**CS6303 COMPUTER ARCHITECTURE II YEAR / 4th SEMESTER, CSE**

**Faculty Name:**

**DEVI.P.P**

**Assistant Professor, CSE Department**

**Aalim Muhammed Salegh College of Engineering**

**UNIT-I OVERVIEW & INSTRUCTIONS**

**1. What are the five classic components of a computer?**

 The five classic components of a computer are input, output, memory, datapath, and control, with the last two sometimes combined and called the processor.

**2. Define –  Response Time**

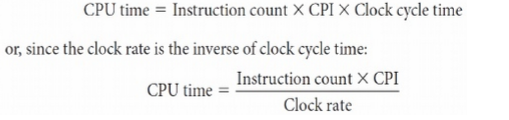
 Response time is also called execution time. The total time required for the computer to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, CPU execution time, and so on is called response time.

**3.Define  –  Throughput**

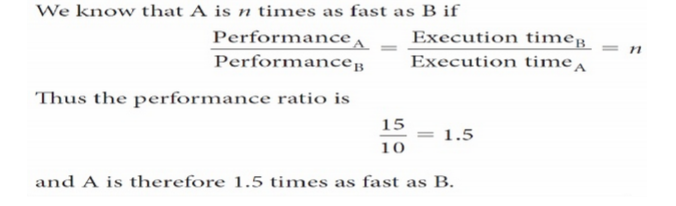
 Throughput or bandwidth is the total amount of work done in a given time.

**4. Write the CPU performance equation.**

 The Classic CPU Performance Equation in terms of instruction count(the number of instructions executed by the program), CPI, and clock cycle time:



**5. If computer A runs a program in 10 seconds, and computer B runs the same program in 15 seconds, how much faster is A over B.10.**

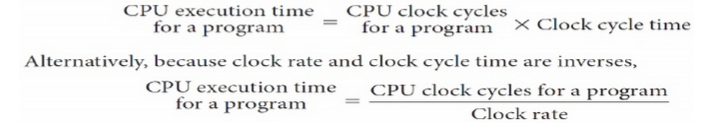


**6. What are the basic components of performance?**

The basic components of performance and how each is measured are:

|  |  |
| --- | --- |
| **Components of Performance** | **Units of measure** |
| CPU execution time for a program | Seconds for the program |
| Instruction count | Instruction executed for the program |
| Clock cycles per instruction(CPI) | Average number of clock cycles per instruction |
| Clock cycle time | Seconds per clock cycle |

**7. Write the formula for CPU execution time for a program. [Dec 2015]**

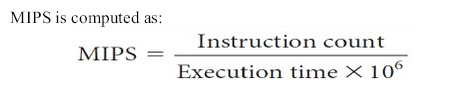
****

**8. Write the formula for CPU clock cycles required for a program.**

****

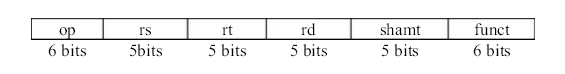
**9. Define  –  MIPS**

Million Instructions Per Second (MIPS) is a measurement of program execution speed based on the number of millions of instructions. MIPS is computed as:

****

**10. What are the fields in an MIPS instruction?**

**MIPS fields are**

****

Where,

op: Basic operation of the instruction, traditionally called the opcode.

rs: The first register source operand.

rt: The second register source operand.

rd: The register destination operand. It gets the result of the operation.

shamt: Shift amount.

funct: Function.

**11. Write an example for immediate operand.**

The quick add instruction with one constant operand is called add immediate or add i. To add 4 to register $s3,

we just write

addi  $s3,$s3,4  # $s3 = $s3 + 4

**12. Define  –  Stored Program Concepts**

 Today’s computers are built on two key principles:

 1. Instructions are represented as numbers.

2. Programs are stored in memory to be read or written, just like data.

These principles lead to the stored-program concept. Treating instructions in the same way as data greatly simplifies both the memory hardware and the software of computer systems.

**13. How to representing instruction in the computer system [May 2016]**

* Instructions are kept in computers as a series of high and low electric signals and represented as number.
* Each piece of an instruction can be considered as an individual number .
* Placing these number side by side forms the instruction.
* R- format: register format
* I-format: intermediate format
* J – format: Jump format

**14. Define  –  Addressing Modes**

The different ways in which the operands of an instruction are specified are called as addressing modes.

The MIPS addressing modes are the following:

1. Register addressing mode
2. Immediate addressing mode.
3. Base or displacement addressing mode
4. Pc-relative addressing mode
5. Pseudo- direct addressing mode
6. In direct addressing mode
7. Auto increment addressing mode
8. 8.Auto decrement addressing mode

**15.Define Register mode and Absolute Mode with examples.**

**Register mode:**

The operand is the contents of the processor register. The name (address) of the register is given in the instruction.

**Absolute Mode(Direct Mode):**

* The operand is in new location. The address of this location is given explicitly in the instruction.

Eg: MOVE LOC,R2

The above instruction uses the register and absolute mode.

The processor register is the temporary storage where the data in the register are accessed using register mode.

The absolute mode can represent global variables in the program.

|  |  |  |
| --- | --- | --- |
| Mode | Assembler Syntax | Addressing Function |
| Register mode | Ri | EA=Ri |
| Absolute mode | LOC | EA=LOC |

Where **EA**-Effective Address

**16.What is a Immediate addressing Mode?**

The operand is given explicitly in the instruction.

Eg: Move 200 immediate ,R0

* It places the value 200 in the register R0.The immediate mode used to specify the value of source operand.
* In assembly language, the immediate subscript is not appropriate so # symbol is used. It can be re-written as

Move #200,R0

Assembly Syntax: Addressing Function

Immediate #value Operand =value

**17.Define Indirect addressing Mode.**



The effective address of the operand is the contents of a register .We denote the indirection by the name of the register or new address given in the instruction.

**Fig: Indirect Mode**

Address of an operand (B) is stored into R1 register. If we want this operand, we can get it through register R1(indirection).

