COURSE FILE

COMPUTER ORGANIZATION

IV B.Tech. - I Semester

Department of Computer Science and Engineering

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN
www.gvpcew.ac.in(Approved by AICTE & Affiliated to JNTUK)Estd. – 2008
Gandhi Nagar, Madhurawada, Visakhapatnam, Andhra Pradesh 530048.

Course Objectives & Outcomes

Course Objectives:--

- Comprehensive knowledge of computer system including the analysis and design of components of the system
- Understanding RTL, Micro operations, ALU, Organization of stored program computer, types of instructions and design of basic components of the system
- Illustration of data paths and control flow for sequencing in CPUs, Microprogramming of control unit of CPU
- Illustration of algorithms for basic arithmetic operations using binary and decimal representation
- Description of different parameters of a memory system, organization and mapping of various types of memories
- Describes the means of interaction devices with CPU, their characteristics, modes and introduction multiprocessors**.**

Course Outcomes:--

After completing this Course, the student should be able to:

Understand the architecture of a modern computer with its various processing units. Also the performance measurement of the computer system. In addition to this the management system of computer.

Students have a thorough understanding of the basic structure and operation of a digital computer.

- Able to discuss in detail the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
- Able to discuss in detail the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
- Students have a thorough understanding of Micro Program Control

- Students can calculate the effective address of an operand by addressing modes
- Understanding of how a computer performs arithmetic operation of positive and negative numbers.
- Explain the function of each element of a memory hierarchy, Cache memory and its importance.
- Students can understand how cache mapping occurs in computer and can solve various problems related to this.
- Study the hierarchical memory system including cache memories and virtual memory.
- Able to identify and compare different methods for computer I/O
- Able to discuss about advantages of parallel processing, multiprocessors.

1. Syllabus

UNIT-I

BASIC STRUCTURE OF COMPUTERS: Computer Types, Functional units, Basic operational concepts, Bus structures, Software, Performance, multiprocessors and multi computers. Data types, Complements, Data Representation. Fixed Point Representation. Floating – Point Representation.

Error Detection codes.

COMPUTER ARITHMETIC: Addition and subtraction, multiplication algorithms, Division Algorithms, Floating point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT-II

REGISTER TRANSFER LANGUAGE AND MICRO-OPERATIONS:

Register Transfer language. Register Transfer, Bus and memory transfer, Arithmetic Micro-operations, logic micro operations, shift micro-operations, Arithmetic logic shift unit. Instruction codes. Computer Registers Computer instructions – Instruction cycle. Memory Reference Instructions. Input Outputand Interrupt.

CENTRAL PROCESSING UNIT - Stack organization. Instruction formats. Addressing modes. DATA Transfer and manipulation. Program control. Reduced Instruction set computer

UNIT-III

MICRO PROGRAMMED CONTROL: Control memory, Address sequencing, micro program example, Design of control unit-Hard wired control. Micro programmed control.

UNIT-IV

THE MEMORY SYSTEM: Memory Hierarchy, Main memory, Auxiliary memory, Associative memory, Cache memory, Virtual memory, Memory management hardware

UNIT-V

INPUT-OUTPUT ORGANIZATION : Peripheral Devices, Input-Output Interface, Asynchronous data transfer Modes of Transfer, Priority Interrupt, Direct memory Access, Input –Output Processor (IOP), SerialCommunication;

UNIT-VI

PIPELINE AND VECTOR PROCESSING: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline Vector Processing, Array Processors. **Multi processors:** Characteristics of Multiprocessors, Interconnection Structures, Inter processor Arbitration. Inter processor Communication and Synchronization, Cache Coherence.

Text Books:

1. M. Moris Mano (2006), Computer System Architecture, 3rd edition, Pearson/PHI, India.
2. Carl Hamacher, ZvonksVranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India.

Reference Books:

1. Computer Organization Architecture- William Stallings (2006), 7th edition, PHI/PEARSON.
2. Computer Architecture and Organization-John P.Hayes ,McGraw Hill, International editions,2002.

Lecture Plan

Lecture no.	***Unit Number***	***Topic***
1.	I	Introduction to Computer architecture and Organization
2.	I	Computer Types, functional Units and Basic Operational Concepts, Bus structure, software, performance multi processor and multi computer
3.	I	Data Types ,Number systems, octal and hexadecimal nos, decimal and alphanumeric representation
4.	I	Complments-10's and 9's Complements; 2's ans 1's Complements; Subtraction of unsigned Nos; fixed point representation and Integer representation;
5.	I	Addition of two numbers in signed magnitude system; 2's complement addition, subtraction; overflow and overflow detection and decimal fixed point representation
6.	I	Floating point representation& Error detection codes
7.	I	Introduction to computer Arithmetic, addition and subtraction algorithm for signed magnitude numbers and hardware implementation; addition and subtraction for signed 2's complement data
8.	I	Multiplication algorithm: Hardware implementation for signed magnitude data, hardware & algorithm;
9.	I	Booth multiplication algorithm; Array multiplier
10.	I	Division Algorithm: Hardware implementation for signed magnitude data, hardware algorithm and other algorithm.
11.	I	Floating point Arithmetic operations: Basic considerations, register configurations, addition, subtraction, multiplication and division
12.	I	Decimal Arithmetic Unit: BCD Adder and BCD Subtraction
13.	I	Hardware and algorithm for Decimal Arithmetic Operations: Addition

		and subtraction.
14.	I	Hardware and algorithm for Decimal Arithmetic Operations: multiplication and division, floating point operations
15.	II	Register transfer Language, Register Transfer, Bus and Memory Transfer
16.	II	Arithmetic Micro operations; Logic Micro operations
17.	II	Shift micro operations, Arithmetic logic shift unit
18.	II	Instruction Codes, Computer register
19.	II	Computer Instruction and format, Instruction Cycle
20.	II	Memory Reference Instructions, Input-Output and Interrupt
21.	II	Stack Organization, register Stack, memory stack, Reverse polish notation, Conversion to RPN, Evaluation of arithmetic expression
22.	II	Instruction Formats, 3-types of CPU organization, 3,2,1,0 address instruction formats, RISC Instructions
23.	II	Addressing modes; Data Transfer and manipulation instructions
24.	II	Program Control Instructions, Subroutine call and return instructions, Program interrupts and types of interrupts
25.	II	RISC Vs CISC; Overlapping Register windows; Berkeley RISCI
26.	III	Control memory
27.	III	Address Sequencing
28.	III	Micro Program Example: computer configuration, instruction format and symbolic microinstructions; The fetch routine, symbolic microprogram and binary microprogram
29.	III	Design of Control Unit: Hard wired control.and Micro programmed control
30.	IV	Memory Hierarchy, Main memory: RAM and ROM chips, memory address map, memory connection to CPU; Auxiliary memory: Magnetic Disks and Magnetic Tapes

31.	IV	Associative Memory: Hardware Organization, Match Logic, read operation and write operation
32.	IV	Direct mapping, Set-Associative Mapping, Writing into cache and cache initialization
33.	IV	Virtual memory: Address Space and Memory Space, Address Mapping using pages; Associative memory page table, Page replacement
34.	IV	Memory management hardware: Segmented page mapping, Numerical example, Memory Protection
35.	V	Peripheral Devices, I/O Interface: I/O bus and interface modules; I/O vs Memory Bus, Isolated vs memory mapped I/O, Examples of I/O Interface
36.	V	Asynchronous Data Transfer: Strobe Control, handshaking, asynchronous serial transfer; Asynchronous Commn Interface, First-in-first out buffer
37.	V	Modes of Transfer: Examples of Programmed I/O, Interrupt initiated I/O software considerations; Priority Interrupt: Daisy Chaining Priority, parallel priority Interrupt, priority encoder,
38.	V	Interrupt cycle, software routines and initial and final operations, Direct Memory Access: DMA Controller and DMA Transfer
39.	V	Input Output Processor(IOP): CPU –IOP Communication,
40.	V	IBM 370 I/o channel, Intel 8089 IOP
41.	V	Serial Communication: Character oriented protocol,
42.	V	Transmission example, Data transparency and Bit-oriented protocol
43.	VI	Parallel Processing
44.	VI	Pipelining, general considerations
45.	VI	Arithmetic Pipeline, Instruction Pipeline
46.	VI	RISC Pipeline; Vector Processing: vector operations, matrix multiplication, memory interleaving,
47.	VI	superscalar processors and super computers

48.	VI	Array Processors: Attached Array Processor, SIMD Array Processor
49.	VI	Characteristics of multi processors
50.	VI	Interconnection Structure: Timeshared Common Bus, Multiport Memory, Crossbar Switch, Multistage switching network, Hypercube Interconnection
51.	VI	Inter processor arbitration: serial arbitration procedure, parallel arbitration logic and dynamic arbitration algorithms
52.	VI	Inter processor communication and synchronization, cache coherence

Unit – I

Basic structure of computers and Computer arithmetic

1.1.1. Unit Objectives:

After reading this Unit, the reader should be able to understand:

- The definition of computer architecture, organization and computer hardware.
- The design aspects of computer hardware and software.
- Functions provided by a digital computer and functional units of a digital computer.
- Processes involved in executing a task, instruction types, connections between memory and processor and data transfer mechanism between memory and processor.
- Processor registers and their functions interrupt and interrupt service routines.
- Bus, single bus, buffer registers, functions of system software and multiprogramming, processor performance and performance parameters.
- Multi Processors and multi computers.
- Data types and their representation, number system and their representation.
- Using complements for subtraction
- Fixed point and floating point representation of signed numbers
- Error generation , error detection and error correction codes

1.1.2. Unit Outcomes:

- Student is able to differentiate the various computer types, hardware Vs software, RISC Vs CISC.
- Student is able to give an outline of functional parts of the computer.
- Student is able to explain basic operational concept of a computer, bus structure & software.
- Student is able to bring out the various performance parameters of a computer
- Student is able to give representation of various data types in Binary Coded format.
- Student is able to do arithmetic operations on signed, fixed point & floating point numbers using complements.
- Student is able to develop algorithms for arithmetic operations in fixed point & floating point and decimal numbers and design hardware circuits for them.
- Student is able to make error detection codes and build error detection circuits.

Lectures for the concerned Unit:

Lecture1


lecture-1.docx

Lecture-2


lecture-2.pptx

Lecture-3


lecture-3'.pptx

Lecture-4


lecture-4.pptx

Lecture-5

Characteristics of floating point numbers:

The characteristics are 1. Precision 2. Gap 3. Range

Precision:

It characterizes how precise a floating point value can be. I**t is defined as the number of bits in the significand**. The greater the number of bits in the significand, the greater is the CPU's precision and the more precise is it's value. Many CPUs have 2 representations for floating point numbers. They are called single precision and double precision here double precision has twice the number of bits.

Gap:

The gap is the difference between two adjacent values. It's value depends on the value of the exponent.

Take the number: X = .10111010 * 2^3 .

It's adjacent values are : .10111001 * 2^3 and .10111011* 2^3 .

Each number produce a gap of .00000001* 2^3 .

In general the gap for floating point value X can be expressed as $\mathbf{2^{(Xe\text{-}precision)}}$

Range:

The range of a floating point representation is bounded by it's smallest and largest possible values.

Overflow and underflow;

Overflow occurs when an operation produces a result that can not be stored in computers's floating point registers. Underflow occurs when an operation produces a result between zero and either the positive or negative smallest possible value.

IEEE 754 Floating point standard:

This standard specifies 2 precision for floating point numbers which are called single precision and double precision floating point representations.

Single Precision Format:

This format has 32 bits. 1 bit for sign; 8 bits for the exponent; 23 for the significand. The significand also includes an implied 1 to the left of its radix point(except for special values and denormalized numbers).

