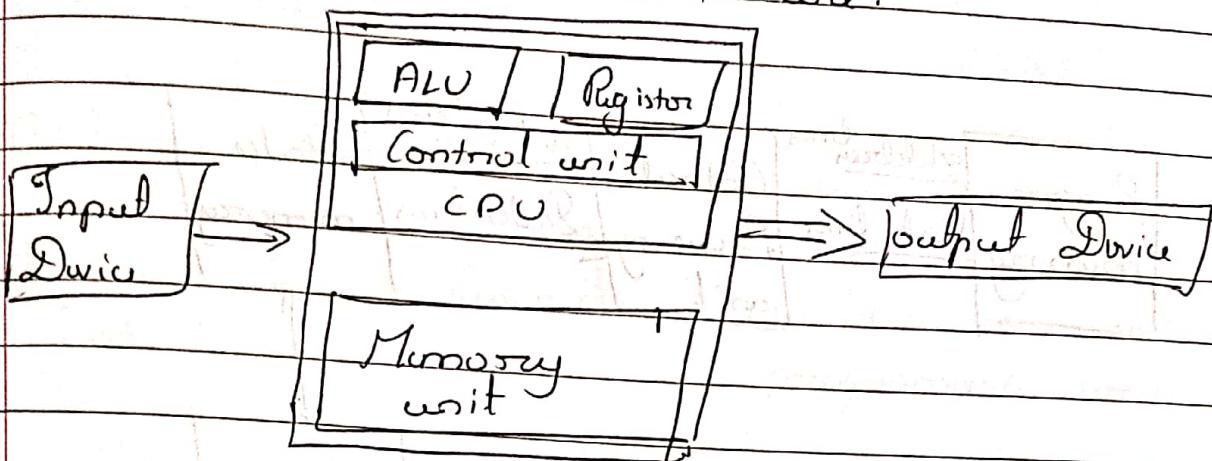


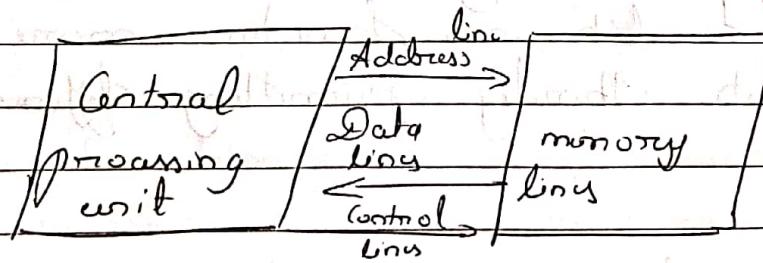
Q1 with a neat diagram, Explain Von Neumann and Harvard architecture.

Ans Q2

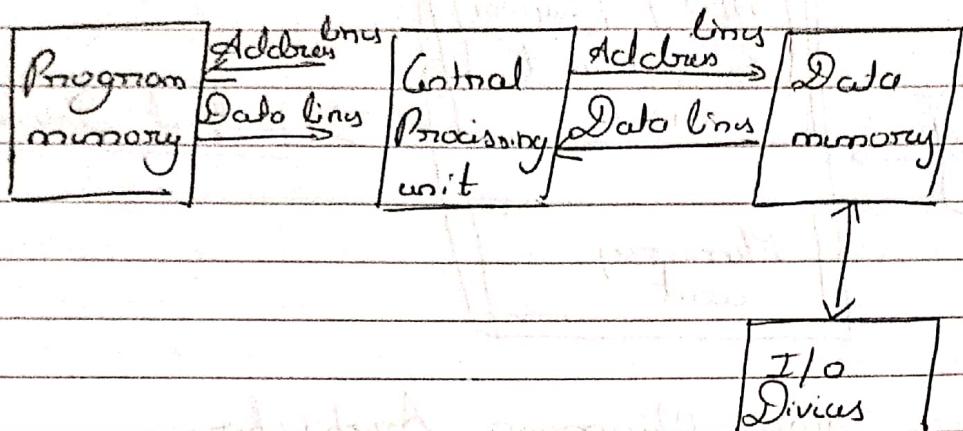


Von - Neumann Architecture

- A Computer architecture that uses a single memory unit within to which both data and instruction get stored is known as Von Neumann architecture.
- The core three major components that constitute this architecture are:
 - CPU
 - Memory unit
 - I/O Interface
- The memory unit also serves as crucial part of the overall system as it stores both data as well as instructions.



After the Execution of the program, data gets

Harvard architecture

- A computer architecture where the memory unit is divided into two parts for individually storing data and instruction is known as Harvard architecture.

- This means, unlike Von Neumann architecture, here data memory and instruction memory is in Separate format. The figure below shows Harvard model.

- In this approach, Efficient resource utilization occurs as instruction bits can sometimes move faster than data bits thereby permitting different cell sizes.

- The Central processing unit present must be more efficient so that it can handle two set of buses & allows simultaneous data transfer.

