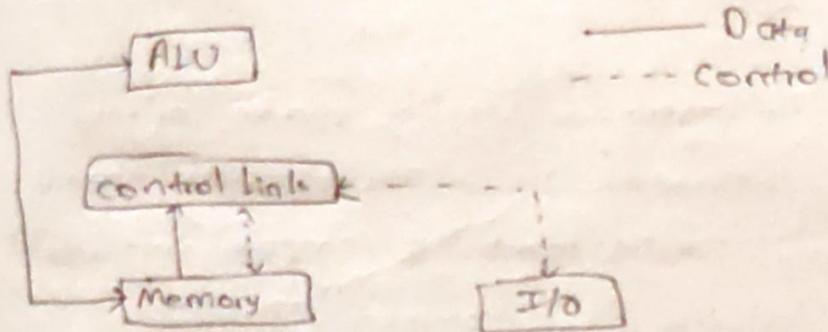


ASSIGNMENT - I① Von-Neuman Architecture

All parts are controlled by control unit CPU and connected together by Bus. Data can pass through bus in half Duplex mode to and from CPU. Memory holds programs & data known as stored program concept. Memory is split to small cells with the same size then original numbers are called address number.

Advantages

Control unit gets data & instruction in the same way from one memory. It simplifies design & development of the control unit.

Disadvantages

- Serial instruction processing does not allow parallel execution. Parallel execution are simulated later by the O.S.
- One Bus is a Bottleneck.
- Instructions are stored in same memory as the data can be accidentally rewritten by an error in a program.

## ② CISC

CISC was developed to make compiler development easier & simple. CISC is complex. Instruction set computer.

## RISC

RISC is designed to perform a smaller number of and types of computer instructions. It is micro processor that is designed to perform smaller number of computer instructions. RISC is reduced instruction set computer.

### CISC

- More set of instructions
- More addressing mode
- Minimum amount of RAM
- Focus is on Hardware.
- High power consumption
- CISC has variable instruction format
- Single register set
- Less pipelined
- Lesser set of instructions
- Lesser addressing mode
- More amount of RAM
- Focus is on Software to optimise.
- Low power consumption.
- RISC has fixed instruction format.
- Multiple register set.
- Highly pipelined.

### RISC

⑧ There are four main types of number systems

- \* Binary Number System (Base-2)
- \* Octal Number System (Base-8)
- \* Decimal Number System (Base-10)
- \* Hexadecimal Number System (Base-16)

The Binary number system is used to store the data in computer.

The advantage of the octal number system is that it has fewer digits when compared to several other systems, hence there would be fewer computation errors.

The decimal number system is the system that we can use in daily life.

The hexadecimal number system is used in computer to reduce the large sized strings of the binary system.

Number Systems helps in representing the numbers in a small symbol set. Binary numbers are mostly used in computer that use digits like 0 and 1 for calculating simple problems. The number systems also help in converting one number system to another.

Conversion from one number system to another number system

To convert a number from one of the binary/octal/hexadecimal system to one of the other systems, we first convert it into decimal system & then we convert it to the required systems by using the below process.

Eg:  $1010\ 111100_2$ , to decimal

$$\begin{array}{ccccccc}
 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
 & | & | & | & | & | & | & | & | & | \\
 & 2^0 & 2^1 & 2^2 & 2^3 & 2^4 & 2^5 & 2^6 & 2^7 & 2^8 & 2^9 \\
 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
 & 2^0 \times 1 = 0 & 2^1 \times 0 = 0 & 2^2 \times 1 = 4 & 2^3 \times 1 = 8 & 2^4 \times 1 = 16 & 2^5 \times 1 = 32 & 2^6 \times 0 = 0 & 2^7 \times 0 = 0 & 2^8 \times 0 = 0 & 2^9 \times 1 = \frac{512}{700}
 \end{array}$$

$$1010\ 111100_2 = 700_{10}$$

$$\begin{array}{r}
 16 \overline{)700} \\
 16 \overline{)43} \\
 \underline{-12} \quad 1 \\
 \underline{2} \quad -11 \\
 \underline{\underline{B}}
 \end{array}$$

$$700_{10} = 2BC_{(16)}$$

$$\begin{array}{r}
 8 \overline{)700} \\
 8 \overline{)87} \\
 \underline{-4} \\
 8 \overline{)10} \\
 \underline{-8} \\
 1 \quad -2
 \end{array}
 = (1274)_8$$

$$\begin{array}{r}
 2 \overline{)700} \\
 2 \overline{)350} \\
 \underline{-6} \\
 2 \overline{)175} \\
 2 \overline{)87} \\
 \underline{-1} \\
 2 \overline{)43} \\
 2 \overline{)21} \\
 \underline{-1} \\
 2 \overline{)10} \\
 \underline{-8} \\
 2 \overline{)5} \\
 2 \overline{)2} \\
 \underline{-1} \\
 1 \quad -0
 \end{array}
 = 1010\ 111100_2$$

⑤ Arithmetic operators are used to perform mathematical calculations. Logical operators are used to perform logical operations & include AND, OR, NOT. Boolean operators include AND, OR, XOR or NOT and can have one of two values true or false.

The arithmetic operators are addition (+) subtraction (-) Multiplication (\*) & division (/) exponential (^).

The 3 common three logical operators are

- \* AND
- \* OR
- \* NOT

The logical operators are often used to help create a test expression that controls program flow this type of expression is also known as Boolean expression because they create a boolean answer or value when evaluated.

