CS6303 COMPUTER ARCHITECTURE L T P C
3 0 0 3

OBJECTIVES:

- To make students understand the basic structure and operation of digital computer.
- To understand the hardware-software interface.
- To familiarize the students with arithmetic and logic unit and implementation of fixed point and floating-point arithmetic operations.
- To expose the students to the concept of pipelining.
- To familiarize the students with hierarchical memory system including cache memories and virtual memory.
- To expose the students with different ways of communicating with I/O devices and standard I/O interfaces.

UNIT I OVERVIEW & INSTRUCTIONS 9

Eight ideas – Components of a computer system – Technology – Performance – Power wall – Uniprocessors to multiprocessors; Instructions – operations and operands – representing instructions – Logical operations – control operations – Addressing and addressing modes.

UNIT II ARITHMETIC OPERATIONS 7

ALU - Addition and subtraction – Multiplication – Division – Floating Point operations – Subword parallelism.

UNIT III PROCESSOR AND CONTROL UNIT 11

Basic MIPS implementation – Building data path – Control Implementation scheme – Pipelining – Pipelined data path and control – Handling Data hazards & Control hazards – Exceptions.

UNIT IV PARALLELISM 9

Instruction-level-parallelism – Parallel processing challenges – Flynn's classification – Hardware multithreading – Multicore processors

UNIT V MEMORY AND I/O SYSTEMS

Memory hierarchy - Memory technologies – Cache basics – Measuring and improving cache performance - Virtual memory, TLBs - Input/output system, programmed I/O, DMA and interrupts, I/O processors.

OUTCOMES:

At the end of the course, the student should be able to:

- Design arithmetic and logic unit.
- Design and anlayse pipelined control units
- Evaluate performance of memory systems.
- Understand parallel processing architectures.

TEXT BOOK:

1. David A. Patterson and John L. Hennessey, “Computer organization and design’, Morgan Kauffman / Elsevier, Fifth edition, 2014.

REFERENCES:

1. V.Carl Hamacher, Zvonko G. Varanesic and Safat G. Zaky, “Computer Organisation“, VI th edition, Mc Graw-Hill Inc, 2012.
2. William Stallings “Computer Organization and Architecture” , Seventh Edition , Pearson Education, 2006.
3. Vincent P. Heuring, Harry F. Jordan, “Computer System Architecture”, Second Edition, Pearson Education, 2005.
4. Govindarajalu, “Computer Architecture and Organization, Design Principles and Applications", first edition, Tata McGraw Hill, New Delhi, 2005.
5. John P. Hayes, “Computer Architecture and Organization”, Third Edition, Tata Mc Graw Hill, 1998.
6. http://nptel.ac.in/.

PART-A-TWO MARK QUESTIONS AND ANSWERS

UNIT-I OVERVIEW & INSTRUCTIONS

1.Define Computer Architecture.

- It is concerned with the structure and behavior of the computer.
- It includes the information formats, the instruction set and techniques for addressing memory.

2.Define Computer Organization.

- It describes the function and design of the various units of digital computer that store and process information.
- It refers to the operational units and their interconnections that realize the architectural specifications**.**

3.What are the components of a computer.

- Input unit
- Memory unit
- Arithmetic and Logic Unit
- Output unit
- Control unit

4.Draw the block diagram of computer.

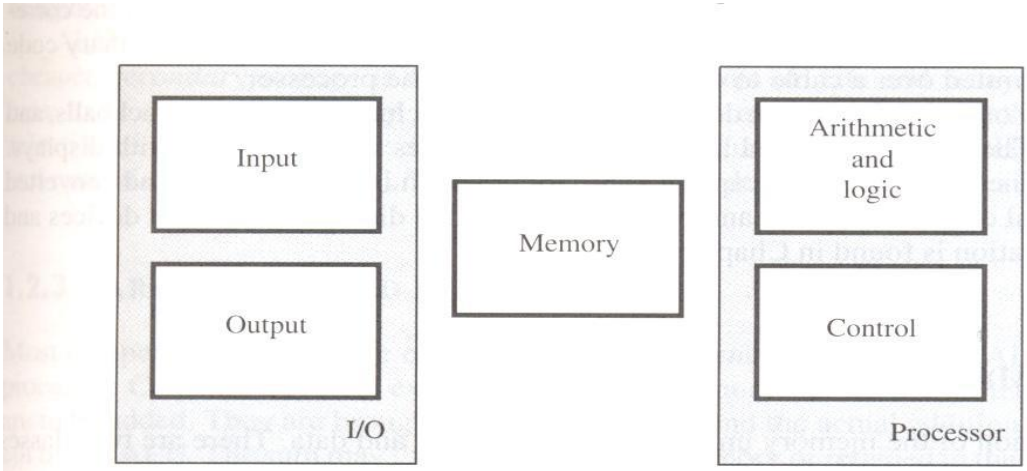


Figure 1.1 Basic functional units of a computer.