Floating point representation.docx

Error Detecting codes.docx

Error Detection codes:

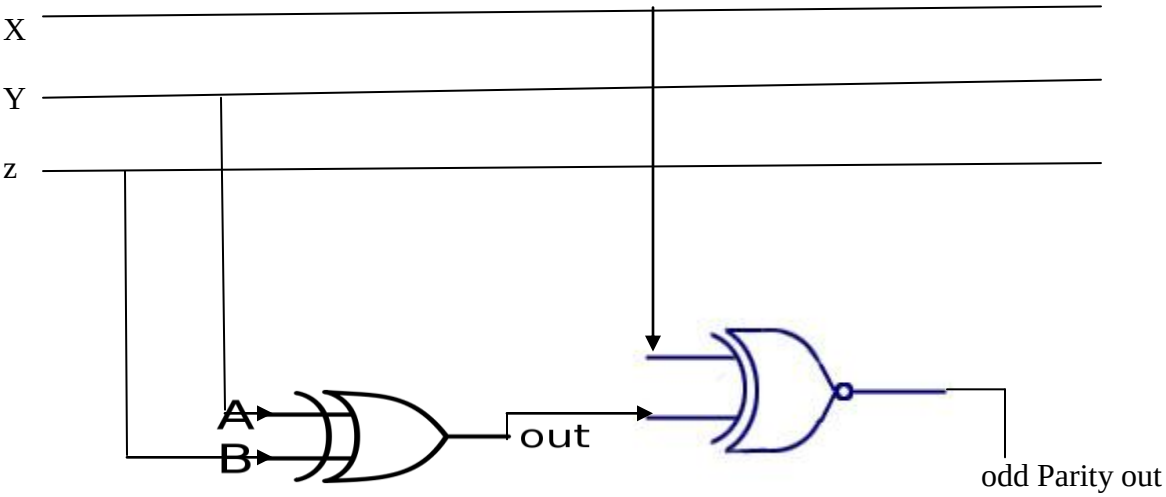
Information is stored as binary codes and are transmitted by serial or parallel communication. During transmission noise is added to the signal and it may change binary bits in the code from 1 to 0, and vice versa. An error detection code is a **binary code that detects digital errors during transmission.** The detected errors can not be corrected but their presence is indicated.

Parity bit:

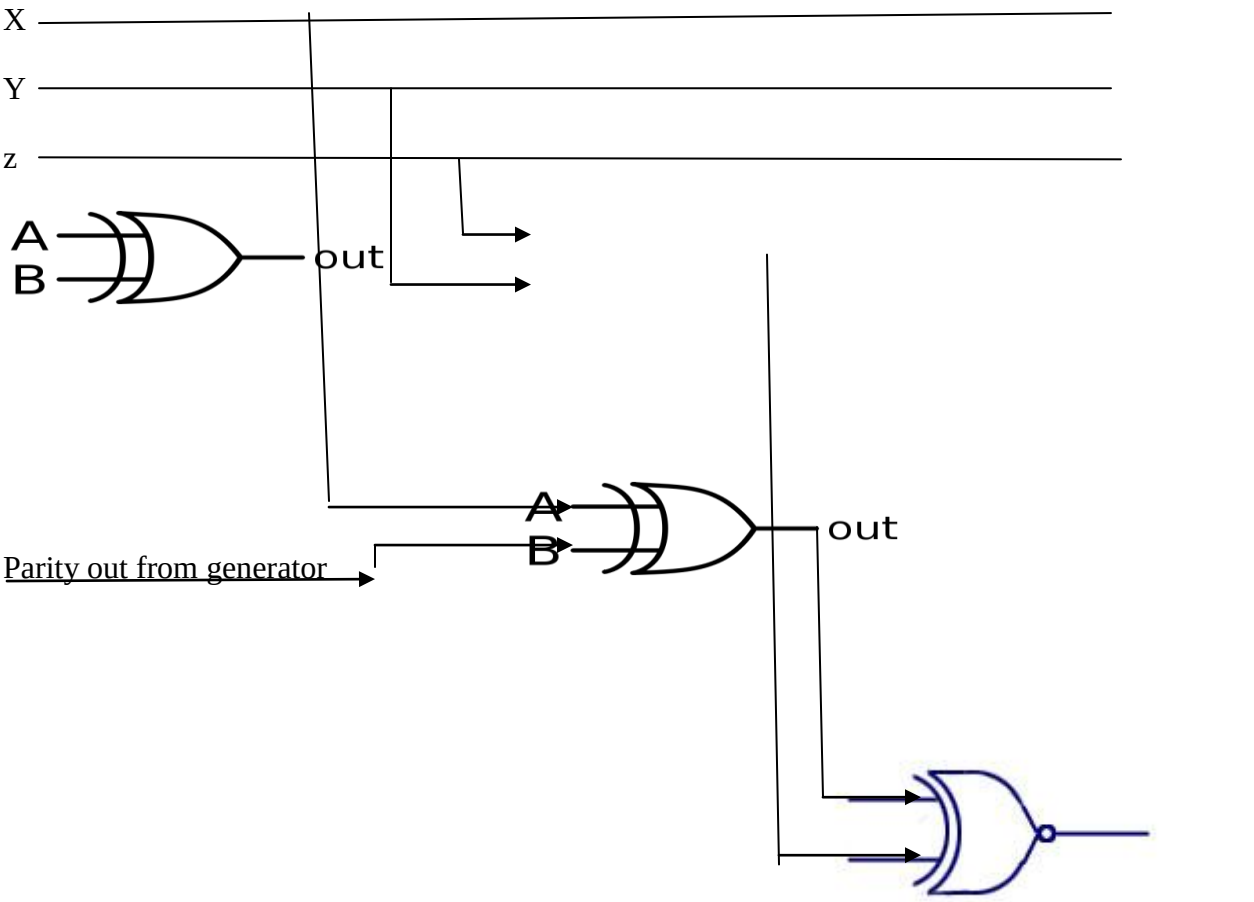
The most common error detection code used is the parity bit. A parity bit is an extra bit included with a binary message to make the total number of 1's either odd or even. If the message consists of n bits , then the error detection code consists of n+1 bits. If the bit added to the message makes the sum of 1's odd in the error detection code, then the scheme is called odd-parity. If the sum of bits is even , the scheme is called even parity scheme.

Message xyz	P(odd)	P(even)	Error detection code, odd parity	Error detection code, even parity
000	1	0	0001	0000
001	0	1	0010	0011
010	0	1	0100	0101
011	1	0	0111	0110
100	0	1	1000	1001
101	1	0	1011	1011
110	1	0	1101	1101
111	0	1	1110	1111

Parity Generator and Parity Checker:



Parity Checker:



The circuit arrangement checks the occurrence of error any odd number of times. An even number of errors is not detected.

We note that P(even) function is the exclusive –OR x,y,z because it is equal to 1 when either one or all 3 of the variables are equal to 1. The P(odd) function is the complement of the P(even) function.

Assume at the sending end the message bits and odd parity bit is generated. The EX-OR gates generate P(even) function and to generate P(odd), the complement of P(even) is used.

The 4 bits transmitted has an odd number of I's. If an error occurs during transmission, then the number of 1's become even. Hence parity checker checks for even parity.

Lecture-6

COMPUTER ARITHMETIC:

Addition, subtraction, multiplication are the four basic arithmetic operations. Using these operations other arithmetic functions can be formulated and scientific problems can be solved by numerical analysis methods.

Arithmetic Processor:

It is the part of a processor unit that executes arithmetic operations. The arithmetic instructions definitions specify the data type that should be present in the registers used . The arithmetic instruction may specify binary or decimal data and in each case the data may be in fixed-point or floating point form.

Fixed point numbers may represent integers or fractions. The negative numbers may be in signed-magnitude or signed- complement representation. The arithmetic processor is very simple if only a binary fixed point add instruction is included. It would be more complicated if it includes all four arithmetic operations for binary and decimal data in fixed and floating point representations.

Algorithm:

Algorithm can be defined as a finite number of well defined procedural steps to solve a problem. Usually, an algorithm will contain a number of procedural steps which are dependent on results of previous steps. A convenient method for presenting an algorithm is a flowchart which consists of rectangular and diamond –shaped boxes. The computational steps are specified in the

rectangular boxes and the decision steps are indicated inside diamond-shaped boxes from which 2 or more alternate path emerge.

Addition and Subtraction:

3 ways of representing negative fixed point binary numbers:

1. Signed-magnitude representation---- used for the representation of mantissa for floating point operations by most computers.
2. Signed-1's complement
3. Signed -2's complement—Most computers use this form for performing arithmetic operation with integers

Addition and subtraction algorithm for signed-magnitude data

Let the magnitude of two numbers be A & B. When signed numbers are added or subtracted, there are 4 different conditions to be considered for each addition and subtraction depending on the sign of the numbers. The conditions are listed in the table below. The table shows the operation to be performed with magnitude(addition or subtraction) are indicated for different conditions.

Sl.No	Operation	Add Magnitudes	Subtract magnitudes		
			When A> B	When A< B	When A=B
1	(+A) + (+B)	+ (A + B)			
2	(+A) + (-B)		+(A-B)	-(B-A)	+(A-B)
3	(-A) + (+B)		-(A-B)	+(B-A)	+(A-B)
4	(-A) + (-B)	- (A + B)			
5	(+A) - (+B)		+(A-B)	-(B-A)	+(A-B)
6	(+A) - (-B)	+ (A + B)			
7	(-A) - (+B)	- (A + B)			
8	(-A) - (-B)		-(A-B)	+(B-A)	+(A-B)

The last column is needed to prevent a negative zero. In other words, when two equal numbers are subtracted, the result should be +0 not -0.

The algorithm for addition and subtraction (from the table above):

Addition Algorithm:

When the signs of A and B are identical, add two magnitudes and attach the sign of A to the result. When the sign of A and B are different, compare the magnitudes and subtract the smaller number from the larger. Choose the sign of the result to be the same as A if A>B or the complement of sign of A if A < B. If the two magnitudes are equal, subtract B from A and make te sign of the result positive.

Subtraction algorithm:

When the signs of A and B are different, add two magnitudes and attach the sign of A to the result. When the sign of A and B are identical, compare the magnitudes and subtract the smaller number from the larger. Choose the sign of the result to be the same as A if A>B or the complement of sign of A if A < B. If the two magnitudes are equal, subtract B from A and make te sign of the result positive.

Hardware Implementation:

Let A and B are two registers that hold the numbers.

A_S and B_S are 2, flip-flops that hold sign of corresponding numbers. The result is stored In A and A_S .and thus they form Accumulator register.

We need to perform micro operation, A+ B and hence a parallel adder.

A comparator is needed to establish if A> B, A=B, or A<B.

We need to perform micro operations A-B and B-A and hence two parallel subtractor.

An exclusive OR gate can be used to determine the sign relationship, that is, equal or not.

Thus the hardware components required are a magnitude comparator, an adder, and two subtractors.

Reduction of hardware by using different procedure:

1. We know subtraction can be done by complement and add.
2. The result of comparison can be determined from the end carry after the subtraction. We find An adder and a complementer can do subtraction and comparison if 2's complement is used for subtraction.

Hardware forsigned-magnitude addition and subtraction:

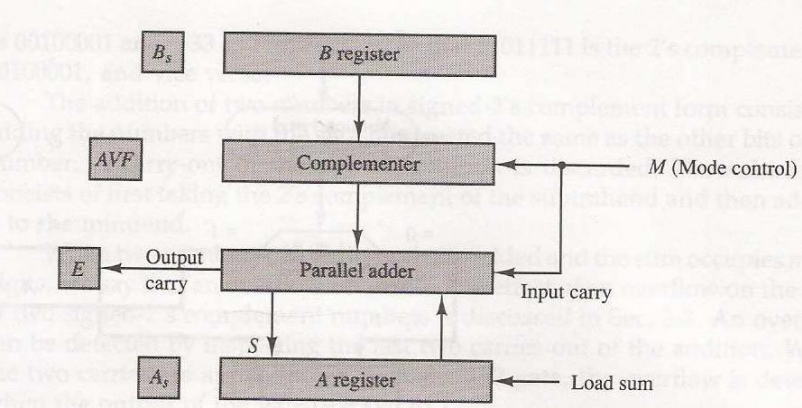


Figure 10-1 Hardware for signed-magnitude addition and subtraction.

AVF Add overflow flip flop. It hold the overflow bit when A & B are added.

Flip flop E—Output carry is transferred to E. It can be checked to see the relative magnitudes of the two numbers.

A-B = A +(-B)= Adding a and 2's complement of B.

The A register provides other micro operations that may be needed when the sequence of steps in the algorithm is specified.

The complementer Passes the contents of B or the complement of B to the Parallel Adder depending on the state of the mode control B. It consists of EX-OR gates and the parallel adder consists of full adder circuits. The M signal is also applied to the input carry of the adder.

When input carry M=0, the sum of full adder is A +B. When M=1, S = A + B' +1= A – B

Hardware algorithm:

Flow Chart for Add and Subtract operations:

The EX-OR gate provides 0 as output when the signs are identical. It is 1 when the signs are different.

A + B is computed for the following and the sum is stored in EA:

1. When the signs are same and addition operation is required.
2. When the signs are different and subtract operation is required.

   The carry in E after addition indicates an overflow if it is 1 and it is transferred to AVF, the addoverflow flag

   A-B = A+ B'+1 computed for the following:

1. When the signs are different and addition operation is required.
2. When the signs are same and subtract operation is required.
   No overflow can occur if the numbers are subtracted and hence AVF is cleared to Zero.

   [the subtraction of 2 n-digit un signed numbers M-N (N≠0) in base r can be done as follows:
   1. Add minuend M to thee r's complement of the subtrahend N. This performs M-N $+r^n$.
   2. If M ≥ N, The sum will produce an end carry r^nwhich is discarded, and what is left is the result M-N.
   3. If M< N, the sum does not produce an end carry and is equal to r^n–(N-M), which is the r's complement of the sum and place a negative sign in front.]