### Truth table

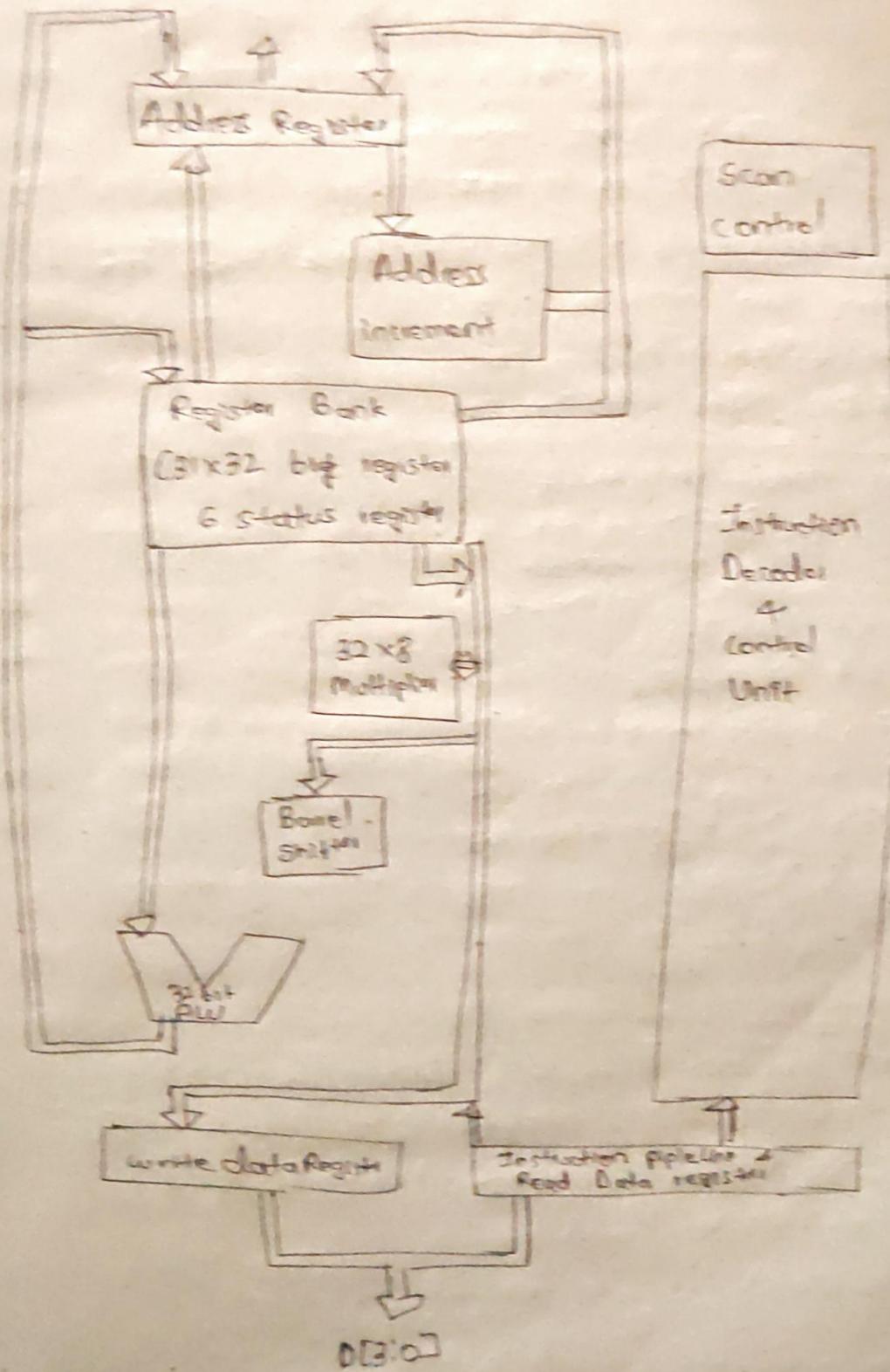
The common way to show logical relationships is in truth table.

| x     | y     | x and y | x or y |
|-------|-------|---------|--------|
| false | false | false   | false  |
| false | true  | false   | true   |
| true  | false | false   | true   |
| true  | true  | true    | true   |

| X     | Not X |
|-------|-------|
| false | true  |
| true  | false |

| operator | Meaning     | Example           | Result |
|----------|-------------|-------------------|--------|
| and      | Logical AND | (true) and (true) | false  |
|          | Logical OR  | (true)    (false) | true   |
| !        | Logical NOT | !(true)           | false  |

③ Write a neat diagram, explain the PRMF Architecture.



## Features used

- Load / store Architecture
- Fixed Length 32 bit Instructions
- 3 address instruction formats.

ARM processor is a 32 bit architecture most ARM's implement two instruction sets

- 32 bit ARM instruction set.
- 16 bit thumb instruction set.
- Von-Neuman Architecture.
- 3 stage Pipeline → fetch
  - decode
  - execute
- 32 bit Data bus
- 32 bit Address Bus.
- 37 32 bit register
- 32 bit ARM instruction set.
- 16 bit thumb instruction set.
- $32 \times 8$  Multiplier.
- Barrel Shifter.

## Data types

ARM processor supports 6 data types

- 8 bit Signed & Unsigned bytes
- 16 bit Signed & Unsigned half word aligned.
- on 2 byte boundaries.
- 32 bit Signed and Unsigned half words

ARM instructions are all 32 bits words word aligned  
thumb instruction are half words aligned on 2 byte boundaries.

Internally all ARM operations are on 32-bit operands  
the shorter data type are only supported by data  
transfer instructions. When a byte is loaded from memory  
it is zero or signed extended to 32-bit.

ARM Co-processor supports floating point values.

#### ④ Programming model for ARM7

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 operating modes.
- Modes changes by software control or external interrupts.

| CPSR[4:0] | Mode   | Use                            | Reg# |
|-----------|--------|--------------------------------|------|
| 00000     | USR    | Normal user code               | USR  |
| 00001     | FIQ    | processing fast interrupt      | FIQ  |
| 00100     | SVC    | processing slow interrupt      | SVC  |
| 00111     | SVC    | processing software interrupt  | -SVC |
| 01000     | Abar   | processing memory fault        | -abt |
| 10111     | Undef  | Handling undefined instruction | -und |
| 11111     | System | Running privileged OS          | User |

- Most programs operate in user mode. ARM has other privileged operating modes which are used to handle exceptions, supervisor calls & system mode.
- More access rights to memory systems + co-processors.
- Current operating mode is defined by CPSR[11:0].

### Privileged modes

#### Supervisor modes:-

- Having some protective privileges.
- System Level functions can be accessed through specified supervisor calls.
- Usually implemented by software interrupt.

ARM has 37 registers all of which are 32 bits

- Program Counter.
- Current program status register.
- 5 dedicated saved program status registers.
- 30 general purpose registers.

#### Each mode can access

- its particular set of R0 - R12 registers.
- stack pointer, Run-Link register.
- Program counter (PC)
- the current program status register (CPSR)

Privileged modes can access a particular SPSR saved program status register.

