Q.23 Which of the following technology can give high speed RAM?

(A) TTL (B) CMOS
(C) ECL (D) NMOS

Ans. (C)

Q.24 In 8085 microprocessor how many I/O devices can be interfaced in I/O mapped I/O technique?

(A) Either 256 input devices or 256 output devices.
(B) 256 I/O devices.
(C) 256 input devices & 256 output devices.
(D) 512 input-output devices.

Ans. (C)

Q.25 After reset, CPU begins execution of instruction from memory address

(A) 0101_H (B) 8000_H
(C) 0000_H (D) $FFFF_H$

Ans. (C)

Q.26 Which is true for a typical RISC architecture?

(A) Micro programmed control unit
(B) Instruction takes multiple clock cycles.
(C) Have few registers in CPU.
(D) Emphasis on optimizing instruction pipelines.

Ans. (A)

Preview from Notesale.co.uk
Page 5 of 129

Q.27 When an instruction is read from the memory, it is called

(A) Memory Read cycle (B) Fetch cycle
(C) Instruction cycle (D) Memory write cycle

Ans. (B)

Q.28 Which activity does not take place during execution cycle?

(A) ALU performs the arithmetic & logical operation.
(B) Effective address is calculated.
(C) Next instruction is fetched.
(D) Branch address is calculated & Branching conditions are checked.

Ans. (D)

Q.29 A circuit in which connections to both AND and OR arrays can be programmed is called

(A) RAM (B) ROM
(C) PAL (D) PLA

Ans. (A)

(C) NOP (D) END

Ans. (D)

Q.44 A demultiplexer can be used as
(A)Encoder (B)Decoder
(C)Multiplexer (D)None of the above

Ans. (B)

Q.45 Excess-3 equivalent representation of $(1234)_H$ is
(A) $(1237)_{Ex-3}$ (B) $(4567)_{Ex-3}$
(C) $(7993)_{Ex-3}$ (D) $(4663)_{Ex-3}$

Ans. (B)

Q.46 Which of the memory holds the information when the Power Supply is switched off?
(A) Static RAM (B) Dynamic RAM
(C) EEROM (D) None of the above

Ans. (C)

Q.47 Minimum no. of NAND gate required to implement 2 Ex-OR function is
(A)2 (B)3
(C)4 (D)5

Ans. (C)

Preview from Notesale.co.uk
Page 8 of 129

Q.48 Which of the following interrupt is maskable?
(A)INTR (B)RST 7.5
(C)TRAP (D)Both (A) and (B)

Ans. (B)

Q.49 Which of the following expression is not equivalent to x?
(A) x NAND x (B) x NOR x
(C) x NAND 1 (D) x NOR 1

Ans. (D)

Q.50 Word 20 contains 40
Word 30 contains 50
Word 40 contains 60
Word 50 contains 70
Which of the following instructions does not, load 60 into the Accumulator
(A) Load immediate 60
(B) Load direct 30
(C) Load indirect 20

(D) both (A) & (C)

Ans. (B)

Q.51 An interrupt for which hardware automatically transfers the program to a specific memory location is known as
(A) Software interrupt
(B) Hardware interrupt
(C) Maskable interrupt
(D) Vector interrupt

Ans. (B)

Q.52 Synchronous means _____
(A) At irregular intervals
(B) At same time
(C) At variable time
(D) None of these

Ans. (B)

Q.53 'n' Flip flops will divide the clock frequency by a factor of
(A) n^2 (B) n
(C) 2^n (D) log n

Ans. (B)

Preview from Notesale.co.uk
Page 9 of 129

Q.54 In DMA the data transfer is controlled by
(A)Microprocessor (B) RAM
(C)Memory (D) I/O devices

Ans. (D)

Q.55 The number of instructions needed to add a numbers an store the result in memory using only one address instruction is
(A)n (B) n - 1
(C)n +1 (D) Independent of n

Ans. (D)

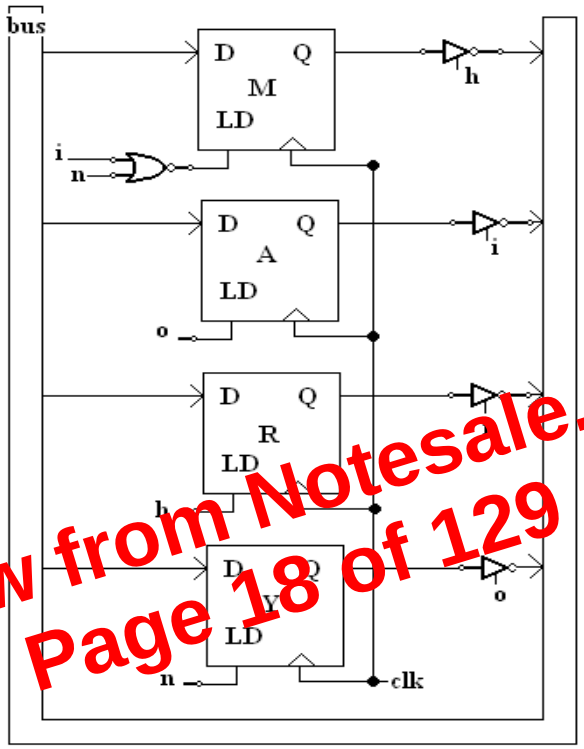
Q.56 Negative numbers cannot be represented in
(A)Signed magnitude form
(B)I's complement form
(C)2's complement form
(D)8-4-2-1 code

Ans. (C)

Q.57 Which of the following architecture is/are not suitable for realizing SIMD
(A)Vector Processor (B) Array Processor

n: Y ←R, M ←R
Assume M, A, R and Y are to be one bit D – flip- flop. (6)

Ans:
The hardware realisation of R.TL behavior is shown below by using common bus and tri state buffer. All the FFs get the same clock pulse. Depending on the control signal of tri state buffer, the source FF is selected, and depending on control signals connected to 'LD' of FF's the destination FF is selected.