   A 1 in E indicates that A ≥ B and the number in A is the correct result.
   If this number in A is zero, the sign A_S must be made positive to avoid a negative zero.
   A 0 in E indicates that A< B. For this case it is necessary to take the 2's complement of the value in A.
   In the algorithm shown in flow chart, it is assumed that A register has circuits for micro operations complement and increment. Hence two complement of value in A is obtained in 2, micro operations. In other paths of the flow chart , the sign of the result is the same as the sign of A, so no change in A_S is required.
   However When A < B, the sign of the result is the complement of original sign of A.
   Hence The complement of A_S stored in A_S.
   Final Result: A_S A
   Flow chart for ADD and Subtract operations:

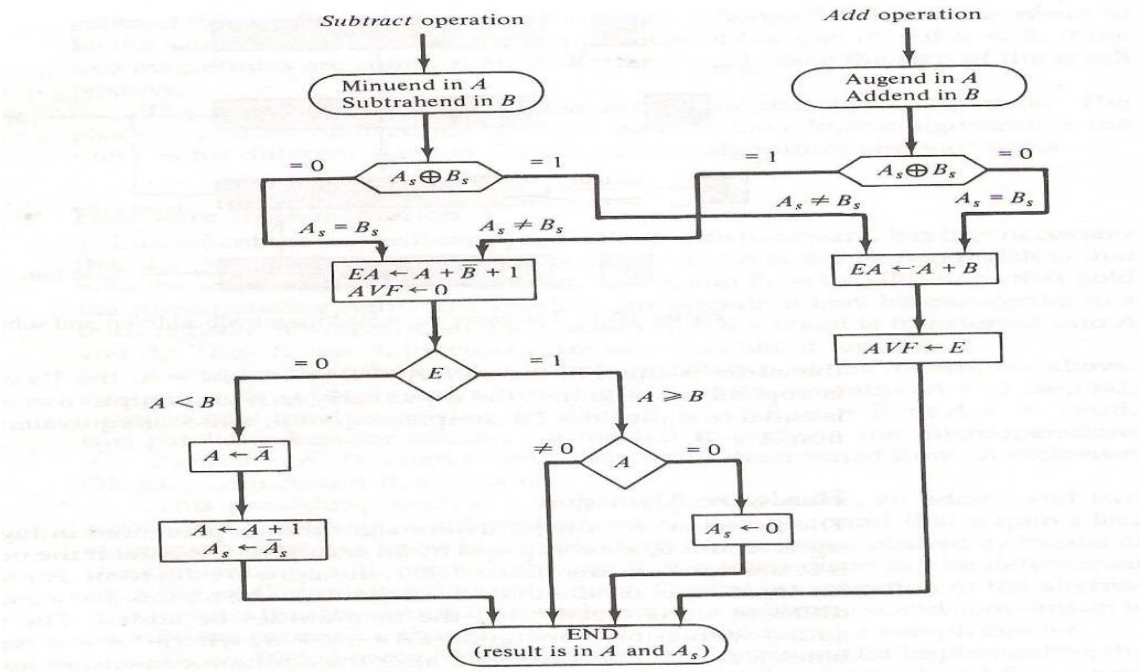


Figure 10-2 Flowchart for add and subtract operations.

Addition and Subtraction with signed-2's complement Data.:

Arithmetic Addition:

This method does not need a comparison or subtraction but only addition and complementation. The procedure is as below:

1. **Represent the negative numbers in 2's complement form.**
2. **Add the two numbers including the sign bits and discard any carry out of sign bit position.**
3. **The overflow bit V is set to 1 if there is a carry into sign bit and no carry out of sign bit or if there is a no carry into sign bit and a carry out of sign bit. Otherwise it is set to zero.**
4. **If the result is negative, take the 2's complement of the result to get a correct negative result.**

Arithmetic Subtraction:

1. **Represent the negative numbers in 2's complement form.**
2. **Take the 2's complement of the subtrahend including the sign bit and add it to the minuend including the sign bit.**

3. **The overflow bit V is set to 1 if there is a carry into sign bit and no carry out of sign bit or if there is a no carry into sign bit and a carry out of sign bit. Otherwise it is set to zero.**
4. **Discard the carry out of the sign bit position.**

Note: A subtraction operation can be changed to an addition operation if the sign of the subtrahend is changed.

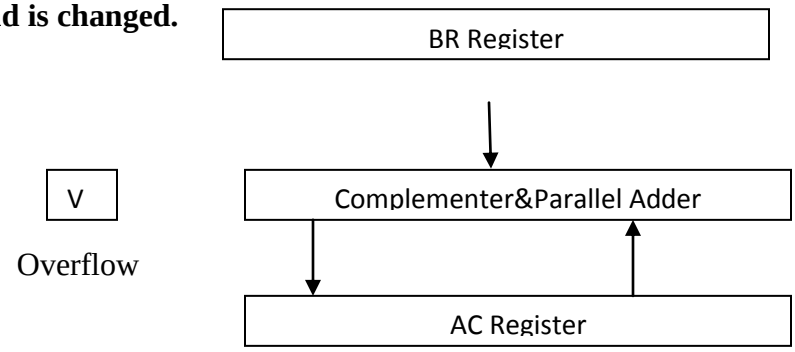


Fig: Hardware for Signed 2/s complement for addition/ subtractioin.

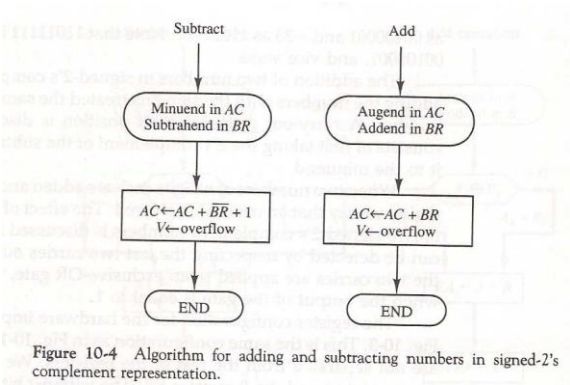


Figure 10-4 Algorithm for adding and subtracting numbers in signed-2's complement representation.

lecture-6.docx

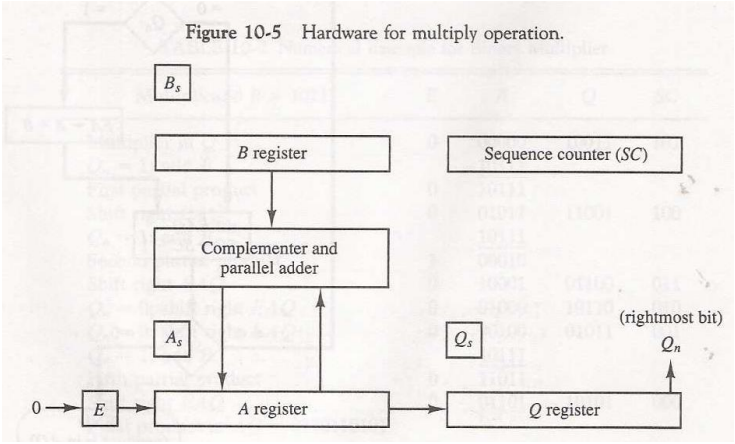
Lecture-7

Multiplication Algorithm:

Hardware implementation of multiplication of numbers in signed – magnitude form:

1. A adder is provided to add two binary numbers and the partial product is accumulated in a register.
2. Instead of shifting the multiplicand to the left, the partial product is shifted to the right, which result in leaving the partial product and the multiplicand in the required relative positions.
3. When the corresponding bit of the multiplier is zero, there is no need to add all zeros to the partial product, since it will not alter it's value.

The hardware consists of 4 flipflops, 3 registers, one sequence counter , an adder and complementer.



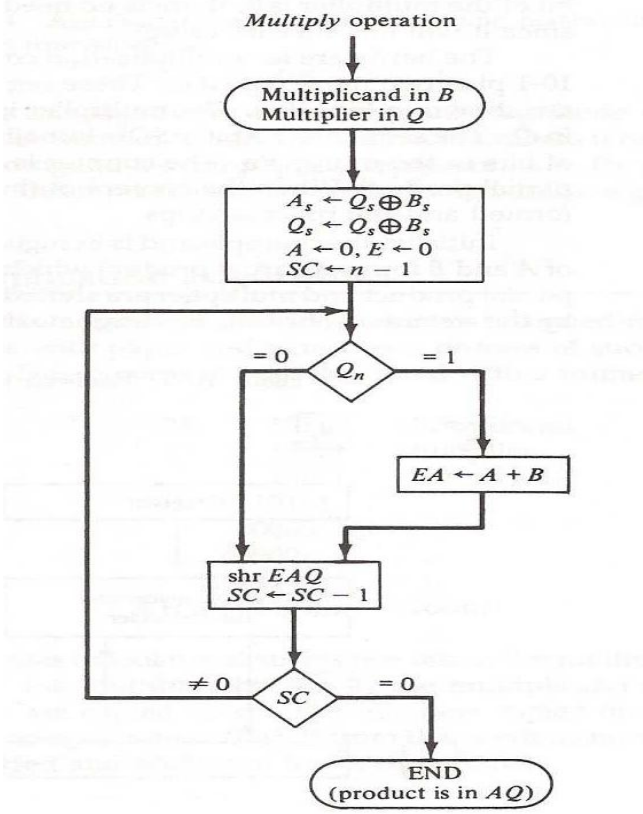
Q register&Q_S flip flop : contains multiplier & Its sign
Sequence counter : It is set to a value equal to the number of bits in the multiplier
B Register& B_S flipflop : It contains the multiplicand,& its sign
A Register, E Flip flop : Initialized to ' 0'. A_S denotes sign of partial product
EA Register : hold partial product, with carry generated in addition being shifted to E .
Qn : Rightmost bit of the multiplier; AQ : will contain the final product.

As **AQ** represent product register, both A_S $Q_S$represent the sign of the partial product or product. The number to be multiplied are stores in memory as n bit sign magnitude numbers and when transferred to register msb bit go to sign flipflop and remaining n-1 bits go to registers. Hence SC is initially set to n-1.
Let the lower order bit of the multiplier in $Q_n$tested.
If it is 1, the multiplicand in B is added to the present partial product in A.
If it is a '0', nothing is done. Register EAQ is then shifted once to the right to form the new partial product. The sequence counter is decremented by 1 and it's new value checked. If it is not equal to zero, the process is repeated and a new partial product is formed. The process stops when SC = 0.

The final product is available in both A and Q, with A holding the most significant bits and Q holding the least significant bits.

Flowchart for multiply operation:

Figure 10-6 Flowchart for multiply operation.



Numerical Example for the above algorithm:

Multiplicand B= 10111	E	A	Q	SC
Multiplier in Q	0	00000	10011	101
Q_n =1;add B		10111		
First Partial Product	0	10111		
Shift Right EAQ	0	01011	11001	100
Q_n =1;add B		10111		
Second Partial Product	1	00010		

Shift Right EAQ	0	10001	01100	011
Q_n =0; Shift Right EAQ	0	01000	10110	010
Q_n =0; Shift Right EAQ	0	00100	01011	001
Q_n =1;add B Fifth Partial Product Shift Right EAQ	 0 0	10111 11011 01101	 10101	 000
Final Product in AQ AQ = 0110110101				


lecture-7.docx

Lecture-8

Booth Multiplication Algorithm:

Multiplication of signed- 2's complement integers:

This algorithm uses the following facts.

1. A string of 0's in the multiplier requires no addition but just shifting.
2. A string of 1's in the multiplier from bit weight 2^k to weight 2^m can be treated as 2^{k+1} - $2^{m.}$

Example: Consider the binary number: 001110(+14)

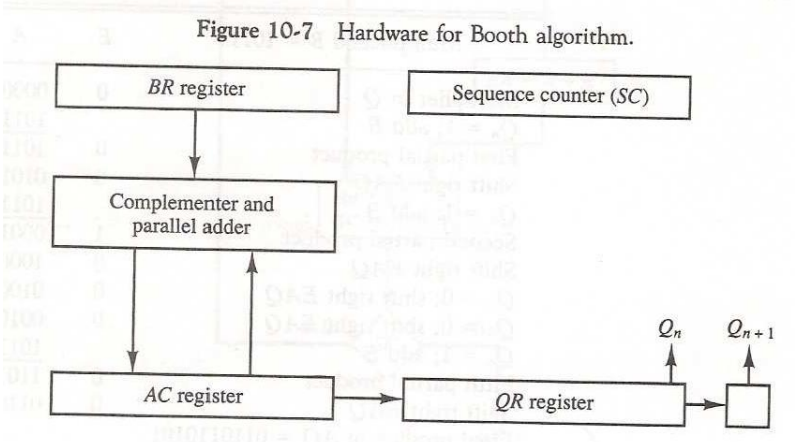
The number has a string of 1's from 2^3 to 2^1. Hence k = 3 and m= 1. As other bits are 0's, the number can be represented as 2^{k+1} - 2^m = $2^4 - 2^1$ = 16-2 = 14. Therefore the multiplication M * 14 , where M is the multiplicand and 14 the multiplier can be done as Mx 2^4 –M x $2^{1.}$

This can be achieved by shifting binary multiplicand M four times to the left and subtracting M shifted left once which is equal to (Mx 2^4 –M x $2^{1.}$).

