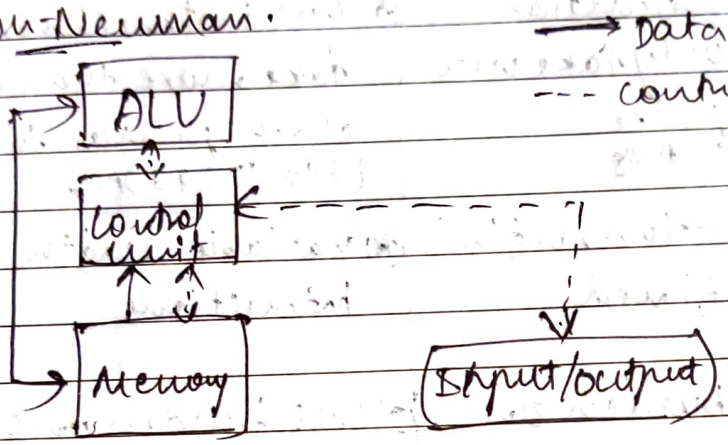
Date
Page
SPLASH

MCA
Assignment-1

(1) With a neat Diagram explain Von-Neuman & Harvard Architecture.

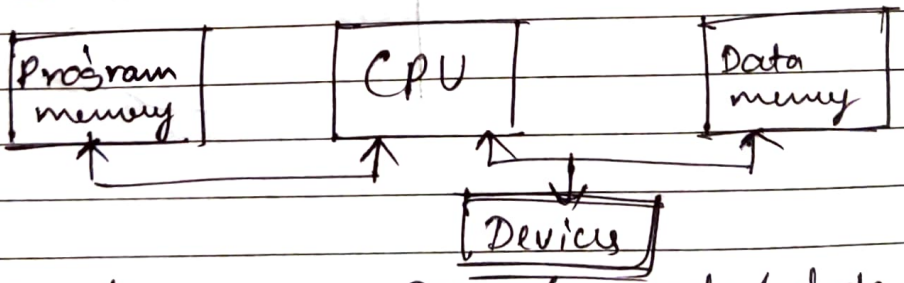
Sol:

Von-Neuman.



- Same memory space is shared b/w program & data in Von-Neuman.
- Single bus is sufficient since data & programs cannot be fetched at a time.
- Cost effective architecture.

Harvard Architecture



- Seperate memory space for code & data
- Additional bus is required
- Cost is more due to seperate bus for code & data.
- Free data in code / program is only specific to code / data mem respectively

Date / /
Page
SPLASH

② List the difference b/w RISC and CISC

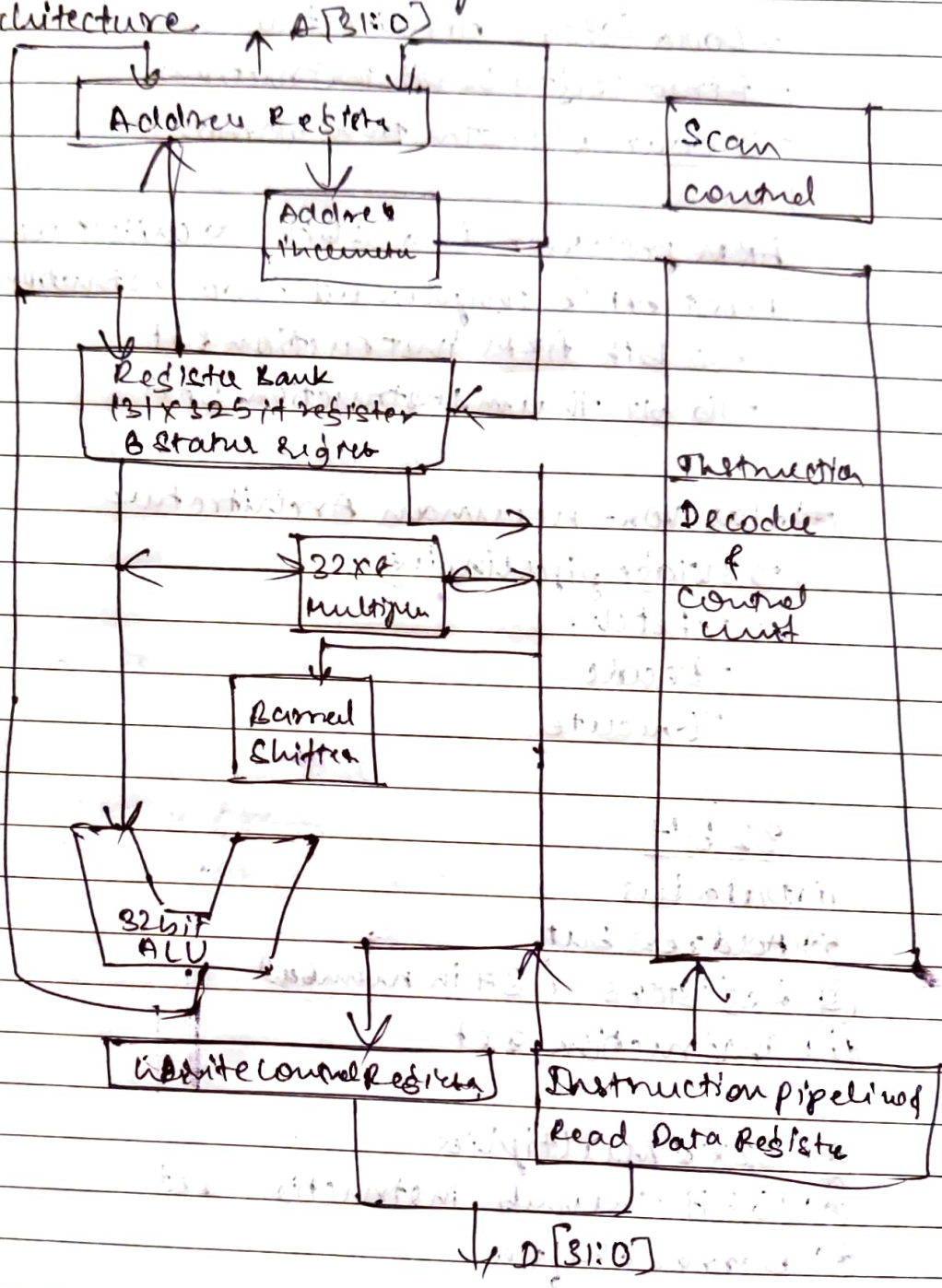
Sol:-

RISC	CISC
★ Instructions are single	★ Complex instructions
★ One instruction will only take once clock pulse (1.5 CPI) Avg	★ One instruction will take more than one clock pulses to execute (2 to 15 CPI) Avg
★ Instruction length is ~~sim~~ fixed.	★ Variable length instructions
★ More burden on compiler ★ Some example has to be difficultly implemented	★ Less burden on compiler Eg:- SQRT is in its instruction set only
★ RAM burden is very less	★ More extensive RAM usage.
★ Code density is more	★ Code density is less

Date __/__/__
Page _____
SPLASH

(5) With a neat diagram explain the ARM7 architecture.