- ② list the difference RISC vs CISC machine

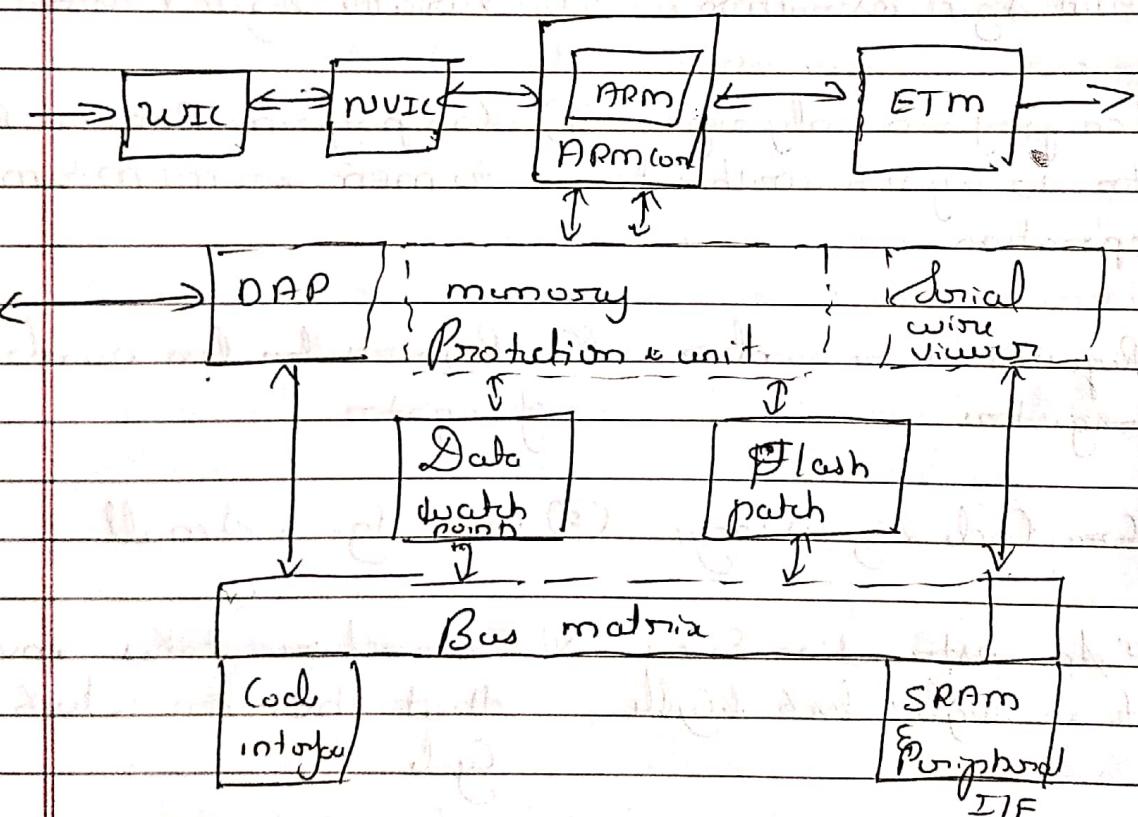
RISC

CISC

- | | |
|--|--|
| ① Focus on software | ① Focus on hardware |
| ② uses only hardwired control unit | ② uses both hardwired and microprogrammed control unit |
| ③ Transistor are used for more register | ③ Transistor are used for storing complex instruction |
| ④ Fixed sized instruction | ④ Variable sized instruction |
| ⑤ Can perform only reg-to-reg arithmetic operation | ⑤ Can perform REG-to-MEM or MEM-to-MEM operation |
| ⑥ Requires more number of registers | ⑥ Requires less number of registers |
| ⑦ Code size is large | ⑦ Code size is small |
| ⑧ An instruction Executed in single clock cycle | ⑧ Instruction takes more than one clock cycle |
| ⑨ An instruction fit in one word | ⑩ Instruction are larger than the size of the word |

(3) with a neat diagram, explain Eth ARM7 architecture.

- The ARM architecture processor is an advanced reduced instruction set computer [RISC] machine & is a 32 bit reduced instruction set computer (RISC) microcontroller.
- The ARM architecture comes with totally different versions like ARM1, ARM11, ARMv2 etc each one has its own advantages & disadvantages.



The DRAM architecture

- (1) arithmetic logic unit
- (2) Booth multiplier
- (3) Barrel Shifter
- (4) Control unit
- (5) Register file

(1) Arithmetic logic unit (ALU)

The ALU has two 32 bit input. The primary comes from the register flags modified by the ALU outputs.

(2) Booths multiplier

The multiplier factor has 3 32-bit inputs and the inputs comes from the register file. The multiplier output is nearly 32-bit significant.

(3) Booths Booth algorithm

Booths algorithm is a nonoverlapped multiplication algorithm rule for 2's complement numbers.

(4) Barrel Shifter

The barrel shifter features a 32 bit input to be shifted. This input is coming back from the register file or it might be immediate data.

③ Control unit-

For any microprocessor, control unit is the heart of the whole processor and it is responsible for system operation. So the control unit design is most important part within the whole design.

4) With a neat diagram, Explain the programming mode of ARM

- a) • ARM is a flexible programming designed architecture with different applications.
- Design is simple, optimum & economic.
- A processor instruction set defines the operation that the programmer can use to change the state of the system incorporating the processor.
- This state usually comprises the values of the data items in processor visible registers and the system's memory.
- Each instruction can be viewed as performing a defined transformation from the state before the instruction is executed to state after it has completed.

ARM Registers

- ARM has 31 general purpose 32 bit registers.
- 16 registers only visible registers.
- 16 registers are user mode, other registers are used to speed up execution processing.

- 16 registers are R0 to R15. To this three register (R13, R14, R15) are special purpose register.
- These three registers used for only important functions in program.

ARM processor model

R0

R1

R2

R3

R4

R5

R6

R7

R8

R8-jig

R9

R9-jig

R10

R10-jig

R11

R11-jig

R12

R12-jig

R13

R13-jig

R14

R14-jig

R15

CRSR SPSR-jig SPSR-SVC -SPSR-abt -SPSR-irq SPRR-und

mod1 Jig SVC abt1 irq und
mod2 mod2 mod2 mod2 mod2 mod2 mod2

(5) With a neat diagram, Explain the programming model of for Thumb.

4 (1) Processing operating Stacks:

- From the programmers point of view, the ARM7TDMI can be in one of two states:

- ARM State which executes 32-bit word aligned ARM instructions

- Thumb State which operates with 16-bit half word aligned THUMB instruction

(2) Switching States:

- Entering Thumb State:

- Entry into Thumb State

can be achieved by executing a BX instruction with my State bit (bit 0) set in the operand register

~~From ARM~~

(3) Memory formats:

- ARM7TDMI views as bytes

Collection of bytes and numbered upwards from zero

- Bytes 0 to 3 hold the first word stored word, bytes 4 to 7 in the second & so on

(4) Big-endian format:

In Big Endian format, the most significant byte of a word is stored at lower numbered byte & and the least significant byte at the highest numbered byte

⑤ Little Endian formats

In Little Endian format, the low-numbered byte in a word is considered the word's least significant byte, the highest numbered byte the most significant.

⑥ Instruction length:

- Instructions are either 32 bits long (in ARM state)
- or 16 bits long (in THUMB state)

With a neat diagram, Explain three stage pipeline of ARM

→ Pipeline is the maximum used by RISC Processor to Execute instructions

→ By Specifying . in the .Execute by fetching instructions, while other instructions are binary decoded & Executed Simultaneously

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

The ARM has Three Stage pipeline:

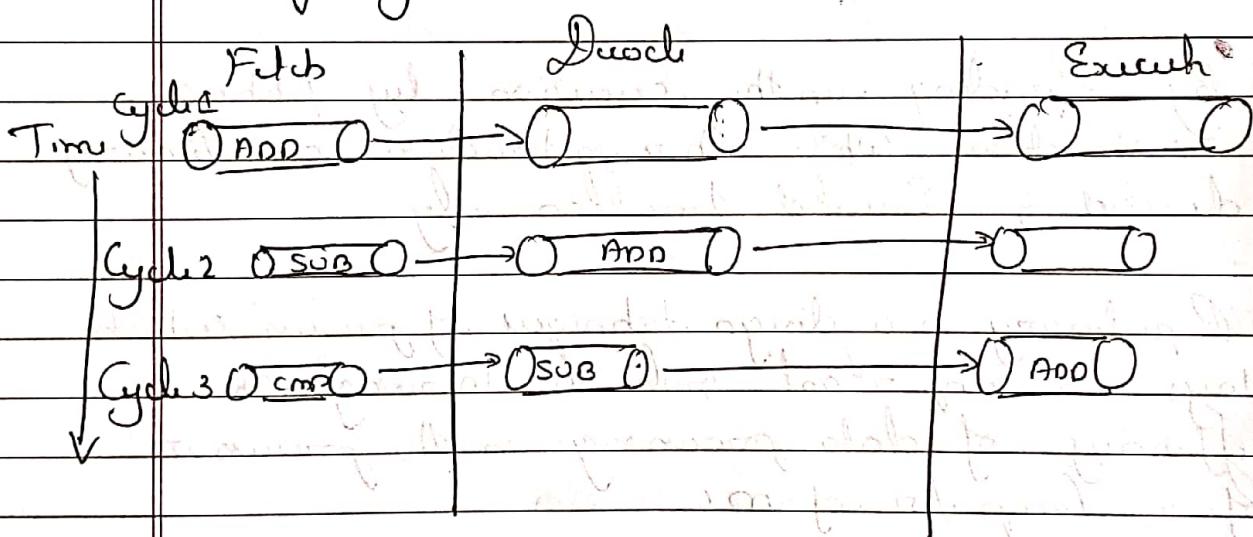
(1) Fetch: The instruction is fetched from memory
 (2) Decode: The Instructions specify operations of p are decoded to determine what function to perform

(3) Execute: The decoded instruction is Executed

Ex. Each of this operation requires one clock cycle for typical instruction. Thus a normal instruction requires three clock cycles to complete. This is known as the latency of

Instruction Execution

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



(7) With the given diagram, Explain CPSR register.

