Name: Manasa M.K.
Assignment
Roll No: 811039034

Embedded Systems

15/10/2021

MCA C Microcontroller and it's Application)

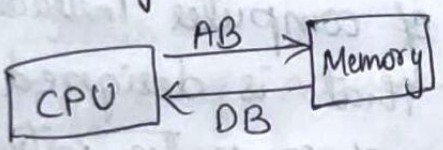
Virtualisation is technology that lets you create useful IT services using resources that are traditionally bound to Hardware.

Assignment

① Von-Neumann & Hardward architecture, Explain them with a neat diagram.

Ans: **Von-Neumann Architecture** [Basic, primitive]

* Widely used, single chip, single memory



{ DB - Data bus
AB - Address Bus

* All parts are controlled by control unit CPU and connected together by BUS.
* Data can pass through bus in Half Duplex mode to and from CPU.
* Memory holds programs & data known as stored program concept.
* Memory is split to small cells with the same size. Their original numbers are called address numbers.

Advantages:

* Control Unit gets data & instruction in the same way from one memory. It simplifies design & development of the control unit.

Disadvantages: Serial instructions processing does not allow parallel execution. Parallel execution are simulated later by the OS.

* One Bus is a bottle neck.
* Instructions are stored in same memory as the data can be accidently rewritten by an error in a program.

---

(2) List the difference b/w RISC & CISC

Ans: Difference b/w RISC & CISC are as follows.

CISC: CISC was developed to make compiler development easier & simple. CISC is complex instruction set computer.

RISC: RISC is designed to perform a smaller numbers of and types of computer instructions. It is a Microprocessor that is designed to perform smaller number of computer instructions. RISC is Reduced instruction set computer.

CISC	RISC
* More set of Instructions	* Lesser set of instructions
* More addressing mode	* Lesser addressing mode.
* Minimum amount of RAM	* More amount of RAM.
* Focus is on Hardware to optimise.	* Focus is on software to optimise
* High Power consumption	* Low power consumption.
* CISC has variable instruction format	* RISC has fixed instruction format.
* Single Register set	* Multiple Register set
* Less pipelined	* Highly pipelined
* Microcode control	* Hardwired control
* Instruction → variable size	* Instructions → fixed mechanisms.

* Instructions take more than 1 clock to execute	* Instructions take one clock cycle to execute.
* Simple compiler is Enough.	* Enough complexity → compiler.
* Segmented memory model.	

③ Subtract two hexadecimal values which are loaded in R3 & R9.

Ans: There

③ List the numbering system available and give example for one numbering system to another.

Ans: Types of number systems are as follows :-

① Binary Number System (Base-2)
② Octal Number System (Base-8)
③ Decimal Number System (Base-10)
④ Hexadecimal number System (Base-16)

* The Binary number s/m is used to store the data in computers.
* The advantage of the octal number s/m is that it has fewer digits when compared to several other systems, hence there would be fewer computation errors.
* The decimal numbers system is the s/m that we use in daily life.
* The hexadecimal number s/m is used in computers to reduce the large sized strings of the binary s/m.
* Number s/m helps in representing the numbers in a small symbol set.

* Binary numbers are mostly used in computers that use digits like 0 and 1 for calculating simple problems.
* The number s/m also help in converting one numbers s/m to another.
* Conversion from one number s/m to another number s/m.
* To convert a no from one of the binary/octal/hexadecimal s/m to one of the other s/m, we first convert it into decimal s/m & then we convert it to the required s/m by using the below process.

Eg: i) 1010111100_2 to decimal

1 0 1 0 1 1 1 1 0 0

$2^0 \times 0 = 0$

$2^1 \times 0 = 0$

$2^2 \times 1 = 4$

$2^3 \times 1 = 8$

$2^4 \times 1 = 16$

$2^5 \times 1 = 32$

$2^6 \times 0 = 0$

$2^7 \times 1 = 128$

$2^8 \times 0 = 0$

$2^9 \times 1 = 512$

700

$$\boxed{1010111100_2 = 700_{10}}$$

16 | 700

16 | 43 – 12 → (C)

2 – 11 → (B)

$\rightarrow 700_{(10)} = 2BC_{(16)}$

8 | 700
8 | 87 - 4
10 - 7
1 - 2 = $(1274)_8$

2 | 700
2 | 350 - 0
2 | 175 - 0
2 | 87 - 1
2 | 43 - 1
2 | 21 - 1
2 | 10 - 1
2 | 5 - 0
2 | 2 - 0
1 - 0

$(1010111100)_2$

---

(4) With a neat example. Explain arithematic & logical shift operations.

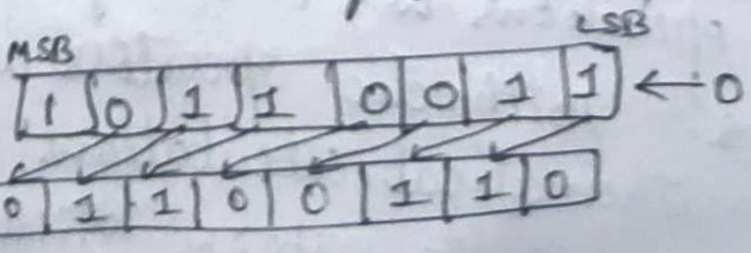
Ans: Logical Shift: * A left logical shift of one position moves each bit to the left by one.

* The varant least significant (LSB) bit is filled with zero and the most significant bit (MSB) is discarded.

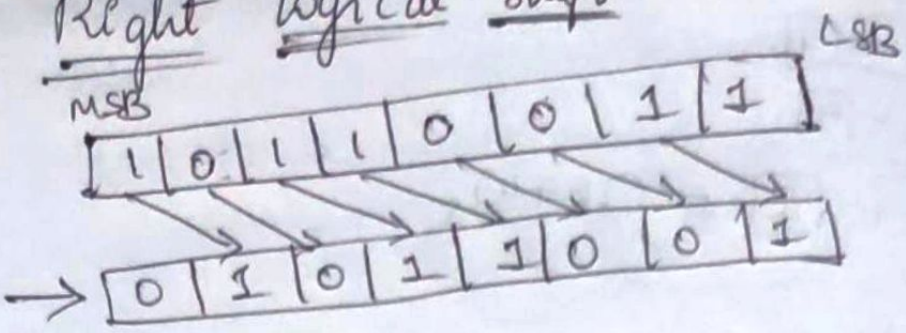
* A right logical shift of one position moves each bit to right by one.

