifts in single clock pulse.

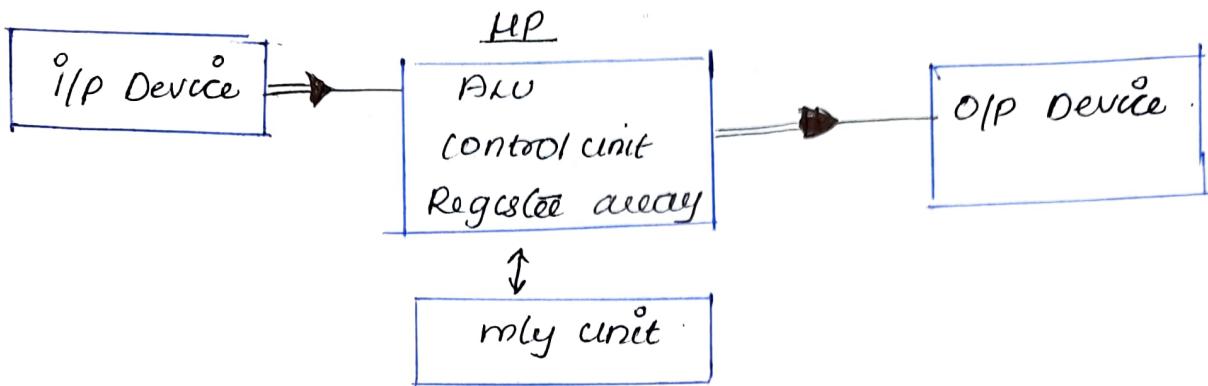
Multiply Accumulate unit (MAC)  $\rightarrow$  complex arithmetic operations like multiply & addition.

Register file - 16 registers (R<sub>0-R15</sub>). among them R<sub>15</sub> (PC) register are blank.

R<sub>14</sub> (LC) SPC(R<sub>13</sub>) and some changes with operation mode.

Address register - hold address of memory operands during load store instructions.

8. With neat diagram explain the block diagram of MP. Explain the units of microprocessor.



1. ALU → actual numeric and logic operation are doing in ALU.

① consist accumulator (8 bit register) to store data and perform arithmetic and logic operations.

- ② flag -
- S - Sign
  - Z - Zero
  - AC - Auxiliary Carry
  - P - Parity
  - CY - carry flag

2. Timing and control unit - generates control sig necessary for communication between MP and peripherals.

3. Instruction register and decoder.

- part of ALU

decoder - decodes the instructions & establish the sequence to follow

IR  $\rightarrow$  It is not programmable.

4. Register area:

B, C, D, E, H, L - 6 8 bit general purpose register.

SP - stack pointer PC - program counter.

One increment & decrement counter.

5. System bus:

data bus - data (8 bit / 16 bit)

address bus - address of operand (16 bit).

control bus - Read/write control logic.

6. Interrupt control -

suspends the routine what is MP doing & brings the control to perform subroutine & completes & returns to main routine.

INTR, TRAP, RST 7.5, 6.5, 5.0.

7. Serial I/O control.

SIO, SOO } to implement serial transmission.

Q. List the architectural differences b/w 8085 v/s 8086.

	8085	8086
Data bus size	8 bit	16 bit
Address bus size	16 bit	20 bit
Clock speed	3MHz	5.8 - 10 MHz
Duty cycle for clock	50%	33%

Data bus size

Address bus size

Clock speed

Duty cycle for clock

Flags

Pipelining

Memory segmentation

No. of transistors

Processor type

presence of max & min mode.

No. of processor

Memory size

1/28 t.

Inst. Queue support

8085

5 flags

Z, S, NC, P, C.

Does not support

Does not support

6500

Accumulator based

Not present

Only one.

64 KB

No multiplication & division inst.

Does not support

8086

9 flags

Overflow, Direction, Intercept, Trap, S, Z, AC, P, C.

Supports

Supports

29000.

General purpose register based.

Present.