5. What is Execution time/Response time?

Response time 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.

6. What is CPU execution time, user CPU time and system CPU time?

CPU time : The actual time the CPU spends computing for a specific task.
user CPU time: The CPU time spent in a program itself.
system CPU time: The CPU time spent in the operating system performing tasks on behalf the program.

7.What is clock cycle and clock period?

clock cycle :The time for one clock period, usually of the processor clock, which runs at a constant rate.
clock period :The length of each clock cycle.

6. Define CPI

The term Clock Cycles Per Instruction Which is the average number of clock cycles each

instruction takes to execute, is often abbreviated as CPI.

7.State and explain the performance equation?

N denotes number of machine Instructions, Suppose that the average number of basic steps needed to execute one machine instruction is S, where each basic step is completed in one clock cycle. If the clock cycle rate is R cycles per second, the processor time is given by

$T = (N \times S) / R$ This is often referred to as the basic performance equation.

8. Define MIPS .

MIPS:One alternative to time as the metric is MIPS(Million Instruction Per Second)

MIPS=Instruction count/(Execution time x1000000).This MIPS measurement is also called Native MIPS to distinguish it from some alternative definitions of MIPS.

9.Define MIPS Rate:

The rate at which the instructions are executed at a given time.

10.Define Throughput and Throughput rate.

- Throughput -The total amount of work done in a given time.
- Throughput rate-The rate at which the total amount of work done at a given time.

11.What are the various types of operations required for instructions?

• Data transfers between the main memory and the CPU registers

• Arithmetic and logic operation on data

• Program sequencing and control

• I/O transfers

12. What is a Program?

A program is a set of instructions that specify the operations, operands and the sequence by which processing has to occur.

13. What is a Computer Instruction?

A Computer instruction is a binary code that specifies a sequence of micro operations for the computer.

14. What is a Instruction Code?

An instruction code is a group of bits that instruct the computer to perform a specific operation.

15. What is a Operation Code (Opcode)?

The operation code of an instruction is a group of bits that define operations as add, subtract, multiply, shift and complement etc.

16. Define Instruction Format.

Instructions are represented as numbers .Therefore, entire programs can be stored in memory to be read or written just like numbers(data).Thus simplifies software/Hardware of computer systems. Each instruction is encoded in binary called machine code.

17.What are the Most Common Fields Of An Instruction Format?

- An operation code field that specifies the operation to be performed.
- An address field that designates, a memory address or register.
- A mode field that specifies the way the operand or the effective address is determined

18.Explain the following the address instruction?

• Three-address instruction-it can be represented as

ADD A, B, C

Operands a,b are called source operand and c is called destination operand.

• Two-address instruction-it can be represented as

ADD A, B

• One address instruction-it can be represented as

LOAD A

ADD B

STORE C

19. What is the straight-line sequencing?

The CPU control circuitry automatically proceed to fetch and execute instruction, one at a time in the order of the increasing addresses. This is called straight line sequencing.

20.Wrie down the MIPS Assembly language notation for arithmetic operations.

Category	Instruction	Example	Meaning	Comments
Arithmetic	add	add $s1,$s2,$s3	$s1 = $s2 + $s3	Three register operands
	subtract	sub $s1,$s2,$s3	$s1 = $s2 – $s3	Three register operands
	add immediate	addi $s1,$s2,20	$s1 = $s2 + 20	Used to add constants

21.Wrie down the MIPS Assembly language notation for data transfer operations.

Category	Instruction	Example	Meaning	Comments
Data transfer	load word	lw $s1,20($s2)	$s1 = Memory[$s2 + 20]	Word from memory to register
	store word	sw $s1,20($s2)	Memory[$s2 + 20] = $s1	Word from register to memory
	load half	lh $s1,20($s2)	$s1 = Memory[$s2 + 20]	Halfword memory to register
	load half unsigned	lhu $s1,20($s2)	$s1 = Memory[$s2 + 20]	Halfword memory to register
	store half	sh $s1,20($s2)	Memory[$s2 + 20] = $s1	Halfword register to memory
	load byte	lb $s1,20($s2)	$s1 = Memory[$s2 + 20]	Byte from memory to register
	load byte unsigned	lbu $s1,20($s2)	$s1 = Memory[$s2 + 20]	Byte from memory to register
	store byte	sb $s1,20($s2)	Memory[$s2 + 20] = $s1	Byte from register to memory
	load linked word	ll $s1,20($s2)	$s1 = Memory[$s2 + 20]	Load word as 1st half of atomic swap
	store condition.	sc $s1,20($s2)	Memory[$s2+20]=$s1;$s1=0 or	Store word as 2nd half of atomic swap
	load upper immed.	lui $s1,20	$s1 = 20 * 2^{16}	Loads constant in upper 16 bits

22.Wrie down the MIPS Assembly language notation for Logical operations.