Ques

|      |
|------|
| 10   |
| 11   |
| 12   |
| 13   |
| 14   |
| 15   |
| 16   |
| 17   |
| 18   |
| P0   |
| R10  |
| R11  |
| R12  |
| R13  |
| R14  |
| R15  |
| CPSR |

|          | FIQ      | IRQ      | SVC      | Undef    | About    |
|----------|----------|----------|----------|----------|----------|
| r8       |          |          |          |          |          |
| r9       |          |          |          |          |          |
| r10      |          |          |          |          |          |
| r11      |          |          |          |          |          |
| r12      |          |          |          |          |          |
| r13 (SP) |
| r14 (LR) |
| r15 (PC) |          |          |          |          |          |
| SPSR     | SPSR     | SPSR     | SPSR     | SPSR     | SPSR     |

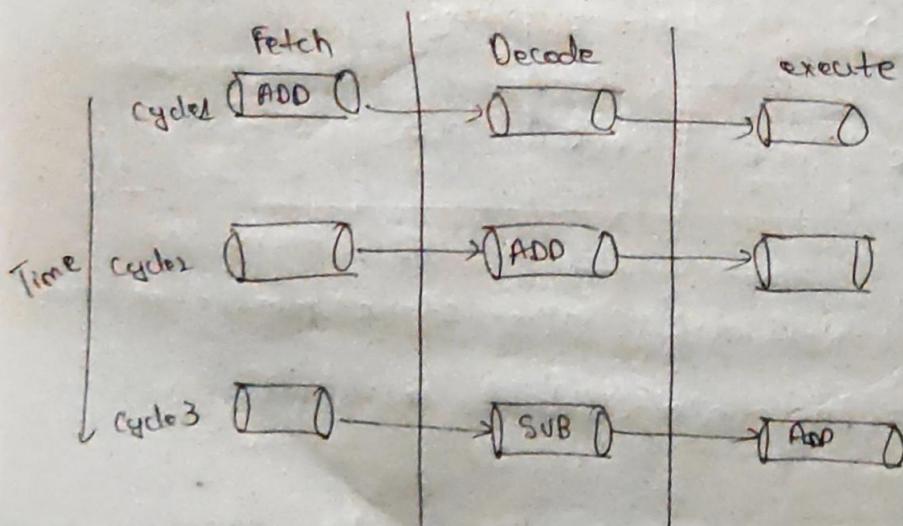
- ⑥ With a neat diagram, explain three stage pipeline of ARM
- A pipeline is the mechanism used by RISC processor to execute instruction.
  - By speeding up the execution by the fetching the instruction while other instructions are being decoded & executed simultaneously.
  - pipelining is a design technique or process which plays an important role in increasing the efficiency of data processing in the process of a computer & etc.

The ARM7 has three stage pipeline:

- Fetch: The instruction is fetched from memory.
- Decode: The instruction is fetched from memory. The instruction's opcode & operands are decoded to determine what function to perform.
- Execute: The decoded instruction is executed.

Each of three operations requires one clock cycle for 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 executes completed in every clock cycle. In other words the pipeline has a throughput of one instruction per cycle.



⑦ With the neat diagram, explain CPSR register.

|   |   |   |   |   |   |   |   |   |   |        |
|---|---|---|---|---|---|---|---|---|---|--------|
| N | Z | C | V | . | . | - | I | F | T | (mode) |
|---|---|---|---|---|---|---|---|---|---|--------|

[Conditional flag]

Condition code flags

- N: Negative result from ALU
- Z: zero result from ALU.
- V: ALU operation overflowed
- C: ALU operation carried out

Sticky overflow flag - S flag

- Architecture STE only
- indicates if saturation has occurred during certain operations

In interrupt Disable bits:-

- I = 1 Disables the IRQ
- F = 0 disables the FIQ

T bit

→ Architecture T only

→ T = 0 processor in ARM state.

T = 1 processor in thumb state

Mode Bits

→ Specify the processor mode.

CPSR → Current processor status register holds the information about the current status of the 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.

### Q) Explain the seven different modes in ARM.

The ARM7TDMI processor has seven modes of operation.

- User mode is the usual ARM program execute on state and is used for executing most application programs.
- Fast interrupt (FIQ) mode supports a data transfer or channel process.
- Interrupt (IRQ) 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 entered after iso 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 mode. Privileged modes are used to service interrupts or exceptions or to access protected resources.

| Mode           | Mode Identifier |
|----------------|-----------------|
| User           | user            |
| Fast Interrupt | Fin             |
| Interrupt      | Irq             |
| Supervisor     | Svc             |
| Abort          | abt             |
| System         | sys             |
| undefined      | und             |

### Q) Explain the nomenclature in ARM

ARM was originally from Acorn computer Ltd first RISC processor for commercial use.

ARM7/D7 Processors

32 bit Processor Advanced machine

T → thumb Architecture extension.

D → Debug extension

M → Enhanced extension

I → In circuit Emulation

ARM {x3-F4} - {z3 TDMI {E3 {S3 {F3 {S3

x → series

y → Memory management Unit

z → cache

T → thumb 16 bit decoder

D → JTAG Debugger.

M → Fast Multiplier.

I → Embedded ICE (In circuit Emulator)

E → Enhanced instruction for DSP

J → JAVA acceleration of JavaCard.

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's a strong chance that we have used JTAG or one of the related standards.

JTAG is Joint Test Action Group is an industry standard for verifying designs & testing printed circuit boards after manufacture.

JTAG is implements standards for on chip instruction is electronics design automation (EDA) as a complement tool to digital simulation. It specifies the use of a dedicated debug port <sup>implementing serial comm interface</sup> for low-overhead access without to the system address by direct buses. The interface connects to an chip. Test access port (TAP) that implements a standard protocol to access a set of test registers.

test logic reset

run test idle

select DR scan

capture DR

shift DR

exit 1DR

pause DR

exit 2DR

update DR

select IR scan

capture IR

shift IR

exit 1IR

pause IR

exit 2IR

update IR

⑪ What is single tasking? Give examples of process or application

Single tasking means doing one task at a time with as little disruption & interruption as possible.

Micro controller are known as computer on chip. They are designed to perform a single task only because its processing power as well as memory is not suitable for installing an O.S.

⑫ What is MMU? why MMU is required? Give ex of MMU support.