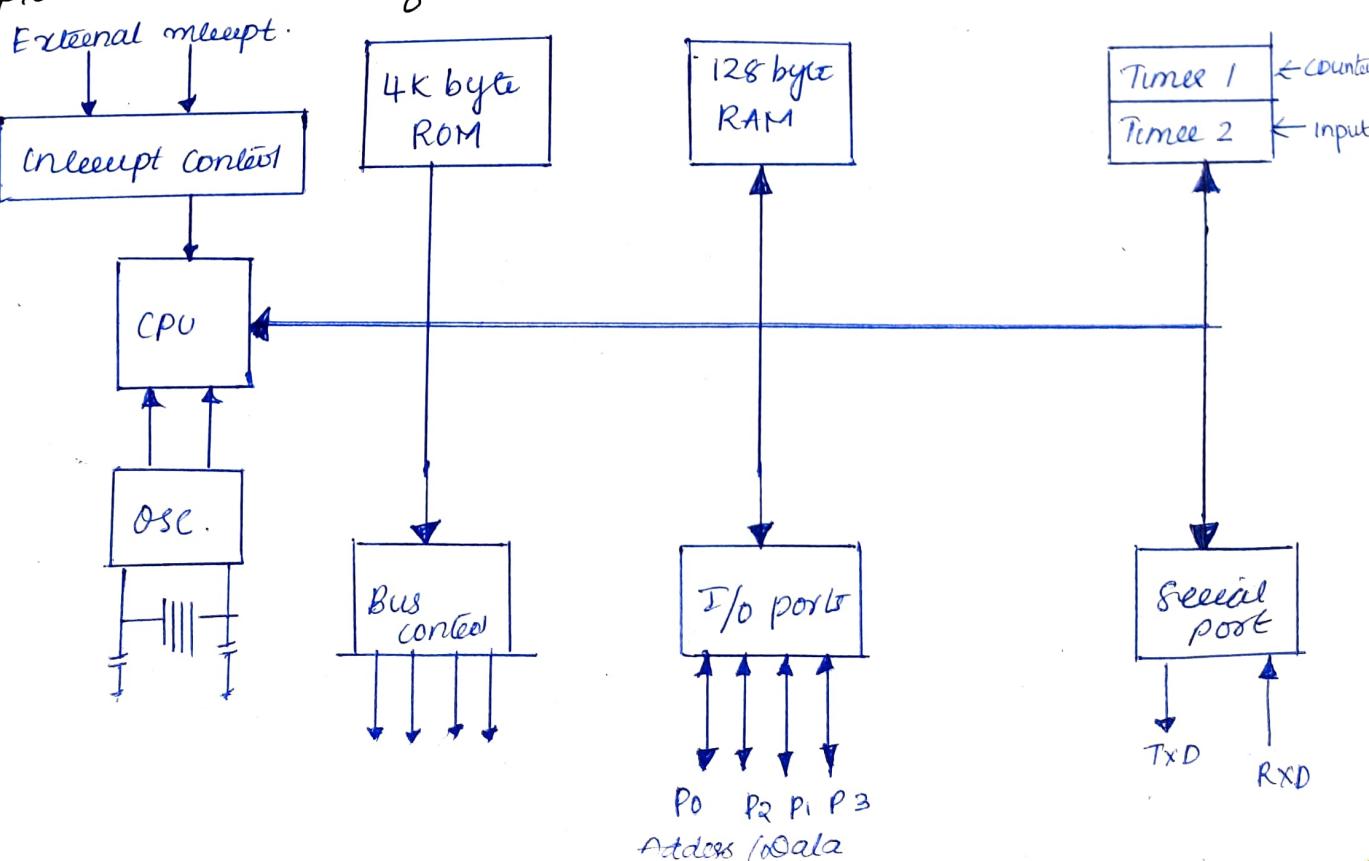
More than 1 processor  
Additional can be employed

1 MB.

Multiplication & Division  
Operations are present

Support.

- Q10. With neat diagram, explain the block diagram of microcontroller.  
Explain the units of microcontroller.



8 bit microcontroller - width of the data bus is 8 bits. data can process 8 bit of data at one time.

memory - MC needs program memory to store pgm/instr. to perform defined task. This mly is called ROM. Also it needs memory to store the Operands / data on Composeay basis. This is RAM. In MC, 4 kb on chip ROM and 128 bytes RAM (8051).

Address bus - A bus of microcontroller can be defined as a group of wire act as medium for transfr. data. (16 B)

Interrupts - It will

- Temporarily suspend the ongoing program
- pass the control to subroutine
- ~~Resume~~ execute subroutine
- Resume the ongoing / main program.

Hardware & software interrupts are there.

INT0 - external hardware

TF0 - Timer 0 overflow interrupt.

INT1 - External Hardware interrupt

TF1 - Timer 1 overflow interrupt

R1/T1 - serial comm. interrupt.

I/P - O/P port -

and also ① 2 16 bit timer & counters

② 1 Data pointer & PC of 16 bit each

③ 128 user defined flags

④ 4 Register banks

⑤ 31 general purpose Registers which are 8 bit each.

(11) what are the addressing modes Explain in general?

① Implied mode - operand specified in inst. itself. dat is 8 bit or 16 bit and is part of instruction. Two address inst. are designed with implied address mode.

eg: CLC (clear reset carry).

② Immediate - data is present in address field of instr.



Eg: MOV AL, 35H.

③ Register mode - operand placed in one of 8 bit/16 bit general purpose registers.



MOV AX, CX.

④ Register indirect mode - addressing operand's offset is placed in any of register BX, BP, SI, DI as specified in inst.

2 register reference is required to access data.

MOV AX, (BX).

⑤ Auto indexed (increment & decrement mode).

After accessing the operand, the contents of register automatically incremented to point to next memory location (increment mode).

$$\text{Add } R_1, (R_2) + \quad R_1 = R_1 + M[R_2]$$

$$R_2 = R_2 + d.$$

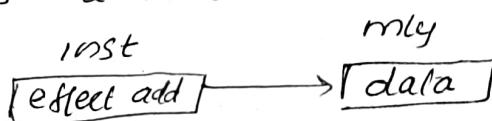
In decrement mode, it will decrement to the memory location

$$\text{Add } R_2, -(R_2)$$

$$R_2 = R_2 - d$$

$$R_1 = R_1 + M[R_2]$$

⑥ Direct addressing - Only memory reference is required to access the data



Add AL, [0301].

⑦. Indirect addressing mode (by @).

- 2 types
  - ① Register indirect: effective address is in register and corresponding register name will maintained in address field.
  - ② Memory indirect: effective address is in memory and corresponding address my will maintained in address field.

⑧ Indexed addressing mode: sum of content of index register SI or DI & 8 bit or 16 bit displacement

mov AX,[SI+05]

⑨. What is code density? What does ARM does to improve code density?

The amount of space that an executable pgm takes up in memory. Code density is important in the devices that contains limited amount of memory. Code density refers loosely to how many MP instructions it takes to perform a requested action and how much space each instruction takes up. If less space an instruction takes and more work it can do, then it has high code density.