Preview from Notesale.co.uk
Page 18 of 129

Q.8 What do you mean by program control instructions? With a neat diagram, explain how the status register containing overflow, zero, sign and carry flags works with the status of the accumulator content obtained from ALU. (3+4)

Ans:
Program control type instructions, when executed by the processor, may change the address value of the Program Counter and cause type flow of control to be altered. Program control instruction specifies conditions for altering the content of Program Counter. This causes break in the sequence of instruction execution. This Instruction also gives the capability for branching to different Program segments.
Examples - Branch., Jump, Skip, Call, Return etc.
Status bits are set or reset depending on the result of a logical or arithmetic manipulation of accumulator data. So status bits are called condition - code bits or flag bits. These status bits constitute status register.
The hardware realization of status register containing overflow, Zero, Sign, Carry flag is shown below –

Ans:
In a general microprocessor system, there is a special register known as stack pointer, which holds the address of the top of the stack. In some microprocessor, register stack is provided. In order to indicate the stack full condition and stack empty condition, two flags are used. These two flags are known as EMPTY flag & FULL flag. The empty flag is set when the stack is completely empty. Full flag is set only when all the stack locations are filled with data. Stack is essential for implementing subroutine call and interrupts.
Stacks operate in two principles
(1) LIFO i.e. Last in First Out
(2) FIFO i.e. first in first out.
These principles of operation depends on stack architecture. Most of general purpose processor use LIFO principle for their stack.
If the stack is organized in R/W memory, than the stack pointer is loaded with same address to initialize. The memory stack grow down word i.e. with each Push operations, stack pointer is decremented. The situation is just reverse on register stack.

Q.11 What are the advantages of assembly language? How is it different from high-level language? (6)

Ans:
Writing program for a computer consists of specifying, directly or indirectly, a sequence of machine instructions. The machine instruction stored in RAM of the computer is in binary format. This binary format is very difficult to use and to – troubleshoot. So programs are written by user by using English like symbols of the alpha numeric character set, which is known as assemble language. The assembler converts these assembly language programs to binary form.
Advantages of assemble language program is it is easy to use. It is easy to troubleshoot, it is fast to execute than high level language program.
A programming language is defined by a set of rules. Users must conform to all format rules of the assembly language if it is to be translated correctly. Each microprocessor has its own assembly language format. The assembly language use predefined rules that specify the symbols that can be used & how they may be combined to form a line of code.
Some of the common rules are
(i) The label field may be empty or it may specify a symbolic address.
(ii) The instruction field specify a machine instruction or a Pseudo instructions.
(iii) The comment field may be empty or it may include a comment.
(iv) The symbolic address consists of up to four alphanumeric characters.
(v) Symbolic address in the label field is terminated by a comma so that it will be recognized as a label by the assembler.
(vi) The comment field is preceded by a slash foe assembler to recognize the beginning of a comment field.

Q.12 What is vertical micro code? State the design strategy of a vertical micro coded control unit. (6)

Ans:

Preview from Notesale.co.uk
Page 20 of 129

Example: Refer table 10-3 from page 348, Morris mano (3rd Edition)

Q. 26 Using 8-bit 2's complement representation of negative numbers, perform the following computations:

(i) - 35 + (-11) (ii) 19 - (- 4)

Ans.

35 = 00100011
-35 = 11011100
11011101
11 = 00001011
-11 = 11110100
= 11110101
-35+ (-11) = 11011101
+ 11110101

111010010

(ii) 19 = 0001 0011
4 = 00000100
-4 = 11111100
19 - (- 4) = 19 + 4
00010011
00000100

00010111

Preview from Notesale.co.uk
Page 33 of 129

Q. 27 Consider a cache (M_1) and memory (M_2) hierarchy with the following characteristics:

M_1 : 16 k words, 50 ns access time

M_2 : 1 M words, 400 ns access time

Assume 8 words cache blocks and a set size of 256 words with set associative mapping.

(i)Show the mapping between M_2 and M_1.

(ii)Calculate the Effective Memory Access time with a cache hit ratio of h = .95.

Ans.

(i) Main Memory = 1M words.
$= 2^{20}$ words.
Block size = 8 words.

main memory = $\frac{2^{20}}{8} = 2^{17}$ blocks

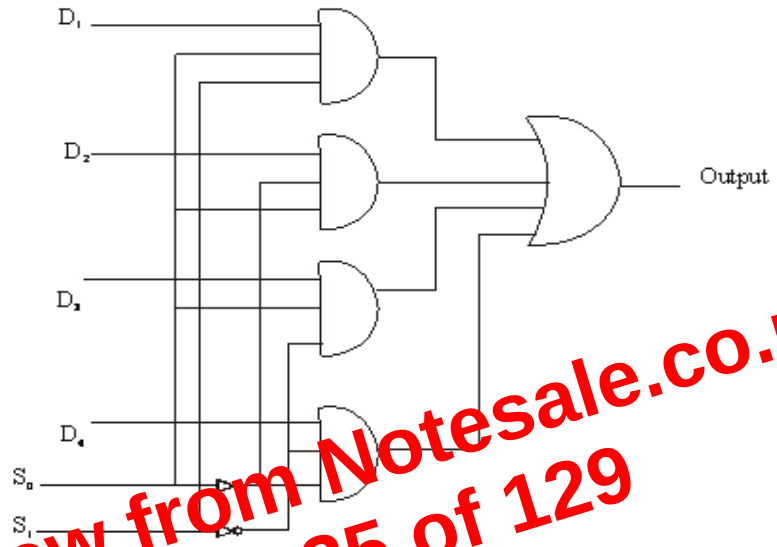
Cache memory = 16 k words.

Therefore Cache memory = $\frac{16K}{8} = \frac{2^{14}}{2^3} = 2^{11}$ blocks.

Set size = 265 words.

Truth Table		
S_0	S_1	O/P
0	0	D_4
0	1	D_a
1	0	D_2
1	1	D_1

Output = $D_4 S_0 S_1 + D_3 S_0 S_1 + D_2 S_0 S_1 + D_1 S_0 S_1$

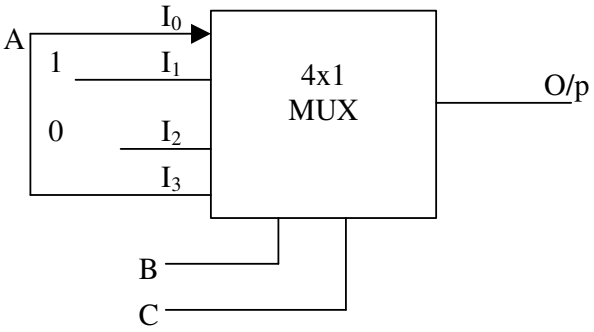


Preview from Notesale.co.uk
Page 35 of 129

Q.28 Implement the following by using 4:1 multiplexer
$P = \Pi(M_0, M_1, M_5, M_7)$.

Ans.

	I_0	I_1	I_2	I_3
A′	0	1	2	3
A	4	5	6	7
	A′	1	0	A



B and C are the selection lines.

Cons:
- Requires enough RAM
- Only runs on the same architecture as the host OS

3. Emulation

Emulators will completely emulate the target CPU and hardware (e.g. sound cards, graphics cards, etc). Emulators are the "old way" of running multiple OSs on a single computer. Emulation on PCs these days is only good for non-OS usages (e.g. game consoles, embedded systems) or specific OS/CPU development purposes.