The register or new location that contains the address of an operand is called the **pointer.**

Mode Assembler Syntax Addressing Function

Indirect Ri , LOC EA=[Ri] or EA=[LOC]

**18.Define Index addressing Mode.**



* The effective address of an operand is generated by adding a constant value to the contents of a register.
* The constant value uses either special purpose or general purpose register. We indicate the index mode symbolically as,

**X(Ri)**

Where **X** – denotes the constant value contained in the instruction

**Ri** –It is the name of the register involved.

The Effective Address of the operand is,

EA=X + [Ri]

* The index register R1 contains the address of a new location and the value of X defines an offset(also called a displacement).

Eg: Add 20(R1) , R2 (or) EA=>1000+20=1020

|  |  |  |
| --- | --- | --- |
| Index Mode | Assembler Syntax | Addressing Function |
|  |  |  |
| Index | X(Ri) | EA=[Ri]+X |
| Base with Index | (Ri,Rj) | EA=[Ri]+[Rj] |
| Base with Index and offset | X(Ri,Rj) | EA=[Ri]+[Rj] +X |

**19. What is a Relative Addressing mode? [Dec 2015]**

It is same as index mode. The difference is, instead of general purpose register, here we can use program counter(PC).

**Relative Mode:**



* The Effective Address is determined by the Index mode using the PC in place of the general purpose register (gpr).
* This mode can be used to access the data operand. But its most common use is to specify the target address in branch instruction. Eg. Branch>0 Loop
* It causes the program execution to go to the branch target location. It is identified by the name loop if the branch condition is satisfied.

|  |  |  |
| --- | --- | --- |
| Mode | Assembler Syntax | Addressing Function |
| Relative | X(PC) | EA=[PC]+X |

**20.Define Auto-increment addressing mode. [May 2016]**



* The Effective Address of the operand is the contents of a register in the instruction.
* After accessing the operand, the contents of this register is automatically incremented to point to the next item in the list.

|  |  |  |
| --- | --- | --- |
| Mode | Assembler syntax | Addressing Function |
| Auto-increment | (Ri)+ | EA=[Ri]; Increment Ri |

**21.Define Auto-decrement addressing mode. [May 2016]**

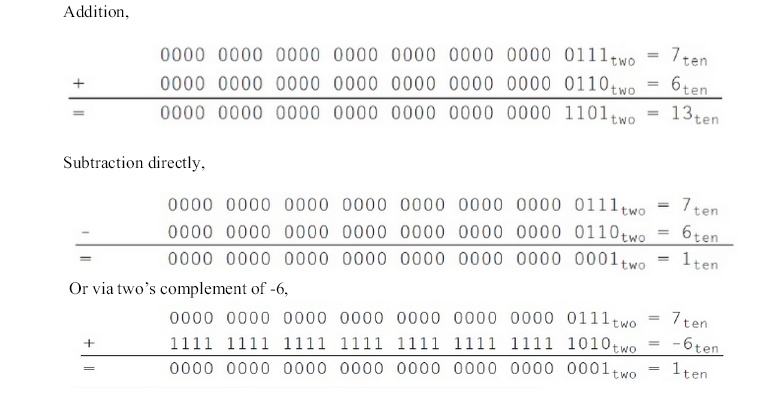


* The Effective Address of the operand is the contents of a register in the instruction.
* After accessing the operand, the contents of this register is automatically decremented to point to the next item in the list.

|  |  |  |
| --- | --- | --- |
| Mode | Assembler Syntax | Addressing Function |
| Auto-decrement | -(Ri) | EA=[Ri]; Decrement Ri |

**UNIT-II ARITHMETIC OPERATIONS**

1. **Add 610 to 710 in binary and Subtract 610 from 710 in binary.**

****

**2. What are the floating point instructions in MIPS?**

MIPS supports the IEEE 754 single precision and double precision formats with these instructions:

* Floating-point addition
* Floating-point subtraction
* Floating-point multiplication
* Floating-point division
* Floating-point comparison
* Floating-point branch

**3. Define Guard and Round**

Guard is the first of two extra bits kept on the right during intermediate calculations of floating point numbers. It is used to improve rounding accuracy.

Round is a method to make the intermediate floating-point result fit the floating-point format; the goal is typically to find the nearest number that can be represented in the format. IEEE 754, therefore, always keeps two extra bits on the right during intermediate additions, called guard and round, respectively.

**4. Define ULP**

Units in the Last Place is defined as the number of bits in error in the least significant bits of the significant between the actual number and the number that can be represented.

**5. What is meant by sub-word parallelism? [May 2015/May 2016]**

Given that the parallelism occurs within a wide word, the extensions are classified as sub-word parallelism. It is also classified under the more general name of data level parallelism. They have been also called vector or SIMD, for single instruction, multiple data . The rising popularity of multimedia applications led to arithmetic instructions that support narrower operations that can easily operate in parallel.

For example, ARM added more than 100 instructions in the NEON multimedia instruction extension to support sub-word parallelism, which can be used either withARMv7 or ARMv8.

**6. What is meant by sticky bit?**

Sticky bit is a bit used in rounding in addition to guard and round that is set when ever there are nonzero bits to the right of the round bit. This sticky bit allows the computer to see the difference between 0.50 … 00 ten and .... 01 ten when rounding

**7. What are the steps in the floating-point addition?**

The steps in the floating-point addition are

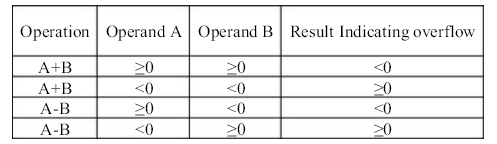
1. Align the decimal point of the number that has the smaller exponent.
   1. Addition of the significands.
   2. Normalize the sum.
   3. Round the result.

**8. Write the IEEE 754 floating point format.**

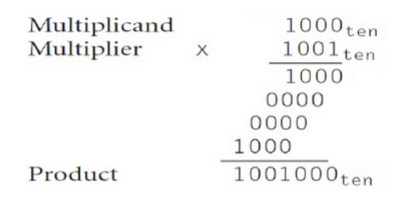
The IEEE 754 standard floating point representation is almost always an approximation of the real number



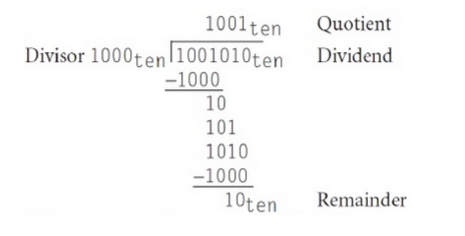
**9.Write the overflow conditions for addition and subtraction [May 2015/Dec 2015]**



**10. Multiply 100010 \* 100110.**



**11.** **Divide 1,001,010 ten  by 1000 ten**



**12. Define Little Endian arrangement? [Dec 2014]**

When lower byte addresses are used for the less significant bytes (rightmost bytes) of the word.