Program can achieve maximum performance in ARM with minimum inst. The Thumb inst. set offers much increased code by reducing code size & my requirement. The thumb's set is 16 bit inst. Length allows it to approach about 65% of standard ARM code size while retaining ARM 32 bit processor performance. Code can switch between ARM and thumb in procedure call.

⑩. What is endianness. With neat example, discuss type of endianness.

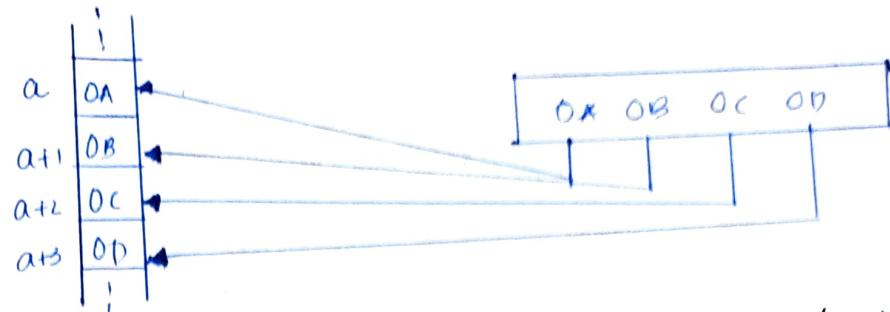
In computing, endianness is the order or sequence of bytes of words of digital data in computer memory. Endianness is primarily expressed as big endian & little endian.

⑪ Big endian

First byte of binary representation of multibyte is placed ~~as~~ first.

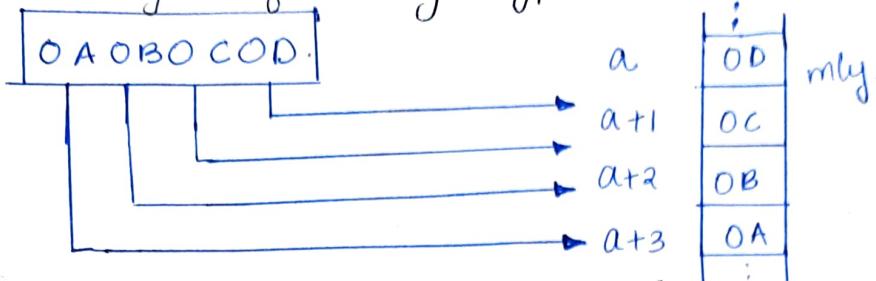
Eg 0x 0A 0B 0C 0D

4/11/2020



little endian

The last byte of binary type data will stored first.



- Q15) What are the difference between Assembly language & High level language. Discuss on the preference on High level language over Assembly language.

	High Level language	Low Level language
1.	programmer friendly language	machine friendly language
2.	language is less memory efficient	high memory efficient
3.	Easy to understand	tough to understand
4.	Simple to debug	Complex to debug
5.	simple to maintain	complex to maintain
6.	portable	non portable
7.	can run on any platform	machine dependable
8.	compiler/interpreter for translation	needs assembler for translation
9.	widely used for programming	not commonly used now a days

High level language is preferred because it will run on any platform and easy to understand and it is programmer friendly. Also it is simple to debug.

24/10/2022  
MAY/MAR

## Differentiate RISC and CISC?

### RISC

- Reduced set of instr. set
  - One clock pulse, 1.5 clock per instr.
  - performance more on software
  - no memory unit & separate <sup>hard</sup> <sub>ware</sub>.
  - Hardwired unit
  - Inst. set is limited
  - multiple register set
  - complex inst. synthesis with software.
  - highly pipelined.
  - Complexity lies in compiler that executes program
  - execution time is very less
  - code expansion is problem.
  - decoding inst. is simple.
  - not require external memory.
- Eg: Alpha, ARC, ARM, MIPS, PA-RISC.

### CISC

- complex inst. set comp.
- 2-15 clock cycle/instr.
- based on hardware.
- memory unit
- microprogrammed unit
- vast inst. set.
- single reg. set.
- support complex inst.
- not pipelined /less pipeline
- complexity lies in FP.
- execution time high.
- code expansion <sup>is not</sup> problem.
- Decoding inst. is complex.
- Requires external memory.
- Eg. Motorola 68000, AMD intel x86 CPUs.

(16.) With neat diagram explain programming model of ARM7.

ARM has 7 basic operating mode:

User - normal program execution mode

FIQ - Used for handling a high priority interrupt

IRQ :- used for handling a low priority interrupt

Supervisor : entered on reset and when software interrupt instruction is executed

Absent : used for handling memory access violations

Undefined : Used for handling undefined instructions

Sysregs : a privileged mode that uses the same registers as user mode.

ARM register set :