Pros:
- Best solution for embedded/OS development
- Doesn't interfere with the underlying host OS
- Can be ported to any architecture

Cons:
- Can be very slow
- No 3D or other exotic PC hardware support
- Requires enough RAM

Being a traditional geek chick myself, I still prefer the manual re - partioning method for my PCs (I like the clean nature of it), but for a MacTel I would much prefer Boot Camp's special portioning scheme (if Vista, Linux are supported properly — otherwise, Virtualization is my next best option on MacTels). Tell us what's your preferred method is below

Q. 34 Explain with an example, how effective address is calculated in different types of addressing modes.

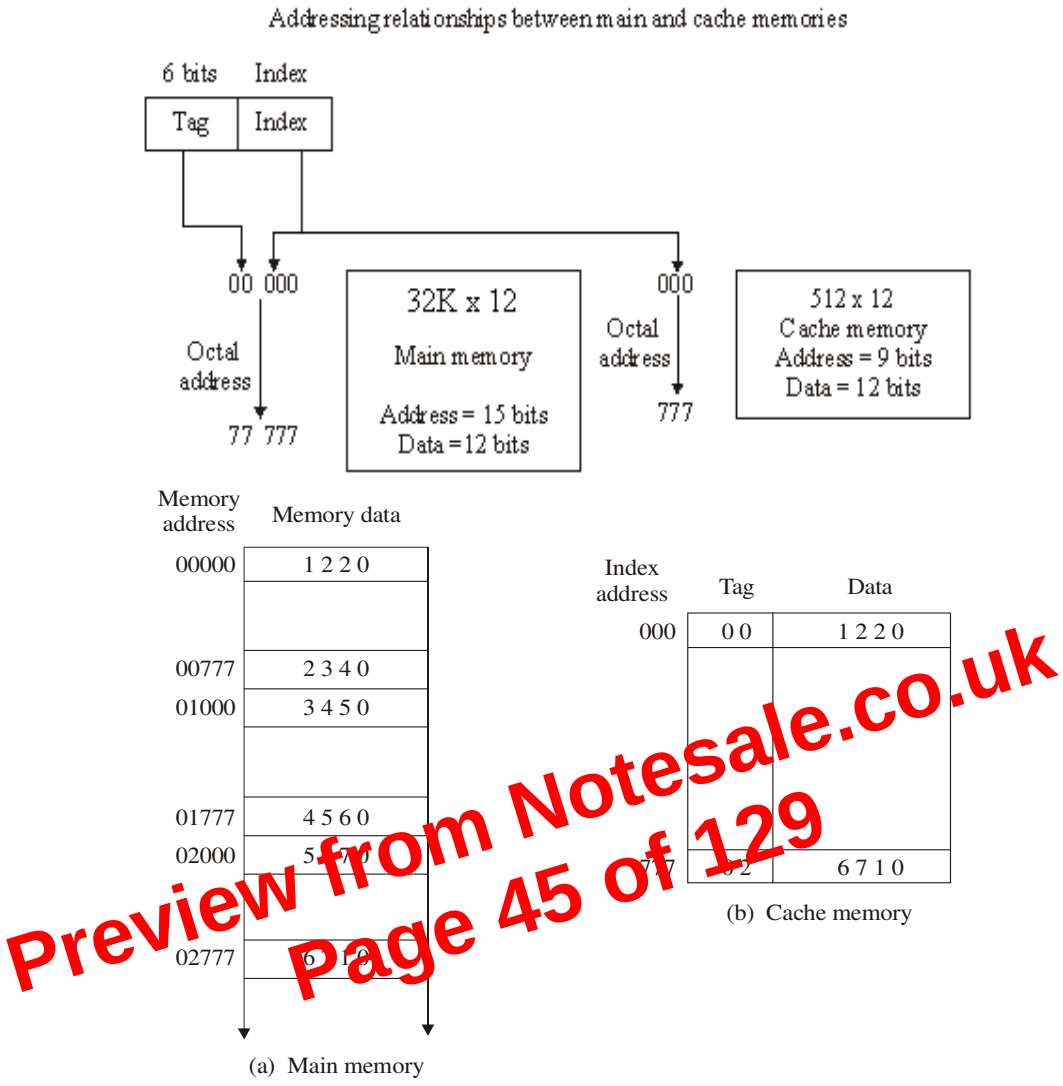
Ans.

To explain the difference between the various modes, the two word instruction at address 200 and 201 is a "load to AC" instruction with an address field equal to 500. The first word of the instruction specifies the operation code and mode, and the second word specifies the address part. PC has the value 200 for fetching this instruction. The content of processor register R1 is 400, and the content of an index register XR is 100. AC receives the operand after the instruction is executed. The figure lists a few pertinent addresses and shows the memory content at each of these addresses for each possible mode. We calculate the effective address and the operand that must be loaded into AC. In the direct address mode the effective address is the address part of the instruction 500 and the operand to be loaded into AC is 800. In the immediate mode the second word of the instruction is taken as the operand rather than an address, so 500 is loaded into AC. In the indirect mode the effective address is stored in memory at address 500. Therefore the effective address is 800 and the operand is 300. In the Index mode the effective address is XR + 500 = 100 + 500 = 600 and the operand is 900. In the register mode the operand is in R1 and 400 is loaded into AC.

The Autoincrement mode is the same as the register indirect mode except that R1 is incremented to 401 after the execution of the instruction. The Autodecrement mode decrements R1 to 399 prior to the execution of the instruction. In the relative mode the effective address is 500+202=702 and the operand is 325.

In the register indirect made the effective address is 400, equal to the content of R1 and the operand loaded into AC is 700.

Preview from Notesale.co.uk
Page 39 of 129



Direct mapping cache organization

To see how the direct-mapping organization operates, consider the numerical example shown. The word at address zero is presently stored in the cache (index = 000, tag = 00, data = 1220). Suppose that the CPU now wants to access the word at address 02000. The index address is 000, so it is used to access the cache. The two tags are then compared. The cache tag is 00 but the address tag is 02, which does not produce a match. Therefore, the main memory is accessed and the data word 5670 is transferred to the CPU. The cache word at index address 000 is then replaced with a tag of 02 and data of 5670.

The direct-mapping example just described uses a block size of one word. The same organization but using a block size of 8 words is shown.