* The vacant least significant bit (LSB) is discarded and the most significant (MSB) is filled

Left logical Shift



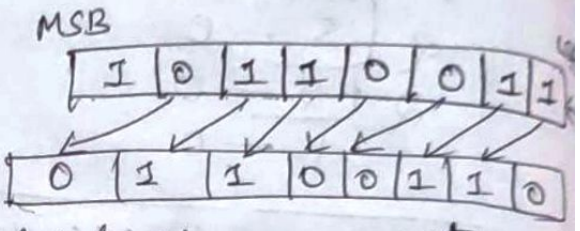
Right logical shift



Arithmetic shift :-

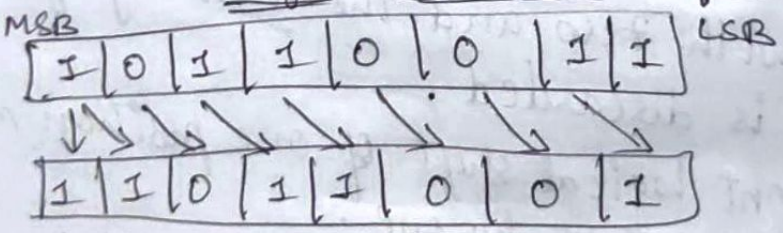
* A left arithmetic shift of one position moves each bit to the left by one.
* The vacant least significant bit (LSB) is filled with zero and the most significant bit (MSB) is discarded.

Left Arithmetic Shift



* Right arithematic shift of one position moves each bit to the right by one. The least significant bit is discarded and the vacant (MSB). Most significant bit is filled with the value of the previous MSB.

Right Arithematic shift



---

(6) Explain the basic arithmetic and logical operations w.r.t binary (AND, OR, XOR).

Ans:- Arithmetic operators are used to perform mathematical calculations. Logical operators are used to perform logical operations & include AND, OR (or) NOT of OR.

* Boolean operators include AND, OR, XOR (or) NOT of OR, Can have one of two values 'True' of 'False'.

* The arithematic operations are addition (+) subtraction (-), multiplication (*) & division (/) Exponential (^)

There are 3 common logical operations are:

* AND
* OR
* NOT

* The logical operators are often used to help create a test expression that controls program flow. This type of expression is also known as a Boolean expression, because they create a Boolean answer (or) value when evaluated.

<u>Truth Tables</u>: The common way to show logical relationship is in truth table.

x	y	x (and) y	x (or) y
false	false	false	false
false	true	false	true
true	false	false	true
true	true	false	true

x	not x
false	true
true	false

Operator	Meaning	Example	Result
&&	logical AND	(5>2) && (5>3)	false
||	logical OR	(5>2) || (5>3)	True
!	logical NOT	! (5>2)	True

(7) What is Endianess? Explain with example little endian and Big Endian along with significance.

Ans: Endianess is the order (or) sequence of bytes of a word of digital data in computer memory. Endianess is primarily expressed as:-

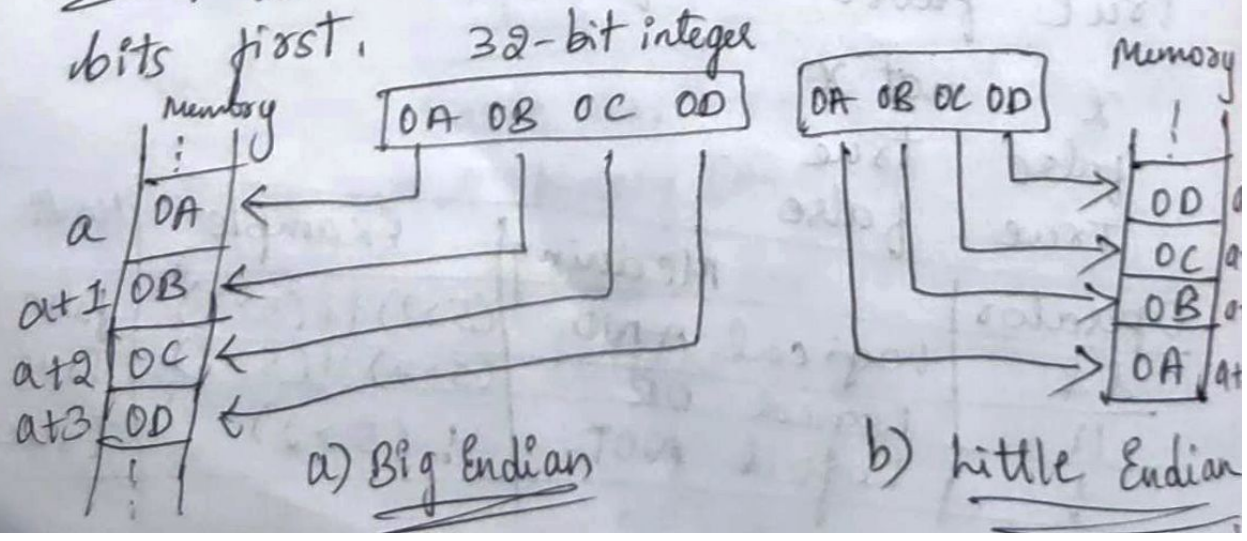
i) Big Endian (BE)
ii) Little Endian (LE)

* A big Endian s/m stores the most significant byte of a word at the smallest memory address and least significant byte of the word at the largest memory address.

* A little Endian s/m stores the least significant byte of the word at the smallest address and most significant byte of the word at the largest memory address.

* Endianess may also be used to describes the order in which the bits are transmitted over a communication channel.

For Eg: Big Endian in a communications channel transmits the most significant bits first.



* The two diagrams show how two computers use different Endianness store a 32-bit integer with the value of 0X0A0B0C0D.
* In both cases the integer is broken into 4-bytes 0X0A 0X0B 0X0C 0X0D are stored in four sequential byte location in memory starting with the memory location with address a, a+1, a+2, a+3. {where a is address}
* The diagram (b) shows a computer using little Endian. This starts the storing of the integer with the LSB 0X0D at address a & ends with the MSB 0X0A at address a+3.
* Since each computer uses its same Endianess to both store & retrieve the integer, the results will be same for both computers. Issues may arise when memory is addressed by bytes instead of integers (or) when memory contents are transmitted b/w computers with different Endiansess.