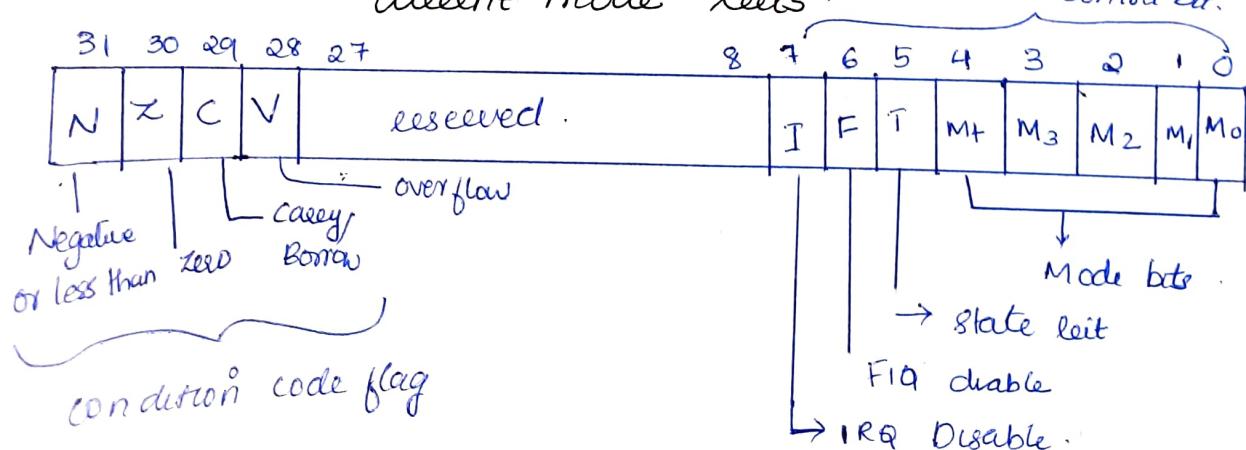
30 general purpose register + 1 program counter +

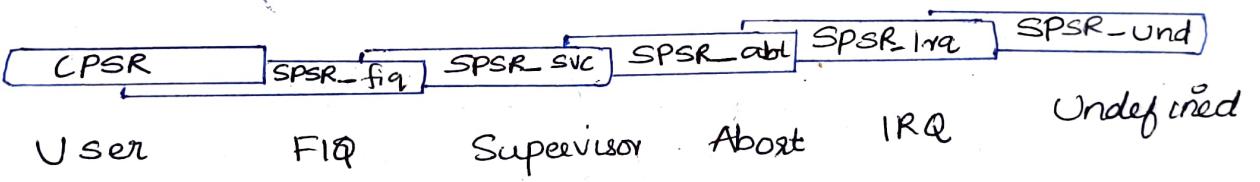
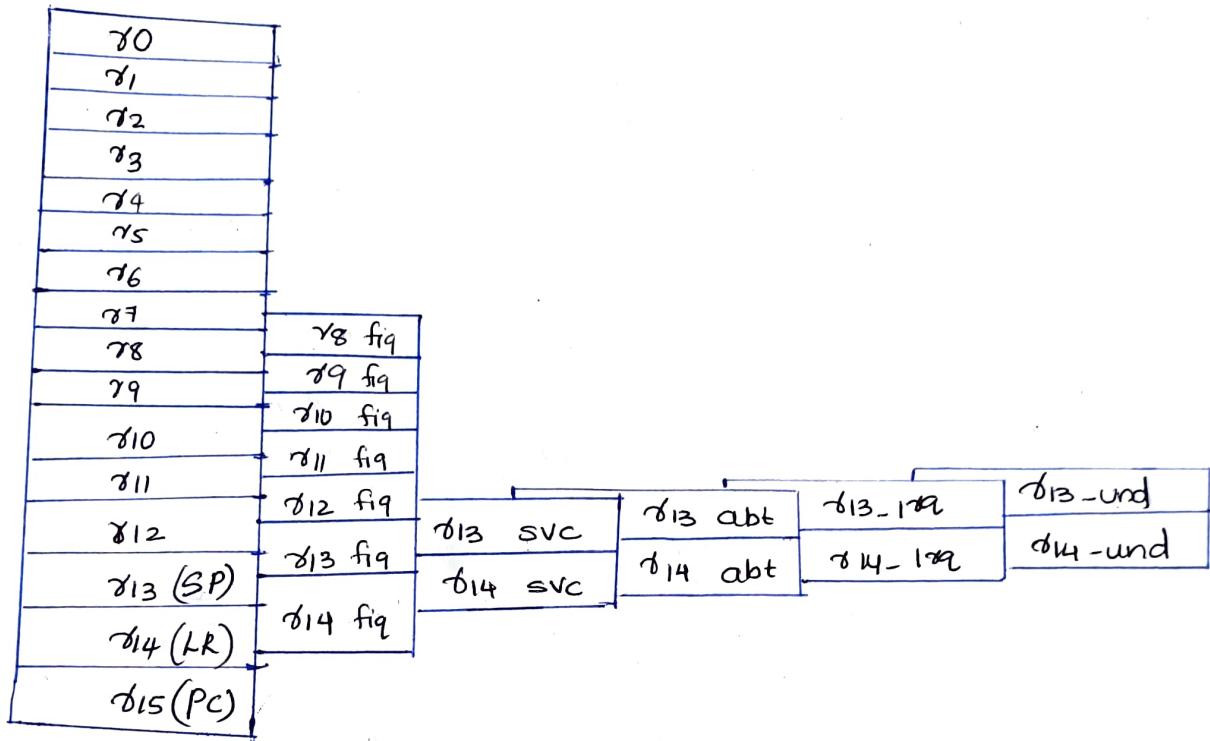
1 dedicated current program status register + 5 dedicated saved program status register

Entry in to thumb state can be achieved by executing a BX instruction and the state bit set to Operand bit. Transition to thumb state occur automatically on return from exception (IRQ, FIQ, etc).

R0 - R15 - except R15 are general purpose to store data/ address values. In addition 17 register to store status information

Register 16 - CPSR - This contains condition code flags and the current mode bits.





18. List and briefly explain the extension significance of an ARM processor.

SVE, SVE2; TME; All mandatory : Thumb-2, Neon, VFPv4-DIG, VFPv4

Obsolete : Jazelle.

Hardware extensions are standard components placed next to ARM core. It improve performance, manage resources & provide extra functionality and are designed to provide flexibility in handling particular applications.

