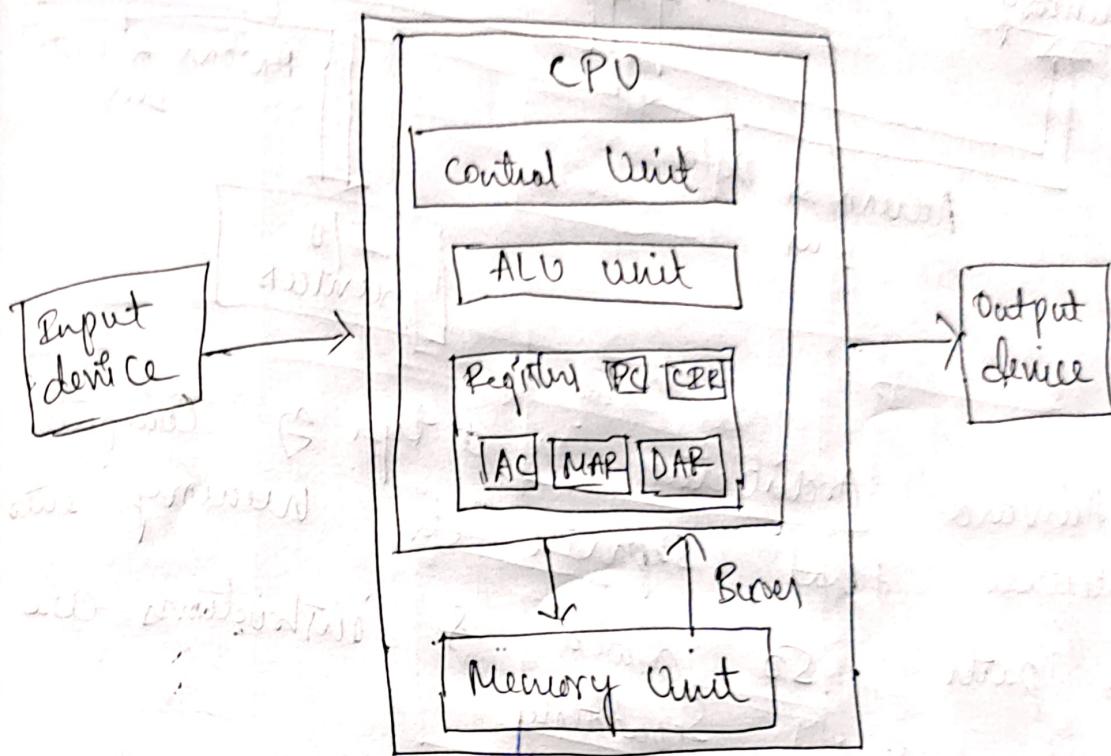


① With Block diagram and Harvard Architecture explain Von - neuman

* Von - neuman Architecture



* Von - neuman Architecture design consists of Arithmetic Logic Unit (ALU), Control Unit, memory Unit, registers, I/O ports and Bus.

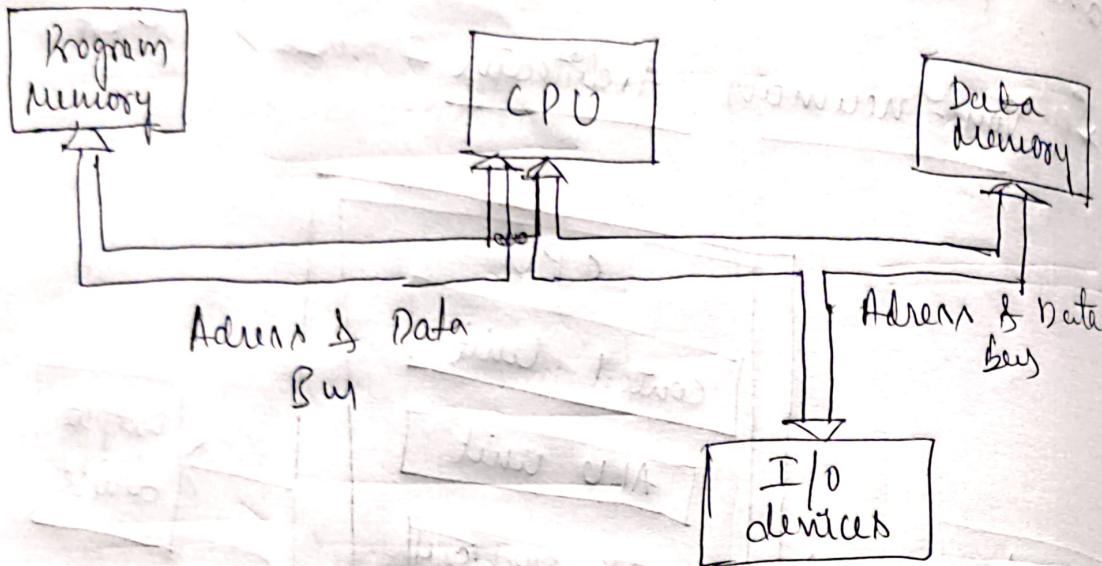
* Von - neuman Architecture is based on the stored - program computer Concept

* The instruction data and the program data are stored in the same memory.

* Only One Common Bus will help to transfer of both the instruction and data

* This method comes to play in case of Personal Computers and small computers.

* Harvard Architecture



* Harvard Architecture is a type of Computer Architecture that Separates its memory into two parts so data & instructions are stored Separately

* The architecture also has a Separate bus for data transfer and instruction transfer this allows the CPU to fetch the data and instruction at the same time.

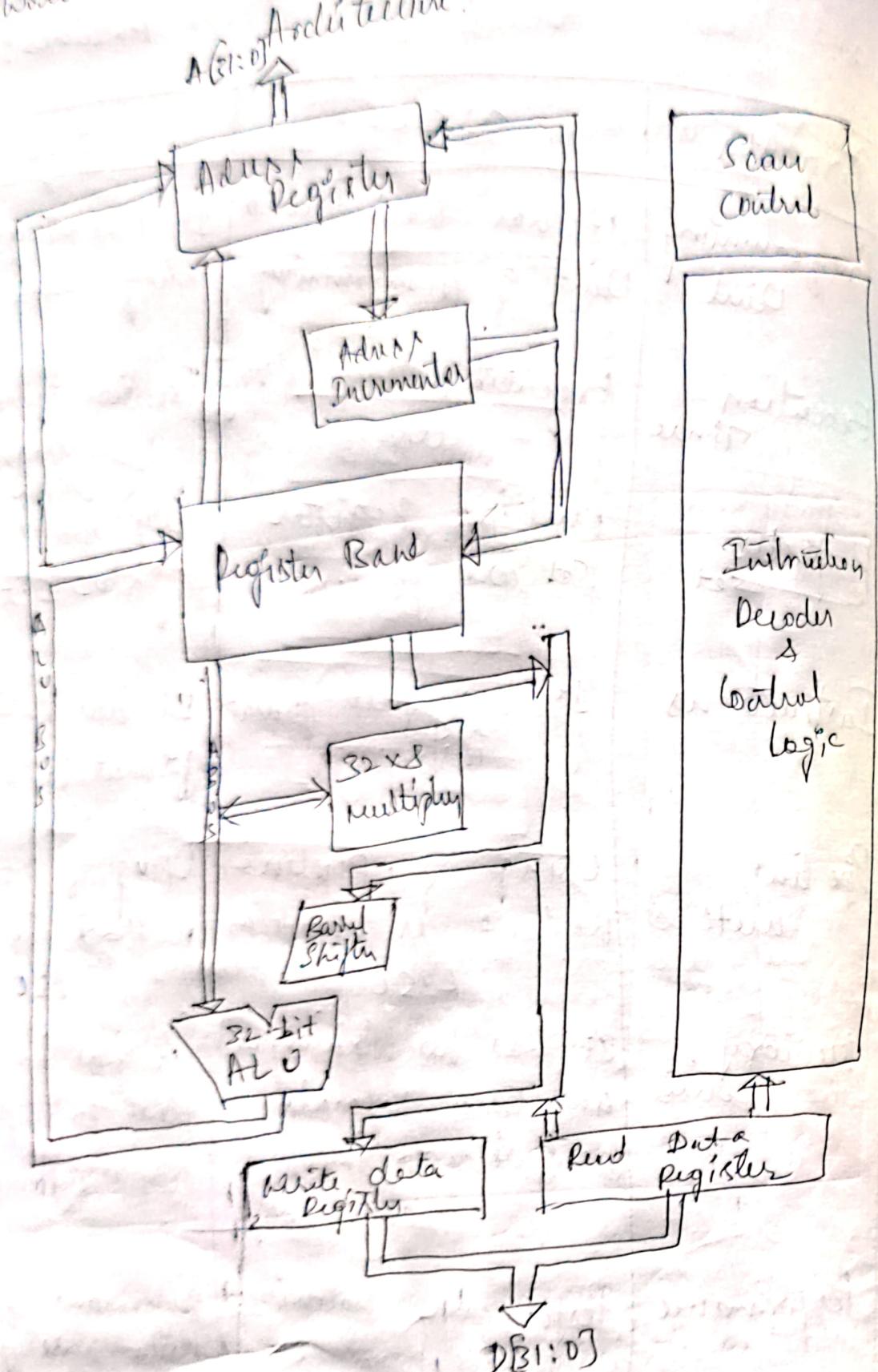
* Harvard Architecture can be faster than the Von-Neumann Architecture because data and instruction can be fetched in parallel instead of competing on the same bus.

