

① Difference between

② Microprocessor



Microcontrollers

- It is the heart of computer system
- It is only a processor so memory and I/O components need to be connected externally.
- Memory and I/O has to be connected externally so the circuit becomes large.
- Cost is high
- we cannot use it in compact System.
- power consumption is high
- It uses External Bus to Interface.
- It is complex and used in personal computer.
- It has processor with memory & I/O components.
- Memory and I/O are already connected / present , internal circuit is small.
- cost is low
- we can use it in Compact System .
- power consumption is low
- It uses Internal Bus
- It is simple and used in washing machines, MP3 players & Embedded System.

③ RISC



CISC

- Reduced Instruction.
- Executes within single Clock cycle.
- Complex Instruction
- In more clock cycles.

- High clock speed.
- It has large general purpose register set
- Separate Load - Store Instruction Register
- Instructions encoded in parallel by pipeline
- Low clock Speed
- It has few dedicated registers.
- No Load Store Registers.
- No such pipeline.

Discuss different types of registers in user Mode.

* ARM processor has 15 active Registers : 16 data registers and 1 processor status register.

* All registers are 32 bits in size which operates in 32 bits.

ie R13 - Sp - stack pointer

R14 - LR - Link register

R15 - PC - Program Counter.

CPSR - Current program status register.

SPSR - Saved program status register.

* R13 - SP - holds head of the stack

* R14 - LR - holds return address, when it calls subroutine.

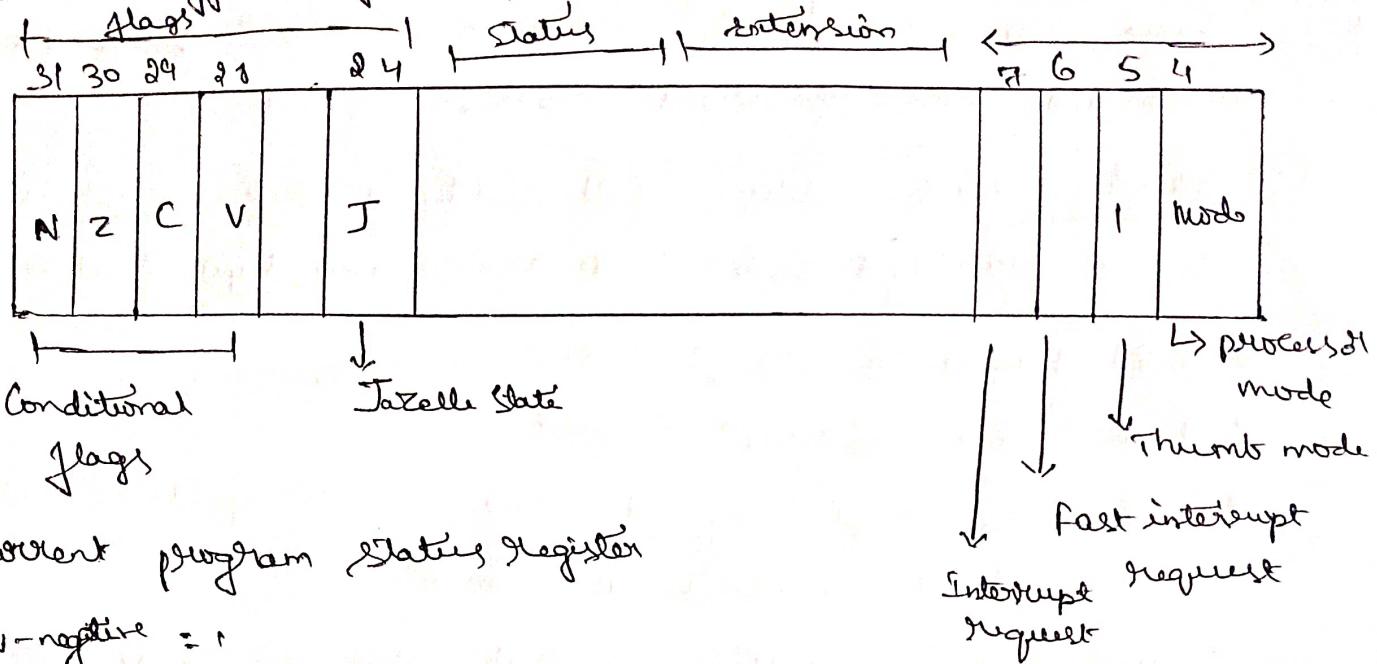
* R15 - PC - holds the address of the next instruction to be executed.