---

8) What is virtualisation? Types of virtualization.

Ans * Virtualisation is the technology that lets you create useful IT services using resources that are traditionally bound to Hardware.

* It allows to use a physical machine's full capacity by distributing its capabilities among many users (or) environments.

Types of Virtualisation:

* Presentation virtualisation: * It is an application delivery method that delivers desktops (or) applications from a shared servers.

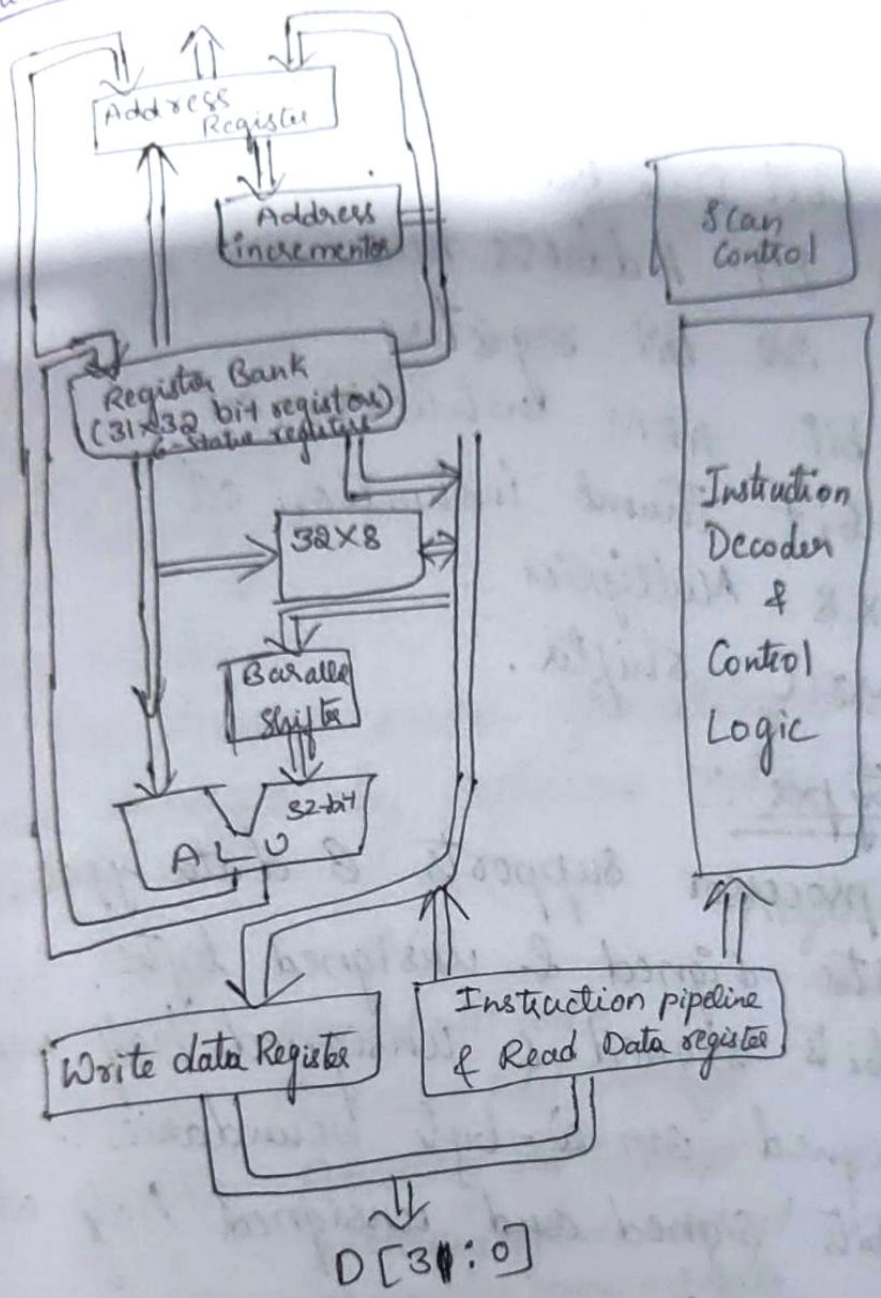
* This Enables access to client applications from a central server that is connected with clients.
* This intiate a presentation instance. The only resources shared with the client is the graphical user interface as well as the mouse/keyboard.

* Virtual Desktop Infrastructure (VDI):
* Sharing similarities with present virtualization, VDI solutions are also a remote display protocol that hosts centrally managed virtual machines that client PC's are connected to one-on-one network.

* Application virtualization:-
* It is capable of allowing applications to run in Environments that are foreign to them.
* For Example, Wine allows microsoft windows apps to run on linux.
* By Establishing a common software baseline across multiple computers with an organization application virtualisation also reduces s/w integration and administrations costs.
* Finally it enables simplified operating s/m migrations, where by applications can be transferred to removable media (or) b/w computers without having to install them; Effectively becoming portable Software.

Write a neat diagram. Explain the ARM 7 architecture

Ans:



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 sets

- 32 bit ARM instruction set
- 16 bit thumb instruction set

* Von - Nueman Architecture
* 3-stage Pipeline — Fetch
  - decode
  - Execute

* 32 - bit Data Bus
* 32 bit Address Bus
* 37 32 bit registers
* 37 - bit ARM Instruction set
* 16 - bit Thumb instruction set
* 32 x 8 Multiplier
* Barrel shifter.

<u>Data types</u>:

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

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

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

ARM Co-processor supports floating point values.

Ans: Most programs operates

Programming model for ARM7.

Each instruction can be viewed as performacy a defined transformation of the states

* Visible Register
* Invisible Register
* System Memory
* User Memory

Processor Modes :

→ ARM has seven basic operating modes.

→ Modes changes by softwares control (or) external intercepts

PSR[4:0]	Mode	USE	Register
10000	User	Normal user code	user
10001	FIQ	Processing Fast Interrupt	-fiq
10010	IRQ	Processing std interrupt	-irq
10011	SVC	processing software interrupt	-svc
10111	Abort	processing memory faults	-abt
11011	Undefined	Handling undefined inst types	-und
11111	System	Running privilged OS	user

Most programs operates in user mode ARM has other privileges operating modes which are used to handle exceptions, Supervisor calls & system mode.