Shifting and addition/subtraction rules for multiplicand in Booth's Algorithm:

1. The multiplicand is subtracted from the partial product upon encountering the first least significand 1 in a string of I's in the multiplier.
2. The multiplicand is added to the partial product upon encountering the first 0 (provided that there was a previous 1)in a string of 0's in the multiplier.
3. The partial product does not change when the multiplier bit is identical to the previous multiplier bit

Hardware Implementation of Booth Algorithm:



Note: Sign bit is not separated from register. QR register contains the multiplier register and $Q_n$represent the least significant bit of the multiplier in QR. Q_{n+1} is an extra flip flop appended to QR to facilitate a double bit inspection of the multiplier.

AC register and appended Q_{n+1} are initially cleared to 0.

Sequence counter Sc is set to the number n which is equal to the number of bits of bits In the multiplier.

Q_nQ_{n+1} are to successive bits in the multiplier

Example for multiplication using Boot h algorithm:

Q_nQ_{n+1}	BR = 1011 ,BR'+1 = 01001	AC	QR	Q_{n+1}	SC
10	Initial Subtract BR ashr	00000 01001 01001 00100	10011 11001	0 1	101 100
11	ashr	00010	01100	1	011
01	Add BR ashr	10111 11001 11100	 10110	 0	 010
00	ashr	11110	01011	0	001

10	Subtract BR Ashr	01001 00111 00011	10101	1	000

Algorithm in flowchart for multiplication of signed 2's complement numbers.

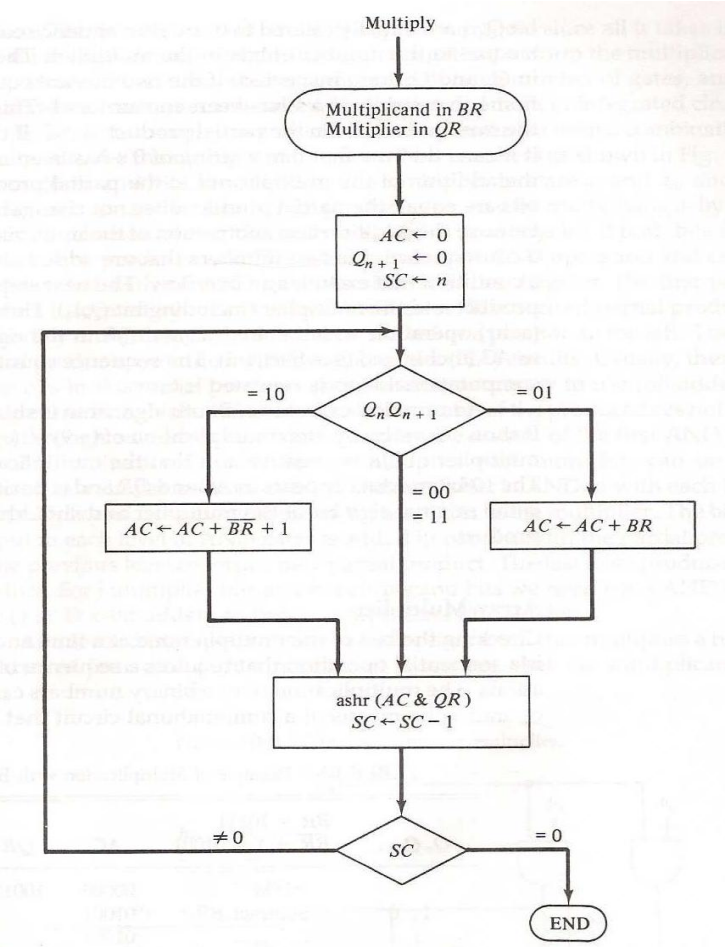
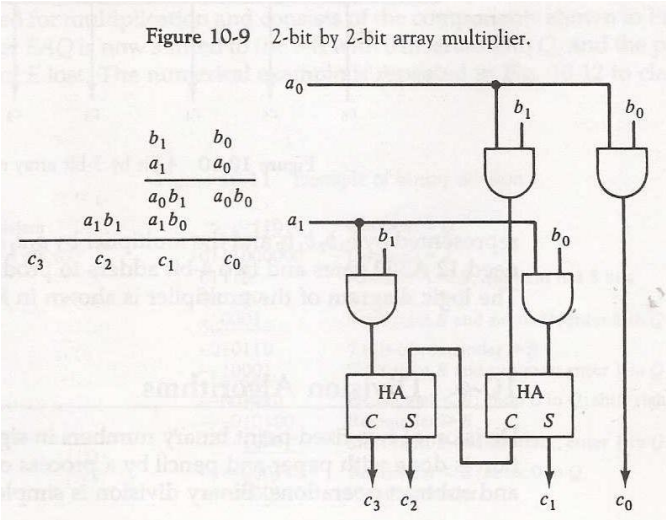


Figure 10-8 Booth algorithm for multiplication of signed-2's complement numbers.

Array Multiplier:

2 -bit by 2- bit Array Multiplier:

Multiplicand bits are b_1 and b_0 .Multiplier bits are a_1 and a_0 .The first partial product is obtained by multiplying a_0 by $b_1 b_0$. The bit multiplication is implemented by AND gate. First partial product is made by two AND gates. Second partial product is made by two AND gates. The two partial products are added with two half adder circuits.



Combinational circuit binary multiplier:

A bit of the multiplier is ANDed with each bit of the multiplicand in as many levels as there bits in the multiplier. The binary output in each level of the AND gates is added in parallel with the partial product of the previous level to form a ne partial product. The last level produces the product. For j multiplier and k multiplicand bits, we need j*k AND Gates and (j-1)*k bit adders to ptoduce a product of j+k bits.

4- bit by 3-bit Array Multiplier:

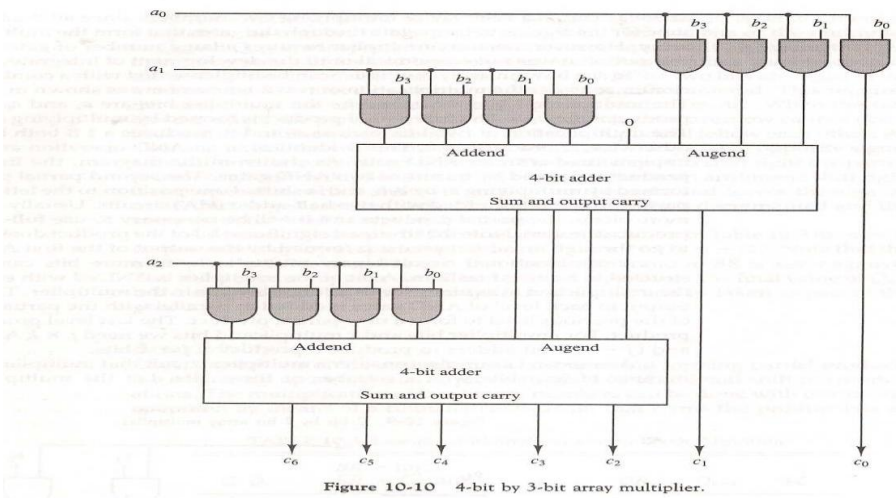


Figure 10-10 4-bit by 3-bit array multiplier.

Lecture-9

Division Algorithms:

Division Process for division of fixed point binary number in signed –magnitude representation:

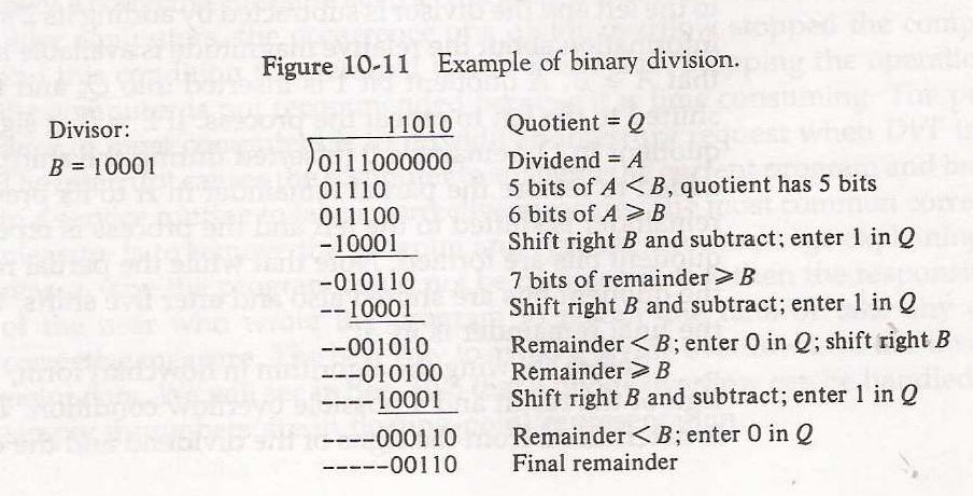


Figure 10-11 Example of binary division.

Divisor:	11010	Quotient = Q
B = 10001	)0111000000	Dividend = A
	01110	5 bits of $A < B$, quotient has 5 bits
	011100	6 bits of $A \geqslant B$
	-10001	Shift right B and subtract; enter 1 in Q
	-010110	7 bits of remainder $\geqslant B$
	--10001	Shift right B and subtract; enter 1 in Q
	--001010	Remainder $< B$; enter 0 in Q; shift right B
	---010100	Remainder $\geqslant B$
	----10001	Shift right B and subtract; enter 1 in Q
	----000110	Remainder $< B$; enter 0 in Q
	-----00110	Final remainder

Let dividend A consists of 10 bits and divisor B consists of 5 bits.

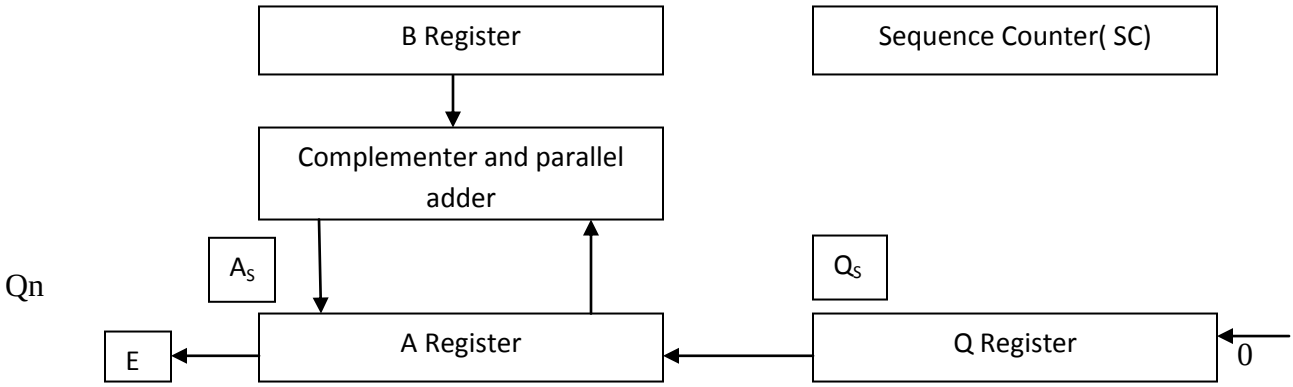
1. Compare the 5 most significant bits of the dividend with that of divisor.
2. If the 5 bit number is smaller than divisor B, then take 6 bits of the dividend and compare with the 5 bit divisor.
3. The 6 bit number is greater than divisor B. Hence place a 1 for the quotient bit in the sixth position above the dividend. Shift the divisor once to the right and subtracted from the dividend. The difference is called partial remainder.
4. Repeat the process with the partial remainder and divisor. If the partial remainder is equal or greater than or equal to the divisor, the quotient bit is equal to 1.The divisor is then shifted right and subtracted from the partial remainder. If the partial remainder is small than the divisor, then the quotient bit is zero and no subtraction is needed. The divisor is shifted once to the right in any case,.

Hardware Implementation of division for signed magnitude fixed point numbers:

To implement division using a digital computer, the process is changed slightly for convenience.