~~the result of the ALU operation is stored in CPSR register.~~

Condition code flags

- N: Negative result from ALU
- Z: Zero result of operations, overflowed
- V: ALU operation - overflowed
- C: ALU operation - carried out

Sticky condition flag - Q flag

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

Interrupt Disable bit

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

T bit

- Architecture STE only
- T → 0 processor in ARM state
- T → 1 processor in Thumb state

mode bits

- Specify the processor mode

CPSR → for current processor. It is a register which holds the information about the current state of the processor.

SPSR → Saved processor status register. holds the information on the processor status before the information on system changed to this mode i.e. processor status just before an or exception.

(8)

Explain the seven different modes in ARM

The ARM7TDMI processor has seven modes of operation.

→ user mode is the usual ARM program execution state and useful for executing most application programs.

→ Fast Interrupt (FIQ) mode supports a data transfer on channel processor.

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

→ Supervisor mode is a privileged mode for the operating system.

→ Abort mode is a privileged user mode for the operating system.

→ System mode is a privileged user mode for the operating system.

We can only enter System mode from another privileged mode by modifying the mode bits of the current program register (CPSR).

if undivided mode is also entered. When an undivided instruction is Executed

modes other than user mode are collectively known as privileged modes. Privileged mode are used to Service interrupts or to access protection, resources.

Mode

undivided mode Today just

Fast interrupt privilege mode

Interrupt privilege mode

Supervisor

about

System

undivided

(a) Explain the nomenclature in ARM

ARM was originally from acorn computer to UCL. It is RISC processor for commercial use.

ARM 7TDMI Processor

32 Bit Processor Advanced machine

T → Thumb . architecture . Extension

n → Debug . Extension

m → Enhanced . Extension

I → Traceunit . Emulation

ARM fxyfzyfzy · TDMI {EYfJyfzyfzy}

x → Sins

y → memory management unit

z → Cache

T → Thumb - 16 bit decoder

n → JTAG . Debugging

m → Fast multiplier

I = Embedded Instruction of for DSP

E = Enhanced Instruction of for DSP

F = Floating point

s → Synthesizable version

(10) What is JTAG? Explain JTAG State Diagram

JTAG has become a standard in Embedded System and it is available in nearly every microcontroller and FPGA on the market.

If we have programmed a microcontroller thoroughly a strong chance that we have used JTAG or one of the related standards.

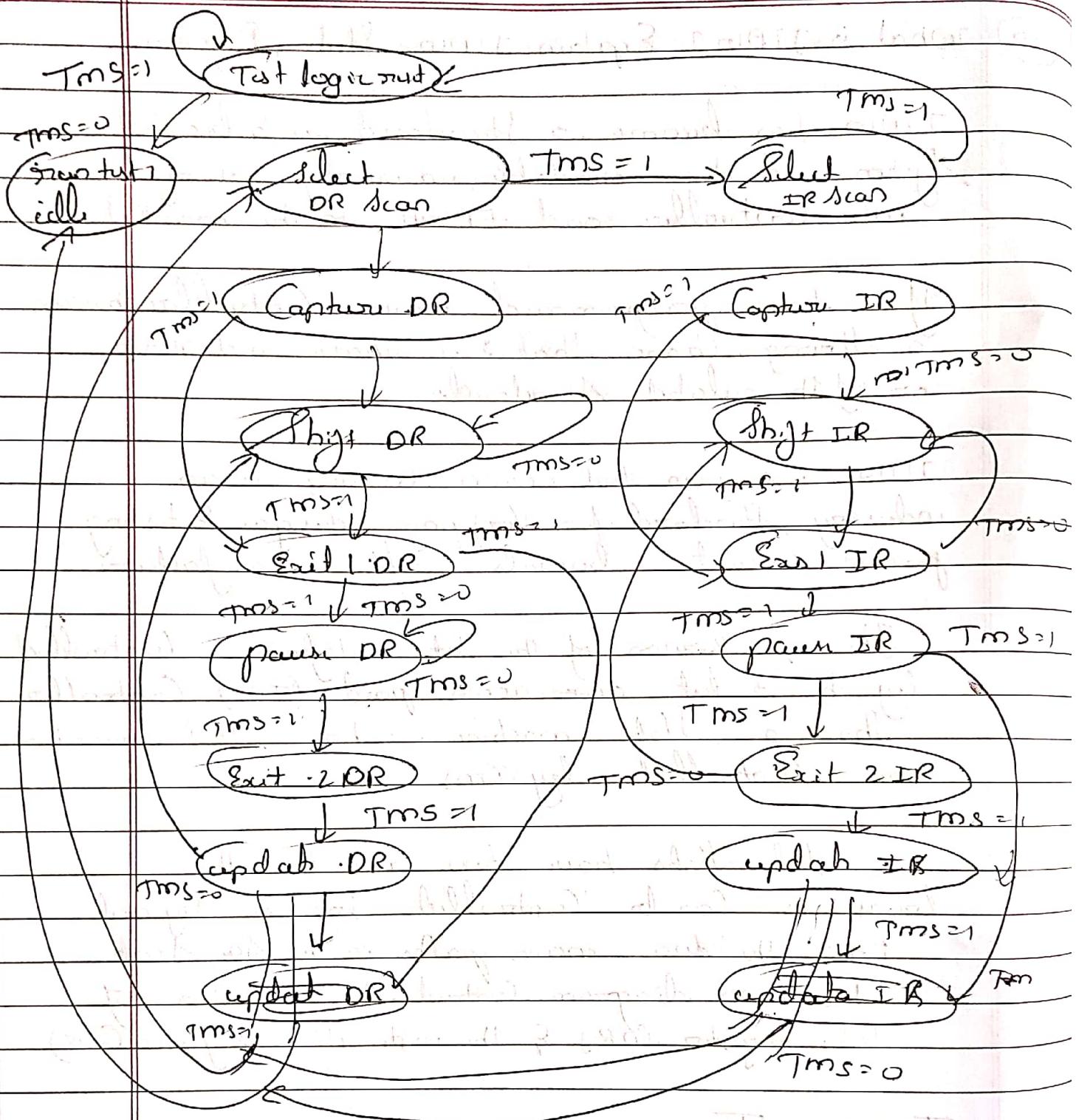
JTAG i.e. joint test action group is an industry standard for verifying designs & testing printed circuit boards after manufacture.

The operation of the test interface is controlled by the test access port (TAP) Controller. This is a state machine whose state transitions are controlled by TMS.