Sol:



PTO.

Features used
→ Load/Store Architecture
- Fixed length 32 bit instructions
- 3 address instruction formats

ARM processor is a 32 bit architecture
Most ARM's implement two instruction set
- 32 bit ARM instruction set
- 16 bit thumb Instruction set

* Has Von-neuman Architecture
→ 3 Stage pipelining
→ Fetch
- Decode
→ Execute

32 bit
1. Data bus
2. Address bus
3. Registers (37 in number)
4. Instruction set

* 32x8 Multiplier
* 16 bit Thumb instruction set
* Barrel shifter

Date / /
Page
SPLASH

Data types

ARM processor supports 6 data types
- 8 bit signed & unsigned bytes.
- 16 bit signed & unsigned half word alligned on 2 byte boundaries.
- 32 bit signed and unsigned half words.

ARM instructions are all 32 bit words alligned thumb instructions are half words alligned on 2 byte boundaries.

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

ARM co-processors supports floating point values.

(4) programming model for ARM 7

Sol: Each instruction can be viewed as performing a defined transformation of the states.

- Visible register
- Invisible registers
- System memory
- User memory

Processor modes

→ ARM has some basic operating modes

→ Modes change by software control or external interrupts.

CPSR[4:0]	mode	use	Registers
0000	User	Normal user code	R0-R14
0001	FIQ	processing fast interrupts	R8_fiq-R14_fiq
0010	IRQ	processing standard interrupts	R13_irq-R14_irq
0011	SVC	processing software interrupts	R13_svc-R14_svc
0100	Abort	processing memory faults	R13_abt-R14_abt
0101	Undef	handling undefined instructions	R13_und-R14_und
0111	System	Running privileged OS tasks	R0-R14

Date __/__/__
Page _____
SPLASH

→ Most programs operate in user mode ARM has other previlaged operating modes which are used to handle exceptions. Supervisor called & system mode.

→ More access rights to memory systems & coprocessors
→ Current operating mode is defined by CPSR[4:0].

Privilaged Modes

Supervisor modes:

→ Have some protective previlages.
→ System level functions can be accessed through specified supervisor calls.

→ Usually implemented by software interrupt.

ARM has 37 registers all of which are 32 bits
→ 1 program counter
→ 1 current program status register
→ 5 dedicated saved program status registers.
→ 30 general purpose register.

Each mode can access
a particular set of R0-R12 registers
→ Stack pointer: R14 - Link register
→ Program counter r15 (PC)
→ the current program status register CPSR

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

[R0 to R15 + CPSR] → User mode

FIQ	IRQ	SVC	Undef	Abort
r8				
r9				
r10				
r11				
r12				
r13 (SP)	r13 (SP)	r13 (SP)	r13 (SP)	r13 (SP)
r14 (LR)	r14 (LR)	r14 (LR)	r14 (LR)	r14 (LR)
SPSR	SPSR	SPSR	SPSR	SPSR

Date
Page
SELASH

(6) With a neat diagram explain 3 stage pipeline of ARM.

→ Pipelining is the mechanism used by RISC & CISC processor to execute instructions.

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

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

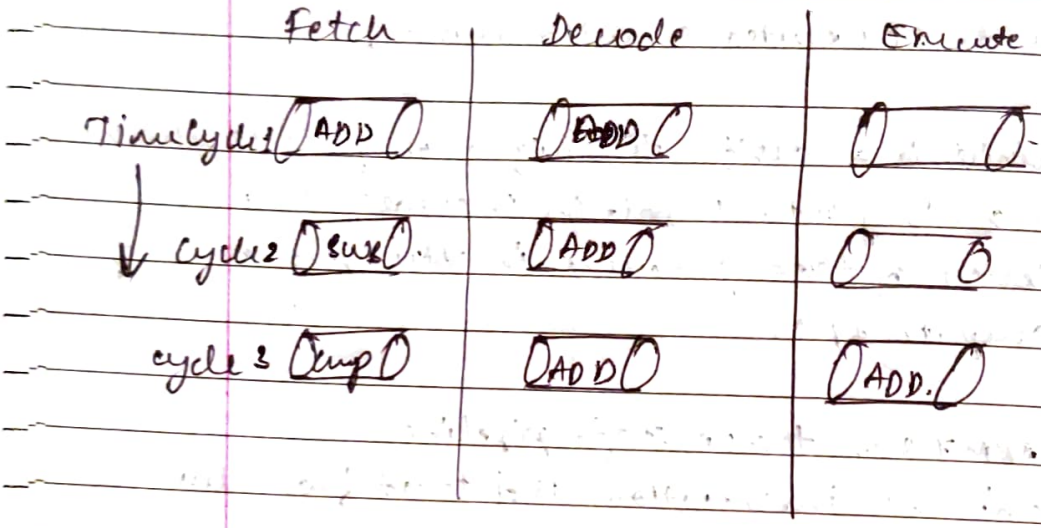
→ The ARM7 has three stage pipeline.

- Fetch: The instruction is fetched from the memory.
- Decode: The instructions opcode & operands are decoded to determine what function to perform.
- Execute: The coo 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 on instructions execution computed in every instructions execution computed in every clock cycle. In other words the pipeline has a throughput of one instruction per cycle.



Date __/__/__
Page _____
SPLASH

(8) With a neat diagram explain CPSR register.