Category	Instruction	Example	Meaning	Comments
Logical	and	and $s1,$s2,$s3	$s1 = $s2 & $s3	Three reg. operands; bit-by-bit AND
	or	or $s1,$s2,$s3	$s1 = $s2 | $s3	Three reg. operands; bit-by-bit OR
	nor	nor $s1,$s2,$s3	$s1 = ~ ($s2 | $s3)	Three reg. operands; bit-by-bit NOR
	and immediate	andi $s1,$s2,20	$s1 = $s2 & 20	Bit-by-bit AND reg with constant
	or immediate	ori $s1,$s2,20	$s1 = $s2 | 20	Bit-by-bit OR reg with constant
	shift left logical	sll $s1,$s2,10	$s1 = $s2 << 10	Shift left by constant
	shift right logical	srl $s1,$s2,10	$s1 = $s2 >> 10	Shift right by constant

23.Wrie down the MIPS Assembly language notation for conditional branch operations.

Category	Instruction	Example	Meaning	Comments
Conditional branch	branch on equal	beq $s1,$s2,25	if ($s1 == $s2) go to PC + 4 + 100	Equal test; PC-relative branch
	branch on not equal	bne $s1,$s2,25	if ($s1!= $s2) go to PC + 4 + 100	Not equal test; PC-relative
	set on less than	slt $s1,$s2,$s3	if ($s2 < $s3) $s1 = 1; else $s1 = 0	Compare less than; for beq, bne
	set on less than unsigned	sltu $s1,$s2,$s3	if ($s2 < $s3) $s1 = 1; else $s1 = 0	Compare less than unsigned
	set less than immediate	slti $s1,$s2,20	if ($s2 < 20) $s1 = 1; else $s1 = 0	Compare less than constant
	set less than immediate unsigned	sltiu $s1,$s2,20	if ($s2 < 20) $s1 = 1; else $s1 = 0	Compare less than constant unsigned

24.Wrie down the MIPS Assembly language notation for Unconditional branch operations.

Category	Instruction	Example	Meaning	Comments
Unconditional jump	jump	j 2500	go to 10000	Jump to target address
	jump register	jr $ra	go to $ra	For switch, procedure return
	jump and link	jal 2500	$ra = PC + 4; go to 10000	For procedure call

25. What is Addressing Modes?

The different ways in which the location of an operand is specified in an instruction is called as Addressing mode.

26.What are the different types of addressing Modes?

- Immediate mode
- Register mode
- Absolute mode
- Indirect mode
- Index mode
- Base with index
- Base with index and offset
- Relative mode
- Auto-increment mode
- Auto-decrement mode

27.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

28.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

29.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]

30.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,

$$X(R_i)$$

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

R_i − 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).

To find operand,

- First go to Reg R1 (using address)-read the content from R1-1000

  Add the content 1000 with offset 20 get the result.

-
- 1000+20=1020

- Here the constant X refers to the new address and the contents of index register define the offset to the operand.
- The sum of two values is given explicitly in the instruction and the other is stored in register.

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

31.What is a Relative Addressing mode?

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 goto 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

32.Define Auto-increment addressing mode.

- 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

33.Define Auto-decrement addressing mode.

- 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-2 ARITHMETIC OPERATIONS

1. 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.

2. What are the main features of Booth's algorithm?

1) It handles both positive and negative multipliers uniformly.

2) It achieves some efficiency in the number of addition required when the multiplier has a few large blocks of 1s.

3. How can we speed up the multiplication process?(CSE Nov/Dec 2003)

There are two techniques to speed up the multiplication process:

1) The first technique guarantees that the maximum number of summands that must be added is n/2 for n-bit operands.

2) The second technique reduces the time needed to add the summands.

4. 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

5. What is the advantage of using Booth algorithm?

1) It handles both positive and negative multiplier uniformly.

2) It achieves efficiency in the number of additions required when the multiplier has

a few large blocks of 1's.

3) The speed gained by skipping 1's depends on the data.

6. Write the algorithm for restoring division.

Do the following for n times:

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

7. Write the algorithm for non restoring division.

Do the following for n times:

Step 1: Do the following for n times:

1) 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.

2) 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.

8. 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.

9. 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.

10. Write the Add/subtract rule for floating point numbers.

1) Choose the number with the smaller exponent and shift its mantissa right a number of steps equal to the difference in exponents.

2) Set the exponent of the result equal to the larger exponent.

3) Perform addition/subtraction on the mantissa and determine the sign of the result

4) Normalize the resulting value, if necessary.

11. Write the multiply rule for floating point numbers.

1) Add the exponent and subtract 127.

2) Multiply the mantissa and determine the sign of the result .

3) Normalize the resulting value , if necessary.

12. 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.

13. What are the ways to truncate the guard bits?

There are several ways to truncate the guard bits:1) Chooping 2) Von Neumann rounding 3) Rounding

14. 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.