R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15

CPSR

Registers in user mode.

③ Explain different fields of CPSR?



N - negative = 1

Z - zero = 1

C - carry = 1

V - overflow = 1

24	5	Mode
1	0	Thumb
0	1	ARM
0	0	ARM

* CPSR is a 32 bit register used to monitor and control internal operations

* CPSR is divided into 4 fields of 8 bit width

① flag ② status ③ Extension ④ control

* control field :- 0 to 4 bits decide mode of operations

* 5th bit = 1 → Thumb mode

* 6th bit = 1 → fast interrupt request

* 7th bit = 1 → Interrupt request

* flag field :-

24th - 1 ⇒ Thumb mode

28th - V ⇒ overflow

29th - C ⇒ carry

30th - Z ⇒ zero

31th - N ⇒ negative

(4) what are processor modes? Explain.

There are mainly 2 modes in processor mode privileged mode and non-privileged mode.

(a) privileged mode: allows full read and write access to CPSR

In privileged mode there are many types such that

- * Abort: failed to attempt access memory.
- * FIR: Fast interrupt Request
- * IR: Interrupt Request which corresponds to interrupt level available on arm processor.
- * Supervisor: It is the mode that processor is in after boot
- * System mode: It is a ~~supervisor~~ revision of user mode.
It allows full read and write access to CPSR.
- * Undefined: It is used when processor encouters undefined instruction.

(b) Non-privileged mode: for read access - control field for read and write access - condition flags.

(5) what are interrupts. Explain ISR and IRET

* It is an event or instruction or external signal which transfers control from main program to sub program.

(a) Sub program is called as Interrupt Service Routine (ISR)
In ISR it executes the instructions, at the end of ISR there is a instruction called IRET, after executing IRET control is again transferred to main program.

(b) Interrupt vector table (IVT):

* for every interrupt there is a ISR. These ISR address

are stored in a table called interrupt vector Table.

* The 30bit address is reserved for vector table is

0x000000000

* In some processor higher address 0xFFFFF0000

vector Table

Exception / Interrupt	Short hand	Address	High address
Reset	RESET	0x000000000	0xfffff 0000
undefined Instruction	UNDEF	0x000000104	0x1ffff 0004
Software Interrupt	SWI	0x00000008	0x1ffff 0008
prefetch abort	PABT	0x0000000C	0x1ffff 000C
Data' abort	DABT	0x000000040	0x1ffff 0010
Reserved	---	0x00000014	0x1ffff 0014
Interrupt Request	IR	0x00000018	0x1ffff 0018
Fast Interrupt Request	FIR	0x0000001C	0x1ffff 001C

- Reset vector: when power is applied the processor executes the instruction from reset vector.
- undefined instruction vector: when the processor is unable to decode an instruction this vector is used.
- Software Interrupt Request: when we execute SWI instruction this instruction is used/called.
- prefetch abort vector: This interrupt occurs when the processor attempts to fetch an instruction without correct access permission.
- Data abort: This interrupt occurs when the processor attempts to fetch data memory without correct access permission.

Interrupt request vector: - This interrupt is used by external hardware, it occurs only when TR bit in CPSR goes high i.e IRQ = 1.

