1) write a neat diagram, Explain Von Neuman and Harvard Architecture.

Harvard Architecture	Von Neumann Architecture
* Two memories with 2 Buses allow parallel access to data and instruction	* Content of the memory is organized & all installed memory can be used.
* Control unit for two buses is more complicated & more Expensive	* one bus is simpler for the control unit design
* Both memories can use different size	* computer with one bus is cheaper.
* Both memories can use different size	* computer with one bus is cheaper
* Development of a complicated control unit needs more time	* Error in a program can overwrite instruction & crash program Execution
* Free data memory can't be used for instruction & vice-versa	* Development of the control unit is cheaper & faster
	* Data & instruction is accessed in the same way
	* one bus (for data, instruction ins & address) is a bottle neck

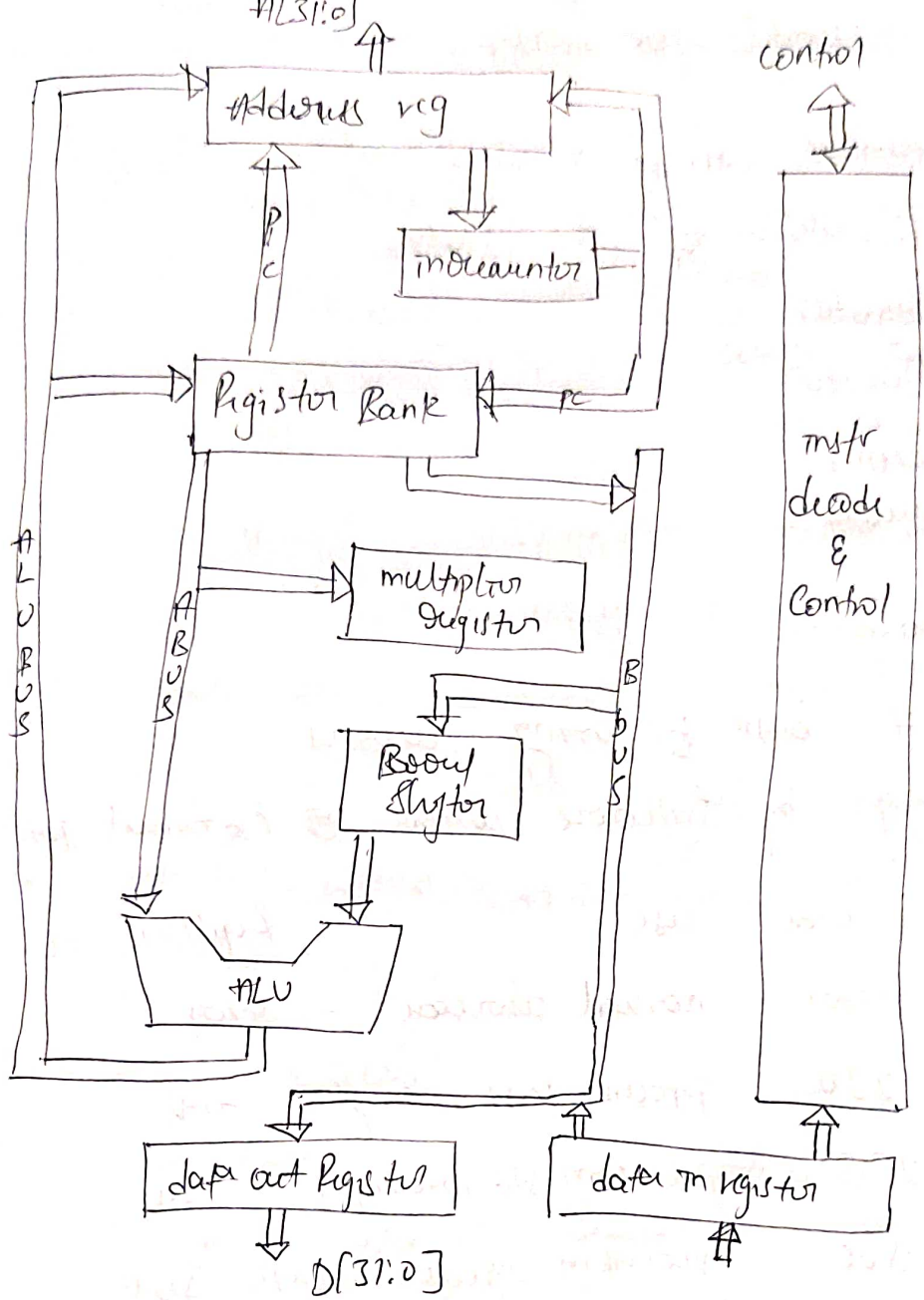
2) List the difference RISC V/S CISC machines.

→

CISC	RISC
* It has more instructions	* It has fewer instructions
* More addressing modes	* use less addressing modes.
* Requires a large amount of RAM	* RISC requires more RAM
* The avg clk cycle/instructions of a RISC processor is b/w 2.5 to 15 cpi's	* The avg clk cycles/instruction of a RISC processor is 1.5
* Large code size	* Small code size
* CISC processor have variable instruction format	* RISC processor have a fixed instruction format
* Single register set	* Multiple register sets are present
* CISC processor are not highly pipelined @ less pipelined	* RISC processor are highly pipelined
* High power consumption	* Low power consumption

3) With a neat diagram Explain ARM7 architecture

→ * Load-store architecture
* Basically 32-bit but supports 16 bits & 32 bits
* Basically RISC architecture
* has Auto increment & Auto decrement modes
* consist of Barrel shifter to increase throughput
* ARM is designed to suit Embedded devices
* ALU & Barrel shifter operations done in same cycle



- All registers are 32 bit
- In user mode, 16 data registers [R0-R15] & 2 status Reg
- 3 special function register out of 16 Reg are for data they are R13, R14, R15, 1.
- R13 → Stack Reg R14 → link Register, R15 - Program counter
- R13 & R14 are used as general purpose register
- CPSR, SPSR are 2 Status register
- It has 7 modes.
- privileged mode: abort, fast interrupt request, interrupt request, supervisor, System & undefined.
- Non privileged mode: user mode.

4) Programming model for ARM7

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

* Visible Register
* Invisible Register
* System memory
* User memory.

Processor modes:-

→ ARM has 7 basic operating modes

→ Mode Changes by Software control or External interrupt

CPSR[4:0]	mode	use	Register
10000	user	normal user code	user
10001	FIQ	Processor Fast interrupt	fiq
10010	IRQ	processing std interrupts	irq
10011	SVC	processing Software interrupt	SVC
10111	Abort	processing memory interrupt	abt
11011	Undef	Handling undefined instruction type	und
11111	System	Running privileged OS	user

→ Most programs operator in user mode ARM other privileges operating modes which are used to handle exceptions, supervision calls and system modes.

