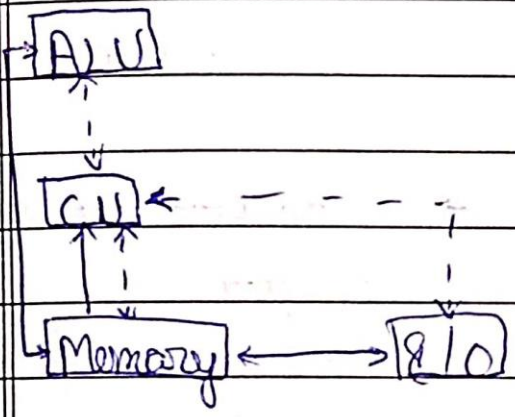
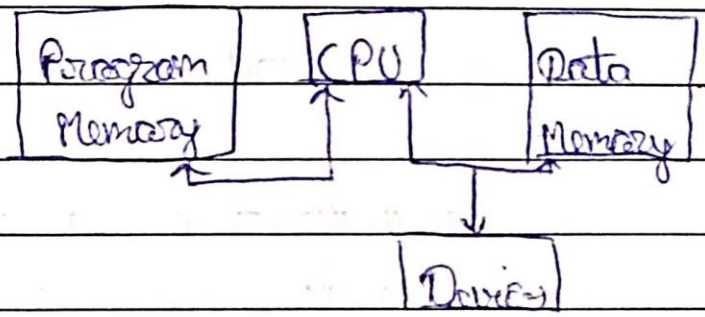
1. With a neat diag, explain Von Neumann and Harvard Architecture.

Von Neumann Architecture	Harvard Architecture.
* 	
* It follows the concept of a stored program computer	* It follows concept of the relay based model by Harvard Mark I.
* It uses one single physical address for accessing & storing both data and instructions	* It uses a seperate physical address during both instruction & data.
* Only single bus is used in transfer of both inst & data	* Seperate buses for transfer both data & inst
* 2 cycles for executing single inst	* 1 clock cycle
* CPU not able to read/write at same time	* Can do simultaneously.

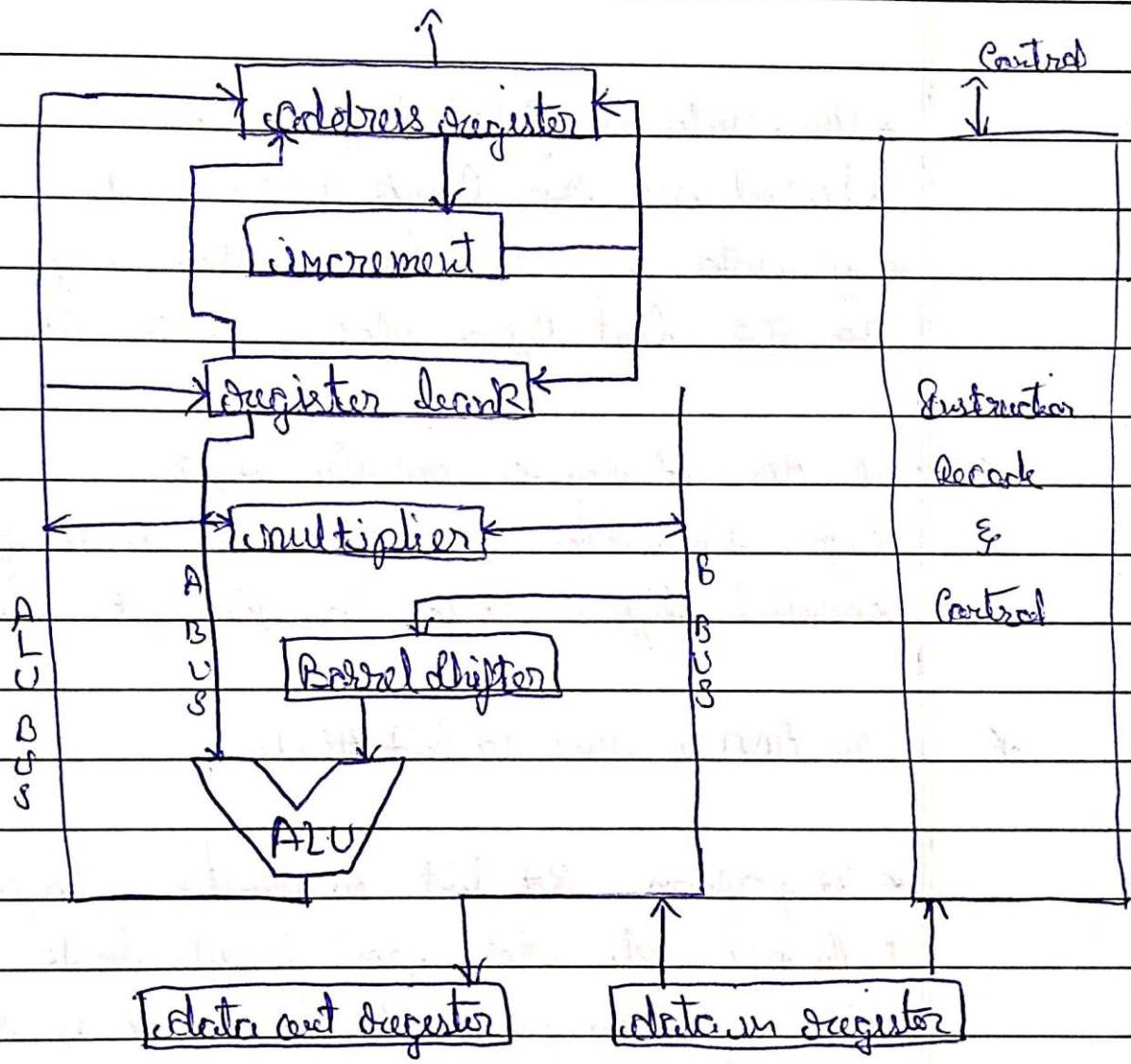
* Execution speed is comparitively slower.
* Comparitively fast

2) List differences RISC v/s CISC m/c.