Draw the diagram in notes

**13.Define ALU [May 2016]**

ALU stands for Arithmetic Logic Unit, ALU is responsible for performing arithmetic operations such as add, subtract, division and multiplication and logical operation such as AND, OR, Inverting etc.

Arithmetic operation to be performed is based on data type.

**Two basic data types:**

1. Fixed point numbers

2. Floating point numbers

**14.State the representation of double precision of floating point number. [Dec 2015]**

* The IEEE-754 double precision occupies a two 32 bit words.

63 62 52 51 0

|  |  |  |
| --- | --- | --- |
| S | E’ | M |

S : sign of number. 0- signifies +ve, 1- signifies –ve

E’ : 11 bit signed exponent in excess-1023 representation.

M : 52 bit mantissa fraction

**15.what are overflow and underflow condition [Dec 2015]**

**Overflow:**

* A situation in which a positive exponent becomes too large to fit in the exponent field.
* In single precision , if the number requires an exponent greater than +127
* In double precision , if the number requires an exponent greater than +1023

**underflow :**

* A situation in which a negative exponent becomes too large to fi t in the exponent

field.

* In single precision , if the number requires an exponent less than -126
* In double precision , if the number requires an exponent less than -1022

**16. State the principle of operation of a carry look-ahead adder.**

The input carry needed by a stage is directly computed from carry signals obtained from all the preceding stages i-1,i-2,…..0, rather than waiting for normal carries to supply slowly from stage to stage. An adder that uses this principle is called carry look-ahead adder.

**17.What are the main features of Booth’s algorithm?**

* It handles both positive and negative multipliers uniformly.
* It achieves some efficiency in the number of addition required when the multiplier has a few large blocks of 1s.

**18.How can we speed up the multiplication process?**

There are two techniques to speed up the multiplication process:

* + The first technique guarantees that the maximum number of summands that must be added is n/2 for n-bit operands.
  + The second technique reduces the time needed to add the summands.

**19.What is bit pair recoding? Give an example.**

* Bit pair recoding halves the maximum number of summands. Group the Booth-recoded multiplier bits in pairs and observe the following:
* The pair (+1 -1) is equivalent to the pair (0 +1). That is instead of adding -1 times the multiplicand m at shift position i to +1 ( M at position i+1, the same result is obtained by adding +1 ( M at position i.

Eg: 11010 – Bit Pair recoding value is 0 -1 -2

**20.What is the advantage of using Booth algorithm?**

* It handles both positive and negative multiplier uniformly.
* It achieves efficiency in the number of additions required when the multiplier has a few large blocks of 1’s.
* The speed gained by skipping 1’s depends on the data.

**21.Write the algorithm for restoring division.**

Do the following for n times:

* Shift A and Q left one binary position.
* Subtract M and A and place the answer back in A.
* If the sign of A is 1, set q0 to 0 and add M back to A. Where A- Accumulator, M- Divisor, Q- Dividend.

**22.Write the algorithm for non restoring division.**

Do the following for n times:

Step 1: Do the following for n times:

* If the sign of A is 0 , shift A and Q left one bit position and subtract M from A; otherwise , shift A and Q left and add M to A.
* Now, if the sign of A is 0,set q0 to 1;otherwise , set q0 to0.

Step 2: if the sign of A is 1, add M to A.

**23.When can you say that a number is normalized?**

When the decimal point is placed to the right of the first (nonzero) significant digit, the number is said to be normalized.

**24. Explain about the special values in floating point numbers.**

The end values 0 to 255 of the excess-127 exponent E are used to represent special values such as:

When E= 0 and the mantissa fraction M is zero the value exact 0 is represented. When E= 255 and M=0, the value is represented.

When E= 0 and M = 0 , denormal values are represented.

When E= 2555 and M =0, the value represented is called Not a number.

**25.What is the purpose of guard bits used in floating point arithmetic**

Although the mantissa of initial operands are limited to 24 bits, it is important to retain extra bits, called as guard bits.

**26.What are the ways to truncate the guard bits?**

There are several ways to truncate the guard bits:

* 1. **Chopping**
  2. **Von Neumann rounding**
  3. **Rounding**

**27.Define carry save addition(CSA) process.**

Instead of letting the carries ripple along the rows, they can be saved and introduced into the next roe at the correct weighted position. Delay in CSA is less than delay through the ripple carry adder.

**28.What are generate and propagate function?**

The generate function is given by Gi=xiyi and

The propagate function is given as

Pi=xi+yi.

**29. What are the difficulties faced when we use floating point arithmetic?**

**Mantissa overflow**: The addition of two mantissas of the same sign may result in a carryout of themost significant bit

**Mantissa underflow**: In the process of aligning mantissas ,digits may flow off the right end of themantissa.

**Exponent overflow**: Exponent overflow occurs when a positive exponent exceeds the maximumpossible value.

**Exponent underflow**: It occurs when a negative exponent exceeds the maximum possibleexponent value.

**UNIT-III PROCESSOR AND CONTROL UNIT**

**1. What is meant by data path element?**

A data path element is a unit used to operate on or hold data within a processor. In the MIPS implementation, the data path elements include the instruction and data memories, the register file, the ALU, and adders.

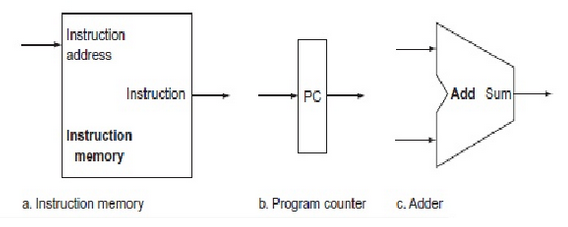
**2. What is the use of PC register?**

Program Counter (PC) is the register containing the address of the instruction in the program being executed.

