Bharath. P. H

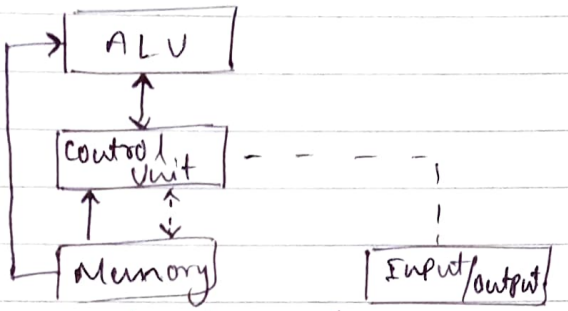
MCA ASSIGNMENT - 1

21103903 1 Page :

Date :

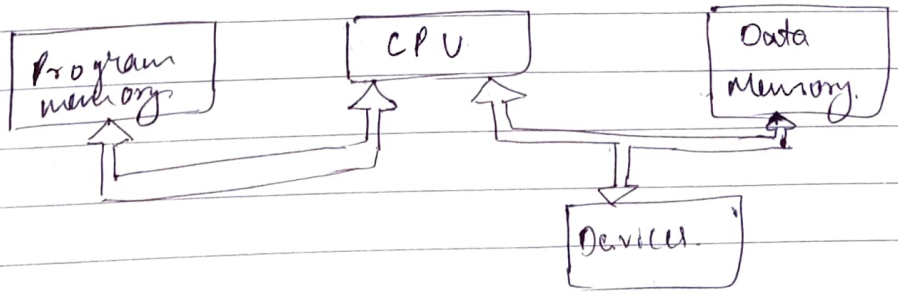
(1) With a neat diagram explain Von Neumann and Harvard Architecture.

* Von-Neumann architecture



* Von Neumann architecture has memory, ALU, control unit and I/P / O/P devices.
* All the components of this are connected together by bus.
* Memory and devices are controlled by CPU.
* Memory hold both program and data known as stored program concept.
* Memory split into small cells with same size
* Program consist of sequence of instructions and they executed in order they are in memory.
* Control unit gets both data and instruction in same way from one memory. Memory organisation is in hands of the programer.

* Harvard architecture:-



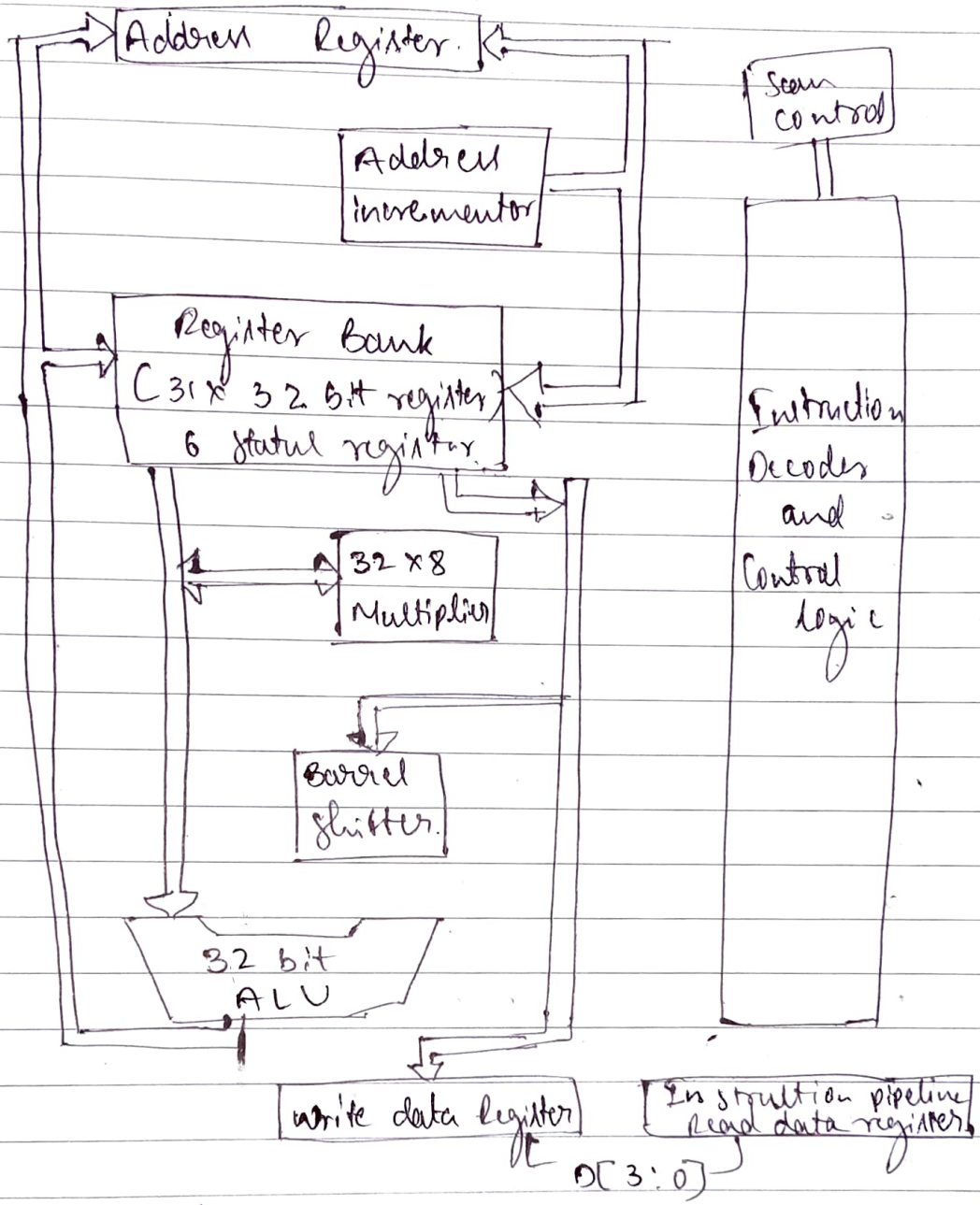
* Memory for data separated from the memory for instruction.