15. What are generate and propagate function?

The generate function is given by

Gi=xiyi and

The propagate function is given as

Pi=xi+yi.

16. What is floating point numbers?

In some cases, the binary point is variable and is automatically adjusted as computation proceeds. In such case, the binary point is said to float and the numbers are called floating point numbers.

17. In floating point numbers when so you say that an underflow or overflow has occurred?

In single precision numbers when an exponent is less than -126 then we say that an underflow has occurred. In single precision numbers when an exponent is less than +127 then we say that an overflow has occurred.

18. 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 the most significant bit

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

Exponent overflow: Exponent overflow occurs when a positive exponent exceeds the maximum possible value.

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

19.In conforming to the IEEE standard mention any four situations under which a processor sets exception flag.

Underflow: If the number requires an exponent less than -126 or in a double precision, if the number requires an exponent less than -1022 to represent its normalized form the underflow occurs.

Overflow: In a single precision, if the number requires an exponent greater than -127 or in a double precision, if the number requires an exponent greater than +1023 to represent its normalized form the underflow occurs.

Divide by zero: It occurs when any number is divided by zero.

Invalid: It occurs if operations such as 0/0 are attempted.

20. Why floating point number is more difficult to represent and process than integer?(CSE May/June 2007)

An integer value requires only half the memory space as an equivalent.IEEEdouble-precision floatingpoint value. Applications that use only integer based arithmetic will therefore also have significantly smaller memory requirement

A floating-point operation usually runs hundreds of times slower than an equivalent integer based arithmetic operation.

21.Give the booth's recoding and bit-pair recoding of the computer.

1000111101000101(CSE May/June 2006)

Booth's recoding

1 0 0 0 1 1 1 1 0 1 0 0 0 1 0 1 0

-1 0 0 +1 0 0 0 -1 +1 -1 0 0 +1 -1 +1 -1

Bit-Pair recoding:

1 0 0 0 1 1 1 1 0 1 0 0 0 1 0 1 0

-2 +1 0 -1 +1 0 +1 1

22.Draw the full adder circuit and give the truth table (CSE May/June 2007)

Inputs			Outputs	
A	B	C	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

UNIT-3 PROCESSOR AND CONTROL UNIT

1.Define pipelining.

Pipelining is a technique of decomposing a sequential process into sub operations with each sub process being executed in a special dedicated segment that operates concurrently with all other segments.

2.Define parallel processing.

Parallel processing is a term used to denote a large class of techniques that are used to provide simultaneous data-processing tasks for the purpose of increasing the computational speed of a computer system. Instead of processing each instruction sequentially as in a conventional computer, a parallel processing system is able to perform concurrent data

processing to achieve faster execution time.

3.Define instruction pipeline.

The transfer of instructions through various stages of the CPU instruction cycle.,including fetch opcode, decode opcode, compute operand addresses. Fetch operands, execute Instructions and store results. This amounts to realizing most (or) all of the CPU in the form of multifunction pipeline called an instruction pipelining.

4. What are the steps required for a pipelinened processor to process the instruction?

- F Fetch: read the instruction from the memory
- D Decode: decode the instruction and fetch the source operand(s).
- E Execute: perform the operation specified by the instruction.
- W Write: store the result in the destination location

5. What are Hazards?

A hazard is also called as hurdle .The situation that prevents the next instruction in the instruction stream from executing during its designated Clock cycle. Stall is introduced by hazard. (Ideal stage)

6. State different types of hazards that can occur in pipeline.

The types of hazards that can occur in the pipelining were,

1. Data hazards.
2. Instruction hazards.
3. Structural hazards.

7. Define Data hazards

A data hazard is any condition in which either the source or the destination operands of an instruction are not available at the time expected in pipeline. As a result some operation has to be delayed, and the pipeline stalls.

8. Define Instruction hazards

The pipeline may be stalled because of a delay in the availability of an instruction. For example, this may be a result of miss in cache, requiring the instruction to be fetched from the main memory. Such hazards are called as Instruction hazards or Control hazards.

9.Define Structural hazards?

The structural hazards is the situation when two instructions require the use of a given hardware resource at the same time. The most common case in which this hazard may arise is access to memory.

10. What are the classification of data hazards?

Classification of data hazard: A pair of instructions can produce data hazard by referring reading or writing the same memory location. Assume that i is executed before J. So, the hazards can be classified as,

1. RAW hazard
2. WAW hazard
3. WAR hazard

11.Define RAW hazard : (read after write)

Instruction ‘j’ tries to read a source operand before instruction ‘i’ writes it.

12. Define WAW hazard :(write after write)

Instruction ‘j’ tries to write a source operand before instruction ‘i’ writes it.

13.Define WAR hazard :(write after read)

Instruction ‘j’ tries to write a source operand before instruction ‘i’ reads it.