* It is used in Microcontrollers and Signal processing.

② List the difference between RISC v/s CISC machines.

Parameter	RISC	CISC
Programming Unit	It has fixed-sized unit of programming.	It has microprogrammable Unit
Execution Time	Execution time is very less.	Execution time is very high.
Registers Set	multiple Register Set are present.	Only has a single Register Set.
Instructions	It has fixed format instructions.	It has variable format instruction.
Pipeline method	Using of pipeline method is simple in RISC.	Using of pipeline method is difficult in CISC.
Memory Unit	It has no memory Unit and uses separate hardware to implement instructions.	It has memory Unit to implement complex instructions.
Performance	Performance Optimised with more focus on Software.	Performance optimised with more focus on hardware.
Examples	ARM, AVR, PIC, SHARC	VAX, PDP-11, X86 CPU's, Motorola 6800.

⑤ Write a unit diagram explain the ALU.



Features Used

- * Load/Store Architecture
- * Fixed length 32-bit instruction
- * 3-address instruction format

- * ARM processes in a 32-bit architecture most ARM implements two instruction sets
 - * 32 bit ARM Instruction set
 - * 16 bit Thumb Instruction set.
- * Von - Neuman Architecture
- * Three stage pipeline = fetch
 - decode
 - execute
- * 32 bit Data Bus and address bus
- * 37 32 bit Registers
- * 32 bit ARM Instruction set
- * 32x8 multiplier.
- * Barrel Shifter.

Data Types

- * ARM processor support 6 data types
 - * 8-bit Signed and Unsigned bytes
 - * 16-bit Signed and Unsigned half words aligned on 2 byte boundaries
- * ARM instructions are all 32 bits words word aligned thumb instructions are half words aligned on 2-byte boundaries.
- * Generally all ARM operations are on 32-bit Operands.

With neat diagram explain programming model for ARM7.

Register	FIQ	IRQ	SVC	undf	Abort
R0					
R1					
R2					
R3					
R4					
R5					
R6					
R7					
R8					
R9					
R10					
R11					
R12					
R13					
R14					
R15					
CPSR		SPSR	SPSR	SPSR	SPSR

Each instruction can be viewed as performing a defined transformation of the states

- * Visible register
- * Invisible register
- * System memory
- * User memory

Processor modes

- * ARM has seven basic Separating modes Control (OS)
- * Modes change by Software external interrupt.

CPSR [4:0]	Mode	use	Register
0 0 0 0	USER	Normal User code.	user
0 0 0 1	FIQ	Processing fast interrupt	- FIQ
0 0 1 0	IRQ	Processing std. Interrupt	- IRQ
0 0 1 1	SVC	Processing Software interrupt	- SVC
0 1 0 0	Abort	Processing memory faults	abt
0 1 0 1	undf	Handling undefined instruction	undf
0 1 1 0	System	Handling System function	sys
0 1 1 1		Privileged OS	- user

Privileged mode

- * Most programs operate in User mode ARM has other privileges operating in User mode which are used to handle exceptions, supervisor mode occurs right to current operating mode in memory system and
- * more occurs right to co-processor mode in memory system and
- * current operating mode in memory system and

Supervising mode

- * mainly some defined by CPSR [4:0]
- * system level protection
- * specified supervisor privilege can be called
- * usually implemented by software
- * ARM has 87 registers
- * program counters all software interrupt

- * B10 dedicated save program status register.
- * 6 general purpose register.
- * mode con accm.
- Each particular set of R0-R12 register
- * stack pointer R14 - link register
- * program counter R15(PL)
- * current program status register

⑥ With a neat diagram, explain three stage pipeline of AMI.

- * Pipeline is the mechanism used by RISC processor to execute instructions.
- * By speeding up the execution by fetching the instruction while other instructions are being decoded & executed simultaneously.

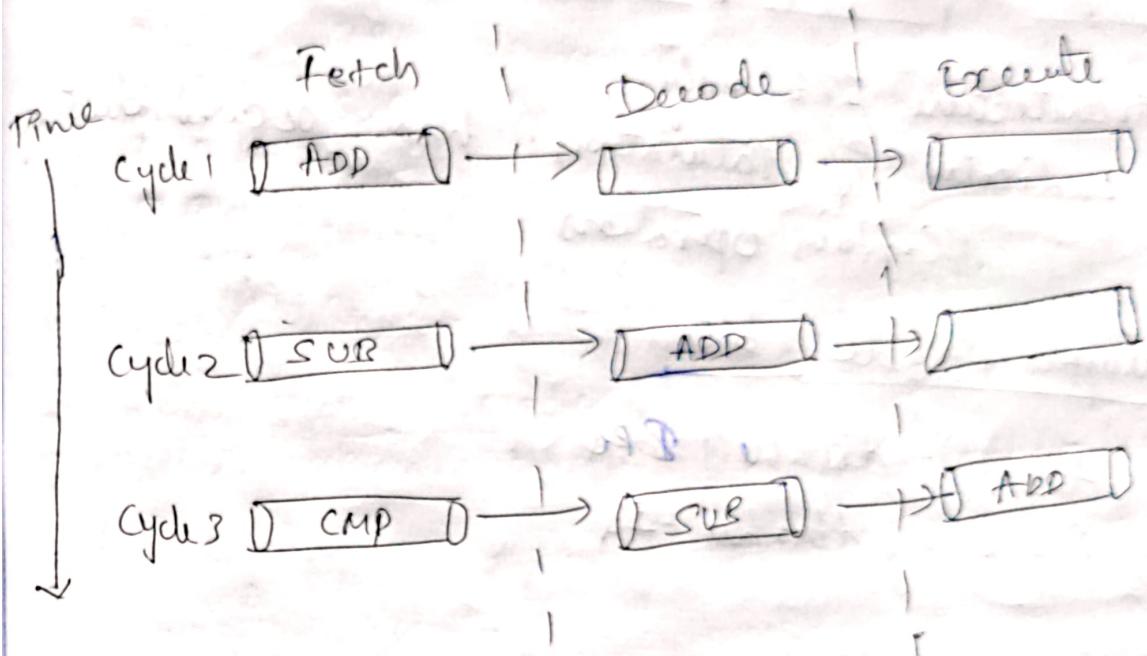
The AMI has 3-stage pipeline:-

- * Fetch - The instruction is fetched from memory.
- * Decode - The instruction opcode & operands are decoded to determine relevant functions to perform.
- * Execute - the decoded instruction is executed.