* fast interrupt Request: This interrupt occurs only when FIR bit in CPSR goes high i.e FIR = 1. It is similar to IRQ but it is reserved for faster response times.

⑥ Discuss different types of interrupts in arm processor.

> There are 2 types of interrupts available in the arm processor. The first type is the interrupt caused by external event from hardware peripherals and the second type is the SWI instruction.

* (IRQ) - Interrupt's request vector: Is used by external hardware to interrupt the normal execution flow of the processor.

* (FIR) - fast interrupt request: Is similar to the interrupt request but is reserved for hardware requiring fast response times.

* Reset vector: It is the location of the first instruction Executed by the processor when power is applied. This instruction branches to the initialization code.

* undefined instruction vector: It is used when the processor can't decode an instruction.

* Software interrupt vector: Is called when you execute a SWI instruction. The SWI instruction is frequently used as the mechanism to invoke an operating system routine.

* Page fault about vector?

occurs when the processor attempts to fetch an instruction from an address without the correct access permission.

* Data fault about vector?

This interrupt occurs when the processor attempts to fetch data memory without correct access permissions.

④

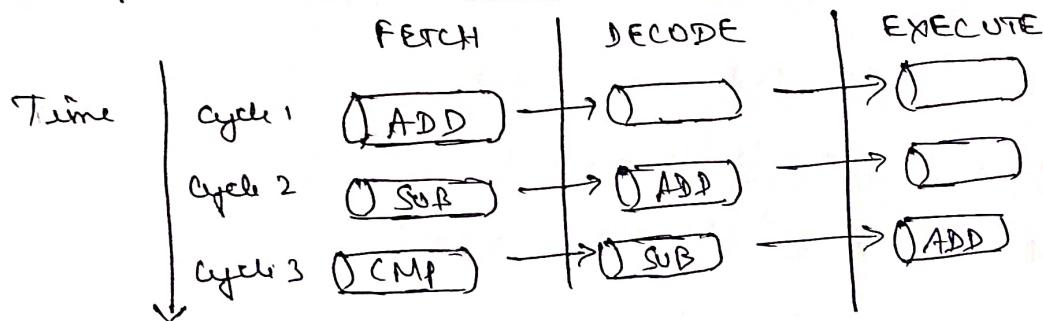
Explain pipelining with example.

A pipelining is the mechanism a RISC processor uses to execute instructions. Using a pipeline speeds up executions by fetching the next instruction while other instructions are being decoded and executed.

→ ARM -7 , 3 stage pipeline (Fetch) → (decode) → (execute)

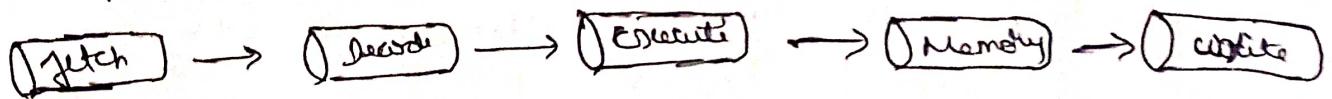
- fetch loads an instruction to be executed.
- decode identifies the instruction to be executed
- execute processes the instruction and writes the result, back to a register.

Pipeline instruction sequence:-



Here 3 instructions are placed in pipeline sequence, In first cycle ADD instruction is fetched. In II cycle ADD is decoded, SUB instruction is fetched. In III cycle CMP is fetched SUB is decoded ADD is executed.

ARM-9 - five stage pipeline



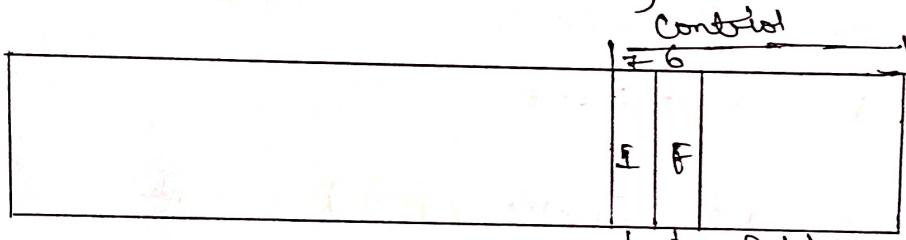
ARM-9 increases 2 stages by adding memory and write back stage.