→ More access rights to memory systems & co-processor

→ Current operating mode is defined by CPSR[4:0]

Privileged Modes :-

→ Having some protective previlages
→ System-level function 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.

<u>Each modes can access</u>
- a particular set of $R_0 - R_{12}$ registers.
- stack pointer, R_{14} - Link register.
- Program counter r_{15} (PC)
- the current program status register (PSR)

Privliged modes can access a particular SPSR saved program status register.

	FIQ	IRQ	SVC	Undefined	Abort
r_0					
r_1					
r_2					
r_3					
r_4					
r_5					
r_6					
r_7					
r_8	r_8				
r_9	r_9				
r_{10}	r_{10}				
r_{11}	r_{11}				
r_{12}	r_{12}				
r_{13}	r_{13} (SP)	r_{13} (SP)	r_{13} (SP)	r_{13} (SP)	r_{13} (SP)
r_{14}	r_{14} (LR)	r_{14} (LR)	r_{14} (LR)	r_{14} (LR)	r_{14} (LR)
r_{15}					
CPSR	SPSR	SPSR	SPSR	SPSR	SPSR

With a neat diagram, Explain three Stage pipeline of ARM

Ans: A pipelining is the mechanism used by RISC processor to execute instructions.

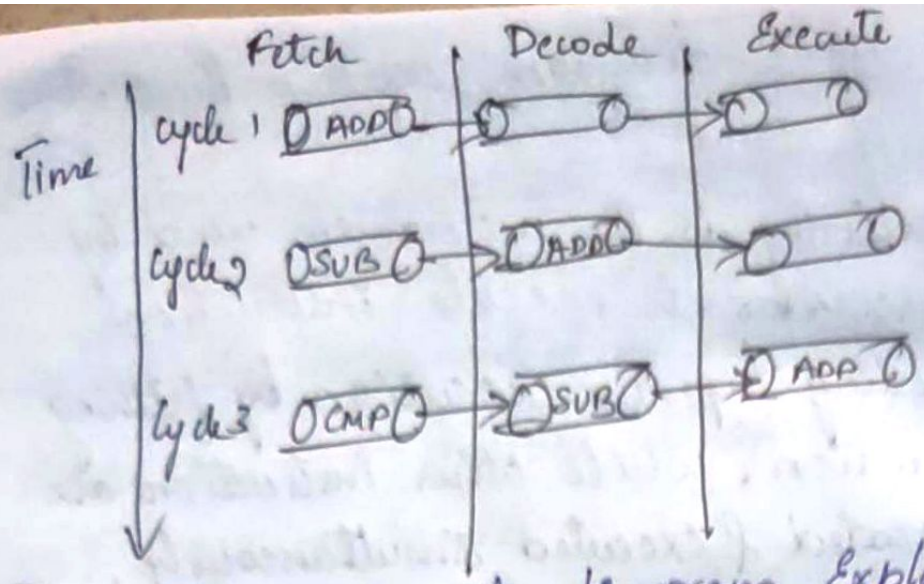
→ By speeding up the execution by fetching the instruction, while other instructions are being decoded & executed simultaneously.

→ 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 or µc.

The ARM 7 has three stages pipeline:

- Fetch: the instruction is fetched from memory.
- Decode: The instructions opcods & operands are decoded to determine what functions perform.
- Execute: The decoded instruction is executed.

Each of these operations require one clock cycle for typical instructions. Thus a normal instruction requires three clock cycles to completely execute, known as the latency of instruction execution. Because the pipeline has three stages an instructions execution computed in every clock cycle. In other words the pipeline has a throughput of one instruction per cycle



(7) With the neat diagram. Explain CPSR register

N Z C V Q undefined IFT mode

Condition code flags.

- N: Negative result from ALU
- Z: zero result from ALU
- V: ALU operation overflowed
- C: ALU operation carried out

Sticky overflow flag - Q flag

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

Interrupt Disable bits

→ I = 1 Disables the IRQ

→ I = 0 Disables the FIQ.

T bit.

- Architecture XT only
- T ⇒ 0 processor in ARM state
- T ⇒ 1 processor in Thumb state

Mode Bits - specify the processor mode.

CPSR → current processor status register Holds the information about the current status of the processor.

SPSR → saved processor status register Holds the information on the processor state before the s/m changed to this mode i.e., processor status just before an exception.

Explain the seven different mode in ARM.

Ans: The ARM7TDMI processor has seven modes of operations:

→ user mode is the usual ARM program execute on 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 reading

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

→ Abort mode is a priviliged ~~user~~ mode for the operating system.

We can only enter system mode from another priviliged mode by modifying the mode bit of the current program status register (CPSR)

Undefined mode is entered when an undefined institution is executed.

Modes other than user mode are collectively called as priviliged modes. Priviliged modes are used service interrupts (or) exceptions (or) to access process resource.

Modes other than user mode are collectively

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

Explain the nomenclature in ARM

Ans:- ARM was originally from Acorn computers Ltd. First RISC processor for commercial use

ARM7TDMI Processor:

32-bit processor Advanced machine

T → Thumb Architecture extension

D → Debug extension

M → Enhanced extension

I → Incircuit Emulation

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

x → Series

y → Memory management Unit

Z → Cache
T → Thumb 16-bit decoder
D → JTAG Debugger
M → Fast Multiplier
I → Embedded ICE (Incircuit Emulation)
E → Enhanced Instruction for DSP
J → JAVA acculeration by Jazelle
F → Floating point
S → Synthesisable version.

What is JTAG? Explain JTAG State diagram.

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

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

JTAG i.e., Join Test Action Group is an industry standard for verifying design & testing printed circuit boards after manufacture.

JTAG implements standards of on chip instruction is 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 to the s/m without requiring direct external access to the s/m address & direct buses. The interface connects to an chip test Access port (TAP)

that implements a stateful protocol to access a set of test register.

Test logic reset
run test idle
Select DR scan
Capture DR
Shift DR
Exit IDR
Pause DR
Exit 2 DR
Update DR

Select IR Scan
Capture IR
Shift IR
Exit IDR
Pause IR
Exit 2 IR
Update IR

What is Single Tasking? Give examples of processor applications.