→ Mode access right to memory system & coprocessor

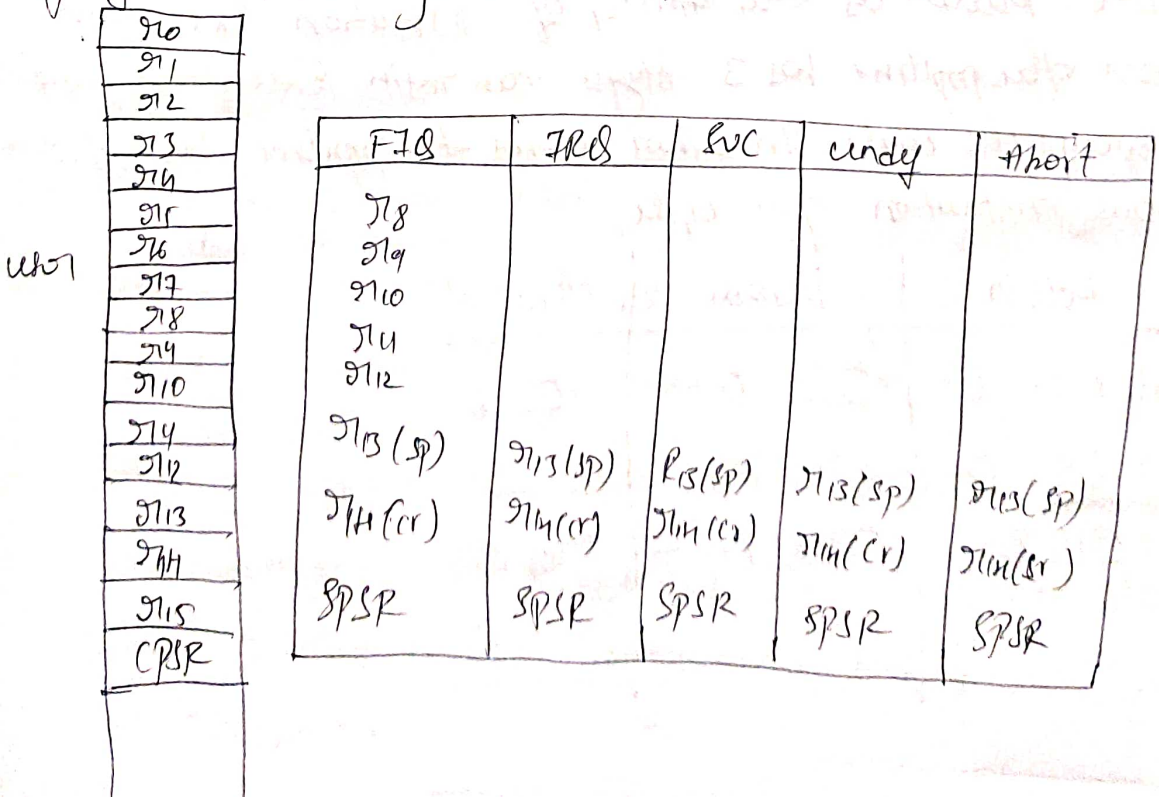
→ Current operating mode is defined by CPSR[4:0]

Privileged modes:

Supervisor mode:

→ Having some protective privilege
→ System level function can be accessed through special Supervisor calls
→ usually implemented by software interrupt.
ARM has 37 registers all of which are 32 bits
→ 1 program counter
→ 1 current program status register
→ 5 dedicated saved program status Registers
→ 30 general purpose register.

Each mode can access
→ a particular set of R0 - R12 registers
→ Stack pointer, R14 - link register
→ Program counter (R15)
privileged modes can access a particular SPSR - saved program status register



6) With a neat diagram, Explain three stage pipeline org

-> A pipeline is the mechanism used by RISC processor to Execute instructions

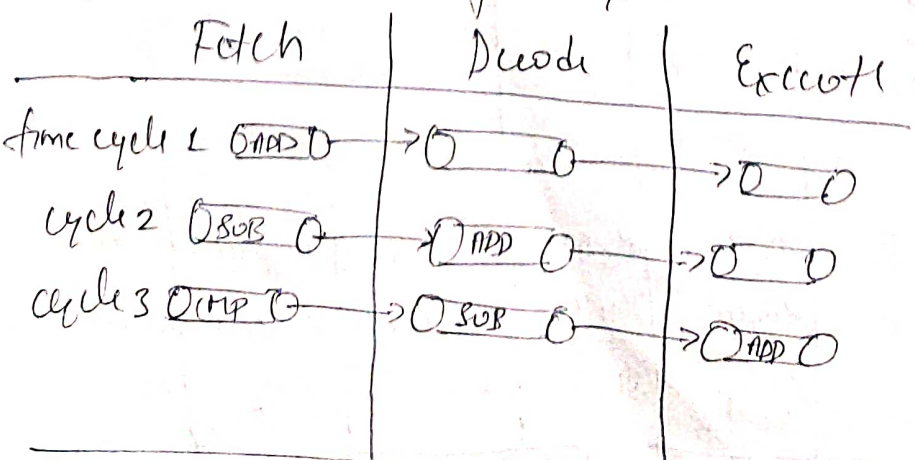
→ By speeding up the execution by fetching the instruction while other instruction are being decoded & executed simultaneously.

→ Pipelining is a design technique of process which plays an important role in increasing the efficiency of data processing in the processor of a computer & PC

The ARM7 has 3 stage pipeline

* Fetch: The instr is fetched from memory
* Decode: The instr opcode & operands are decoded to determine what function to perform
* Execute: The decoded instr is executed.

Each of these operations require one clock cycle for typical instructions Require one clock cycle for typical instruction thus a normal instn require three clk cycle to completely Execute, Reason as the latency of instruction execution Because the pipeline has 3 stages can instn Execution completed in every clk cycle. In other words the pipeline has a throughput of one instruction per cycle.



3) With the neat diagram explain CPSR register N Z C V Q and updated IFT mode control code flags

→ N – Negative Result from ALU
→ Z – Zero Result from ALU
→ C ⇒ ALU operation overflowed
→ V – ALU operation carried out

Sticky overflow flag. Q flag
– Architecture 5TE only
– Indicate if saturation has occurred during certain operation

Interrupt disable bits
→ I = 1 Disable the IRQ
→ I = 0 Disable the FIQ

T bit
→ Architecture XT only
→ T = 0 Processor in ARM State
→ T = 1 Processor in Thumb State

Mode Bits :-
→ Specify the processor mode
– CPSR → current processor status reg holds things about the current processor
→ SPSR → Saved processor status register holds the information on the processor state before the system changed to this mode. i.e processor status just before an exception.

