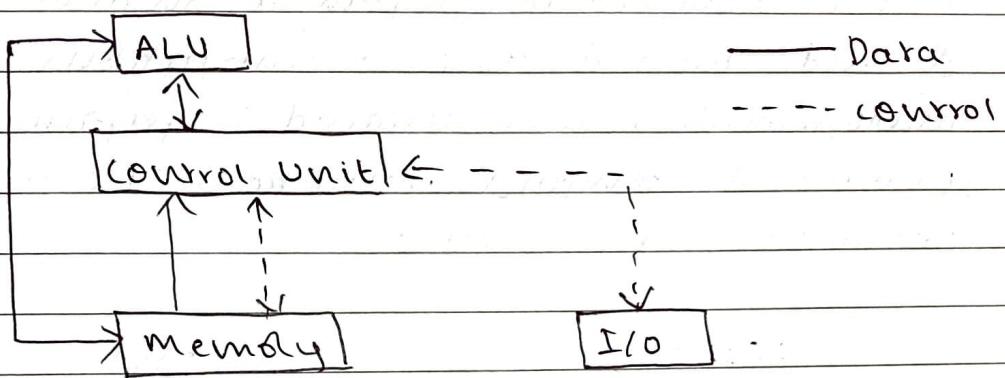


Microcontroller and its Applications - Assignment 1

1. With a neat diagram, explain von neumann and Harvard Architecture.

→ Von-Neumann 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 and data known as stored program concept.

Memory is split to small cells with the same size. Their original numbers are called address numbers.

Advantages:

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

Disadvantages:

- Serial instruction processing does not allow parallel execution. Parallel execution are simulated layer by the OS.
- One bus is a bottleneck.

2. List me difference between RISC and CISC machines

→ CISC:

CISC was developed to make compiler development easier and simple. CISC is a complex instruction set computer.

RISC:

RISC is designed to perform a smaller number of types of computer instructions. It is a microprocessor that is designed to perform smaller number of computer instructions. RISC is Reduced Instruction set computer.

CISC

- more set of instructions
- more addressing modes
- minimum amount of RAM.
- focus is on hardware to optimise.
- high power consumption
- CISC has variable instruction format.
- single Register set.
- less pipelined

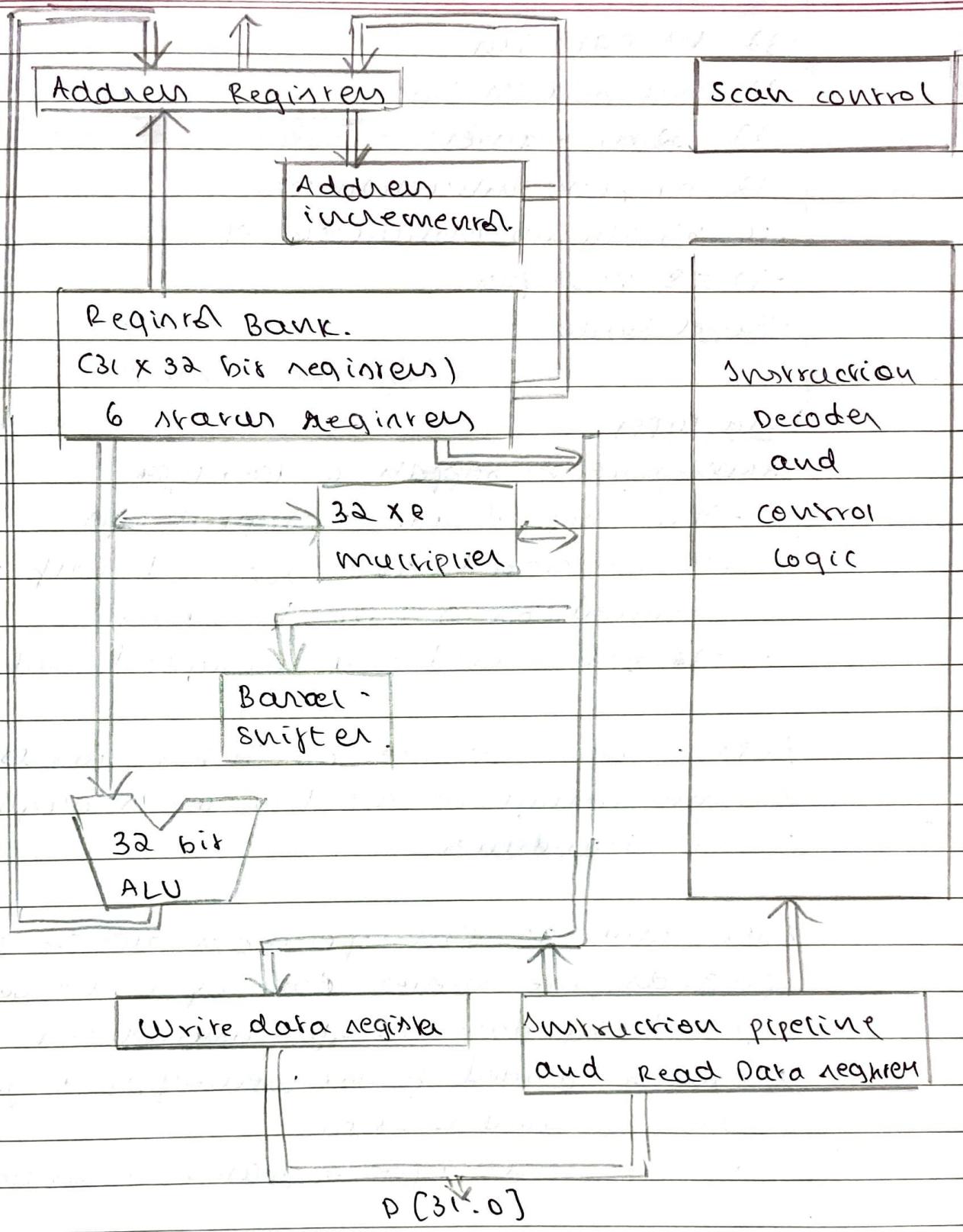
RISC

- 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.

3. Write a neat diagram, explain me ARM7 architecture.

→ Features used

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



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 Neumann Architecture.

→ 3 stage pipeline - fetch, decode, execute.

- 32 bit data bus
- 32 bit Address bus
- 37 32 bit registers
- 32 bit ARM instruction set.
- 16 bit Thumb instruction set.
- 32 x 8 multiplier.
- Barrel Shifter.

Data types.