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

Microcontrollers are 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.

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

The memory can be defined as a collection of data in a specific format. It is used to store instructions & processed data. The memory comprises a range array (or) group of words (or) bytes, each with its own location. The primary method of a computer s/m is to collect & execute programs. These programs along with the information by access, should be in the main memory during execution. The CPU fetches the instructions from memory according to the value of the program counter.

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

Main Memory is also known as RAM (Random access memory). This memory is a volatile memory. RAM lost its data when a power interruption occurs.

Memory management

In a multiprogramming computer the operator/O.S/m resides in a part of memory and is used by multiple processes. The task of subdiving the memory among different process is called memory management. Memory management is a method in operating system to manage operations b/w main memory & disk during process execution. The main aim of memory management is to achieve efficient utilization of memory.

Why Memory Management is Required?

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

Eg:- IBM system/360 Model 67, IBM system 1370

ARM ARM architecture based application processors implement an MMU defined by ARM's virtual memory system architecture. The current architecture defines PTE's for describing 4KB and 64KB pages 1MB sections & 16MB super sections legacy versions also defined 1KB tiny space.

Write a ASM program to find the endianess of a given number.

```
AREA EX24, CODE, READWRITE
ENTRY

START
    ADR R0, LIST
LIST MOV R10, #0X2
    LDR R1, [R0]
    LDR R3, [R0, #4]!
    LDRB R3, [R0], #1
    LDRB R4, [R0, R10]!
    LDRSB R5, [R0], #1
    LDRSH R6, [R0]

STOPB STOP
LIST DCB 0X34, 0XF5, 0X32, 0XE5, 0X01, 0X02,
         0X08, 0X8E, 0X56, 0X58, 0X67

    END
```

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

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

A group of 4-bits is called a nibble.

A group of 8-bits make a byte.

P.T.O.

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

Word is another length that gets thrown out from time to time. Word is much less sounding & more ambiguous. The length of a word is usually dependent on the architecture of a processor. It could be 16 bits, 32, 64 (or) even more. The terms half word (or) single word are often used in contemporary computing to refer to common word sizes relative to a 32-bit base word size.

half word = 16 bits

Word = 32 bits

Explain the word align & Half word align in ARM memory.

Ans:- Different processor have different defination of words. As the name implies, a half word for 32-bit processors, a word is 32-bit (4 bytes) As the name implies, a half word is 16-bit, for a 8-bit processor, a word a 16-bit (2 bytes), for 8 bit processor word is 8 bits. Word alignment: The stored address are adjacent and can be divided by 4, the last two digits are 00.

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

ARM architecture requires 32-bit ARM instructions that must be word aligned & stored in memory and 16 bit thumb Instructions requires Half Word aligned & stored. Therefore in ARM state the value of R15 is always divisible by 4, that is lowest 2 bits of the R_{15} 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 (or) 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 instruction forms a program to perform a specific tasks.

2 Components
- Opcode field → Specifies how data manipulates
- Address field → Specifies the data locations

When 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 Instructions.

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.

One-address Instructions :-

This uses an implied accumulator register for data manipulation and the other is in the register & memory location. Implied means that the CPU already knows that one operand is in the accumulator so there is no need to specify it. Eg :- LDR addr

ACC ← (addr)

Two address Instructions :-

Here two address can be specified in the instruction. In the one address instruction, the result can be stored in the accumulator, Here the result can be stored in different locations i.e., Register (or) memory location. But require more number of bit to represent the address.

Eg : MOV R1, R2

R1 ← [R2]

Three address Instructions :-

This has three address field to specify a register (or) memory location. Program created are much short in size but no of bits per instructions increases. Program created are much short in size but no of bits per instruction increases. These instructions make creation of program much easier but it does not mean that program will run much faster because how instruction only contain more information but each micro operation will be performed in one cycle only Eg: ADD R_3, R_1, R_2

$R_3 = R_1 + R_2$.

19. Explain the LPC2148 Microcontroller Block diagram.

Ans: LPC2148 Features

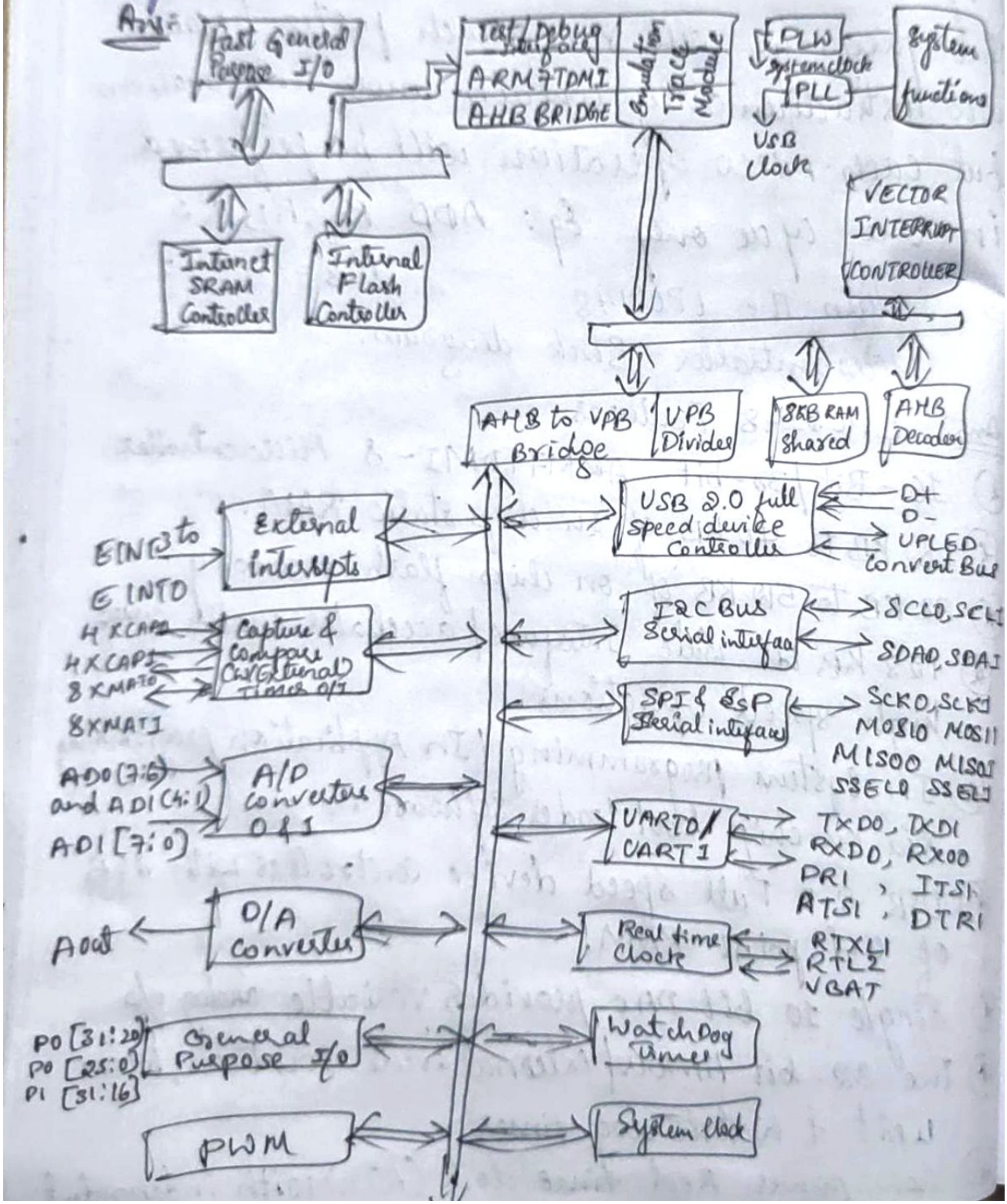
- 16-Bit/32-bit ARM7TDMI-S Microcontroller
- 8 KB to 40 KB of on chip static RAM.
- 32 KB to 512 KB of on chip flash memory.
- 128 KB bit wide Interface/accelator enables 60MHz high speed operations.
- In System programming / In Application programming via on chip boot loader software.
- USB 2.0 Full speed device controller with 2KB of end point RAM.
- Single 10 bit DAC provides variable analog o/p
- Two 32-bit timers/external event counters, PWM unit & Watch dog timer.
- Low power Real time clock (RTC) with independent