8) Explain the 7 different modes in ARM

The ARM 7 TDMI, processor has 7 modes of operation

-> user mode is the usual ARM program can state & is used for executing most application program

-> Fast interrupt mode supports a data transfer channel process

-> Interrupt mode is used for general purpose interrupt handling

-> Supervisor mode is a protected mode for the operating system

-> Abort mode is entered after a data or instruction prefetch abort

-> System mode is a privileged user mode for the operating system.

We can only enter system mode from another privileged mode by modifying the mode bit of the current program status register (CPSR)

-> Undefined mode is entered when an undefined instruction is executed

Modes other than user mode are collectively known as privileged modes. Privileged modes are used to service interrupt or exception to access protected resources.

Mode	Mode Identifier
User	usr
Fast interrupt	fiq
Interrupt	irq
Supervisor	svc
Abort	abt
System	sys
Undefined	und

9) Explain the nomenclature in ARM

-> * first RISC processor for commercial use ARM7TDMI processor.

32-bit processor Advanced Risc line

T- Thumb architecture extension

D- debug extension

M- Enhanced extension

I- Inchrut simulation

ARM {x} {y} {z} TDMI {E} {J} {F} {S}

x - Series

y - Memory management unit

z -> cache

T - Thumb 16 bit decoder

D - JTAG debugger

M - Fast multiplier

I - Embedded ICE

E - Enhanced instruction for DSP

J - Jave acceleration by Jazzle

F -> floating point

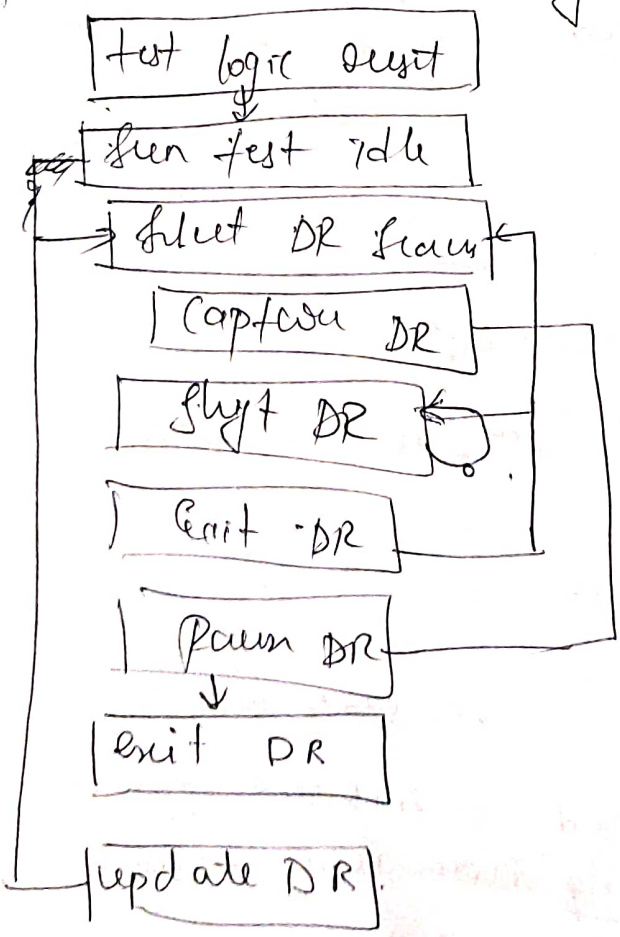
S - Synthesizable version

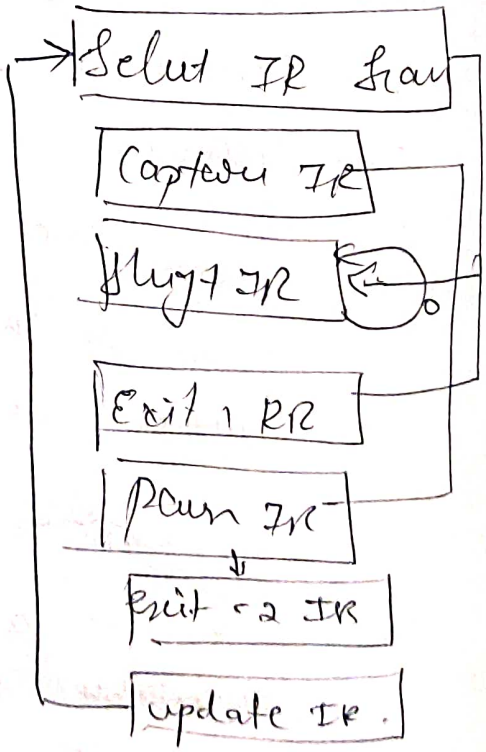
10) what is JTAG? Explain JTAG State diagram

* JTAG has become a standard in Embedded systems 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 related standards

* JTAG is joint test action group is an industry standard for verifying design and testing printed circuit boards after manufacture
* JTAG implements standard for on chip instrument is Electronic design automation as a complementary tool to digital simulation
* It specifies the use of a dedicated debug port implementing a serial communication interface for low overhead without requiring direct external access to the system address by direct buses. the interface connects to an chip and test acroos port (TAP) that implements a stateful protocol to access a set of test registers





11) What is Real time? Give example & its applications

* Real time system means doing all task data from both as little interaction & interruption as possible

* Microcontroller are known as computer on chip they are designed to perform a Real time task only because its processing power as well as memory is not suitable for multiple os

Example:- * Multiple management
* prioritization
* Scheduling
* workspace
* Tools
* Locks

Applications:- MIS DSS, global os etc.

12) What is RMO? why RMO is required give example of RMO support

→ * The memory can be stored as a collection of data in a specific format. It is used to store instruction and processed data.

* The main memory is central to the operation of a computer main memory is a large array of words & bytes ranging in size from hundreds to thousand to billions

Memory management

In multiprogramming computers the OS resides in a part of memory and rest is used by multiple process as.

the task of subdividing the memory among different processes is called memory management. Memory management is a method in os to manage operation bet main memory and disk during process execution. The main aim of memory management is to achieve efficient utilization of the memory.

* Allocation & deallocation the memory b/w and after the process execution
* To keep track of used memory space by the process
* To minimize fragmentation issues
* To proper utilization of main memory
* To maintain data integrity while executing process

Ex:- IBM system / 360 was by, IBM system / 370.

13) what is Endianness? list types and give example

→ Endianness means that the bytes in computer memory a used to represent a number Endianness is classified 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 value

Ex:- The big Endian representation of the integer 123 places the hundreds value in 1st position followed

by first value and then other value

12 3 2 [12 3]

Little Endian

Little endian stores the least significant value first followed by increasingly more significant value.