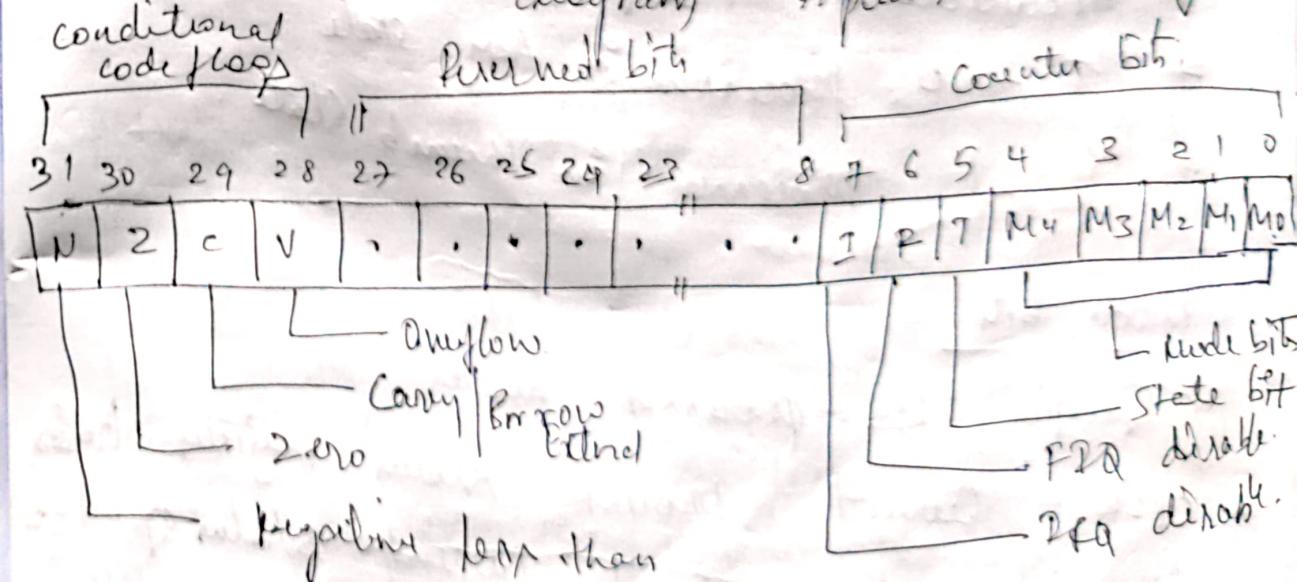
clock cycle for each of these operations required are typical instructions. Thus a normal

instruction requires three clock cycles to completely execute known as the latency of instruction execution

Because the pipeline has three stages an instruction execution is completed in every clock cycle, in other words, the pipeline has a throughput of one instruction per cycle.



⑦ With conditional code flags meet diagram explains CPSR register



Condition code

- * N - negative result from ALU
- * Z - zero result from ALU
- * V - ALU operation overflowed.
- * C - ALU operation carried out.

strictly Overflow flag - Q flag

- * Architecture STE only
- * Indicates if saturation has occurred during certain operations

Interrupt disable bit

- * I = 1 - disable IRQ
- * I = 0 - enable IRQ

T-bit

- * Architecture XT only
- * T = 0 processor in ARM state
- * T = 1 processor in Thumb state

Mode bit

- * Specifies the processor mode.
- * CPSR - Current Processor Status Register holds the information about current status of the processor.
- * SPSR - Saved Processor Status Register holds the information on processor state before the system changes to this mode.

(2) Explain the seven different modes in ARM.

The ARM+TDMI has seven modes of operation.

- * User mode is the usual ARM program execute state and is used for executing most application programs.
- * Fast interrupt (FAA) mode supports a data transfer (or) channel process.
- * Interrupt (IQA) mode is used for general purpose interrupt handling.
- * Supervisor mode is a protected mode of the operating system.
- * Abort mode is entered after a data (or) instruction or prefetch Abort.
- * System mode is privileged user mode for the operating system.
- * We can only exit System mode from another privileged mode by modifying the mode bit of the CPSR.
- * Undefined mode is taken when an undefined instruction is executed.
- * Modes other than the user mode are collectively known as privileged mode. Privileged modes are used to service interrupt (or) exceptions (or) to access particular resources.

mode

mode

Identifier

User

User

Fast interrupt

fin

Interrupt

int

Supervision

src

Abort

abt

System

sys

undefined

und

Q) Explain the nomenclature in ARM.

* First RISC processor for commercial use

ARM=Thumb processor

32-bit processor Advanced machine

T - Thumb architecture extension

D - Debug extension

M - Enhanced memory

I - Instruction set extensions

ARM {x3 {y4} {z2} TDMI {E}{J1} {F} } {G}

x - Series

y - memory Management Unit

z - Cache

T - Thumb 16 bit decoder

D - JTAG debugger

M - Fast multiplex

1 - Embedded I^cC

2 - Enhanced Instruction for DSP

3 - JAVA acceleration by Jazelle.

4 - Floating point

5 - Synthesizable Version.

(10) What is JTAG? Explain JTAG state diagram

- * JTAG has become a standard in embedded system and it is available in nearly every microcontroller and FPGA on the market.
- * If we have programmed a microcontroller there is a strong chance that we have used JTAG of the selected standards.
- * JTAG is join Test action group is an industry standard for verifying designs and testing printed circuit boards after manufacture.
- * JTAG implements standard for on chip interface in electronic design automation (EDA) as a complementary tool to digital simulation.
- * It specifies the use of a dedicated debug port implementing a serial communication interface for low overhead access without requiring direct external access to the system area by direct bus. The interface connects to an on-chip and Test action port (TAP) that implements a stateful

Protocol to access a set of registers

test logic needed

then test idle.

Select DR scan.

Capture DR

Shift DR

exit 1 DR.

Pause DR

Select DR scan.

Capture DR

Shift DR

exit 1 DR

Pause DR

⑩ What is Single Tasking? Give example & its applications.

* Single Tasking means doing one task at a time with as little distraction & interruption as possible

* Microcontrollers are known as computers on chip they are designed to perform a single task only because its processing power or limited memory is not suitable for installing and

Example :- * File Management

* Prioritization

* Scheduling

* Workspace

* Tools

* Flows

Applications :- MS. DOS,

Palm OS etc

Q2 What is MMU? Why MMU is required
give example of MMU support.

- * the memory can be defined as a collection of data in a specific format. It is used to store instructions and processed data.
- * The main memory is located central to the operation of a computer. Main memory is a large array of words of bytes. Its size ranges from hundreds to thousands and up to billions.

Memory management

In multiprogramming, computer has OS resides in a part of memory and gets it needed by multiple processes. The task of subdividing the memory among different processors is called memory management. Memory management is a method in OS to manage operations b/w main memory and disk memory. The main aim of memory management is to achieve efficient utilization of the memory.

- * Allocation & deallocation of memory before and after the program execution
- * To keep track of used memory space by the processes
- * To minimise fragmentation issue.
- * To proper utilisation of main memory
- * To maintain data integrity while executing of programs

Eg: IBM System /360 model 67, IBM System /370

Q) What is Endianness? List types and give example

Endianness means that the bytes in occupying memory are stored in a certain manner

Endianness represented in two ways -

- * Big Endian (BE)
- * Little Endian (LE)

Big Endian

Big Endian is the most common way to store the Binary data - it places the most significant value first followed by less significant values

Ex1 - The Big Endian representation of the integer 123 places the hundred's value in 1st position followed by few value and then one Value

$$123 = [1 \ 2 \ 3]$$

Little Endian

Little Endians store the least significant Value first, followed by Increasingly more Significant Value.

Ex1 - The number 123 in little endian

is stored as

$$123 = [3 \ 2 \ 1]$$

- ⑯ Write a C program to find the endianness of a given number.

#include <stdio.h>

int main()

{ Unsigned int x = 0x76543210;

char *c = (char *);

If (*c == 0x10)

{ printf("It is Little Endian"); }

```
    {  
        printf("It is big endian\n");  
    }  
    return 0;  
}
```

(b) Explain bit, byte, nibble, half-word, word.