ARM-10 - 6 stage pipeline.



ARM-10 increases the pipe length by adding sixth stage issue.

⑧ what is masking and unmasking in ARM?



Interrupt masks are used to stop specific interrupt requests from interrupting the processor.

CPSR has 2 interrupt mask bits 7 and 6 (I and F) out of which I controls interrupt request (IRQ), F → FIQ

if the bit is set to 1 it is activated or unmasked set to 0 it is deactivated or masked.

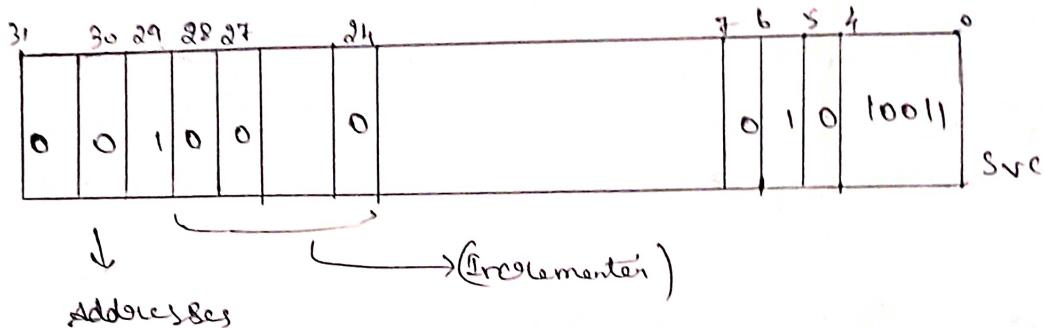
7th bit = 1 if Q is unmasked

6th bit = 1 FIQ is also unmasked.

⑨ what are conditional flags?

conditional flags are updated by comparisons and the result of ALU operations that specify the S instruction suffix
Q - saturation, V - overflow, C - carry, Z - zero, N - negative

- Saturation
 - The result causes saturation (overflow or underflow)
 - Wrap
 - The result causes a signed overflow
 - Carry
 - The result causes an unsigned overflow
 - Zero
 - The result zero, frequently used to indicate equality
 - Negative
 - Bit 31 of the result is binary.



$C=1$ Mode = supercritical $CPSF = n^2 \int_{\text{var}} f(t) - \bar{f}^2 dt$

$j \rightarrow$ Jargelle mod. ≈ 0 proses

$t \rightarrow$ Thumb mode

IRR - ARM mode of instruction is 32 bit and supervisor mode of operation.

