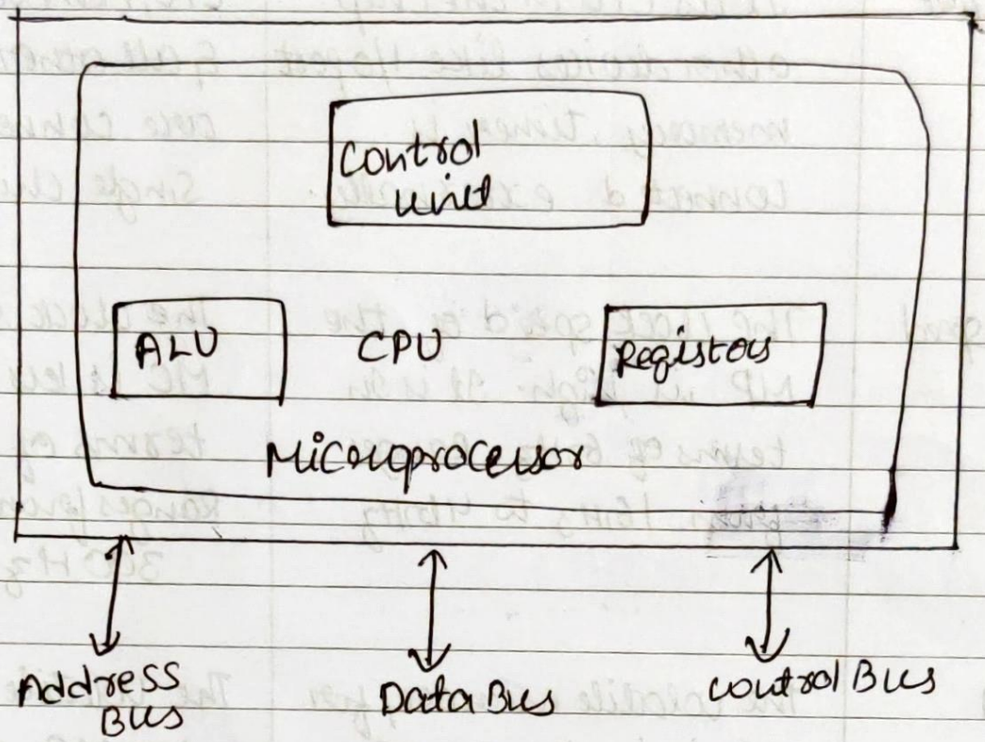
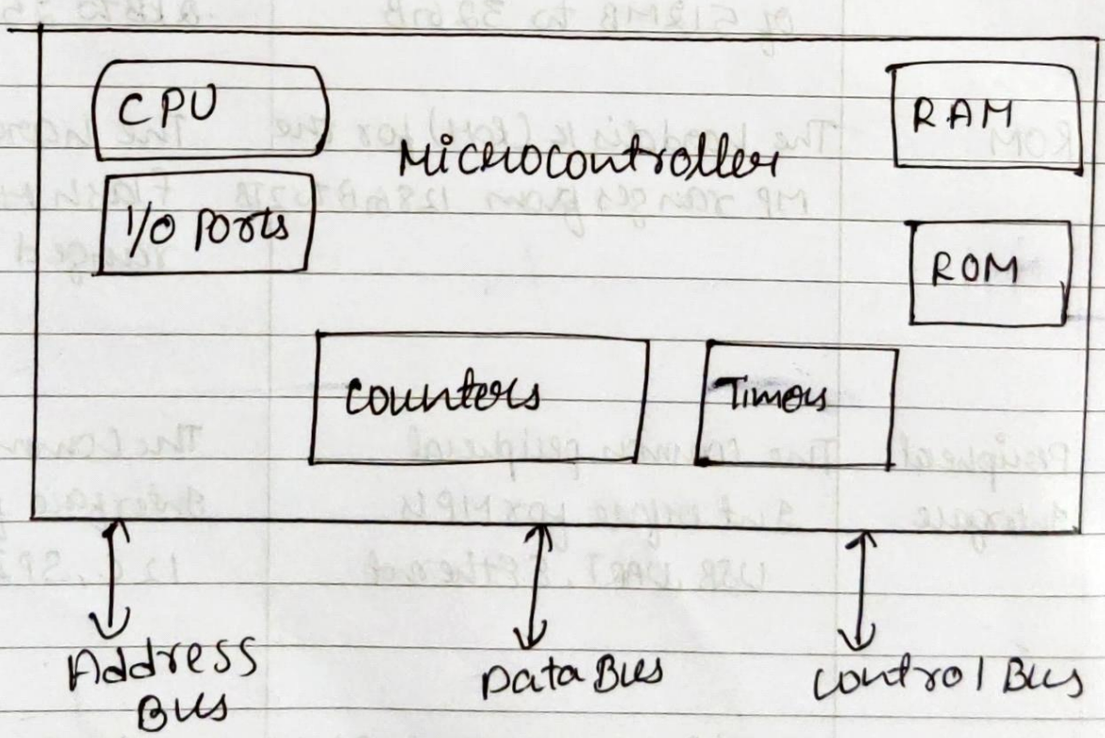
1. List and differentiate b/w Microprocessors & Microcontroller with neat diagrams.

parameters	Microprocessor	Microcontroller
Structure	It as CPU in the chip other devices like I/O port memory, timer is connected externally.	CPU, Memory, I/O port & all other devices are connected on single chip.
clock speed	The clock speed of the MP is high. It is in terms of GHz. Ranges from 1GHz to 4GHz	The clock speed of the MC is less, It is in terms of MHz. Ranges from MHz to 300 Hz.
RAM	The volatile memory for MP is in the range of 512MB to 32GB	The volatile memory for MC is from 2KB to 256KB.
ROM	The hard disk (ROM) for the MP ranges from 128GB to 2TB	The hard drive or flash memory is ranged of 32KB to 2MB.
Peripheral Interface	The common peripheral interface for MPs is USB, UART, & Ethernet	The common peripheral interface for MC is I2C, SPI & UART
Bit size	available from 32bit & 64bit	available from 8, 16 & 36bit
Cost	High	Cheaper

power consumption	High	less
Size	Loage	Small





2. List few Microprocessors & its features

1 Intel pentium :- The 1st chip in 1991 that come up with 60 & 66 MHz clock speed, used 3.1 million transistor, 4GB of RAM.

* These was 5th 1st MP that come with pipeling design based on x86 Micro architecture (32 bit MP).
* faster floating point unit.
* separate code & data caches.

