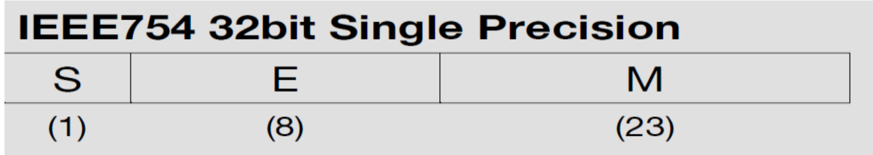
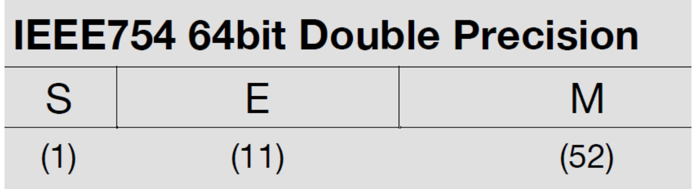
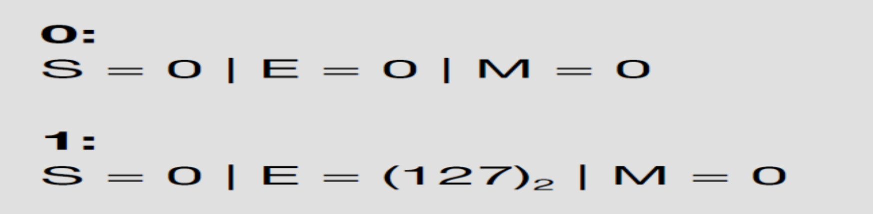
1.	What are the three types of buses?	**Address bus** | Carries the address for the current memory or I/O operation. **Data bus** | Transfers data between the processor and memory or I/O. **Control bus** | Issues control signals to decide the type of operation like read, write etc. Together they are called the SYSTEM bus
2.	What is an instruction cycle?	The total process of fetching an instruction from the memory, decoding the opcode to understand the operation and finally executing the instruction is called an instruction cycle.
3.	What is PC?	**PC | Program Counter** It is a register present inside the processor. It contains address of the next instruction. PC gets incremented as soon as any instruction is fetched.
4.	What is SP? Where it is used?	**SP | Stack Pointer** It is a register present inside the processor. It contains address of the top of stack. SP gets decremented in every Push and incremented in every Pop operation. **Used in Interrupt Handling (when multiple interrupts occur together)**
5.	What are GPRs?	**GPRs | General Purpose Registers** These are registers present inside the processor. They are available to the programmer for storing operands and getting results in a program.
6.	What is the difference between Von Neumann and Harvard Model?	**Von Neumann Model** Common memory used to store programs and data. E.g.:: 8085, 8086. **Harvard Model** Separate program memory and data memory. E.g.:: 8051.
7.	Explain Flag register?	It gives the status of the current result. It has flag bits that indicate if there was a carry, if the result was zero, the sign of the result, its parity etc. Flags are changed by the ALU after every operation.

8.	Why are signed numbers stored in 2's complement form?	Because it does not produce a negative zero. 2's complement of any number gives –(that number) but 2's complement of 0 gives back 0 itself.
9.	What is half adder and a full adder?	**Half Adder |** Adds the two current bits. Produces a sum and a carry. Cannot be scaled to a bigger adder. **Full Adder |** Adds the two current bits along with the carry of the previous stage. Produces a sum and a carry. Can be scaled (combined) to form a bigger adder.
10.	How is subtraction performed?	We never really "subtract" Subtraction is performed by the same circuit used for addition by simply adding the 2's complement of the number. Note: Draw Adder/ Subtractor circuit
11.	Why cant we do multiplication by repeated addition?	Because it is too slow! 1000 x 1000 will require 1000 steps! Hence we use faster algorithms like Booth's Algorithm!
12.	Briefly explain Booth's algorithm?	Keep checking bit transitions in the multiplier from right to left starting with an imaginary 0 beyond the LSB. Do the following actions: **0 to 1 |** Subtract Multiplicand, Right Shift **1 to 0 |** Add Multiplicand, Right Shift **0 to 0 |** Right Shift **1 to 1 |** Right Shift
13.	Why cant we do division by repeated subtractions?	Because it is too slow! 1000 ÷ 1 will require 1000 steps! Hence we use faster algorithms like restoring and non restoring division!
14.	Explain restoring division in brief?	At each step, **left shift** the dividend and **subtract** the divisor. **If result is positive** (or 0) then successful. Quotient bit is 1. Restoration not needed.