- ARM processor supports 6 data types
 - 8 bits signed and unsigned bytes
 - 16 bits signed and unsigned half words aligned on 2 byte boundaries.
 - 32 bits signed and unsigned half words

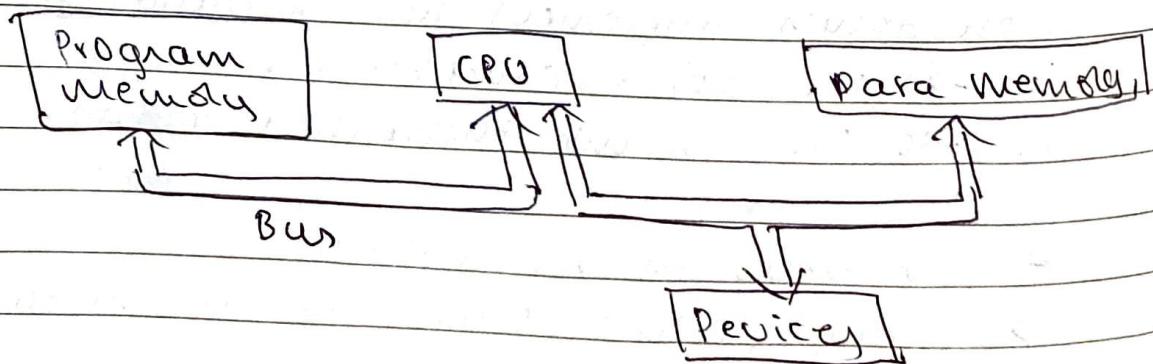
ARM instructions are all 32 bits words aligned
 Thumb instructions are half words aligned on
 2 byte boundaries.

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

- ARM processor supports floating point values

1. Continue.

Harvard Architecture:



It was introduced with Von Neumann architecture.

- It has a slightly different architecture. Memory for data was separated from the memory for instruction.

Advantages:

- since it has two memories, this allows parallel access to data and instruction.
- Development of the control unit is expensive and needs more time.
- Both memories can use different cell sizes

Disadvantages:

- Free data memory can't be used for instruction and vice-versa
- Production of a computer with two buses is more expensive and needs more time.

Q. With a neat diagram, explain the programming model of ARM.

- Each instruction can be viewed as performing a defined transformation of the state.
 - * Visible registers.
 - * Invisible registers.
 - * System memory
 - * User memory.

Processor modes:

- ARM has ~~had~~ seven basic operating modes.
- Modes changes by software control or external interrupt.

~~CPSR~~

<u>(CPSR[4:0]) mode</u>	<u>use</u>	<u>Register</u>
0000 User	Normal user code	User
0001 FIQ	Processing Fast interrupt	-fiq
0010 IRQ	Processing 1rd interrupt	-irq
0011 SVC	Processing software interrupt	-svc
0100 ABT	Processing memory fault	-abt
0101 Undef	Handling undefined instr.	-und
1111 System	Running privileged OS	User

Privileged modes:

- Most programs operate in user mode ARM has other privilege operating modes which are used to handle exceptions, supervising calls and system mode

- more access rights to memory, system and co-processors
- current operating mode is defined by CPSR[4:0]

Supervised 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 bit

- 1 program counter.
- 1 current program status register.
- 5 dedicated saved program status registers
- 30 general purpose registers.

each mode can access,

- a particular set of R0 - R12 registers
- stack pointer, PC, link register.
- Program Counter PC.

- the current program status register CPSR.

privileged mode can access a portion of SPSR saved in program status register.

r0

r1

r2

wcn

r3

r4

r5

r6 F18 IRQ SVC undef Abtr.

r7

r8

r9

r10

r10

r11

r11

r12

r12 r13(sp) r13(sp) r13(sp) r13(sp) r13(sp)

r13 r14(r) r14(r) r14(r) r14(r) r14(r)

r14

r15

rPSR SPSR SPSR SPSR SPSR SPSR

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

→ A Pipelining is a mechanism used by RISC processor to execute instructions.

- By speeding up the execution by fetching the instruction, while other instructions are being decoded and executed simultaneously.

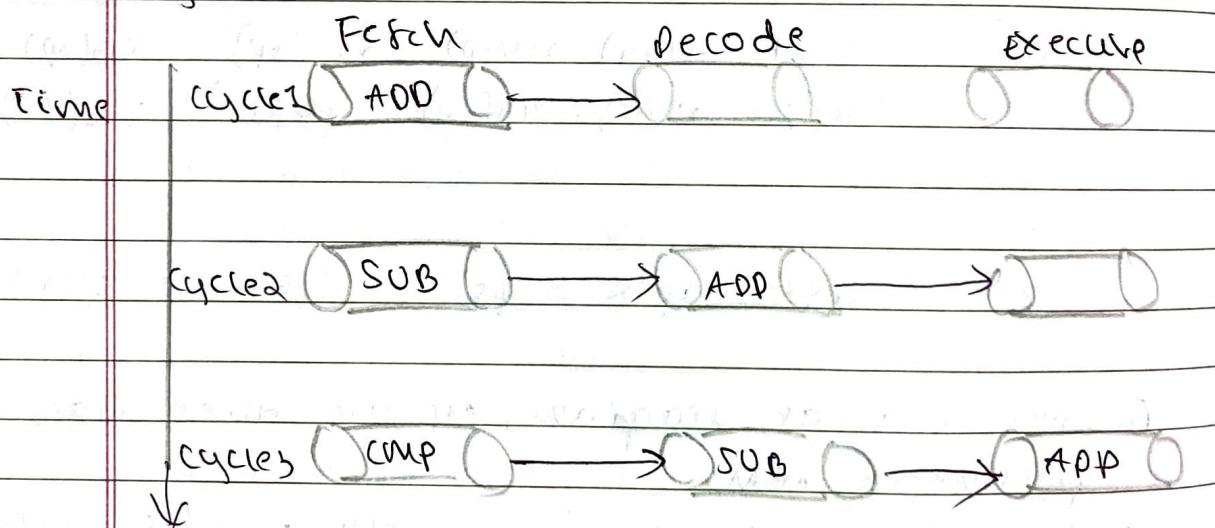
- By pipelining is a design technique a process which plays an important role in increasing the efficiency of data processing in the processor of a computer and etc.

The ARM7 has 3 stage pipeline:

- Fetch: the instruction is fetched from memory.
- Decode: the instruction opcode and operands are decoded to determine what function to perform.
- Execute: the decoded instruction is executed.

Each of these operations requires one clock cycle for typical instructions. Thus, a normal instruction requires 3 clock cycles completely execute known as the latency of instruction execution.

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



7. With a neat diagram, explain CPSR register
→ N Z C V Q undefined I F T mode

Condition code flags:

- N: Negative result from ALU.
- Z: Zero result from ALU.
- V: ALU operation overflowed.

-C: ALU operation carried four.

Negative overflow flag - OF flag:

- Architecture XTE only (Z80, 6502 etc)

- indicates if saturation has occurred during certain operations

Interrupt disable bit:

$\rightarrow I = 1$ Disables me IRQ

$\rightarrow I = 0$ enables me FIQ

I bit:

- Architecture XT only

$\rightarrow T = 0$ present in ARM state

$\rightarrow T = 1$ present in Thumb state

Mode bits:

- specify me process mode

CPSR \rightarrow current process status register:

Holds the information about me current status of me process.