- Since 2 memories allow parallel access of data and instruction.
- Both memory are of different size.
- Development of control unit is expensive and more time.
- free data memory cannot be used for instruction and vice versa.
- Production of a computer with 2 buses is more expensive and need more time.

(2) List the difference RISC vs CISC machines.

RISC	CISC
* Reduced Instruction set Computer	* Complex Instruction set computer.
* An instruction set architecture that is designed to perform a smaller number of computer instructions so that it can operate at a higher speed.	* A full set of computer instructions that intends to provide the necessary capabilities in an efficient way.
* Utilizes a small, highly optimized set of instruction.	* Utilizes a large, specialized and a complex set of instructions
* More machine oriented	* More programmer oriented
* Simple and required one clock cycle to execute instructions.	* Complex, & required multiple clock cycles to execute an instruction.
* More registers	* less registers
* Required more RAM	* Required minimum amount of RAM
* Had simple instructions the program length is long.	* Had complex instructions the program length is short.

(3) With a neat diagram, explain the ARM7 architecture.



* Features used
  -> Load/store architecture
  -> Fixed length 32 bit instruction
  -> 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 registers
-> 32 bit ARM instruction set.
-> 16 bit Thumb instruction set.
-> 32 x 8 multiplier
-> Barrel shifter.

Data types

* ARM processor support 6 data types
  - 8 bit signed and unsigned byte.
  - 16 bit signed and unsigned half words aligned on 2 byte boundries.
  - 32 bits signed and unsigned half words.

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

Internally all ARM operations are on 32 bit operands. the shorter data types are only supported by data transfer instructions, when a byte is loaded from memory, it is zero or signed extended to 32 bits.
ARM co-processor supports floating point values.

(4) With a neat diagram, explain the programming model of ARM

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

Processor modes

* ARM has seven basic operating modes.
* Mode changes by software control or external interrupts.

CPSR[4:0]	mode	Use	Register.
0000	User	Normal user code	User
0001	FIQ	Processing fast interrupts	- fiq
0010	IRQ	Processing std interrupts	- irq
0011	SVC	Processing software interrupts	- svc
0111	Abort	Processing memory faults	- abt
1011	undef	handling undefined instruction types	- und
1111	System	Running privileged OS user.	

Each mode can access

-> Most programs operate in user mode ARM has other privileged operating modes which are used to handle exceptions, supervisor calls and system mode.

* Have access registers to memory system and co-processors
* Current operating mode is defined by CPSR[4:0]

Supervisor Mode:

* Having some protective privileges.
* System-level function can be accessed through specified supervisor calls.

- Usually implemented by software interrupt.

ARM has 37 registers all of which are 32 bit.

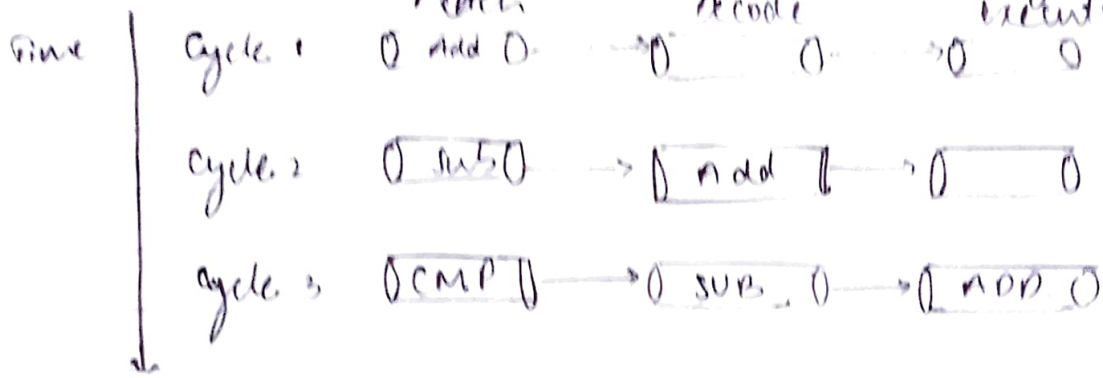
- Program counter
- 1 current program status register
- 5 dedicated saved program status registers
- 30 general purpose registers

Each mode can access
-> a particular set of R_0 -> R_{12} registers.
-> stack pointer, R_{14} - link register.
-> Program counter r15 (PC)
-> The current program status register CPSR.

User
r_0
r_1
r_2
r_3
r_4
r_5
r_6
r_7
r_8
r_9
r_{10}
r_{11}
r_{12}
r_{13}
r_{14}
r_{15}
CPSR

FIQ	IRQ	SVC	Undef	Abort
r_8				
r_9				
r_{10}				
r_{12}				
r_{12}				
$r_{13}(SP)$	$r_{13}(SP)$	$r_{13}(SP)$	$r_{13}(SP)$	$r_{13}(SP)$
r_{14} lr	$r_{14}(lr)$	$r_{14}(lr)$	$r_{14}(lr)$	$r_{14}(lr)$
Spsr	Spsr	Spsr	Spsr	Spsr

(6) With a neat diagram explain three stage pipeline organization.



-> A pipelining is the mechanism used by RISC processor to execute instructions.
-> By speeding up the executed by fetching the instruction while other instructions are being decoded and executed simultaneously.
-> Pipelining is a design technique or process which play an important role in increasing the efficiency of data processing in the processor of a computer & etc.