* Bit - A bit is the smallest unit of information that can be stored in a computer. bits
Computers are grouped to form a larger unit of information

* Byte - A byte is a combination of 8 bits.
8 bits represent character and is called a byte

* nibble - A nibble is a combination of 4 bits, otherwise nibble is half a byte

* Half word - An area of storage one half the size of the word in a particular system. Usually two bytes

* Word - A word is a combination of 16 bits, 32 bits (or) 64 bits depending on the Computer. 16 is known as operand word.

Length	Name	Example
1	bit	0
4	Nibble	0110
8	Byte	01100110
16	Halfword	10110111 01101110
32	Word	10101111 10110111 1011101 0111011

Q) Explain word Align and half word align in memory

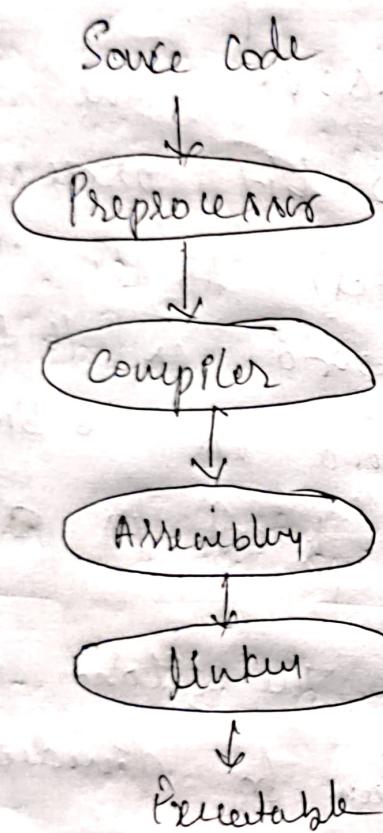
* Different processor have different definition of words. e.g. 32 bit processor, a word is 32 bit (4 bytes). At the same implies a ~~half~~ half word is 16 bit for a 16 bit processor, a word is 16 bit (2 bytes); for 8 bit processor word is 8 bits.

Word Alignment :- The stored address are adjacent and can be divided by 4, the last two digits are 00.

Half word Alignment :- That is stored address are adjacent & divisible by 2, that is last bit is 0.

ARM architecture requires 32 bit ARM instructions
that must be word aligned & 16 bit memory and 16 bit thumb instructions require
half word aligned & 8 bit.

- Q17 Explain the software tool involved in
and processing C source file with a
diagram.



- * Normally the C program building process involves four stage and utilize different tools such as a preprocessor, compiler, assembler & linker.
* At last there should be a single executable file.

- * Preprocessing :- It is the first part of any compilation, it processes instruction & macros.
- * Compiler :- In the second part, it takes the output of the preprocessor and the source code, and generates assembly source code.
- * Assembly :- is the third stage of compilation of take assembly source code & produces an assembly listing with offsets. the assembly o/p is stored in an object file.
- * Linking :- is the final stage, it takes one or more object files (or) libraries as i/p. and combines them to produce a single executable file.

- ⑧ Explain the following addressing modes in ARM
- Three address
 - Two address
 - One address

One address Instructions

This uses an implied Accumulator register for data manipulation and the other operand is the register (R) memory location, implied means that the CPU already knows that one operand is in the accumulator so there

there is no need of specifying it.

Ex:- LDR addr

Acc ← [addr]

Two address Instruction

Here two address can be specified in the instruction, in one address instruction the result was stored in the accumulator, here the result can be stored in different location, i.e. Register or memory location. but requires more number of bit to represent the address.

Ex:- Mov R1, R2

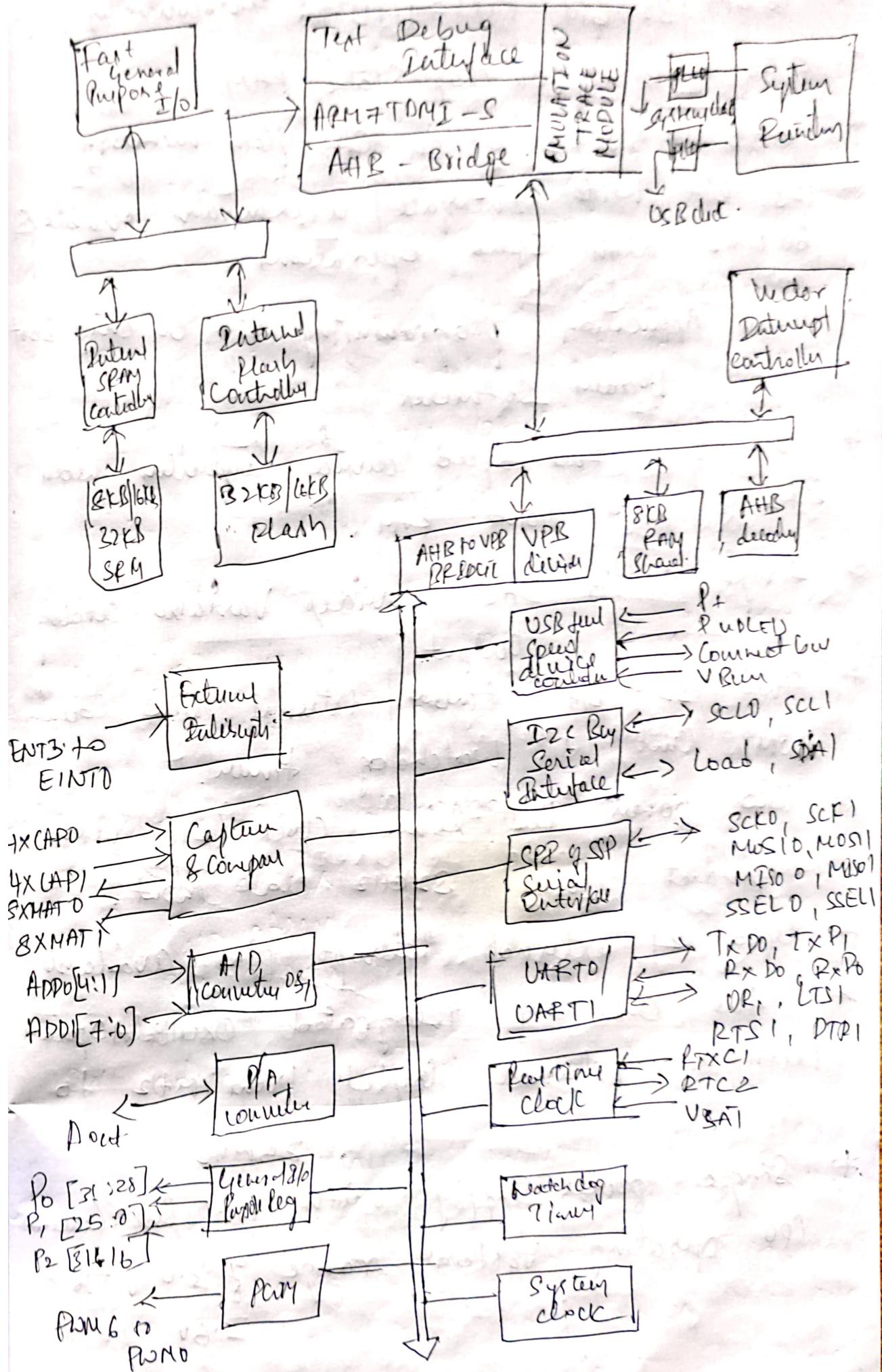
R1 ← [R2]

Three address Instructions

This has three address field to specify a register or memory location. Program created are much short in size but no of bit per instruction increase. These instructions make creation of programs much easier, but if don't mean that program will run much faster because now instruction only contains more information but each microprogram will be performed in one cycle only.