SPSR \rightarrow saved process status register:

Holds the information on me process state before me system changed to this mode i.e. process status just before an exception

Q. Explain me seven different modes in ARM.

\rightarrow The ARM7TDMI process has seven modes of operations.

- User mode is me usual ARM program execution state and is used for executing most application programs.

- Fast interrupt (FIQ) mode supports a data transfer to channel process.
- Interrupt (IRQ) mode is used for general purpose interrupt handling.
- Supervisor mode is a protected mode for the operating system.
- Abar mode is entered after a data or instruction prefetch Abar.
- System mode is privileged user mode for the operating system.

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

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 & exception & to access protected resources.

Mode	Mode Identifier
User	usr
Fast interrupt	fiq
Interrupt	irq
Supervisor	src
Abar	abt
System	sys
Undefined	und

Q. Explain the nomenclature in ARM.

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

ARM7TDMI processor:

32 bit processor Advanced machine

T → Thumb Architecture extension

D → Debug extension

M → Enhanced extension

I → In circuit Emulation

ARM 7x4-144-127 TDMI (E4 E54 E74 K54)

x → series

y → memory management unit

z → cache

t → thumb 16 bit decoder

d → JTAG Debugger

m → fast multiplier

i → embedded ICE (In circuit emulation)

e → Enhanced instruction for DSP

j → JAVA acceleration by Jazelle

f → Floating point

s → synthesizable version

10. What is JTAG? Explain JTAG state diagram

→ JTAG has become a standard in embedded systems and it available in nearly every microcontroller and FPCB on the market.

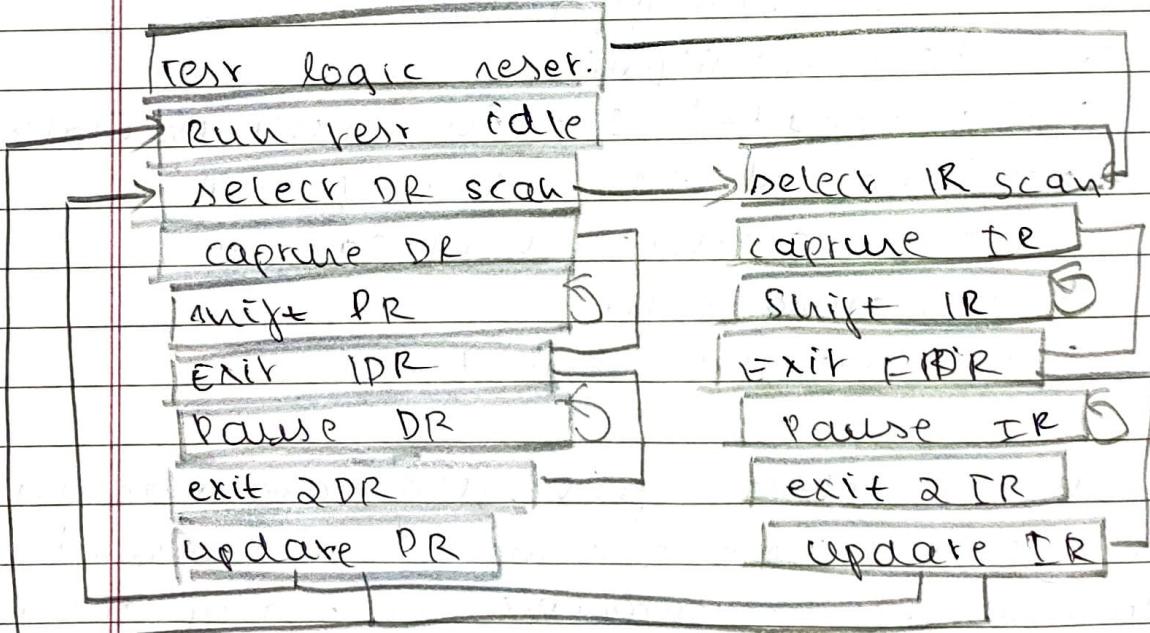
If we have programmed a microcontroller, there's a strong chance that we have used JTAG. & the related standards

JTAG is Joint Test Action Group is an

industry standard for verifying design and testing printed circuit boards after manufacture.

JTAG implements standards for on-chip instructions 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 address bus. The system address is direct buses. The interface connects to an chip Test Access port (TAP) that implements a stateful protocol to access a set of test registers.



Q1. What is single tasking? Give examples & Processor applications.

→ Single Tasking means doing one task at a time with as little distraction and interruption as possible. Microcontrollers are known as computers 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 OS.

12. What is MMU? Why MMU is required? Give eg of MMU support.

→ The memory can be defined as a collection of data in a specific format. It is used to store instruction of processed data. The memory comprises a large array of group of words & bytes, each of its own location. The primary motive of a computer 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 value of the program counter.

The main memory is central to the operation of a computer. Main memory is a large array of words and bytes, ranging in size from hundreds/hundreds to thousands/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 and instructions are kept when the processor is effectively utilizing mem. Main memory is associated with the processor in extremely fast.

Main memory is also known as RAM (Random Access Memory). This memory is a

volatile memory RAM, its data is lost when a power interruption occurs.

Memory management

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

Why memory management is required?

- Allocate & deallocate the memory before & after the process execution.
- To keep track of used memory space by processes.
- To minimize fragmentation on issue.
- To proper utilization of main memory.
- To maintain data integrity while executing of process.

e.g.: IBM System/360 Model 67, IBM System/30, ARM

ARM architecture based application processor implements an MMU defined by ARM's virtual memory system architecture. The current architecture defines PTE's for describing 4KB and 64KB pages, 2MB sections and 16MB super sections.

legacy versions also defined memory space as 16 bits and 32 bits.

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

→ ~~#include <stdio.h>~~

int main()

{

aligned int x = 0.76543210;

char *c = (char *)&x;

if (*c == 0x10)

printf("Underlying architecture is little endian\n");

y

else

{

printf("Underlying architecture is big endian\n");

y

return 0;

}

15. Explain following

→ ① 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.

② Byte: A byte is a combination of eight bits. Eight bits represent a character and is called a byte.

③ Nibble: A nibble is a combination of four bits. In other words, a nibble is half a byte.

④ word: a word is a combination of 16 bits, 32 bits or 64 bits depending on the computer is known as quad word.

Explain

- (a) Bit (b) Byte (c) Nibble (d) Halfword (e) Word

Binary values are often grouped into a common length of 1's and 0's, their number of digits is called the length of a number. Common bit lengths of binary numbers include bit, nibble, byte. Each 1 or 0 in a binary number is called a bit.