The ARM7 has three 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 three clock cycles to completely execute, known as the latency of instruction execution. Because the pipeline has three stages an instruction execution 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 Q undefined IFT mode

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 - Q flag.
-> Architecture 5TE only.
-> Indicates if saturation has occurred during certain operations.

Interrupt disable bits.
-> I = 1 disables the IRQ
-> I = 0 disables 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 register
Holds the information about the current state 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.

(8) Explain the seven different modes in ARM.

* User mode :- It is the usual ARM program execution state and is used for executing most applications programs.
* Fast interupt (FIQ) mode supports a data transfer or channel process.
* Interupt (IRQ) mode is used for general purpose interupt 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 previliged user mode for the operating system.

We can only enter system mode from another privilged 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 previliged modes previliged modes are used to serviced interupts or exceptions or to access protected resources.

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

Camlin

(9) Explain the nomenclature in ARM.

ARM was originally from Acron computer ltd first RISC processor for commercial use

ARM 7 TDMI Processor.

32 bit processor Advanced machine.
T -> Thumb architecture extension
D -> Debug extention
M -> Enhanced extention
I -> Encircuit emulation.

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 (In circuit emulation)
E -> Enhanced Instruction for DSP
J -> JAVA accderation by Jazelle
F -> Floating point
S -> Synthesisable 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 i.e. Joint Test action group is a industry standard for verifying designs and testing printed circuit board after manufacture.

JTAG implements standards for on chip instrument in electronic design automation (EDA) as a complementary tool to digital simulation. It specifies the use of a dedicated debug part implementing a serial communication interface for low-overhead access without requiring direct external access to the system addresses and direct buses. The interface connects to an on chip test access port (TAP) that implements a stateful protocol to access a set of test registers.

Test logic reset
run test idle
select DR Scan
Capture DR
Shift DR
Exit 1 DR
Pause DR
exit 2 DR
update DR

select IR Scan
Capture IR
Shift IR
exit 1 IR
Pause IR
exit 2 IR
update IR

(11) What is Single Tasking? Give examples of processor application

Single tasking means doing one task at a time with as little distraction and interruption as possible.

Micro controllers 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 example of MMU support.

The Memory can be defined as a collection of data in a specific format. It is used to store instructions and processed data. The memory comprises a large array or group of words or bytes. each with its 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 values 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 to thousands to billions. main memory is a repository of rapidly available information shared by the CPU and I/O devices.

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 operating system to manage operations b/w 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 and deallocate the memory before and after the process execution.
* To keep track of used memory space by processes.
* To minimize fragmentation issues
* To proper utilization of main memory
* To maintain data integrity while executing of process.

ex:- IBM system/360 mode 67, IBM system 370

ARM

ARM architecture based application processors implement an MMU defined by ARM's virtual memory system architecture. The current architecture defined VMSAv8 for describing 4kB and 64kB pages. 1MB sections and 16MB supersections. Legacy versions also defined 1kB tiny space.

(13) What is Endianness? List the types give examples.

It is a term that describes how data is stored. It defines which end of a multi-byte data type contains the most significant values. There are two types.

Big endian and little-endian.

→ Big-Endian

It is the most common way to store binary data. It places the most significant value first followed by less significant value.

example. Integer 123 it places 1 first followed by the 2 then the last value 3.

[1 2 3]. Most significant bit has least address.

→ Little-Endian

Little-Endian stores the least-significant value first, followed by increasingly more significant values.

example the number 123. in little-endian

[3 2 1]. 3 is stored first followed by 2 and 1. Least significant bit had the least address