All states have two exits to the transitions can be controlled by one signal TMS. The two main paths in the state transition diagram control the operation of data register (DR) & the instruction register (IR).

$$T_{DR} = -T$$





The normal operation of a JTAG-based system is to enter an instruction which specifies the sort of test to be carried out next & the

data register to be used for that test & into the instruction register & is then done. The data register to carry out the test.

Instruction may be public or private, public instruction are declared & available for several test types, and the Standard specifies a minimum set of public instruction that must be supported by all devices that comply with the Standard. Private instruction are for specialized on-chip test purposes & the Standard does not specify how they should operate or how they should be used.

- The interface works with 5 dedicated signals which must be provided in each chip that supports the test standard
 - TRST is a test reset input which initializes the test interface.
 - TCK is the test interface independent from any system clock.
 - TMS is the test mode Select which control the operation of the test interface state machine.
- TDI is the test data input line which carries the sampled values from the boundary scan chain & propagates data to the next chip in the serial test circuit.

(ii) ~~Explain what is single tasking. Give examples of microprocessors which work with MMU support.~~

Ans.

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

Microcontroller can be known as computer on chip. They are designed to perform a single task only because its processing power as well as memory is not suitable for installing an OS.

(12) What is MMU? Why MMU is required? Give Example of MMU Supervisor

The memory can be defined as a collection of data in a specific format. It is used to store instructions & processing data. The memory consists of a large array or group of words or bytes, each with its own location.

The primary motto of a computer system is to execute programs. The programs along with the info. information they access should be in the main memory during execution. The control fetches the instruction from memory according to the value of the PC.

The main memory is central to the operation of a computer. Main memory is a large array of words to bytes ranging in size from hundreds to thousands to billions. Main memory is a repository of rapidly available info. shared by the CPU & all devices. Main memory is the place where programs info. are kept when the processor is effectively utilizing. The main memory is associated w/ the processor is extremely fast. Main memory RAM contains data when a power interruption occurs.

Memory management

In a multi-programming computer, the tasks in a part of memory & the task is used by multiple processor. The task

The task of subdividing the memory among different processes is called memory management. memory management is a method used to manage operation b/w main memory & disk during process execution. The main aim of memory management is to achieve efficient utilization of memory.

memory management is required to,

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

Ex. IBM System/360 model 67, IBM System 1370.

ARM architecture based application memory implement as mmu defined by ARM's virtual memory system architecture. The current architecture defines PTE's for dividing 4KB & 0.4KB Pages or 1MB Sections & 1GB Super Sections. legacy VMS also defined 1KB tiny space.

- (13) What is Endians? list the types Give examples
 Endians is a term that describes how data is stored. It defines which end of a multi byte data type contains the most significant value.
 The two types of endians

(1) little Endian:

In little endian machine first byte of binary representation of the 4 multi byte data type is stored first.

Ex :-

Suppose integer is stored as 4 bytes then a variable x with value 0x01234567 will be stored as following

0x100	0x101	0x102	0x103
01	23	45	67
67	45	23	010

Fig:- little Endian

- (4) write a C program to find the Endians of a given number

Program 1:

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
  unsigned int Value = 0x11223344;
```

```
  char *x = ((char *) &Value);
```

```

int main()
{
    int arr[4];
    for (int i = 0; i < 4; i++)
        arr[i] = i;
    printf("Address of arr[i] = %d) \n", &arr[i] & arr[i]);
    return 0;
}

```

Programs