A group of 4 bits is called a nibble.

A group of 8 bits makes a byte.

Length	Name	Example
1	Bit	0
4	Nibble	1011
8	Byte	10110101
16	Halfword	10110101 1001001

word is another length that gets thrown out from time to time. word is much less sounding and more ambiguous. The length of a word is usually dependent on the architecture of a processor. It could be 16 bits, 32, 64, or even more.

The terms half word & single word are often used in contemporary computing.

refer to common word sizes relative to a 32 bit base word size.

half word = 16 bits

word = 32 bits

16. explain me word align and half word align in ARM memory.

→ different processors have different definitions of words. For 32 bit processors, a word is 32 bits (4 bytes). As the name implies, a half word is 16 bits for a 16 bit processor, a word is 16 bit (2 bytes), for 8 bit processor word is 8 bit.

word alignment: the stored addresses are adjacent and can be divided by 4, the last two digits are 0P.

half word alignment: that is the stored addresses are adjacent and divisible by 2, that is the last bit is 0.

ARM architecture requires 32 bit. ARM instructions must be word aligned and stored in memory and 16 bit thumb instructions requires half word aligned and stored. therefore in ARM state the value of RIS is always divisible by 4, that is lowest 2 bits of the RIS register are always 00.

In the thumb state, the value of RIS is always divisible by 2, which is lowest bit of the RIS register always 0. One word consists of one or more bytes.

are usually integer bit of bytes.

18. Explain the following addressing modes in ALU.

a) three address

b) two address

c) one address instruction using APM.

→ Sequence of instructions forms a program to perform a specific task.

2 components

opcode field - specifies how to manipulate

Address field - specifies the data location.

When data to be read from or write in two, & 1 more address field may have one address.

* Three address instruction.

* Two address instruction.

* One address instruction.

* zero address instruction.

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

Unique binary sequence pattern must be unique. This process is called opcode encoding.

One address instructions:

This uses an implied accumulator register for data manipulation and the other is in

One 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 addr

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 location. But require more number of bits to represent the address.

Eg: MOV R1, R2
 $R_1 \leftarrow [R_2]$

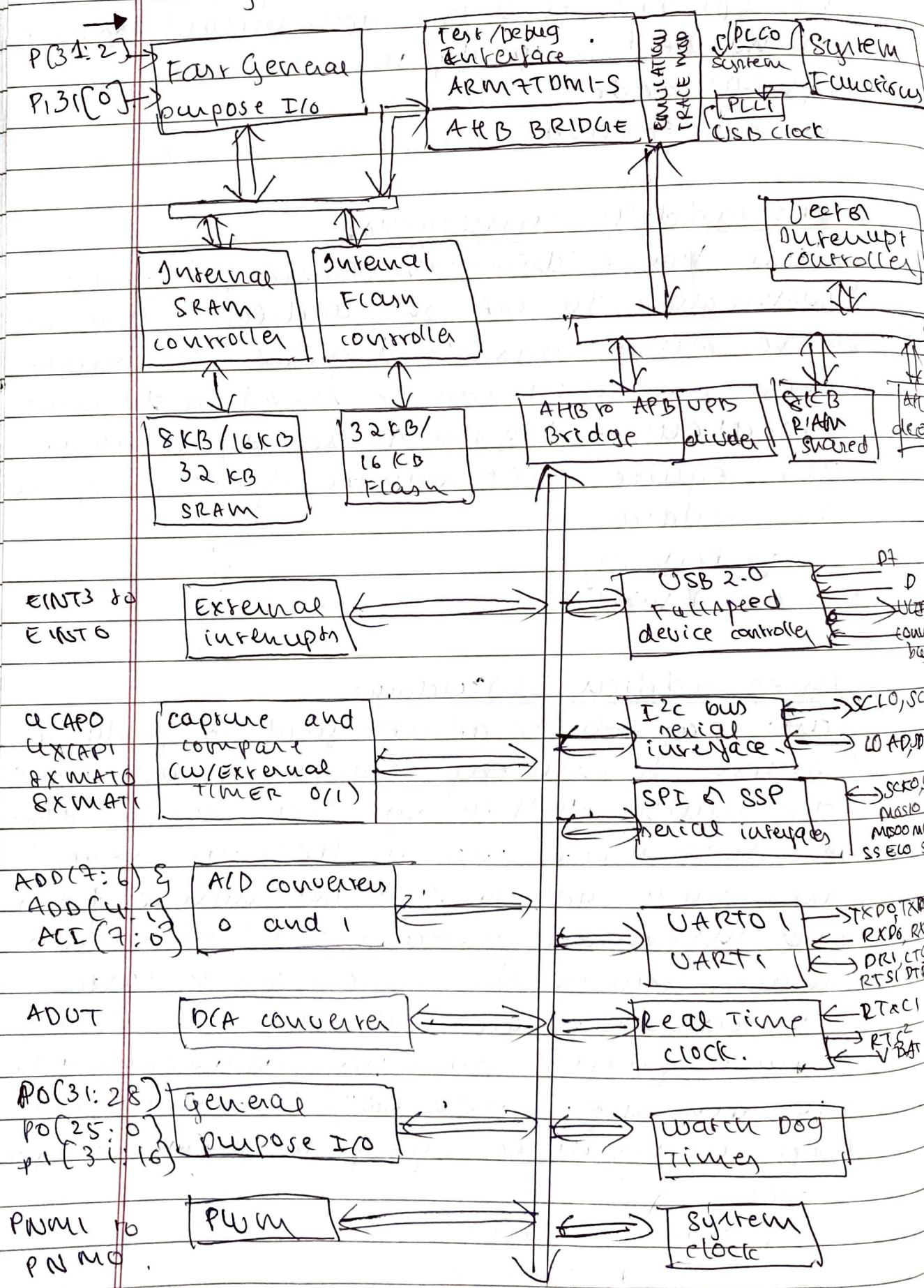
Three address instruction:

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 instruction increases. 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 information but each micro operation will be performed in one cycle only.

Eg: ADD R3, R1, R2

$$R_3 = R_1 + R_2$$

19 Explain me LPC2148 microcontroller block diagram.



LPC2148 Features

- 16 bit/32 bit ARM7TD MI-S microcontroller
- 8 kB to 40 kB of on chip static RAM
- 32 kB to 512 kB of on chip flash memory
- 128 bit wide interface/accumulator enables 66 MHz high speed operation
- In system programming / in application programming via on chip boot loader software.
- USB 2.0 full speed device controller with 2 kB end point RAM.
- single 10 bit DAC provides variables analog output
- two 32 bit timers/ external event counters, POWER unit and watch dog timer.
- low power real time clock (RTC) with independent power and 32 kHz clock input
- up to 21 external interrupt pins available
- one on chip integrated oscillator operating with an external crystal from 1 MHz to 25 MHz.
- single power supply with POR, BOR circuits
- CPU operating voltage range of 3.0 V to 3.5 V.