CPSR - Current program status Register

N Z C V Q undefined ĪF T 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 STE only
- Indicates if Saturation has occurred during certain operations

Interrupt Disable bits.
Ī = 1 Disable the ĪRQ
I = 0 disables the ĪRQ

T - bit
- Architecture xT Only
- T = 0 processor in ARM State
- T = 1 in thumb state

Mode bits.
- Simply the processor mode

CPSR → holds the information about the current State of the processor

SPSR → Saved processor state register
holds the information on the processor state before the system changed to the mode.
ie. processor State just before an exception

(8) Explain the seven different modes in ARM

Sol:-

ARM7TDMI has 7 modes of operation

* Usermode
  - used ARM programs execution state and is used for executing most application programs.

* Fast interrupt (FIQ) mode supports a data transfer or channel process.

* Interrupt (IRQ) mode is used for general purpose interrupt handling

* Supervisor mode - is a protected mode for the O.S.

* Abort mode is entered after a data or ~~interrupt~~ instruction prefetch Abort

* System mode is a privilaged user mode for the O.S.
  * we can only enter System mode from another privilaged mode by modifying the mode bit of the CPSR.

* Undefined mode is entered when an undefined instruction is executed

Date / /
Page
SPLASH

Modes other than user mode are collectively known as privilaged mode. Privilaged modes are used to service interrupts or exception or to access protected resources.

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

(3) Explain the nomenclature in ARM.

ARM was originally from Acron computers Ltd. First RISC processor for commercial use

ARMTDMI

32 bit processor (Advance RISC Machine)
T → Thumb Architecture enextension
D → Debug extension
M → Enhanced extension
I → Incircuit Emulation

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

x → Series	I → Embedded ICE (Incircuit Emulation)
y → Memory management unit	E → Enhanced instruction for DSP
z → Cache	J → JAVA acceleration for DSP
T → Thumb 16 bit decoder	F → Floating point
D → JTAG Debugger	S → Synthesizable version
M → fast multiplier	

(10) What is JTAG? Explain State Diagram.

Ans:- *) JTAG has become a standard in embedded systems and it is available in nearly every microcontroller & FPGA on the market.

*) If we have programmed a microcontroller there is a strong chance that we have used JTAG of the relatable standards.

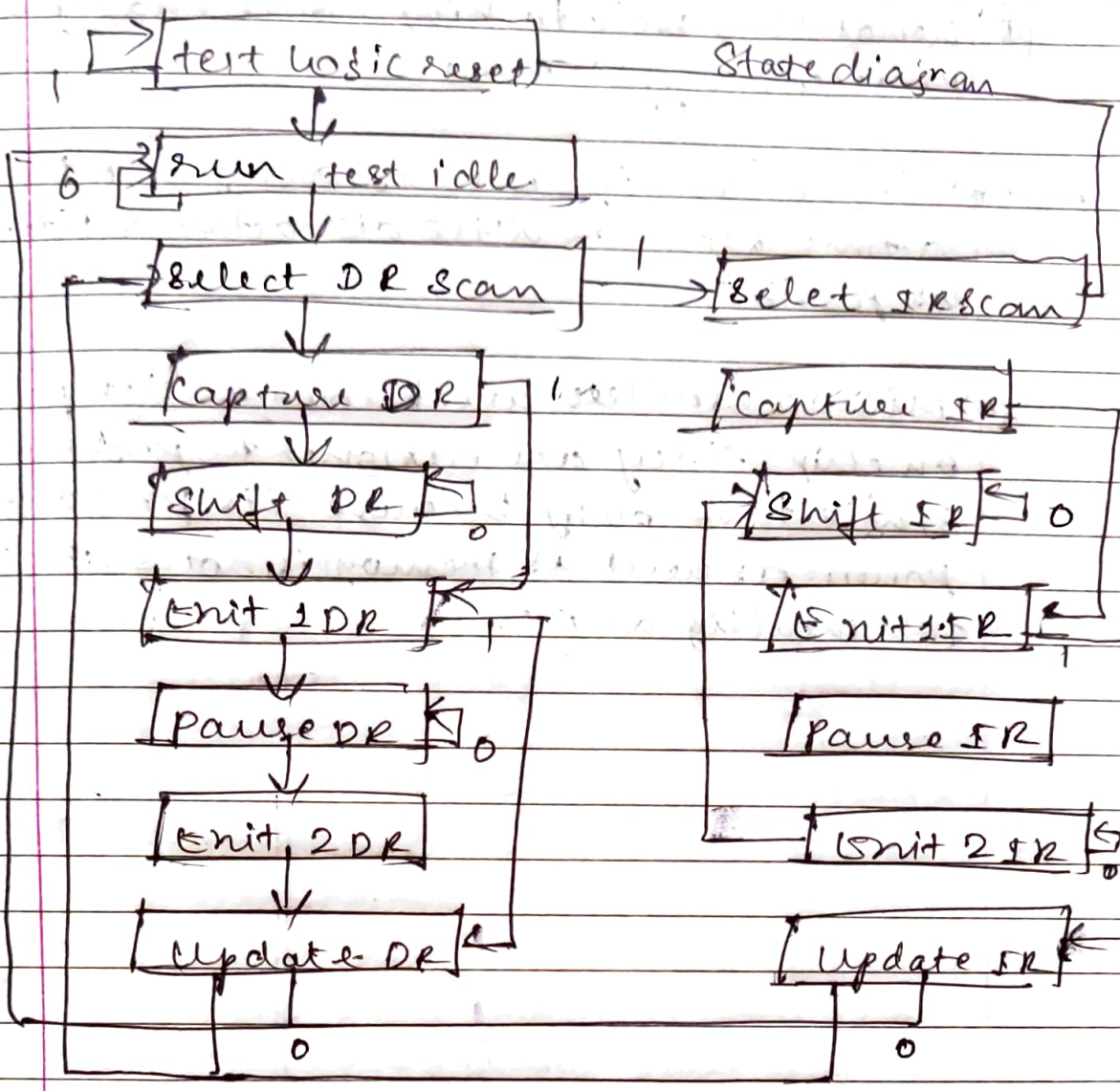
(*) JTAG - Joint test Action Group. is an Industry standard for verifying designs & testing PCB's after manufacturing.