```
#include < stdio.h>
```

```
int main (void)
```

```
    unsigned int value = 0x12345678;
```

```
    if (value == 1)
```

```
        printf ("your system is little Endian \n");
    else
```

```
        printf ("your system is Big Endian \n");
    return 0;
}
```

(15)

Explain.

8

Binary values are often grouped into a common lengths of 3/4/ 8's etc. This number of digits is called the bit length of numbers. Common bit-length of binary numbers include bits ; nibble ; nibbles ; bytes . Each 1 or 0 in a binary number is called a bit.

A group of 4 bits \rightarrow called a nibble.

A group of 8 bits makes a byte.

length	Name	Example
1	Bit	0
4	Nibble	1011
8	Byte	10110101
16	Half word	10110101 10011001

word is another length that is thrown out from time to time word is much less sounding & more ambiguous. The length of a word is usually dependent on the architecture of a processor. It could be 8 bit, 32, 64 or even more.

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

$$\text{half word} = 16 \text{ bits}$$

$$\text{word} = 32 \text{ bits}$$

(16) Explain the word align & half word align.

ARM - memory

Different processors have different definition of words. For 32 bit processor, a word is 32 bits as the name implies, a half word is 16 bits for 16 bit processor, a word is 16 bits (2 bytes) for 8 bit processor word is 8 bits.

word alignment :- The stored addresses are adjacent & can be divided by 4, the last two digits are 00

half word aligned : That is the start addresses are adjacent & divisible by 2
, that is the last bit is 0.

ARM architecture requires 32-bit ARM instruction that must be word aligned

& stored in memory & 16-bit Thumb instructions required half word aligned and is stored. Therefore in ARM state the 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 or 1 one word consists of one or more bytes i.e. it is usually integer bit of bytes

(17)

Explain the software tools involved in processing the C source file with a neat diagram

Software development for the ARM is

Supported by a coherent range of tools - development by ARM & there are many third party & public domain tools available such as an ARM back-end for the gcc "C" Compiler

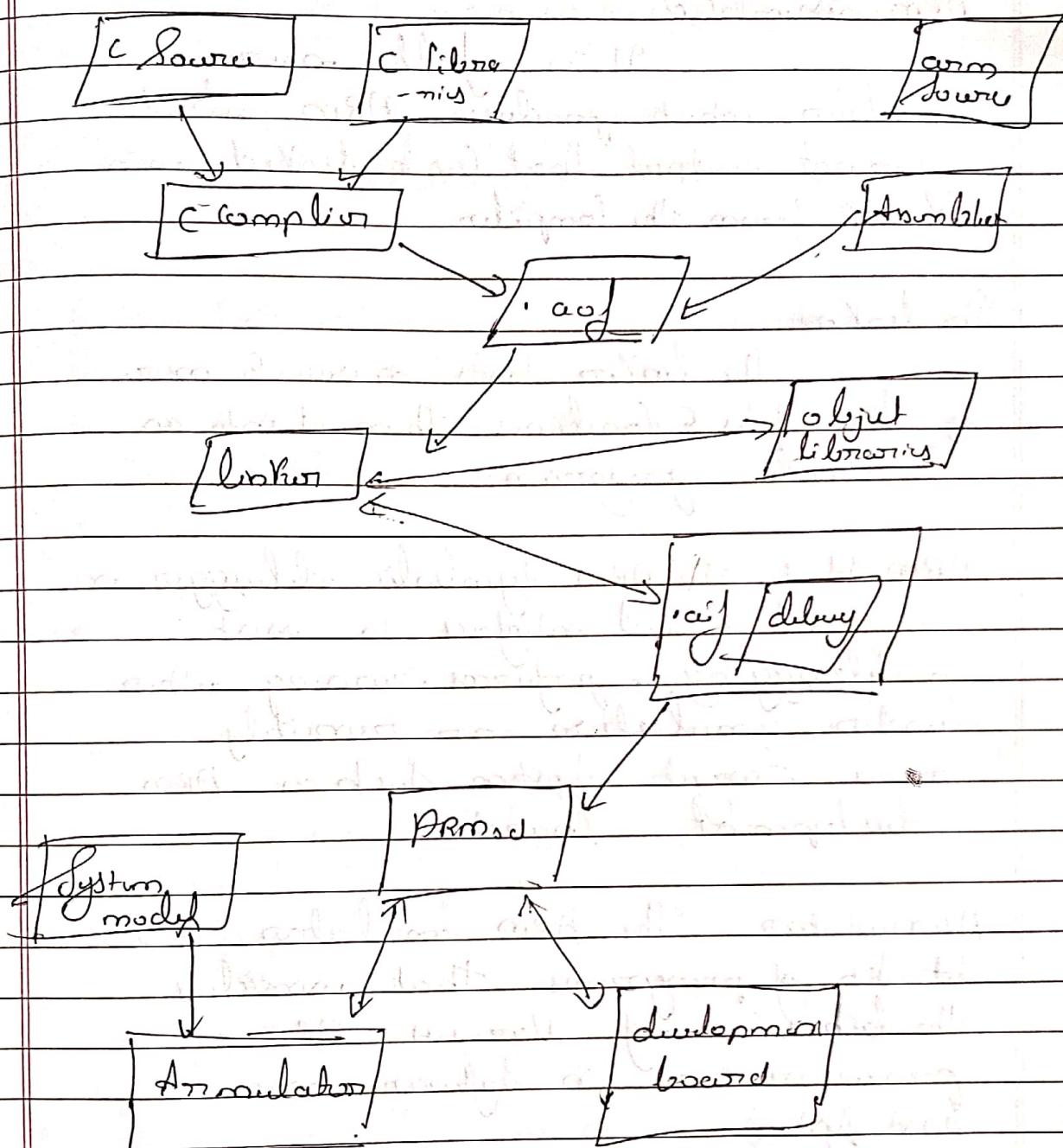


Fig :- Structure of ARM GRIM - Development
TOOLKIT

ARM "C" Compiler:

It uses the ARM procedure
Call Standard for all externally available
functions

ARM assembler

It is a full source assembler which produces ARM object format output that can be linked with O/P from the compiler.

The linker

The linker takes one or more of object files & combines them into an executable program.

ARM SD - The ARM Symbolic debugger is a front end interface to assist in debugging, performing either under simulation or remotely on a target system such as ARM development board.

ARMulator - The ARM emulator is a set of programs that mimics the behaviour of various ARM processor cores in software on a host system.

Q8 Explain the following addressing modes in ARM

- Three address
- Two address
- Single address instructions with respect to ARM

Sequence of instructions form a program to perform a specific task

Components →
 1. Operation field → specify how data manipulated
 2. Address field → specify the data location

When data is to be read from or stored into two or more address fields may have one or more than one address

- Three address instruction
- Two →
- One →
- Four →

Processor can execute an instruction only if it is represented in binary sequence unique. Binary sequence pattern must be assignment. This process is called on-code decoding.

one-address Instruction

This uses an implied accumulator register for data manipulation & the other is in the register or memory location. Implied means that the CPU also already knows that one operand is in the accumulator. For known that one operand is in the accumulator, there is no need to specify it.

Eg: LDR addr

Acc ← (addr)

Two-address Instructions

How two addresses can be specified in the instruction. In the one address instruction, in the one address instruction, the result can be stored in different locations i.e., registers or memory location. But requires more number of bits to represent the address.

Ex : $\text{MOV } R_1, R_2$
 $R_1 \leftarrow [R_2]$

Three-address Instructions

This has three address field to specify a register or memory loc - fice. Programs created are much smaller in size, but number of bits per instruction increases. Thus instruction length, execution of programs, a much easier, but it does not mean that program will run much faster because, here instruction only contains more steps but each micro operation will be performed in one cycle only.