14. How data hazard can be prevented in pipelining?

Data hazards in the instruction pipelining can prevented by the following techniques.

a)Operand Forwarding

b)Software Approach

15.How Compiler is used in Pipelining?

A compiler translates a high level language program into a sequence of machine instructions. To reduce N, we need to have suitable machine instruction set and a compiler that makes good use of it. An optimizing compiler takes advantages of various features of the target processor to reduce the product N*S, which is the total number of clock cycles needed to execute a program. The number of cycles is dependent not only on the choice of instruction, but also on the order in which they appear in the program. The compiler may rearrange program instruction to achieve better performance of course, such changes must not affect of the result of the computation.

16. How addressing modes affect the instruction pipelining?

Degradation of performance is an instruction pipeline may be due to address dependency where operand address cannot be calculated without available informatition needed by addressing mode for e.g. An instructions with register indirect mode cannot proceed to fetch the operand if the previous instructions is loading the address into the register. Hence operand access is delayed degrading the performance of pipeline.

17. What is locality of reference?

Many instruction in localized area of the program are executed repeatedly during some time period and the remainder of the program is accessed relatively infrequently .this is referred as locality of reference.

18. What is the need for reduced instruction chip?

- Relatively few instruction types and addressing modes.
- Fixed and easily decoded instruction formats.
- Fast single-cycle instruction execution.
- Hardwired rather than micro programmed control

19. Define memory access time?

The time that elapses between the initiation of an operation and completion of that operation ,for example ,the time between the READ and the MFC signals .This is Referred to as memory access time.

20. Define memory cycle time.

The minimum time delay required between the initiations of two successive memory operations, for example, the time between two successive READ operations.

21.Define Static Memories.

Memories that consist of circuits capable of retaining the state as long as power is applied are known as static memories.

22. List out Various branching technique used in micro program control unit?

a) Bit-Oring

b) Using Conditional Variable

c) Wide Branch Addressing

23. How the interrupt is handled during exception?

* CPU identifies source of interrupt

* CPU obtains memory address of interrupt handles

* PC and other CPU status information are saved

* Pc is loaded with address of interrupt handler and handling program to handle it.

24. List out the methods used to improve system performance.

The methods used to improve system performance are

1. Processor clock
2.Basic Performance Equation
3.Pipelining
4.Clock rate
5.Instruction set
6.Compiler

UNIT-4 PARALLELISM

1. What is Instruction Level Parallelism? (NOV/DEC 2011)
Pipelining is used to overlap the execution of instructions and improve performance. This potential overlap among instructions is called instruction level parallelism (ILP).

2. Explain various types of Dependences in ILP.

- Data Dependences
- Name Dependences
- Control Dependences

3. What is Multithreading?
Multithreading allows multiple threads to share the functional units of a single processor in an overlapping fashion. To permit this sharing, the processor must duplicate the independent state of each thread.

4. What are multiprocessors? Mention the categories of multiprocessors?
Multiprocessor are used to increase performance and improve availability. The different categories are SISD, SIMD, MIMD.

5. What are two main approaches to multithreading?

- Fine-grained multithreading
- Coarse-grained multithreading

•

6. What is the need to use multiprocessors?

1. Microprocessors as the fastest CPUs Collecting several much easier than redesigning 1
2. Complexity of current microprocessors
   Do we have enough ideas to sustain 1.5X/yr?
   Can we deliver such complexity on schedule?
3. Slow (but steady) improvement in parallel software (scientific apps, databases, OS)
4. Emergence of embedded and server markets driving microprocessors in addition to desktops Embedded functional parallelism, producer/consumer model
5. Server figure of merit is tasks per hour vs. latency.

7. Write the advantages of Multithreading.

If a thread gets a lot of cache misses, the other thread(s) can continue, taking advantage of the unused computing resources, which thus can lead to faster overall execution, as these resources would have been idle if only a single thread was executed. If a thread cannot use all the computing resources of the CPU (because instructions depend on each other's result), running another thread permits to not leave these idle.

If several threads work on the same set of data, they can actually share their cache, leading to better cache usage or synchronization on its values.

8. Write the disadvantages of Multithreading.

Multiple threads can interfere with each other when sharing hardware resources such as caches or translation lookaside buffers (TLBs). Execution times of a single-thread are not improved but can be degraded, even when only one thread is executing. This is due to slower frequencies and/or additional pipeline stages that arc necessary to accommodate thread-switching hardware. Hardware support for Multithreading is more visible to software, thus requiring more changes to both application programs and operating systems than Multi processing.

9. What is CMT?

Chip multiprocessors - also called multi-core microprocessors or CMPs for short - are now the only way to build high-performance microprocessors, for a variety of reasons. Large uniprocessors are no longer scaling in performance, because it is only possible to extract a limited amount of parallelism from a typical instruction stream using conventional superscalar instruction issue techniques. In addition, one cannot simply ratchet up the clock speed on today's processors, or the power dissipation will become prohibitive in all but water-cooled systems.