RISC	CISC
* Instruction set simple & small	* Instruction set large & complex
* Less addressing modes	* More addressing modes
* Fixed no of operands	* Variable no of operands
* Mostly for single Tasking	* Mostly for Multi Tasking
* Execution speed is comparitively faster	* comparitively slower.
* High degree of pipelining	* Less degree of pipelining
* compiler design is simple	* Compiler design is complex.

3. With a neat diagram. Explain ARM7 architecture

→



* ARM7 processor has a 32 bit address bus & the PC of the RB gives 32 bit address of the instruction to be fetched. 32 bit data bus used to fetch both instruction and data.

* It is based on Von Neumann architecture.

* The instruction is fetched from memory through instruction decoder, that decode the instruction.

* The data is fetched from memory, then it get stored in reg bank (R0 - R15).
* If data size is lesser, then sign extended it to 32 bit before placing into RB.

* The ARM 7 has a Barallel shifter.
Before the data gives to ALU, it is given to Barallel shifter used to preshift operands.

* The ARM 7 has 32 bit ALU.

* It perform 32 bit Arithmetic & Logic operation.
* The ALU gets data from register bank through A Bus, after operation performed, the result is stored back to register through B Bus & status flag of CPSR are updated correspondingly.

* Multiplier is used to perform complex Arithmetic operation like x^{th} & $\sqrt{x}$.

* The address reg holds address of memory operand during load & store inst.

papergrid
Date: / /

① There are 7 modes of operation.
In each mode, programmer does access to 16 GPRS (R0 - R15) & a CPSR (Flag Reg). Additionally, a saved PSR (SPSR) is also available in all modes except user / System mode. All Reg are of 32 bits. In all modes the 3 Reg have common function.

② R15 : PC (Program Counter):
PC gives a 32 bit address of instruction to be fetched.
This address is put on address bus & instruction is fetched from memory through 32 bit data bus.
PC is periodically incremented after every instruction is fetched.

③ R14 : LR (Link Register)

It stores return address when we perform a BL instruction. While returning back to main program value of LR will be put back into PC.

④ R13 : SP (Stack pointer):
Hold the address of the top of stack.

(5) Additionally there are some Banked registers that are available on specific operating modes.

(6) FIQ mode:
* R8 - R14 provide fresh set of GPRs
* This eliminates need to store original values of these registers elsewhere & hence make execution of ISR faster.

(7) All other privileged modes get their own version of SP & LR

(8) This gives a total of 37 register all 32 bit each.

6. With a neat sdig, Explain 3 stages pipeline of ARM.

-> * Pipelining is the mechanism used by RISC processor to execute instruction

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

papergrid
Date: / /

* It is a design technique for processor which play an imp role in increasing efficiency of data processing in processor of computer & MC

ARM 7 has 3 stages pipeline

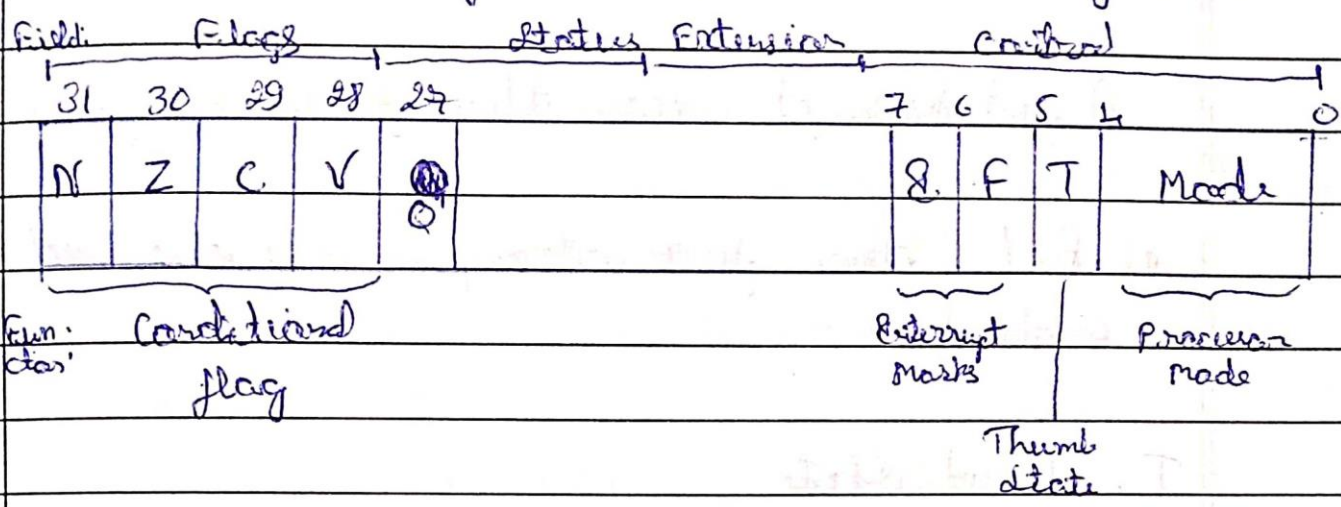
* Fetch : Instruction is fetched from memory
* Decode : Instruction opcode & operand are decoded to determine what has to perform.
* Execute: Decoded instruction is executed.

Each of these operation require one clock cycle for typical inst. Thus a normal instruction require 3 clock cycles to complete. execute shows the latency of instruction execution. Because it has 3 stage an instruction execution is completed in every clock cycle.

	Fetch	Decode	Execute
Cycle 1	ADD		
Cycle 2	SUB	ADD	
Cycle 3	CMP	SUB	ADD

Time

7. With neat diagram. Explain CPSR register.



Conditional flags:

V → ALU operation is overflowed
C → It is set when carry after MSB in ALU operation
Z → Zero result from ALU.
N → Negative result from ALU.