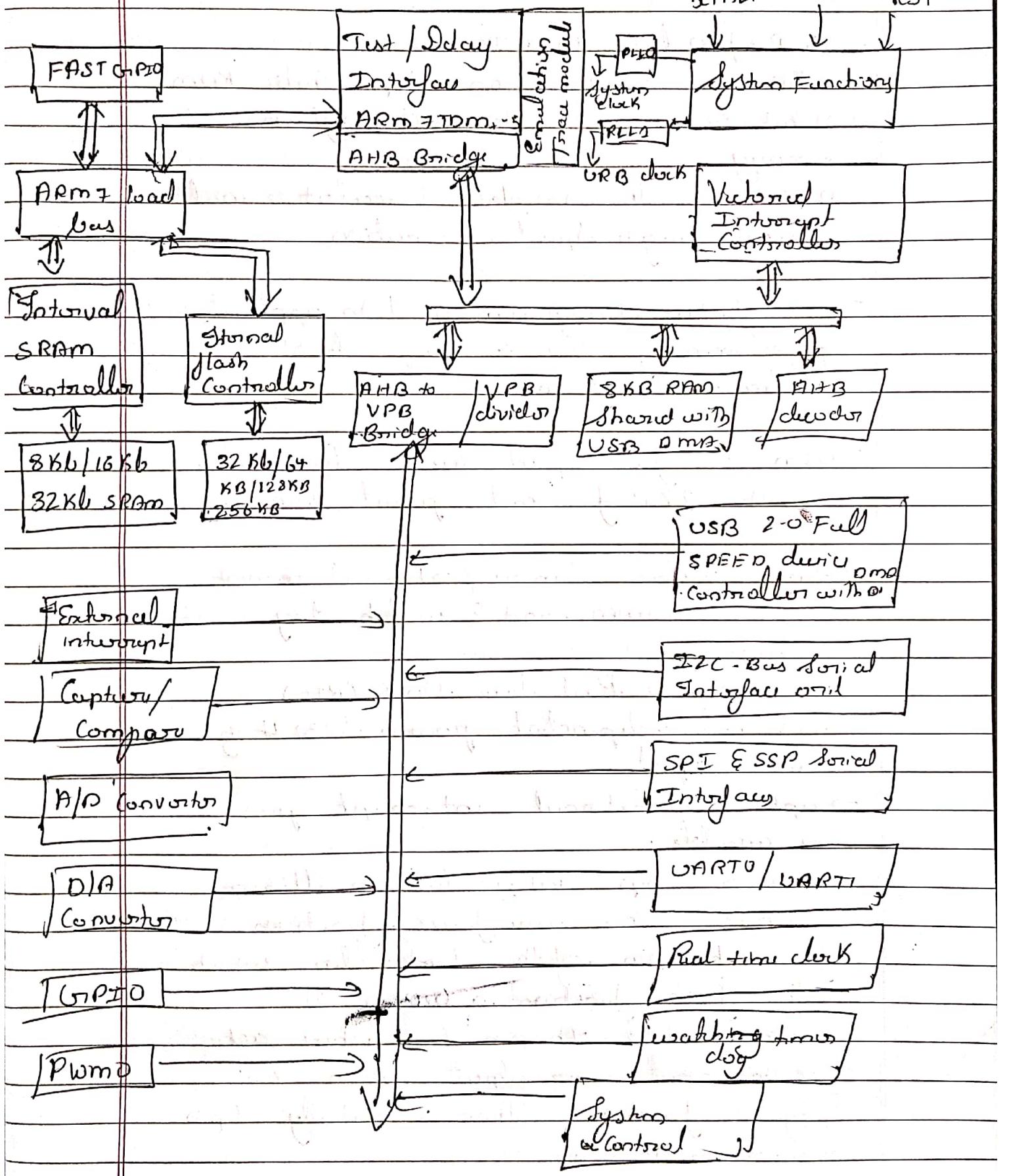
Ex. $\text{ADD } R_3, R_1, R_2$

$$R_3 = R_1 + R_2$$

(19)

Explain the LPU2148 using Block diagram

ITALY ITAL2 PST



Features:

- 16 Bit / 32 bit programming is supported.
- 8 KB to 4096 KB of on-chip static RAM
- 32 KB to 512 KB of on-chip flash memory.
- 128 bit wide Software I/O accelerator enables 60MHz high speed operation.
- In system programming / In application programming via on chip boot loader / software.
- USB 2.0 full speed device controller with 2 KB of end point RAM.
- Single 16 bit timer / external or internal counter, PWM unit & watch dog timer.
- Low power Real time clock (RTC) with its independent power of 32.768 Hz clock input.
- up to 21 External interrupt pins available.
- The on chip integrated oscillator operates thru four junctions for each pin of the controller which the first junction is open. It means that the pins can either act as an input or output with no specific function. There is initially 3 pin register in LPI 2148.

Controllers in Andes to control the functions of pins in the suprachiasmatic nucleus - The classification given below:

PINS ELO : Controls functions of Port 0-0 - Port 1-15

PRIMUSIEL = Controls functions 0.1b - Post A-3

Prasetyo - Controls Junctions ... , 1.16 to Part

When the use of a 31 bit register can give configuration

00	GPOU Part 0-1
01	TXD (UART4)
10	RUMI
11	Res Reserved

LPC 2148 has 2-32 bit GPIO ports. A total of 8 bits
can be mapped to a single output. Only pin out of
32 pins are available on PORT0, PORT1
has upto 16 available for GPIO functions. PORT0
& PORT1 are controlled via 2 groups of 4
registers.

- * TOPIN
 - * JOSET
 - * TOPIR
 - * TOCLR

IOPIN : This register provides the value of port pins that are configured to perform only digital functions. The register will show the pin is configured for input or output.

or as GPIO or alternate digital functions.

IODET - This register is used to produce a HIGH level output at the port pins. Configured as GPIO in as output mode, writing 1 to it produces a high level at the corresponding port pins even if any pin has no job. If any pin is configured as an input or a secondary function writing 1 to the corresponding bit in the IODET has no effect.

IODR

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

IOCLR:

This register is used to produce a low level at the port pins. Configured as GPIO in as output mode writing 1 to it produces a low level at the corresponding bit in IODET register. Writing 0 has no effect. If any pin is configured as an input or a secondary function writing IOCLR has no effect.

21]

with a neat diagram explain Band width & Bit rate. Explain the calculation.

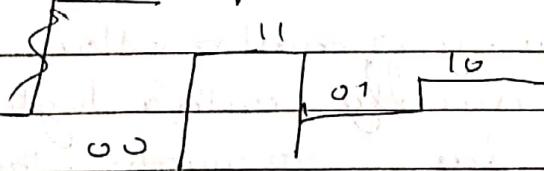
→ Band: How many times a signal changes per second.

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

Bit rate is controlled by band of number of signal. i.e levels

Band width:

- Number of times bits changed per second
- If Band width by 4 (4 change per second)
- If 2 bit per time change in 2
- Bit rate = 8 bits per second
- Bit rate = $\times 2$ Band width in this. Example



- Band width defines the switching of a.
- Bit rate defines the rate at which info can occur. a data link measured in bits/sec

1 Bit \rightarrow 1 Symbol

$$\text{Bit rate} = \text{Band width}$$

$$1000 = \text{Band width}$$

$$1000 = \text{Band width}$$

bits/sec

If 1 bit / symbol (or data rate) \Rightarrow 'B'
 Band - rate (or Symbol rate) \Rightarrow 'S'
 $B = S \times n$

$B \rightarrow$ data rate (bit per second)
 $S \rightarrow$ symbol rate (symbol/sec)
 $n \rightarrow$ no of bits per second

If $n=1$, Band rate = Bit rate
 $n=4$, Bit rate = $4 \times$ Band rate

22] with a neat diagram. Explain working
 Features of SPI protocol.

SPI \Rightarrow Synchronous Serial Communication. Interface Specification used for short distance communication.

- * It was developed by Motorola
- * SPI devices communicate in full duplex using a master-slave architecture usually with a single master
- * The master device originates the frame for reading & writing
- * SPI is called a four wire serial bus
- * Contrasting with three, two or one wire serial bus

The SPI may be accurately described as Synchronous Serial Interface

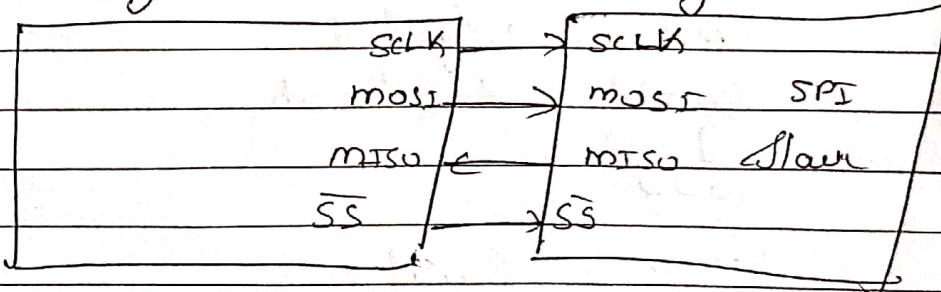


Fig :- Single master to single slave

23] In SPI with a neat timing diagram. Explain CPHA & CPOL usage.

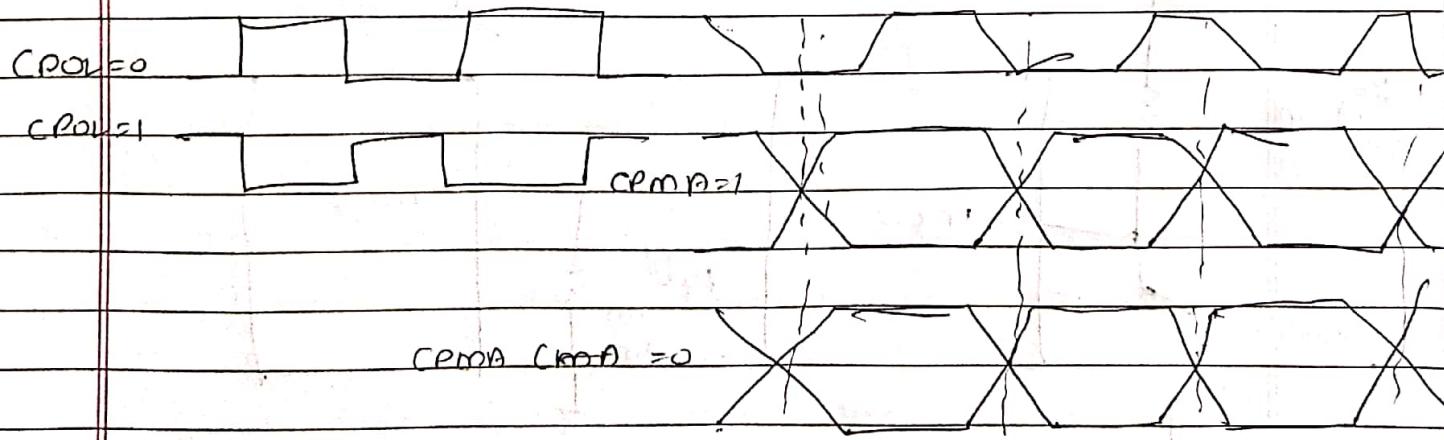
CPOL determines the polarity of the clock. The polarities can be converted with a simple inverter.