The index field is now divided into two parts: the block field and the word field. In a 512-word cache there are 64 blocks of 8 words cache, since 64 x 8 = 512. The block number is specified with a 6-bit field and the word within the block is specified with a 3-bit field. The tag field stored within the cache is common to all eight words of the same block. Every time a miss occurs, an entire block of eight words must be transferred from main memory to cache memory. Although this takes extra

Preview from Notesale.co.uk
Page 45 of 129

	MOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
ANDOP	AND	U	JMP	FETCH
ADD	ORGO			
	MOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ADD	U	JMP	FETCH
STORE	ORG8			
	NOP	I	CALL	INDRCT
	ACTRD	U	JMP	NEXT
	WRITE	U	JMP	FETCH
COMPLEMENT	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	COM	U	JMP	FETCH

Q.68 What do you mean by software of hardware interrupts? How there are used in a microprocessor system?

Ans.

Software and hardware interrupt:

The software interrupts are program instructions. These instructions are inserted at desired location in a program. A program generated interrupt also called trap, which stops current processing in order to request a service provided by the CPU. While running a program if software interrupt instruction is encountered the CPU initiates an interrupt. For example a program might generate a software interrupt to read input from keyboard. Hardware interrupt is a type of interrupt generated either externally by the hardware devices such as input/ output ports, keyboard and disk drive etc or internally by the microprocessor. External hardware interrupts are used by device to request attention from CPU. Internal hardware interrupts are generated by the CPU to control events.

Preview from Notesale.co.uk
Page 75 of 129

Q.69 What are the reasons of Pipe-Line conflicts is a Pipe Lined processor? How are they resolved?

Ans.

There are three major difficulties that cause the instruction pipeline to deviate form its normal operation.

1) Resource conflicts caused by access to memory by two segments at the same time. Most of these conflicts can be resolved by using separate instruction and data memories.

2) Data dependency conflicts arise when an instruction depends on the result of a previous instruction, but this result is not available.

3) Branch difficulties arise from branch and other instructions that change the value of PC. In computer, for solving conflicts problems to the compiler that translates the high level programming language into a machine language program. The compiler for such computer is designed to detect a data conflict and re order the instructions, to delay the loading of the conflicting data by inserting no-operation instructions. This method is referred to as delayed load.

The fetch cycle - The fetch cycle, which occurs at the beginning of each instruction cycle and causes an instruction to be fetched from memory.

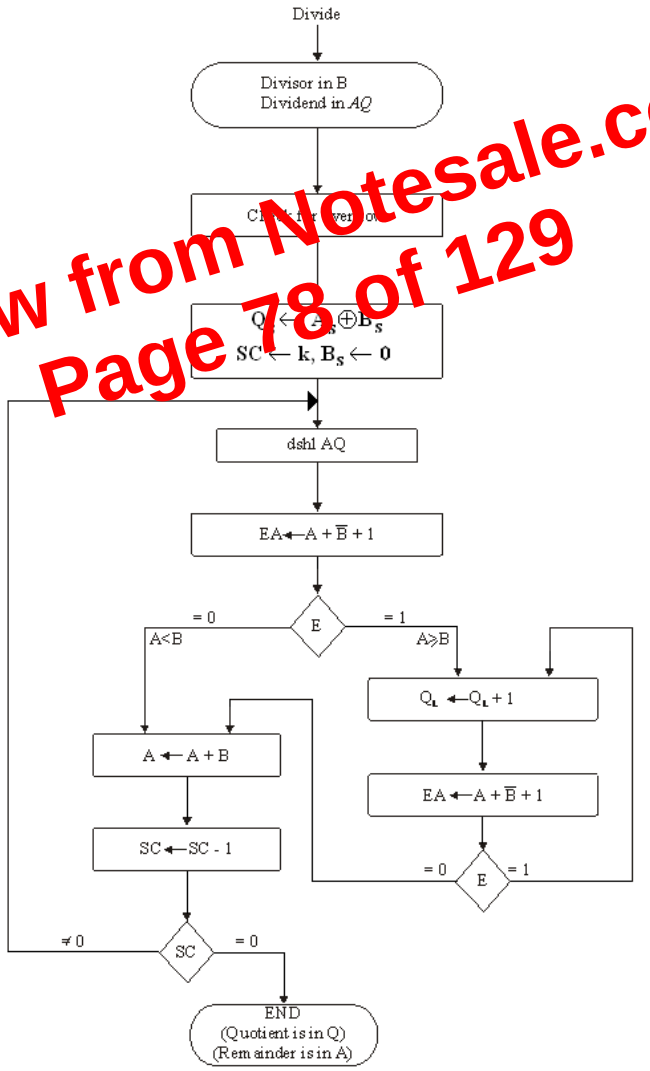
Instruction cycle:- The instruction cycle can be decomposed into a sequence of elementary micro-operations. There is one sequence each for the fetch, indirect and interrupt cycle and for execute cycle, there is one sequence of micro-operations for each opcode.

Micro - operations :-

– A computer executes a program
– Fetch/ execute cycle
– Each cycle has a number of steps are pipelining
– Called micro-operation
– Each step does very little

Q. 73 With neat flow chart, explain the procedure for division of floating point numbers carried out in a computer.

Ans:



Flowchart for decimal division

Preview from Notesale.co.uk
Page 78 of 129

Decimal division is similar to binary division except of course that the quotient digits may have any of the 10 values from 0 to 9. In the restoring division method, the divisor is subtracted from the dividend or partial remainder as many times as necessary until a negative reminder results. The correct remainder is then restored by adding the divisor. The digit in the quotient reflects the number of subtractions up to but excluding the one that caused the negative difference. The decimal division algorithm is shown. It is similar to the algorithm with binary data except for the way the quotient bits are formed. The dividend (or partial remainder) is shifted to the left, with its most significant digit placed in A_e .The divisor is then subtracted by adding its 10's complement value. Since B_e is initially cleared, its complement value is 9 as required. The carry in E determines the relative magnitude of A and B, If E=0, it signifies that A<B. In this case the divisor is added to restore the partial remainder and Q_L stays at 0 (inserted there during the shift). If E=1, it signifies that A≥B. The quotient digit in Q_L is incremented once and the divisor subtracted again. This process is repeated until the subtraction results in a negative difference which is recognized by E being 0. When this occurs, the quotient digit is not incremented but the divisor is added to restore the positive remainder. In this way, the quotient digit is made equal to the number of times that the partial remainder "goes" into the divisor. The partial remainder and the quotient bits are shifted once to the left and the process is repeated k times to form k quotient digits. The remainder is then found in register A and the quotient is in register Q. The value of E is neglected.

Preview from Notesale.co.uk
Page 79 of 129

Q.74 Give the flow table for register contents used in implementing booth's algorithm for the multiplier = - 6 and multiplicant = + 5.

Ans.