**3. What is meant by register file?**

The processor’s 32 general purpose registers are stored in a structure called a register file. A register file is a collection of registers in which any register can be read or written by specifying the number of the register in the file. The register file contains the register state of the computer.

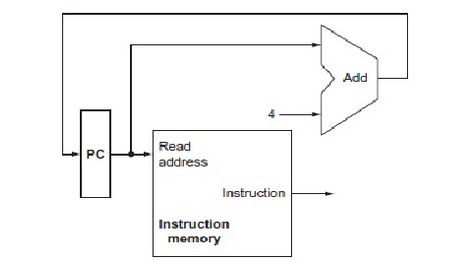
**4. What are the two state elements needed to store and access an instruction?**



Two state elements are needed to store and access instructions, and an adder is needed to compute the next instruction address. The state elements are the instruction memory and the program counter.

**5. Draw the diagram of portion of data path used for fetching instruction.**

A portion of the data path is used for fetching instructions and incrementing the program counter. The fetched instruction is used by other parts of the data path



**6. Define Sign Extend**

Sign-extend is used to increase the size of a data item by replicating the high-order sign bit of the original data item in the high order bits of the larger, destination data item.

**7. What is meant by branch target address?**

Branch target address is the address specified in a branch, which becomes the new program counter (PC) if the branch is taken. In the MIPS architecture the branch target is given by the sum of the off set field of the instruction and the address of the instruction following the branch.

**8. Differentiate branch taken from branch not taken.**

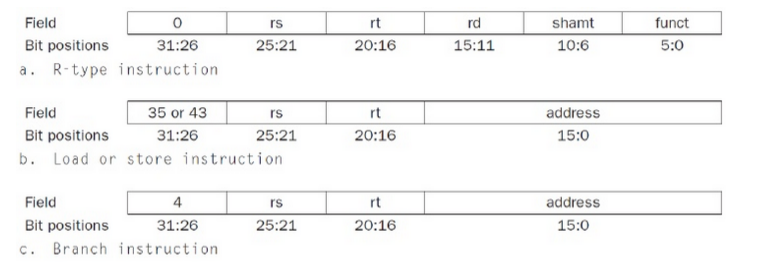
Branch taken is a branch where the branch condition is satisfied and the program counter (PC) becomes the branch target. All unconditional jumps are taken branches. Branch not taken or (untaken branch) is a branch where the branch condition is false and the program counter (PC) becomes the address of the instruction that sequentially follows the branch.

**9. What is meant by delayed branch?**

Delayed branch is a type of branch where the instruction immediately following the branch is always executed, independent of whether the branch condition is true or false

**10. What are the three instruction classes and their instruction formats?**

The three instruction classes (R-type, load and store, and branch) use two different instruction formats.



**11. Write the instruction format for the jump instruction.**

The destination address for a jump instruction is formed by concatenating the upper 4 bits of the current PC + 4 to the 26-bit address field in the jump instruction and adding 00 as the 2 low-order bits.



**12. What is meant by pipelining? [May 2016]**

Pipelining is an implementation technique in which multiple instructions are overlapped in execution. Pipelining improves performance by increasing instruction throughput, as opposed to decreasing the execution time of an individual instruction.

**13. What is meant by forwarding?**

Forwarding, also called bypassing, is a method of resolving a data hazard by retrieving the missing data element from internal buffers rather than waiting for it to arrive from programmer visible registers or memory.

**14. What is pipeline stall?**

Pipeline stall, also called bubble, is a stall initiated in order to resolve a hazard. They can be seen elsewhere in the pipeline.

**15. What is meant by branch prediction?[May 2015/Dec 2015]**

Branch prediction is a method of resolving a branch hazard that assumes a given outcome for the branch and proceeds from that assumption rather than waiting to ascertain the actual outcome.

**16. What are the five steps in MIPS instruction execution?**

1. Fetch instruction from memory.

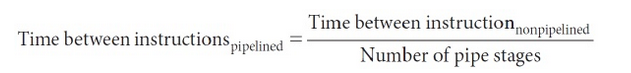
2. Read registers while decoding the instruction. The regular format of MIPS instructions allows reading and decoding to occur simultaneously.

3. Execute the operation or calculate an address.

4. Access an operand in data memory.

5. Write the result into a register.

**17. Write the formula for calculating time between instructions in a pipelined processor?**



**18. What are hazards? Write its types. [Dec 2015]**

 There are situations in pipelining when the next instruction cannot be executed in the following clock cycle. These events are called hazards, and there are three different types.

1. Structural Hazards

2. Data Hazards

3 . Control Hazards

**19. What are the 5 pipeline stages?**

The 5 stages of instruction execution in a pipelined processor are:

1. IF: Instruction fetch

2. ID: Instruction decode and register file read

3. EX: Execution or address calculation

4. MEM: Data memory access

5. WB: Write back

**20. What are exceptions and interrupts?[Dec 2014/ May 2016]**

Exception, also called interrupt, is an unscheduled event that disrupts program execution used to detect overflow. Eg. Arithmetic overflow, using an undefined instruction. Interrupt is an exception that comes from outside of the processor.Eg. I/O device request

**21. Define  –  Vectored Interrupts**

Vectored interrupt is an interrupt in that the address to which the control is transferred is determined by the cause of the exception

**UNIT-IV PARALLELISM**

**1. What is meant by ILP?[Dec 2015/May 2016]**

Pipelining exploits the potential parallelism among instructions. This parallelism is called instruction-level parallelism (ILP). There are two primary methods for increasing the potential amount of instruction-level parallelism.

1. Increasing the depth of the pipeline to overlap more instructions.

2. Multiple issue.

**2. What is multiple issue? Write any two approaches.**

Multiple issue is a scheme whereby multiple instructions are launched in one clock cycle. It is a method for increasing the potential amount of instruction-level parallelism. It is done by replicating the internal components of the computer so that it can launch multiple instructions in every pipeline stage.

The two approaches are:

1. Static multiple issue (at compile time)

2. Dynamic multiple issue (at run time)

**3. What is meant by speculation?[Dec 2014]**

One of the most important methods for finding and exploiting more ILP is speculation. It is an approach whereby the compiler or processor guesses the outcome of an instruction to remove it as dependence in executing other instructions. For example, we might speculate on the outcome of a branch, so that instructions after the branch could be executed earlier.