Ex:- ADD R3, R1, R2 R3 = R1 + R2

Q9 Explain the Uc2148 microcontroller Block diagram



Processor Features

- * 16MHz / 32-bit ARM7TDMI - 8 Microcontroller
- * 8 to 40K of on chip static RAM
- * 32KB to 50 KB of on chip flash memory
- * 128 bit wide Dataflow / accumulators, enable 60MHz high speed operations.
- * In Application programming via on chip board software.
- * USB 2.0 Full Speed data controller, with 2KB of end point RAM.
- * Single 10 bit DAC provides variable analog output.
- * Two 32 bit timer / external event counter PWM out & watchdog timer.
- * low power real-time clock (RTC) with independent power 32kHz clock input.
- * Up to 21 external interrupt pins available
- * the on-chip integrated Oscillators with an external crystal from 1MHz to 25MHz
- * Single power supply with POR, BOD with 100 operating voltage range of 3.0V to 3.6V

Explain the LPC2148 microcontroller GPIO pins

LPC2148 has two 32 bit general purpose I/O pins

* PORT 0

* PORT 1

Port 0

- * Out of 32 pins, 28 pins can be configured as either general purpose I/O.
- * 2 of these 32 pins can be configured as O/P only (P0.30, P0.31)

Port 1

- * It is also a 32 bit port.
- * Out of only 16 (P1.16 - P1.32) pins are available for general I/O.
- * The functionality of each pin can be selected using the pin function select register.

PINSELECT

- * Pin select registers are 32-bit registers.
- * These are used to select and configure specific pin functionality.

PINSEL0: It is used to configure PORT0 pins from P0.0 to P0.15.

PINSEL1 from port 1 to Po.16 to Po.32
in port 2 from Po.16 to Po.32

PINSEL2 1- It is used to configure pins
in port 2 from Po.16 to Po.32

Slow GPIO Registers

- * D0XPIN
- * D0XSET
- * D0XDIR
- * D0XCLR

D0XPIN

- * 32 bit wide register, this register is used to read/write value on Port1/Port2.
- * Care should be taken while writing. masking should be used to ensure write to the desired pin

Ex)- writing 1 to Po.4 using D0PIN

$$D0PIN = D0PIN \text{ (1cc9)}$$

D0XSET

- * This is 32 bit wide register. the register is used to make pins of port0/port1 as high

* Writing 1 to specific pin makes high
writing 2 to has no effect.

Port X DDR

- * This is a 32 bit wide Register. This register individually controls the direction of each port pin.
- * Setting a bit to '1' configures the corresponding pin as the output pin. Setting a bit '0' configures the corresponding pin as an I/P pin.

Port X CLR

- * This is a 32-bit wide Register.
- * This reg is used to make pins of port low.
- * Writing 1 to specific bit makes that pin low, writing 2_{no} has no effect.

Eg:- Configuring pin P0.4 as O/P they set the pin to low.

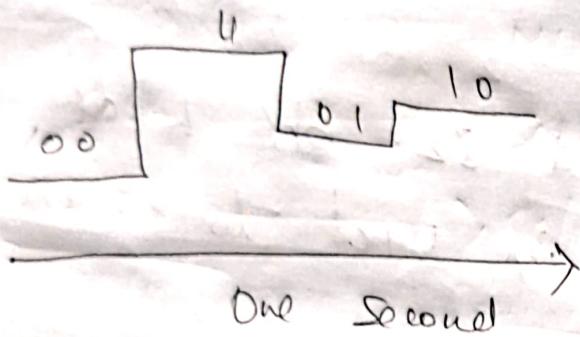
$$\text{PORTDR} = 0x0000\ 0010;$$

$$\text{PORTCLR} = (1cc4);$$

Given a Mat diagram explain Baud rate and bit rate. Explain the calculation.

Baud rate and Bit rate

- * Number of time line changed per second
- * Let Baud rate be 4
- * Let bits per line change be 2.
- * Bit rate = 8 bits per second
- * Bit rate = $\times 2$ Baud rate per example



- * Baud rate defines the switching speed of a signal
- * Bit rate defines the rate at which information flows across a data link measured in bits/second.
- * For binary two-level Signal. A data rate of one bit/second. is equivalent to one Baud.

1 bit \rightarrow 1 Symbol

$$\text{Bit rate} = \text{Baud rate}$$

$$\text{Bit rate} = 1000 \times 4 = 4000 \text{ bps.}$$

Formula $b = s \times n$

b = Data Rate (bits per second)

s = Symbol rate (Symbol/sec)

n = Number of bits per second.

if $n = 1$ Band rate = Bit rate

$n = 4$ Bit rate = $4 \times$ Band rate

(Q2) with neat diagram explain the working features of SPI protocol

* The Serial Peripheral Interface (SPI)

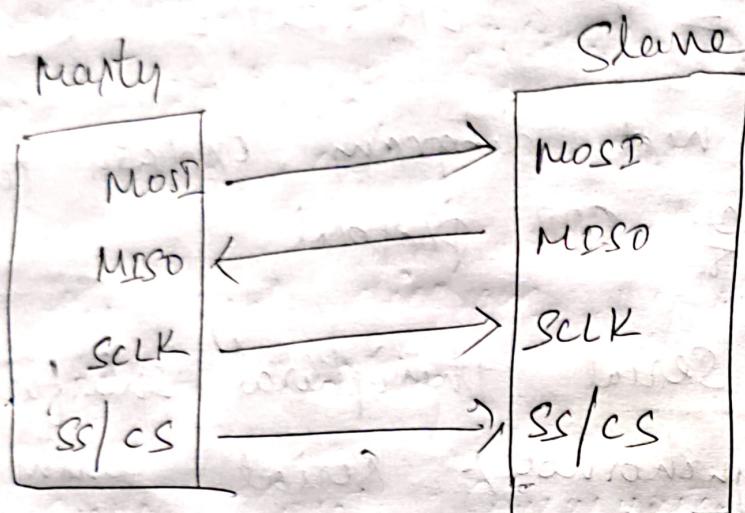
↳ a Synchronous Serial Communication interfacing Specification used for short distance communication primarily in embedded systems

* SPI devices communicate in full duplex using a master slave architecture usually with one single master. The master device generates the frame for reading and writing. multiplex slave may be supported through Selection with individual chip select (cs) sometimes called slave select (ss)

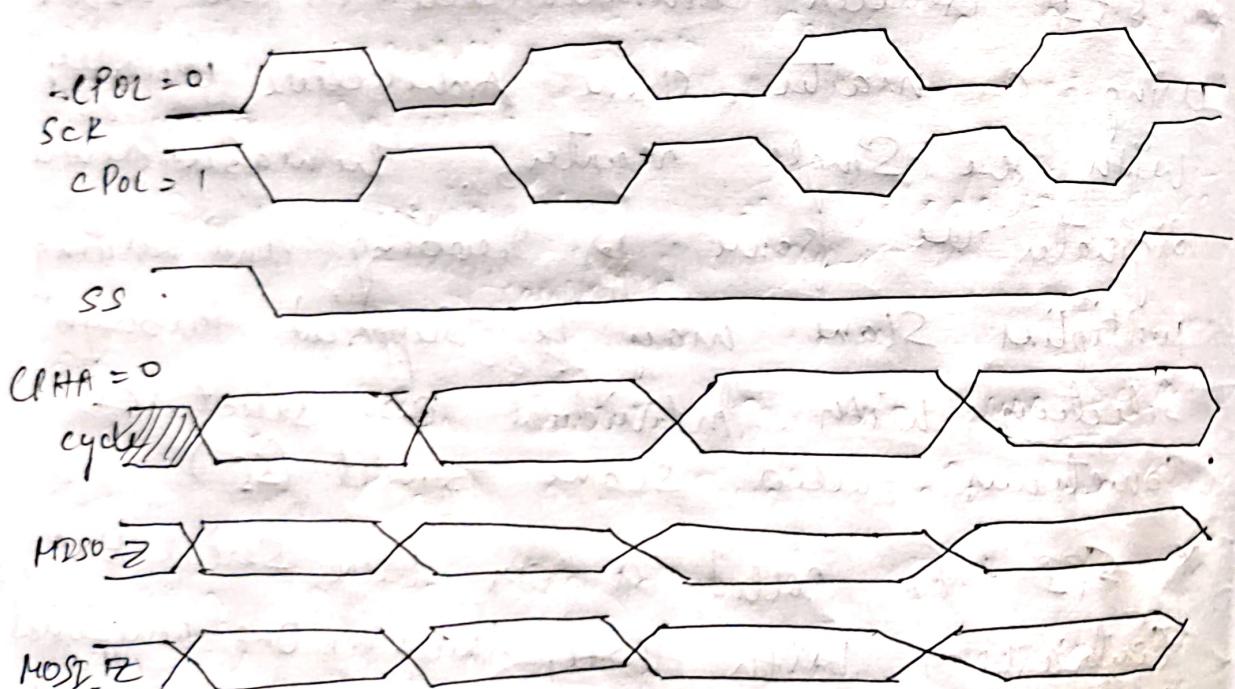
* SPI is called a four wire serial Bus, contrasting with three, two, or one wire serial bus. In SPI may be accurately described as a Synchronous Serial Interface.