		If result is negative, unsuccessful. Quotient bit is 0. Restore by adding back the divisor. **Repeat** for all bits of dividend.
15.	Explain non restoring division in brief?	At each step, **left shift** the dividend and **subtract** the divisor. **If result is positive** (or 0) then successful. Quotient bit is 1. Restoration not needed. **If result is negative**, unsuccessful. Quotient bit is 0. Don't Restore! Next step will be addition instead of subtraction. **Repeat** for all bits of dividend.
16.	Describe Normalization and normalized form of a number ?	Normalization represents a number such that there must be only one non-zero digit to the left of the point. **Normalized form of a number** $(-1)^S$ x 1.M x 2^E S: Sign | M: Mantissa | E: Exponent
17.	Convert 0101.001 into Normalized form?	**$(-1)^0$ x 1.0101001 x 2^2**
18.	Explain Single Precision format?	**IEEE754 32bit Single Precision**  IEEE754 32bit Single Precision S E M (1) (8) (23) Bias value: 127
19.	Explain Double Precision format?	 IEEE754 64bit Double Precision S E M (1) (11) (52)

		Bias value: 1023
20.	Explain Floating point Addition in brief?	Compare exponents. If Equal, add the mantissas, else Shift the mantissa till exponent becomes equal, then add the mantissa.
21.	What is Floating point overflow exception?	When the exponent is too large to be stored. It is denoted by exponent all 1’s. Such a number is also called NaN.
22.	What is Floating point overflow exception?	When the exponent is too small to be stored. It is denoted by exponent all 0’s. Such a number is also called a DeNormal Number.
23.	How do you represent 0 and 1 in Single precision format? (any number can be asked)	0: $S = 0 \mid E = 0 \mid M = 0$ 1: $S = 0 \mid E = (127)_2 \mid M = 0$
24.	What is the difference between a subroutine (procedure) and a macro?	When we CALL a subroutine, the program control shifts from the main program to the subroutine, and after execution, returns to the next instruction after CALL. When we invoke a macro, the macro code is pasted into our program at the location where the macro was invoked.
25.	What is an instruction format?	Label: Opcode, Operands; Comment **Label |** Location Identifier (optional) **Opcode |** Binary code of the operation (compulsory) **Operands |** Values taking part in the operation (optional) **Comment |** Purpose of the instruction (optional)
26.	What are the types of instructions?	**Data Transfer** E.g.: MOV, LOAD, STORE etc. **Data Processing** E.g.: ADD, SUB, AND etc. **Control instructions**

		E.g.: JMP, CALL etc.
27.	What is instruction level pipelining?	Overlapping different stages of an instruction is called pipelining. E.g.:: Fetch, Decode, Execute, Store etc.
28.	Give examples of pipelined processors?	**8086 | 2 stage** Fetch, Exec. **80386 | 3 stage** Fetch, Decode, Exec. **Pentium | 5 stage** Fetch, Decode, Addr Gen, Exec, Store
29.	What are Pipelining Hazards?	Data Hazard Structural Hazard Control Hazard
30.	What do you mean by addressing modes. List some with examples?	It is the manner in which an operand is given in the instruction. Most popular ones are… Immediate | Register | Direct | Indirect | Implied
31.	What are the micro-operations required for fetching?	T1: MAR ← PC T2: MDR ← instruction from memory T3: IR ← MDR PC ← PC + 1
32.	What is an instruction cycle, machine cycle and T-state?	**Instruction cycle** |Total process of fetching, decoding and executing an instruction. **Machine cycle** | One complete operation of the system bus. **T-state** | One clock cycle. Instruction cycle has several machine cycles. Machine cycles have several Tstates.
33.	Give characteristics of a RISC processor?	**RISC |** Reduced Instruction Set Computers