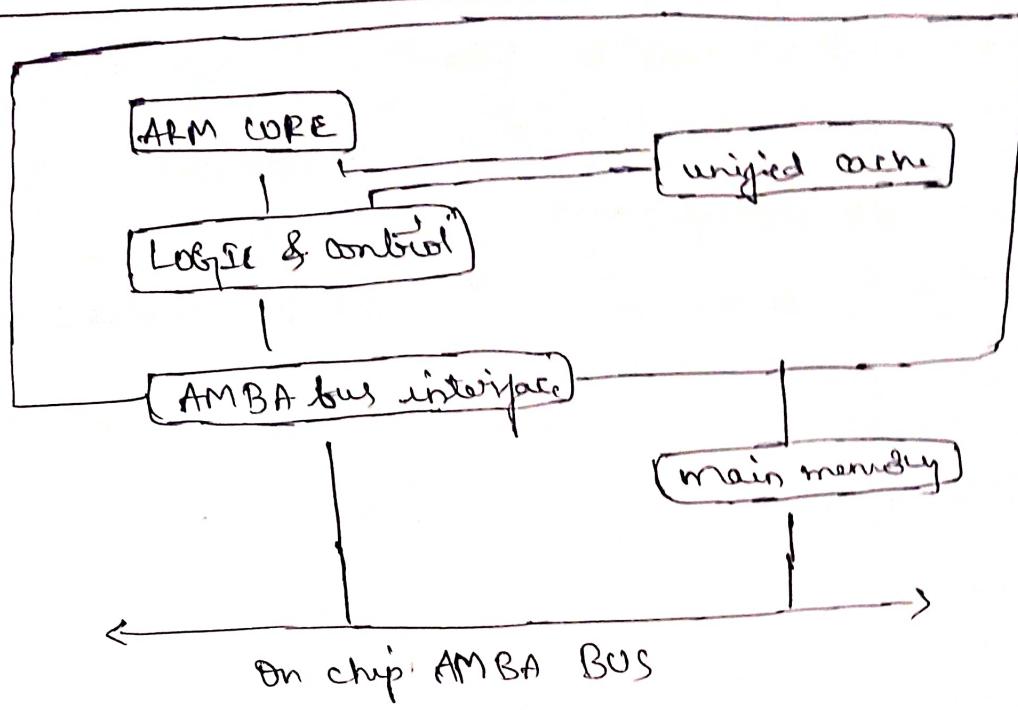
⑩ Discuss different types of cache.

→ ARM has 2 forms of cache

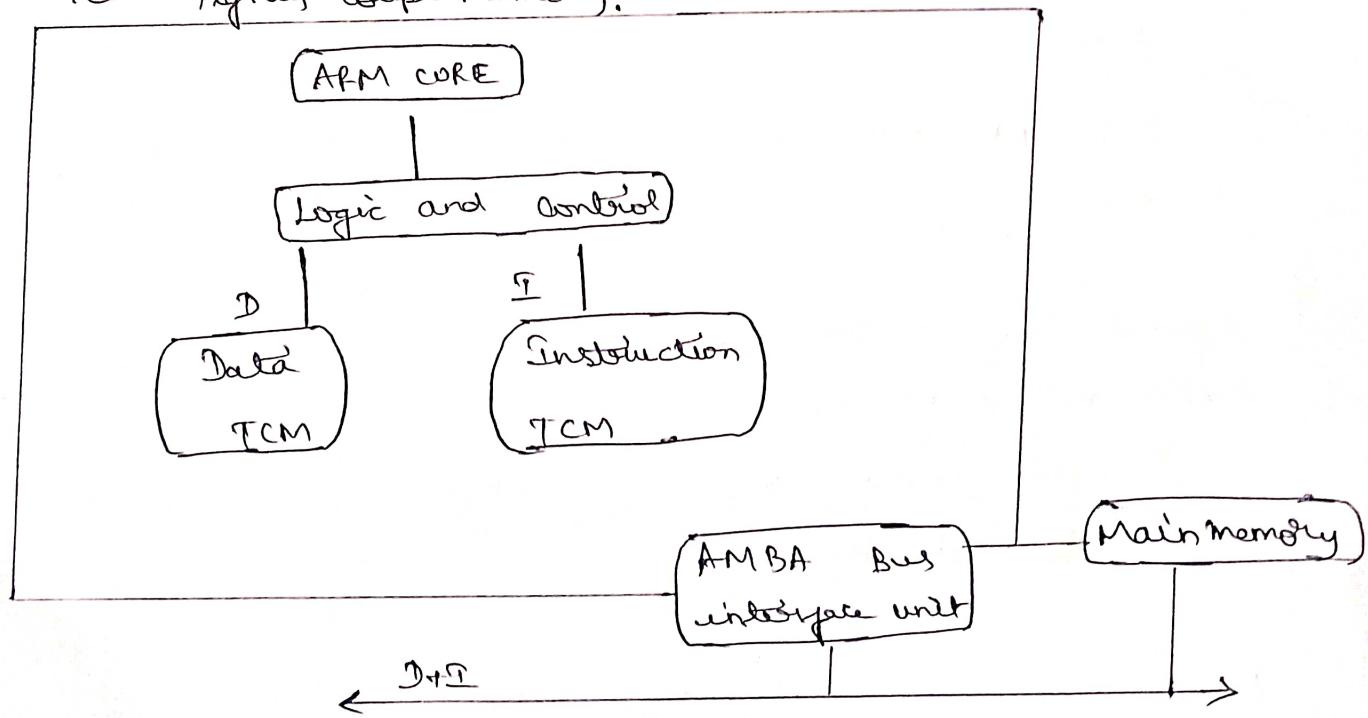
- 1). von neumann architecture style
 - 2). Harvard Style.