2 Motorola 68000 :- Materials first 16 bit MP & support Pipeling.

* There are 17, 32 bit data registers.
* 32 bit address Registers.
* The data registers can be used to handles 8bit, 16bit 32 bit.
* There are 7 General purpose registers
* To implement OS & protection feature it operates on 2 mode 1. Superior mode 2) user mode.

3. RCA COSMAL CDP 1802 (NASA voyager 1) :-

* The RCA has static core CMOS design with no max. clock freq. that means you halt the CPU & restart from the same point where you paused.
* It has separate 8 pin buses, 8 pin bidirectional data bus & time Multiplexed address bus

4) The 1st MP was 'INTEL 4004' in 1971

* Mainly used in calculators, ATM's & other Simple Systems
* Maxm clockspeed was 740 KHz
* upto 92600 Instructions per second
* 12 bit address
* It had a 4bit address bus
* 8bit Instructions
* 4bit words.

3. List two popular microcontrollers & its features

a) STM32F103C8T6.

features

It is a popular no. of STM32F103xx, medium density performance line family of Microcontrollers that feature a high performance. ARM Cortex M3, 32 bit RISC core operating at 72MHz frequency & possess an extensive range of enhanced I/O & peripherals connected to two APB buses.

All members of the STM32F103x family, including the CT86, offer two bit ADC's, three general purpose 16bit timers plus one PWM timer, as well as standard & advanced communication Interfaces. upto 2 I2C, 2 SPI's & 3 USARTs & USB & a CAN.

b) ATMega 328.

1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timers with compare modes, internal & External Interrupts, serial programmable USART, a byte-oriented 2-wire Serial interface, SPI Serial port, 6-channel 10 bit A/D converter

c) PIC16F877A.

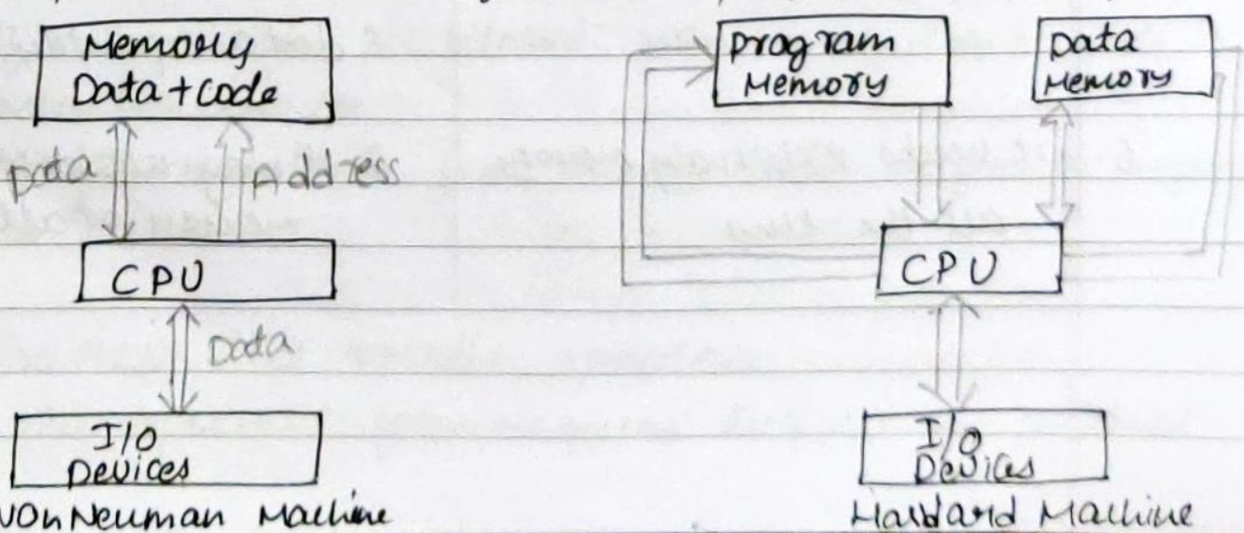
total number of Pins - 40
total number of ports - 5 (port A, B, C, D & E)
operating voltage - 2 to 5.5V
no of I/O pins - 33
no of ADC pins - 14
ADC Resolution - 10bit.
NO of comparators - 2
NO of timers - 3

communication protocols - UART, SPI, I2C
external oscillator - upto 20MHz.
prog Memory - 14KB
RAM - 368 Bytes
EEPROM - 256 bytes Max
Supports both ~~microprocessor~~ Hardware pin & timer Interrupts

4. which is fist Microprocesor, list the features
~~features of~~ The 1ST Microprocessor was Intel 4004
i) Maximum clock speed of 740KHz
ii) up to 92600 instructions per second
iii) Separate program & data storage
iv) 12-bit addresses
v) 8-bit instructions
vi) 4bit words.

5. which is first Microcontroller, list the features
The first Microcontroller was TMS 1802.
the features of TMS 1802.
i) It had 5 thousand transistors
ii) 3000 bits of program memory
iii) 128 bits of access memory
iv) possible to programitto perform a ~~vari~~ range of functions