Ex: The number 123 in little endian is stored as

123 = [3 2 1]

14) write a program to find the endianess of a given number

```
#include <stdio.h>
int main()
{
  unsigned int x = 0x76543210;
    char *c = (char*);
    if (*c == 0x10)
    {
      printf ("It is Little Endian \n");
      }
else
  {
  printf ("It is big Endian \n");
  }
return 0;
}
```

15) Explain bit, byte, nibble, halfword, word

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

* **Byte** :- A Byte is a combination of 8 bits, it represents character and is called a byte

* **nibble** :- A nibble is a combination of 4 bits, otherwise nibble is half of 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 bit 64 bits depending on the computer 16 is known as opened word

Length	Name	example
1	bit	0
4	nibble	0110
8	Byte	01100110
16	Halfword	0110101011001111
32	word	11001100111100001111010110101001111

16) Explain word allign and halfword allign in memory

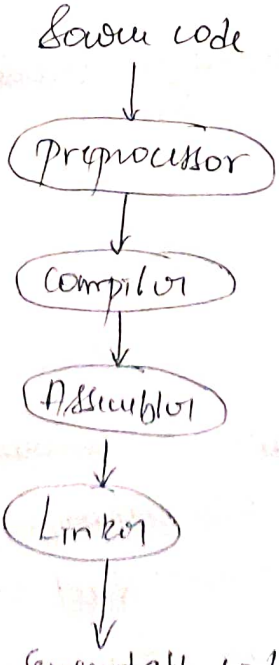
→ Different processor have different definition of words for 32-bit processor, a word is 32 bit. As the same implies a half word is 16 bit for a 16 bit processor a word is 16 bit (2 bytes) for 8 bit processor word in 8 bits.

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

Half word alignment :- that is stored address are adjacent & divisible by 2 that is the last bit is 0

ARM architecture require 32 bit ARM instruction that must be word allgned & stored in memory and 16 bit thumb instruction require halfword allgned & stored

17) Explain the software tool involved in and processing (source file with a neat diagram



* Normally the program building process involves four stages and utilizes different tools such as a preprocessor compiler, assembler & linker
* As far as there should be a single executable file
* preprocessing :- It is the first pass of any compilation it process includes files, condition compilation instruction & macros.
* compiler :- is the second pass, it takes the o/p of the preprocessor and the source code, and generates assembler source code
* Assembly :- is the third stage of compilation it takes assembly source code & produces an assembly listing with offset. 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.

(18) Explain the following addressing modes in ARM

a) three address
b) two address
c) one address

one address instructions

This uses an implied accumulator register for data manipulation and the other in the register(s) 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

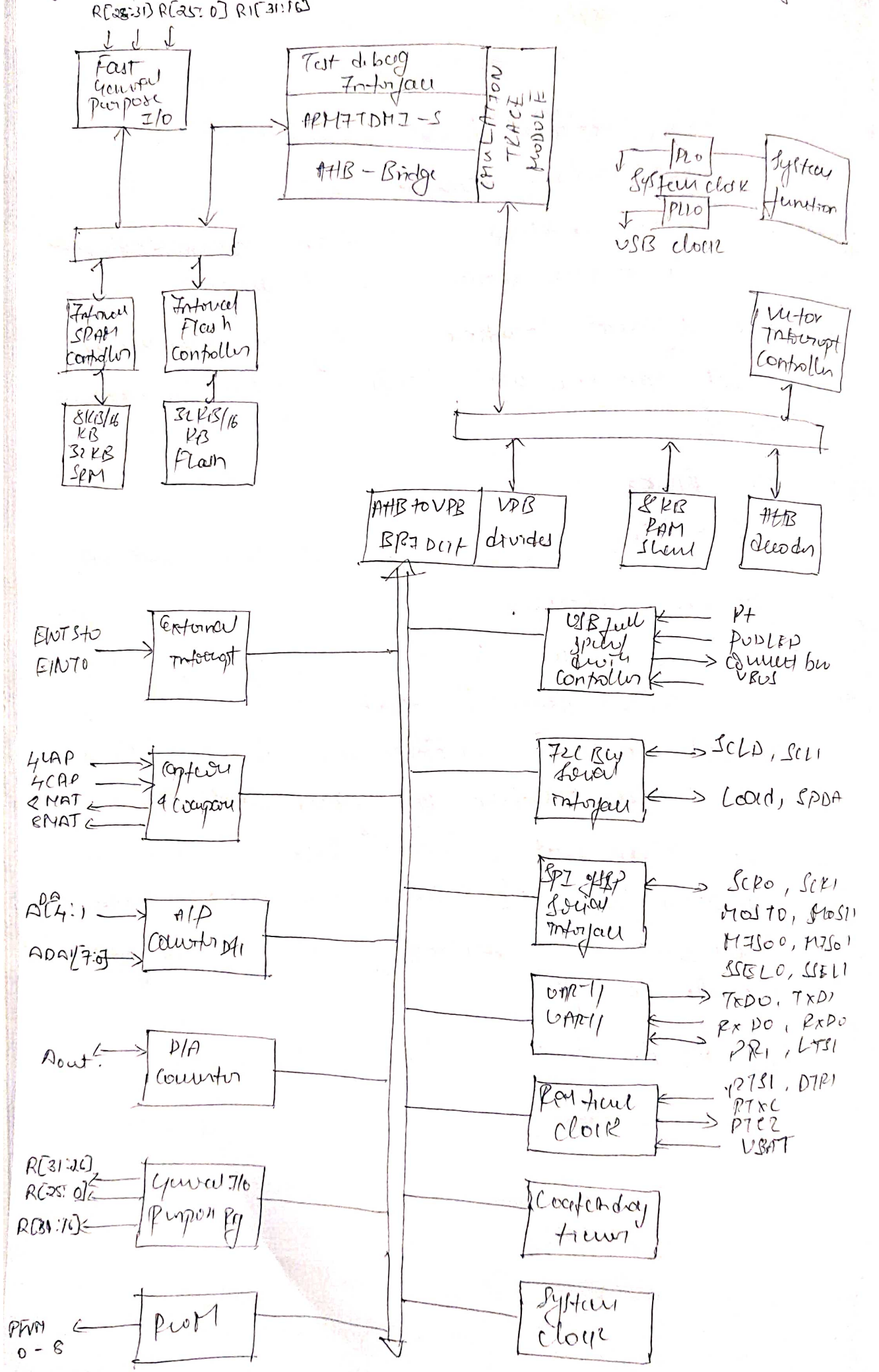
$R_1 \leftarrow [R_2]$

three address instructions

This has three address field to specify register (or) memory location program created are much short in size but no of bit per execution increases. This instruction make duration of program much easier, but it does not mean that program have been much faster because how instruction only contain more information but each microprogram have be performed in one cycle only