The SPI bus specifies four logic signals.

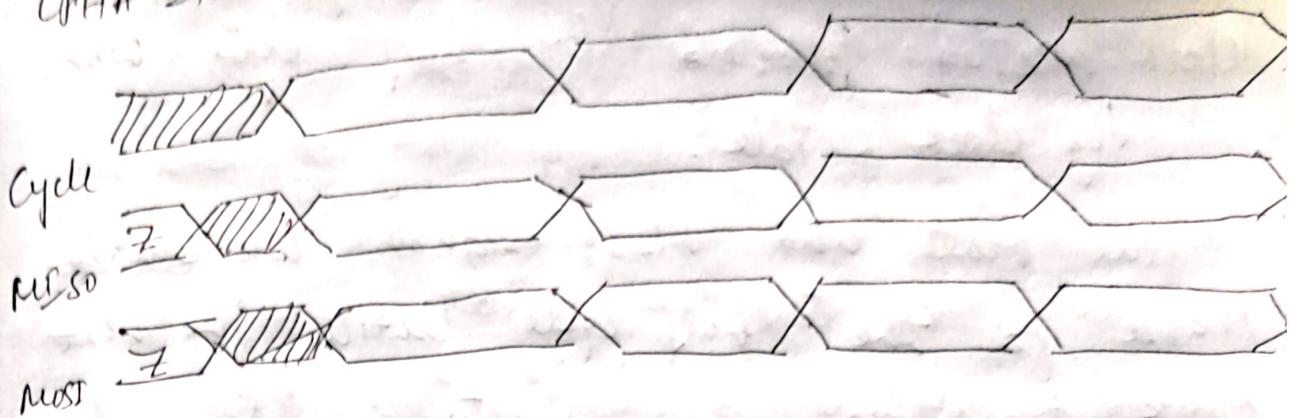
- SCLK :- serial clock (Opp from master)
- MOSI :- master out slave in (data from master)
- MISO :- master in slave out (data O/P from slave)
- SS/CS :- chip / slave select



Q23) With neat timing diagrams explain CPHA and CPOL usage in SPI.



CPHA = 1



CPOL determines the polarity of the clock the polarities can be converted with a simple truthy

CPOL = 0 in a clock which idles at 0, and each cycle consists of a pulse of 1. That is the leading edge is the rising edge and the trailing edge is the falling edge.

CPOL = 1 in a clock which idles at 1 and each cycle consists of a pulse of 0. That is leading edge is a falling edge and trailing edge is the rising edge

For CPHA = 1 the out side changes the data on the leading edge of the clock cycle, which inside capture the data. On the trailing edge of the clock cycle the outside holds the data valid until the leading edge of the following clock cycle. For the last cycle, the slave holds the MISO line valid until the slave select is disabled. An alternative way of considering it is to say that a

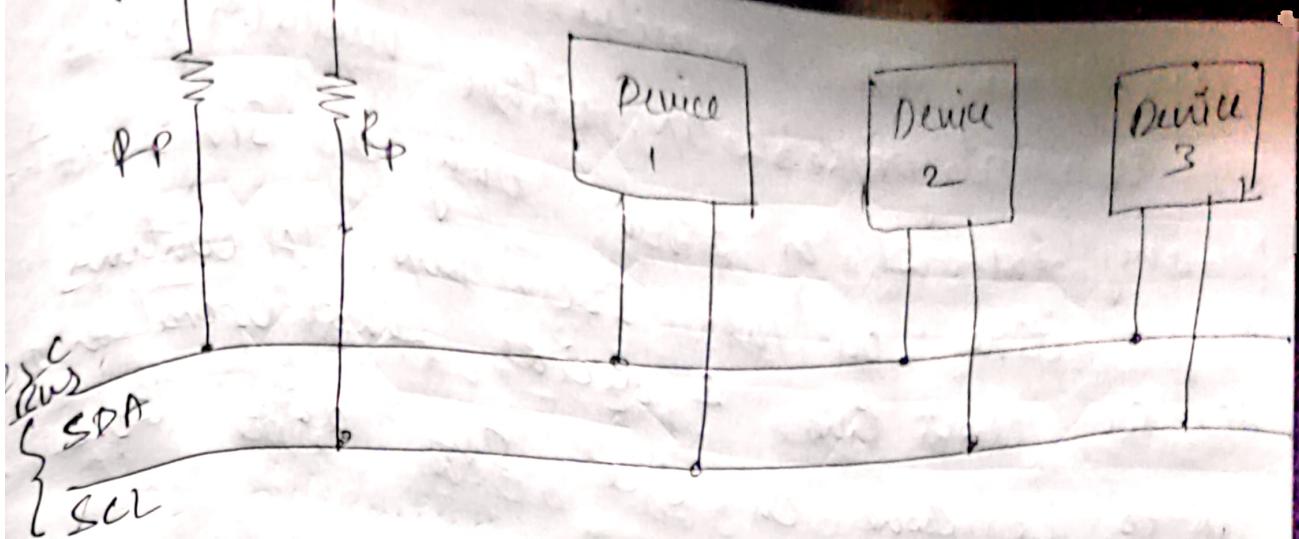
CPHA = 1 cycle consists of the half cycle clock asserted followed by half cycle ^{high}
^{low} the clock idle.

The MISO and MOSI signals are usually valid for the half cycle until the next clock transition, SPI master and Slave ^{will} sample data at different points ^{of} _{in fact} ^{PD} _{half cycle}.

Q) With the neat diagram explain the ^{of} I₂C & its working.

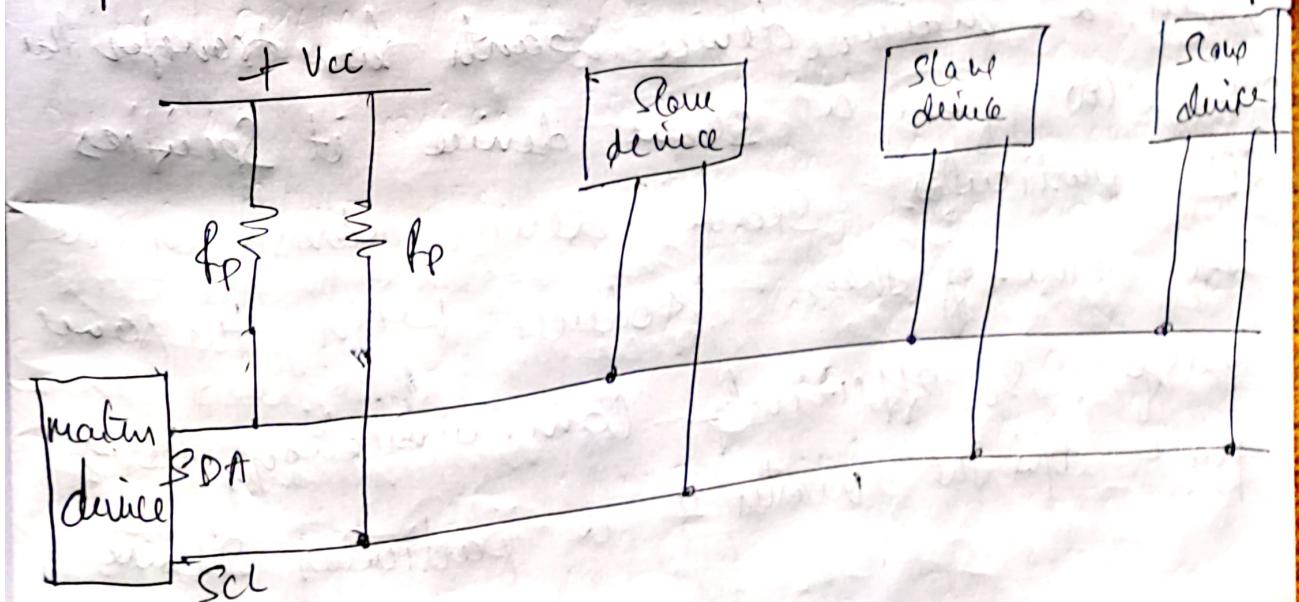
- * Half duplex Serial Communication
- * Synchronous communication protocol
- * Only two common bus lines are required to control any device I₂C or I₂C(N)
- * Data transfer speed can be adjusted between required
- * Simple mechanism for validation of data transferred.