1. Instead of shifting the divisor to the right, the dividend or the partial remainder, is shifted to the left so as to leave the two numbers in the required relative position.
2. Subtraction may be achieved by adding A (dividend)to the 2's complement of B(divisor). The information about the relative magnitude is then available from end carry.

3. Register EAQ is now shifted to the left with 0 inserted into Q_n and the previous value of E is lost..
4. The divisor is stored in B register and the double length dividend is stored in registers A and Q.
5. The dividend is shifted to the left and the divisor is subtracted by adding it's 2's complement value.
6. If E= 1, it signifies that A ≥ B.A quotient bit is inserted into $Q_n$and the partial remainder is shifted to the left to repeat the process.
7. If E = 0, it signifies that A < B so the quotient Q_n remains 0(inserted during the shift). The value of B is then added to restore the partial remainder in A to its previous value. The partial remainder is shifted to the left and the process is repeated again until all 5 quotient bits are formed.
8. At the end Q contains the quotient and A the remainder. If the sign of dividend and divisor are alike, the quotient is positive and if unalike, it is negative. The sign of the remainder is the same as dividend.



Hardware for implementing division of fixed point signed- Magnitude Numbers

Example of Binary division with digital hardware: Divisor B = $\overline{1}$0001, B + 1 = 01111

		E	**A**	**Q**	**SC**
	Dividend:		01110	00000	5
	Shl EAQ		11100	00000	
	Add , $\overline{\mathbf{B}}$ **+ 1**		01111		
	E = 1	1	01011		
	Set Q_n= 1	1	01011	00001	4
	Shl EAQ	0	10110	00010	

	Add , **B + 1**		01111 _____		
	E = 1	1	00101		
	Set Q_n= 1	1	00101	00011	3
	Shl EAQ	0	01010	00110	
	Add , **B + 1**		01111 _____		
	E= 0; Leave Q_n= 0	0	11001	00110	
	Add B		10001 _____		
	Restore remainder	1	01010		2
	Shl EAQ	0	10100	01100	
	Add , **B + 1**		01111 _____		
	E = 1	1	00011		
	Set Q_n= 1	1	00011	01101	1
	Shl EAQ	0	00110	11010	
	Add , **B + 1**		01111 _____		
	E= 0; Leave Q_n= 0	0	10101	11010	
	Add B		10001 _____		
	Restore remainder	1	00110	11010	0
	Neglect E				

	Remainder in A		00110	11010	
	Quotient in Q				

Divide overflow:

When the dividend is twice as long as the divisor, the condition for overflow can be stated as follows:

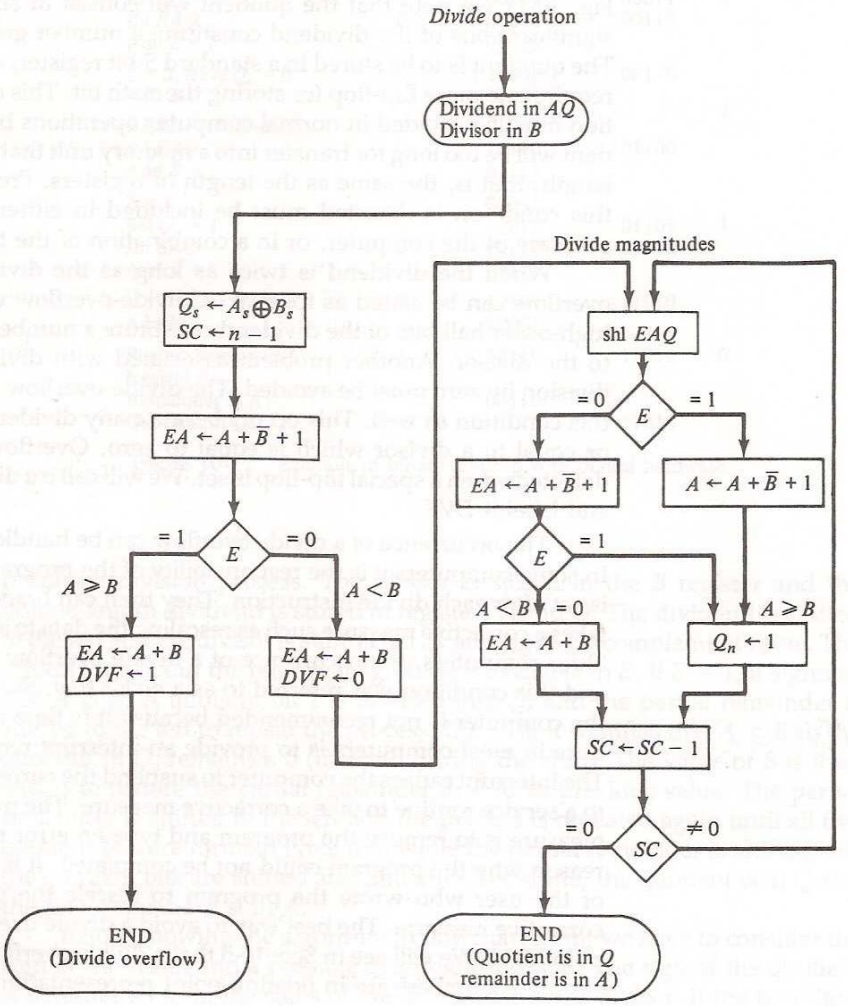
A divide-overflow condition occurs if the higher order half bits of the dividend constitute a number greater than or equal to the divisor. If the divisor is zero, then the dividend will definitely be greater than or equal to divisor. Hence divide overflow condition occurs and hence the divide-overflow –flip flop will be set. Let the flip flop be called DVF.

Handling DVF:

1. Check if DVF is set after each divide instruction. If DVF is set, then the program branches to a subroutine that takes corrective measures such as rescaling the data to avoid overflow.
2. An interrupt is generated if DVF is set. The interrupt causes the processor to suspend the current program and branch to interrupt service routine to take corrective measure. The most common corrective measure is to remove the program and type an error message that explains the reasons.
3. The divide overflow can be handled very simply if the numbers are represented in floating point representation.

Flow chart for divide operation:

Figure 10-13 Flowchart for divide operation.



Assumption:

Operands are transferred from memory to registers as n bit words.n-1 bit form magnitude and 1 bit shows the sign.

A divide overflow condition is tested by subtracting the divisor in B from half of the bits of dividend stored in A. If vA ≥ B, the DVF is set and the operation is terminated prematurely. If A < B, no DVF occurs and so the value of dividend is restored by adding B to A.

The division of the magnitudes starts by shifting the dividend in AQ to the left, with the higher order bit shifted into E. If the bit shifted into E is 1, we know that EA is greater than B because EA consists of a 1 followed by n-1 bits while B consists of only n-1 bits. In this case, B must be subtracted from EA and 1 inserted into Q_n for the quotient bit. Since register A is missing the

higher order bit of the dividend (which is in E), it's value is $EA - 2^{n-1}$. Adding to this value the 2's complement of B results in

$(EA-2^{n-1}) + (2^{n-1} - B) = E-B$. The carry from the addition is not transferred to E if we want E to remain a 1.

If the shift left operation inserts a zero into E, the divisor is subtracted by adding it's 2's complement value and the carry is transferred into E. If E = 1, it signifies that $A \geq B$ and hence Q_n is set to 1. If E = 0, it signifies that A < B and the original number is restored by adding B to A. In the latter case we leave a 0 in Q_n (0 was inserted during the shift).

This process is repeated again with register A holding the partial remainder. After n-1 times, the quotient magnitude is formed in the register Q and the remainder is found in register A.


lecture-9.docx

1.1.2.1. Lecture-10

Floating-Point Addition and Subtraction

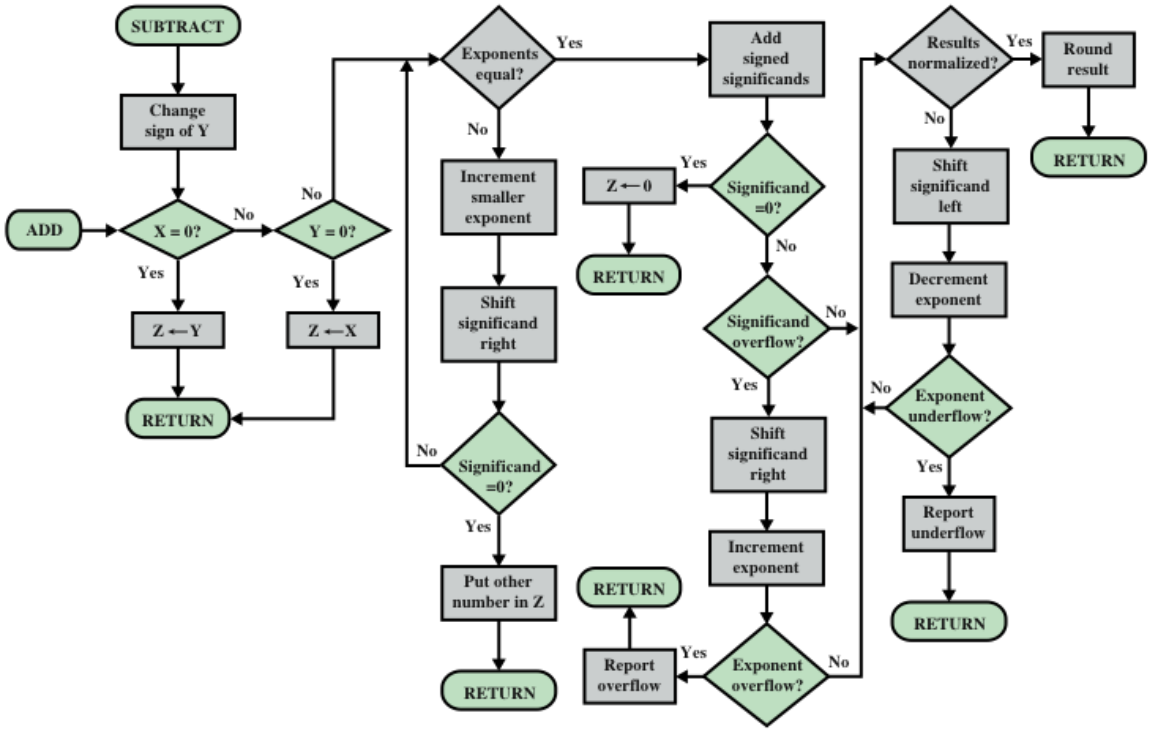


Figure 10.22 Floating-Point Addition and Subtraction (Z← X ± Y)

Floating-Point Multiplication

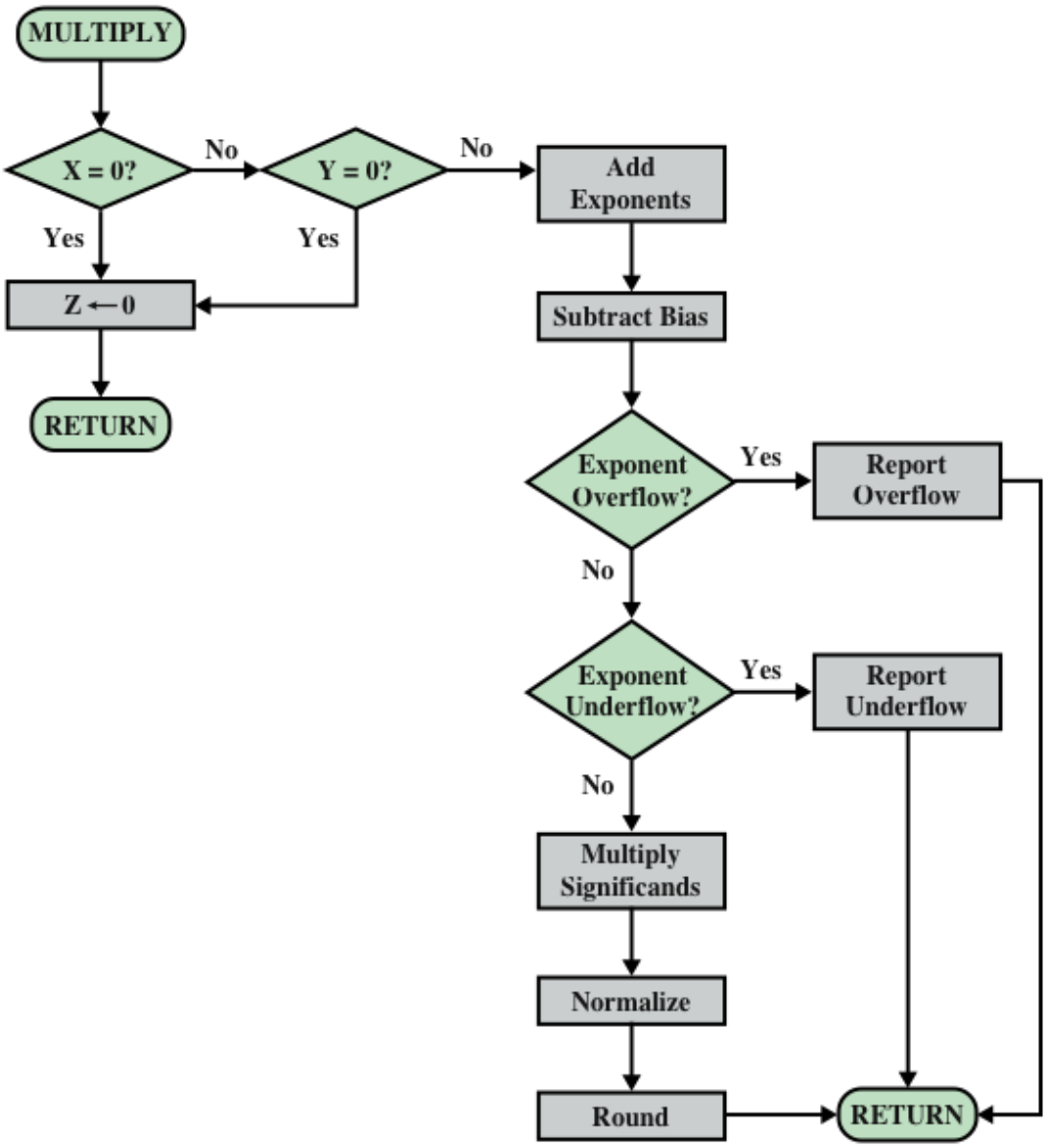


Figure 10.23 Floating-Point Multiplication (Z← X× Y)