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

```
# include <stdio.h>
int main()
{ unsigned int x = 0x 1234567890;
    char *c = (char*);
  if (*c == 0x10)
  { printf ("underlying architecture is little endian");
  }
  else
     printf ("Big endian");
  }
  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 : It a combination of eight bits. Eight bits represent a character and is called byte

nibble : It is a combination of four bits, In other words a nibble is half a byte

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

Length	bit	example
1	Bit	0
4	Nibble	1011
8	Byte	10110101
16	halfword	1011010110010011

(16) Explain the word align and halfword align in ARM memory

Different processors have different definition of word for 32 bit processors 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 for 8 bit processor word is 8 bits.

Word alignment : The stored addresses and adjacent and can be divided by 4, the last two digits are 00.

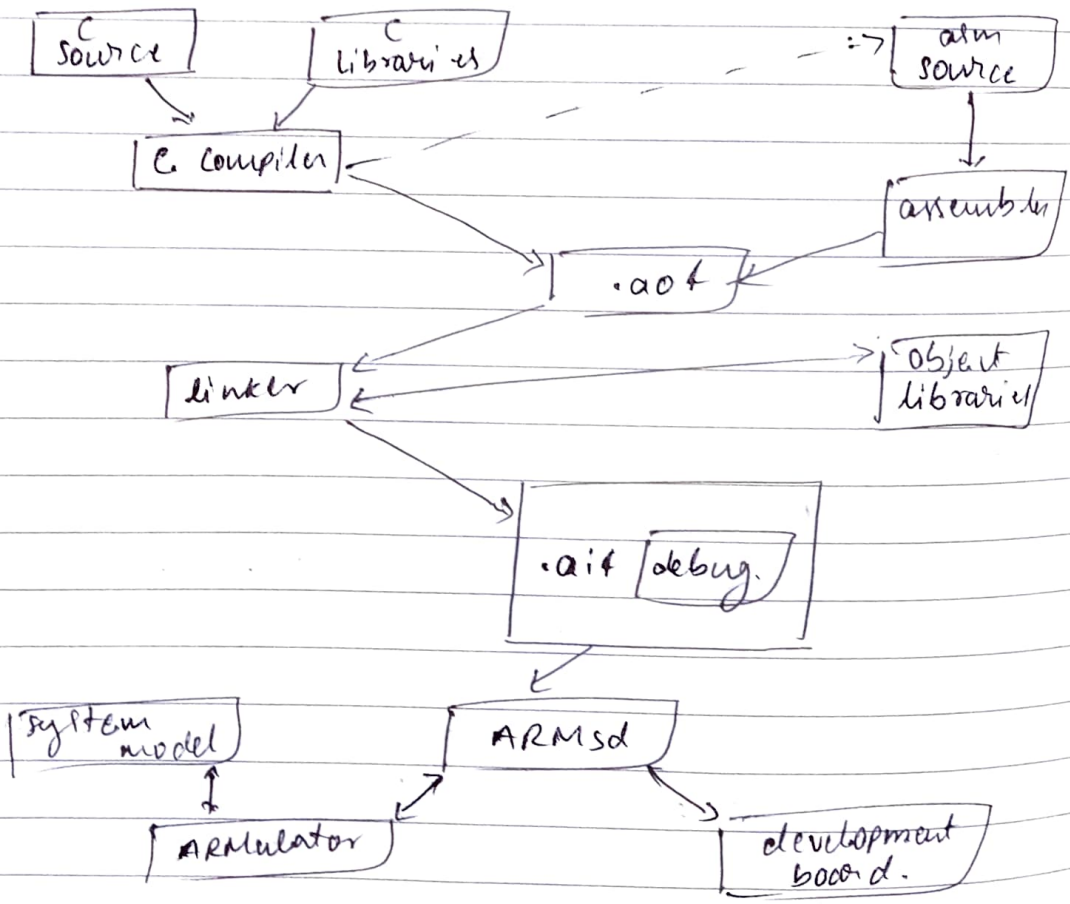
Half word alignment. That is the stored addresses are adjacent and divisible by 2 that is the last bit is 0.

ARM architecture requires 32 bit arm instructions that must be word aligned and stored in memory and 16 bit thumb instructions required halfword

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 the lowest bit of the RIS register always 0. One word consists of one or more bytes. i.e usually integer bits of bytes.

17. Explain the software tools involved and processing the C source file with a neat diagram.

Software tool flow:



Software development for the ARM is supported by a coherent range of tools developed by ARM and there are many third party and public domain tools available such as a ARM backend for the gcc C compiler.

Since ARM is widely used as an embedded controller where the target H/w will not make a good env for software development, the tools are intended for cross-development from a platform such as a PC running windows or a suitable UNIX workstation.

* From the figure

C or assembler source files are compiled or assembled into ~~Arm~~ object format (.aof) files, which are then linked into ARM image format (.ai*) files. The image format files can be built to include the debug tables required by the ARM symbolic debugger.

The ARMulator has been designed to allow easy extension of the software model to include system features such as caches, particular memory timing characteristics and so on.

ARM C compiler: It uses the ARM procedure call standard for all externally available function. The compiler can also produce Thumb code.

ARM assembler: It is a full macro assembler which produces ARM object format output that can be linked with o/p from the C compiler.

-> Assembly source language is near machine level with most assembly instructions translating into single ARM.