Cache and TCM.

Memory management (MPU & MMU) - prevent apps from in appropriate access to hardware co-processor interface.

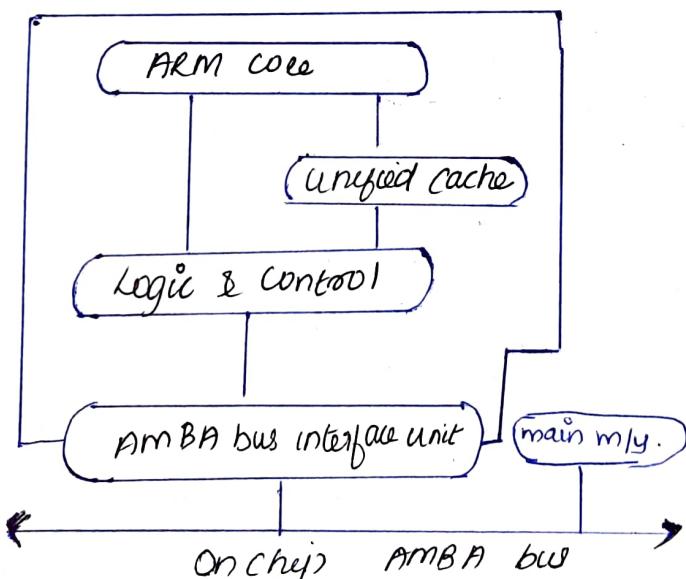
ARM core extensions with co-processors.

Co-processors can be attached to the ARM processor.

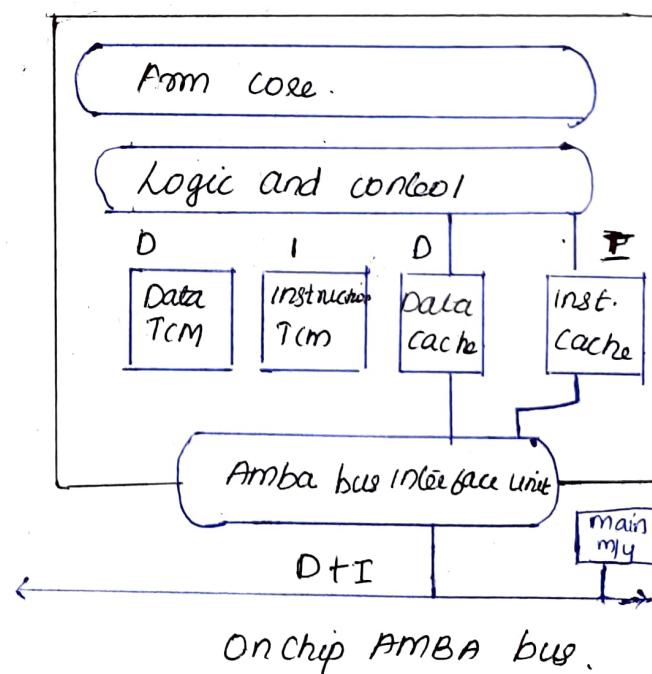
Extends the processing feature of a core by extending the instruction set or by providing configuration registers.

More than one co-processor can add to ARM core via co-processor interface. Co-processor can access through group of dedicated ARM Inst that provide load-store type interface.

Eg: Co-processor 15, the ARM processor uses Co-processor 15 register to control the cache, TCMs and memory management.  
(Highly unified memory).

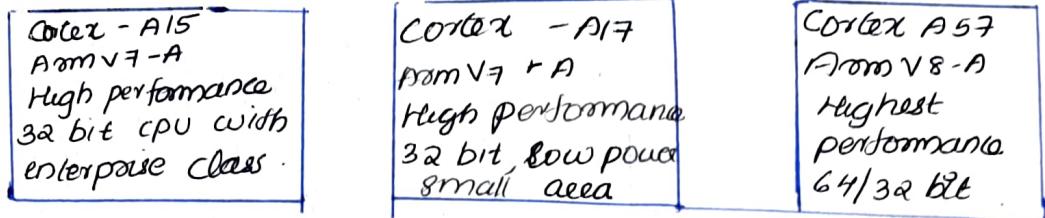


Simplified architecture with von Neumann style cache.



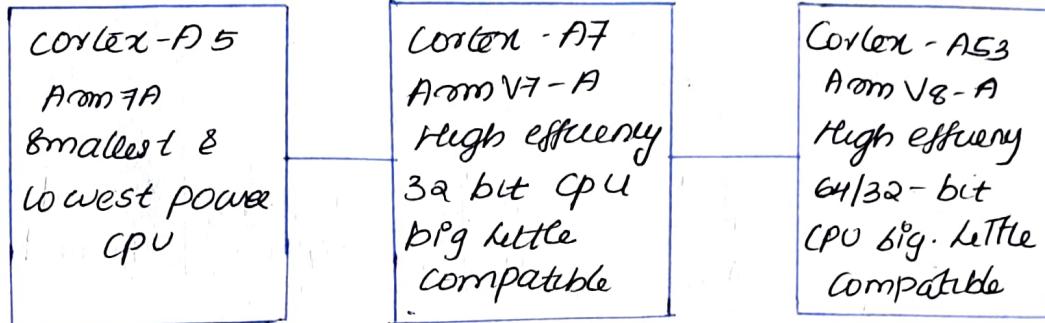
Simplified ARM arch with Harvard style.

17) With a neat diagram, explain the architecture road map of ARM?