Floating-Point Division

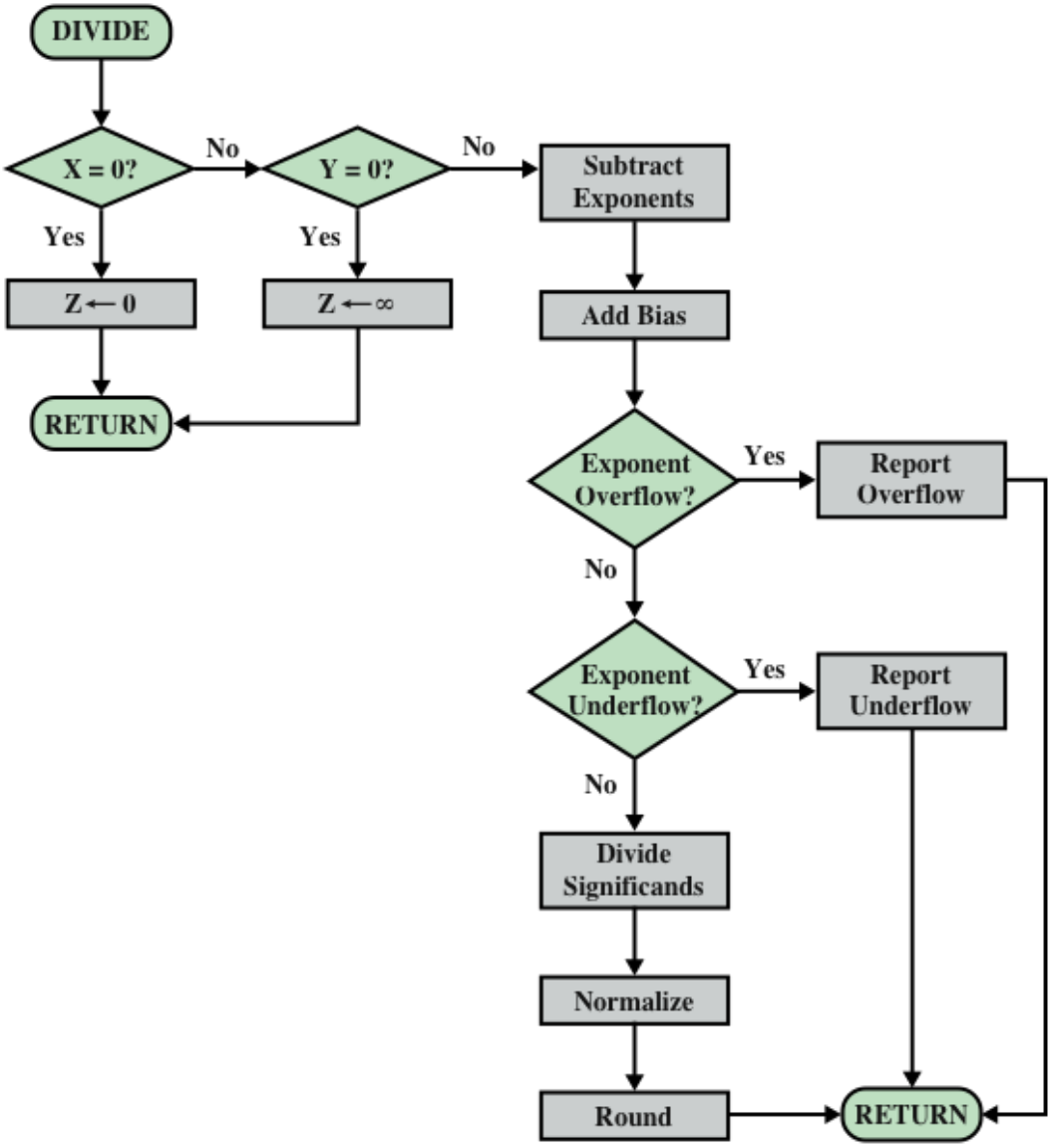


Figure 10.24 Floating-Point Division (Z← X/Y)

Lecture-11

lecture-11.docx

Lecture-12

lecture-12.docx

Lecture-13

lecture-13.docx

Test Questions

4.1.5.a.docx


4.1.5.b.docx


true or false.docx

Fill in the blanks type of questions <Minimum of ten>

a. The decimal representation for hex number F3 is _____ .

b. The binary equivalent for the decimal number 41.6875 is _____

c. The BCD code for the decimal number 248 is _____ .

d. For a given number N in base r having n digits, the (r-1)'s complement of N is defined as _____

e. The 10's complement of a decimal number is obtained by adding _____ to the 9's complement value.

f. When 2 unsigned numbers are added, an overflow is detected from the _____ of the most significant position.

g. An overflow for addition/ subtraction of two signed numbers is detected when the carry into the sign bit position and carry out of the sign bit position are _____ .

h. Booth multiplication algorithm is followed when the binary integers are represented in _____

i. When Booth algorithm is used for multiplication, the partial product does not change when the multiplier .is identical to the previous multiplier .

j. Floating point multiplication and division do not require an alignment of the _____ .

Answers: (1). 243 (2) 101001.1011 (3) 0010 0100 1000

(4) (r^n-1)-N (5) 1 (6) carry out (7) not equal

(8) signed 2's complement representation for negative integers. (9) bit, bit

(10) mantissa

Multiple choice questions<Minimum of ten>

1. Floating point representation is used to store
(A) Boolean values (B) whole numbers (C) real integers (D) integers
Ans: C

2. In computers, subtraction is generally carried out by
(A) 9's complement (B) 10's complement (C) 1's complement (D) 2's complement

Ans: D

3. The circuit used to store one bit of data is known as
(A) Register (B) Encoder (C) Decoder (D) Flip Flop
Ans: D

4. Which of the following is not a weighted code?
(A) Decimal Number system (B) Excess 3-cod
(C) Binary number System (D) None of these
Ans: B

5. Assembly language
(A) uses alphabetic codes in place of binary numbers used in machine language
(B) is the easiest language to write programs
(C) need not be translated into machine language
(D) None of these
Ans: A

6. The multiplicand register & multiplier register of a hardware circuit implementing booth's algorithm have (11101) & (1100). The result shall be
(A) (812) 10 (B) (-12) 10 (C) (12) 10 (D) (-812) 10
Ans: A

7. What characteristic of RAM memory makes it not suitable for permanent storage?
(A) too slow (B) unreliable (C) it is volatile (D) too bulky
Ans: C

8. (2FAOC) 16 is equivalent to
(A) (195 084) 10 (B) (001011111010 0000 1100) 2 (C) Both (A) and (B) (D) None of these
Ans: B

9. The average time required to reach a storage location in memory and obtain its contents is called the
(A) seek time (B) turnaround time (C) access time (D) transfer time
Ans: C

10. In signed-magnitude binary division, if the dividend is (11100) 2 and divisor is (10011) 2 then the result is
(A) (00100) 2 (B) (10100) 2 (C) (11001) 2 (D) (01100) 2
k. **Ans:** B

Fill the blank with true or false.

1. EEPROM comes under volatile memory category. _____
2. Thumb drive or pen drive is semiconductor memory. _____
3. The control unit generates the appropriate signal at the right moment. _____
4. While executing a program, CPU brings instruction and data from disk memory. _____
5. A memory module of capacity 16 * 4 , indicates a storage of 128 bits. _____
6. A memory module of capacity of 1024 locations, the required address bus size is 10. _____
7. The program counter PC is used to store the address of the next instruction to be fetched from Accumulator. _____
   .
8. For n-bit signed integer, the range of numbers that can be represented is -2^{n-1} to 2^{n+1}. _____
9. Given a number N in base r having n digits, the (r-1)'s complement of N is defined as

   $(r^n-1) - r$. _____
10. Floating point representation uses mantissa and an exponent part of radix R . _____

Answers: (1). false (2) true (3) true (4) false (5) false
(6) true (7) false (8) false (9)false (10) true

Review Questions


4.1.6.a.docx


4.1.6.b.docx


4.1.6. d.docx


4.1.6.e.docx


4.1.6.c.docx

a. Objective type of questions(Very short notes)<Minimum of ten>
b. Analytical type questions<Minimum of ten>
c. Essay type Questions<As per requirements>
d. Problems<As per required Number>

e. Case study<As per required Number>

Skill Building Exercises/Assignments

a. Take the mother board of a computer and identify CPU, memory, peripheral ICs, BUS etc.

b. Buy the components of a computer, assemble, install the software and make it to function.

Eg:- -Prepare a model of something
-Trace something
-Prepare a report on something etc.,

Previous Questions (Asked by JNTUK from the concerned Unit)


JNTUK questions unit-1.docx

GATE Questions (Where relevant)

Subject is not in gate syllabus

Interview questions (which are frequently asked in a Technical round-Placements)


4.1.10.docx

Real-Word (Live) Examples/Case studies wherever applicable

a. List out the intel CPUs in various generation with their specifications. Write how the performance was improved in each generation.

Suggested "Expert Guest Lectures" (both from in and outside of the campus)

Literature references of Relevant NPTEL Videos/Web/You Tube videos etc.


neptel vidieo ref.docx

Reference Text Books / with Journals Chapters etc.

T1: M. Moris Mano (2006), Computer System Architecture, 3rd edition, Pearson/PHI, India.

T2: Carl Hamacher, ZvonksVranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India

R1: Computer Organization Architecture- William Stallings (2006), 7th edition, PHI/PEARSON.

Unit – II

REGISTER TRANSFER LANGUAGE AND MICRO-OPERATIONS

REGISTER TRANSFER LANGUAGE AND MICRO-OPERATIONS:
Register Transfer language. Register Transfer, Bus and memory transfer,Arithmetic Micro-operations, logic micro operations, shift micro-operations,Arithmetic logic shift unit. Instruction codes. Computer Registers Computerinstructions –Instruction cycle. Memory Reference Instructions. Input Onput and Interrupt. **CENTRAL PROCESSING UNIT -** Stack organization.Instruction formats. Addressing modes. DATA Transfer and manipulation.Program control. Reduced Instruction set computer*UNIT SYLLABUS*

Unit Objectives:

After reading this Unit, the reader should be able to understand:

- Register Transfer language & use it to express micro-operations in symbolic form.
- The symbol definition for Arithmetic , logic, shift micro-operations
- The development of a composite Arithmetic logic shift unit to show the hardware design of the most common micro operations.
- The organization and design of a basic digital computer. That is registers, common bus system, memory system, computer instructions, timing and control, micro operations involved in instruction cycle of various instructions.
- The functional capabilities of a stored program general purpose device.
- The description of internal operation of the computer by RTL and providing design requirements by RTL
- Input-output by means of programmed control transfer and an interrupt cycle.
- The operation of the memory stack and stack applications
- Various instruction formats and Addressing modes.
- The operation of most common Instructions & their operation.

- The RISC concept its characteristics and advantages.