(8) Explain the following addressing modes with example
(a) Three address
(b) Two address
(c) One address instruction used in ALU.

One address Instruction

This uses an implied accumulator register for data manipulation and the other is in the registered memory location. Implied means that the CPU already know that one operand is in the accumulator so there is no need to specify it ex: LOR addr

ACC ← (addr)

Two address instruction.

Here two address can be specified in the instruction. In one address 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 bit to represent the address.

ex: MOV R1, R2

$R_1 \leftarrow [R_2]$

Three address Instruction.

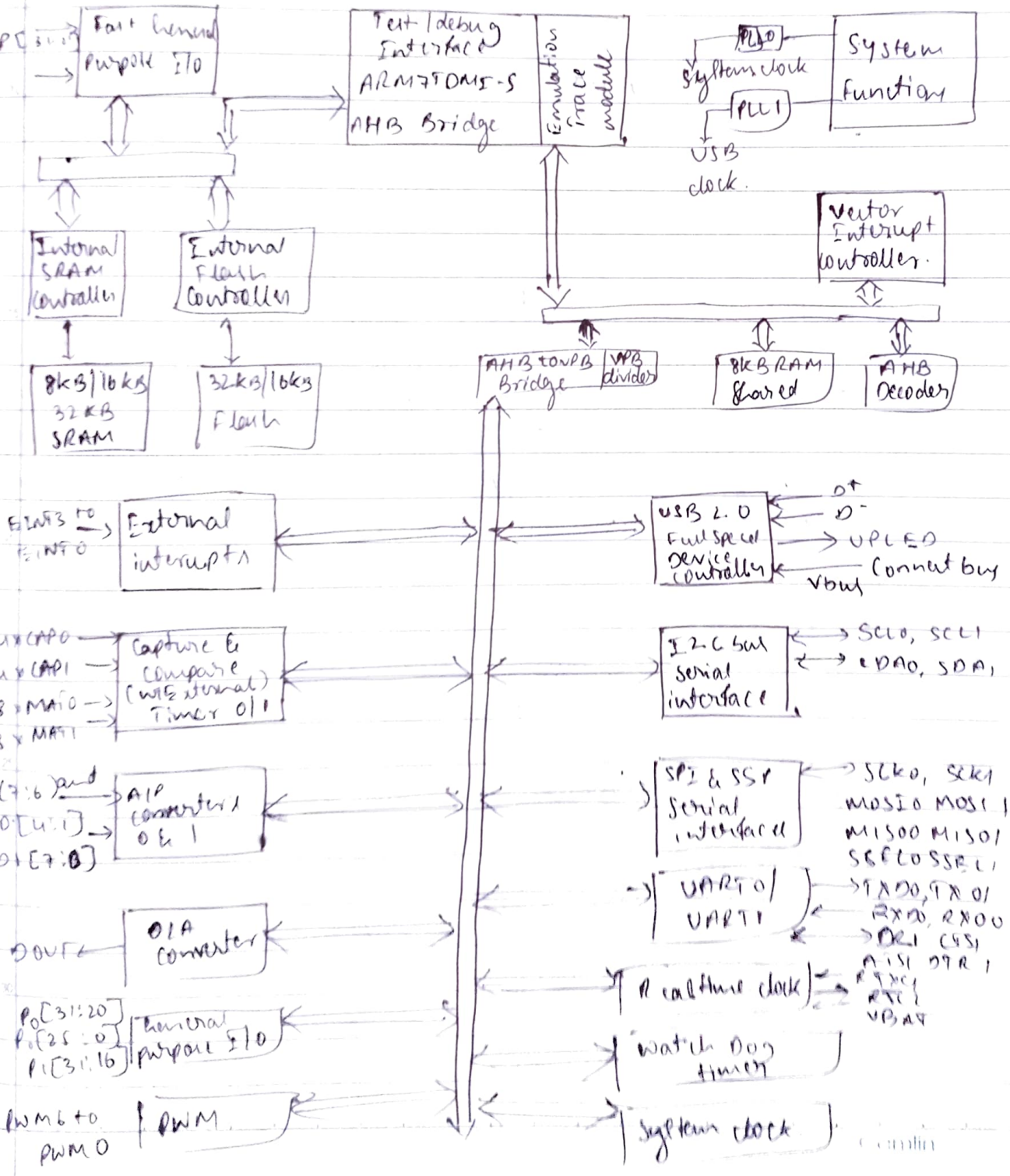
This has three address field to specify a register or memory location. Programs created are most short in size but number of bits per instruction increase. Programs created are much short in size but number of bits per instruction increase. These instructions make creation of program much easier but it doesnot mean that program

will run much faster because how instruction only contain more information but each micro operation will be performed in one cycle only.

ex: ADD R_3, R_1, R_2

$R_3 : R_1 + R_2$.

(19) 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 512 kB of on chip flash memory
-> 128 bit wide interface / accelerator enables 60 MHz high speed operation.
-> In system programming / In Application programming via on chip boot loader software.
-> USB 2.0 full speed device controller with 2kB of end point RAM.
-> Single 10 bit DAC provides variable analog output.
-> Two 32 bit timers / external event counter, PWM unit and watch dog 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 1MHz to 25 MHz.
-> Single power supply with POR, BOD circuits (3V operating voltage range of 3.0V to 3.6V.

(10) Explain the LPC 2148 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 four 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 function.

There is totally three PIN register in LPC 2148

Page :
Date :

controller in order to control the functions of the pins in the respective ports. The classification is given below.

PINSEL0 - controls functions of port 0.0 - Port 0.15
PINSEL1 - Controls functions of Port 0.16 - Port 0.31
PINSEL2 - controls function of Port 1.16 - Port 0.31

where the 00 of 32 bits registers are GPIO configuration.

LPC2148 has two 32 bit GPIO ports. A total of 30 I/O and a single output only pin out of 32 pin are available on PORT0. PORT1 has upto 16 pins available for GPIO function. PORT0 & PORT1 are controlled via two groups of 4 registers.

- IOPIN
- IOSET
- IODIR
- 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 weather the pin is configured for input or output or as GPIO or an alternate digital functions.

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 pin. writing 0 has no effect.