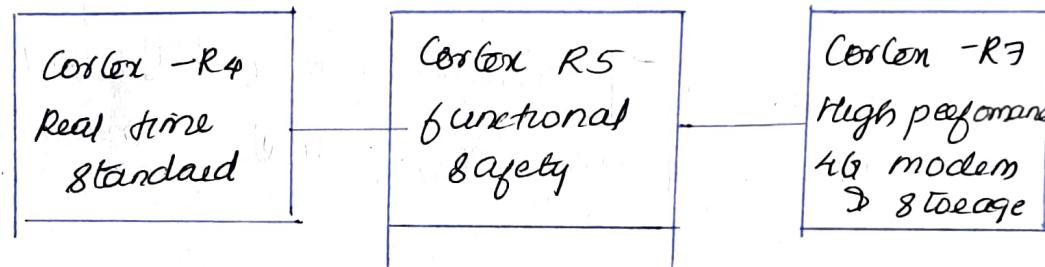
High performance

Cortex A9



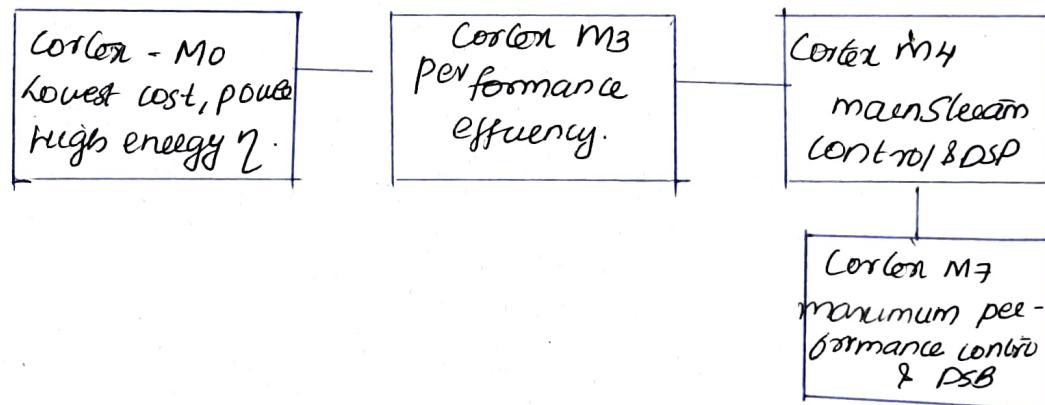
High efficiency

R



Real time

M



Control

Classic

Arm processor

↓

↓

↓

Arm II

↓

↓

ARM 0

↓

↓

ARM 7

Embedded

Cortex processor

↓

↓

Cortex Rx

↓

↓

Cortex M4

↓

↓

Cortex M3

↓

↓

Cortex M1

↓

↓

Cortex M0

Application

Cortex processor - Cortex A15

↓ - - - Cortex A9

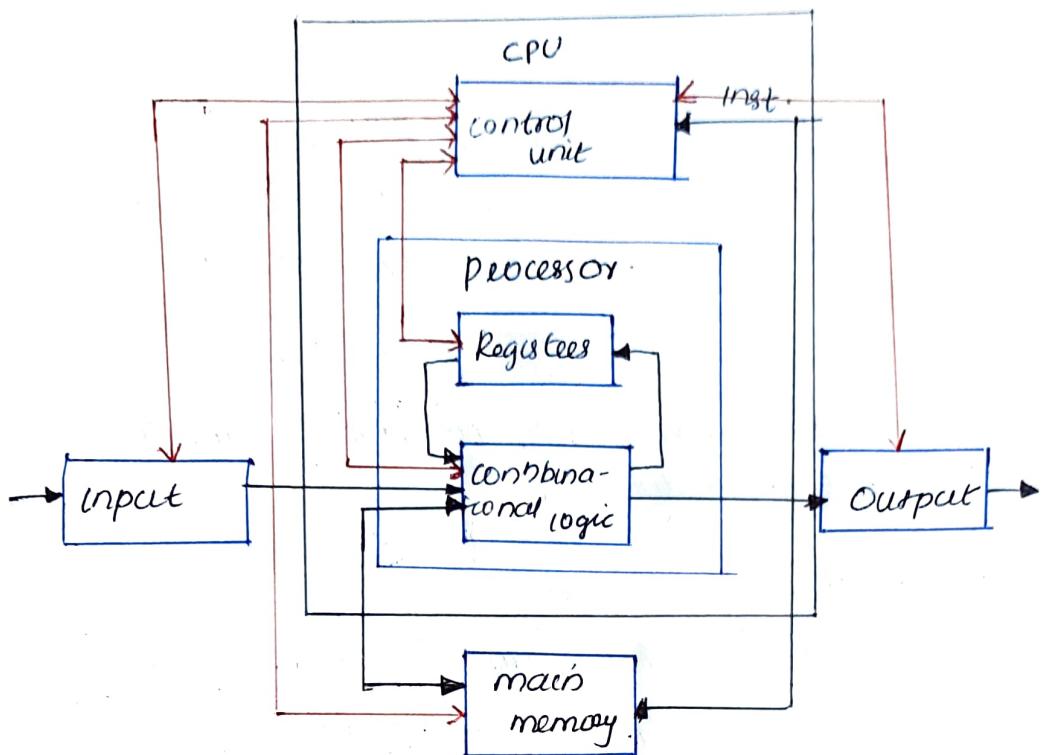
↓ - - - Cortex A8

↓ - - - Cortex A7

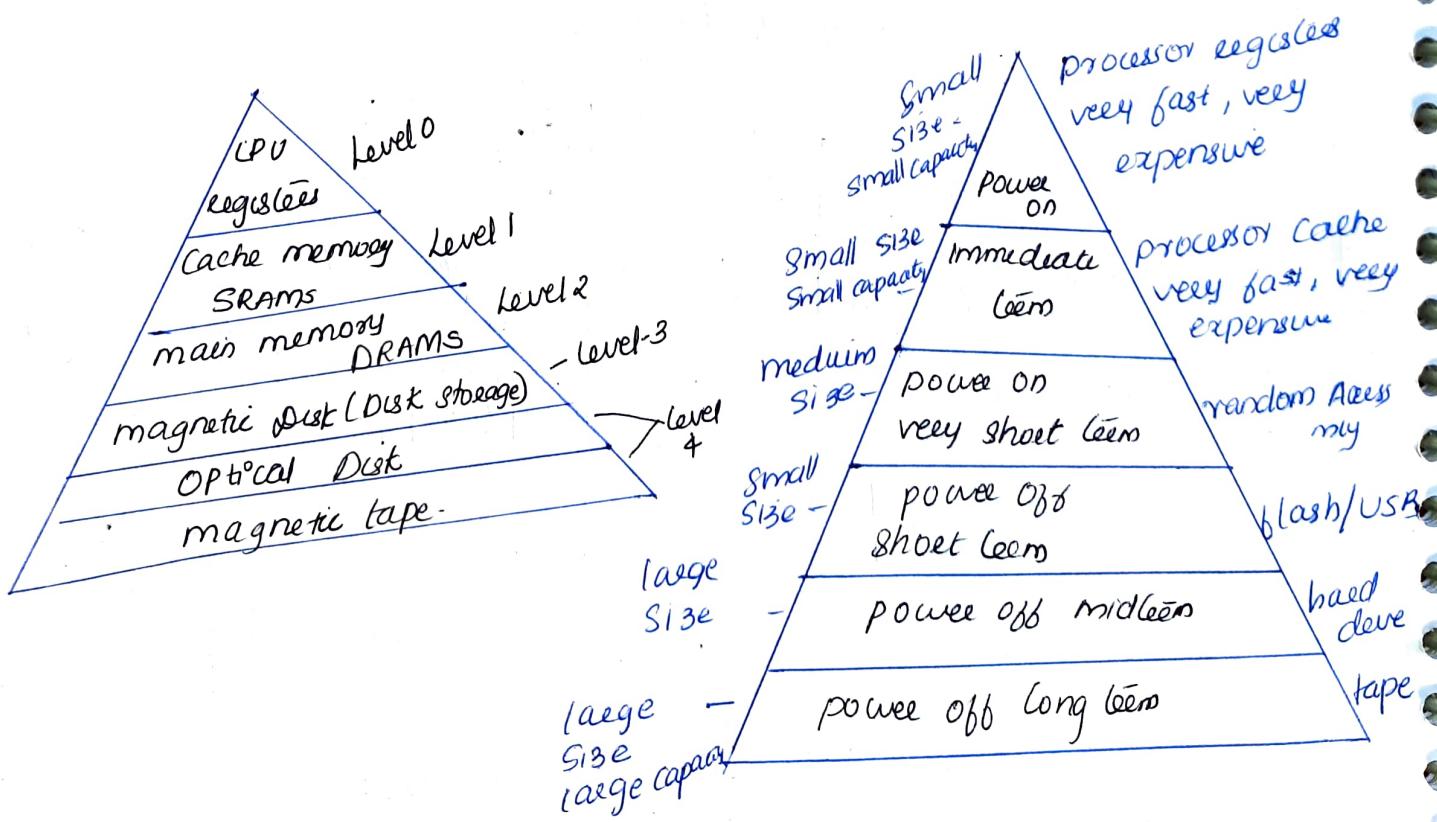
↓ - - - Cortex A5



18. From a birds eye view, draw the architecture of typical computer architecture.



20. List the hierarchy of memory with neat diagram.



## 20. continued :

**Registers:** Static RAM / SRAM used to store 64 or 128 bits

PC register is most important