1.1.3. Unit Outcomes:

Student is able to

- give the definition for micro operation, register transfer and symbols for RTL
- Student understands what is register transfer, bus transfer and control condition for transfer; differentiate between common bus with multiplexer and 3-state buffers.
- Able to represent arithmetic, logic and shift micro operations using RTL.
- Able to design circuits for arithmetic, logic and shift operations.
- Able to explain instruction code, addressing modes for instruction operands and stored program concept.
- List out registers for a basic computer, explain their connections to common bus and control signals to effect micro operations.
- Explain different instruction formats, set of basic computer instructions that provide instruction set completeness.
- Explains control unit, timing signals, hardwired control and micro programmed control.
- Able to provide micro operations in the execution of instructions.
- Able to provide micro operations for input – output operations, programmed I/O and interrupt based I/O.
- Explain the micro operations of register stack, memory stack required for instructions used with stack
- Able to differentiate between infix notation and reverse polish notation in arithmetic expressions and the conversions from one to another and thereby evaluate arithmetic expressions efficiently.
- Explain instruction format, number of operands addresses used in the instruction and their advantages and disadvantages; different addressing modes used for operand addresses.
- Able to list out instructions for data transfer, data manipulation and program control instructions and explain various types of interrupts.
- Able to differentiate between RISC and CISC machines

Lectures for the concerned Unit:

Lecture-1

unit-2.Lecture-1.pptx

Lecture-2

unit-2.Lecture-2.pptx

Lecture-3

unit-2.Lecture-3.pptx

Lecture-4

unit-2.Lecture-4.pptx

Lecture-5

unit-2.Lecture-5.pptx

Lecture-6

unit-2.Lecture-6.pptx

Lecture-7

unit-2, l7.ppt

Lecture-8

unit-2, l8.ppt

Lecture-9

unit-2, l9.ppt

Lecture-10

unit-2,l-10.ppt

Lecture-11

unit-2.Lecture-11.pptx

Test Questions

4.2.5.b.docx


4.2.5.c.docx

Fill in the blanks using true or false.

1. The operations executed on data stored on memory is called Micro operations. _____
2. The symbolic notation used to describe the micro operation transfers among registers is called register transfer language. _____
3. The control function is included in the RTL as: P R2 ← R1. _____ .
4. Arithmetic micro operations perform arithmetic operations on binary data stored in registers. _____
5. The AND micro operation can be used to selectively set bits of a register. _____
6. The EX-OR micro operation can be used to selectively complement bits of a register. _____ .
7. The AND micro operation can be used for selectively clearing bits in a register. _____
8. The instruction read from memory is placed in the register PC . _____
9. Effective address is defined as the address of the operand in a computation type instruction or the target address in a branch type instruction. _____
10. Inn hardwired control, control logic is implemented with gates, flip-flops, decoders,and other digital circuits. _____

Answers: (1) True (2) True (3) false (4) false (5) false (6) True (7) True

(8) false (9) True (10) True.

a. Multiple choice questions<Minimum of ten>

b. <u>UNIT-II</u>

c.
d. 1. In Reverse Polish notation, expression A*B+C*D is written as
e. (A) AB*CD*+ (B) A*BCD*+ (C) AB*CD+* (D) A*B*CD+
f. **Ans:** A

g.
h. 2. The addressing mode used in an instruction of the form ADD X Y, is
i. (A) Absolute (B) indirect (C) index (D) none of these
j. **Ans:** C

k. 3. The BSA instruction is
l. (A) Branch and store accumulator (B) Branch and save return address
m. (C) Branch and shift address (D) Branch and show accumulator
n. **Ans: B**
o.
p. 4. In a program using subroutine call instruction, it is necessary
q. (A) initialise program counter (B) Clear the accumulator
r. (C) Reset the microprocessor (D) Clear the instruction register
s. **Ans:** D

t. 5. A Stack-organised Computer uses instruction of

u. (A) Indirect addressing (B) Two-addressing (C) Zero addressing (D) Index addressing
v. Ans: C

w. 6. Logic X-OR operation of (4ACO) H & (B53F) H results
x. (A) AACB (B) 0000 (C) FFFF (D) ABCD
y. Ans: C

z. 7. When CPU is executing a Program that is part of the Operating System, it is said to be in (A) Interrupt mode (B) System mode (C) Half mode (D) Simplex mode
aa. Ans: B

bb. 8. A three input NOR gate gives logic high output only when
cc. (A) one input is high (B) one input is low
dd. (C) two input are low (D) all input are high
ee. **Ans: D**
ff. 9. n bits in operation code imply that there are _____ possible distinct operators (A) 2n (B) 2n (C) n/2 (D) n2
gg. Ans: B

hh. 10. The instruction 'ORG O' is a
ii. (A) Machine Instruction. (B) Pseudo instruction.
jj. (C) High level instruction. (D) Memory instruction.
kk. Ans: B

II.

mm.

c.True or False questions<Minimum of ten>

Fill in the blanks with true or false statement.

1. An instruction code is a group of bytes that instruct the computer to perform a specific operation. _____ .
2. The number of bits required for the operation code of an instruction depends on the total number of operations available in the computer. _____ .
3. When the second part of an instruction code specifies an operand, the instruction is said to have direct address _____ .
4. If the memory address register has 12 bits, then the program counter register will have 16 bits. _____ .
5. If the load input of a register is enabled, then it will receive data from the bus during the next clock pulse transition. _____
6. The timing signals to the control logic can be derived by decoding the output of a sequence counter. _____
7. The operation of deletion in a stack is called push or push down operation. _____
8. Arithmetic, logical and shift instructions come under data manipulation instructions. _____
9. The instruction that transfers program control to a subroutine is known as branch and save address. _____
10. Interrupts are classified as traps and faults. _____

Answer: (1) false (2) True (3) false (4) false (5)true (6) true (7) false (8) true (9) true (10) false

1.1.4. Review Questions


4.2.6.c.docx


4.2.6.a.docx


4.2.6.b.docx

a. Objective type of questions(Very short notes)<Minimum of ten>
b. What do you mean by RTL?.
c. What do you mean by common bus system?
d. What is the use of 3 state buffers?

e. List out different arithmetic operations.
f. What do you mean by stored program organization.
g. List out registers for a basic computer.
h. What is meant by hardwired control unit?
i. What is meant by micro programmed control?
j. What do you mean by interrupt based data transfer?
k. What is reverse polish notation?

b.Analytical type questions<Minimum of ten>

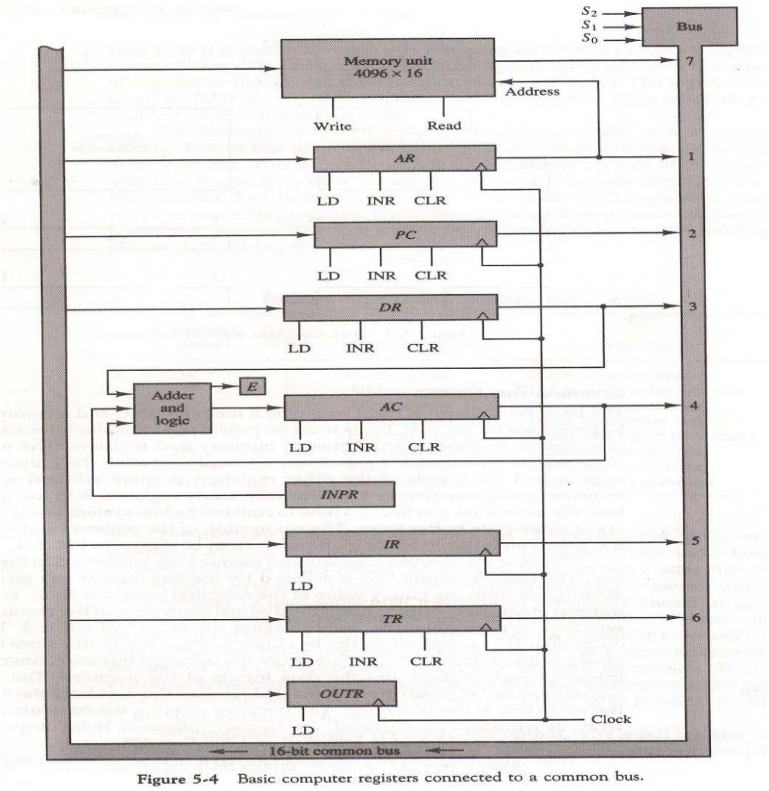


Figure 5-4 Basic computer registers connected to a common bus.

1. Referring to the bus system shown above, explain why each of the following micro operations cannot be executed during a single clock pulse. Specify a sequence of micro operations that will perform the operation.

   A. IR ← M[PC] B. AC ← AC +TR
   C. DR ← DR + AC (AC does not change).

2. Referring to the above bus system, the following control inputs are active. For each case , specify the register transfer that will be executed during the next clock transition.

	S_2	S_1	S_0	**LD** of register	Memory	Adder
a.	1	1	1	IR	Read	———
b.	1	1	0	PC	———	———
c.	1	0	0	DR	Write	———
d.	0	0	0	AC	———	Add

3. The following register transfers are to be executed in the above system. For each transfer specify: (1) The binary value that must be applied to bus select inputs S2, S1, S0; (2)the register whose load control input must be active(if any); (3) A memory read or write operation (if needed); and (4) the operation in the adder and logic circuit (if any).

a. $AR \leftarrow PC$
b. $IR \leftarrow M[AR]$
c. $M[AR] \leftarrow TR$
d. $AC \leftarrow DR,\ DR \leftarrow AC$ (done simultaneously)

4. The content of *PC* in the basic computer is 3AF (all numbers are in hexadecimal). The content of *AC* is 7EC3. The content of memory at address 3AF is 932E. The content of memory at address 32E is 09AC. The content of memory at address 9AC is 8B9F.
   a. What is the instruction that will be fetched and executed next?
   b. Show the binary operation that will be performed in the *AC* when the instruction is executed.
   c. Give the contents of registers *PC*, *AR*, *DR*, *AC*, and *IR* in hexadecimal and the values of *E*, *I*, and the sequence counter *SC* in binary at the end of the instruction cycle.

5. A digital computer has a memory unit with a capacity of 16,384 words, 40 bits per word. The instruction code format consists of six bits for the operation part and 14 bits for the address part (no indirect mode bit). Two instructions are packed in one memory word, and a 40-bit instruction register *IR* is available in the control unit. Formulate a procedure for fetching and executing instructions for this computer.

Convert the following arithmetic expressions from infix to reverse Polish notation.

a. $A * B + C * D + E * F$

b. $A * B + A * (B * D + C * E)$

c. $A + B * [C * D + E * (F + G)]$

d. $\dfrac{A * [B + C * (D + \mathrm{E})]}{F * (G + H)}$

Convert the following arithmetic expressions from reverse Polish notation to infix notation.

a. $A\ B\ C\ D\ E + * - /$

b. $A\ B\ C\ D\ E * / - +$

c. $A\ B\ C * /\ D - E\ F\ /\ +$

d. $A\ B\ C\ D\ E\ F\ G + * + * + *$

8.

Write a program to evaluate the arithmetic statement:

$$X = \frac{A - B + C * (D * E - F)}{G + H * K}$$

a. Using a general register computer with three address instructions.

b. Using a general register computer with two address instructions.

c. Using an accumulator type computer with one address instructions.

d. Using a stack organized computer with zero-address operation instructions.

9.

How many times does the control unit refer to memory when it fetches and executes an indirect addressing mode instruction if the instruction is (a) a computational type requiring an operand from memory; (b) a branch type.

10.

Given the 16-bit value 1001101011001101. What operation must be performed in order to:

a. clear to 0 the first eight bits?
b. set to 1 the last eight bits?
c. complement the middle eight bits?

l. Essay type Questions<As per requirements>

Essay Type Questions:

UNIT-II

1. What is the need of addressing modes? Explain different types of addressing modes
2. List out the instruction formats used in the processor and discuss with example
3. Explain about machine instruction characteristics
4. What are the differences between direct and indirect addressing instructions? How many references to memory are needed for each type of instruction to bring an operandin to a process register?
5. Discuss about the register organization in computer
6. Explain about registers for floating point arithmetic operation
7. Discuss about one stage of a decimal arithmetic unit.
8. Discuss about adding of decimal numbers methods.
9. Discuss about the design of the control unit
10. ExplainBooth'salgorithm. ApplyBooth'salgorithm tomultiplythetwodecimal numbers14and12.Assumethemultiplierandmultiplicandtobeof5bitseach

m. Problems<As per required Number> N.A
n. Case study<As per required Number>N.A

Skill Building Exercises/Assignments

a. Compare Instruction set of 8 bit and 16 bit and 32 bit processors

b. Compare addressing mode of 8 bit, 16 bit and 32 bit processors.

c. Compare features of 8,16 and 32 bit processors

Eg:- -Prepare a model of something
-Trace something
-Prepare a report on something etc.,

Previous Questions (Asked by JNTUK from the concerned Unit)

JNTUK questions unit-2.docx

1. A. Consider 4 , 4-bit registers A,B,C and D connected to a common bus system using multiplexers. What has to be done to the bus system so that information can be transferred from any register to any other register?.
   B. Represent the following conditional control statements by two register transfer statements with control functions. If (p=1) then (R1 ← R2) else if (Q =1) then (R1 ← R3).