Ex: ADD R3, R2, R1 $R_3 = R_1 + R_2$

19) Explain the LPC 2148 microcontroller Block diagram



LPC 2148 Features

* 16 bit/ 32 bit ARM 7TDMI - S microcontroller
* 8 to 40 KB of on chip Static RAM
* 32 KB to 512 KB of on chip flash memory
* 128 bits wide interface/ accelerator enables high speed operation
* USB 2-0 full speed device controller with 2KB of endpoint RAM
* Single 10 bit DAC provides variable analog output
* Two 32 bit timer/ external event location PWM unit & watchdog timer.
* Low power Real time clock (RTC) with independent power 32 KHz clock input
* upto 21 external interrupt pins available
* the on-chip integrated oscillator operates with an external crystal from 1 MHz to 25 MHz
* Single power supply with POR, BOD circuitry CPU operating voltage range of 3.0 V to 3.6V

10) Explain the LPC2148 Microcontroller GPIO pins

LPC2148 has two 32 bit general purpose I/O ports

* PORT 0
* PORT 1

Port 0

* Out of 32 pins, 28 pins can be configured either general purpose I/O.
* Rest of the 32 pins can be configured as o/p only (P0.31)

Port 1

* It is also a 32 bit port
* Out of only 16 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
* They are used to select & configure specific pin functionality

PINSEL 0:- It is used to configure PORT 0 pins from P0.0 to P0.15

PINSEL 1:- It is used to configure pins from PORT P0.16 to P0.32

PINSEL 2:- It is used to configure pins mport 2 from P0.16 to P0.32

Slow GPIO Register

* IOxPIN
* IOxSET
* IOxDIR
* IOxCLR

IOxPIN

=> 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 P0.4 using IOPIN

IOBPIN = IO0PIN | (1<<4)

IOxSET

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

-> writing 1 to respective pin makes high, writing zero has no effect.

IOxDIR

-> 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.

* IOxCLR

→ This is a 32-bit write Register

→ this reg is used to make pins of port low

→ writing 1 to specific bit make that pin low, writing zero has no effects

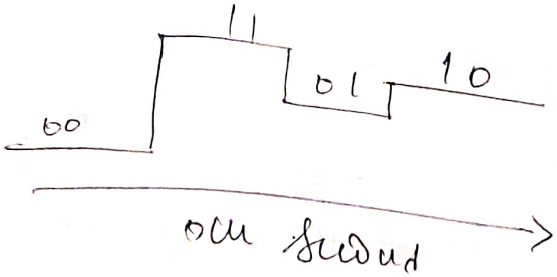
Ex: Configure pin P0.4 as o/p then set the pin to low

```
IO0DIR = 0x0000 0010;
IO0CLR = (1<<4);
```

21) With a neat diagram explain Baudrate and bit rate Explain the calculations.

Baudrate and Bit rate

* Number of times line changed per second
* Let Baud rate be y
* Let bits per line change be 2
* Bit rate = 8 bits per second
* Bit rate = x 2 baud rate in Example



* Baud rate defines the switching speed of a signal
* Baud rate denotes the rate at which information flows across a data link measured in bit/second

* For binary two-level signal data rate/bit rate is equivalent to baud

1 bit = 1 Symbol

1000 = Baud rate

Bit rate = 1000 × 4 = 4000 bps

Formula $b = s \times n$

b = Data Rate (bit per second)

s = Symbol rate (Symbol / sec)

n = number of bits per second

If n = 1 Baud rate = Bit rate

n = 4 Bit rate = 4 × Baud rate

(22) With neat diagram explain the working principle of SPI protocol

* The serial peripheral interface (SPI) is a synchronous serial communication interfacing specification used for short distance communication primarily in embedded systems

* SPI device communication in full duplex using a master slave architecture usually with the single master. The master device originates the frame for reading and writing multiple slave may be existed through selection with individual chip select lines called slave select.

* SPI is called a four wire serial bus controlling with three, two, or one wire serial bus. In SPI only is accurately described as synchronous serial interface

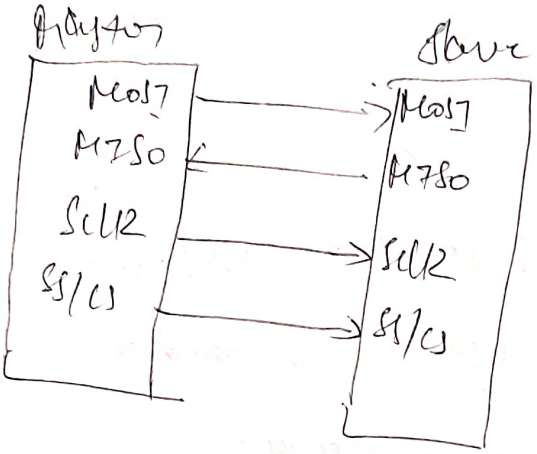
The SPI bus specifies four logic signals

SCLK :- Serial clock (o/p from master)