Status word and accumulator are used to ~~to~~ decision making & arithmetic operation accordingly.

RISC - has no registers

**Cache memory:** It holds the chunk of data which frequently used from main memory. When processor has single core, then it will have 2 or more cache levels easily.

**Main memory:** memory in CPU which communicates directly with bus interface unit.

Fast and large memory used.

Made up of RAM as well as ROM.

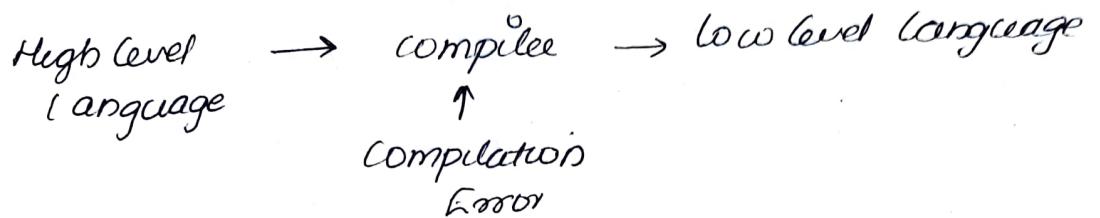
**Magnetic Disk:** circular plates fabricated of plastic other wise metal by magnetized material.

2 faces of disk are utilized as well as many disks may be stacked on one spindle by read or write heads obtainable on every plane. The tracks in the computer are nothing but bits which are stored within the magnetized plane in spots next to concentric circles.