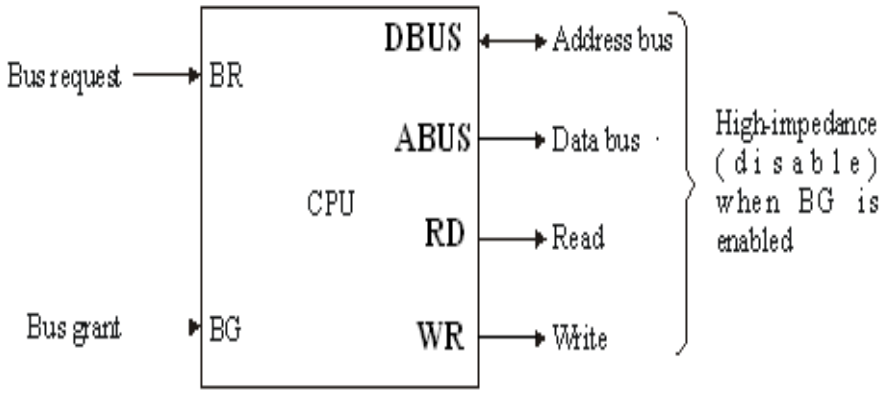
Q_n	Q_{n+1}	BR = 0101 BR +1 = 1011	AC	QR	Q_{n+1}	SC
		Initial	0000	1010	0	100
0	0	Ashr	0000	0101	0	011
1	0	Subtract BR	$\frac{1011}{1011}$			
		ashr	1101	1010	1	010
0	1	Add BR	$\frac{0101}{0010}$			
		ashr	0001	0101	0	001
1	0	Subtract BR	$\frac{1011}{1100}$			
		ashr	1110	0010	1	000

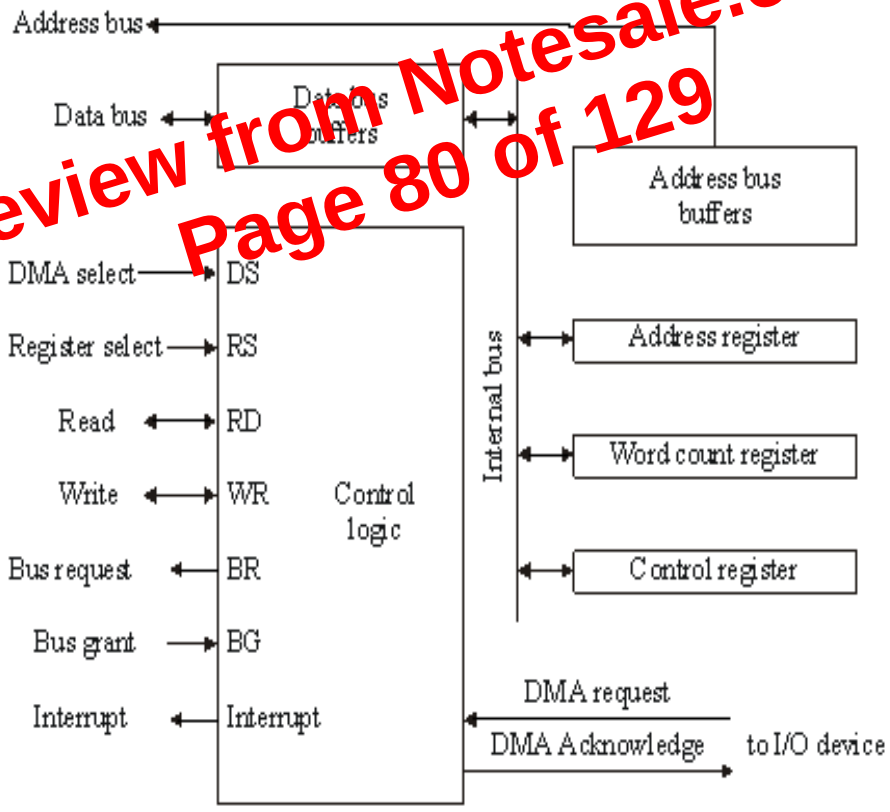
Final product is 11100010

Q.75 What do you mean by initialization of DMA controller? How DMA controller works? Explain with suitable block diagram.

Ans.
Figure below shows two control signals in the CPU that facilitate the DMA transfer. The bus request (BR) input is used by the DMA controller to request the CPU to

relinquish control of the buses. The CPU activates the bus grant (BG) output to inform the external DMA that the buses are in the high-impedance state. The DMA that originated the bus request can now take control of the buses to conduct memory transfers without processor intervention. When the DMA terminates the transfer, it disables the bus request line. The CPU disables the bus grant.

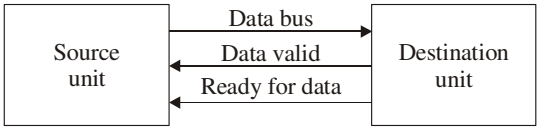




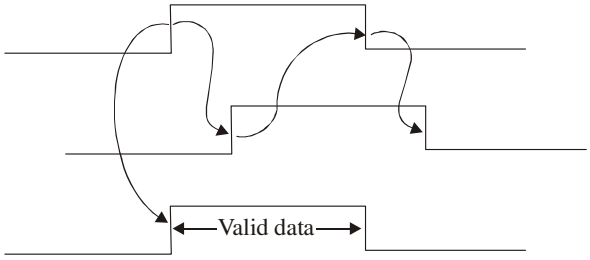
Preview from Notesale.co.uk
Page 80 of 129

When the DMA takes control of the bus system, it communicates directly with the memory. Figure shows the block diagram of a typical DMA controller. The unit communicates with the CPU via the data bus and control lines. The registers in the DMA are selected by the CPU through the address bus by enabling the *DS* (DMA

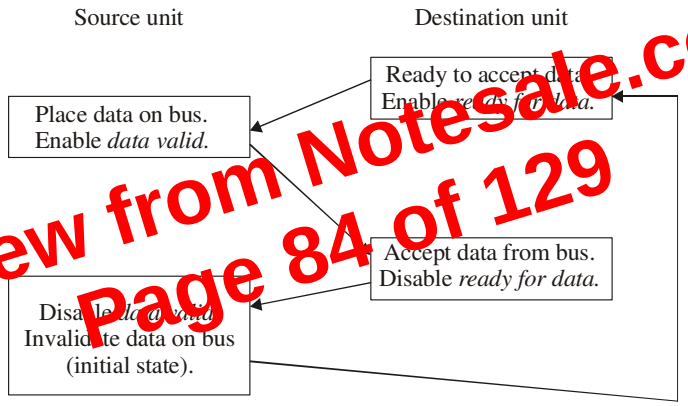
Destination-initiated transfer using handshaking.