I: Interrupt Request mask:

If I = 1 normal interrupts are disabled else enabled.

A normal interrupt occurs through nIRQ pin.

Date: / /

F: Fast Interrupt mask:

A fast Interrupt occurs through nFIQ pin.

If F=1, then fast interrupts are disabled else enabled.

T: Thumb State

It is a special state where 'Thumb' instruction set having 16 bit instructions are used. This is useful if memory is implemented as a 16 bit memory.

If T=1, processor is in Thumb state else in normal ARM7 state.

Bit 27: Q flag:

It indicates if a saturation has occurred or not.

If Q=1 saturation occurred else not occurred.

Date: / /

8. Explain different Modes in ARM.

→ *) User Mode:

* It is normal mode in which all user programs are executed.
* It has limited access to memory, I/O components & flags.
* It is only non privileged mode.

*) Fast interrupt Request Mode:

It support a data transfer or channel process. This mode is entered when high priority interrupts occurs through nFIQ pin.

*) Interrupt Request Mode:

It is a normal Interrupt Mode & is invoked when a low priority interrupt occurs on nIRQ pin.

*) Supervisor mode:

ARM7 enter this mode on RESET used to execute BIOS programs.

*) Abort Mode:

This mode is entered when an unsuccessful attempt is made to access a memory location.

and the program that tried to access this location is aborted.

*) Undefined Mode:

It is used to enter undefined instructions. When the processor enters the undefined mode, the program exception in the instruction is terminated.

*) System Mode:

It is the privileged version of User Mode. User can invoke this mode to gain full control over CPSR & Memory.

3. Explain the nomenclature in ARM.

→ It is originally from ACRON Computers Ltd, It is the 1st RISC processor for commercial use.

ARM7 TDMI is the 32 bit processor where T indicates Thumb Architecture extension and D indicates Debug extension and M indicates Enhanced extension and I indicates Embedded emulation.

ARM [x][y][z] TDMI [E][J][F][S]

x → series
y → Memory Management unit
z → cache
T → Thumb instruction set
D → JTAG Debugger
M → Fast Multiplier
I → In-circuit Emulation
E → Enhanced instruction
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 system & it is available in nearly every MC & FPGA on market.

JTAG (Joint Test Action Group) is a industry standard for verifying design & testing printed circuit boards after manufacture

papergrid
Date: / /

It implements standard for on chip instruction in Electronic design automation (EDA) as a complementary tool to digital simulation. It specifies the use of dedicated debug port implementing a serial communication interface for low overhead access without requiring direct external access to system address direct lines. The interface connects to on chip TAP implements a stateful protocol to access a set of test registers

12 What is MMU ? Why MMU is required ? Give any of MMU

→ 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 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 they access should be in main memory during execution. The CPU fetches the instructions from memory acc to value of program counter (14)

Date: / /

Main memory is the place where program & information are kept when processor is effectively utilizing them. It is also known as RAM. This memory is a volatile memory RAM lets data when a power interruption occurs.

Why MMU 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

Ex: IBM System/360 Model 67, IBM System/370.

Write a C program to find endianness of given machine.

papergrid
Date: / /

```
#include <stdio.h>
int main()
{
    unsigned int i = 1;
    char *c = (char*) &i;
    if (*c)
        printf ("Little Endian");
    else
        printf ("Big Endian");
    getchar();
    return 0;
}
```

15. Explain
a) Bit b) Byte c) Nibble d) Halfword e) Word

→ Binary Values are often grouped into convenient lengths of 0's & 1's. This use of digits is called the length of a machine. Computer bit length of binary use include bits, nibble, bytes. Both 1 or 0 in a binary number is called a bit.

papergrid

Date: / /

Length	Name	Ex.
1	Bit	0
4	Nibble	1011
8	Byte	1011 0101
16	Half word	10110101 10010001

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

The term half word or single word are often used in contemporary to refer to common word e.g. relative to a 32 bit core each size.

half word = 16 bits

word = 32 bits

Date: / /

16. Explain the word align & half word align in ARM memory.

→ * Different processor have diff defn for a word.
* For a 32 bit processor a word is 32 bit
* As the name implies, a word is 16 bit for 16 bit processor.
* For 8 bit processor 8 bit means a word.

word alignment:

The stored address are aligned & can be divided by 4, the last 2 digits are 00

Half word alignment:

The stored address are aligned divisible by 2 that is the last bit is 0.

ARM architecture requires 32 bits ARM inst that must be word aligned & stored therefore in ARM stack the value of R15 is always divisible by 4 that is lowest 2 bits of the register are always 00.

papergrid

Date: / /

In the thumb state, the value of R15 is always %ed by 2 which is lowest bit of R15 register always one word consists of one more bytes i.e usually integer bits of bytes.

18. Explain the following addressing modes in ARM.
a) Three address b) Two address
c) One address instruction using ARM.

⇒ Sequence of instruction forms a program to perform a specific tasks

2 Components { Opcode field / Address field.

When data to be read from or store into 2 or more address field may have one or more than one address.

One address Instruction:

* This uses an implicit accumulator register for data manipulation and the other is in register or memory location.

* Implied means that the CPU already knows that one operand is in the accumulator so there is no need to specify it

ex: LDR addr
ACC ← (addr)

Two address instructions:

* In this the results are stored in one of the operands but here the results can be stored in diff location

It requires more no of bits to represent the address

ex: MOV R1, R2 R1 ← [R2]

Three address instructions:

* It has 3 address fields to specify a register or memory location
* Program created are much short in size but no of bits per inst increases
* These inst make execution of the program much easier but it doesn't means program will run much faster because here inst only contain

more if bit each micro operation will be performed in one cycle only.