I₂C Bus consists of two wires Serial clock wire (SCL) and Serial data line (SDA) the data to be transferred is sent through the SDA wire & synchronized with the clock signal from SCL. All the devices I₂C or I₂C(N) are connected to same SCL/SDA lines.



Both I₂C bus lines (SDA, SCL) are operated on open drain devices. It means that any device on the I₂C bus can drive SDA and SCL low, but they can not drive them high, so a pull up resistor R_P is used for each bus line to keep them high by default.

The reason for using an open drain system is that there will be no chance of shorting which might happen when one device tries to pull the line high & some other tries to pull the line low.



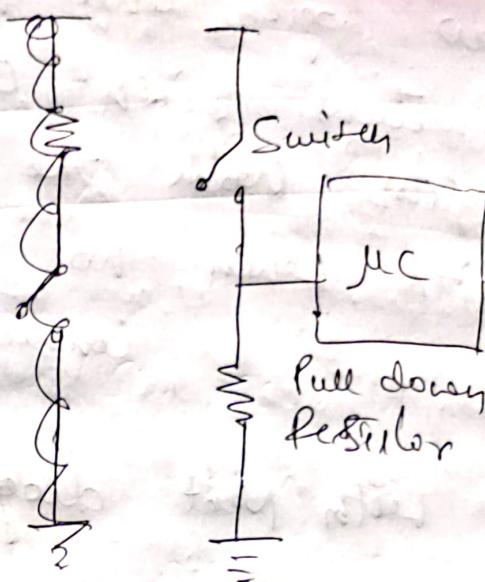
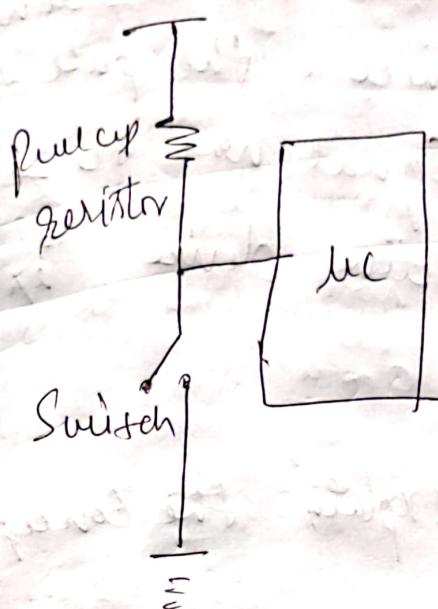
The devices are connected to the I₂C bus & are categorized as either master or slave.
At any instant of time only a single device stays active on the I₂C bus. It controls the SCL clock line & decides what operation is to be done on the SDA data bus.

All the devices that respond to instruction from this master device & slaves. At any instant of time only a single master stays active on I₂C bus. It controls the SCL line & decides what operation is to be done on the SDA data line.

All the devices that respond to instruction from this master device are slaves for differentiating b/w multiple slave devices connected to the same I₂C bus, each slave device is physically.

When a master device wants to transfer data (R/W) from a slave device it specifies this particular slave device address on the SDA line & then producer writes the raw frame. So effectively communication takes place b/w the master & a particular slave.

Q5 with neat diagram explain pull-up and pull-down register.



Pull up resistor

* A pull up resistor is used to establish an additional loop over the critical components while making sure that the voltage is well defined even when the switch is open.

* It is used to ensure that a switch is pulled to a high logical level in the absence of an input signal. pull up resistor with a fixed value was used to connect the V_D supply and a particular pin in digital circuit.

Pull down resistor

* pull down resistors is used to ensure that IP to the logic systems settle at expected logic level whenever the external drivers

are disconnected (or) of high impedance
* It ensures that the line is at a ^{high} logic level when there are no additional connections with other device. The pull register holds the logic signal near to 5 Volts when no open active device is connected.

Q2 With neat diagram explain the concept of arbitration in I₂C

* I₂C is designed for multimaster purpose this means that more than one device can initiate transfer

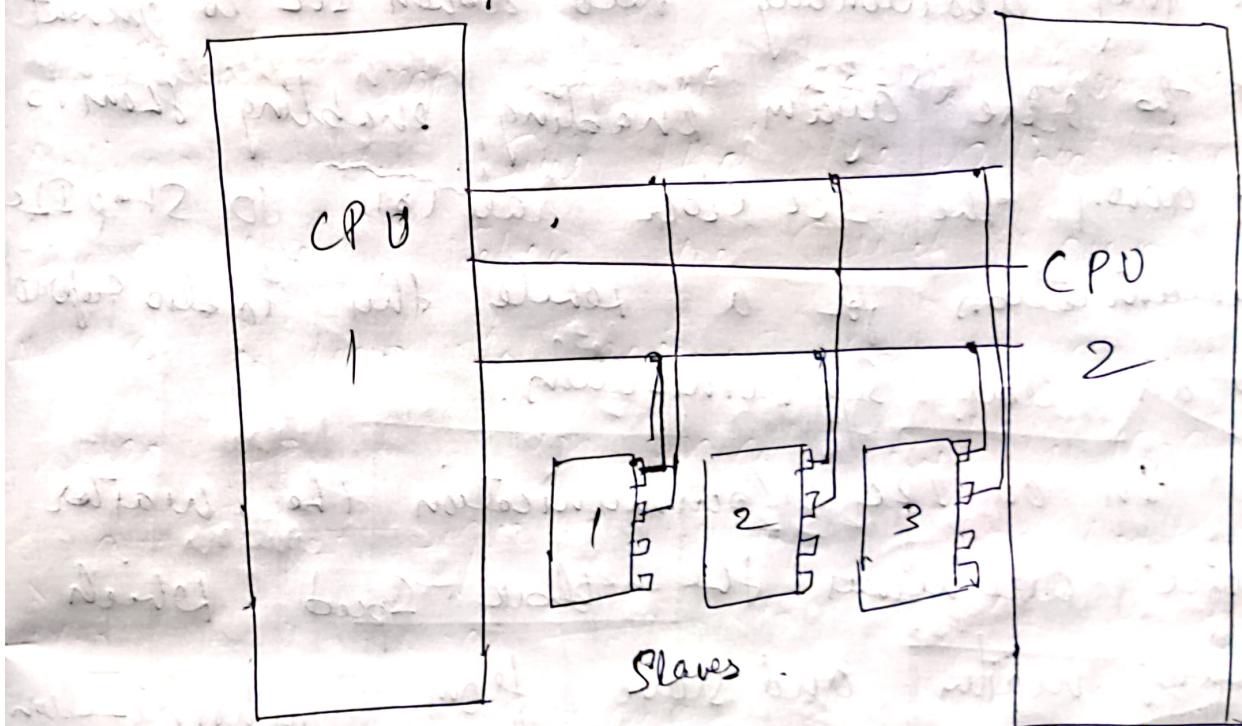
* Bus arbitration occurs when two or more masters start a transfer at the same time.

* I₂C bus was originally developed as a multimaster bus. this means that more than one device initiating transfer can be active in the system.

* By using only one master on the bus there is no real distribution of captured data except if a slave device is malfunctioning or if there is a fault condition involving in the SDA/SCL lines.

& send the address all slave chip listen.
if the address does not match the address
of CPU2, this device has to hold back any
activity until the bus becomes idle again
after a Stop Condition

As long as two MCUs monitor what is going
on the bus and as long as they are aware
that a transaction is going on because that
last issued Command was not a STOP there
is no problem.