(*) JTAG implements standards for on chip instrumentation in electronic design automation (EDA). as a complementary tool to digital Simulation.

It specifies the use of a dedicated debug port implementing a serial communications interface for low-overhead access without requiring direct external access to the system address & 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.

Date __/__/____
Page _____
SPLASH

State diagram



(11) What is Single Tasking? Give example processor applications.

Ans: Single tasking means doing one task a time with as little distraction & interr as possible

Microcontrollers are known as Comput on chip. They are designed to perform a single task only because its process power as well as memory is not suitable for installing an O.S..

Date __/__/__
Page _____
SPLASH

(12) What is MMU? Why MMU is required? Give uses of MMU support.

Ref: ★ 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 the main memory during execution. The CPU fetches the instructions from the memory according to the value of the program counter.

★ The main memory is central to the operation of the computer. Main memory is a large array of words & bytes, ranging in size from hundreds of thousands to billions.

★ Main memory is a repository of rapidly available information shared by the CPU & I/O devices.

★ Main memory is the place where programs & information are kept when the processor is effectively utilizing them. Main memory is associated with the processor & extremely fast

Date
Page
SPLASH

- Main memory is also known as RAM (Random access memory). This is a volatile memory. means RAM's contents are lost when a power interruption occurs.

Memory Management

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

Why memory management is required?

- Allocate & de-allocate the memory before and after the process execution.
- To keep track of the used memory space by process.
- To minimize the fragmentation issue.

Date / /
Page
SPLASH

④ To proper utilization of main memory.

⑤ To maintain data integrity while executing of process

Eg IBM system/360 Model 67, IBM system/370.

ARM

① ARM architecture based application processor implement an MMU designed by ARM's. virutal 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 1KB tiny space.

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

Sol

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

(15) Explain
(a) Bit (b) Byte (c) Nibble (d) Halfword (e) word

Binary values are often grouped into common length of 0's & 1's this number of digits is called the length of a number. Common bit length of binary numbers include bits, nibble, bytes Each 1 or 0 in a binary number is called a bit.

Length	Name	Example
1	Bit	0
4	Nibble	1011
8	Byte	1011 0101
16	Half word	1011 0101 1001 0011

Word is another length that gets thrown around from time to time, word is much less sturdy more ambigous. The length of a word is usually dependant on the architecture of a processor. It could be 16 bits, 32 bits, 64 bits or even more.

The terms half word or single word are often used in contemporary computing to refer to common word size relative to a 32 bit base code size.

half word = 16 bits
word = 32 bits

(16) Explain the word alisn & half word align in ARM memory.

Sol. Different processors have different definitions for a word.

- For a 32 bit processor a word is 32 bit (4 byte)
- 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 adjacent & can be divided by 4, the last two digits are 00.

Half word alignment: That is the stored address are adjacent & divisible by 2 that is the least bit is 0.

ARM architecture requires 32 bit ARM instructions that must be word aligned & stored in memory & 16 bit thumb instructions require half word aligned & stored Therefore in ARM state the value of R15 is always divisible by 4 that is lowest 2 bits of the R15 register are always 00.

In the thumb state, the value of R15 is always divisible by 2. which is the lowest bit of the R15 register always 0.

One word consists of one 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

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

2 components / Op code field → specifies how data manipulate
Address field - Specifies the data location

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

(*) Three address instructions
(*) Two address Instructions
(*) One address Instructions
(*) Zero address Instruction

(*) Processor can execute an instruction only if it is represented in Binary sequence

(*) Unique Binary sequence pattern must be assigned. This process is called op-code encoding

Date __/__/__
Page _____
SPLASH

One address instructions.

② This uses an implied accumulator register for data manipulation and the other is in the Register or memory location

③ 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

④ In one address instructions the results are stored in only accumulator.
but here the results can be stored in different locations.
ie. Register or memory location.

But requires more number of bit to represent the address

eg MOV R1, R2
R1 ← [R2]

Date
Page
SPLASH

Three address instructions

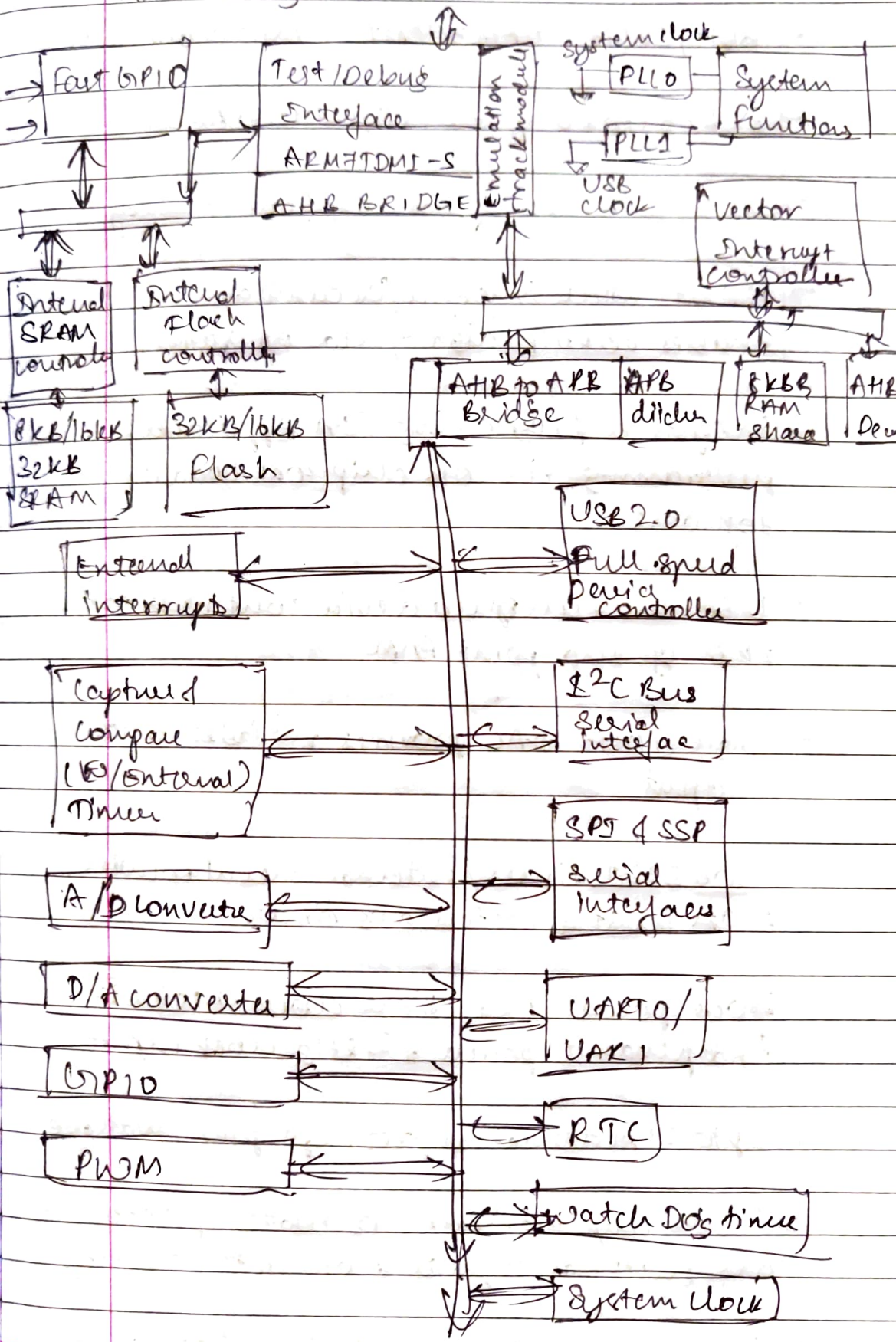
- This has three address field to specify a register or memory location.
- Program created are much short in size but number of bits per instruction increases.
- These instructions make creation of program much easier but it does not mean the program will run much faster because these instructions only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