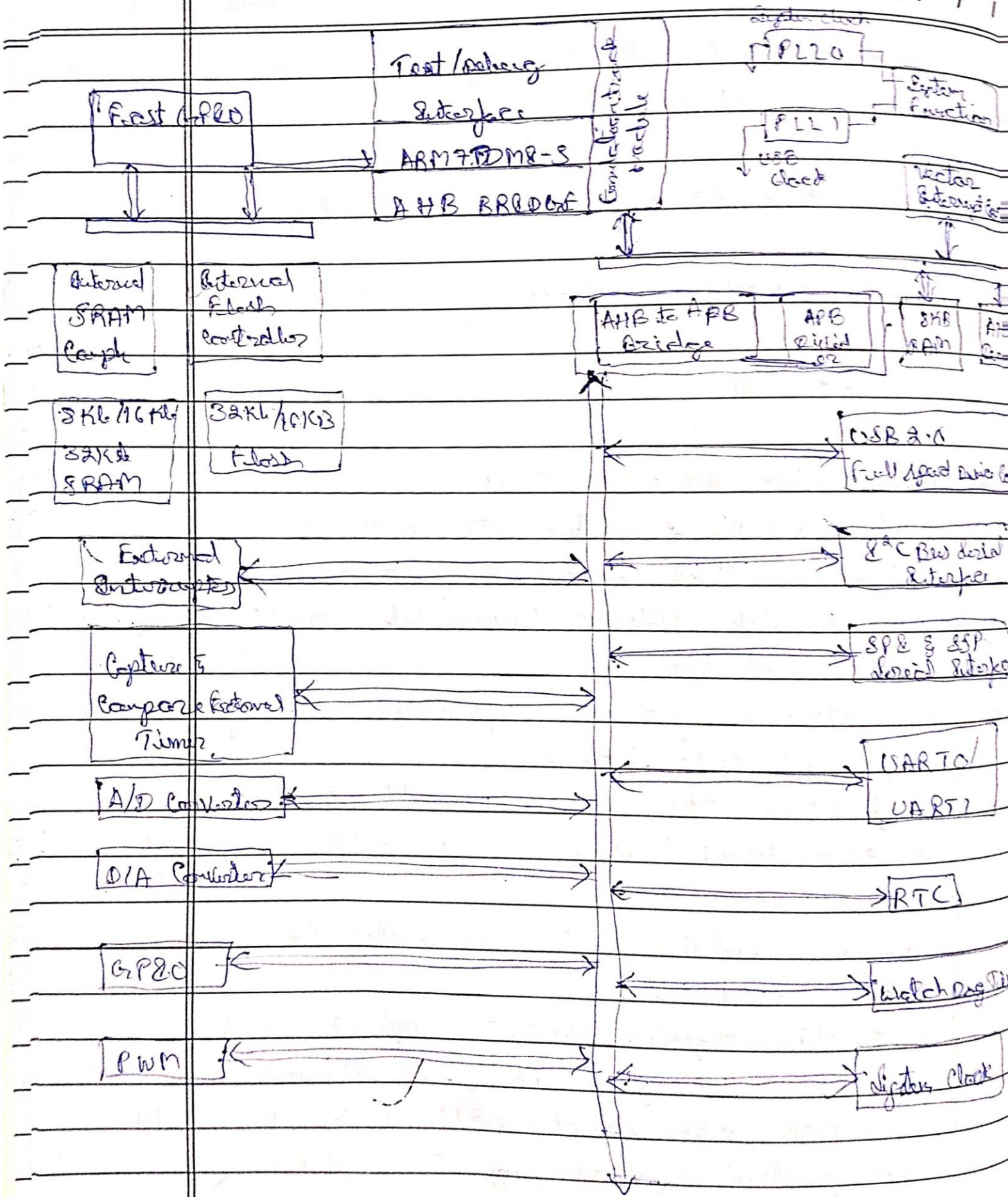
ex: ADD R3, R1, R2. $R_3 = R_1 + R_2$.

19. Explain LPC2148 MC block dig.

LPC2148 features:

* 16 bit / 32 bit ARM7TDMI → MC
* 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 / application programming via on chip boot loader software.
* Single 10 bit DAC provides variable analog O/P.
* 2 32 bit timers / external events PWM unit & watch dog timer.
* Low power real time clock with independent power & 32 KHz clock I/P.
* Two on chip integrated oscillators operates with an external crystal from 1 MHz to 25 MHz.
* Single RAM power supply with POR, BOD ckt. CPU operating V/g range of 3 V to 3.6 V.

Date: / /



20 Explain the LPC2148 MC GPIO

→ * GPIO → General purpose Input / output. A 32 bit register used to select the fun" of pins in which the user needs to operate.
* There are 4 fun" for each pin of the controller which the 1st fun" is GPIO.
* It means that the pin can either act as an I/P or O/P with no specific fun".

There are 3 pins register in LPC2148 controller in order to control the fun" of the pins in the respective ports. i.e,

PINSEL0 - controls fun" of port 0.0 to 0.15
PINSEL1 - control fun" of port 0.16 to 0-31
PINSEL2 - control fun" of port 1-16 to 1-31.

Port 0 & 1 are controlled via 4 group of 4 registers.
IOPIN, IOSET, IODIR, IOCLR.

IOPIN:

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

* The register will give the logic value of the pin regardless of whether the pin is configured for I/P or O/P or as GPIO or for alternate digital fun".

BOSET:

* This register is produced the high level O/P in the port pins configured as GPIO in the O/P mode.
* Writing 1 produces a high level at the corresponding port pin writing a 0 has no effect
* If any pin is configured as a I/P pin & it's fun", writing 1 to corresponding bit in BOSET has no effect
* If any pin is configured as an output pin & it's fun" writing 1 to corresponding bit in BOSET has no effect. Reading the BOSET returns the value of this register as determined by previous writes to BOSET return value of this register as determined by previous writes to BOSET & BOCLR. This value does not affect of any outside world influences on the I/O pins.