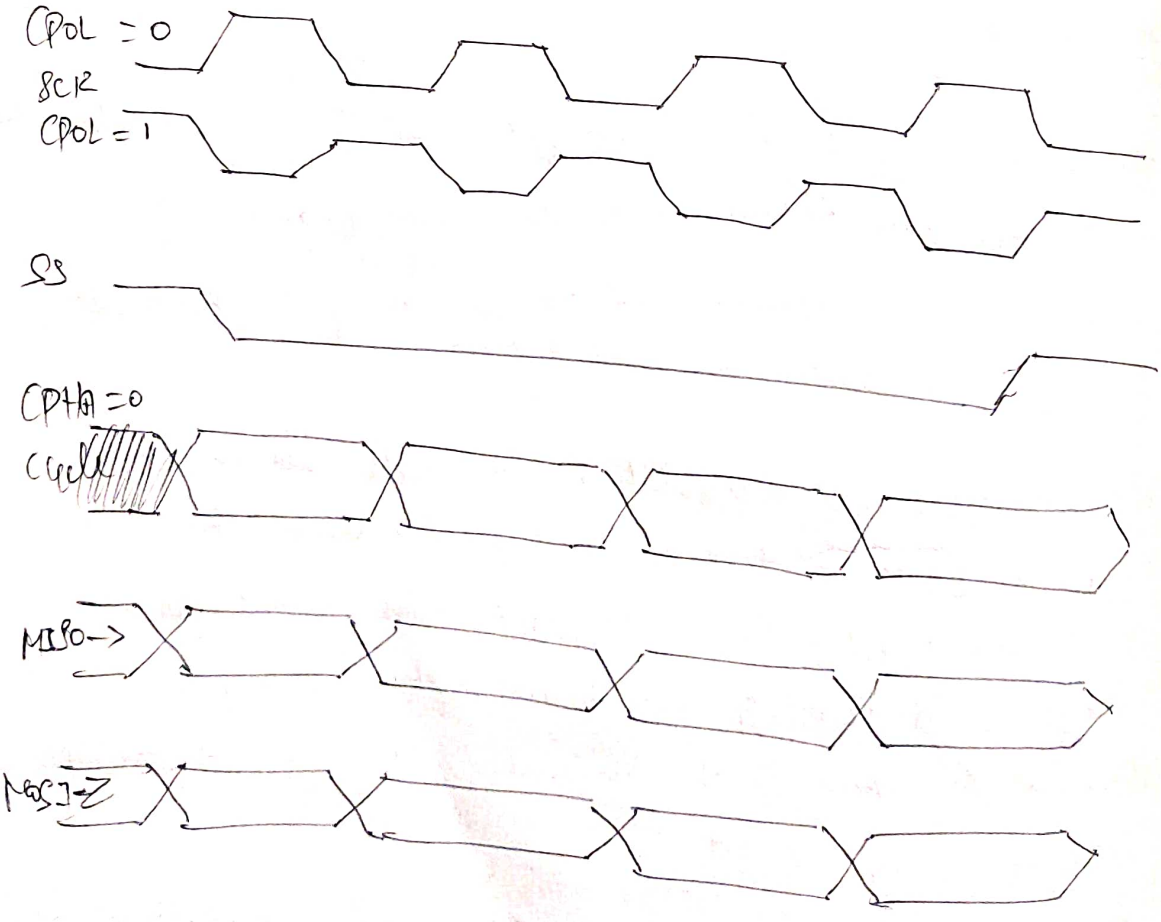
MOSI :- Master out slave in (data from master)

MISO :- Master in slave out (data o/p from slave)

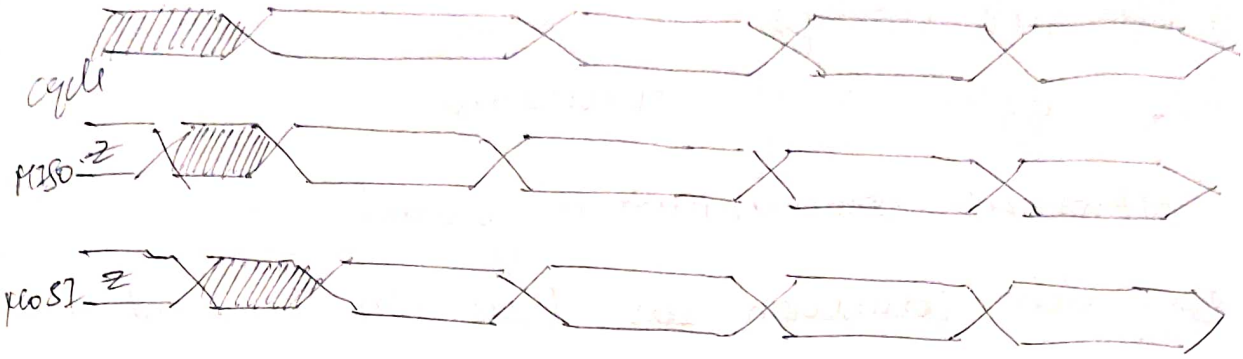
CS/SS :- chip / slave select



23) With neat timing diagram explain CPHA & CPOL usage in SPI



CPHA = 1



CPOL determines the polarity of the clock the processor can be counted with a simple instruction

CPOL = 0 is a clock which idle 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

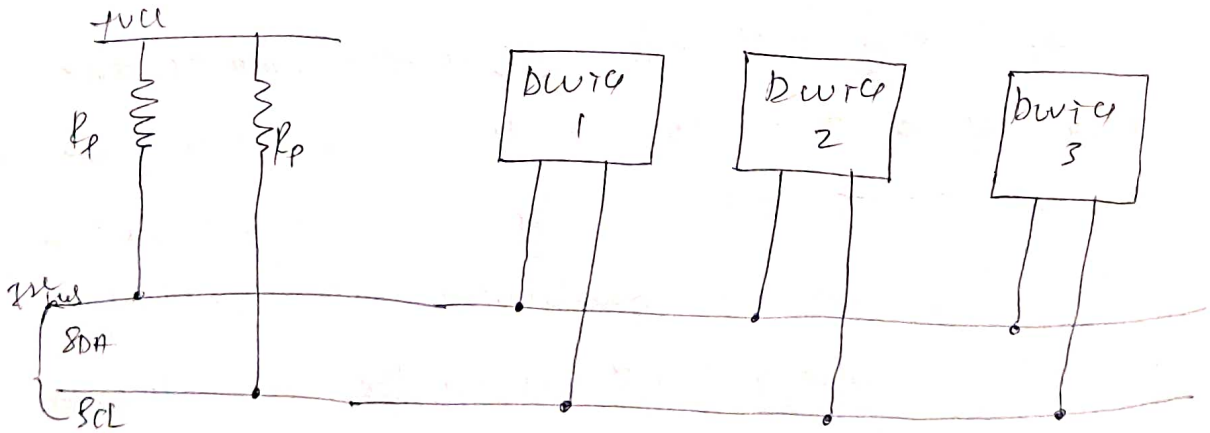
CPOL = 1 is a clock which idle at L and each cycle consists of a pulse of 0 that is leading edge is a falling Edge and trailing Edge is the rising Edge.

For CPHA = 1 the out side changes the data on the leading Edge of the clock cycle, which inside captured the data on the trailing Edge If the clock cycle the outside holds the data ~~maintained~~ until the leading Edge of the following clock cycle for the last cycle, the same holds the MISO line valid until the slave select is deserted An alternative copy of considering it is to say that a CPHA = 1 consists of the half cycle with clock asserted followed by half cycle with the clock idle.