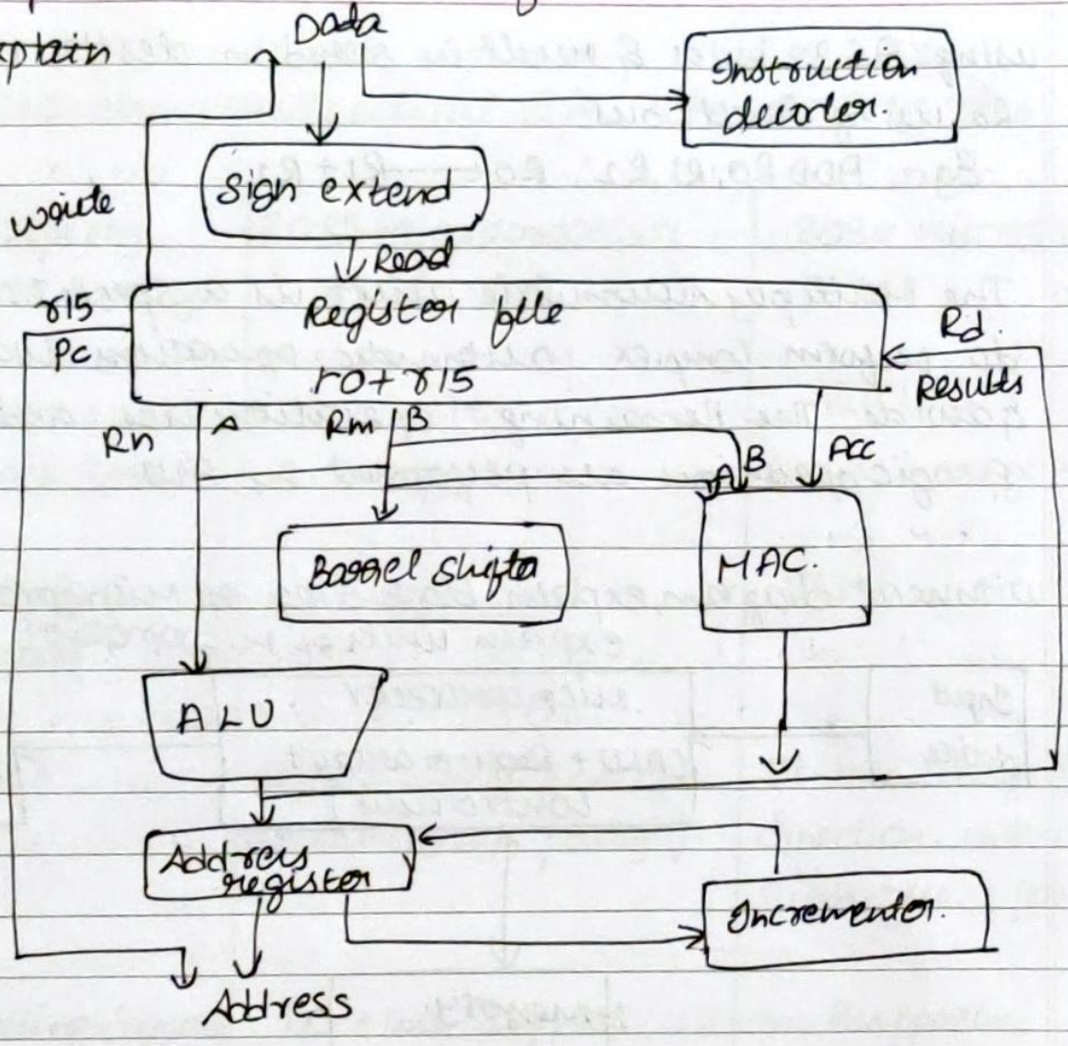
6 With the neat diagram, list d/f b/w Vonneuman & Harvard



	Von-Neuman Architecture	Harvard Architecture
1.	under pure Von Neuman architecture the CPU can be either reading an instruction or reading writing data from to the memory. Both cannot occur at the same time since the instruction & data use same s/m.	1. In a computer using Harvard architecture the CPU can both read an instruction & perform a data memory access at same time, even without a cache.
2.	Von-Neuman architecture is much slower as it has a single communication pathway	2. It is more faster for a given circuit complexity b/c instruction fetches & data access do not contend for single memory pathway.
3.	A Von Neuman architecture does not have distinct code & data address spaces.	3. It has distinct code & data address spaces
4.	It is not possible to have 2 separate memory s/m for Von Neuman architecture.	4. It is possible to have 2 separate memory s/m for Harvard architecture.
5.	It as single unified cache which stores both instruction & data	5. Harvad architecture usally have a multiple cache. which stores both instructions & data separately
6.	it needs externaly memory all the time	6. it may not external memory at all

7) Explain Architecture of ARM7

7). explain



1) The ARM7 processor has a 32 bit address bus. Hence it can access a total of 2^{32} = 4GB memory addrs space. as it based on Von Neuman Model this space contains both program & data.

2) The ARM7 processor has 32 bit data bus - This bus is used to fetch both instructions as well as data, both Instructions & data are 32 bits in size.

3) If a Instruction is fetched from memory than it goes to Instruction decoder.

4) If data is fetched from memory than it goes to register file (R0-R15)

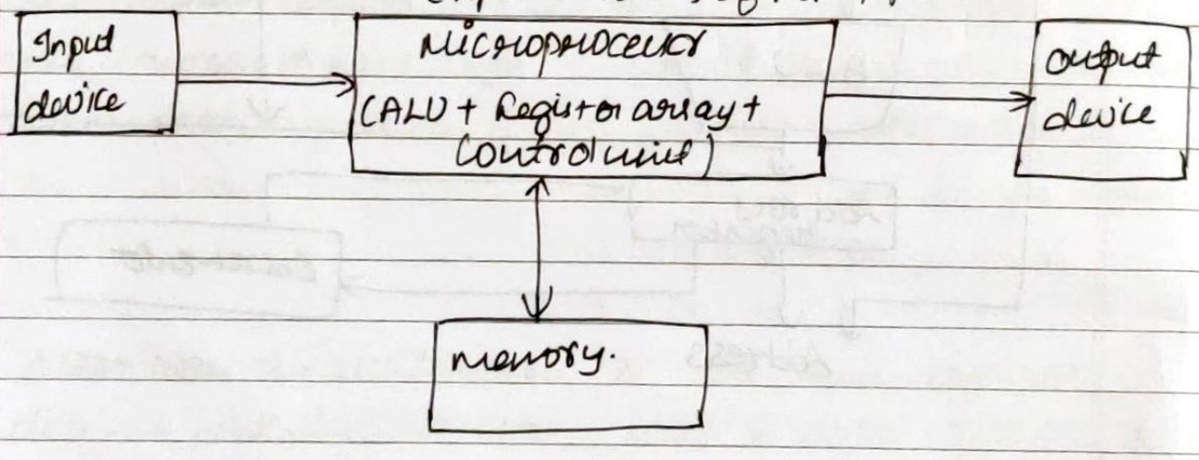
5) The ALU does triadic operations.
Two operands from registers Rm & Rn are obtained

using A&B buses & result is stored in destn register R0 using Result bus.

Eg - ADD R0, R1, R2; R0 ⟵ R1 + R2.

* The Multiply Accumulate unit - is a special unit of to perform complex arithmetic operations like Multiply & divide. The remaining operations like add, sub, & logic operations are performed by ALU

8) with neat diagrams, explain block diag of Microprocessor. explain units of Microprocessor