Ex:- ADD R_3, R_1, R_2

$R_3 = R_1 + R_2$

Date / /
Page
SPLASH

(9) Explain the LPC2148 Microcontroller. Block diagram.



Date / /
Page
SPLASH

LPC 2148 features

- 16bit / 32 bit ARM7TDMI-S Microcontroller
- 8kb to 40kb of on chip static RAM
- 32kb to 512kb of on chip flash memory
- 128 bit wide Interface/accumulator enables 60MHz high speed operation
- In system programming (ISP) Application programming via on chip boot loader software.
- USB 2.0 full speed device controller with 2kb of end point RAM
- Single 10bit DAC provides variable analog output
- Two 32bit timers / external event counter, PWM unit & watch dog timer
- Low power Real time clock (RTC) with independant power & 32kHz clock input
- Upto 21 external interrupt pins available
- The on chip integrated oscillator operates with an external crystal from 1MHz to 25MHz
- Single power supply with POR, BOD circuits CPU operating voltage range of 3.0V to 3.6V

Date / /
Page
SPLASH

(20) Explain the LPC2148 microcontroller GPIO pins.

Sol:- ⊛ GPIO → General purpose Input/output
A 32 bit register used to select the function of pins in which the user needs to operate

⊛ There are fourse functions (max) of each pins 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 is to actually three PIN register in LPC2148 controller in order to control the functions of the pins in the respective ports. The classification is given below.

PINSEL0 - controls function of port 0.0 - port 0.
PINSEL1 - controls function of port 0.16 - Port 0.31
PINSEL2 - controls function of port 1.16 - Port 0.31

Lower two 00 of 32 bits register are GPIO configuration

⊛ LPC2148 has two 32bit General purpose I/O ports.

⊛ A total of 30 I/O and a single output only pinout of 32 pins are available on PORT0. PORT1 has upto 16 pins available for GPIO functions. PORT0 & PORT1 are controlled via two groups of 4 registers.

Date
Page
SPLASH

* 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 whether the pin is configured for input or output or as GPIO or an alternate digital functions.

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

→ IOSET

* This register is produce the 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 a 0 has no effect.
* If any pin is configured as an input or a secondary function, writing 1 to the corresponding bit in the IOSET has no effect.

Date __/__/__
Page _____
SPLASH

If any pin is configured as an input or a secondary function, writing 1 to the corresponding bit in the FIOSET has no effect. Reading the FIOSET register returns the value of this register as determined by previous writes to FIOSET register. Returns the value of this register, as determined by previous writes to FIOSET & FIOCLR. This value does not reflect the effect of any outside world influence on the I/O pins.

FIODIR

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 function ability.

FIOCLR

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 & clear the corresponding bit in FIOSET register. Writing 0 has no effect. If any pin is configured as an input or a secondary function, writing to FIOCLR has no effect.

Date / /
Page
SPLASH

(21) With a neat diagram explain Baud rate & Bit rate. Explain calculations

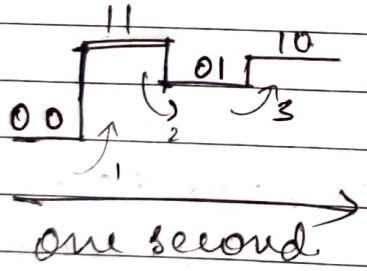
Baud:- says how many times a signal changes per second.

Bit rate:- says how many bits can be sent per unit time (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 = x2 Baud rate in this example



Bit rate = 2
Baud rate = 3

→ Baud rate defines the switching speed of a signal.

→ Bit rate defines the rate at which information flows across a data link measured in bits/second.

→ For a binary two level signals a data

Date __/__/__
Page _____
SPLASH

rate of one bit per second is equivalent to one baud.

→ An analog signal carries 4 bit in each signal unit. If 1000 signal unit are sent per second, find the baud rate & the bit rate.

1 bit → 1 symbol.

1000 bauds/sec = Baud rate
↓
Symbols

Bit rate = 1000 × 4 = 4000 bps.

→ If bit rate (or data rate) is 'b' baud rate or symbol rate) is 's'.

General formula:-

$$b = s \times n$$

b → Data Rates (bits per second)
s → Symbol rate (symbols/sec).
n → number of bits per second.

if n=1, Baud rate = bit rate
n=4, Bitrate = 4x Baud rate

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

Sol: (4) 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.

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

(★) SPI is called a 4 wired serial bus, contrasting with 3, 2 or 1 wired serial bus.