Q. Explain me LPC2148 microcontroller GPIO pins

→ GPIO → General Purpose Input/Output.

A 32 bit register used to select the functions of pins in which the user needs it to operate. There are 4 functions 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 functions.

There are totally three pin registers in LPC2148 controller in order to control the function of the pins in the respective port. The classification is given below.

PINSEL0 - controls functions of P0R0.0-P0R0.7

PINSEL1 - controls functions of P0R0.8-P0R0.15

PINSEL2 - controls functions of P0R1.0-16-P1R0.0

where the set of 32 bits registers are CPIO configuration - control and status register.

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 by Port 0. Port1 has upto 16 pins available for CPIO functions. PORT0 and PORT1 are controlled via two groups of 4 registers.

* T0PIN .

* T0SET .

* T0DIR .

* T0CCR .

T0PIN :

This register provides the value of port pins that are configured to perform only digital functions.

For example: That particular port pin may have CPIO input, output, UART receiver,

~~push output as selectable functions.~~

If a pin has an analog function as one of its options the pin state cannot be read if the analog configuration is selected. The pin value read in the 10 PIN register is not valid writing to the 10 pin register sets the value in the port output register.

I0SET:

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.

I0DIR:

This word accessible register is used to control the direction of the pins when they are configured as GPIO port pins. Direction bit for any pin must be set according to the pin functionality.

I0CLR:

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 and ~~whereas~~ clears the corresponding bit in I0SET register. Writing 0 has no effect.

If any pin is configured as an input or a secondary function, writing to I0CLR has no effect.

21. With a neat diagram explain Baud rate and Bit rate. Explain the calculation.

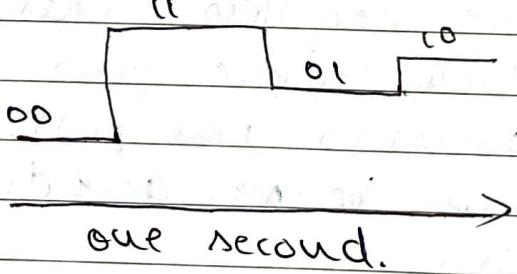
→ Baud: How many times a signal changes per second.

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 times lines 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 = $\times 2$ 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 circuit measured in bits/second.

An analog signal carries 4 bits in each signal unit. If 1000 signal units are occur per second, find the baud rate and the bit rate.

1 Bit \rightarrow 1 symbol

Bit rate = Baud rate

1000

$1000 = \text{Baud rate}$
baud/sec

Bit rate = $1000 \times 4 = 4000$ bps.

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

Baud rate (or symbol rate) is "s"

General formula:

$$b = s \times n$$

$\rightarrow b$ = Data Rates (bit per second).

$\rightarrow s$ = symbol rate (symbols/sec)

$\rightarrow n$ = number of bit per second.

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

$n=4$, Bit rate = $4 \times$ Baud rate.

Q2. 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 system. The interface specification was developed by Motorola.

SPI devices communicates in full duplex using a master-slave architecture usually single master. The master device divides the frame for reading & writing. Multiple slave may be supported through electrical with individual chip select (CS), sometimes called slave select lines.

SPI is called a four wire serial bus, contrasting with there two or one wire serial bus. The SPI may be accurately described as a synchronous serial interface.

But it is different from the synchronous SCI protocol, which is also a four wire synchronous serial communication protocol.

The SPI Bus specifies four logic signals.

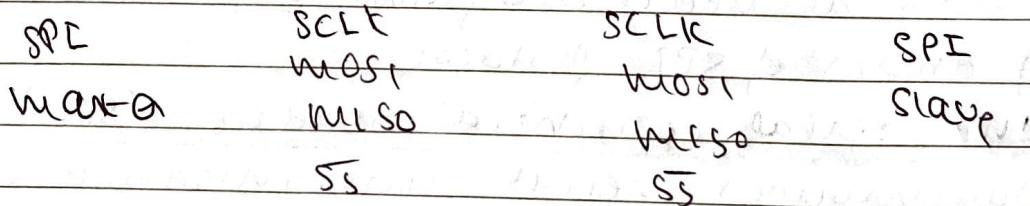
SCLK: Serial clock (output from master)

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 of from master to indicate the data is being sent)



Q3. In SPI with a neat diagram, explain CPOL and CPHA usage.

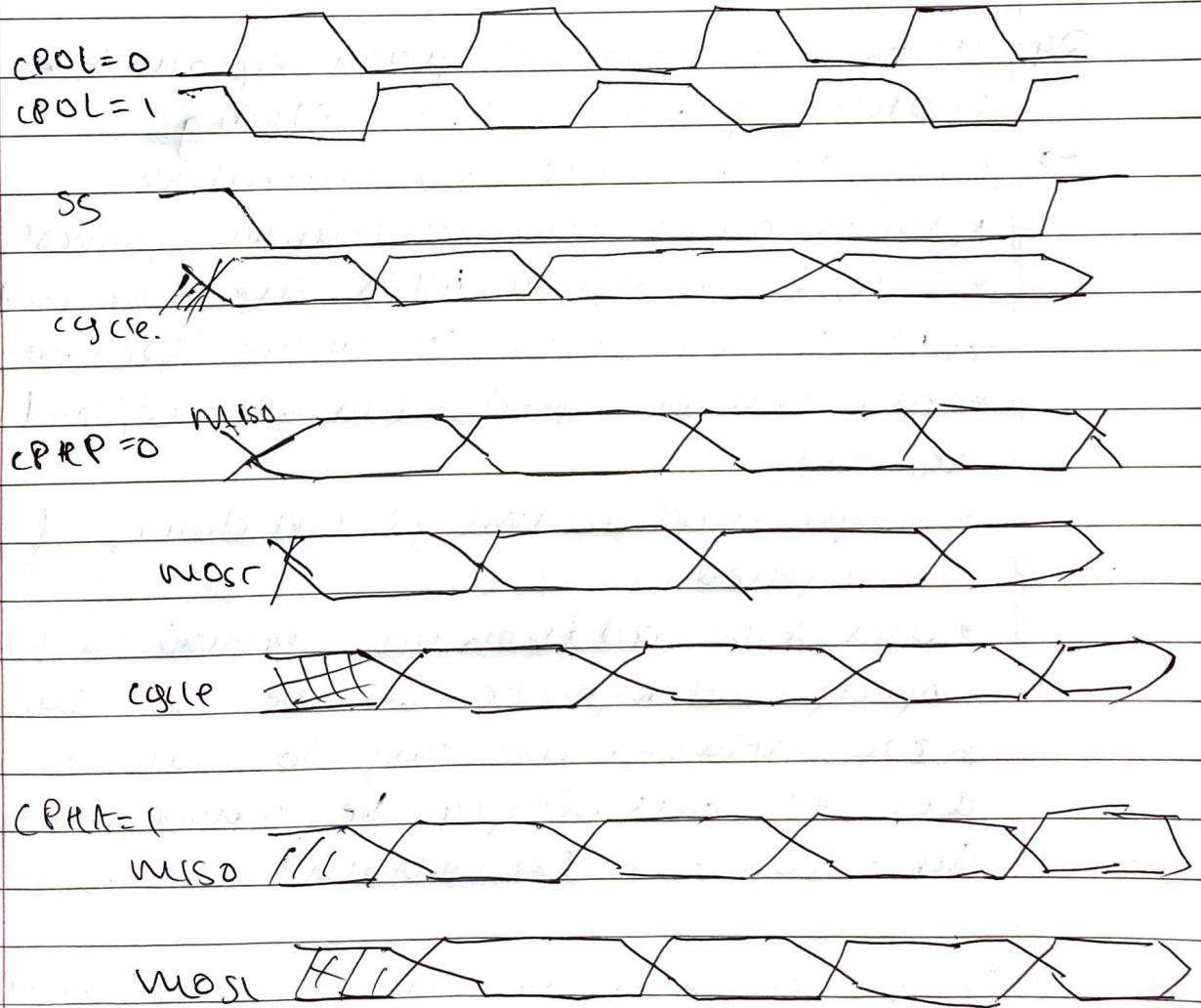
→ CPOL determines the polarity of the clock. The polarities can be converted using a single inverter.