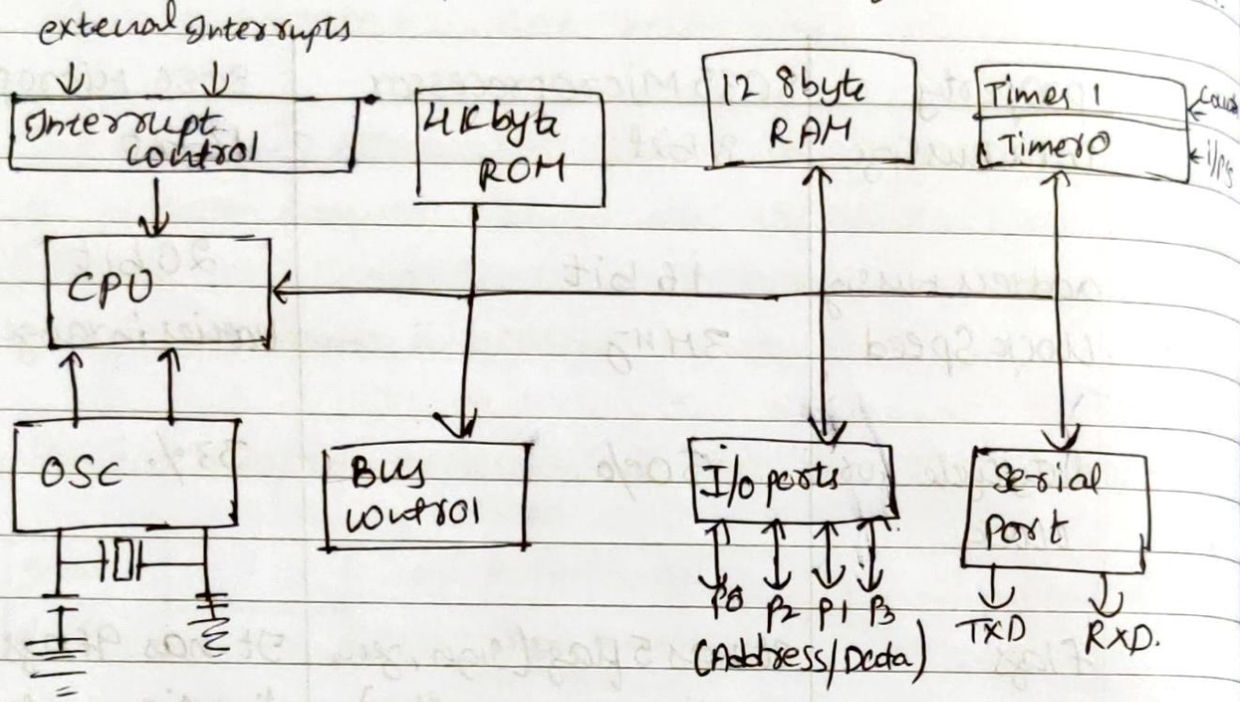
* Microprocessor is a controlling unit of microcomputer, fabricated on small chip cable of performing ALU operation & communicating with other devices connected to it.

* Microprocessor consist of ALU, register array, & a control unit, ALU performs arithmetical & logical operation on data received from memory or an I/p device.
* Micro processor follows a sequence: Fetch decode & execute.
* Initially, the Instructions are stored in memory in a Sequential order. The Microprocessor fetches the instructions from memory, & decodes it & executes the instructions till Stop instructions is reached. Later it sends result in binary to o/p port. B/w these process register stores temporarily data & ALU perform computing functions.

9. List the architectural d/f b/w 8085 vs 8086.

property	8085 Microprocessor	8086 Microprocessor
Data Bus size	8 bit	16 bit
address bus size	16 bit	20 bit
clock Speed	3MHz	Varies in range 5-8-10MHz
duty cycle for clock	50%	33%.
flags	It has 5 flags (sign, zero, auxiliary carry, parity).	It has 9 flags (overflow, direction, interrupt trap sign, zero, parity carry).
pipeling support	Does not support	Supports
Memory segmentation supports	Does not support.	Support.
No of transis-tors	Nearly 6500	Nearly 29000
processor type	Accumulator based	Genral pupose register based.
No. of processors	only one processor is used	More than one processor is used.
Memory size.	64 KB	1MB.
Instruction	No mul & div instruction	mul & div instruction are provid
Instruction Cycle & Support	Does not support	supports.

10. With a neat diagram, explain block diagram of a Microcontroller. Explain the units of Microcontroller.



Memory

a microcontroller needs prog ~~to us~~ memory to store prog/instructions to perform defined tasks. This memory is termed as ROM. Furthermore the Microcontroller also requires data memory to store the operands on temporary basis. This memory is known as RAM

address Bus

a bus of the Microcontroller can be defined as group of wire which can acts a medium for transfer data. there are 2 buses present in 8051 MC.

2) Address bus - used to address memory locations, it is 16-bit wide furthermore, the address bus can also be used to transfer data from CPU Hence for obvious reasons address bus is unidirectional

Interrupts - The most powerful attribute of 8051 Microcontroller is concept of Interrupts. The interrupt is a mechanism to -

* temporarily suspend the ongoing program,
* pass the control to a subroutine
* Restore the ongoing / main program.

Interrupts can be various types.
* INT0 → external Hardware Interrupt
* TF0 - Timer overflow Interrupt
* INT1 - External Hardware Interrupt
* TF1 - Timer 1 overflow Interrupt
* RI/TI - Serial communication Interrupt.

Input/output ports -
The 8051 Microcontroller needs to be connected to the peripheral devices in order to control their operations The I/o ports are responsible for the connection of the Microcontroller to its peripheral devices. There are 4 bit I/o ports present in Microcontroller.

11. What are addressing mode. Explain the different addressing modes in genral.
addressing mode - The term addressing mode refers to the way in which operand of instruction is specified. The addressing mode specifies a rule of interpreting or modifying the address field of instruction before operand is actually executed.

* Immediate addressing mode.
in this immediate Addressing mode, the data is provided in the instruction itself. The data is provided immediately after the opcode.

Eg MOV A, #0AFH;
MOV R3, #45H;
MOV DPTR, #FE00H.

Register addressing mode.

In the addressing mode the source or destn data should be present in register (R0 to R7).

Eg- MOV A, R5;
MOV R2, #45H;
MOV R0, A;

Direct addressing mode.

The source or destn address is specified by using 8bit data in instruction. Only the internal data memory can be used in this mode.

Eg- MOV A, R0;
MOV R2, 45H;
MOV R0, 05H;

Register Indirect addressing mode.

The source or destn address is given in register. By using register indirect addressing mode, the internal or external addresses can be accessed.

MOV @R0, 45H;
MOV @R1, 80H.

Indexed addressing mode.

The source memory can only be accessed from program memory only. The destn operand is always the register A.

MOV A, @A+DPTR,
MOV DPTR, #0180H

implied addressing mode

In this mode, there will be single operand. These type of instruction can work specific registers only.

RLA;
SWAPA;

12. Difference b/w RISC & CISC.

* in RISC, the instruction set size is small while in CISC the instruction set size is large.
* RISC uses fixed format (32 bits) & mostly register-based instructions whereas CISC uses variable formad ranges from 16-64 bits per instruction.
* RISC uses a single clock & limited addressing mode (i.e, 3-5). on the other hand, CISC use multi-clock 12 to 24 addressing modes
* The number of general-purpose registers that RISC uses ranges from 32-192, on contrary CISC uses 8-24 GPR's.
* RISC has split data & instruction cache design, as against CISC uses unified cache for data & instructions, although latest designs also uses split caches.
* Most of CPU control in RISC is hardwired without having a control memory, Conversly, CISC is microcode & uses control memory (ROM), but modern CISC also uses hardwired control.