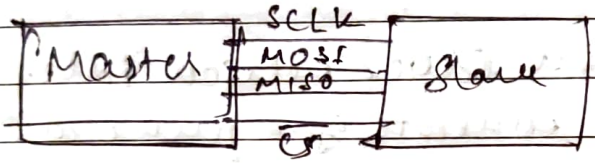
(★) The SPI may be accurately described as a synchronous serial Interface.

(★) But it is different from the synchronous serial Interface (SSI) protocol, which is also a four wire synchronous serial communication protocol.

Date / /
Page
SPLASH

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 from Slave)
$\overline{CS}$ or $\overline{SS}$ → chip select or slave select

So when $\overline{CS}$ = 0 the master has selected the slave to communicate



Working of SPI:

① Master will generate clock whenever it wants to write data to a slave device. After 8 clock pulses data in the master device and data in the Slave device (B7 - B0) is transferred from the master device.

Features of SPI.

① Bidirectional serial data transfer
② High speed / data rates
③ Simple communication for selecting the Slave.

(23) In SPI with a neat timing diagram explain CPHA & CPOL usage

Sol: (*) CPOL determines the polarity of the clock. The polarities can be converted with a simple inverter.

(*) CPOL = 0, is a clock which idles at 0, and each cycle consists of a pulse of 1. That is the leading edge is the rising edge and the trailing edge is a falling edge

(*) CPOL = 1 is a clock which idles at 1, each cycle consists of a pulse at 0. That is the leading edge is a falling edge and trailing edge is the rising edge.

(*) CPHA determines the timing (ie) of the data bits relative to the clock pulses. Conversion b/w them two forms is non trival.

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

Page
SPLASH

iming diagram-
sage

polarity of the
can be converted
ter.

hich idles at 0,
rts of a pulse of 1.
e is the rising
edge is a falling

hich idles at 1 &
pulse at 0.
dge is a falling
e is the rising

timing (ie) of the
ve clock pulses.
two forms.

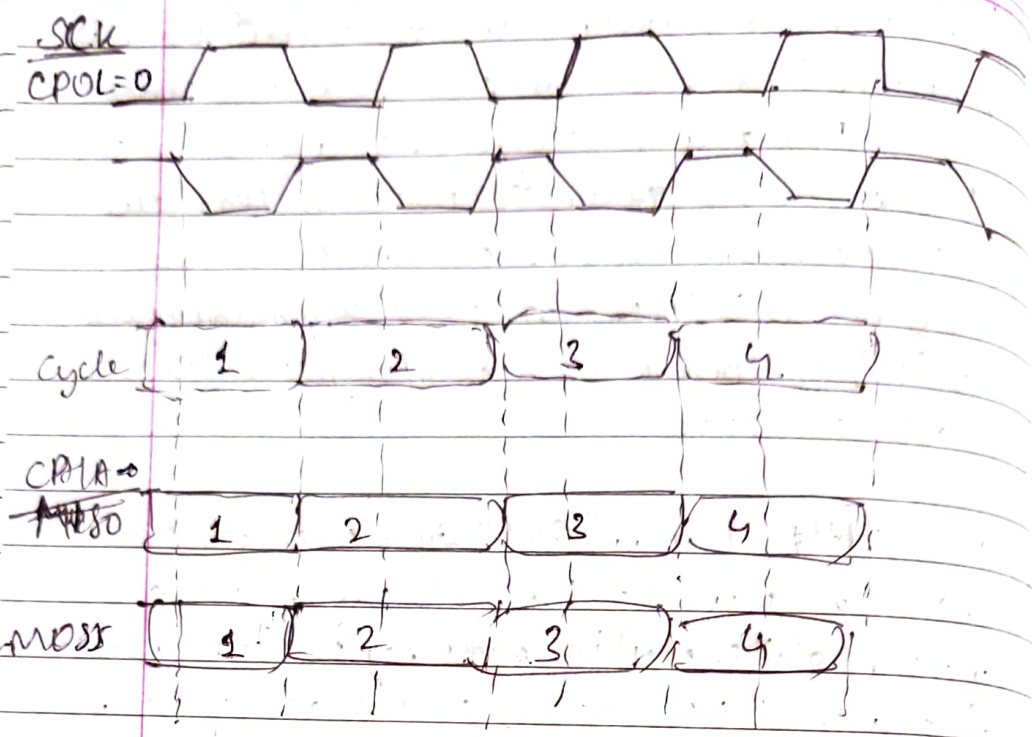
side changes the
dge of the
, while the 'in'
a on the leading
e.

Date
Page
SPLASH

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. An alternative way of considering it is to say that clock idle, followed by a half cycle with the clock asserted.

FOR CPHA = 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 outside holds the data valid until the leading edge of the following clock cycle. For the last cycle, the same holds the MISO line valid until the slave holds. the MISO line valid until the slave select is disserted. An alternative way of considering it is to say that CPHA = 1 clock consists of the half cycle with clock asserted, followed by a half cycle of clock idle.

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



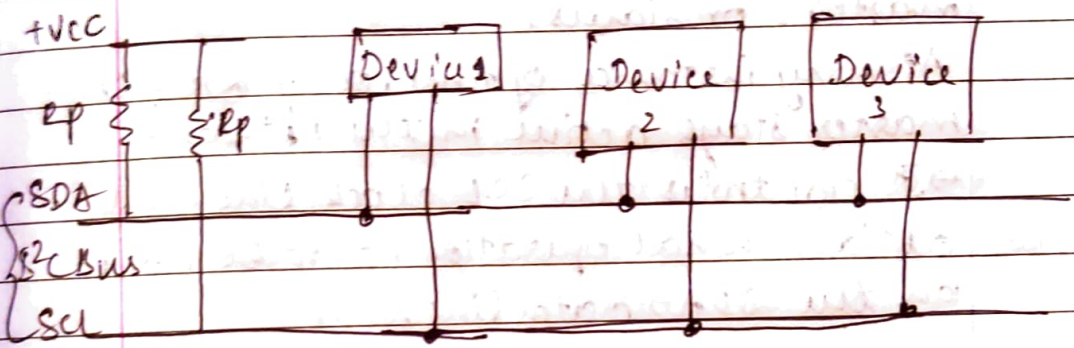
(24) With the neat diagram explain the features of I2C & its working.