1). Von - neumann architecture style!

data and instruction into single multiplexed code glue logic is used to connect memory to AMBA Bus.



④. Harvard style: It is achieved using fast SRAM.
 It has separate caches for data and instruction.
 Cache provides overall increase in performance,
 TCM - Tighty coupled memory.



⑪ What is memory management? Explain.

It is usually necessary to have a method to help organize these devices and protect the system from applications trying to make inappropriate access to hardware.

ARM CORE have 3 different types of memory management hardware

① Non-protected memory: is fixed and provides very little flexibility. It is normally used for small, simple embedded systems that require no protection from rogue applications.

② MPUs: Memory protection unit, employ a simple system that uses a limited numbers of memory regions. These regions are controlled with a set of special coprocessor registers.

③ MMUs: Memory management unit, are the most comprehensive memory management hardware available on the ARM.

⑫ Explain ARM core Data flow model.

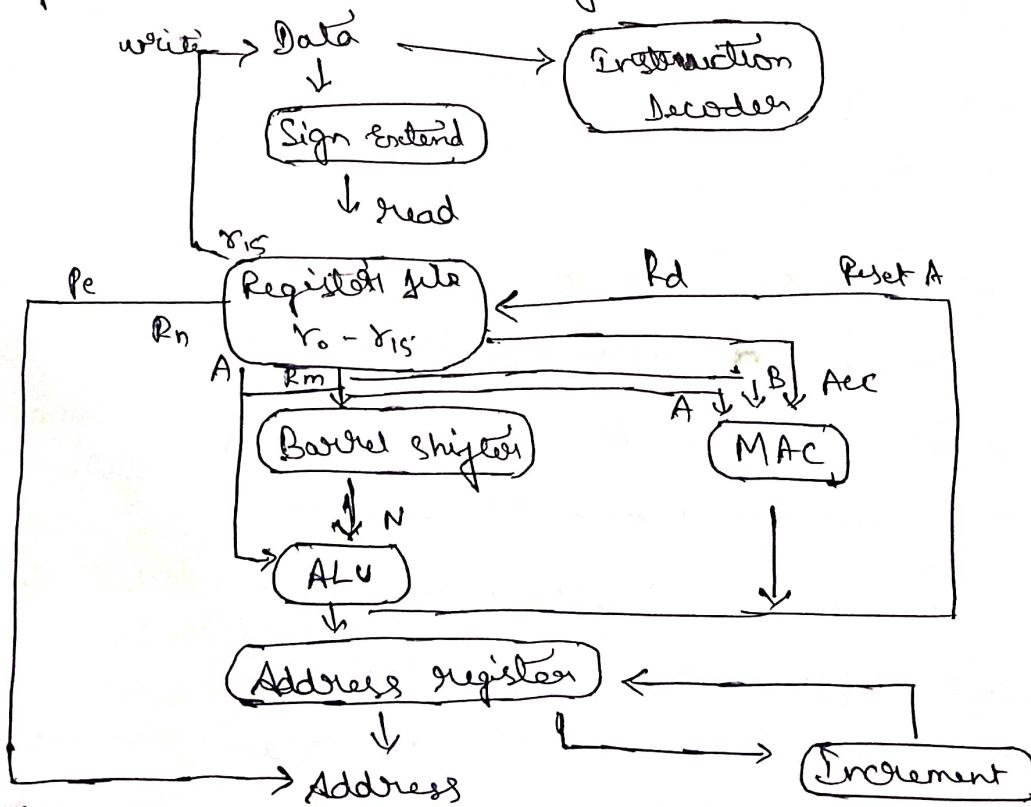


Figure shows Von-Neumann architecture ARM uses data flow model. Here data and instruction share same bus. ARM processor uses load-store architecture.

Load :- Instruction will copy data from memory to register.

Store :- Copy data from Register to memory.

Register file :- It is a store bank of 32 bit registers R0-R15.

Sign-Extend :- hardware converts a signed 8 bit and 16 bit numbers to 32 bit values.

Instruction Decoder :- It translates instructions before they executed. ARM has source registers Rn and Rm from register file using internal buses A and B.

* Data processing instruction write the result in destination register Rd to the register file.

* Load and store instructions use ALU to generate an address.

* The content of register Rm is pre-processed in barrel shifter before it enters ALU.

* The increments update the address register before the core reads or writes the next register value.

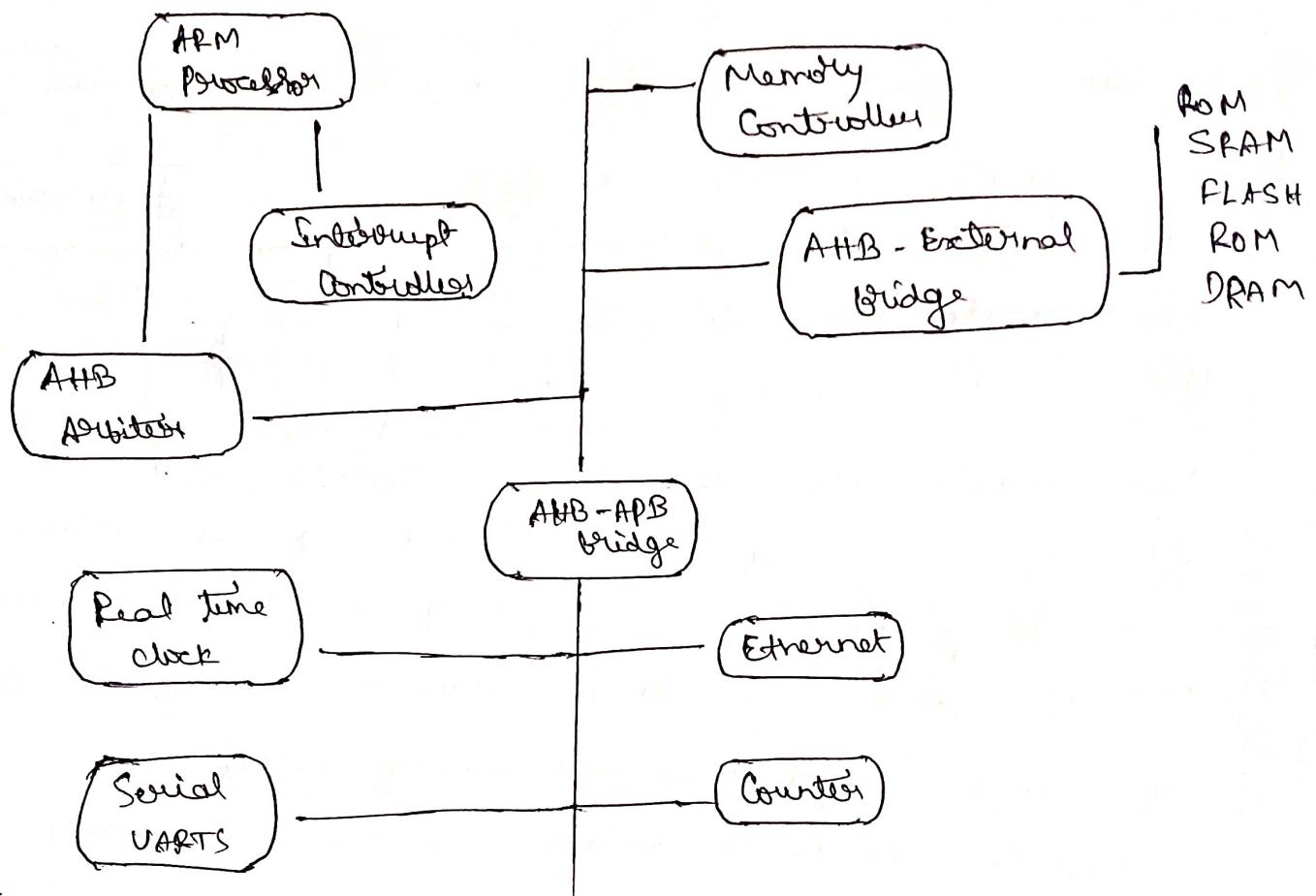
(13) Explain ARM based Embedded System.