The memory can be defined as a collection of data in a specific format. It is used to store instructions & processed data. The memory comprises a large array of group of words or bytes, each with own location. The primary motive of a computer system is to execute programs. These programs along with the information by access, should be in the main memory during execution. The CPU fetches the instructions from memory according to the value of the program counter. The main memory is central to the operation of a computer. Main memory is a large array words & bytes, ranging in size from hundreds to thousands to billions. Main memory is a repository of rapidly available information shared by the CPU.

and I/O devices. Main memory is the place where programs & information are kept when the processor is effectively utilizing them. Main memory is associated with the processor is extremely fast.

Main memory is also known as RAM. This memory is a volatile memory. RAM loses its content a power interruption occurs.

### Memory Management

In a multiprogramming computer the operating system resides in a part of memory and rest is used by multiple processes. The task of subdividing the memory among different processes is called memory management. Memory management is a method in O.S. to manage operations b/w main memory & disk memory during process execution. The main aim of memory management is to achieve efficient utilization of memory.

With memory management is required?

- Allocate & de-allocate the memory before & after the process execution.
- To keep track of used memory space by process.
- To minimize fragmentation issues.
- To proper utilization of main memory.
- To maintain data integrity while executing of process.

E6: IBM System/360 model 67, IBM System 1370

### APM

ARM architecture based application processes implement of MMU designed by APM's virtual memory system architecture. The current architecture defines PTE's for describing 4KB and 64KB pages, 1MB sections and 16MB Super sections. Legacy versions also defined 1KB tiny pages.

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

→ #include <stdio.h>

```
int main()
{
    unsigned int x = 0x70543210;
    char *C = (char*)&x;
    if (*C == 0x10)
        {
            printf("underlying architecture is little endian\n");
        }
    else
        {
            printf("underlying architecture is big endian\n");
        }
    return 0;
}
```

(15) Explain following

- Bit: A bit is the smallest unit of information that can be stored in a computer. Bits in computer are grouped to form a larger unit of information.
- Byte: A byte is a combination of eight bits. Eight bits represent a character and is called a byte.
- nibble: A nibble is a combination of four bits, in other words a nibble is half byte.
- word: A word is a combination of 16 bits, 32 bits or 64 bits depending on the computer. As known as quad word.

| Length | Name     | Example            |
|--------|----------|--------------------|
| 1      | Bit      | 0                  |
| 4      | Nibble   | 1011               |
| 8      | byte     | 1011 0101          |
| 16     | Halfword | 1011 0101 1001 001 |

(16) Explain the word align & Half word align in ARM memory

Different processors have different definition of word. In a 32-bit processor, a word is 32-bit (4 bytes). As the name implies, a half word is 16-bit for a 16-bit processor, a word is 16-bit (2 bytes). In a 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 the stored address are adjacent & divisible by 2 , that is the last bit is 0.

ARM architecture requires 32 bit ARM instructions that must be word aligned & stored in memory & 16 bit thumb instructions requires half word aligned & stored. Therefore in ARM state the value of R15 is always divisible by 4 , that is lowest 2 bits of the R15 register are always 00. In the thumb state the value of R15 is always divisible by 2. which is the lowest bit of the R15 register always 0. One word consists of one or more bytes i.e usually integer bits of bytes.

⑧ Explain the following addressing modes in ARM.

- ④ Three address
- ⑤ Two address
- ⑥ One address instruction using ARM.

Ans Sequence of instructions forms a program to perform a specific tasks.

2 components

- op-code field → specifies how data manipulate
- Address field → specifies the data location

When data to be read from or store in two or more address field may have one or more than one address

- \* Three address Instruction
- \* Two address Instruction
- \* One address Instructions
- \* Zero address Instructions

Processor can execute an instruction only if it is represented in binary sequence.

Unique binary sequence pattern must be assigned. This process is called op-code encoding.

### One address Instructions

This uses an implied Accumulator register for data manipulation and the other is in the register memory location. Implied means that the CPU already knows that one operand is in the accumulator so there is no need to specify it.

Eg: LDR add1  
Acc ← [addr]

### Two address instructions

Here two address can be specified in the instruction. In the one address instruction, the result was stored in the accumulator. Here the result can be stored in different locations, i.e. register or memory locations. But requires more numbers of bit to represent the address.

Eg: MOV R1, R2  
R1 ← [R2]

### Three address Instructions

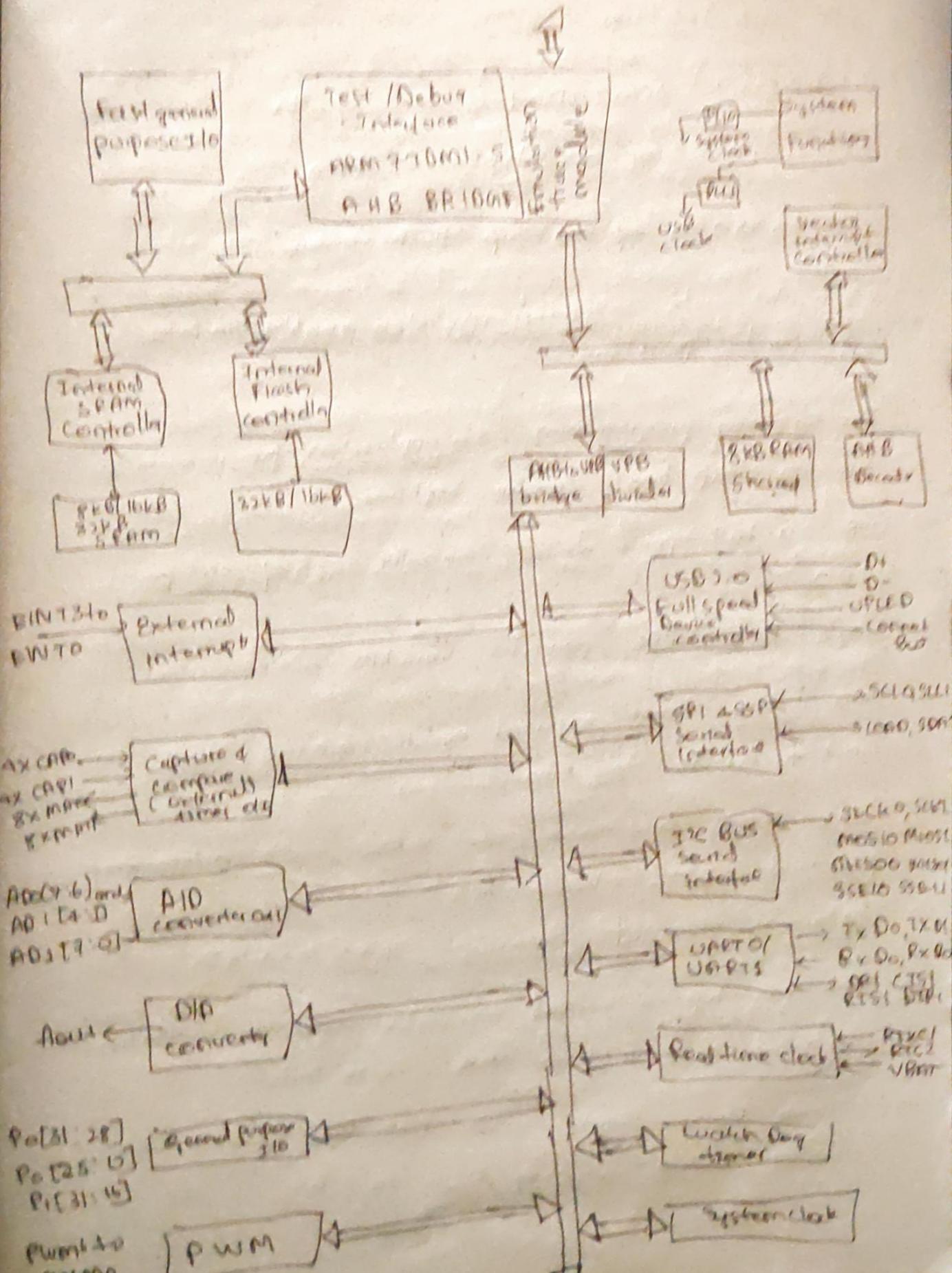
This has three address field to specify a register or memory location. Program created are much short in size but number of bits per instruction increase. Program created are much short in size but number of bits per instructions are increased. These instructions make creation of program much easier but it does not mean that program will run much faster because now instructions only contain more target information but each micro-operation will be performed in one-cycle only.