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

Camlin

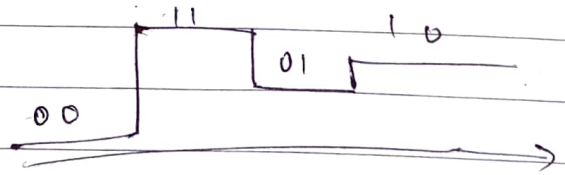
IO CLR: This register is used to produce a low level output at port pins configured as GPIO in an output mode. writing 1 produces low level at the corresponding port pin & clears the corresponding bit in IOSET register. writing 0 has no effect.

(21) With a neat diagram Explain Band rate and bit rate. Explain the calculations.

Band: 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 band and number of signal levels.

Band rate

-> Number of time lines changed per second.
-> Let Band rate be 4 (4 changes per second)
-> Let bits per line change be 2.
-> Bit rate = 8 bits per second.
-> Bit rate = x 2 Band rate in this example



One second.

* Band 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.

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

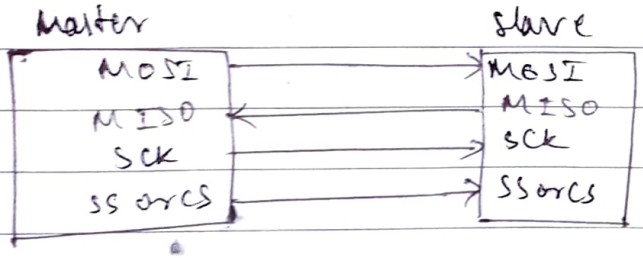
Serial peripheral Interface or SPI is a synchronous serial communication protocol that provides full-duplex communication at very high speeds.

-> It is a master-slave type protocol that provides a simple and low cost interface between a microcontroller and its peripherals.

-> SPI protocol, the devices are connected in a master-slave relationship in a multi-point interface. In this type of interface, one device is considered the master of the bus (usually a microcontroller) and all the other devices are considered as slave.

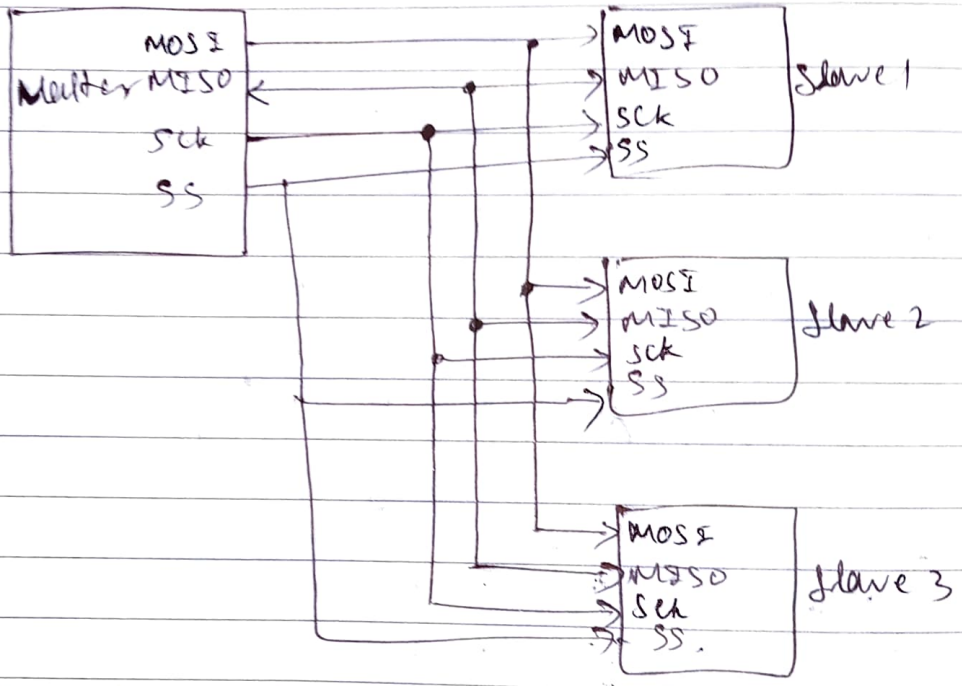
-> SPI bus consists of 4 signals or pins. They are

- Master-Out (Slave-In (MOSI)
- Master-In (Slave-Out (MISO)
- Serial Clock (SCLK) and
- Chip Select (CS) or Slave Select (SS)



-> Master-Out (Slave-in or MOSI as the name suggests is the data generated by the Master and received by the slave. MOSI pins on both the master and slave are connected together. Master-In (Slave-Out or MISO is the data generated by slave and must be transmitted to Master.

MISO pin on both the master and slave are tied together. Even though the signal in MISO is produced by the slave. the line is controlled by the Master. The Master generates a clock signal at SCLK and is supplied to the clock input of the slave. Chip select (CS) or Slave select (SS) is used to select a particular slave by the master.



(23) Modes of operation

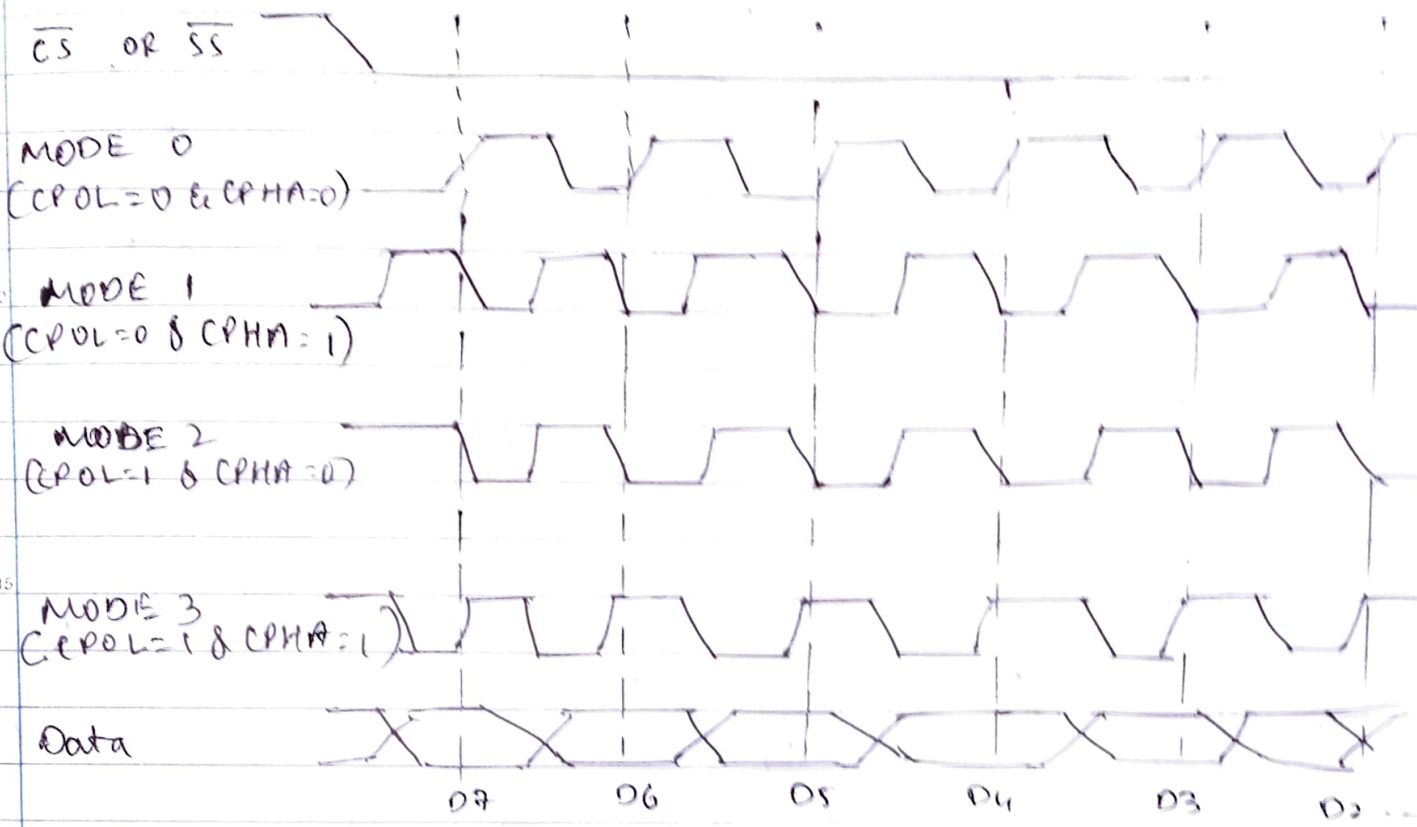
→ Mode 0 :- Mode 0 occurs when clock polarity is low and clock phase is 0 (CPOL = 0 and CPHA = 0) During this data transmission occurs during rising edge of the clock.

→ Mode 1 :- It occurs when clock polarity is low and clock phase is 1. data transmission occurs during falling edge of the clock.

→ Mode 2 :- It occurs when clock polarity is high and clock phase is 0. Data transmission occurs during rising edge of the clock.

Mode 3.

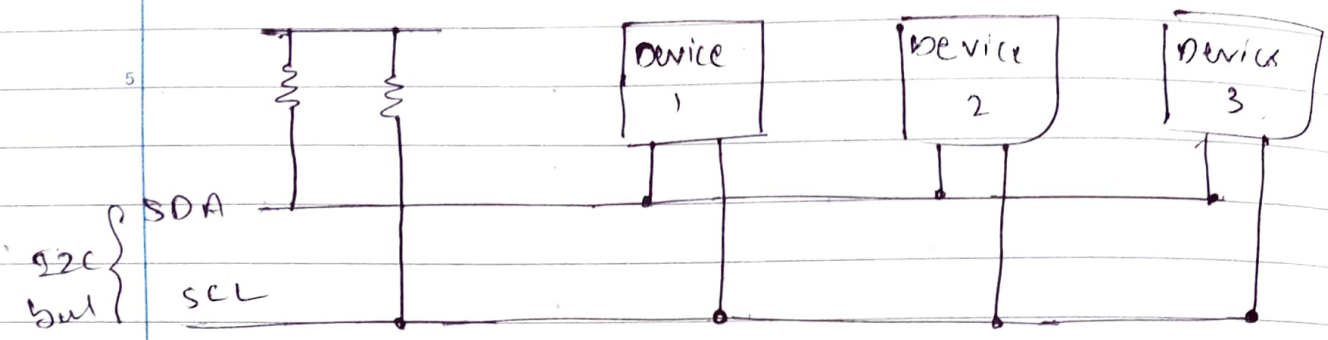