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

$CPOL = 1$ is a clock which idles at 1 & each cycle consists of a pulse of 0, i.e. leading edge - falling edge - trailing edge - rising edge.

$CPHA = 0$ 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.

$CPHA = 1$ 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.



$(CPOL=0)$

SCK

 $(CPOL=1)$

Cyl

mJSU

GP.HD0

mosI

Cycle

CPHASE1

mJSU

mosI

Q. with... circuit diagram Explain the
pull up pull down registers

R_s
pull up
resistor

switch

mcu

Pull
down
resistor

mcu

Pull up Resistor:

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

It is used to ensure that a wire is pulled to a high logic level in the absence of an input signal. Pull up resistor with a fixed value was used to connect the voltage supply to a particular pin in the digital circuit.

Pull down resistor:

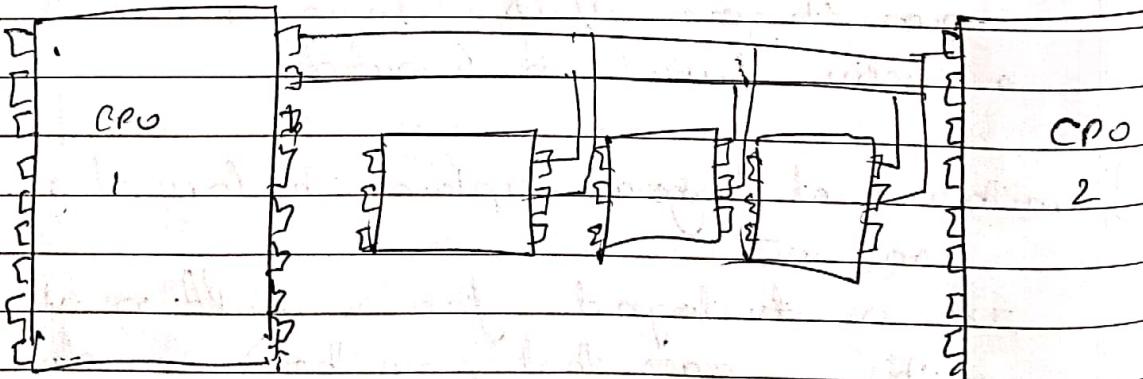
A pull down resistor is used to ensure the input to the logic system settle at expected logic level whenever the external device are disconnected or of high impedance. This means that the signal is at a digital low logic level when there are no active connections with other device. The pull down resistor holds the logic signal near to zero Volt when no other active device is connected.

2.6] With neat diagrams explain the concept of arbitration in I2C.

I₂C is designed for multi master purpose. This means that more than one device can initiate transfer.

Bus arbitration occurs when 2 or more masters 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 transaction can be active at a time in the system.
- When using only one master on the bus there is no real need of capturing or data. Except by if a slave device is malfunctioning. If there is a fault condition involving in the serial bus.
- Who monitors issues a start condition & since the address all slaves will listen. If the address does not match the address of CPU. This device has to hold back any activity until the bus becomes idle again after a stop condition.
- As long as the devices monitors what's 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 then there is no problem.



27]

What is clock stretching? Explain clock stretching in I2C.

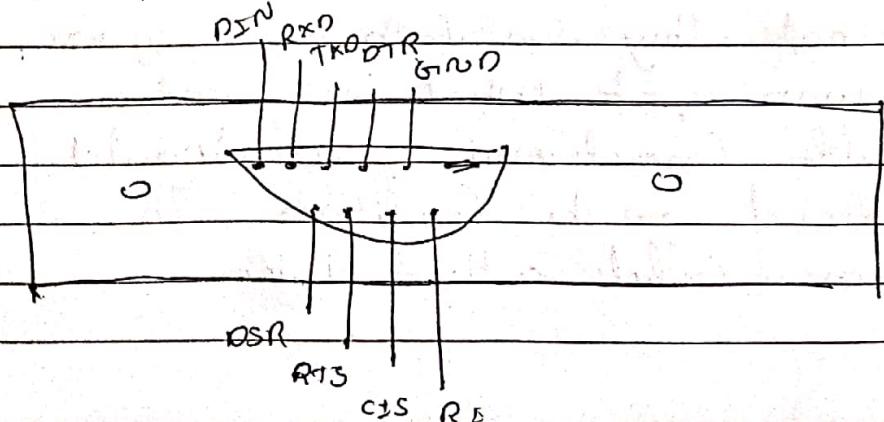
If allows an I2C slave device to force the master device into a wait state. A slave device may program clock stretching when it needs more time, like to manage data such as store received data or prepare that the to transmit another byte of data.

Clock stretching in I2C devices can slow down communication by stretching SCL during an SCL low phase. Any I2C device on the bus may, & additionally hold down SCL to prevent it to rise again, enabling them to low down the SCL clock rate, or to stop I2C communication for a bit while. This is also referred to as clock synchronization.

In an I2C communication the master device determines the clock speed, which rules master & slave from synchronization exactly at a particular baud rate.

28)

Explain working of DSR-PMS-E bands stretching with the waveform



Pin	Name	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

Handshaking medium

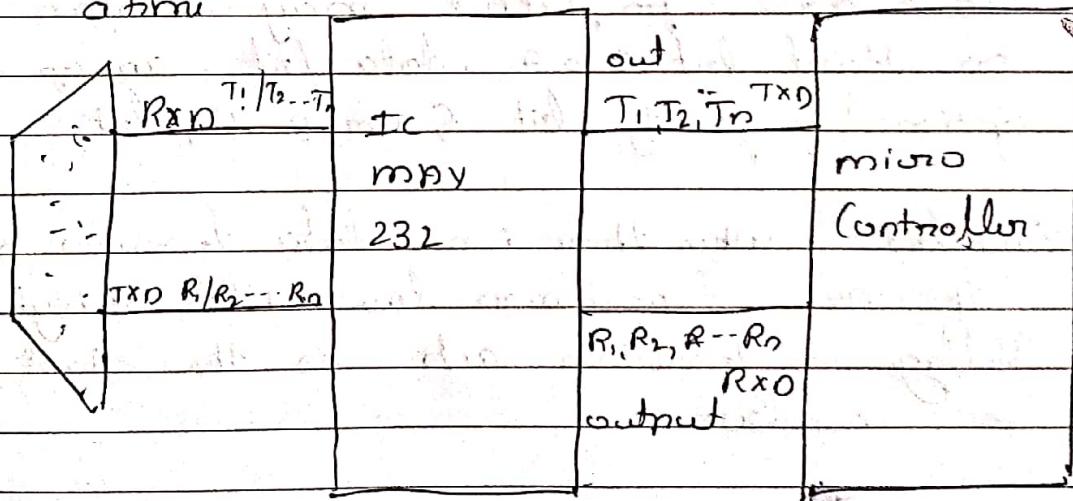
A medium = handshake

What occurs when the receiving medium answers the phone call is the two modems begin to communicate

Before anything else happens the medium must negotiate the quality of the link, negotiate error control protocols, & data compression that they can both recognize & more, put what the most suitable connection speed should be based on the conditions. This process is called a Handshake

29] Explain RS232 connection with a note
 30] Several devices collect data from Sensors & need to send it to another unit like a computer for further processing. Data transfer communication is generally done in two ways is fast & uses more number of lines.

Serial communication on the other hand, uses only one or two data lines to transfer data & is generally used for long distance communication. In serial comm - variation in the data is done on one bit at a time.



A important parameter considered while interfacing - Signal port → The Board rate which is the speed at which data is transmitted serially, i.e., can be set to transfer & receive signal data at different board rates using software instructions.

(BQ)

Explain the frame format in UART communication

START | D₀ | D₁ | D₂ | D₃ | D₄ | D₅ | D₆ | D₇ | D₈ | STOP

Band width

It is a data transmission of reference to the number of symbols transmitted per second. A symbol is a group of a fixed set of number of bits.