Eg: ADD R<sub>3</sub>, R<sub>1</sub>, R<sub>2</sub>

$$R_3 = R_1 + R_2$$

⑨

Explain the LPC2148 Microcontroller Block diagram.



## LPC 2148 Features

- 16 bit / 32 bit ARM7TDMI-S microcontroller
- 8 kB to 40 kB of on chip static RAM.
- 32 kB to 8512 kB of on chip flash memory.
- 108 bit bit wide Interface/accelerator enables 60 MHz high speed operations.
- In System programming / In application programmer via on chip boot loader software.
- USB 2.0 Full speed device controller 2 kB of end point RAM.
- Single 10 bit DAC provides variables analog output.
- Two 32 bit timers / external event counters, PWM unit + Watch dog timer.
- Low power Real time clock (RTC) with independent power & 32 MHz clock input.
- Up to 21 external interrupt pins available.
- The on chip integrated oscillator operates within an external systems from 1MHz to 25 MHz.
- Single power supply with POR, BOD circuits CPU operating Voltage range of 2.30V to 3.6V.

Q) Explain the LPC2148 microcontroller GPIO Pins.

A) GPIO → General purpose Input/Output. A 32bit register used to select the functions of pins in which the user needs to operate. There are four bits for each pin of the controller, which the first function is GPIO. It means that the pin can either act as an input or output with no specific function.

There are totally three PIN register in LPC2148 controller in order to control the functions of the pins in the respective parts. The classification is given below.

PIN SEL0 → controls functions of Port 0.0 - Port 0.15  
PIN SEL1 - controls the function of Port 0.16 - Port 0.31  
PIN SEL2 - controls functions of Port 1.0 - Port 0.31

where all of 32bits registers are GPIO configurator.

LPC2148 has two 32-bit general purpose I/O ports. A total of 30 I/O and a single output only pinout of 32 pins are available to GPIO function. PORT0 & PORT1 are controlled via two groups of 4 registers.

|         |         |
|---------|---------|
| * IOPIN | * IODIR |
| * IOSET | * IOCLR |

## IOPIN

This register provides the value of port pins that are configured to perform only digital functions. The register will give the logic value of the pin regardless of whether the pin is configured to input or output as GPIO or an alternate digital functions.

As an example: that particular port pin may have GPIO, input, output, UART receiver, PWM output as selectable functions. Any configuration of that pin will allow its current logic state to be read from the IOPIN register.

If a pin has an analog function as one of its options, the pin state cannot be read if the analog config is selected. While the pin as an A/D input disconnects the digital features of the pin, the pin value read in the IOPIN register is not valid. Writing to the IOPIN register stores the value in the port output register, bypassing the need to use both the IOSET & IOCLR registers to obtain the entire written value. This feature should be used carefully in an application since it affects the entire port.

### IOSET

This register is used to produce a high level output at the port pins configured as GPIO in an output mode. Writing 1 produces a high level at the corresponding port pins. Writing 0 has no effect if any pin is configured as an IIP or a secondary function, writing 1 to the corresponding bit in the IOSET has no effect. Reading the IOSET register returns the value of this register as determined by previous writes to IOSET register return the value of this register as determined by previous write to IOSET + IOCLR. This value does not reflect the effect of any outside world influence on the IO pins.

### IODIR

This word accessible register is used to control the direction of the pins when they are configured as GPIO port pins in output mode. Writing 1 produces a low level at the corresponding port pin + clear the corresponding bit in IOSET register. Writing 0 has no effect. If any pin is configured as an input or a secondary function, writing to IOCLR has no effect.

## IOCR

This register is used to produce a low-level output at port pins configured as GPIO in an output mode writing 1 produces a low level at the corresponding port pin & clears the corresponding bit in IOSBT register. writing 0 has no effect. If any pin is configured as an input or a secondary function, writing to IOCR has no effect.

- Q) write a neat diagram explain Baud rate + Bit rate. Explain the calculations.

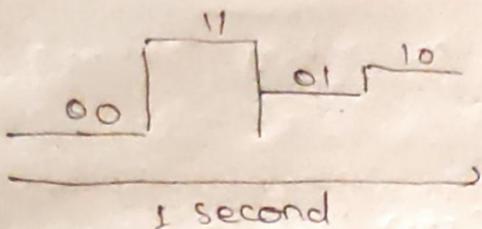
Baud: How many time a signal changes per second

Bit: How many bits can be sent per time unit (usually per second).