		• Fixed sized instructions • Mainly Register based operations • Fewer addressing modes • Smaller instruction set • Simpler instructions • Prefer Hardwired Control Units • Suited for µControllers
34.	Give characteristics of a CISC processor?	**CISC |** Complex Instruction Set Computers • Variable sized instructions • Mainly Memory based operations • More addressing modes • Bigger instruction set • Complex instructions • Prefer µ-Programmed Control Units • Suited for µProcessors
35.	What are the two main types of Control Units?	**Hardwired Control Unit** **Microprogrammed Control Unit**
36.	What is a Hardwired Control Unit?	It generates control signals using dedicated HARDWARE. Common types are 1) State Table method 2) Delay Element method 3) Sequence Counter method
37.	What is a Microprogrammed Control Unit?	It generates control signals for any instruction using its µinstructions. Typical modern Microprogrammed Control Units are made following Wilke’s Design.

38.	What are µPrograms?	A set of µinstructions responsible for generating control signals for a particular instruction is called its µ program. M programs are stored in the control memory.
39.	What are µ instruction sequencing techniques?	**Two main techniques** 1) Dual Address Field 2) Single Address Field
40.	What are µ instruction formats?	**Two main formats** 1) Horizontal µ instructions 2) Vertical µ instructions
41.	Compare Hardwired Control Unit and Microprogrammed Control Unit?	**Hardwired Control Unit** Faster | No Control Memory needed | Suited for RISC **Hardwired Control Unit** Flexible | Supports Emulation | Control Memory needed | Suited for CISC
42.	List “typical” sizes of these memories?	Hard Disk | **1 TB** CD | **700 MB** DVD | **4.7 GB** Pen Drives | **2 GB – 128 GB** RAM | **4GB, 8GB, 16GB, 32GB** Cache | **1 MB – 4 MB**
43.	Compare SRAM and DRAM?	**Static RAM** Faster | Uses Flip-Flops | Expensive | used in Cache **Dynamic RAM** Slower | Uses Capacitors | Cheaper | used in main memory

44.	What are advanced DRAMs?	**RDRAM |** Rambus DRAM **SDRAM |** Synchronous DRAM
45.	What are the types of ROM?	**ROM |** Read Only Memory **PROM |** Programmable ROM **EPROM |** Erasable PROM **EEPROM |** Electrically EPROM **Flash ROM |** Electrically EPROM Most Popular in modern systems
46.	Where does your phone store images, songs etc?	In secondary storage implemented using Flash ROM
47.	What does primary memory consist of?	**RAM and ROM.** RAM for operations ROM mainly for the BIOS
48.	List important Page replacement policies?	FIFO, LRU, LFU, OPT
49.	What are dirty pages?	These are pages modified in the main memory.
50.	What is TLB?	**TLB |** Translation Look-aside Buffer Contains the most recently used entries of the page table to make page translation faster.
51.	What is a page fault?	**Page Fault** When the desired page is not present in the main memory it is called a miss or a page fault.
52.	Most memory systems contain both a translation lookaside buffer (TLB) used by the virtual memory system, and a cache memory. Both the cache and the TLB have similar functions: they store main memory data in a faster “cache-like” memory that	Although the TLB and the cache both hold copies of main memory data, they are used in different parts of the memory subsystem, and accesses to them have different characteristics. Each entry in the TLB contains a pair of addresses: a virtual page number and a page frame address. Because individual TLB entries translate a large number of virtual addresses (a full page each), relatively few of them are active at any time. So a TLB tends to be