(a) Block diagram



(b) Timing diagram



(c) Sequence of events

Preview from Notesale.co.uk
Page 84 of 129

has been changed to *ready for data* to reflect its new meaning. The source unit in this case does not place data on the bus until after it receives the *ready for data* signal from the destination unit. From there on, the handshaking procedure follows the same pattern as in the source-initiated case. Note that the sequence of events in both cases would be identical if we consider the *ready for data* signal as the complement of *data accepted*. In fact, the only difference between the source-initiated and the destination-initiated transfer is in their choice of initial state.

iii) **Isolated vs memory mapped I/O:-**

In the isolated I/O configuration, the CPU has distinct input and output instructions, and each of these instructions is associated with the address of an interface register. When the CPU fetches and decodes the operation code of an input or output instruction, it places the address associated with the instruction into the common address lines. The isolated I/O method isolates memory and I/O addresses so that memory address values are not affected by interface address assignment since each has its own address space. The other alternative is to use the same address space for

both memory and I/O. This configuration is referred to as *memory-mapped I/O*. In a memory-mapped I/O organization there is no specific input or output instructions. The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words. Computers with memory-mapped I/O can use memory-type instructions to access I/O data. The advantage is that the load and store instructions used for reading and writing from memory can be used to input and output data from I/O registers. In a typical computer, there are more memory-reference instructions than I/O instructions. With memory-mapped I/O all instructions that refer to memory are also available for I/O.

iv) **RISC architecture**:-

The concept of RISC architecture involves an attempt to reduce execution time by simplifying the instruction set of the computer. The major characteristics of a RISC processor are:

1. Relatively few instructions.
2. Relatively few addressing modes.
3. Memory access limited to load and store instructions.
4. All operations done within the registers of the CPU.
5. Fixed-length, easily decoded instruction format.
6. Single-cycle instruction execution.
7. Hardwired rather than microprogrammed control.
8. Faster execution.

Other characteristics attributed to RISC architecture are:

1. A relatively large number of registers in the processor unit.
2. Use of overlapped register windows to speed-up procedure call and return.
3. Efficient instruction pipeline.
4. Compiler support for efficient translation of high-level language programs into machine language programs.

Preview from Notesale.co.uk
Page 85 of 129

A large number of registers is useful for storing intermediate results and for optimizing operand references. The advantage of register storage as opposed to memory storage is that registers can transfer information to other registers much faster than the transfer of information to and from memory. Thus register-to-memory operations can be minimized by keeping the most frequent accessed operands in registers. Studies that show improved performance for RISC architecture do not differentiate between the effects of the reduced instruction set and the effects of a large register file.

Q. 78 Explain direct mapping of cache memory system.

Ans.

Associative memories are expensive compared to random-access memories because of the added logic associated with each cell. The possibility of using a random-access memory for the cache is investigated. The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the *index* field and the remaining six bits form the *tag* field. The figure shows that main memory needs an address that includes both the tag and the index bits. The number of bits in the index field is equal to the number of address bits required to access the cache memory. In the general case, there are 2^k words in cache memory and 2^n words in

zero is presently stored in the cache (index = 000, tag = 00, data = 1220). Suppose that the CPU now wants to access the word at address 02000. The index address is 000, so it is used to access the cache. The two tags are then compared. The cache tag is 00 but the address tag is 02, which does not produce a match. Therefore, the main memory is accessed and the ata word 5670 is transferred to the CPU. The cache word at index address 000 is then replaced with a tag of 02 and data of 5670.

Q.79 What do you mean by locality of reference?

Ans.
The references to memory at any given internal of time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of locality of reference. The locality of reference property, that over a short internal of time, the addresses generated by a typical program refer to a few localized area of memory repeatedly, while the remainder of memory is accessed relatively infrequently.

Q.80 A virtual memory system has an address space of 8k words, memory space of 4k words and Page & Block size of 1k words. The following page reference changes occur during a given time interval.

4, 2, 0, 1, 2, 6, 1, 4, 0, 1, 0, 2, 3, 5, 7

Determine the four pages that are resident in main memory after each Page reference change if the replacement algorithm used is (i)FIFO (ii) LRU.

Preview from Notesale.co.uk
Page 87 of 129

Ans.
An address space of 8K and a memory of 4K words and page Block size of 1K words. Four pages of address space may reside in main memory in any one of the four blocks.

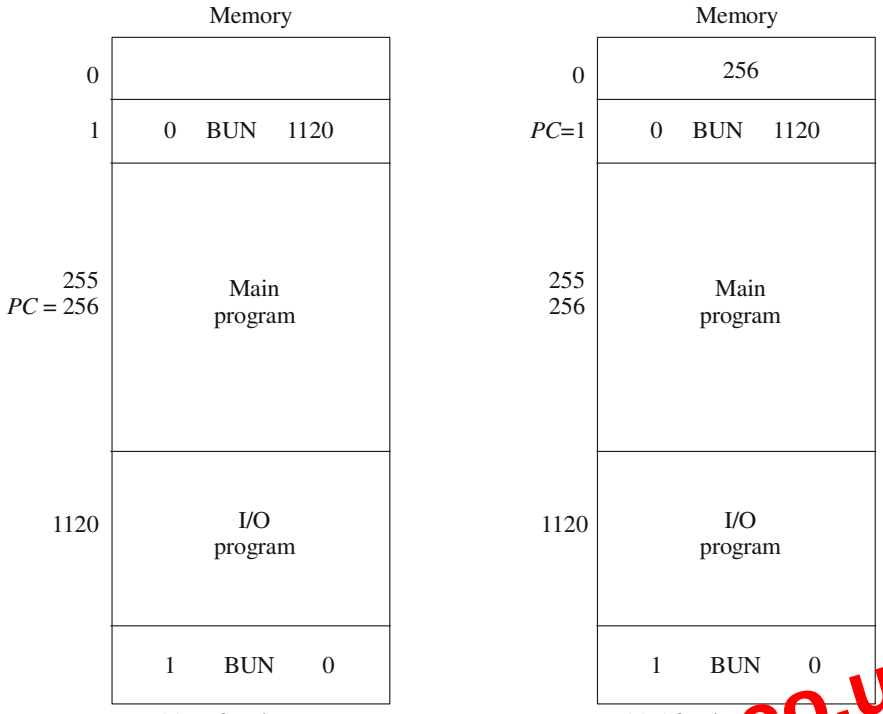
Page 0	
Page 1	
Page 2	Block 0
Page 3	Block 1
Page 4	Block 2
Page 5	Block 3
Page 6	
Page 7	

Address Space
$N = 8K = 2^{13}$

Memory Space
$M = 4K = 2^{12}$

(1) **FIFO**
String 4, 2, 0, 1, 2, 6, 1, 4, 0, 1, 0, 2, 3, 5, 7

4	2	0	1	2	6	1	4	0	1	0	2	3	5	7
4	4	4	4		6		6				6	6	5	5
	2	2	2		2		4				4	4	4	7
		0	0		0		0				2	2	2	2
			1		1		1				1	3	3	3



Register transfer statement:-

The interrupt cycle is initiated after the last execute phase if the interrupt flip-flop is equal to 1. This flip-flop is set to 1 if IEN = 1 and either FG1 or FG0 are equal to 1. When timing signal T_0 T_1 or T_2 are active. This can be expressed with the following register transfer statement:

$T_0'T_1'T_2'$(IEN)(FG1+FG0):R ← 1

Preview from Notesale.co.uk
Page 103 of 129

Q.101 Explain all the phases of instruction cycle.