power & 32KHz clock i/p.

- 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 ckt CPU operating voltage range of 3.0V to 3.6V.

Explained the LPC2148 Block diagram as follows

Ans:



explain the LPC2148 Micro controller GPIO pins.

Ans:- GPIO → General Purpose Input/output A 32-bit register used to select the functions of the pins in which the user needs it to operate. There are four functions per each pin of the controller, which the first function is GPIO. It means that the pin can either act as an i/p (or) o/p with no specific functions.

There is totally three PIN registers in LPC2148 controller in order to control the functions of the pins in the respective ports. The classification is given below:

PINSEL0 – Controls functions of port 0.0 - Port 0.15
PINSET1 – Controls functions of port 0.16 - Port 0.31
PINSET2 – Controls function of port 0.16 - Port 0.31

where the 00 of 32 bits registers are GPIO config.

LPC2148 has two 32 bit General purpose I/O ports. A total of 30 I/O and a single o/p 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.

* IOPIN
* IODIR
* IOSET
* 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 i/p, o/p (or) as GPIO (or) an alternate digital functions.

As an Example:- that particular port pin may have GPIO i/p, o/p, UART receive, PWM o/p

as selectable functions. Any configuration of that pin will allow its current logic state take read from the IOPIN register.

If a pin as an analog function as one of its options the pin state cannot be read if the analog configuration is selected. Selecting the pin as a A/D i/p disconnects the digital features of the pin. The pin value read in the IOPIN register is not valid writing to the IOPIN register stores the value in the port o/p register, bypassing the need to use both the IOSET & IOCLR registers to obtain the entire written value. This feature should be used carefully in an application since it affects the entire port.

<u>IOSET</u>: This register is used to produce a HIGH level output at the port pins configured as GPIO is an OUTPUT mode. Writing 1 produces HIGH level at the corresponding port pins writing 1 to the corresponding bit in the IOSET has no effect. Reading the IOSET register returns the value of this registers as determined by previous writes to IOSET register as determined by previous writes to IOSET register returns the value of this register, as determined by previous writes to IOSET register returns the value of this registers are determined by previous writes to IOSET register returns the value of this register, as determined by previous

writer to IOSET & IOCLR. This value does not reflect the effect of any outside world influence on the IO pins.

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

IOCLR : This register is used to produce a Low-level o/p at port pins configured as GPIO in an output mode. Writing 1 produces a Low level at the corresponding port pin & clears the corresponding bit in IOSET register. Writing 0 has no Effect. If any pin is configured as an input (or) a secondary function, writing to IOCLR has no Effect.

With a neat diagram Explain Baud rate & Bit rate. Explain the Calculations.

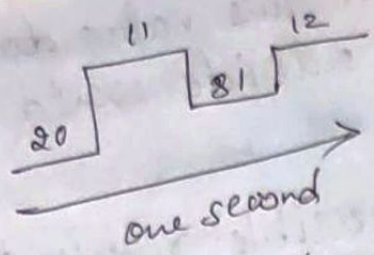
Ans:- Baud :- How many times a Signal changes per second.

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

Bit rate is controlled by based and no of signal levels.

Baud rate :

→ Number of time lines changed per second.
→ Set Baud rate be 8 (4 changes per second)
⇒ Set bits per line changes ~~per~~ & be 2
→ Bit rate = 8 bits per second.
→ Bit rate = 8 × Baud rate in this example.



→ Baud rate defines the switching speed of a signal.

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

→ For a binary two level signal a data rate of one out per second is equivalent to one baud.

An analog signal carries 4 bits in each signal unit. If 1000 signal units are sent per second find the Baud rate of the Bit rate.

1 Bit → Symbol

Bit rate = Baud rate

$\frac{1000}{1000}$ = Baud rate

bands/sec

Bit rate = 1000 × 4 = 4000 bps.