13. What is code density ? what does ARM DOES to improve the code density ?

Code density - ~~computer diff~~ · the amount of space that an executable program takes up in memory. Code density is important in devices contain a limited amount of memory

The arm Instruction set is designed so that a program can achieve maxm performance with the min. no of instructions. most ARM9TDMI instructions are executed in a single cycle.

The Simpler thumb Instruction set offers much increased code density reducing code size & memory requirement.

* code can switch b/w ARM & Thumb instruction sets on any procedure call.

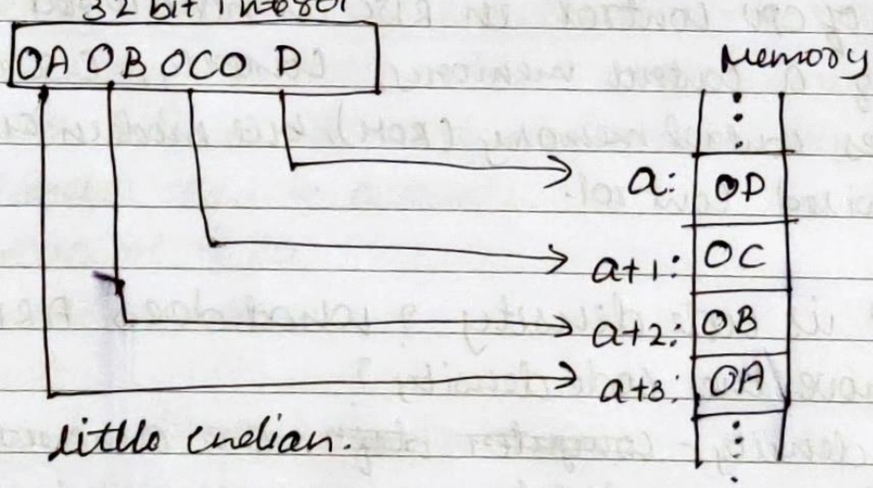
14. little Endian & Big Endian.

Endianness refers to order of storing & reading multibyte words in memory. Endianness determines if least significant byte of a word that we want to store in memory will go to the highest address of assigned memory space.

There are two possibilities of Endianness :- little Endian & Bi g Endian

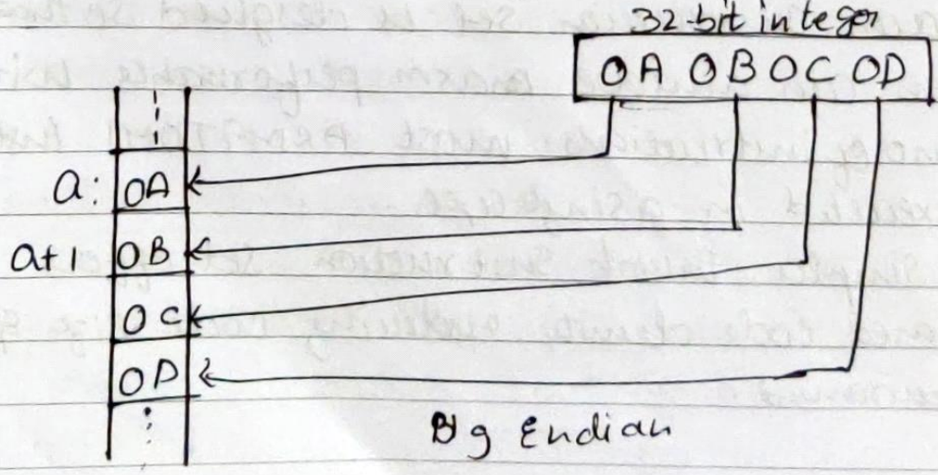
little endian

last byte will be stored first.



Big endian.

first byte of multibyte data stored first.



15. What are the d/f b/w assembly & High level language.

features	Assembly level language	High level language
1. Abstraction	Negligible abstraction with computer language	Strong abstraction with computer language.
2. Use of Interpreter	* it does not make use of compiler & interpreter	* it uses compiler as well as interpreter to convert instruction into Machine code
3. Flexibility	* low level language is difficult to use as it requires to elaborate technical details at each step.	* It is readible & Machine friendly language that can be easily interpreted & executed.
4. Execution.	* Faster execution of Programs	* Slow execution of Programs.
5. Modification	* Modification of prg is difficult	* Easy modification of prg written in high level language
6. Hardware.	* it is closely related to Hardware & hence used to write hardware programs	* it has no correspondance with hardware & used only to write software appln programs
7. Examples	* ARM, MIPS, Z80	* C, Fortan, Lisp, Prolog.

High level is prefered over assembly b/c, High level programmer does n't need to know details about Hardware like registers in processors as compared to assembly programmer. Code of assembly language is difficult to understand & debug than high level & high level prgms run independently of processor type.

16. What a neat diagram explain the programming model of ARM7

Arm7 supports VonNeumann Architecture.
* 3 Stage pipeling.

* Instructions are either 32 bits long (in Thumb State)

Data types.
Arm 7TDMI supports byte (8 bit), halfword (16 bit) & word (32 bit) data types.
* words must be aligned to 4 byte boundaries
* ARM7TDMI has total 37 Registers.
31 general purpose, 32 bit registers & 6 status registers but these cannot all be seen at once.

* The processor State & operating mode dictate which registers are available to the programmer.
* The ARM7TDMI contains a current PSR, plus five Saved program Status Registers (SPSR's) for use by exception handlers.

These registers.
* hold information about the most recently performed ALU operation
* control the enabling & disabling of interrupts
* set the processor operating mode.

Operating modes.
* ARM7TDMI supports seven modes of operation:
* User - The normal ARM program execution State
* FIQ - Designed to support a data transfer or channel process.
* IRQ (irq) - used for general purposes interrupt handling.

* Supervisor (SVC) - protected mode for the operating system.