**Magnetic tape:** normal magnetic recording which is designed with slender magnetizable coating on an extended, plastic film of the thin strip. Mainly used to backup huge data. Whenever computer requires to access strip, first it will mount to access data. The access time of tape memory is slower here.

Q1. How does a compiler store the data & the program how the user defined variables are stored by the compiler onto memory.

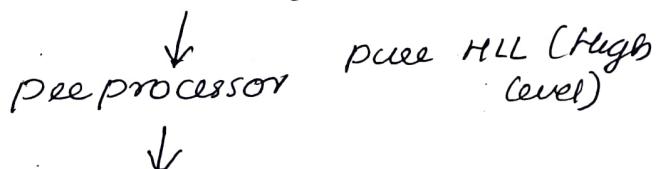
Compiler is software which convert high level language to low level language.



Cross compiler - run on machine A & produce code for other machine B.

Source to source compiler → transforms source code of 1 program

High Level Language.



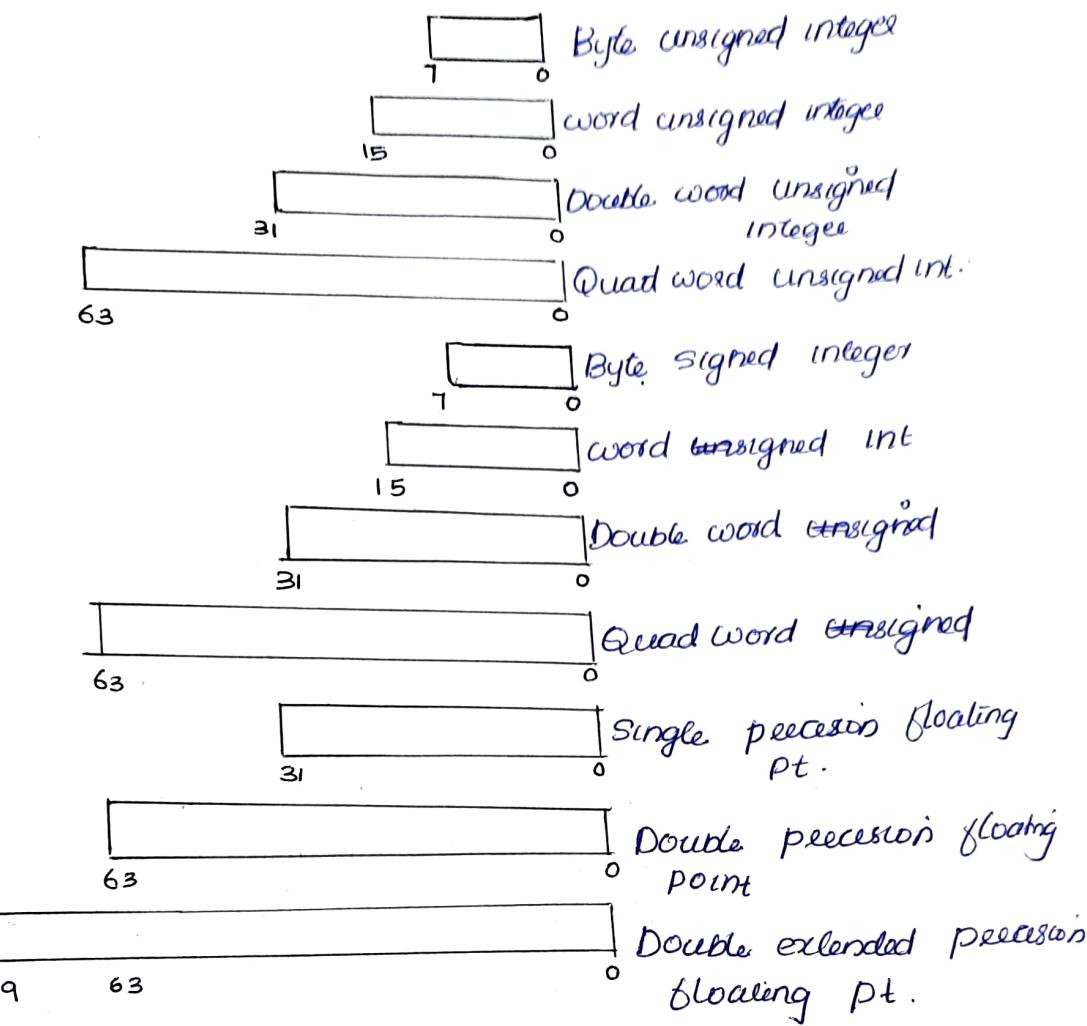
↓  
Compiler - intermediate stage that combines assembly language machine inst and some other data.

Output of - Assembly  
Assembly is Object

file. It translates to machine code. ↓ Relocate machine code.  
loader / linker - it converts relocate code into absolute code.

Absolute machine code.

22) With neat diagram, explain the alignment of data and supported data type of ARM?



Signed datatype can hold both positive and negative values, and are therefore cover in range.

Unsigned can hold large values, including 0, but cannot hold negative value (called ordinal).

Unpacked binary coded decimal: BCD in range 0-9

Packed BCD : 0-99.

Near pointer : 16 bit, 32, 64 bit effective address that represent offset within a segment used for pointer to unsegmented memory or a segment in segmented memory.

Far pointer : 16 bit segment selector and an offset of 16, 32, 64 bit

Bit field : position of each bit considered as an independent unit. It can contain up to 32 bit

Bit string: contains  $2^{32}-1$  bits which are continuous.

Byte string: bytes, words or double word contains  $2^{32}-1$  bytes.

Floating point: used by floating point-point unit and operated on by floating point instruction.