8051R:

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

8051R:

It is used to produce a low level o/p at port pins configured as GPIO in o/p mode. Writing 1 produce low level at corresponding port pin & clear the corresponding bit in the 8051SET writing 0 has no effect.

Q1. With a neat diag explain Baud rate & Bit rate Explain calculation.

→ Baud: Says how many times a signal changes per sec.

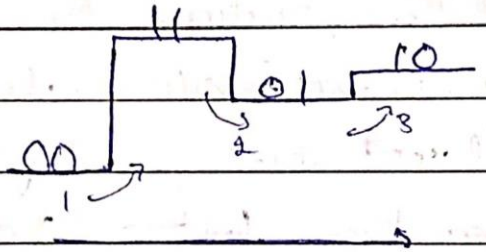
Bit rate: Says how many bits can be sent per unit time (per sec)

Bit rate is controlled by baud and no of signal levels

Date: / /

Baud rate:

* No of times lines changed per second
* Let Baud rate be 4
* Let bits per line change be 2
* Bit rate = 8 bit per sec.
* Bit rate = x2 Baud rate in this ex.



Bit rate = 2
Baud rate = 3

* Baud rate defines switching speed of a signal
* Bit rate defines the rate at which information flows across a data link measured in bit/sec
* For a binary 2 levels signals a data rate of one bit per sec is equivalent to one baud
* An analog signal carries 4 bit in each signal unit. If 1000 signal units are sent per sec. Find the Baud rate & bit rate

1 bit -> 1 baud.

1000 baud/sec = Baud rate

Bit rate = 1000 × 4 = 4000 bps.

* If bit rate is 'b' baud rate is 's'

General formula:

$$b = s \times n$$

b → Data rates
s → Symbol rate
n → no of bits per sec
If n = 1, Baud rate = bit rate
n = 4, Bit rate = 4 × Baud rate.

22. With the neat diag, explain the working features of SPI protocol.

→ * The serial peripheral interface (SPI) is a synchronous serial communication interface specification used for short distance in Embedded system.
The interface specification was developed by Motorola.

* SPI devices communication is full using a master slave architecture usually with single master

papergrid
Date: / /

The master device originates the frame for reading & writing. Multiple slave devices may be supported through selection with individual chip select, sometimes called slave select (ss) lines.

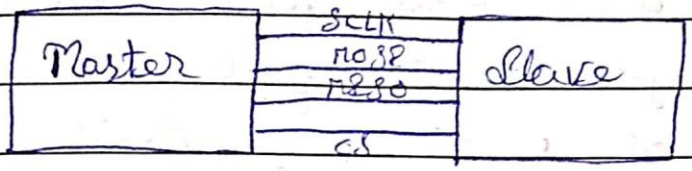
* SPI is called as 4 wired serial bus, contrasting with 3, 2 or 1 wired serial Bus

* The SPI may be accurately described as a synchronous serial interface.

* But it is diff from the synchronous serial interface (SSI) protocol. Each is also a 4 wire synchronous serial communication protocol

The SPI bus specifies four logic signals
SClk : Serial clock
MOSI : Master out slave in
MISO : Master in slave out.
$\overline{CS}$ or $\overline{SS}$: Chip select or slave select.

When $\overline{CS} = 0$ the master has selected the slave for communication.



Working:

Master will generate clock where ever it wants to write data to a slave device. After 8 clock pulses data in the master delivers and data in slave device (B7-B0) is transfered from the master device.

Features of SPI:

* Bidirectional serial data transfer
* Higher speed / data rates.
* Simple communication for selecting the slave

Q3. Explain SPI with a neat timing diag. explain CPHA & CPOL usage.

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

Date: / /

* CPOL=0 is a clock which idles at 0 & each cycle consists of a pulse of 1 that is leading edge is rising & trailing edge is falling edge.

* CPOL=1 is a clock which idles at 1 & each cycle consists of a pulse at 0 that is the leading edge is falling & trailing edge is raising edge.

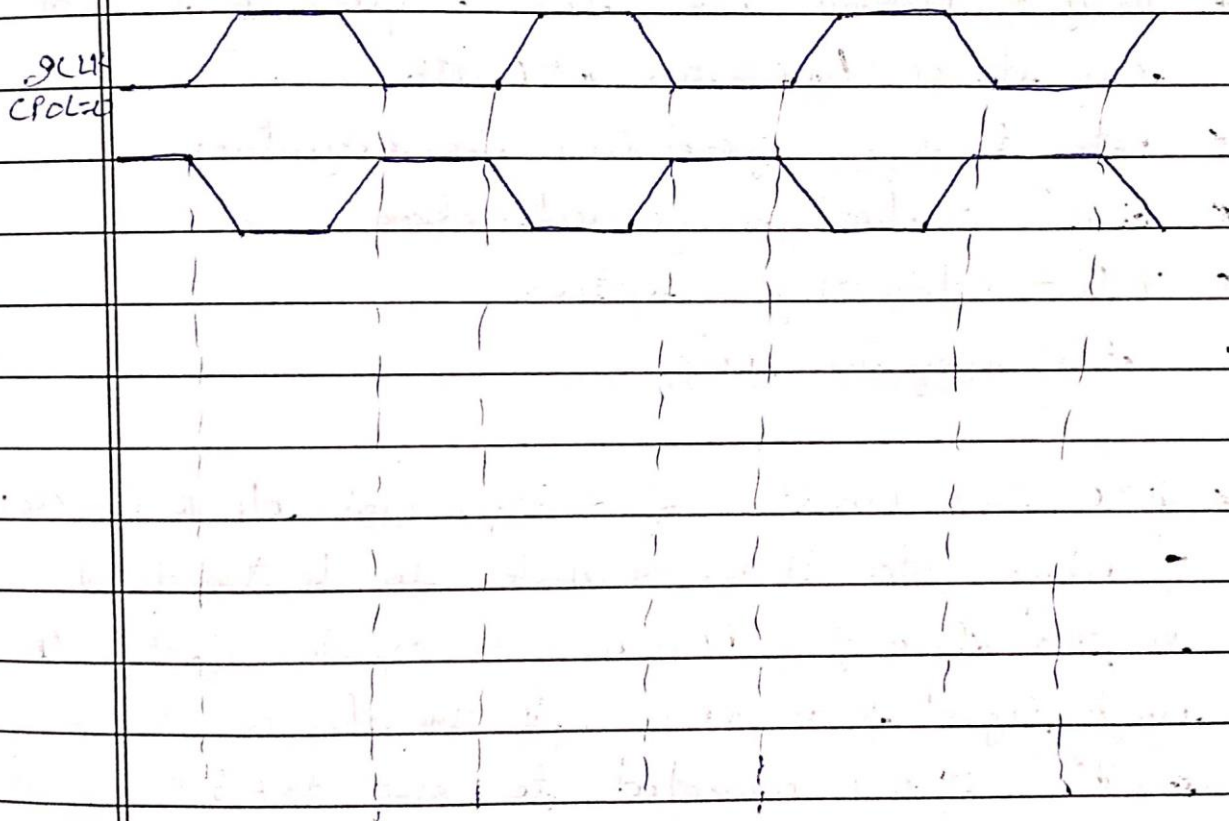
* CPHA determines the timing of data bits relative to the clock pulse conversion 0/1 then & zero is near trivial.