Bit rate is controlled by baud and number of signal levels

### Baud rate

- Number of time line changed per second.
- Let Baud rate be 4 (4 changes per second).
- Let bits per line change be 2.
- Bit rate = 8 bits per second.
- Bit rate = x2 Baud rate in this 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 a binary two-level signal a data rate of one bit per second is equivalent to one baud.

An analog signal carries  $n$  bits in each signal unit. If 1000 signal units are sent per second, find the Baud rate & the bit rate.

1 bit  $\rightarrow$  1 symbol

$$\text{Bit rate} = \frac{\text{Baud rate}}{1000}$$

$$1000 = \text{Baud rate}$$

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

If bit rate (or data rate) is "b"

Baud rate (or symbol rate) is "s"

General formula

$$b = s \times n$$

$b$  = Data Rates (bits per second)

$s$  = Symbol rate / symbol/sec

$n$  = Number of bits per second

If  $n=1$ , Baud rate = Bit rate

$n=4$ , Bitrate =  $4 \times$  Baud rate

(72) With a neat diagram, Explain the working features of SPI protocol.

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

The interface specification was developed by Motorola.

SPI devices communicate in full duplex using a master-slave architecture usually single master. The master device originates the frame for reading or writing. Multiple slave device may be supported through selection with indirect chip select (Cs), sometimes called slave select (SS) lines.

SPI is called a four wire serial bus, contrasted with three, two or one wire serial bus. In the SPI may be accurately described as a symmetrical serial interface. But it is different from the synchronous interface (SSI) protocol, which is also a four wire synchronous serial communication protocol.

The SPI Bus specifies four logic signals SCLK:  
Serial Clock (output from master)

MOSI : Master out Slave In (data from master)

MISO : Master in Slave out (data output from slave).

Cs/SS : Chip / slave select.

(Often active Low output from master to indicate the data is being sent).

|        |                |                |       |
|--------|----------------|----------------|-------|
| SPI    | SClk           | SCLK           | SPI   |
| Master | MOSI           | MOSI           | slave |
|        | MISO           | MISO           |       |
|        | S <sub>S</sub> | S <sub>S</sub> |       |

(2) In SPI with a neat timing diagram explain (CPHA, CPOL).

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

CPOL=0 is 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 a falling edge.

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

CPHA determine the timing (i.e) of the data bits relative to the clock pulses. Conversion b/w the two forms is non-trivial.

For CPHA=0 the "out side" changes the data on the trailing edge of the preceding clock cycle, while the "in" side capture the data on the leading edge of the clock cycle.

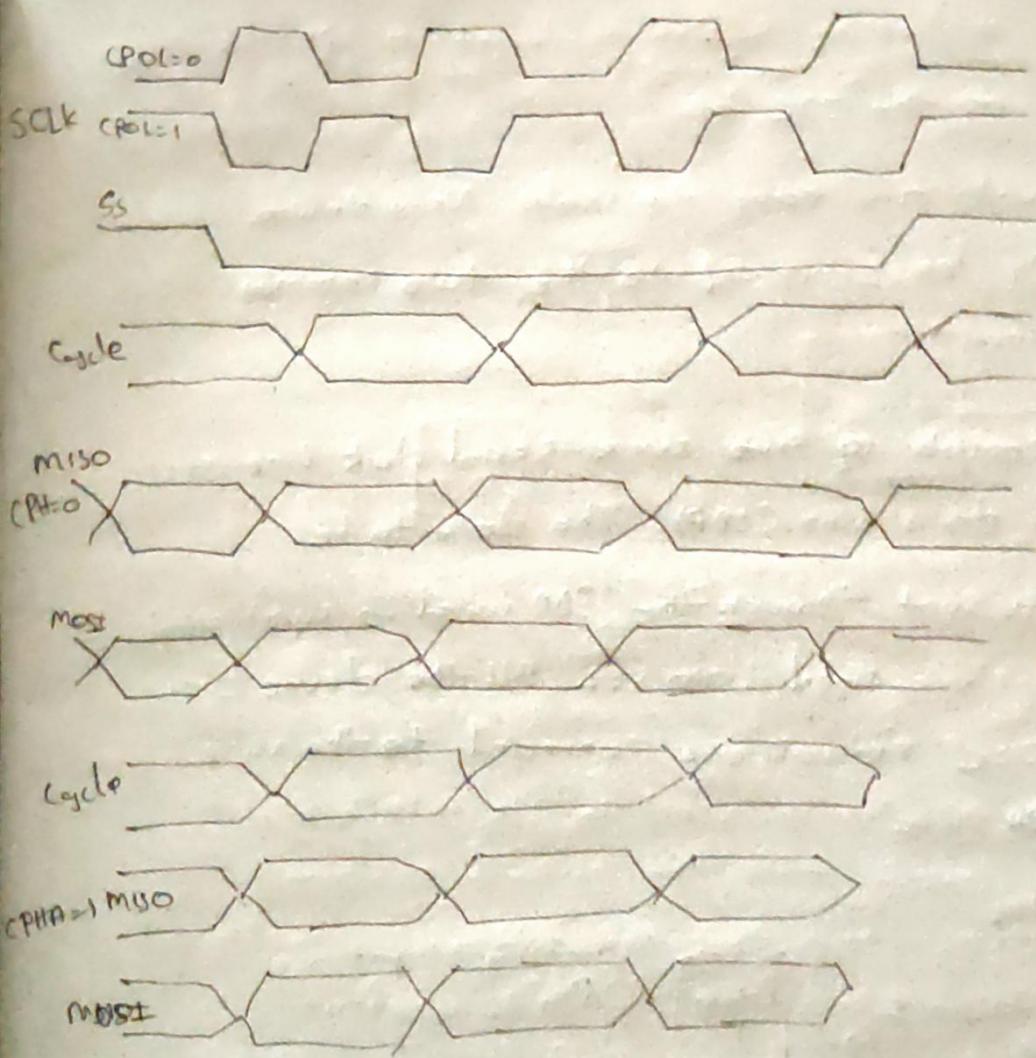
The out side holds the data valid until the trailing edge of the current clock cycle. For the first cycle, the first bit must be on the mosi line before the leading clock edge. An alternative way of considering it is to

that a CPHA=0 cycle consists of a half cycle with the clock idle, followed by a half cycle in the clock asserted.