Sol:

- Half duplex - Serial communication
- Synchronous communication protocol
- Only two common bus line (wire) are required to control any device/IC on the I2C network.
- Data transfer speed can be adjusted as per requirement.
- Simple mechanism for validation of data transfered
- Uses 7 bit addressing system devices/IC on the I2C bus.
- I2C networks are easy to scale. New devices can simply be connected to the two common I2C bus line.

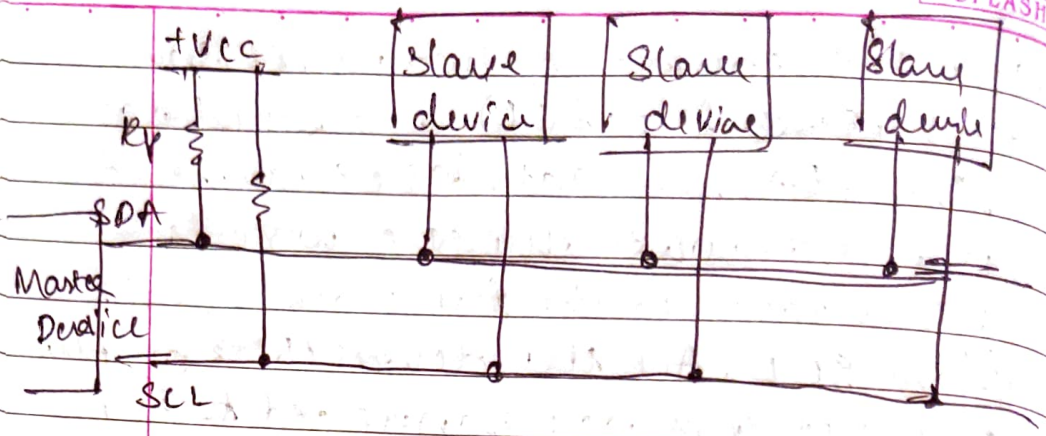
Date
Page
SPLASH

(1) I2C bus consists of two wires. Serial clock line (SCL) and Serial data line (SDA). The data to be transferred is sent through the SDA wire is synchronised with the clock signals from SCL. All the devices I2C's on the I2C network are connected to the same SCL & SDA lines



(2) Both the I2C bus lines (SDA, SCL) are operated at an open drain drivers. It means that any devices on the I2C network can drive SDA & 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

(3) 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.

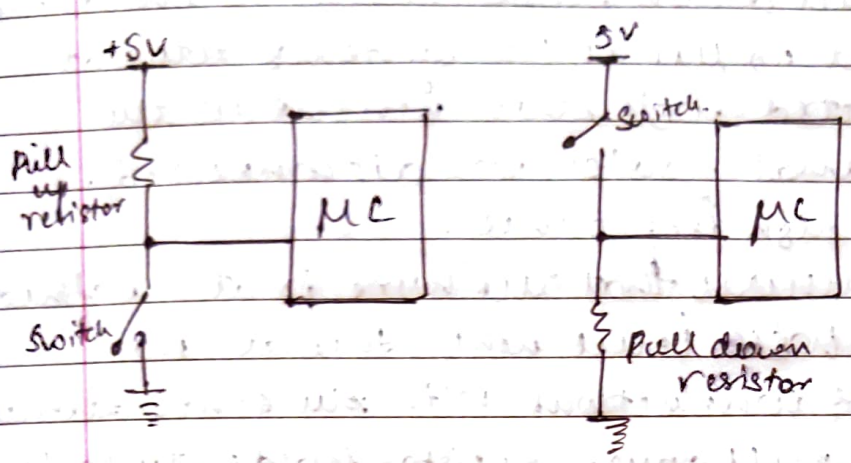


(*) The devices that are connected to the I2C bus are either categorized as masters or slaves.
At any instance of time, only a single master stays active in the network.
→ It controls the SCL clock line and decides what operation has to be done on the SDA → data line.

(*) All other devices that responds to the master are slaves.
When master wants to transfer data or from a slave device, it specifies this particular slave device address on the SDA line and then proceeds with the transfer so effectively the communication takes place b/w the master device & a particular slave.

Date __/__/__
Page _____
SPLASH

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



Pull-up resistor.

* A pull up resistor is used to establish an additional loop over the critical component 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 level in the absence of an i/p signal. Pull-up resistor with a fixed value was used to connect the voltage supply and a particular pin in the digital circuit

Pull-down Resister

* A pull down resister is used to ensure the input to the logical systems settle at expected logic level whenever the external device are disconnected, or of high impedence.
* It ensures that the wire is at a defined logic level when there are no active connections with the other devices.
→ The pull down resistor holds the logic signal near to zero volts when no other active devices are connected.

(26) With a neat diagram explain the concept of arbitration in I2C.

Sol:- * I2C is a multimaster communication means more than one device can initiate the transfer, but not at once.

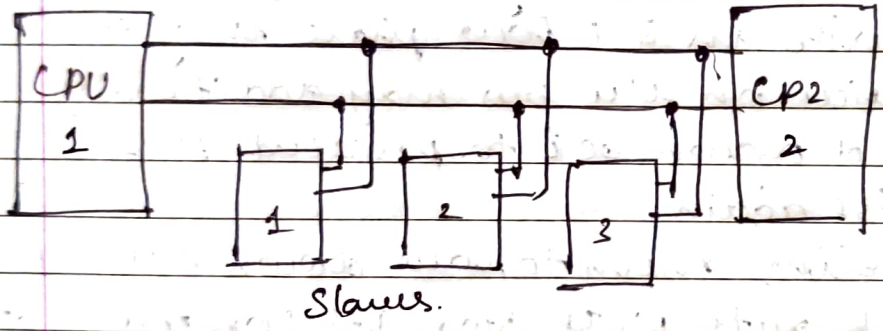
* Bus arbitration happens when two or more masters start to transfer at the same time.

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

Date __/__/__
Page _____
SPLASH

(1) When MCU1 issues the start condition and sends the address the 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.

(2) As long as the two 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.



(27) What is clock streaching & explain clock streaching in I²C.