2) With the neat diagram Explain the features of I2C & its working

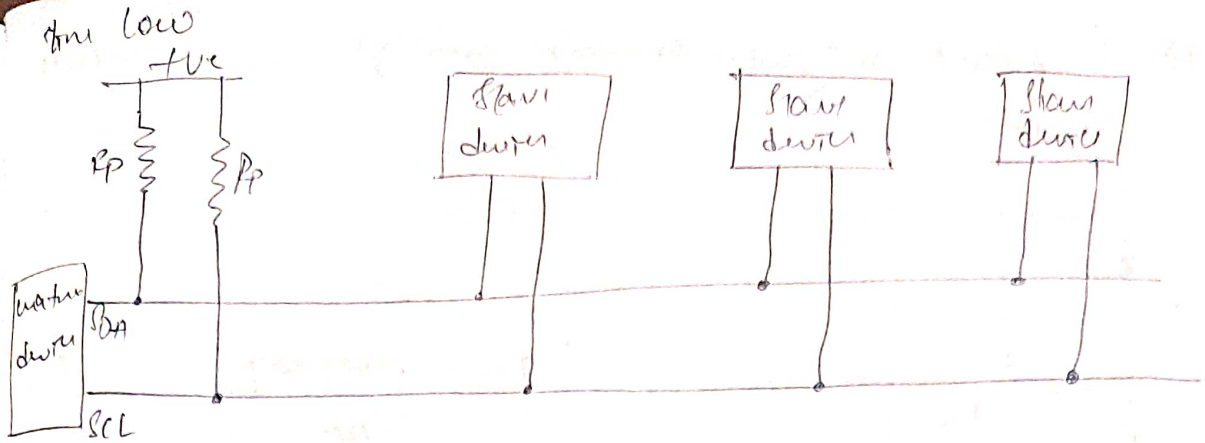
→ * Half duplex serial communication
* Synchronous communication protocol
* Only two communication bus lines are required to control any device / IC on I2C Nw
* Data transfer speed can be adjusted bw required
* Simple mechanism for validation of data transfered

I2C bus consists of two bi-directional clock com and serial data line the data to be transfered is sent through the SDA line synchronised with the clock signal from SCL. All the devices / IC on the I2C Nw are connected to same SCL / SDA lines.



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

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



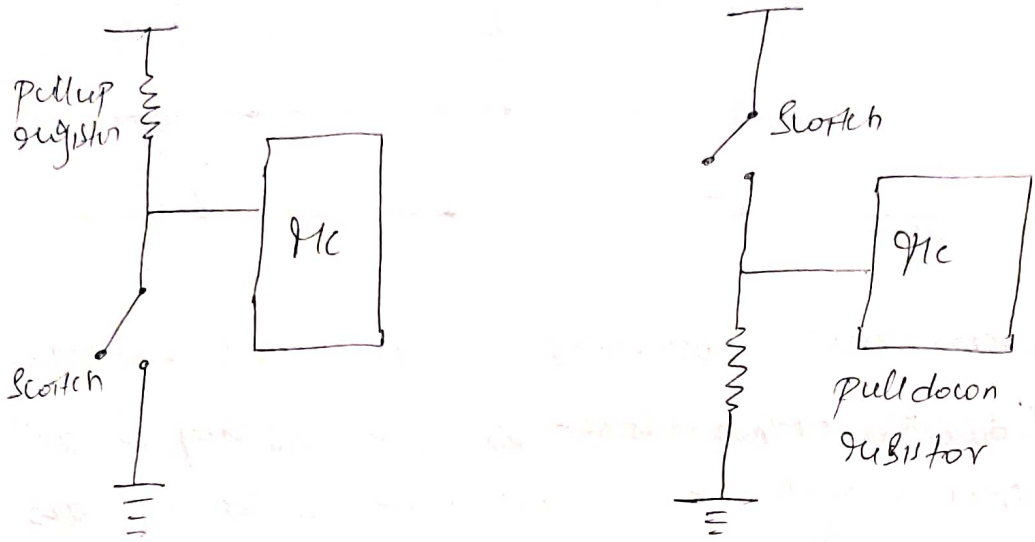
The devices are connected to the I2C bus are categorized as either master or slave. At any instant of time only a single master stays active on the I2C bus. It controls the SCL clock line & decides what operation is to be done on the SDA data line.

All the devices that responds to instruction from this master devices are slaves. At any instant of time only a single master stays active & decides what operation is to be done on the SDA data line.

All the devices that respond to instructions from this master devices are slaves for different bus multiple slave device connected to the same I2C bus, each slave device is physically

When a master device wants to transfer the data or from a slave device it specifies this particular slave device acting on the SDA line & then proceed with the same transfer. So effectively communication takes place b/w the master & a particular slave.

25) With Neat diagram explain pullup and pulldown resistor



Pullup resistor

* A pull up resistor is used to establish an additional loop over the critical components while ensuring sure that the voltage is well defined even when the switch is open
* It is used to ensure that a switch is pulled to high logical level in the absence of an input signal. Pull up resistor with a fixed value coad used to connect the I/p supply and a particular pin in digital circuit.

Pull down resistor

* pull down resistor is used to ensure that I/p to the logic systems still at expected logic level when the external devices are disconnected or high impedance
* It ensures that the line is at a defined LOW logic level. when there are no active connections with other device. the pull down resistor holds the logic signal down to zero volts when no other active device is connected.

26) with neat diagram Explain the concept of arbitration in I2C

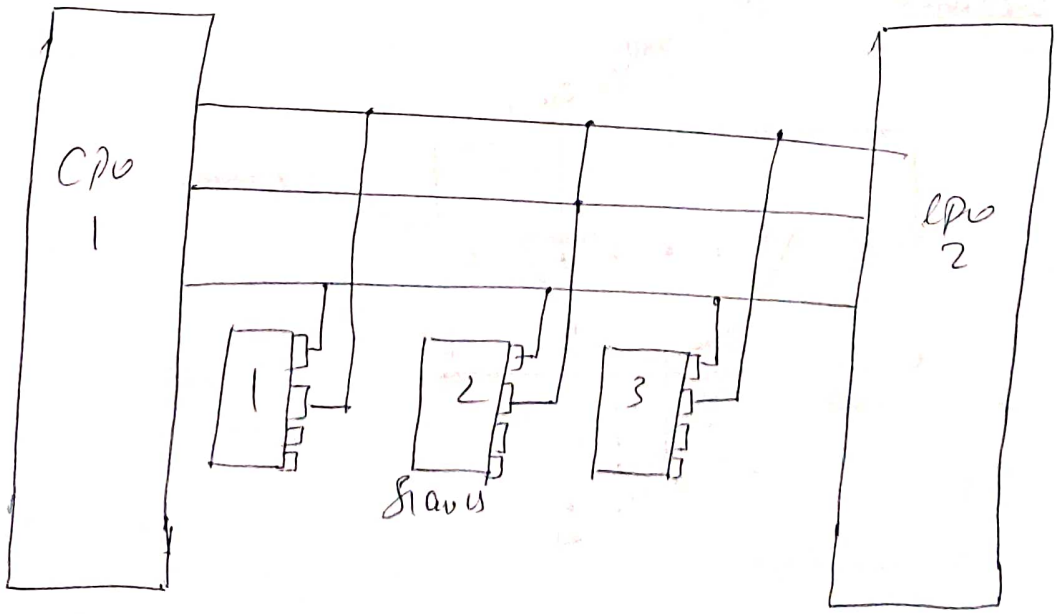
⇒ * I2C is designed for multi master purpose this means that more than one device can initiate transfer

* Bus arbitration occurs when two (or) more master start a transfer at the same time

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

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

when MCU1 issues a start condition & finds the address all slave with listen if the address does not match the address of CPU2, this device has to hold back any activity until the bus breaks with after a stop condition As long as two MCU's monitor what is going on the bus and as long as they can see that a transaction going on because that to12 should command was not a stop there is no problem.