**4. Define Static Multiple Issue?**

Static multiple issue is an approach to implement a multiple-issue processor where many decisions are made by the compiler before execution.

**5. Define Issue Slots and Issue Packet?**

Issue slots are the positions from which instructions could be issued in a given clock cycle. By analogy, these correspond to positions at the starting blocks for a sprint. Issue packet is the set of instructions that issues together in one clock cycle; the packet may be determined statically by the compiler or dynamically by the processor.

**6. Define VLIW?**

Very Long Instruction Word (VLIW) is a style of instruction set architecture that launches many operations that are defined to be independent in a single wide instruction, typically with many separate opcode fields.

**7. Define Superscalar Processor?[Dec 2015]**

Superscalar is an advanced pipelining technique that enables the processor to execute more than one instruction per clock cycle by selecting them during execution. Dynamic multiple-issue processors are also known as superscalar processors, or simply superscalars.

**8. What is meant by loop unrolling?**

An important compiler technique to get more performance from loops is loop unrolling, where multiple copies of the loop body are made. After unrolling, there is more ILP available by overlapping instructions from different iterations.

**9. What is meant by anti-dependence? How is it removed?**

Anti-dependence is an ordering forced by the reuse of a name, typically a register, rather than by a true dependence that carries a value between two instructions. It is also called as name dependence. Register renaming is the technique used to remove anti-dependence in which the registers are renamed by the compiler or hardware.

**10. What is the use of reservation station and reorder buffer?**

Reservation station is a buffer within a functional unit that holds the operands and the operation. Reorder buffer is the buffer that holds results in a dynamically scheduled processor until it is safe to store the results to memory or a register

**11. Differentiate in-order execution from out-of-order execution.**

Out-of-order execution is a situation in pipelined execution when an instruction is blocked from executing does not cause the following instructions to wait. It preserves the data flow order of the program. In-order execution requires the instruction fetch and decode unit to issue instructions in order, which allows dependences to be tracked, and requires the commit unit to write results to registers and memory in program fetch order. This conservative mode is called in-order commit.

**12. What is meant by hardware multithreading?[Dec 2014]**

Hardware multithreading allows multiple threads to share the functional units of a single processor in an overlapping fashion to try to utilize the hardware resources efficiently. To permit this sharing, the processor must duplicate the independent state of each thread. It Increases the utilization of a processor

**13. What are the two main approaches to hardware multithreading?**

There are two main approaches to hardware multithreading. Fine-grained multithreading switches between threads on each instruction, resulting in interleaved execution of multiple threads. This interleaving is often done in a round-robin fashion, skipping any threads that are stalled at that clock cycle. Coarse-grained multithreading is an alternative to fine-grained multithreading. It switches threads only on costly stalls, such as last-level cache misses.

**14. What is SMT?**

Simultaneous Multithreading (SMT) is a variation on hardware multithreading that uses the resources of a multiple-issue, dynamically scheduled pipelined processor to exploit thread-level parallelism. It also exploits instruction level parallelism.

**15. Differentiate SMT from hardware multithreading.**

Since SMT relies on the existing dynamic mechanisms, it does not switch resources every cycle. Instead, SMT is always executing instructions from multiple threads, leaving it up to the hardware to associate instruction slots and renamed registers with their proper threads.

**16. What are the three multithreading options?**

The three multithreading options are:

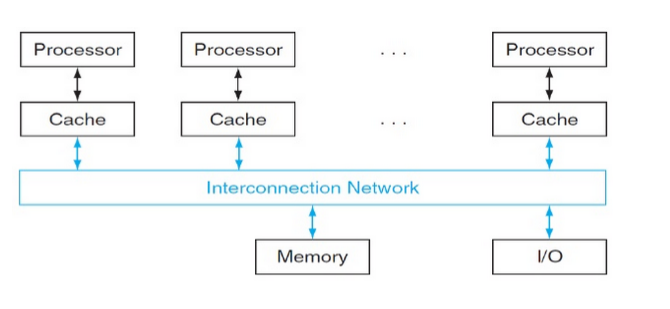
1. A superscalar with coarse-grained multithreading

2. A superscalar with fine-grained multithreading

3. A superscalar with simultaneous multithreading

**17. Define  –  SMP**

Shared memory multiprocessor (SMP) is one that offers the programmer a single physical address space across all processors – which is nearly always the case for multicore chips. Processors communicate through shared variables in memory, with all processors capable of accessing any memory location via loads and stores.



**18. Differentiate UMA from NUMA. [May 2015]**

Uniform memory access (UMA) is a multiprocessor in which latency to any word in main memory is about the same no matter which processor requests the access.Nonuniform memory access (NUMA) is a type of single address spacemultiprocessor in which some memory accesses are much faster than others depending on which processor asks for which word.

**19.What is flynn’s classification.[Dec 2014]**

* In 1966, Michael Flynn proposed a classification for computer architectures based on the number of instruction steams and data streams (Flynn’s Taxonomy).
* Flynn uses the stream concept for describing a machine's structure.
* A stream simply means a sequence of items (data or instructions).
* SISD: Single instruction single data
* SIMD: Single instruction multiple data
* MISD: Multiple instructions single data
* MIMD: Multiple instructions multiple data

**20.What is fine grained multithreading [May 2016]**

* Switches between threads on each instruction. causing the execution of multiple threads to be interleaved
* as the processor switches from one thread to the next, a thread that is currently stalled is skipped over
* CPU must be able to switch between threads at every clock cycle so that it needs extra hardware support

**21.Difference between strong and weak scaling[May 2015]**

1. **Strong scaling:** speedup achieved on a multiprocessor without increasing the size of the problem.
2. **Weak scaling:** speedup achieved on a multiprocessor while increasing the size of the problem proportionally to the increase In the number of processors.

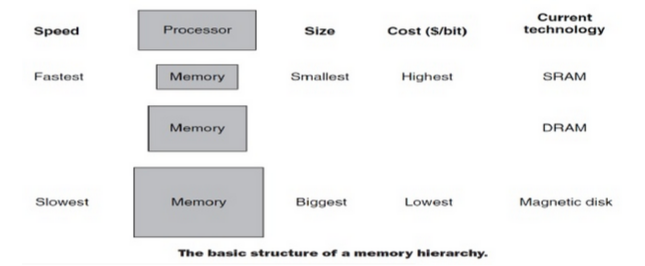
**UNIT-V MEMORY AND I/O SYSTEMS**

**1. What are the temporal and spatial localities of references?**

Temporal locality (locality in time): if an item is referenced, it will tend to be referenced again soon.