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

Baud rate (or) symbol rate) is "s"

General formula :-

$b = s \times n$

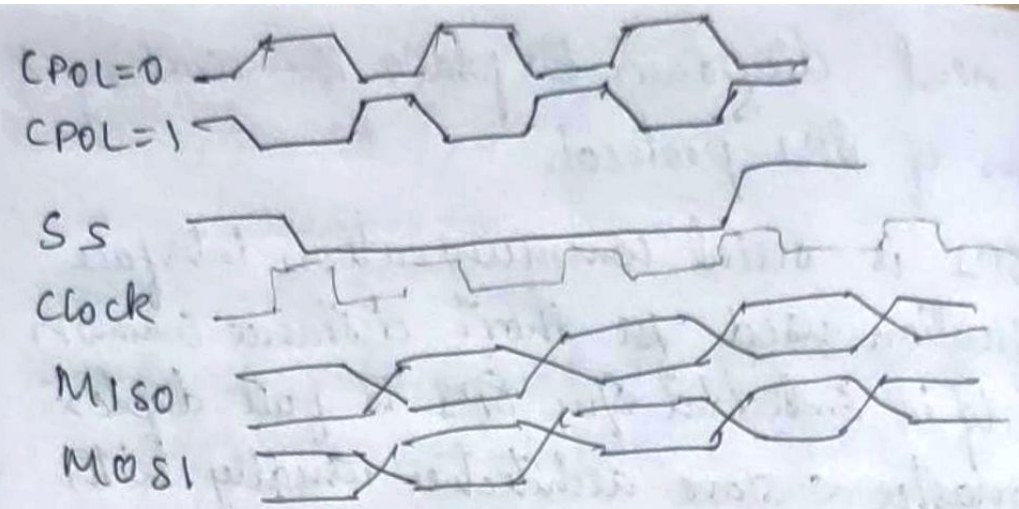
→ b = Data Rates (bits per second)

→ s = Symbol rates (symboles/sec)

→ n = no of bits per second

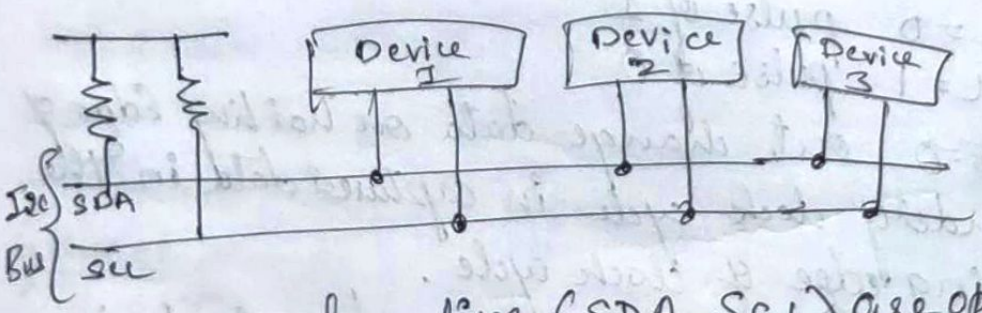
if n = 1 → Baud rate = Bit rate

n = 4 → Bit rate = 4x Baud rate.



With neat diagram explain features of I2C & its working.

Ans: (*) Half Duplex Serial communication.
(*) Synch communication protocol.
(*) Only two communication bus lines;
(*) Adjustable Data rate speeds.
(*) Simple mechanism for validation of transfered.
(*) Uses 7-bit addressing s/m to Target specific device on Bus.

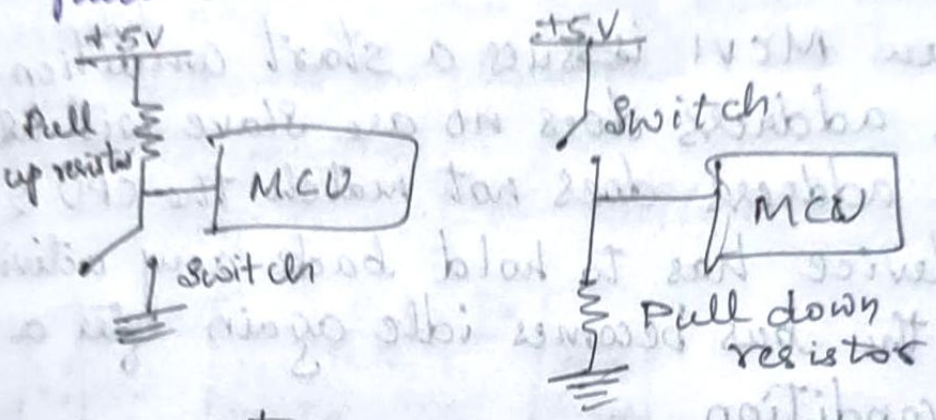


Both I2C bus line (SDA, SCL) are operated as open drain drivers. It means that any device on I2C network can drive SDA & SCL but they cannot drive them high, So a pull up resistor is used for each bus line to keep them high by default, when

master device wants to transfer data (or) from a slave device, it specifies this particular slave device address on SDA line & then proceeds with the transfer so effectively communication takes place b/w master device & particular slave.

With a neat diagram, explain pull up & pull down resistor.

Ans:



Pull up resistor

It is used to establish an additional loop over the critical components while making sure that voltage is while making well-defined even when switch is open. It is used to ensure that a wire is pulled to a high logical level in absense of an input signal. Pull up resistor with a fixed value was used to connect voltage supply and a particular pin in digital circuit.

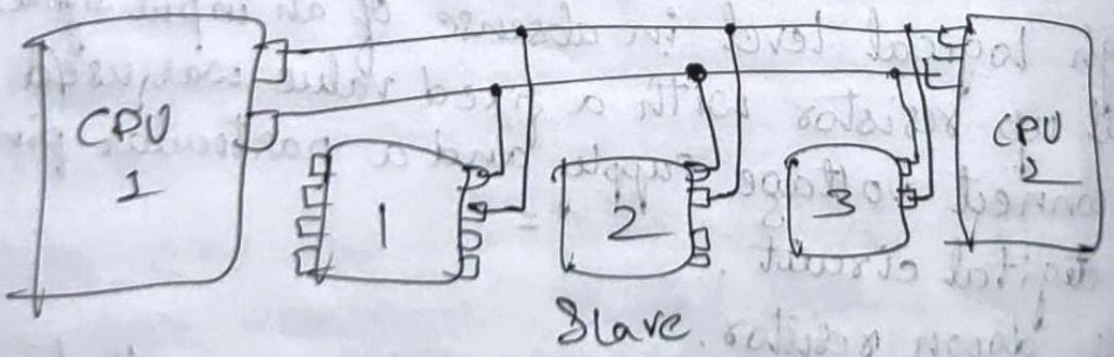
Pull down resistor.