> There are 4 main hardware Components,

- ARM processor controls the embedded device. It consists of ARM for memory management and cache.

- controllers coordinate important functional block of the system 2 types of controllers (1) interrupt
(2) Memory controllers.

- peripherals includes input - output devices from embedded device.
- Bus used to communicate b/w different parts.



Q14. Discuss different types of ARM Bus.

> Embedded Systems use different bus technologies. The most common type is PCI Bus, This Bus connects processor with video cards and Hard disk controllers.

These are 2 different classes of device attach to the bus. The ARM processor core is a bus master and peripheral are called as Bus slaves.

AMBA Bus:

The advanced microcontrollers bus architecture (AMBA)

AMBA Bus introduced 3 types of bus.

① ARM System Bus (ASB).

① ARM peripheral Bus (APB)

② ARM High performance bus (AHB)

AHB provides higher data throughput than ASB.

15

Explain Software Abstraction layers on hardware.

Each software component in the stack uses a higher level of abstraction to separate the code from the hardware device.

The initialization code is the first code executed on the board and is specific to target.

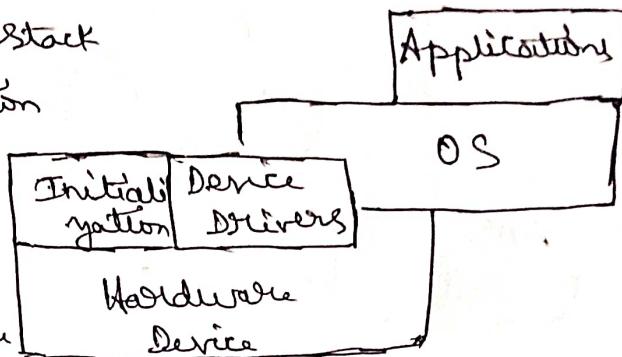
It sets up the minimum parts of board before handing control over to O.S. The O.S provides an infrastructure to control application and manage hardware system resources. The device drivers provide a consistent software interface to the peripherals on hardware device. finally an application performs one of the tasks required for device.

The software components can run from ROM or RAM. ROM code that is fixed on the device is called firmware.

* Initialization code: It takes the processor from reset state to state where the O.S runs. It is operand in 3 phases.

(1) Initialization Hardware (2) Diagnostics (3) booting

operating System (OS) :- The initialization process prepares the hardware for an O.S to take control. O.S organizes the system resources (peripherals, memory and processing).



time. 2 types of OS

- (i) Real time OS (RTOS)
- (ii) Platform OS.

Applications: ARM processor are found in numerous market segments; networking, automobile, imaging and consumeris device.