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

The master holds the data valid until the trailing edge of current clock cycle. For the 1st cycle, the 1st must be set MOSI time before leading clock edge. An alternative way of considering it is to say that clk idle, followed by a half cycle with clock asserted.

Date: / /

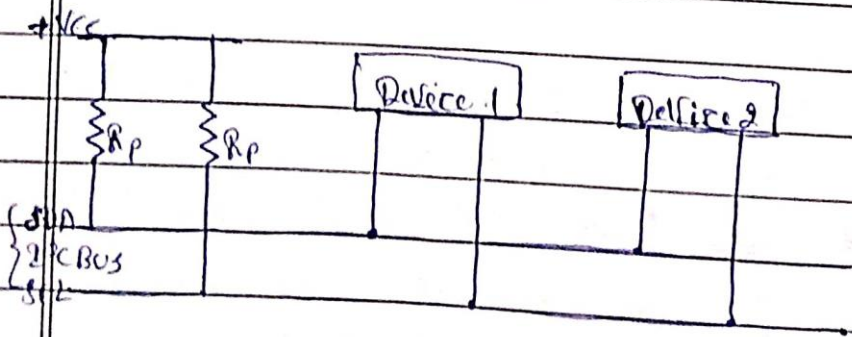
* The MOSI & MISO signals are usually stable for only until the next clock transition. SPI master & slave device may well sample data at diff points in half cycle.



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

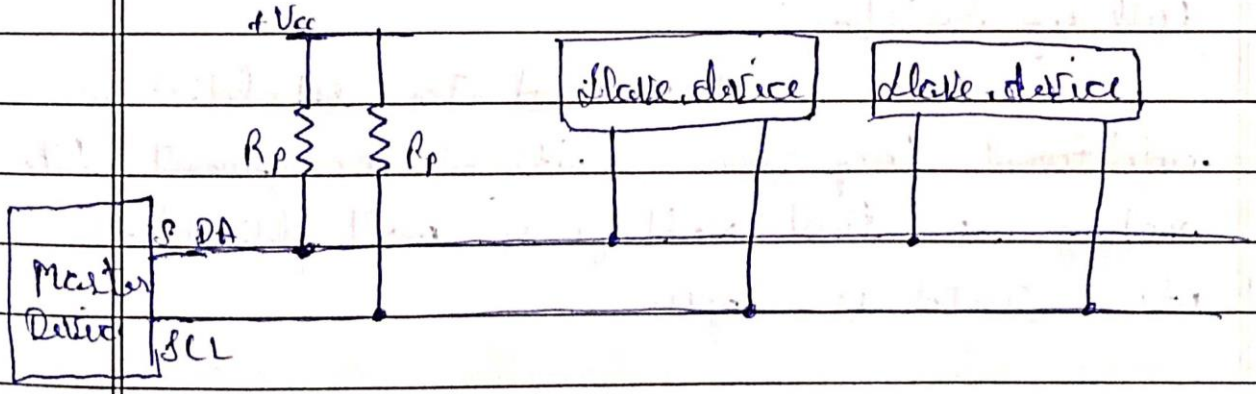
→ * Half duplex - Serial communication
* Synchronous communication protocol.
* Only 2 common bus line required to control any device IC on I²C network.
* Data transfer speed can be adjusted.
* Simple mechanism for validation
* 7 bit addressing system.
* It is easy to scale.

* I²C bus consists of 2 wire serial clock line (SCL) & serial data line. The data to be transferred is send through SDA wire & synchronized with clock signal from SCL. All the devices IC's on I²C network are connected to bus. SCL & SDA lines.



* Both lines are operated at an open drain driver. It means that any device on I2C with can drive SDA & SCL down but they cannot drive them high. So a pull up resistor is used for each bus line to keep them high by default.