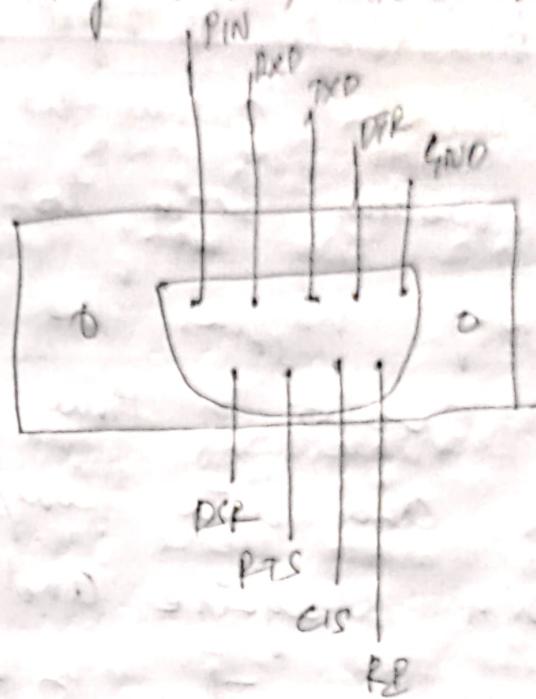


Q) What is clock stretching? Explain clock stretching in I₂C.
Clock stretching allows an I₂C slave device to pull the master device into a wait state. A slave device may perform clock stretching when it waits more time to acknowledge back or send some data (or) it may transmit another type of data.

clock stretching in I₂C devices can slow down communication by stretching SCL. During an SCL low phase any I₂C devices on the bus may additionally hold down SCL to prevent it from rising again enabling, enabling slave to slow down the SCL clock rate (0.1) to stop communication for a while this is also known as an clock synchronization.

In an I₂C communication the master device determines the clock speed which releases master and slave from synchronization mostly to predefined baud rate.

Q2 Explain the working of DCE pins and
also detailing both the modems



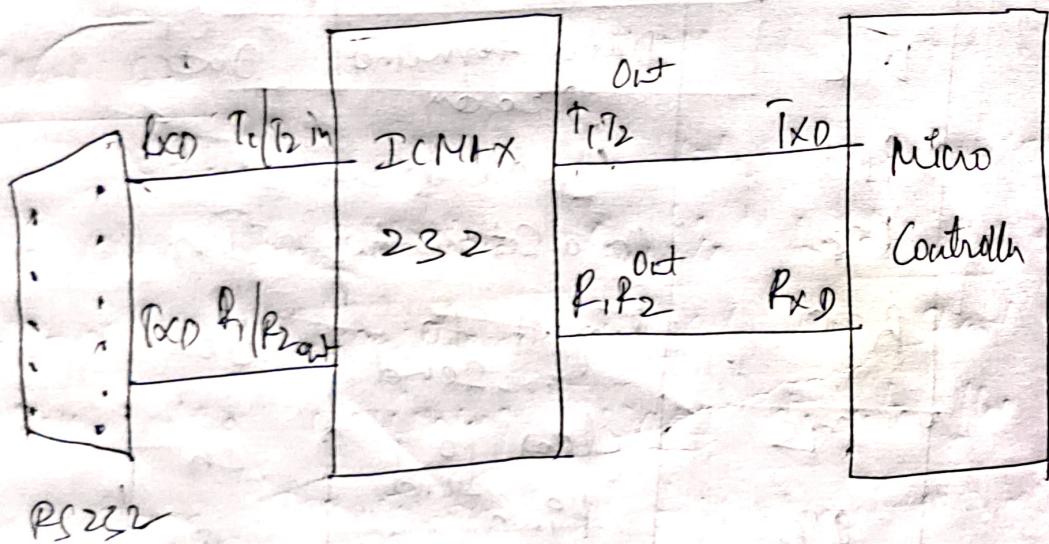
Pin	Signal	Signal name	DTE Signal direction
1	DCD	Data carrier detect	In
2	RXD	Receive data	In
3	TXD	Transmit data	Out
4	DTR	Data terminal ready	Out
5	GND	Ground	-
6	DSR	Data Set Ready	In
7	RTS	Request to send	Out
8	CTS	Clear to send	In
9	RI	Ring Indicator	In

Handshaking

A modem handshake is 1st occurs when the receiving modem answer the phone call and the two modem begin to communicate.

Before anything happens the modem may evaluate the quality of line negotiate control protocols and data compression that they can both recognise and work and the most suitable connection speed should based on the conditions this process is called a handshake.

- 29) Explain RS232 connection with Microcontroller



Several devices collect data from Sensors & need to send it to another unit like a computer for further processing. Data transfer/communication is generally done in two ways. It is fast and uses more number of lines.

Serial communication on the other hand has only one or two data line transfer and it is generally used for long distance communication. In serial communication the data is sent at a on bit at a time.

An important parameter considered while interfacing signal port is the baud rate which is the speed at which data is transmitted serially. Microcontroller can be set to transmit and receive signal data at different baud rate using software instructions.

Q) Explain the frame format in UART communication.

Baud rate :- Baud rate in a data transmission refers to the no of symbols transferred per second. A symbol is a group of a fixed no of bits.

Data framing :- UART transmits data in packets. Each data packet may contain one start bit, 5 to 9 data bits, an optional parity bit

and 1 or 2 stop bits

START	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	PR	STOP.
-------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----	-------

The UART receives the data from the data bus and this data are being sent by CPU memory (or) we:

The data transmission from the data bus to UART is in parallel mode.

UART adds the start bit, parity bit and a stop bit to the data received from the data bus which forms a data packet.

This data packet is finally transmitted to the receiving UART by the transmitter the receiving pin of the receive UART reads the data bits by bit.

Start Bit

When there is no data transmission the UART transmissiby line is held at high voltage, its transition acts as the start bit. When the receiving UART detects the high low voltage transition, it begins reading the data frame at the frequency of the baud rate.

Data Frame

The data bits are usually 5 to 8 bits.

number if no parity bit is used, it can be 9 bit long. In general case the LCR of the data is transmitted first.

Parity bit

The parity bit is used to indicate the change in data during transmission. However if the change in the data is mismatched band noise, electromagnetism over long distance data transfer.

Stop-bit

To mark the end of the data packet the sending UART drives the data transmission line from a low voltage to a high voltage for a minimum of two bit duration.

- In
- Q) Explain the difference between
- ① Serial v/s parallel
 - ② Analog v/s Digital
 - ③ Synchronous v/s Asynchronous

Serial	Parallel
* Data is transmitted bit after the bit in a single line.	* Data is transmitted simultaneously through group of lines.
* Data Congestion takes place.	* No Data Congestion.
* Low Speed transmission.	* High Speed transmission.
* Implementation of serial link is not an easy task.	* parallel data lies on early implemented at hardware.
* No crosstalk problem.	* Crosstalk creation see interface G/w parallel lines.
* Bandwidth of serial line is much higher.	* The Bandwidth of serial line is much lower.

Analog	Digital
* Transmitted modulated signal is analog in nature.	* Transmitted Signal is digital in form of digital pulse.
* Amplitude, frequency or phase variation in the transmitted signal represent message.	* Amplitude levels of transmitted pulse in digital form of code words.

- | | |
|--|--|
| * Noise immunity is poor for FM & PM | * noise immunity Excellent |
| * FDM is used for Multiplexing | * TDM is used for the Multiplexing. |
| * Analog modulation Systems are AM, PM, PM, PAM, DWM | * Digital Modulation Systems are PCM, DM, ADM, DPCM. |

Synchronous

* Communicated in fixed time

* Greater latency in a holding

* Sends the data and receives the data on the same clock frequency

* Faster

* There is no overhead of extra start & stop bit

* Uses constant time interval

* Used in chat rooms such as Video Conferencing

Asynchronous

* Communicated in the real time

* Eliminates interrupts

* Sends the data & receives the data on different clock frequency

* Slower

* Over's start and stop bit

* uses random or irregular time interval

* Used in email