Ans.

Instruction Cycle:-

A program residing in the memory unit of the computer consists of a sequence of instructions. The program is executed in the computer by going through a cycle for each instruction. Each instruction cycle in turn is subdivided into a sequence of sub cycles or phases. In the basic computer each instruction cycle consists of the following phases:

1. Fetch an instruction from memory.
2. Decode the instruction
3. Read the effective address from memory if the instruction has an indirect address.
4. Execute the instruction.

Fetch and Decode:-

The program counter PC is loaded with the address of the first instruction in the program. The sequence counter SC is cleared to 0, providing a decoded timing signal T_0. After each clock pulse, SC is incremented by one, so that the timing

signals go through a sequence T_0, T_1, and so on. The microoperation for the fetch and decode phases can be specified by the following register transfer statements.

T_0 : AR ← PC

T_1 : IR ← M[AR], PC ← PC + 1

T_2 : D_0 . . . . D_7 ← Decode IR(12-14), AR ← IR (011),
I ← IR(15)

To provide the data path for the transfer of PC to AR we must apply timing signal T_0 to achieve the following connection:

1. Place the content of PC onto the bus by making the bus selection inputs S_2 S_1 S_0 equal to 010.
2. Transfer the content of the bus to AR by enabling the LD input of AR.

T_1 : IR ← M[AR], PC ← PC + 1

The next check transition initiates the transfer from PC to AR since T_0 = 1. In order to implement the second statement T_1: IR ← M[R], PC ← PC +1
It is necessary to use timing signal T1, to provide the following connections in the bus system.

1. Enable the read input of memory.
2. Place the content of memory onto the bus by making
   S_2 S_1 S_0 = 111.
3. Transfer the content of the bus to IR by enabling the LD input of IR.
4. Increment PC by enabling the INR input of PC.

The three instruction types are subdivided into four separate paths. The selected operation is activated with the clock transition associated with timing signal T_3.

This can be symbolized as follows:

D_7'I'T_3 : AR ← M[AR]

D_7'I'T_3 : Nothing

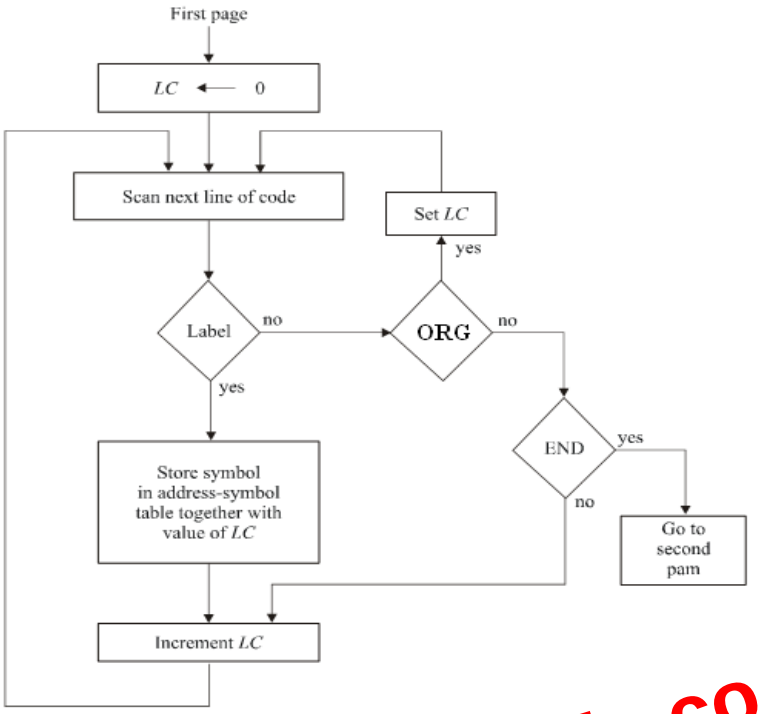
D_7'I'T_3 : Execute a register-reference instruction

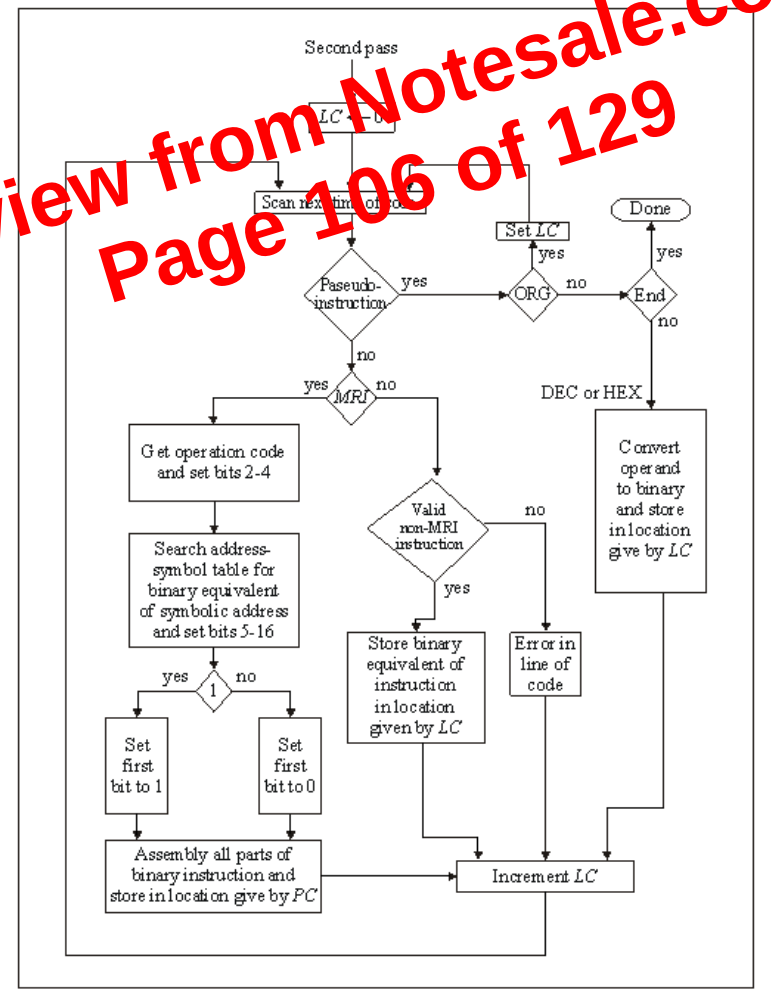
D_7'I'T_3 : Execute an input-output instruction

When a memory-reference instruction with I = 0 is encountered, it is not necessary to do anything since the effective address is already in AR. However, the sequence counter SC must be incremented with D_7'I'T_3 = 1, so that the execution of the memory-reference instruction can be continued with timing variable T_4. A register-reference or input-output instruction can be executed with the clock associated with timing signal T_3. After the instruction is executed, SC is cleared to 0 and control returns to the fetch phase with T_o = 1.