10. What is SMT?

Simultaneous multithreading, often abbreviated as SMT, is a technique for improving the overall efficiency of superscalar CPUs with hardware multithreading. SMT permits multiple independent threads of execution to better utilize the resources provided by modern processor architectures.

11. Write the advantages of CMP?

CMPs have several advantages over single processor solutions energy and silicon area efficiency

i. By Incorporating smaller less complex cores onto a single chip
ii. Dynamically switching between cores and powering down unused cores
iii. Increased throughput performance by exploiting parallelism
iv.Multiple computing resources can take better advantage of instruction, thread, and process level

12. What are the Disadvantages of SMT?

Simultaneous multithreading cannot improve performance if any of the shared resources are limiting bottlenecks for the performance. In fact, some applications run slower when simultaneous multithreading is enabled. Critics argue that it is a considerable burden to put on software developers that

they have to test whether simultaneous multithreading is good or bad for their application in various situations and insert extra logic to turn it off if it decreases performance.

13. What are the types of Multithreading?

- Block multi-threading
- Interleaved multi-threading

14. What Thread-level parallelism (TLP)?

- Explicit parallel programs already have TLP (inherent)
- Sequential programs that are hard to parallelize or ILP-limited can be speculatively parallelized in hardware.

15. List the major MIMD Styles

Centralized shared memory ("Uniform Memory Access" time or "Shared Memory Processor")
Decentralized memory (memory module CPU) get more memory bandwidth, lower memory
Drawback: Longer communication latency
Drawback: Software model more complex

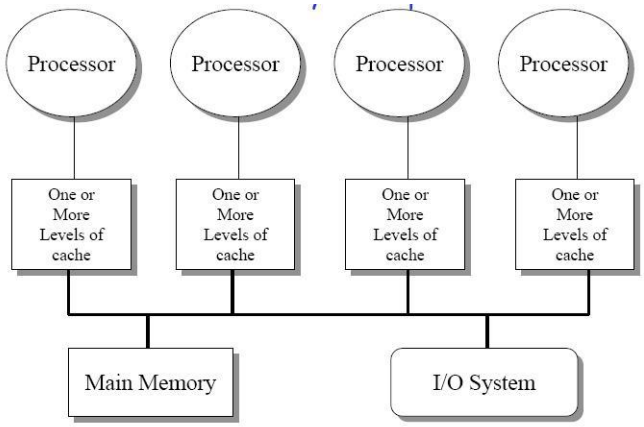
16. Distinguish between shared memory multiprocessor and message-passing multiprocessor.

- A multiprocessor with a shared address space, that address space can be used to communicate data implicitly via load and store operations is shared memory multiprocessor.
- A multiprocessor with a multiple address space, communication of data is done by explicitly passing message among processor is message-passing multiprocessor.

17. What is multicore'?

At its simplest, multi-core is a design in which a single physical processor contains the core logic of more than one processor. It's as if an Intel Xeon processor were opened up and inside were packaged all the circuitry and logic for two (or more) Intel Xcon processors. The multi-core design takes several such processor "cores" and packages them as a single physical processor.The goal of this design is to enable a system to run more tasks simultaneously and thereby achieve greater overall system performance.

18.Draw the basic structure of Basic Structure of a Symmetric Shared Memory Multiprocessor



19. Write the software implications of a multicore processor?

Multi-core systems will deliver benefits to all software, but especially multi-threaded programs.All code that supports HT Technology or multiple processors, for example, will benefit automatically from multi-core processors, without need for modification. Most server-side enterprise packages and many desktop productivity tools fall into this category.

20. What is coarse grained multithreading?

It switches threads only on costly stalls. Thus it is much less likely to slow down the execution an individual thread.

UNIT-5 MEMORY AND I/O SYSTEMS

1. What are the multimedia applications which use caches?

Some Multimedia application areas where cache is extensively used are
*Multimedia Entertainment
*Education
*Office Systems
*Audio and video Mail
*Computer Architecture - Set 6

2. Explain virtual memory technique.

Techniques that automatically move program and data blocks into the physical memory when they are required for execution are called virtual memory technique.

3. What are virtual and logical addresses?

The binary addresses that the processor issues for either instruction or data are called virtual or logical addresses.

4. Define translation buffer.

Most commercial virtual memory systems incorporate a mechanism that can avoid the bulk of the main memory access called for by the virtual to physical addresses translation buffer. This may be done with a cache memory called a translation buffer.

5. What is branch delay slot?

The location containing an instruction that may be fetched and then discarded because of the branch is called branch delay slot.

6. What is optical memory?

Optical or light based techniques for data storage, such memories usually employ optical disk which resemble magnetic disk in that they store binary information in concentric tracks on an electromechanically rotated disks. The information is read as or written optically, however with a laser replacing the read write arm of a magnetic disk drive. Optical memory offer high storage capacities but their access rate is are generally less than those of magnetic disk.