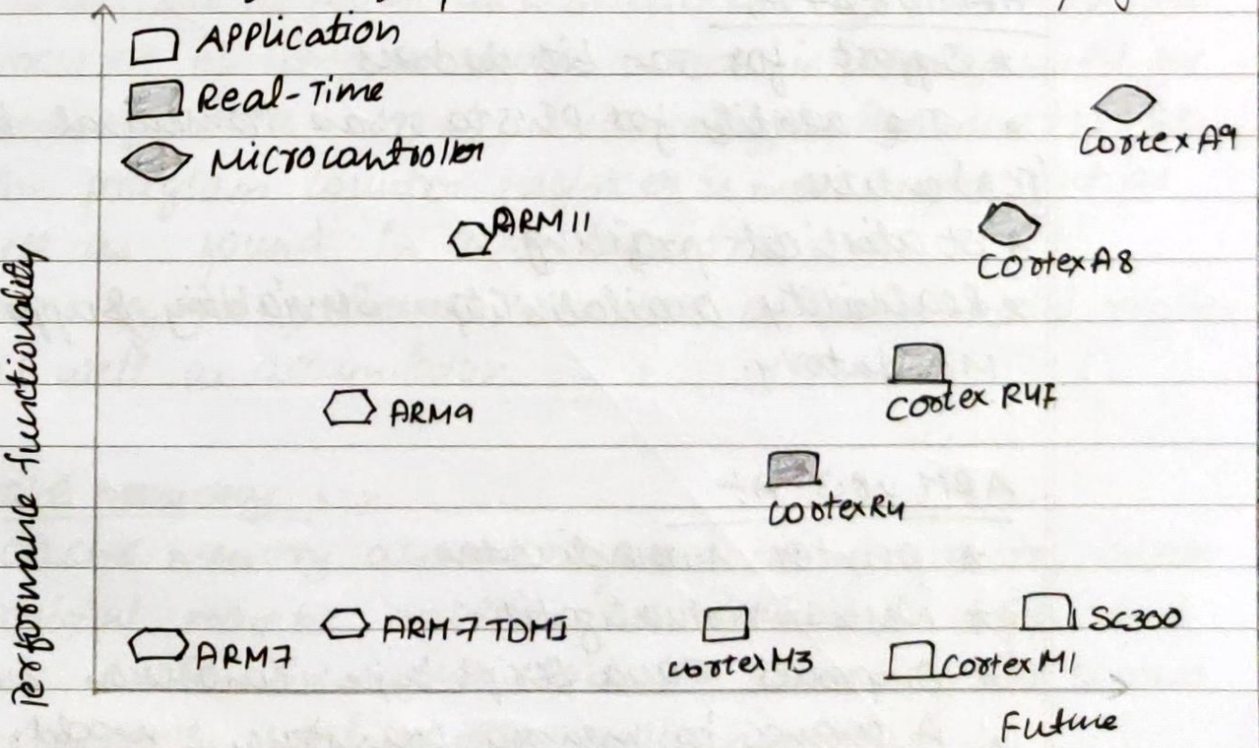
* Abort mode (abt) - Entered after a data or instruction prefetch abort.

* System (sys) - A privileged user mode for the operating system.

Exception priorities.

* Highest priority :-
1. Reset.
2. Data abort.
3. FIQ
4. IRQ
5. prefetch abort.
- Lowest priority
6. Undefined instruction, Software Interrupt.

17. With neat diagram explain architectural Roadmap of ARM.



Processor	Nog of pipelines	Memory organisation	Clock rate	Future M/PS/MHz
ARM6	3	Von Neuman	25MHz	
ARM7	3	Von Neuman	66MHz	0.9
ARM8	5	Vonneuman	72MHz	1.2
ARM11	8	Vonneuman/ Harvard	550MHz	1.2

18. List & briefly explain extension significance of ARM processor

development of major version of arm architecture can take many yeas. for eg ARM7 was released in 2007 & ARM V8 was released 6 yeas later in 2013. B/C the architecture needs to evolve B/W major versions. the extensions are added in B/W 3 major updates in architecture.

ARMV 8.1-A;

* atomic memory access Instructions
* Limited order regions.
* Increased Virtual Machine Identifiers Size & virtualisation Host extensions.
* privileged access N/W (PAN)

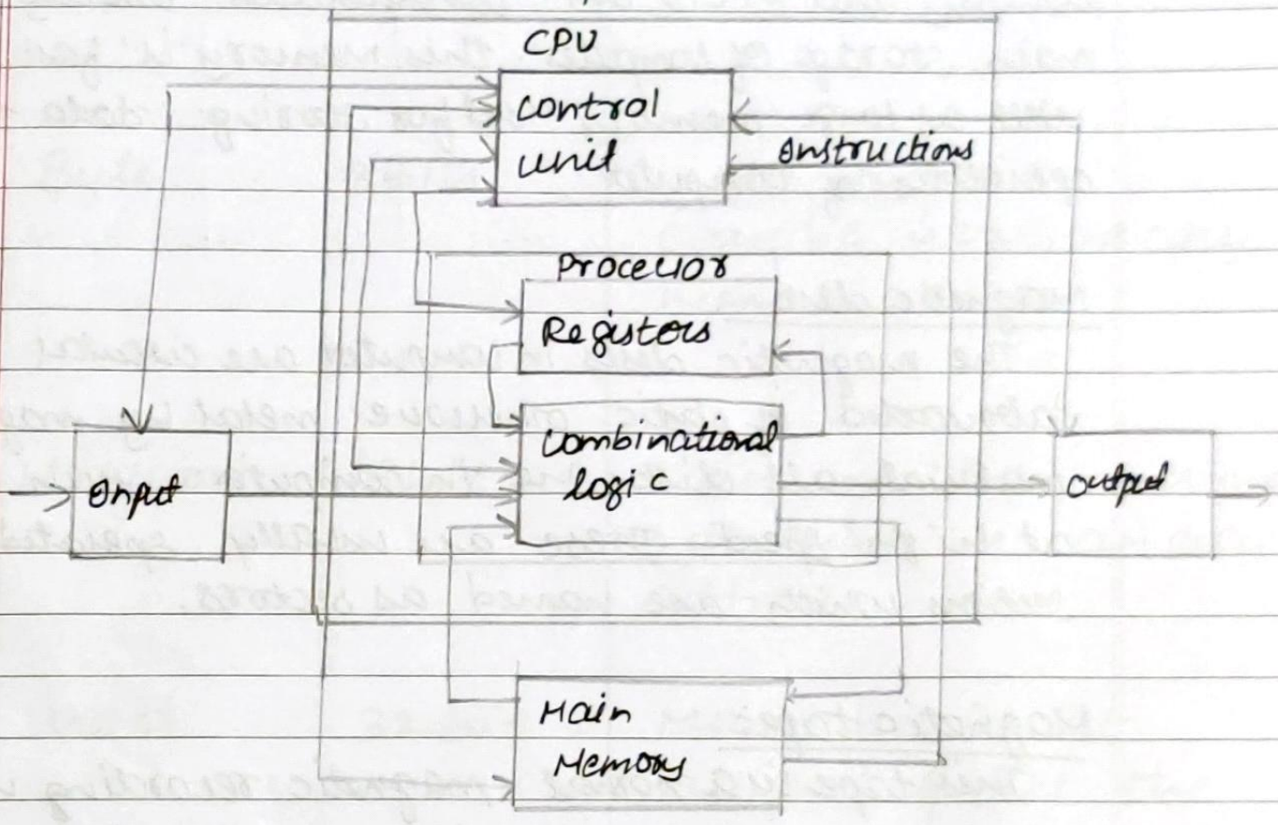
ARM v8.2-A;

* Support for 52-bit address
* The ability for PE's to share Translational Lock Buffer (TCB) entries.
* Statistical profiling
* Realibility availability Serviability & support becomes Mandatory.