Mode3 occurs when clock polarity is high, and clock phase is 1 and data transmission occurs during rising edge of the clock



(24) With the neat diagram explain the features of I2C and its working.

- Half duplex serial communication.
- Synchronous communication protocol.
- Only two common bus lines (wire) are required to control any device IIC 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 IIC on the I2C bus.

* I2C networks are easy to scale. New devices can simply be connected to the two common I2C bus line.



SDA - Serial Clock line, SDA serial data line. All the devices to the I2C network are connected to the same SCL & SDA line.

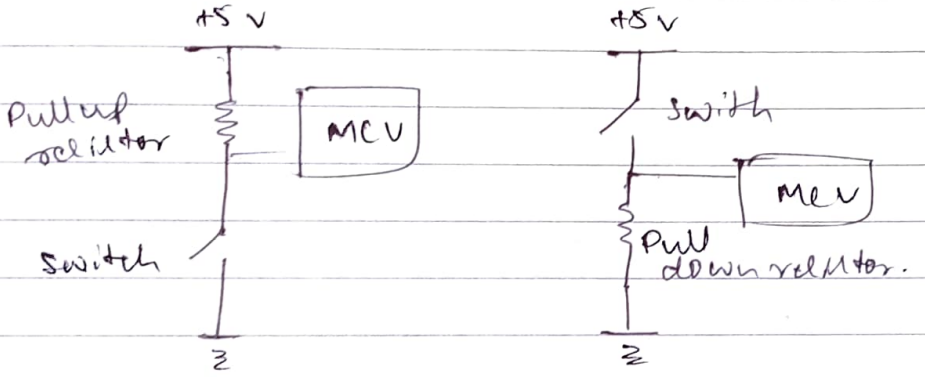
Both the I2C bus lines (SDA, SCL) are operated open drain drivers. It means that any device on the I2C network can drive SDA and SCL low but they cannot drive them high so a pull up resistor is used for each bus line to keep them high by default.

The reason for 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 & some other device tries to pull the line low.

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

When a master device wants to transfer data or receive from a slave device, it specifies this particular slave device address on the SDA line of the proceeds with the transfer. So effectively communication takes place between the master device and a particular slave.

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



Pull up Resistor

A pull up resistor is used to establish an additional loop onto 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 resistors with a fixed value was used to connect the voltage supply and a particular pin in the digital circuit.

Pull down resistor :-

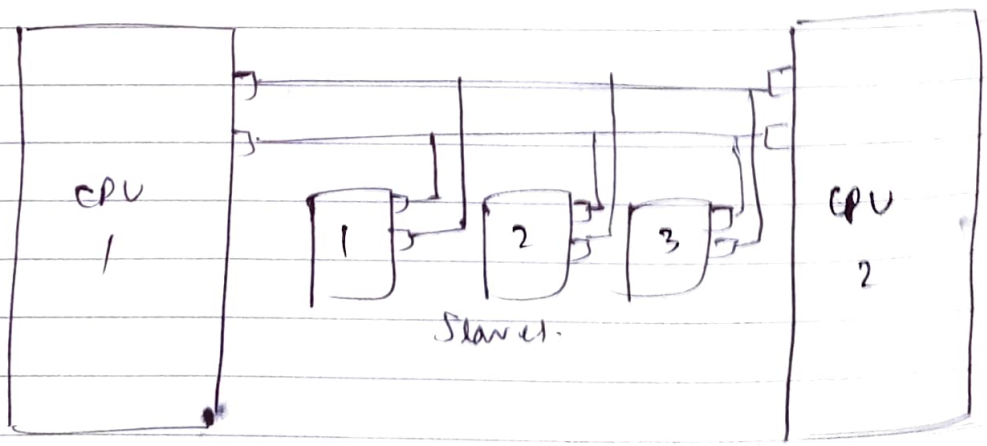
A pull down resistor is used to ensure that input to the logic system rests at expected logic level whenever the external devices are disconnected or at high impedance.

It ensure that the wire is at a defined low logic level when there are no active connections

with other devices. The pull down resistor holds the logic signal near to zero volts when no other outline devices is connected.

(26) 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.
* Bus arbitration occurs when two or more master start a transfer at the same time.
* The I2C bus was originally developed as a multimaster 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 risk of captured data, except if a slave device is malfunctioning or if there is a fault condition involving the SDA / SCL bus.

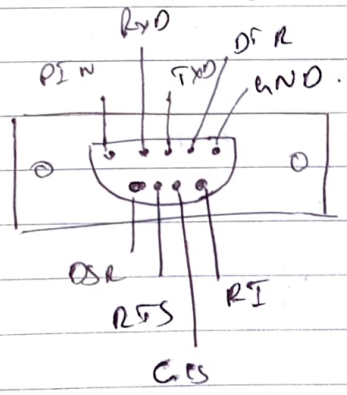


(27) What is clock stretching. Explain clock stretching in I2C.

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 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 low phase any I2C device on the bus may additionally hold down SCL to prevent it to rise again enabling, enabling them to low down the SCL clock rate or to stop I2C communication for a while. This is also referred to as clock synchronisation.