Spatial locality (locality in space): if an item is referenced, items whose addresses are close by will tend to be referenced soon.

**2.Write the structure of memory hierarchy?[may 2016/may2015/Dec 2015]**



**3. What are the various memory technologies?**

The various memory technologies are:

1.SRAM semiconductor memory

2. DRAM semiconductor memory

3. Flash semiconductor memory

4. Magnetic disk

**4. Differentiate SRAM from DRAM.**

SRAMs are simply integrated circuits that are memory arrays with a single access port that can provide either a read or a write. SRAMs have a fixed access time to anydatum.

SRAMs don’t need to refresh and so the access time is very close to the cycle

 time. SRAMs typically use six to eight transistors per bit to prevent the information from being disturbed when read. SRAM needs only minimal power to retain the charge instandby mode.

In a dynamic RAM (DRAM), the value kept in a cell is stored as a charge in a capacitor. A single transistor is then used to access this stored charge, either to read the value or to overwrite the charge stored there. Because DRAMs use only a single transistor per bit of storage, they are much denser and cheaper per bit than SRAM. As DRAMs store the charge on a capacitor, it cannot be kept indefinitely and must periodically be refreshed.

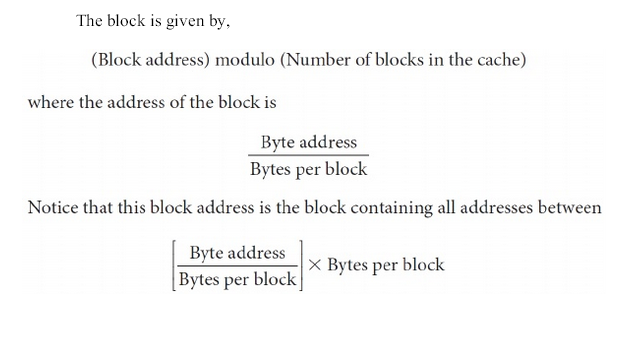
**6. Define − Rotational Latency**

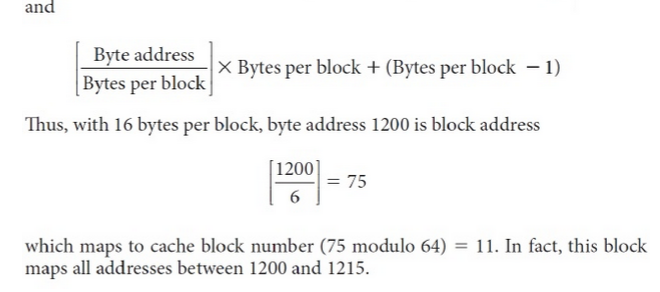
Rotational latency, also called rotational delay, is the time required for the desired sector of a disk to rotate under the read/write head, usually assumed to be half the rotation time.

**7. What is direct-mapped cache?**

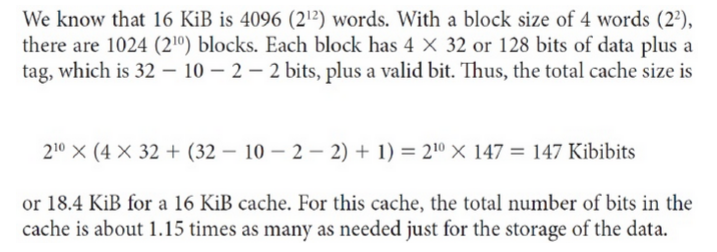
Direct-mapped cache is a cache structure in which each memory location is mapped to exactly one location in the cache. For example, almost all direct-mapped caches use this mapping to find a block, (Block address) modulo (Number of blocks in the cache)

**8. Consider a cache with 64 blocks and a block size of 16 bytes. To what block number does byte address 1200 map?**

****



**9. How many total bits are required for a direct-mapped cache with 16 KiB ofdata and 4-word blocks, assuming a 32-bit address?**



**10. What are the writing strategies in cache memory?**

Write-through is a scheme in which writes always update both the cache and the next lower level of the memory hierarchy, ensuring that data is always consistent between the two. Write-back is a scheme that handles writes by updating values only to the block in the cache, then writing the modified block to the lower level of the hierarchy when the block is replaced.

**11. What are the steps to be taken in an instruction cache miss?**

The steps to be taken on an instruction cache miss are

1. Send the original PC value (current PC  –  4) to the memory.

2. Instruct main memory to perform a read and wait for the memory to complete its access.

3. Write the cache entry, putting the data from memory in the data portion of

the entry, writing the upper bits of the address (from the ALU) into the tag field, and turning the valid bit on.

4. Restart the instruction execution at the first step, which will re fetch the instruction, this time finding it in the cache.

**12. Define  –  AMAT**

Average memory access time is the average time to access memory considering both hits and misses and the frequency of different accesses. It is equal to the following:



**13. What are the various block placement schemes in cache memory?**

Direct-mapped cache is a cache structure in which each memory location is mapped to exactly one location in the cache.

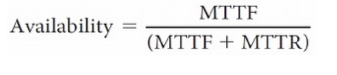
Fully associative cache is a cache structure in which a block can be placed in any location in the cache. Set-associative cache is a cache that has a fixed number of locations (at least two) where each block can be placed.

**14.Define  –  MTTF and AFR**

Reliability is a measure of the continuous service accomplishment or, equivalently, of the time to failure from a reference point. Hence, mean time to failure(MTTF) is a reliability measure. A related term is annual failure rate (AFR), which is just the percentage of devices that would be expected to fail in a year for a given MTTF.

**15. Define  –  Availability**

Availability is then a measure of service accomplishment with respect to the alternation between the two states of accomplishment and interruption. Availability is statistically quantified as



**16. What are the three ways to improve MTTF?**

The three ways to improve MTTF are:1. Fault avoidance: Preventing fault occurrence by construction.2. Fault tolerance: Using redundancy to allow the service to comply with the service specification despite faults occurring.3. Fault forecasting: Predicting the presence and creation of faults, allowing the component to be replaced before it fails.