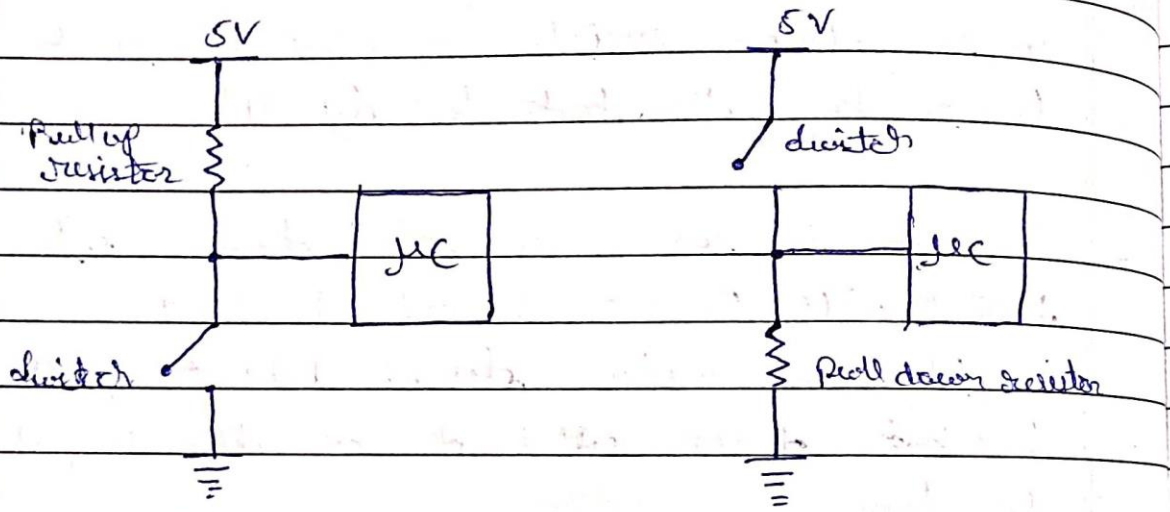
* The reason for using an open drain system is that it will, be no chances of shorting, which might happen when device tries to pull the line high of some other devices that to pull the line down.



* The devices that are connected to I2C bus are either categorized as master or slave.
At any instance of time only a single master stays active in it. It controls SCL clk line & decides what operation has to be done on SDA -> data line.

25 With neat diag explain pull up & pull down resistor

→



Pull up resistor:

It is used to establish an additional loop over critical component while making sure that voltage is well defined even when switch is open.

It is used to ensure that a wire is pulled to a high level in absence of I/P signal It is with a fixed value was used to connect Voltage supply & A particular pin in digital ckt.

papergrid

Date: / /

Pull down resistor:

* It is used to ensure that I/P to logical system settle at expected to logical system 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. It holds the logic signal near to OV when no other active devices are connected.

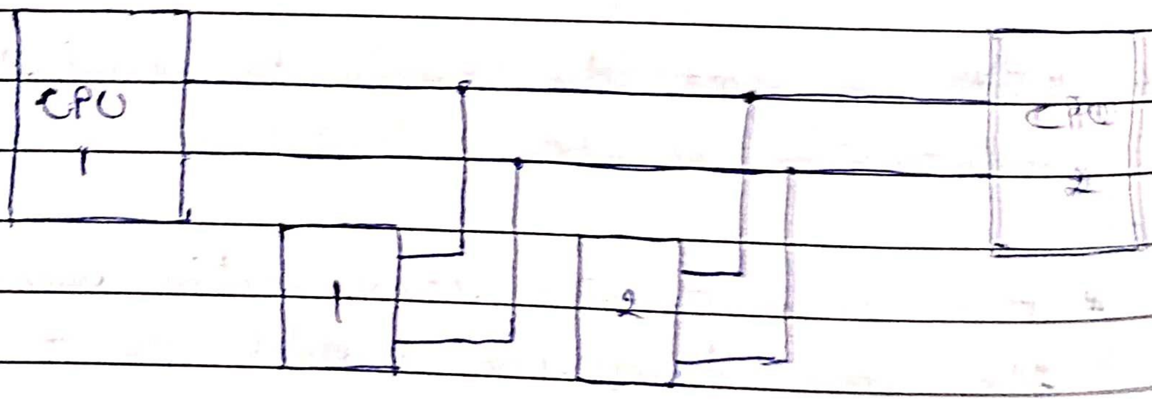
26. With a neat diag explain the concept of arbitration in I2C.

→ * It is a multimaster communication meaning more than one device can initiate the transfer but not at once.

* Bus arbitration happen when 2 or more master start to transfer at same time.

* When MCU issues the start condition and sends the address, the slave will listen. If the address doesn't match address of CPU 2, this device has to hold back any activity until bus become idle again after a stop condition.

* As long as the 2 MCU's monitors 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.



Slaves.

Date: / /

Q7. What is clock streching Explain clock stretching in I²C.

→ * It allows any I²C slave device to force the master device into waiting state.

* A slave device may perform clock stretching when it needs more time to manage data such as store, receive data or prepare transmit another byte of data

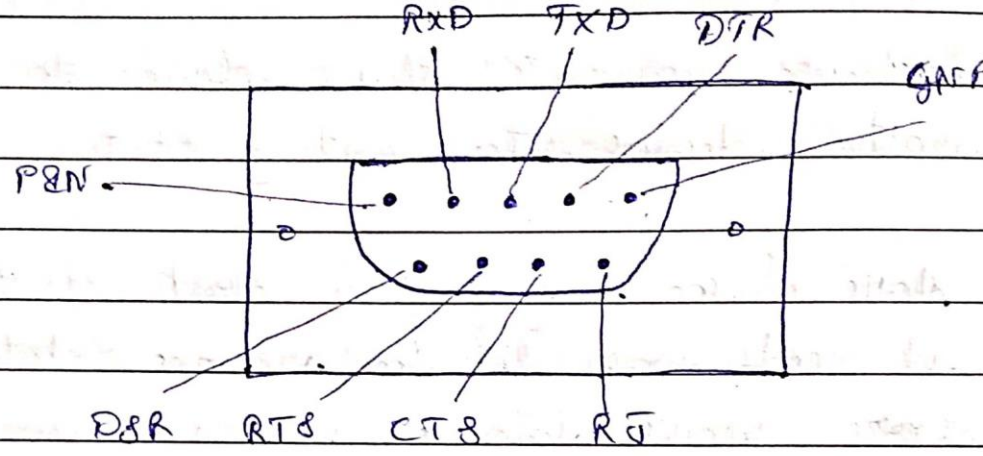
* Clock Stretching in I²C can slave down communication by stretching SCL. During any SCL low phase any I²C device on bus may additionally hold down SCL to prevent it to raise again. Enabling them to low down SCL clock rate or to stop I²C communication for a while. This is also refered to as clock synchronization.