Data framing: UART transmits data in packets. Each data packet may contain one start bit, 5 to 9 data bits, an optional parity bit & 1 or 2 stop bits.

Start bit: When there is no data transmission, the UART transmission line is held at high voltage. This transition acts as the start bit in number.

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

Parity bits

The Parity bit is used to indicate the change in data transmission. The reasons for the change in data is mismatched band width, hysteresis magnetism or long distance data transfer.

Stop bit:

To mark the end of the data packet
The sending UART drives the data transmission
from a low voltage to a high voltage
for minimum of two bit duration

- (3) Explain the difference b/w in detail
- (4) Serial v/s Parallel

Serial

Parallel

- | | |
|--|---|
| • Data is transmitted
bit in a single line | • Data is transmitted
simultaneously through
group of lines |
| • Data confusion take
place | • no data confusion |
| • low speed transmission | • High speed transmission |
| • Implementation of
Serial links is not an
easy task | • Parallel - data links
are easily implemented
as H/w |
| • no Grounds Problem | • Go Grounds - Great
interface b/w parallel
lines |
| • The Bandwidth of
Serial wire is much
higher | • The Bandwidth of parallel
wire is much lower |

① Analogy V/s Digital

Analog

Digital

- Transmitted modulated signal → Transmitted Signal
Signal is analog in nature i.e. form of digital pulses.
 - Amplitude, frequency or phase Variations in the transmitted signal represent the message.
 - Noise immunity is poor for FDM & PDM.
 - Coding is not possible.
 - FDM is used for multiplexing.
 - Analog modulation System are Am, Fm, Pm, PAM, PWM.
- Amplitude width or position of the transmitted pulse is transmitted in the form of code words.
 - Noise immunity is excellent.
 - Coding techniques can be used to detect & correct the errors.
 - TDM is used for multiplexing.
 - Digital modulation system are PCM, PDM, ADPCM, DPCM.

② Synchronous V/s Asynchronous

Synchronous

Asynchronous

- Communicated in real time.
 - Guards interrupt in a working.
- not communicated in real time.
 - In Elements interrupt

- Sends the data & receiver the data & receiver the data on the same clock frequencies.
- Sends the data & receiver the data on different clock frequencies
- Faster
- Slower
- There is no overhead between Start & Stop of Extra Start & Stop bit
- uses Constant time interval
- used in Encrypted messages, or irregular time intervals
- used in chat rooms & Video
- used in E-mail

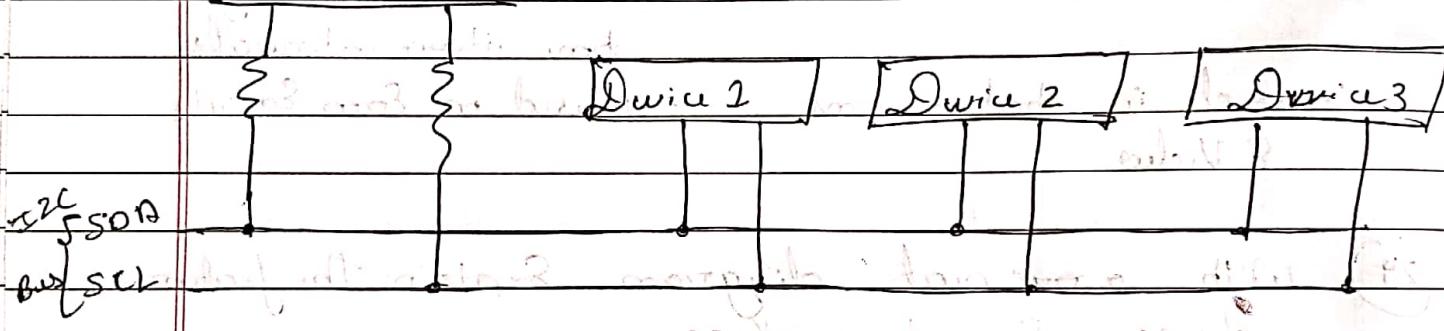
(24). With a ~~format~~ diagram Explain the features of I₂C & its working

- I₂C Communication is the short form for integrated circuit
- It is a communication protocol developed by Philips Semiconductors for the transfer of data between Central processor & multiple IC's on the same circuit board using just two common pins

Features:

- only two common bus lines (wires) are required to control any device on the I₂C network
- uses 7 bit addressing system to target a specific device on the I₂C bus lines

- I₂C Bus consists of just two wires & are named as Serial clock line (SCL) & Serial Data lines (SDA). The data to be transferred is sent through the SDA wire & is synchronized with the clock signal from SCL. All the devices on the I₂C network are connected to the same SCL & SDA lines.



- Both the I₂C bus lines (SDA, SCL) are operated as open drain drivers. It means that any device 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.

Master & Slave Devices

The devices connected to the I₂C bus are categorized as either master or slaves. At any instant of time only a single master say is active on the I₂C bus. It controls the SCL clock line & decides what operation is to do.

Working

In I₂C Communication is initiated by a master device either to send data to a slave slave devices or to receive data from it.

Sending data to a Slave device:

- The master device sends the start condition.
- The master device sends the 7 address bit which corresponds to the slave device to be targeted.
- The master device sets the Read/Write bit to '0' which signifies a write.
- Now 2 situations is possible:

If no slave device matches with the address sent by the master device then set ACK/NACK bit starts at '1' (default). This signals the master device that the slave device identification is unsuccessful.

Reading data from Slave:

The master device sets the Read/Write bit to '1' instead of 0 which signals the targeted slave device that the master device is expecting data from it.