For CPHA=1 the "outside" changes the data on the leading edge of the clock cycle, while the "in" side 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. In the last cycle, the slave holds the miso line valid until the slave select is deasserted. An alternative way of considering it is to say that CPHA=1 cycle consists of the half cycle with clock asserted, followed by a half cycle with the clock idle.

The mosi and miso signals are usually stable for the half cycle until the next clock transition. SPI master and slave device may will sample data at different points in that half cycle.

P.T.O

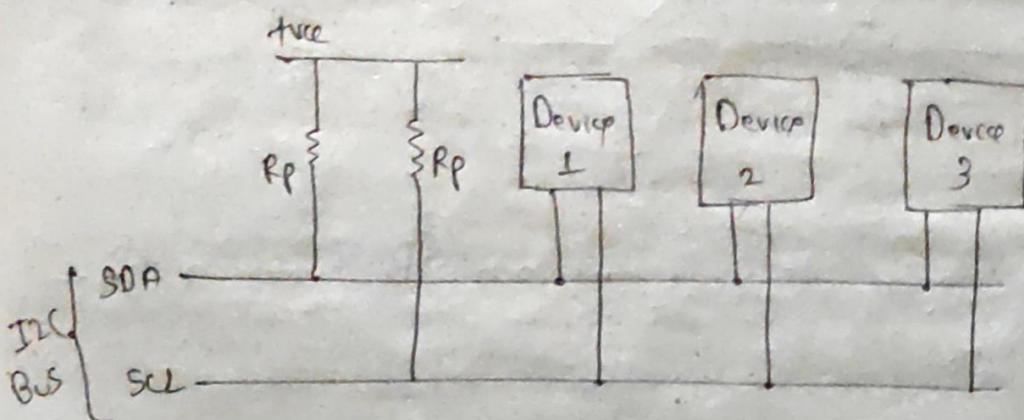


Q1) With the neat diagram explain the features of I2C & its working.

- Ans → Half duplex - serial communication.  
 → Synchronous communication protocol.  
 → Only two common bus line (wire) are required to control any device / IC on the I2C network.  
 → Data transfer speed can be adjusted whenever required.  
 → Simple mechanism for validation of data transferred.

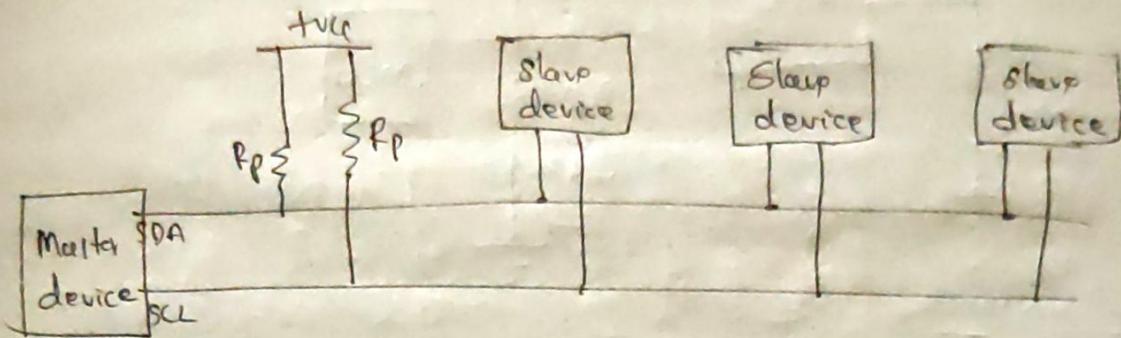
- Uses 7 bit addressing system to target a specific device/IC on the I<sub>2</sub>C bus
- I<sub>2</sub>C network are easy to scale. New devices can simply be connected to the two common I<sub>2</sub>C bus lines.

I<sub>2</sub>C Bus consists of two wires serial clock line (SCL) and serial data line (SDA). The data to be transferred is sent through the SDA wired in synchronised with the clock signal from SCL. All the device/ICs on the I<sub>2</sub>C network are connected to the same SCL & SDA Lines.



Both the I<sub>2</sub>C bus lines (SDA, SCL) are operated in open drain drivers. It means that any device on the I<sub>2</sub>C bus can drive SDA and SCL lines but they cannot drive them high, so a pull up resistor is used for each bus line to keep them high by default.

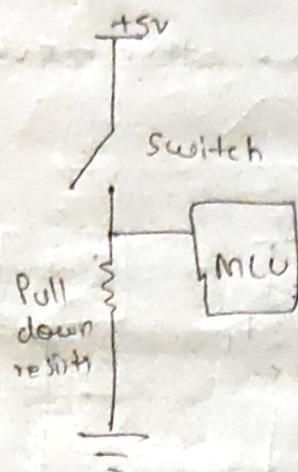
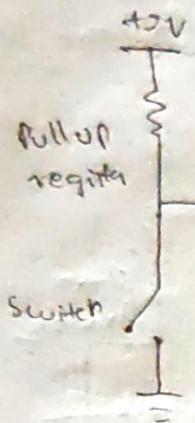
The reason to using an open drain system is that there will be no chances of shorting, which might happen when one device tries to pull the line high and some other device tries to pull the line low.



The devices connected to the I<sub>2</sub>C bus are categorized as either masters or slaves. At any instant of time only a single master stays active on the I<sub>2</sub>C bus. It controls the SCL clock line & decides what operation is to be done on the SDA data line.

All the devices that respond to instructions from the master device are slaves. For differentiating b/w multiple slave devices connected to the same I<sub>2</sub>C bus, each slave device is physically unique. When a master device wants to transfer data on & from a slave device, it specifies this particular slave device address on the SDA line & then proceeds w/ the transfer. So effective communication takes place b/w the master device & a particular slave.

(25) With a neat diagram, explain the pull up/pull down resistor.



### Pull up resistor

A pull up resistor is used to establish an addition 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 wire 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 voltage supply and particular pin in the digital circuit.