	can be accessed at nearly processor speeds. The question is why do these two separate mechanisms exist when they have such a similar purpose? Why have both a cache and a TLB? Why not just a single “supercache” that would hold both regular memory data as well as frequently used page table entries?	fairly small, and the hardware to look up a virtual page number is fully associative. The regular cache, on the other hand, has much more data per row to exploit locality of instruction and data memory references. It also tends to be much larger than a TLB to get a reasonably large hit rate, so it is not practical to use fully associative addressing. Instead, most caches tend to be 2-way or 4-way set associative. A TLB miss and a cache miss are also handled in quite different ways. A TLB miss requires walking the page table; a cache miss does not use a secondary data structure. A final difference is that the TLB and the cache are used in different parts of a memory access. A TLB uses the virtual address to locate data, while a cache normally uses the physical (translated) address as the index and tag. Because of these differences it makes sense to use specialized hardware optimized for each function rather than having a single “super cache”.
53.	What are interleaving techniques?	Lower order interleaving and higher order interleaving.
54.	What is cache consistency?	When data in cache memory and main memory has same value, the cache is consistent. Main policies: Write Through and Write back.
55.	What are cache layouts?	Look Through Cache and Look Aside Cache.
56.	What are cache mapping techniques? Adv and Disadv	**Associative Mapping** **Direct Mapping** **Set Associative Mapping**
57.	What are the main types of multiprocessor systems?	**Closely Coupled and Loosely Coupled** **systems.**
58.	What are Bus Arbitration techniques? Adv and Disadv	Daisy Chaining, Polling and Independent requests.
59.	What are I/O data transfer techniques?	Polling/ Programmed I/O Interrupt driven I/O

		DMA I/O Processors
60.	What are I/O mapping techniques?	Memory mapped I/O and I/O mapped I/O (also called Isolated I/O)
61.	Give characteristics of interrupts?	Software, hardware, vectored, nonvectored, edge triggered, level triggered, Maskable, non Maskable.
62.	What are some popular DMA transfer techniques?	Block Transfer (Burst Mode) Cycle Stealing (Single Byte Transfers) Demand Transfer Hidden Mode (Transparent Mode)
63.	Name a popular I/O Processor?	8089 I/O Processor is used with 8086 µP.
64.	Give Flynn's Classification?	It classifies multiprocessor systems into 4 different categories, based on the instruction stream and the data streams. 1) **SISD |** Single Instruction Single Data 2) **SIMD |** Single Instruction Multiple Data 3) **MISD |** Multiple Instruction Single Data 4) **MIMD |** Multiple Instruction Multiple Data
65.	What is serial and parallel communication?	**Serial Communication** One bit at a time | Slower | Cheaper | Long Distance **Parallel Communication** Multiple bits at a time | Faster | Costlier | Short Distance
66.	What is VLIW? Superscalar architecture.	Very long instruction word (VLIW) describes a computer processing architecture in which a language compiler or pre-processor breaks program instruction down into basic operations that can be performed by the processor in parallel (that is, at the same time). These operations are put

		into a very long instruction word which the processor can then take apart without further analysis, handing each operation to an appropriate functional unit. VLIW is sometimes viewed as the next step beyond the reduced instruction set computing (RISC) architecture, which also works with a limited set of relatively basic instructions and can usually execute more than one instruction at a time (a characteristic referred to as *superscalar*). The main advantage of VLIW processors is that complexity is moved from the hardware to the software, which means that the hardware can be smaller, cheaper, and require less power to operate. The challenge is to design a compiler or pre-processor that is intelligent enough to decide how to build the very long instruction words.
67.	Module Test 1 (Amdahl's law), Test 2 and Test 3 Questions (Problems on Cache)	
68.	Refer this link for additional questions https://mucomputergraphics.wordpress.com/2017/05/04/coa-viva-question/	