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. Since the loading edge is a falling edge and trailing edge is the rising edge.

$CPHA$ determines the timing of the data bit relative to the clock pulse. Conversion between these 2 forms is non-trivial.

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



The outside holds the data valid until the trailing edge of the current clock cycle.

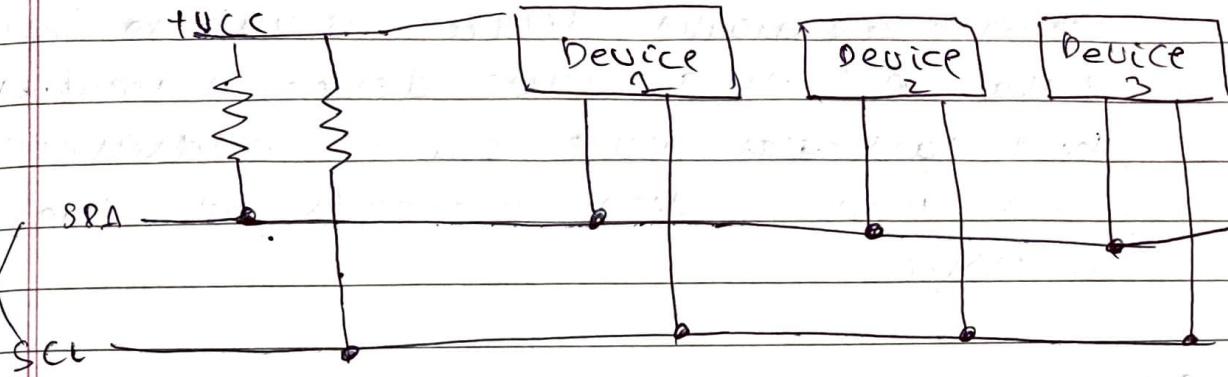
For the first cycle, the first bit must be on the MOSI line before the leading clock edge.

For PPA=1, the "out" side changes the data on the leading edge of the clock cycle, while the "in" side captures the data on the trailing edge of the clock cycle.

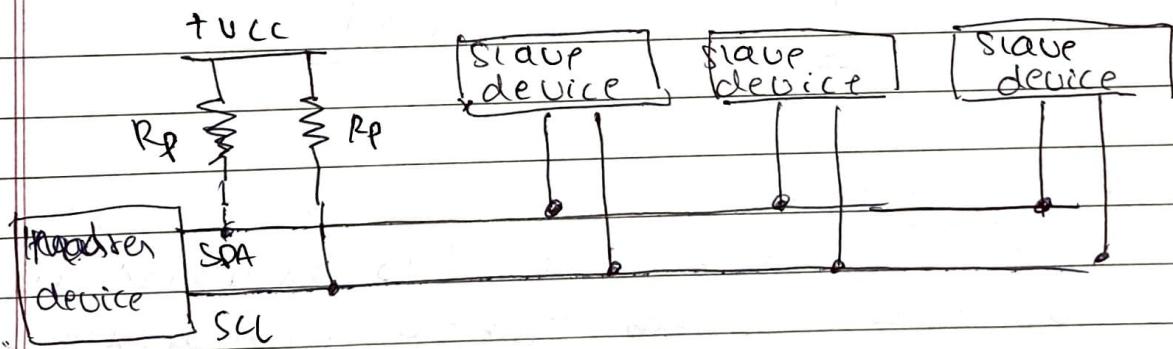
The MOSI and MISO signals are usually stable for the half cycle until the next clock transition. SPI master and devices may still sample data at different points in that half cycle.

- Q4. With the neat diagram explain the features of I²C and its working
-
- * Half Duplex-Serial communication.
 - * Synchronous communication protocol.
 - * Only 2 common lines are required to control any device (IC) on the I²C network.
 - * Data transfer speed can be adjusted whenever required.
 - * Simple mechanism for validating of data transferred.
 - * Uses 7 bit addressing system to target a specific design (IC) on the I²C bus.
 - * I²C networks are easy to scale. New devices can simply be connected to the two common I²C bus line.

I²C bus consists of two wires serial clock line and serial data line (SDA).



Both the I₂C bus lines (SDA, SCL) are operated open drain driver. It means that any device on the I₂C network can drive SDA and SCL but may cannot drive them high, so a pull up register is used for each bus line to keep high by default.

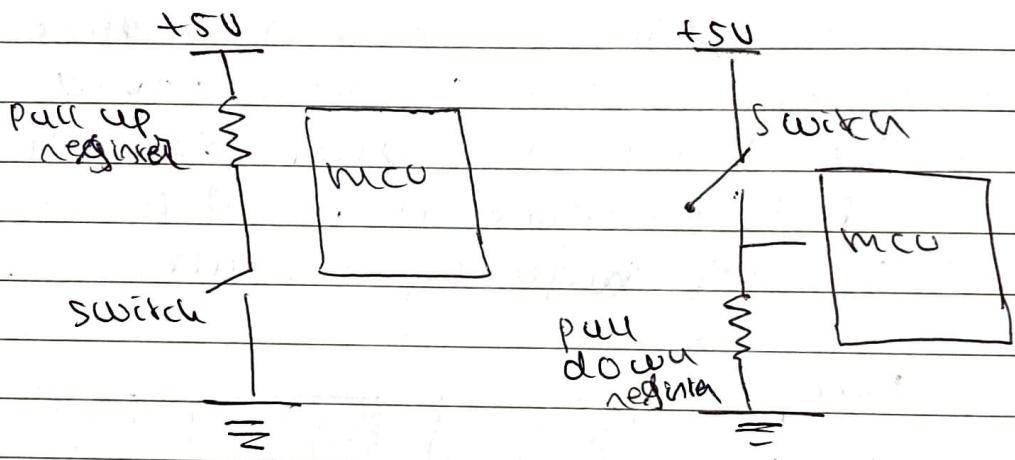


The devices are connected to the I₂C bus are categorized as either master or slaves. It controls the SCL clock line and decides what operation is to be done on the SDA data line.