Sol:-

(*) Clock streaching allows an I²C slave device to force the master device into a waiting state.

(*) A slave device may perform clock streaching when it needs more time to manage data such as store, retrieve data or prepare the transmit another byte of data.

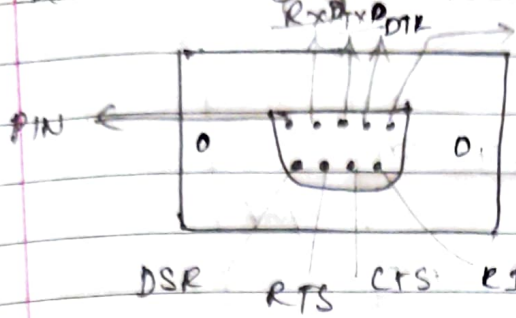
(*) Clock streaching in I²C can slow down communication by streaching SCL. During an SCL low phase any I²C device on the bus may additionally hold down SCL to prevent it to raise again. Enabling them to low down the SCL clock rate or to stop I²C communication for a while. This is also refferd to as clock synchronization

(*) In an I²C communication the master device determines the clock speed which releases master and slave from synchronization exactly to a predefined baud rate

Date __/__/__
Page _____
SPLASH

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

Soln



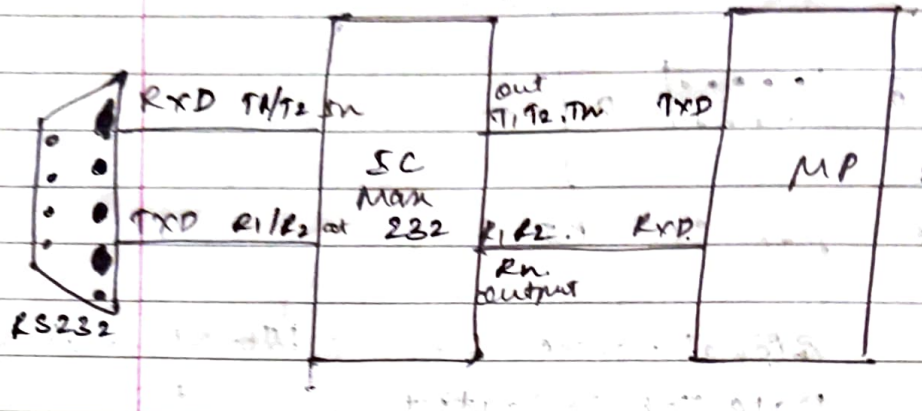
Pin	Signal	Signal name	DTE Signal direction
1	DCD	Data carrier detect	In
2	RXD	Reciever data.	In
3	TXD	Transmitt data.	Out
4	DTR	Data terminal ready	Out
5	GND	Ground	—
6	DSR	Data set ready	In
7	RTS	Request to send	Out
8	CTS	Clear to send	In
9	RI	Ring indicator	In

Handshaking. Modem:

(*) When recieving modem answer the phone call & the two modems begin to communicate.

(*) Before anything else happens, the modem must evaluate the quality of the line, negotiate error control protocols and data compression, that they can both recognise & work but what the most suitable communication speed should be based on the conditions. This process is called a handshake.

(29) Explain the RS232 communication with a microcontroller.



* Several devices collect data from surrounding and need to send it to another unit like a computer for further processing. Data transfer/communication is generally done in 2 ways i.e fast and cis more number of lines.

* Serial communication on the other hand use only one or two data lines transfer data & 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 transferring of signal port is the Baud rate which is the speed at which data is transmitted serially and recieve signal data at different baud rates using software instructions

Date / /
Page
SPLASH

(20) Explain frame format in UART communication

Baud rate:- No. of symbol changes per second.
(*) A group of fixed number of bits.
↳ Signal (Symbol is a group of) ?

Data Framing
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.

Start bit	D0	D1	D2	D3	D4	D5	D6	D7	PB	Stop bit

(*) The UART recieves data from the data bus and this data are being sent by CPU, memory or MC.

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

(*) UART adds the start bit, parity bit and a stop bit to the data recieved from the data bus. which creates a packet.

(*) This packet is now transmitted serial. to the recieving UART reciever.

(*) At reciever, the UART reciever reads the data bit by bit. This serial data is again converted into parallel form at the reciever.

and then sent to the data bus.

④ At receiver the receiver UART controller removes all the start bit, stop bits and if included parity bit to check if the signal has been modified.

Start bit:

⑤ 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 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, electromagnetic at long distance data transfer 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 minimum of two bit duration.

(31) Explain the difference b/w.

(a) Serial v/s parallel

(b) Analog vs Digital

(c) Synchronous vs Asynchronous

Serial	Parallel
(*) Data is sent bit after bit in a single line.	(*) Data is transmitted simultaneously through group of lines.
(*) Data congestion takes place	(*) No data congestion
(*) Transmission speed is low.	(*) Transmission speed is high.
(*) Software implementation	(*) Hardware implementation
(*) No crosstalk issues	(*) Cross talk creates interference b/w parallel lines
(*) Bandwidth is greater	(*) Lower bandwidth.

Analog	Digital
★ Transmitted modulated signal is analog in nature	★ Transmitted signal is digital i.e. train of digital pulse.
★ Message is represented in form of phase, frequency or amplitude variations	★ Amplitude, width & position is transmitted pulse is in the form of code words.
★ Noise immunity is poor for FM & PM.	★ Excellent noise immunity
★ Coding techniques are not possible	★ Coding techniques can be used to detect & correct the error
★ FDM is used for multiplex	★ TDM is used for multiplexing
Ex:- AM, FM, PM, PAM, PWM	Ex:- PCM, DM, ADM, DPCM.

Date __/__/__
Page _____
SPLASH

Synchronous	Asynchronous
⭐ Communication is in real time	⭐ Not in real time.
⭐ Creates interrupts while communicating.	⭐ No interrupts will be present.
⭐ Faster communication	⭐ Slower communication.
⭐ No overheads as it start & stop bits	⭐ Uses start & stop bits.
⭐ Uses constant interval of time	⭐ Uses irregular amount of time. / non constant amount of time.
Ex:- Video conferencing, chat rooms	Ex:- Sending emails