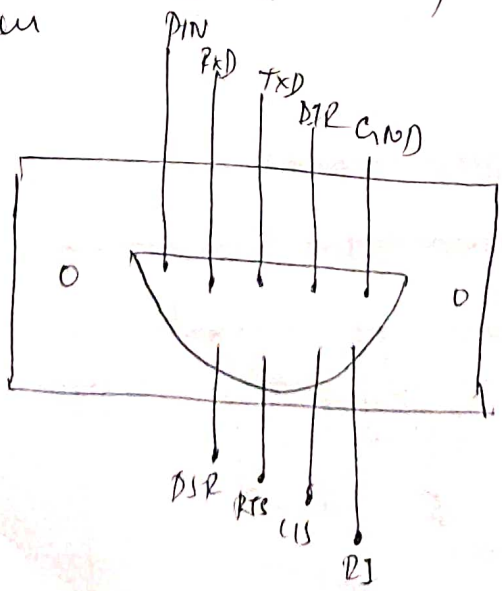


27) What is clock stretching? Explain clock stretching in I2C

→ Clock stretching allows an I2C slave device to force the master device into a wait state a slave device may perform clock stretching when it needs more time to manage data such as storing data or prepare the transmit another byte of data. Clock stretching in I2C device can slow down communication by stretching SCL. during an SCL low phase any I2C device on the bus may additionally hold down SCL to prevent it to rise again enabling enabling them to slow down the SCL clock rate or to stop I2C communication for a while then it also respond to an clock synchronization.

In an I2C communication the master device determines the clock speed while during master and slave from synchronization exactly to predefined baud rate

28) Explain the working of DB 9 pins and handshaking with the modem

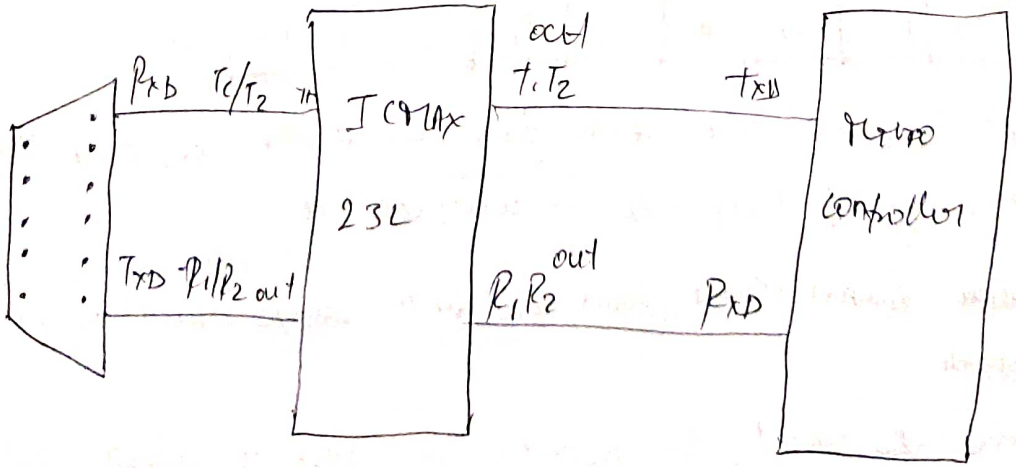


Pin	Signal	Signal name	DTE Signal direction
1	DCD	Data carrier detect	In
2	RXD	Receive data	In
3	TXD	Transmit data	out
4	DTR	data terminal Ready	out
5	GND	Ground	—
6	DSR	Data set ready	In
7	RTS	Request to send	out
8	CTS	Clear to send	In
9	RI	Ring indicator	In

Handshaking Protocol

A modem handshake is what occurs when the receiving modem answers the phone call and the two modems begin to communicate. Before anything happens the modem must evaluate the quality of line negotiate used control protocol and data compression that they can both recognise and decide what the most suitable connection speed should be based on the condition. Speed this process is called a handshake.

29) Explain RS232 connection with microcontroller



several devices collects 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 is fast and when more number of bits send communication on the other hand has only one or two data line transfer and its generally used for long distance communication. In serial communication the data is sent as a one bit at a time

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

3a) Explain the frame format in UART communication.

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

Data framing: UART transmit 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	DB	Stop

the UART receives the data from the data bus and the data are being sent by CPU memory or file

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 is a data packet.

This data packet is serially transmitted to the receiving UART by the transmitter the receiving pin of the receiving UART reads the data bit by bit

Start bit:

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

Data frame:

The data bits are usually 8 to 8 in number if no parity bit is used, it can be 9 bit long in general case the LSB of the data is transmitted first

parity bit:

The parity bit is used to indicate the change in data during transmission the reason for the change in the data is mismatched baud rate, Electromagnetism or 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

3) Explain the difference between in detail
(a) Serial v/s parallel
(b) Analog v/s digital
(c) 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 links is not an easy task	* parallel data is very accurate Easily implemented as hardware
* No crosstalk problem	* cross talk occurs the interference blw parallel lines
* Bandwidth of serial com is much higher	* The bandwidth of serial wires is much lower

Analog	Digital
* Transmitted modulated signal is analog in nature	* Transmitted signal is digital i.e form of digital pulse
* Amplitude frequency or phase variation in the transmitted signal represent message	* Amplitude width of transmitted pulse is transmitted pulses is transmitted in the form of code words
* Noise immunity is poor for AM & PM	* Noise immunity is Excellent
* FDM is used for multiplexing	* TDM is used for multiplexing
* Analog modulation system are AM, FM, PM, PAM, DWM	* Digital modulation system are PCM, DM, ADM, DPCM.

Synchronous	Asynchronous
* communicated in real time	* communicated in the real time
* Continuously interrupts in a working	* Eliminates interrupts
* Sends the data and receives the data on the same clock frequency.	* Sends the data & receives the data on different clock frequency.
* Faster	* Slower
* There is no overhead of Extra Start & Stop bit	* user's Start and Stop bit
* user constant time interval	* uses random & irregular time interval
* used in chat rooms, video conferencing	* Used in email.