ARM v8.3-A:-

* pointer Authentication.
* Nested virtualization.
* Improved Java Script type conversion support
* A change to memory consistency model.

19. From abirds eye view, draw architectural of typical computer architecture



20. List the hierarchy of Memory with a neat diagram

Registers:

- Usually, the register is a static RAM (or) SRAM in the processor of computer device which is usually used for holding data word which is typically 64 or 128 bits
- The program counter register is most important as well as found in all processors.
- Most of the processors are used as status word register as well as accumulator.

Cache memory:

Cache memory also be found in processor, however rarely it may be another IC which is separated into levels. The cache holds the chunk of data which are frequently used from main memory

main Memory:

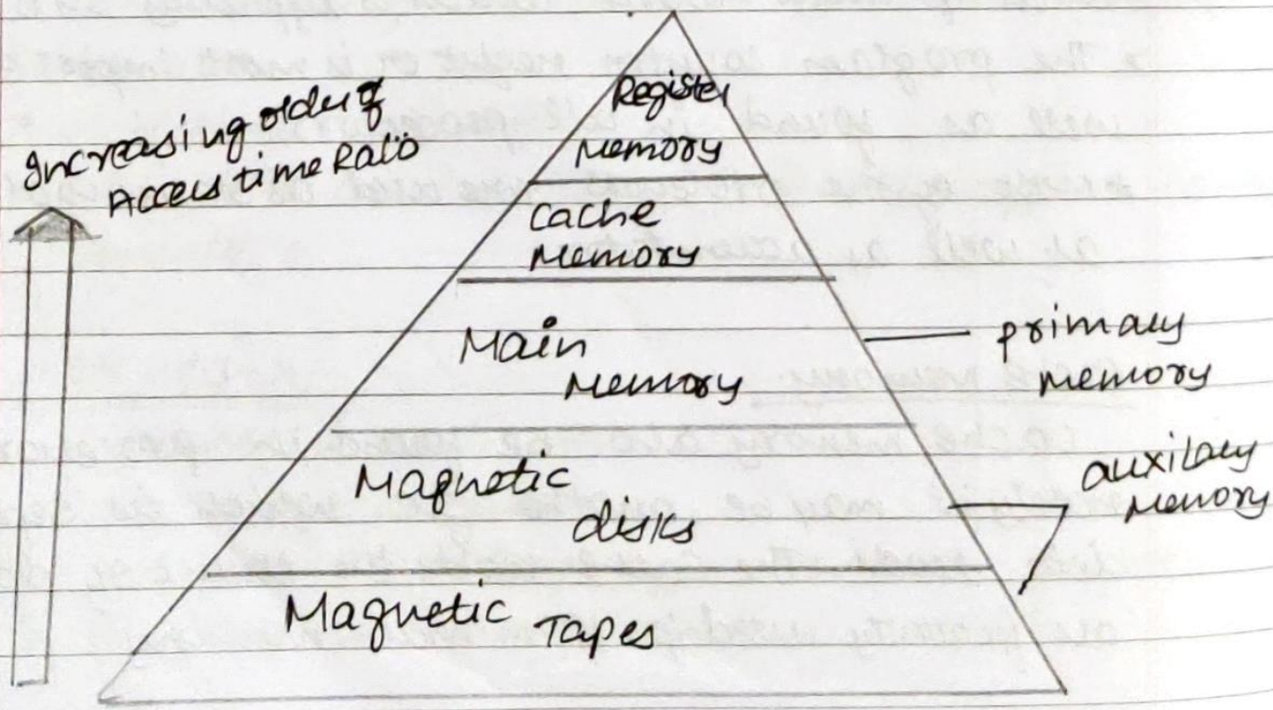
The main memory in computer is nothing but memory unit in CPU that communicates directly. it is main storage of computer. this memory is fast as well as large memory used for storing data through operations of computer.

Magnetic disk.

The magnetic disks in computer are circular plates fabricated of plastic otherwise metal by magnetized material. all disks are in computer turn jointly at high speed. These are usually speiated into sections which are named as sectors.

Magnetic tape:-

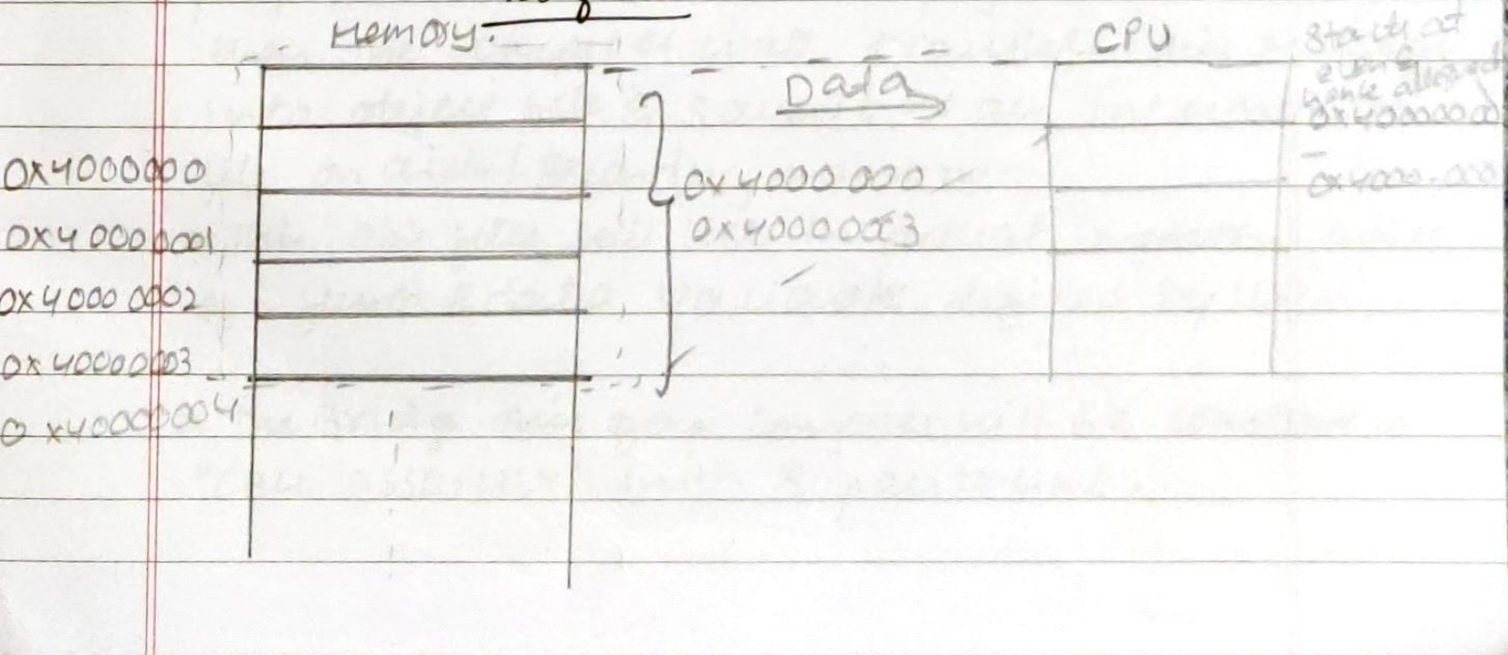
This tape is a normal magnetic recording which is designed with a slender magnetizable covering on an extend plastic film of thin strip.
Once data is allowed, then it will be unmounted.
* The access time of memory will be slower within magnetic strip as well as it will take a few minutes for accessing a strip.