7. What are static and dynamic memories?

Static memory are memories which require periodic no refreshing. Dynamic memories are memories, which require periodic refreshing.

8. What are the components of memory management unit?

A facility for dynamic storage relocation that maps logical memory references into physical memory addresses. A provision for sharing common programs stored in memory by different users .

9. What is the role of MAR and MDR?

The MAR (memory address register) is used to hold the address of the location to or from which data are to be transferred and the MDR(memory data register) contains the data to be written into or read out of the addressed location.

10. Distinguish Between Static RAM and Dynamic RAM?

Static RAM are fast, but they come at high cost because their cells require several transistors. Less expensive RAM can be implemented if simpler cells are used. However such cells do not retain their state indefinitely; Hence they are called Dynamic RAM.

11. Distiguish between asynchronies DRAM and synchronous RAM.

The specialized memory controller circuit provides the necessary control signals, RAS And CAS ,that govern the timing. The processor must take into account the delay in the response of the memory. Such memories are referred to as asynchronous DRAMS.The DRAM whose
operations is directly synchronized with a clock signal. Such Memories are known as synchronous DRAM.

12. What do you mean associative mapping technique?

The tag of an address received from the CPU is compared to the tag bits of each block of the cache to see if the desired block is present. This is called associative mapping technique.

13. What is SCSI?

Small computer system interface can be used for all kinds of devices including RAID storage subsystems and optical disks for large- volume storage applications.

14. What are the two types of latencies associated with storage?

The latency associated with storage is divided into 2 categories
1. Seek Latencies which can be classified into Overlapped seek,Mid transfer seek and Elevator seek.
2. Rotational Latencies which can be reduced either by Zero latency read or Write and Interleave factor.

15. What do you mean by Disk Spanning?

Disk spanning is a method of attaching drives to a single host uadapter. All drives appear as a single contiguous logical unit. Data is written to the first drive first and when the drive is full, the controller switches to the second drive, then the second drive writes until its full.

16. What is SCSI?

Small computer system interface can be used for all kinds of devices including RAID storage subsystems and optical disks for large- volume storage applications.

17. Define the term RELIABILITY

"Means feature that help to avoid and detect such faults. A realible system does not silently continue and delivery result that include interrected and corrupted data, instead it corrects the corruption when possible or else stops.

18. Define the term AVAILABLITY:

"Means features that follow the systerm to stay operational even offen faults do occur. A highly available systerm could dis able do the main functioning portion and continue operating at the reduced capacity".

19. How the interrupt is handled during exception?

* cpu identifies source of interrupt
* cpu obtains memory address of interrupt handles
* pc and other cpu status information are saved
* Pc is loaded with address of interrupt handler and handling program to handle it.

20. What is IO mapped input output?

A memory reference instruction activated the READ M (or)WRITE M control line and does not affect the IO device. Separate IO instruction are required to activate the READ IO and WRITE IO lines ,which cause a word to be transferred between the address aio port and the CPU. The memory and IO address space are kept separate.

21. Specify the three types of the DMA transfer techniques?

--Single transfer mode(cyclestealing mode)
--Block Transfer Mode(Brust Mode)
--Demand Transfer Mode
--Cascade Mode

22. What is an interrupt?

An interrupt is an event that causes the execution of one program to be suspended and another program to be executed.

23. What are the uses of interrupts?

*Recovery from errors
*Debugging
*Communication between programs
*Use of interrupts in operating system

24. Define vectored interrupts.

In order to reduce the overhead involved in the polling process, a device requesting an interrupt may identify itself directly to the CPU. Then, the CPU can immediately start executing the corresponding interrupt-service routine. The term vectored interrupts refers to all interrupthandling schemes base on this approach.

25. Name any three of the standard I/O interface.

*SCSI (small computer system interface),bus standards
*Back plane bus standards
*IEEE 796 bus (multibus signals)
*NUBUS & IEEE 488 bus standard

26. What is an I/O channel?

An I/O channel is actually a special purpose processor; also called peripheral processor. The main processor initiates a transfer by passing the required information in the input output channel. The channel then takes over and controls the actual transfer of data.

27. What is a bus?

A collection of wires that connects several devices is called a bus.

28. Define word length?

Each group of n bits is referred to as a word of information and n is called the word length.

29. Why program controlled I/O is unsuitable for high-speed data transfer?

In program controlled i/o considerable overhead is incurred. Because several program instruction have to be executed for each data word transferred between the external devices and MM. Many high speed peripheral; devices have a synchronous modes of operation. That is data transfers are controlled by a clock of fixed frequency, independent of the CPU.

30. What is the function of I/O interface?

The function is to coordinate the transfer of data between the CPU and external devices.

31. What is NUBUS?

A NUBUS is a processor independent, synchronous bus standard intended for use in 32 bitmicro processor system. It defines a backplane into which upto 16 devices may be plugged each in the form of circuit board of standard dimensions.