**17. Define  –  TLB**

Translation-Look aside Buffer (TLB) is a cache that keeps track of recently used address mappings to try to avoid an access to the page table.

**18 . What is meant by virtual memory?**

Virtual memory is a technique that uses main memory as a “cache” for secondary storage. Two major motivations for virtual memory: to allow efficient and safe sharing of memory among multiple programs, and to remove the programming burdens of a small, limited amount of main memory.

**19. Differentiate physical address from logical address?**

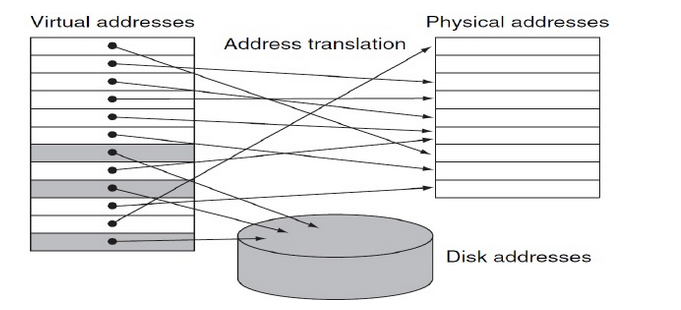
Physical address is an address in main memory. Logical address (or) virtual address is the CPU generated addresses that corresponds to a location in virtual space and is translated by address mapping to a physical address when memory is accessed.

**20. Define Page Fault?**

Page fault is an event that occurs when an accessed page is not present in main memory.

**21. What is meant by address mapping?**

Address translation also called address mapping is the process by which a virtual address is mapped to an address used to access memory.



**16 – Marks: UNIT-1 OVERVIEW AND INSTRUCTIONS**

**1. Explain the various components of computer System with neat diagram.**

[**Dec 2014/Dec 2015 /May 2016]**

* Input devices
* Output devices
* Memory
* CPU or processor
* Network

**2. Discuss in detail the various measures of performance of a computer.** [**Dec 2014]**

* Performance Evaluation
* Measuring performance
* CPU performance Equation
* SPEC CPU benchmark o Implementation of Amdahl’s law

**3. Define Addressing mode and explain the basic addressing modes with an example for each.** .[ **May 2015/Dec 2015/May 2016**]

1. Register addressing mode
2. Immediate addressing mode.
3. Base or displacement addressing mode
4. Pc-relative addressing mode
5. Pseudo- direct addressing mode
6. In direct addressing mode
7. Auto increment addressing mode
8. Auto decrement addressing mode

**4. Explain operations and operands of computer Hardware in detail.**

* operands of computer Hardware
* compiling a C assignments using registers
* memory operands

**5. Discuss the Logical operations and control operations of computer.**

* Logical operations o Bitwise operation AND o Bitwise operation OR
* Control operations o MIPS encoding for jump instruction
  + MIPS code
  + Compiling loop statements
  + Case/switch statements
  + Jump and link o Coprocessor instructions
  + Summarization of MIPS instruction format

**6.Discuss about the various techniques to represent instructions in a computer system.[May 2014]**

* Instruction format
* R- format: register format
* I-format: intermediate format
* J – format: Jump format
* Opcode and function code for each operation:
* Example

**UNIT-II ARITHMETIC OPERATIONS**

**1. Explain the Multiplication algorithm in detail with diagram and examples[Dec 2014/May 2015/Dec 2015/May 2016]**

* Signed multiplication
* Booth algorithm
* Booth algorithm for Signed multiplication
* Faster Multiplication
* Multiply in MIPS

**2. Discuss in detail about division algorithm in detail with diagram and examples.[Dec 2014/Dec 2015]**

* Signed division
* Faster division
* Division in MIPS

Step 1: Test divisor<dividend

Step 2: if divisor<dividend.

o Step 1: shift the divisor right by 1 bit

**3. Explain in detail about floating point representation**

* ∑ IEEE 754 standard
* ∑ Scientific notation in binary o Single precision floating point IEEE 754 standards
  + Double precision floating point IEEE 754 standard
* Normalization representation
* Sizes o Sign bit o Exponent
* Special values
* Denormalized representation
* Floating point under flow and over flow
* Guard and rounding.

**4. Explain in detail about floating point arithmetic operation.[May 2015/May 2016]**

* Floating point addition and subtraction
* Floating point addition o Procedure

o Example

* Floating point Multiplication

o Procedure

o Example

* Floating point in MIPS

**5. Explain in detail about basic concepts of ALU design**

* 1-bit ALU design
* Full adder
* 32-bit ALU design
* MIPS ALU design
* Arithmetic for multimedia

**6. Explain in detail about arithmetic operation**

* Boolean addition
* Boolean subtraction
* Overflow
* MIPS Overflow handling
* Ripple carry adder
* Fast adder circuit
* Carry look ahead adder

**7.Explain the Working of a Carry-Look Ahead adder. (NOV/DEC 2015)**

* Design Issues
* **Pi** **= Ai** **⊕ Bi**
* **Gi** **= Ai** **· Bi**
* Logical Circuit
* Carry vector equation for
* Design of Carry Lookahead Adders

**UNIT III PROCESSOR AND CONTROL UNIT**

**1. Explain the basic MIPS implementation of instruction set [Dec 2015¸**

The memory –reference instruction load word(lw) and store word(sw)

The arithmetic –logical instruction add,sub,and,or,and slt ¸

The instruction branch equal (beq) and jump (j)

Clocking methodology

**2. Explain in detail about building a data path [Dec 2014 ]**

* Data path elements
* Data path for branch instruction
* Creating a single Data path

**3. Explain in detail about control implementation scheme.[Dec 2014]**

* The ALU control
* Designing the control unit
* Operation of the Data path for an R type instruction
* Finalizing the control
* A multi cycle implementation

**4. What are control hazards? Explain the methods for dealing with the control hazards. [Dec 2014/May 2015/May2016]**

* Reducing the delay of branch
* Pipeline branch
* dynamic branch prediction

o 1-bit prediction scheme

o 2-bit prediction scheme

**5. Discuss the data hazards and forwarding in pipelining [Dec 2014/May2015/May2016]**

1a) EX/MEM. RegisterRd =ID/EX. register RS

1b) EX/MEM. RegisterRd =ID/EX. register RT

2a) MEM/WB. RegisterRd =ID/EX. register RS

2b) MEM/WB. RegisterRd =ID/EX. register RT

* Dependence detection
* Sub $1,$2,$3
* add $1,$2,$3
* or $1,$2,$3
* EX Hazards
* MEM Hazards