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

When a master device wants to transfer data or from a slave device, it specifies this particular slave device address on the SDA line & then proceeds with the transfer.

25. With a neat diagram, explain the pull up and pull down resistors.



Pull up resistor

A pull up resistor is used to establish an addition loop over vertical components while making sure that the voltage is well defined even when 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-down resistor:

A pull down resistor is used to ensure that input to the logic system reflects an expected logic level whenever the external device are disconnected at 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 controlled.

Q6. With a neat diagram, explain the concept of arbitration in I₂C.

→ I₂C is designed for multimaster purpose, it means that more than one device can initiate transfer.

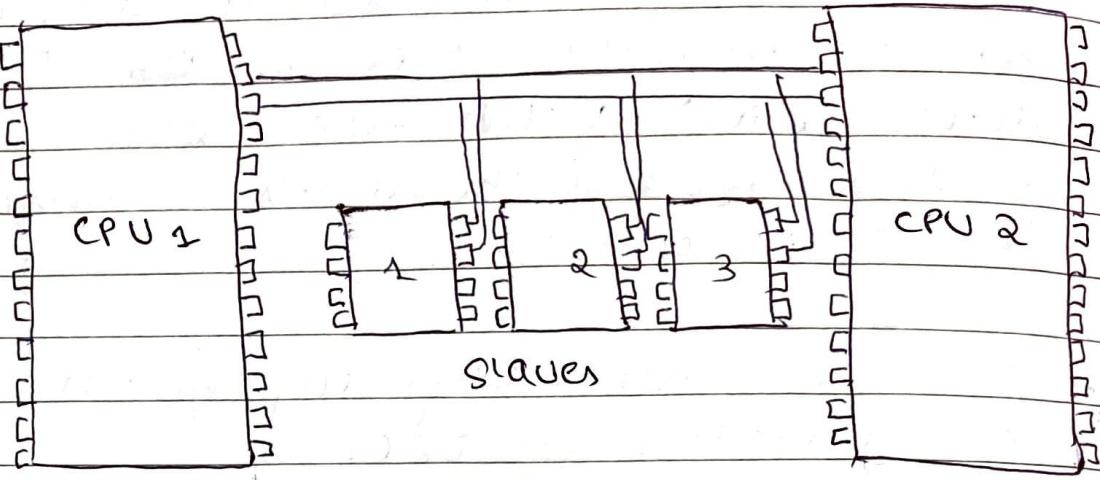
* Bus arbitration occurs when two slave starts a transfer at the same time.

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

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

When MCU issues a start condition and sends me address all 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 microcontrollers 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.



Q7 what is clock stretching. Explain clock stretching in I₂C

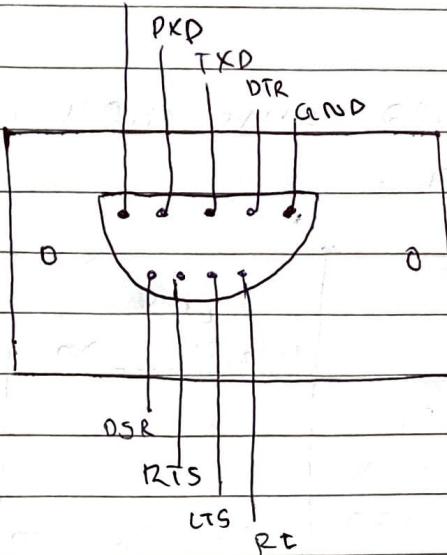
→ Clock Stretching allows an I₂C slave device to force the master device into a wait state. A slave device may allow clock stretching when it needs more time to manage data such as store received data & prepare to transmit another byte of data.

Clock stretching in I₂C devices 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.

In an I₂C communication the master device determines the clock speed which relieve master and slave from synchronisation exactly to a predefined baud rate.

Q8. Explain the working of D89 pins and handshaking with the modem

PINS



Pin	Signal	signal Name	DTE Signal Dir
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

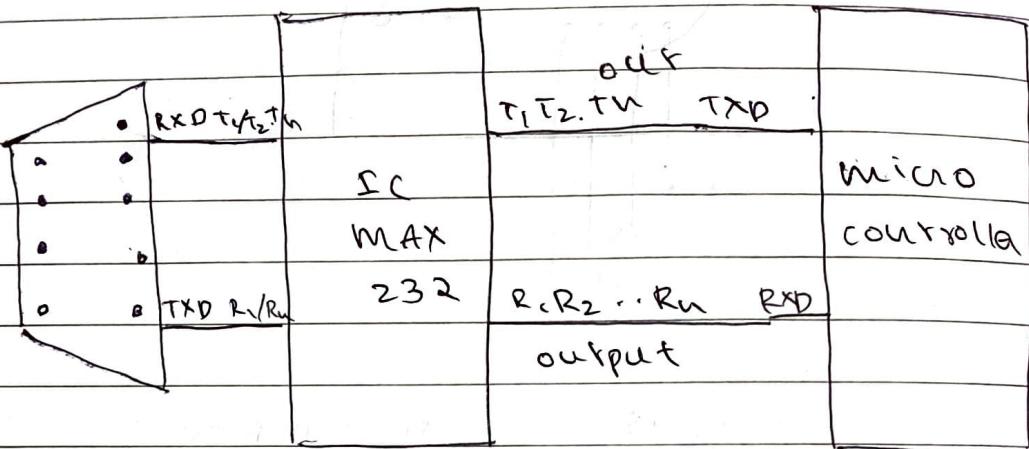
Handshaking modem:

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

Before anything else happens the modem must evaluate the quality of line negotiate and control protocols and data compression that they can both recognise and work out what the most suitable connected speed should be based on.

the condition. This process is called handshake.

29. Explain the RS232 connection using a microcontroller.



Several devices collect data from sensors and need to send it to another unit like a computer for further processing. Data transfer communication is generally done in 2 ways in fact and we use bus lines.

Serial communication on the other hand uses only one or two data lines to transfer data and is generally used for long distance communication. In serial communication, the data is sent as 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. Microcontroller can be set to transfer and receive signal data at different baud rate using software instructions.