A pull down resistor is used to ensure that input to logic systems settle at expected logic level whenever the external the external device are disconnected (or) of high impedance.

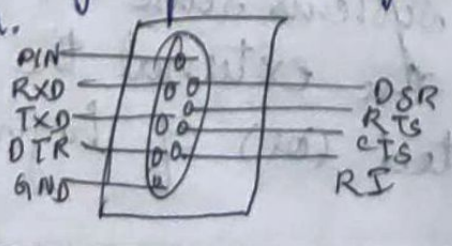
* It ensures that the wire is at a defined low logic level when there are no active connections with other device is connected.

Arbitration in I2C

* Bus arbitration occurs when two (or) more master start a transfer at same time.
* More than one device can be active in the system.
* When MCU1 issues a start condition and sends address does no are slave will listen. If the address does not match the CPU 2 this device has to hold back any activity until the bus becomes idle again after a stop condition.
* As long as two MCU's monitors what is going on bus and as long as they are aware that a transaction is going on because that back issued command that back issued command was not a STOP there is no problem.



Explain working of DB9 pins & handshaking with the modem.

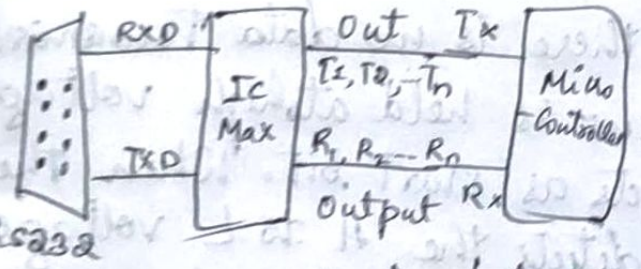


PIN	SIGNAL	SIGNAL NAME	DTE Sig direction
1	DCD	Data carrier detect	In
2	RXD	Receive data	In
3	TXD	Transfists data	Out
4	DTR	Data Terminal ready	Out
5	GND	Ground	–
6	DSR	Data set ready	In
7	RTS	Request to send	Out
8	CTS	Clear to send	In
9	RI	Ring indicator	In

Handshaking modem is what occurs when the receiving modem answers the phone call & two modems begin to communicate.

Explain RS232 Connection with ML.

Ans:-



Several Devices collect data from sensors and need to send it to another unit like a Computer for further processing.

Serial communication on 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 on bit at a time.

An Important parameter considered while interfacing serial signal port is Baud rate which is speed at which data is transmitted serially. Microcontrollers can be set to transfer and receive signal data at difference baud rate using software Instruction.

Explain frame format in UART Communication

Ans:- **Baud Rate**:- Baud rate in a data Tx refers to the number of symbols transfered per second. A symbol is a group of a fixed no of bits.

Data Framming:- 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	D0	D1	D2	D3	D4	D5	D6	D7	D8	STOP

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

Start Bit: When there is no data Transmission the UART Tx line is held at high voltage To transition acts as start bit. When the receiving UART detects the H to L voltage transition, it begins reading the data frame at freq of baud rate.

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

Parity bits: This is used to indicate the change in data during tx. The reasons in ~~data during~~ tx. The reas. from the change in data is mismatched baud rates, Electro-magnetism & long distance data transfer.

Stop bits:- To mark the end of data packet the sending UART drives the data

transmission line from a low voltage to high voltage from a minimum of bit duration.

Explain difference b/w Serial v/s Parallel, Analog v/s Digital, Synchronous v/s Asynchronous

Ans:-

Serial	Parallel
• Data is transmitted bit after bit in a single line	• Data is transmitted simultaneously through group of lines.
• Data Congestion takes place.	• No data congestion.
• Low speed transmission	• High speed transmission
• Implementation of serial linker is not an easy task.	• Parallel data links are easily implemented as hardware
• No cross talk problem	• Cross talk creates interference between parallel lines.

Analog	Digital
Transmission modulated signal is analog in nature	• Transmission signal is digital i.e., train of pulses
Noise immunity is poor for FM & PM	• Noise immunity is excellent
Coding is not possible	• Coding technique can be used to detect & correct error.
FDM is used for multiplex	• TDM is used for multiplexing
Analog modulation s/m are AM, FM, PM, PAM, PWM.	• Digital modulation s/m are PCM, DM, ADM, PCM

P.T.O.

Synchronous	Asynchronous
• Communicated in real time	• Communicated in real time
• Creates interrupts in a working	• Eliminates Interrupts
• Faster	• Slower
• No overhed of extra start & stop bit.	• User start & stop bit
• Uses constant time interval.	• Uses random or irregular time intervals.
• Used in chat rooms.	• Used in emails.

With neat diagram. Explain the working & features of SPI protocol.

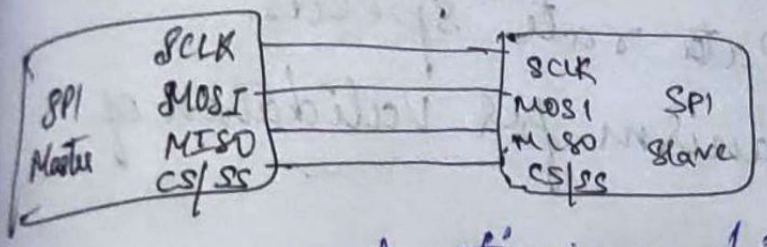
Ans: SPI is serial communication interface specification used for short distance common primarily in Embedded S/m SPI is full duplex with master & slave architecture usually with single master SPI is wire serial Bus.

SCLK - Serial clock

MOSI - Master out Slave In

MISO - Master in Slave Out

CS/SS - chip/Slave Select.



In SPI neat timing diagram. Explain CPHASE & CPOL usage.

Ans:- CPOL determines the polarity of clock

CPOL=0 pulse of 1

CPOL=1 pulse of 0

CPHASE=0 out change data on trailing edge of precendery clock cycle in captures data in the leading edge of clock cycle.

CPHASE=1 Out side changes data in leading edge of clock cycle, while 'in' slide captures data on trailing edge of clock cycle.