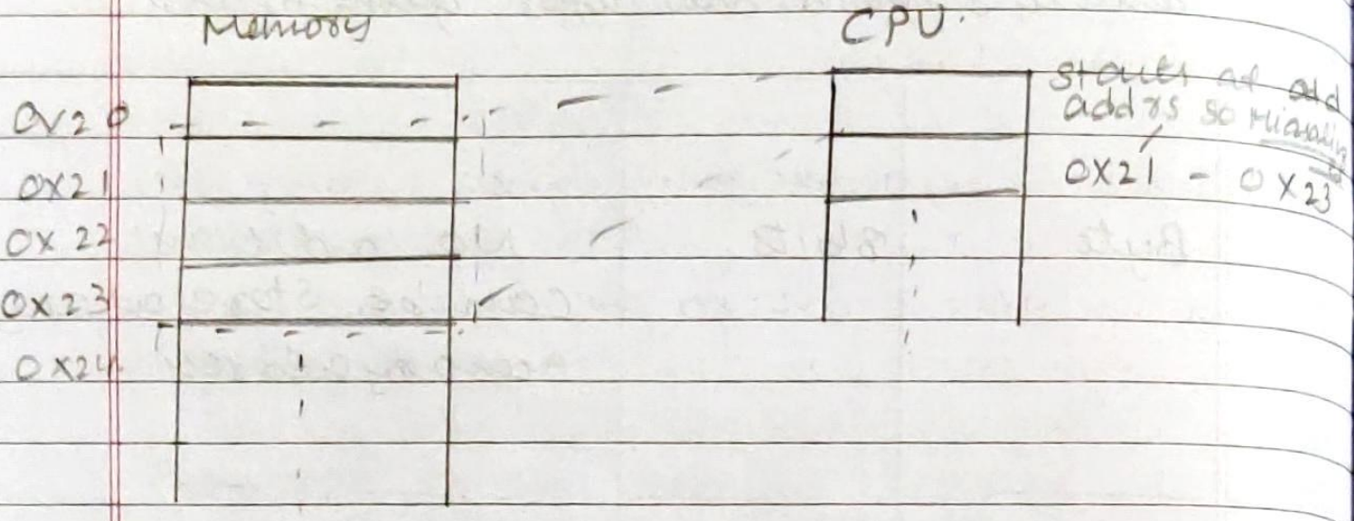
22. With a neat diagram, explain alignment of data & supported data types from ARM7.

Byte	8bits	No conditions can be store at any memory address
Halfword	16bit	Must be 16-bit assigned & it must be an even address
Word	32bit	Must be 32 bit assigned & all the address should be multiple of 4.

Alligned.

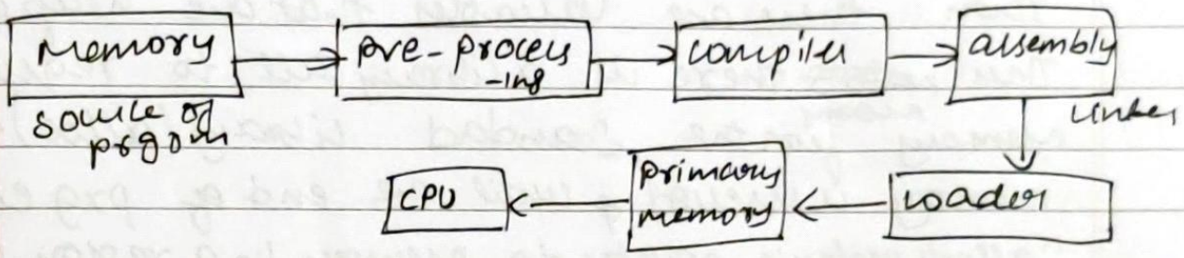


Misaligned.


Memory
CPU.
0X20
0X21
0X22
0X23
0X24
starts at odd address so misaligned
0X21 - 0X23

21. How does a compiler store data & program. How user defined variables stored by compiler onto memory.

* When the programmer creates the program & stores in the disk.
* this program should be compiled first before it can be executed & this ~~operation~~ compilation has d/f stages



* The compiler creates the object file
* The linker/assembler will link object code with the libraries & create an executable file.
* Then the loader will put this file onto the primary memory/ RAM so the CPU will execute the ~~diagr~~ prog / instruction.

The compiler acts as a translator which converts the High level languge prg into Machine code.
* The compiler will automedically invokes the preprocess & will translate prg into Macro file then the compiler will translate this Macro file into object file & saves it as an intermidiate file on disk / secondary memory.
* This obj files still lack in actual memory addrs of func & data variable defined by user.

* the Bridge this gap compiler will be constand a "CALL ASSEMBLY" instn & pass to link.

Typically local variable / user defined variables are put on 'stack'. This means that compiler assigns an offset to the stack pointer which can be d/f depending on the invitation ~~process~~ progress The compiler assumes that the memory location like Stack pointer +4 or Stack pointer +8 etc.. can be accesible & usable by prog. Hence this is mapped into assembly instruction.

Then there are variables that are heap allocated. This ~~means~~ means there is a library call to request. memory from the standad library (alloc). The memory is reserved ~~@~~ until the end of prg execution "alloc" return pointer to memory in a region of memory called "heap"