* In an I²C communication master device determines clock speed which allows master & slave from synchronization exactly to a predefined Baud rate.

Date: / /

28. 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	Reciver data	In
3	TXD	Transmitter 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

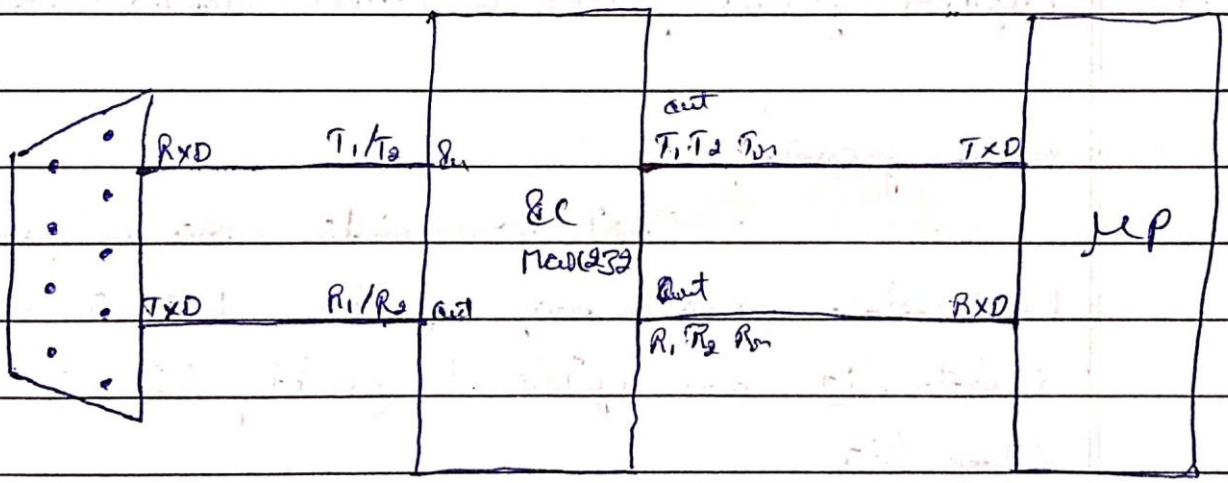
Handshaking Modem:

* When receiving modem answers the phone call of modems begins to communicate

* Before anything else happens, the modems must evaluate the quality of line, negotiate error control protocols & data compression that they can both recognise & work out what the most suitable communication speed should be based on condition. It is called as handshake.

29. Explain RS232 communication with a µC.

→



Serial device collect data from device and need to send it to another unit like a computer for further processing. Data transfer is generally done in 2 ways is fast & uses more no of lines.

Serial communication or other hand use only one or 2 data line transfer data & is generally

used for long distance communication. In serial communication the data is sent bit by bit at a time.

* In this parameters considered while transferring signal port is baud rate which is speed at which data is transmitted serially and receive signal data at diff baud rate using software instructions.

30 Explain frame format in UART communication.

Data framing:

UART transmits data in packets. Each data packets may contain 1 start bit, 5 to 9 data bit and optional parity bit & 1 or 2 stop bit.

Start bit	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	PB	Stop Bit

* The UART received data from data bus & this data are being by CPU memory or I/O

* The data transmission from the data bus to UART

is in I^2C mode.

* UART adds the start bit, parity bit & stop bit to data received from data bus which create packets.

* This packet is then transmitted serial to receiving UART receiver.

* At receiver the UART receiver reads data bit by bit. This serial data is again converted into I^2C form at the receiver & then sent to data bus.

* At receiver the receiver UART controller removes all the start bit stop bit and if included parity bit to check is after signal has been modified.

Start Bit:

* When there is no data transmission the UART transmission line is held at high voltage. Its transmission ends at the start bit. When the receiving UART detects the high to low voltage transmitter it begins reading the data frame at the frequency of baud rate.

Data frame:

The data bit are usually 5 to 8 in no. If no parity bit is used, it can be 9 bit long. In general case the LSB of data is transmitted first.

Parity bit:

It is used to indicate the change in data during transmission. The reasons for the change in data is mismatched baud rates at long distances data etc.

Stop bit:

* To mark the end of the data packet the sending UART drives the data transmission line from a low voltage to a high voltage for a min of 2 bit durations

31. Explain the difference b/w

a) Serial v/s Parallel
b) Analog v/s Digital
c) Synchronous v/s Asynchronous

Serial	Parallel
* Data is sent bit by bit in single line	* Data is transmitted simultaneously through group of lines
* Transmission speed is low	* Transmission speed is high
* Software implementation	* Hardware implementation
* No cross talk issue	* Cross talk creates interfaces b/w n lines.
* Bandwidth is greater	* Lower Bandwidth

Date: / /

	Analog	Digital
*	Transmitted modulated signal is analog in nature	* Transmitted signal is digital i.e train of digital pulse
*	Message is rep uniform of phase, freq or amplitude variation	* Amplitude, width is position is transmitted pulse is in the form of code words.
*	Noise immunity is poor	* Excellent noise immunity
*	Coding technique are not possible	* Coding techniques are to detect, correct the
*	FDM is used for multiplexing	* TDM is used for multi
*	ex: AM, FM, PM	* ex: PCM, DM, ADM

	Synchronous	Asynchronous
*	Communication is in real time	* Not in real time
*	Creates interrupts while creating communication	* No interrupts
*	Faster communication	* Slower communication
*	No overhead wrt start & stop bits	* uses start & stop bit
*	Uses constant interval of time	* Uses irregular amount of time
*	ex: Video conferencing chat rooms	* Sending emails.