Note that the sequence counter SC is either incremented or cleared to 0 with every positive clock transition. We will adopt the convention that if SC is incremented, we will not write the statement SC ← SC + 1, but it will be implied that the control goes to the next timing signal is sequence. When SC is to be cleared, we will include the statement SC ← 0.

Preview from Notesale.co.uk
Page 104 of 129





Preview from Notesale.co.uk
Page 106 of 129

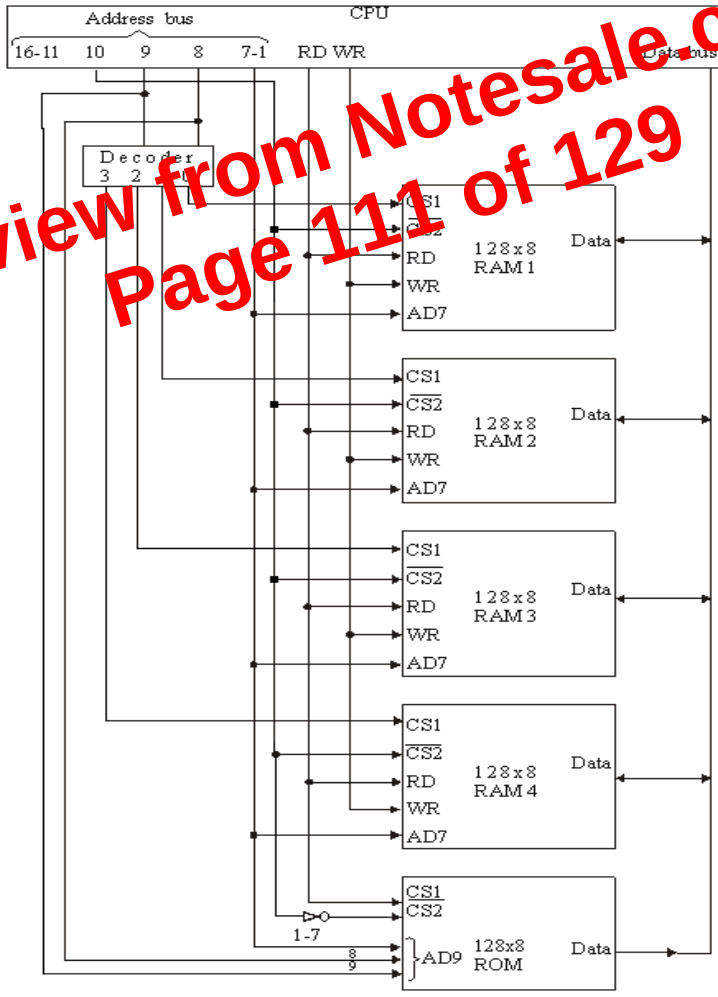
3. Calculate the effective address.
4. Fetch the operands from memory.
5. Execute the instruction.
6. Store the result in the proper place.

There are certain difficulties that will prevent the instruction pipeline from operating at its maximum rate. Different segments may take different times to operate on the incoming information. Some segments are skipped for certain operations. For example, a register mode instruction does not need an effective address calculation. Two or more segments may require memory access at the same time, causing one segment to wait until another is finished with the memory.
The design of an instruction pipeline will be most efficient if the instruction cycle is divided into segments of equal duration. The time that each step takes to fulfill its function depends on the instruction and the way it is executed.

Q. 108 Show the memory organization (1024 bytes) of a computer with four 128x8 RAM Chips and 512x8 ROM Chip. How many address lines are required to access memory.

Ans.

Memory organization:-

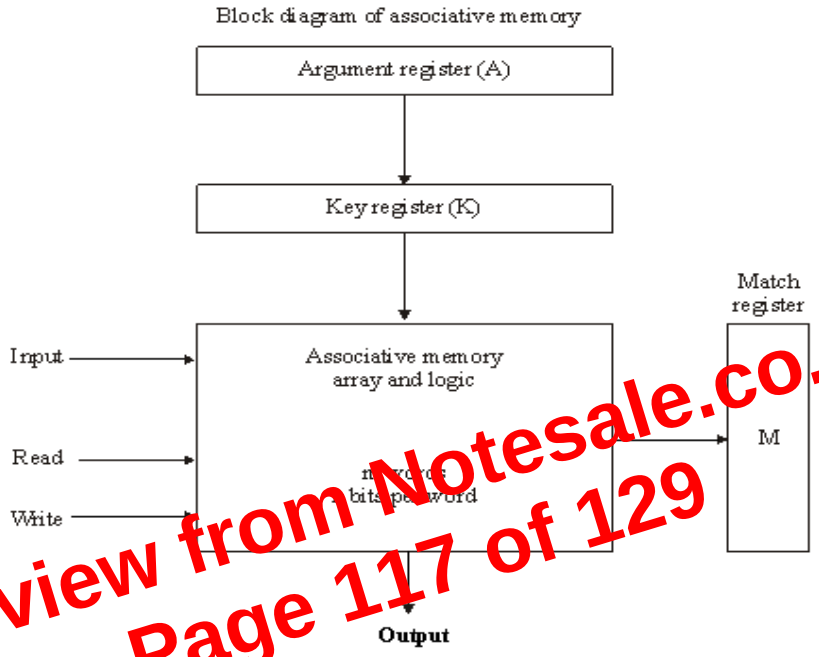


Preview from Notesale.co.uk
Page 111 of 129

Address lines:-

(iv) **Associative Memory**:-

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit access by the content of the data itself rather than by an address. A memory unit accessed by content is called an associative memory or content addressable memory (CAM) . This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.



Preview from Notesale.co.uk
Page 117 of 129

Q.112 a. Show the Truth Table's for the Following functions:-

i) $f(w, x, y, z) = w + x + y + z$

ii) $f(w, x, y, z) = wx + xz + \overline{y}$

Ans. (i) $f(w, x, y, z) = w + x + y + z$

w	x	y	z	output
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1