**6. How exceptions are handled in MIPS**

* Instruction fetch and memory stages o Memory protection violation
* Instruction decode stages o Undefined illegal opcode
* Execution stage o Arithmetic exception
* Write back stages

**7.What is pipelining ? Discuss about pipelined datapath and control [May 2016]**

**PIPELINING DATAPATH:**

* Five components correspond roughfly to the way thedatapath
* Five stages as they complete exectuion. Returning to
* Pipeline version of datapath
* Pipelined Datapath highlighting the pipeline registers
* Execution of Load /store instruction in a five stage pipeline

**PIPELINED CONTROL:**

* Control signals can be categorized by the pipeline stage that uses them
* Write and draw ALU tabular column from (control implementation scheme\_)

**UNIT IV PARALLELISM**

**1. Explain Instruction level parallelism [Dec 2014]**

* Dynamic, hardware intensive approach
* Static, complier intensive approach
* Loop level parallelism
* Data dependences and hazards
* Data dependences
* Control dependences
* Structure dependences
* ILP architecture

**2. Explain the difficulties faced by parallel processing programs [Dec 2014**

* First major challenges

–goods speedup

* Second major challenges

–remote access in parallel processing

**3. Explain in detail Flynn’s classification of parallel hardware [May 2015/Dec 2015/May 2016]**

* Introduction
* Flynn’s taxonomy
* Single instruction stream, single data stream (SISD)
* Single instruction stream, multiple data stream (SIMD)
* Multiple instruction streams, single data stream (MISD)
* Multiple instruction stream, multiple data stream(MIMD)

**4. Explain in detail hardware Multithreading [Dec 2014/May 2015/Dec 2015/May 2016]**

* Fine grained multithreading
* Advantages
* Disadvantages
* Coarse grained multithreading
* Advantages
* Disadvantages
* SMT
* A super scalar without multithreading
* A super scalar Fine grained multithreading
* A super scalar Coarse grained multithreading
* A super scalar SMT multithreading

**5. Explain Multicore processor [Dec 2014/May 2016]**

* ∑ Centralized shared memory architecture
* ∑ Classification based on communication models
* Distributed shared memory
* Message passing multiprocessor

**UNIT V MEMORY AND INPUT OUTPUT SYSTEMS**

**1. Explain in detail about memory hierarchy.**

The principle of locality, states that program access a relatively small portion p

of their address space at any instant of time.

* Temporal locality
* Spatial locality

**2. Explain Memory Technologies [May 2015]**

Main memory is the next level hierarchy It satisfies the demand of caches and serves as I/o interface

Types

* SRAM
* CMOS
* DRAM
* SYCHRONOUS DRAM
* DOUBLE DATA RATE DRAM
* RAMBUS MEMORY
* ROM
* PROM

cpu Register c a c h e I/O Devices Memory

* EPROM
* EEPROM
* FLASH MEMORY
* FLASH CARD
* FLASH DRIVE

**3. Explain about cache memory in detail [Dec 2014/May 2016]**

• Cache memory is a small amount of fast memory

* Placed between two levels of memory hierarchy
* To bridge the gap in access times

– Between processor and main memory (our focus)

– Between main memory and disk (disk cache)

How Cache Memory Works

* Prefetch data into cache before the processor needs it
* Need to predict processor future access requirements
* Not difficult owing to locality of reference

• Important terms

* Miss penalty
* Hit ratio ∗ Miss ratio = (1 – hit ratio)
* Hit time

**4. Explain about DMA Conroller [Dec 2014/Dec 2015/May 2016]**

**DMA channel:**

DMA channel is issued to transfer data between main memory and peripheral device in order to perform the transfer of data. The DMA controller access rs address and data buses. DMA with help of schematic diagram of controller on tile needs the dual circuits of and e to communicate with -CPU and I/O device. In addition, it nee s an address register; a word count register, and a set of, es The address register and address lines are used for communication with memory to word count register specifies the no. of word that - must be transfer may be done directly between the device and memory .

**5. Explain about 1/0 processor [Dec 2014]**

Input/Output processor/information processor: It is designed to handle input/ output processes of a device or the computer. This processor is separate from the main processor (CPU). I/O processor is similar to CPU but it controls input output operations only. The computer having I/O processor relieves CPU from Input/output operations only. CPU is the master processor of the computer and it instructs the I/O processor to handle the input output tasks. I/O processor cannot work independently and is controlled by the CPU.

The I/O processor is composed of commercially available TTL logic circuits that generate the micro instructions necessary to implement the I/O instructions. The I/O processor is fully synchronous with the system clock and main processor. it receives starting control from the main processor (CPU) whenever an input output instruction is read from memory. The I/O processor makes use of system buses after taking the permission from the CPU. It can instruction the I/O processor 1/0 processor responds to CPU by placing a status word at prescribed location to be checked out by the CPU later on CPU informs the 1/0 processor to find out the 1/0 program and ask 1/0 processor to transfer the data. I/O

**6. What are the advantages you got with virtual memory? [May 2015/Dec 2015]**

permit the user to construct program as though a large memory space were available, equal to totality auxiliary memory. Each address that is referenced by CPU goes through an address mapping from so called virtual address to physical address main memory.

There are following advantages we got with virtual memory:

1. Virtual memory helps in improving the processor utilization.

2. Memory allocation is also an important consideration in computer programming due to high cost of main memory.

3. The function of the memory management unit is therefore to translate virtual address to the physical address.

4. Virtual memory enables a program to execute on a computer with less main memory when it needs.

5.Virtual memory is generally implemented by demand paging concept In demand paging, pages are only loaded to main memory when they are required

6.Virtual memory that gives illusion to user that they have main memory equal to capacity of secondary stages media. The virtual memory is concept of implementation which is transferring the data from secondary stage media to main memory as and when necessary. The data replaced from main memory is written back to secondary storage according to predetermined replacement algorithm.