(28) Explain the working of DB9 pins and handshaking with the modem.

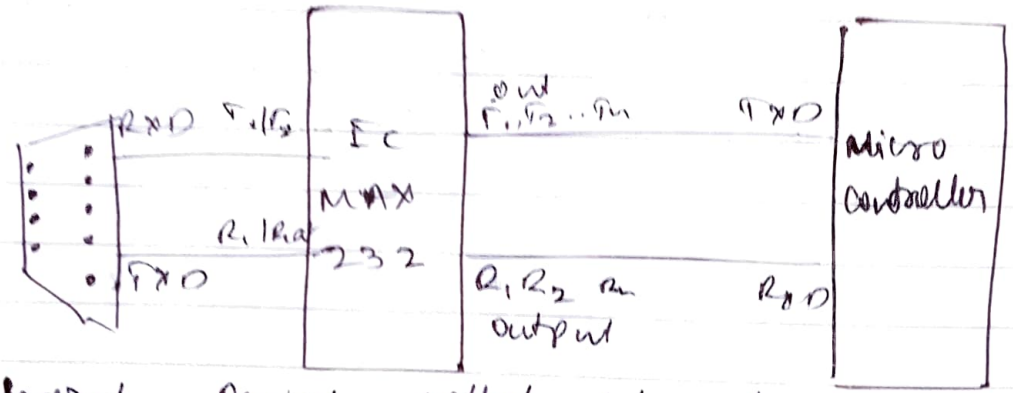


Hand shaking modem:

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

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

(29) Explain the RS232 connection with 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 two ways in Fast and uses more number of lines.

Serial communication on the other hand use only one or two data lines transfer data & it is generally used for long distance communication. In serial communication the data is sent as one bit at a time.

(30) Explain the frame format in UART communication

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

UART transmits data in packets each data packet may contain one start bit 5 to 9 data bits an optional parity bit and 1 or 2 stop bits The UART receives the data from the data bus and this data are being sent by CPU, memory or microcontroller.

The data transmission is in parallel mode.

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

The receiver end of after receiving UART removes the start bit, parity bit and stop bits.

Start bit:

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

Data frame: The data bits are usually 5 to 8 in number. If no parity bit is used it can be 9 bit long. In general can 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 data is mismatched baud rate. electromagnetic at long distance data transfer.

Stop bits: To mark the end of the data packet the sending UART driver 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.

(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 links are easily implemented on 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 lowest.

Analog	Digital
* Transmitted modulated signal is analog in nature	* Transmitted signal is digital i.e. train of digital pulse.
* Amplitude frequency or phase variations in the transmitted signal represent the information or message	* Amplitude width or position of the transmitted pulse is transmitted in the form of code words

* Noise immunity is poor for FAM & PAM
* Coding is not possible

* FDM is used for multiplex

* noise immunity is excellent
* coding techniques can be used to detect and correct the error.
* TDM is used for multiplexing

Synchronus

* Communicated in real time
* Created interupts in a working
* sends the data and receives the data on the same clock frequency
* Faster
* There is no overhead of extra start & stop bit
* Use constant time interval
* Used in chat room & radio communication

Asynchronus

* Communicated in real time
* Eliminated Interrupts
* sends the data and receives the data on different clock frequency
* slower
* Use start & stop bit
* Use random or irregular time intervals
* Used in emails.