### Pull down resistor

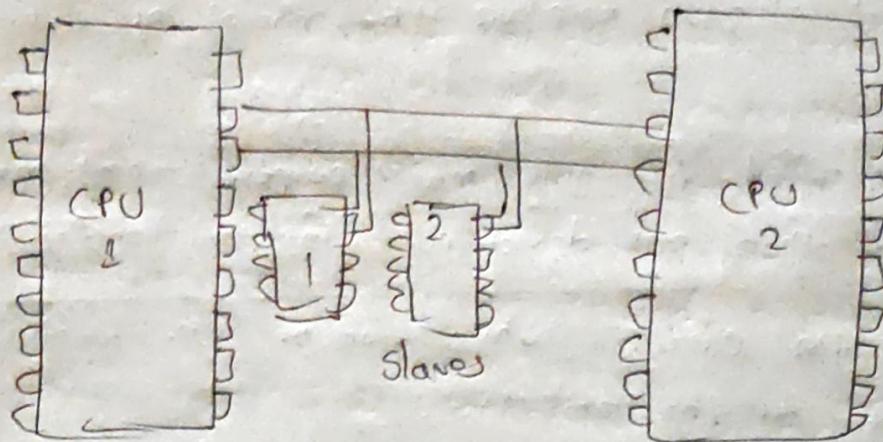
A pull down resistor is used to ensure that up to the logic systems settle at expected logic level whenever the external device are disconnected or of high impedance.

It ensures that the wire is at a defined low logic level when there are no active connections with other device. The pull down resistor holds the logic signal near to zero volts when no other active device is connected.

Q) With a neat diagram explain the concept of arbitration in I2C

- I2C is designed for multi master purpose that means that more than one device can initiate transfer.
- But arbitration occurs when two or more masters start a transfer at the same time.
- The 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 thus no real risk of captured data except if a slave is malfunctioning or if there is a fault condition involving in the SDA / SCL bus.
- When MCU1 issues a start condition it sends the address of slave will 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 the 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.



- (27) What is clock stretching. Explain clock stretching in I2C

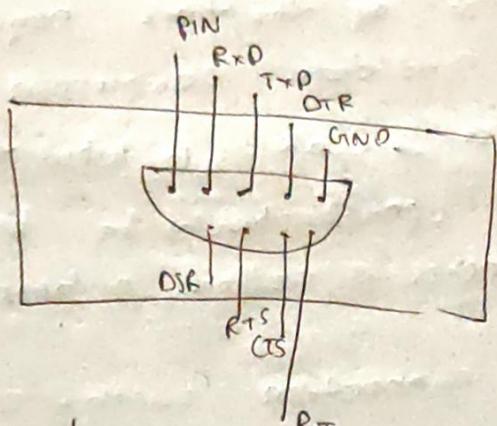
An clock stretching allows an I2C slave device to force the master device into a wait state. A slave device may perform clock stretching when it needs more time to manage data such as store received data or prepare the transmit another byte of data.

clock stretching in I2C devices can slow down communication by stretching SCL during an SCL phase any I2C device on the bus may not again enabling, enabling them to lower down the SCL Clock rate or to stop I2C communication for awhile. This also referred to as Clock

## Synchronisation.

In an I<sub>2</sub>C communication the master device determine the clock speed which relieve masters and slaves from synchronization exactly to a predefined baud rate.

- Q) Explain the working of DB9 pins and handshaking with the modem.



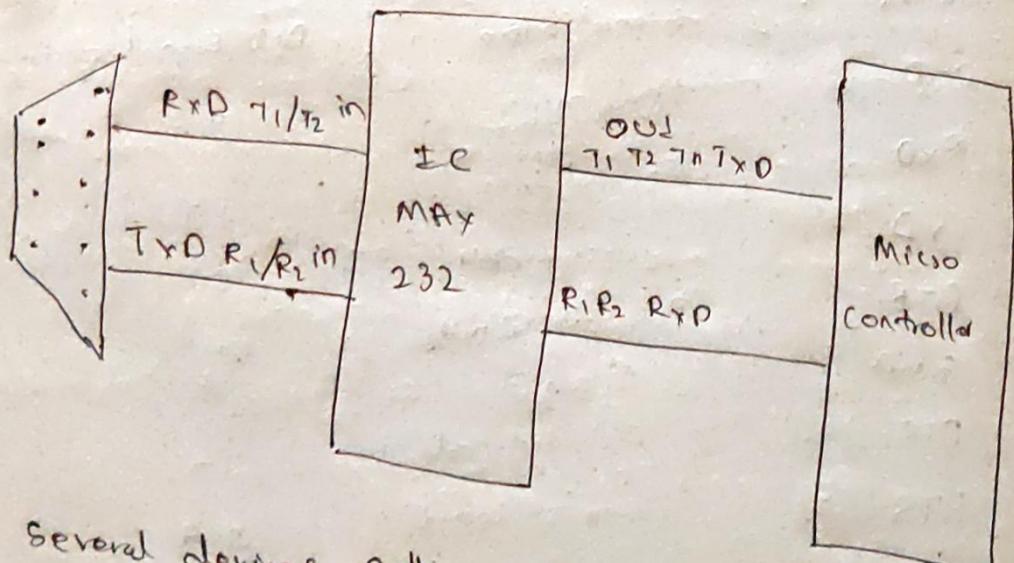
| Pin | Signal | Signal name         | DTE Signal direction |
|-----|--------|---------------------|----------------------|
| 1   | DCD    | Data carrier detect | In                   |
| 2   | RxD    | Receive data        | In                   |
| 3   | TxD    | Transmit data       | In                   |
| 4   | DTR    | Data terminal ready | Out                  |
| 5   | GND    | Ground              | Out                  |
| 6   | DSR    | Data Setready       | -                    |
| 7   | RTS    | Request to send     | Out                  |
| 8   | CTS    | Clear to send       | In                   |
| 9   | RI     | Ring indicator      | In                   |

## Handshaking Modem

A modem handshake is what occurs when the receiving modem answers the phone call and the two modems start begin to communicate.

Before anything else happens the modem must calculate the quality of the time negotiate <sup>etc</sup> control protocols and data compression that they can both recognise & work to what the most suitable connection speed should be <sup>and other</sup> conditions. This process is called a handshake.

- Q) Explain the RS232 connection with a microcontroller



Several devices collect data from sensors and need to send it to another until like a computer for further processing. Data transfer / communication is generally used for long distance communication. In serial communication the data is sent one 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. Microcontrollers can be set to transfer and receive signal data at different baud rate using software instructions.

Q6) Explain the frame format in UART communication.

Baud rate: Baud rate is a data transmission refers to the number of symbols transferred per second. A symbol is a group of a fixed number of bits.

Data Framing