30. Explain me frame format in UART communication.

→ Baud rate: Baud rate in 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

UART transmits data in packets. Each data packet may contain one start bit & 8 data bits an optional parity bit and 1 or 2 stop bits.

START	D0	D1	D2	D3	D4	D5	D6	D7	PB	STOP
-------	----	----	----	----	----	----	----	----	----	------

The UART receiver receives data from the data bus and this data are being sent by CPU, memory or microcontroller.

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

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

This data packet is serially transmitted to the receiving UART by the transmitter.

The receiving pin of the receiver UART reads the data bit by bit. This data are again converted into parallel form at the receiver UART and being sent to the data bus at the receiver end after the receiver UART removes the start bit, parity bit and stop.

grant bit: When there is no data transmission the UART transmission line is held at

high voltage to transition acts as the start bit. When we receiving UART detect the high to low voltage transition, it begins reading the data from at the frequency of the baud rate.

Data Frame: The data bits are usually 8 bit 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 reasons for the change in data is mismatched baud rates, electromagnetism & long distance data transfer.

Stop bit:

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

31. Explain the difference between in detail
- Serial v/s parallel
 - Analog v/s digital
 - Synchronous v/s Asynchronous

Serial

- Data is transmitted bit after the bit in a single line

Parallel

- 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 links are easily implemented as hardware. |
| • No crosstalk problem. | • Cross talk creates interface b/w parallel lines. |
| • The bandwidth of serial wire is much higher. | • The bandwidth of parallel wire is much lower. |

Analog

- * Transmitted modulated signal is analog in nature.
- Amplitude frequency & phase variations in transmitted signal represent me into a message.
- * Noise immunity is poor for FSK and PSK.
- Coding is not possible.
- FSK is used for multiplexing.
- Analog modulation systems are AM, PSK, PAM, DPM, SSBM.

Digital

- Transmitted signal is digital i.e. train of digital pulse.
- Amplitude width & position of the transmitted pulse is transmitted in the form of code words.
- Noise immunity is excellent.
- Coding techniques can be used to detect & correct the error.
- TDM is used for multiplexing.
- Digital modulation systems are PCM, DPCM, ADPCM.

Synchronous

- communication in real time
- lessers interruptions in a working
- sends me data and receives me data on the same clock frequencies
- * Fast
- There is no overhead of extra start and stop bit
- Uses constant time intervals.
- Used in chat rooms and video computing.

Asynchronous

- communication not in real time.
- Eliminates interruptions
- Sends me data and receives the data on different clock frequencies
- Slower.
- Uses start & stop bit
- uses random or irregular time intervals
- Used in emails

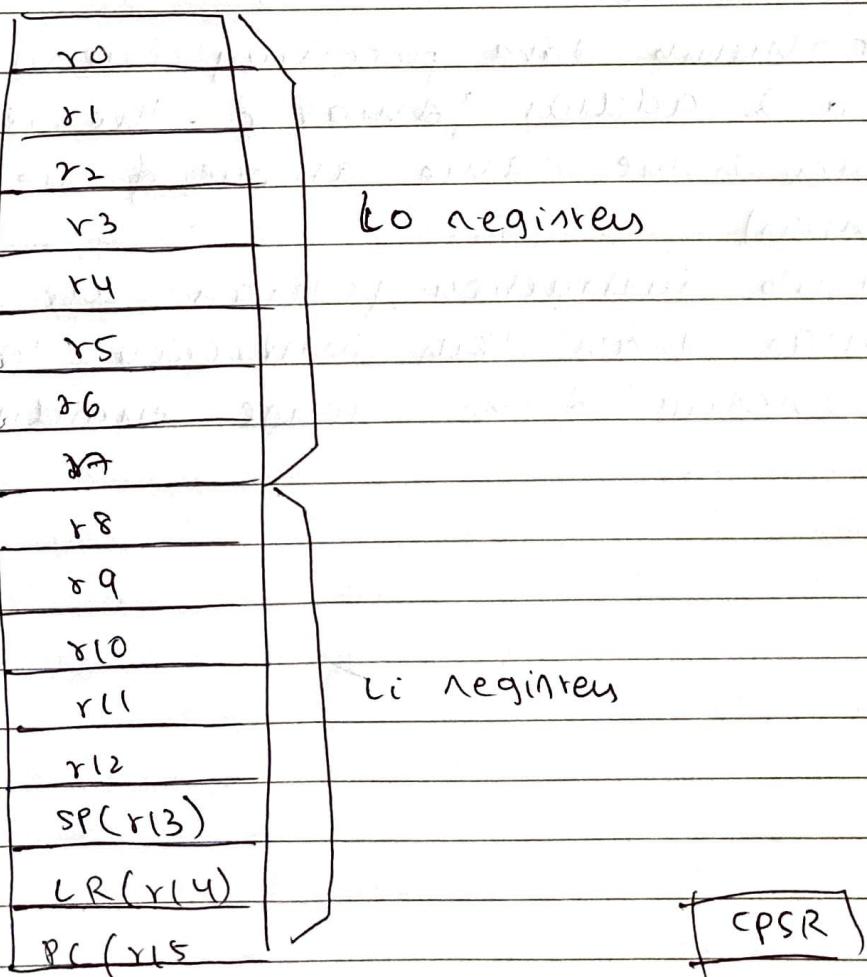
5. With a neat diagram, explain me programming model of Thumb.

→ The Thumb instruction set is a subset of the ARM instruction set and the instructions operate on a restricted view of the ARM registers.

- The instruction set gives full access to the eight 32-bit general purpose registers r0 to r7, and makes extensive use of r13 and r15 for special purpose.
- r13 is used as a stack pointer.
- r14 is used as the linker register.
- r15 is the program counter.

These rules follow very closely the way these registers are used by the ARM instruction set. Though the use of r13 as a stack pointer in ARM code is purely a software convention, whereas in Thumb code it is more or less hard-wired. The remaining registers (r0 to r12 and cpsr) have only restricted access.

- The CPSR code flags are set by arithmetic and logical operations and control conditional branching.
- All thumb instructions are 16 bits long. They map onto ARM instructions so they inherit many properties of ARM instruction set.



CPSR

Fig: thumb accessible registers

- The Load-store architecture with data processing, data transfer and control flow instructions
 - support for 8-bit byte, 16-bit halfword and 32-bit word data types where half-word are aligned on 2-byte boundaries and words are aligned on 4-byte boundaries
 - A 32-bit unsegmented memory
- However, in order to achieve a 16-bit instruction length a number of characteristic features of the ARM instruction set have been abandoned.
- most thumb instructions are executed unconditional.
 - most thumb data processing instructions use a 2 address format i.e. the destination register is the same as one of the source registers)
 - thumb instruction formats are less regular than ARM instruction formats as a result of the dense encoding