32. Name some of the IO devices.

*Video terminals
*Video displays
*Alphanumeric displays
*Graphics displays
* Flat panel displays
*Printers
*Plotters

33. What are the steps taken when an interrupt occurs?

*Source of the interrupt
*The memory address of the required ISP
* The program counter &CPU information saved in subroutine
*Transfer control back to the interrupted program

34. Define interface.

The word interface refers to the boundary between two circuits or devices

35. What is programmed I/O?

Data transfer to and from peripherals may be handled using this mode. Programmed I/O operations are the result of I/O instructions written in the computer program.

36. Types of buses.

-Synchronous bus
-Asynchronous bus

37.Define Synchronous bus.

- Synchronous bus on other hand contains synchronous clock that is used to validate each and every signal.
- Synchronous buses are affected noise only when clock signal occurs.
- Synchronous bus designers must control with meta stability when attempting different clock signal Frequencies
- Synchronous bus of meta stability arises in any flip flop. when time will be violated.

38. Define Asynchronous bus.

- Asynchronous buses can mistake noise pulses at any time for valid handshake signal.
- Asynchronous bus designer must deal with events that like synchronously.
- It must contend with meta stability when events that drive bus transaction.

-When flip flop experiences effects can occur in downstream circuitry unless proper design technique which are used

PART-B-16 MARK QUESTIONS

UNIT-I OVERVIEW & INSTRUCTIONS

1. Explain the Eight ideas of the Computer architects in detail.(8)
2. Explain the components of a computer with the block diagram in detail.(16)
3. Explain the technologies for building computer over time with a neat graph.(6)
4. Explain the chip manufacturing process with a neat diagram in detail.(10)
4.Explain the techniques used to measure the performance of a computer.(8)
5.(i)Prove that how performance and execution are inverse to each other.(2)
(ii) If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?(2)
(iii)Write the formula to calculate the CPU execution time for a program.(2)
(iv) Write the formula to calculate the CPU clock cycles.(2)
(v) Write the formula to calculate the classic CPU Performance equation.(2)
6. Explain how clock rate and power are related to each other in microprocessor over years with a neat graph.(6)
7. Explain the need to switch from uniprocessors to multiprocessors and draw the performance chart for processors over years. (6)
8. Explain the basic instruction types with examples.(6)
9. (i)Explain the different types of instruction set architecture in detail(6)
(ii)Explain MIPS assembly language notation for arithmetic, Data transfer,logical,conditional branch and unconditional branch operations.
10. What do you mean by addressing modes? Explain various addressing modes with the help of examples.(16)

UNIT-2 ARITHMETIC OPERATIONS

1.Explain the design of ALU in detail.(16)
2.Explain with an example how to multiply two unsigned binary numbers.(8)
3.Describe in detail booth's multiplication algorithm and its hardware implementation?
4.Explain the Working of a Carry-Look Ahead adder.(16)
5.Derive and explain an algorithm for adding and subtracting two floating point binary numbers.(8)
6.Design a 4-bit adder/subtractor circuit using full adders and explain it's function?(8)
7.Describe the algorithm for integer division with suitable examples.(16)
8.Perform the multiplication using Carry save addition of summands. (6)
45 X 45
9.Perform the integer division using non-restoring and restoring division. (10)
9 / 4

UNIT-3PROCESSOR AND CONTROL UNIT

1.Discuss the basic concepts of pipelining.
2.State and explain the different types of hazards that can occur in a pipeline.
3.Draw and explain the modified three-bus structure of the processor suitable for four –stage pipelined execution. How this structure is suitable to provide four-stage pipelined execution?
4.What is data hazard?Explain the methods for dealing with the data hazards
5.Describe the data and control path techniques in pipelining.(10)
6.What is instruction hazard?Explain in detail how to handle the instruction hazards in pipelining with relevant examples.(10)
7.Describe the techniques for handling control hazards in pipelining.(10)
8.Write note on exception handling.(6)

UNIT – IV PARALLELISM

1. Explain instruction level parallelism in detail?
2. Explain parallel processing challenges in detail?

3. Explain briefly the Flynn's classification?
4. What is hardware multithreading? Explain the various approaches in detail?
5. Explain multicore processors in detail?
6. Compare SISD, SIMD, MISD, and MIMD in detail?
7. Explain the following:
   i) Implicit and Explicit multithreading.
   ii) Interleaved, Blocked and Simultaneous multithreading.
8. What are multicore processors? Explain the common configurations that support multiprocessing?

UNIT – V MEMORY AND I/O SYSTEMS

1. Explain with the block diagram the DMA transfer in a computer system.
2. Describe in detail about IOP organization.
3. Describe the data transfer method using DMA.
4. Write short notes on the following
   (a) Magnetic disk drive (8)
   (b) Optical drives (8)
5. Discuss the design of a typical input or output interface.
6. What are interrupts? How are they handled?