2. A digital computer has common bus system for 16 registers of 32 bits each. The bus is constructed with multiplexers.

a. How many selection inputs are there in each multiplexer?

b. What size of multiplexers are needed?

c. How many multiplexers are in the bus?

3a.Draw the block diagram of the hardware that implements the following statements:

X +yz: AR ← AR + BR

, where AR and BR are 2 n bit registers and x, y and z are control variables. Include the logic gates for the control function.

b. Show the hardware that implements the following statements.Include the logic gates for the control function and a block diagram for the binary counter with a count enable input.

$xyT_0 + T_1 + y' T_2$: AR ← AR +1 .

4. Design a 4 bit combinational circuit decrementer using 4 full adder circuits.

5. What is wrong with the following register transfer statements?

a. xT: AR ← $\overline{AR}$, AR ← 0 b. yT: R1 ← R2, R1 ← R3 c. zT: PC ← AR, PC ← PC + 1

6. (a) Explain the various Instruction types.
(b) Draw and explain the flow chart for instruction cycle.

7 (a) Explain various instruction formats with examples.
(b) Write short notes on process organization.
8. (a) List and explain the characteristics of machine instructions
(b) Describe various addressing modes in detail
(c) Give a short note on instruction pipelining

9a. Draw the flowchart for memory reference instructions and explain.
b. Discuss about stack organization with register stack and memory stack.

10a. Design a combinational circuit for i) 4-bit shifter ii) 4-bit decrementer and explain.
b. Briefly explain the computer registers for basic computer.

11a. Design a combinational circuit for the following arithmetic operations

(i) Addition (ii) Subtraction (iii) Increment (iv) decrement

b. Discuss about instruction codes and stored program organization

Interview questions (which are frequently asked in a Technical round-Placements)


4.2.10.docx

4.2.10 Interview questions

1. What is micro operation in a digital computer.

2. What is register transfer language?

3. What do you mean by addressing mode?

4. What is indirect address?

5. What is meant by instruction set of a computer?

6. What is the use of PC and IR in a digital computer?

7. What is stack in a computer?

8. Explain instruction format.

9. What is interrupts and interrupt service routine?

10. What is RISC and CISC?

Text Books:

3. M. Moris Mano (2006), Computer System Architecture, 3rd edition, Pearson/PHI, India.
4. Carl Hamacher, Zvonks Vranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India.

Reference Books:

3. Computer Organization Architecture- William Stallings (2006), 7th edition, PHI/PEARSON.
4. ComputerArchitectureandOrganization-John P.Hayes ,Mc Graw Hill, International editions,2002.

Unit – III –Micro programmed Control

MICRO PROGRAMMED CONTROL: Control memory, Addresssequencing, micro program example, Design of control unit-Hard wiredcontrol. Micro programmed control.

3.1. Unit Objectives

After reading this Unit, the reader should be able to understand:

- The concept of microprogramming and define control memory, microinstruction, micro program, control address register, address sequencer, pipeline register etc.
- writing microcode for a typical set of instructions.
- The design of micro programmed control unit, by designing hardware for the microprogram sequencer.

3.2. Unit Outcomes:

Student is able to

- Differentiate between hardwired control and micro programmed control
- Explain control word, microinstruction, micro program, control memory, control address register, micro program sequencer and pipeline register.
- Understands micro program routine, sequencing microinstructions in a routine and branching from one routine to another.
- Understands how microprogram routine is reached from instruction code.
- Develop a micro program for a given computer hardware configuration and convert into binary micro program.
- Develop a control unit with the design of micro program sequencer.

3.3. Unit Lecture plan

Lecture no.	*Topic*	*Methodology*	*Quick reference*
1.	Control memory	Chalk & Board, PPT	T1, page 213 -216
2.	Address Sequencing	Chalk & Board	T1, page 216-220

3.	Micro Program Example: computer configuration, instruction format and symbolic microinstructions; The fetch routine, symbolic micro program and binary micro program	Chalk & Board	T1, page 220-230
4.	Design of Control Unit: Hard wired Control and Micro programmed control	Chalk & Board	T1, page 231-235

3.4. Teaching Material / Teaching Aids as per above lecture plan.

3.4.1 Lecture-1

unit-3, L1, control memory.pptx

3.4.2. Lecture-2

Unit-3, L2, address sequencing.pptx

3.4.3 Lecture-3

Unit-3, L3, Microprogram exampl

3.4.4 Lecture-4

Unit-4, L4, Design of control unit.pptx

3.4.2 Test Questions

a. Fill in the blanks type of questions

Fill in the blanks type questions.

1. The function of control unit in a digital computer is to initiate _____ Of micro operations.
2. When the control signals are generated by hardware using conventional _____ techniques , the control unit is said to be hardwired.
3. The control variables at any given time can be represented by a string of _____ Called a control word.
4. A control unit whose _____ control variables are stored in memory is called a micro programmed control unit.
5. A _____ of micro instructions constitutes a micro program.
6. The next address generator in a control unit is sometimes called micro program _____
7. The data register in a micro programmed control unit is sometimes called _____
8. .The transformation from the instruction code bits to an address in control memory where the routine is located is referred to as a _____ .
9. _____ are programs that are used by other routines to accomplish a particular task.
10. In a microinstruction format , the CD field select _____ bit conditions.

b. Multiple choice questions

1. Translation from symbolic program into Binary is done in
(A) Two passes. (B) Directly (C) Three passes. (D) Four passes.
Ans: A

2.A microprogram sequencer
(A) generates the address of next micro instruction to be executed.
(B) generates the control signals to execute a microinstruction.
(C) sequentially averages all microinstructions in the control memory.
(D) enables the efficient handling of a micro program subroutine.
Ans: A

3.The operation executed on data stored in registers is called
(A) Macro-operation (B) Micro-operation
(C) Bit-operation (D) Byte-operation
Ans: B

4. Microinstructions are stored in control memory groups, with each group specifying a
(A) Routine (B) Subroutine (C) Vector (D) Address
Ans: A

5. PSW is saved in stack when there is a
(A) interrupt recognised (B) execution of RST instruction
(C) Execution of CALL instruction (D) All of these
Ans: A

6. In a vectored interrupt.
(A) the branch address is assigned to a fixed location in memory.
(B) the interrupting source supplies the branch information to the processor through an interrupt vector.
(C) the branch address is obtained from a register in the processor
(D) none of the above
Ans: B

7. The communication between the components in a microcomputer takes place via the address and
(A) I/O bus (B) Data bus (C) Address bus (D) Control lines
Ans: B

8. In a program using subroutine call instruction, it is necessary
(A) initialise program counter (B) Clear the accumulator
(C) Reset the microprocessor (D) Clear the instruction register
Ans: D

9. A microprogram written as string of 0's and 1's is a a. symbolic microinstruction b. binary microinstruction c. symbolic microprogram d. binary microprogram
Ans d

c. fill the blanks with true of false statement.

1. A word in control memory location is called microinstruction. _____ .
2. A group of microinstructions constitute micro program. _____ .
3. In dynamic microprogramming, the micro program can be initially loaded from disk. _____
4. Each machine instruction initiate a microinstruction in control memory. _____
5. The microinstruction specifies various internal control signals for execution of register micro operations. _____ .
6. Each machine instruction has associated micro program in control memory. _____
7. While the micro operations are are being executed, the next address is computed in the next address generator circuit and then transferred into the control data register. _____

8. If we want to establish a different control sequence for the system, we need to simply replace the control ROM with suitable micro programs , in the control unit of micro programmed control unit. ——————————

9. The subroutine register in the control unit stores the starting address of the subroutine. ——————————

10. Some of the bits in the instruction code is used to reach the starting address of the micro program routine associated with that instruction and this process is called mapping of instruction. —————————— .

Answer:

(1) True (2) false because a sequence of microinstructions constitute a microprogram.

(3) True (4) false because it initiates a series of microinstructions.

(5) True (6) True (7) false (8)True (9) false 9100 True

3.4.3 Review Questions

a. Objective type of questions(Very short notes)

1. What is the difference between a microprocessor and a micro program?
2. Is it possible to design a micro program without a micro program?
3. Are all micro programmed computers also microprocessors?
4. Is it possible to have hardwired control associated with a control memory?
5. Explain the difference between hardwired control and micro programmed control.
6. Te4ll the mapping process from instruction code to microinstruction address using ROM.
7. Specify a minimum of 3 components used in a micro program sequencer.
8. What is the advantage of writable control memory?
9. What is the function of CD and AD fields in the Microinstruction Format ?
10. Hat is the use of program counter.

b. Analytical type questions<Minimum of ten>

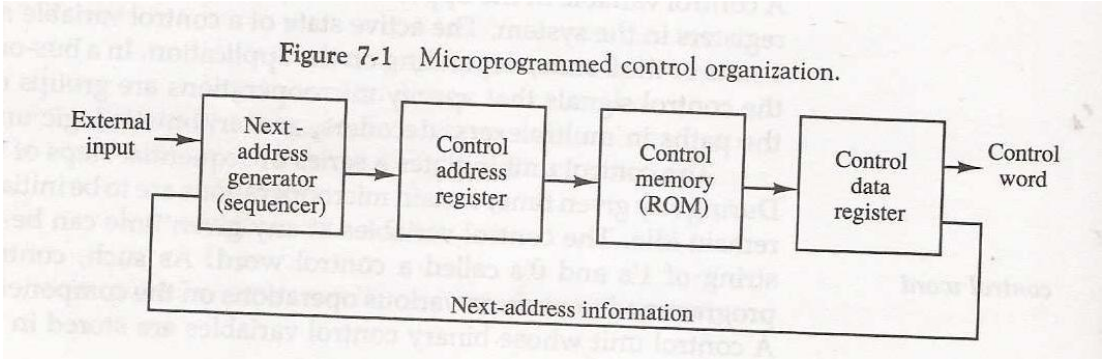


Figure 7-1 Microprogrammed control organization.

1. With reference to the above figure, the propagation delay timesare : 1) to generate the next address is 40ns, 2) to transfer the address into the control address register 10ns, 3) to access the control memory ROM 40 ns, 4) to transfer the microinstruction into the control data register 10ns, 5) to perform the required micro operation specified by the control word 40ns. What is the maximum clock frequency that the control can use?.
2. Formulate a mapping procedure that provide 8 consecutive microinstructions for each routine. The operation code has 6 bits and the control memory has 2048 words.
3. Show how a 9-bit micro operation field in a microinstruction can be divided into subfields to specify 46 micro operations. How many micro operations can be specified in one micro instruction?.
4. A computer has 16 registers, an ALU with 32 operations, and a shifter with 8 operations, all connected to a common bus system.
   a. Formulate a control word for a micro operation.
   b. Specify the number of bits in each field of the control word and give a general encoding scheme.
   c. Show the bits of the control word that specify the micro operation R4 ⟵ R5 + R6.

TABLE 7-1 Symbols and Binary Code for Microinstruction Fields

F1	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC + DR$	ADD
010	$AC \leftarrow 0$	CLRAC
011	$AC \leftarrow AC + 1$	INCAC
100	$AC \leftarrow DR$	DRTAC
101	$AR \leftarrow DR(0\text{–}10)$	DRTAR
110	$AR \leftarrow PC$	PCTAR
111	$M[AR] \leftarrow DR$	WRITE

F2	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC - DR$	SUB
010	$AC \leftarrow AC \vee DR$	OR
011	$AC \leftarrow AC \wedge DR$	AND
100	$DR \leftarrow M[AR]$	READ
101	$DR \leftarrow AC$	ACTDR
110	$DR \leftarrow DR + 1$	INCDR
111	$DR(0\text{-}10) \leftarrow PC$	PCTDR

F3	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow \overline{AC}$	COM
011	$AC \leftarrow$ shl AC	SHL
100	$AC \leftarrow$ shr AC	SHR
101	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Reserved	

CD	Condition	Symbol	Comments
00	Always = 1	U	Unconditional branch
01	$DR(15)$	I	Indirect address bit
10	$AC(15)$	S	Sign bit of AC
11	$AC = 0$	Z	Zero value in AC

BR	Symbol	Function
00	JMP	$CAR \leftarrow AD$ if condition = 1
		$CAR \leftarrow CAR + 1$ if condition = 0
01	CALL	$CAR \leftarrow AD$, $SBR \leftarrow CAR + 1$ if condition = 1
		$CAR \leftarrow CAR + 1$ if condition = 0
10	RET	$CAR \leftarrow SBR$ (Return from subroutine)
11	MAP	$CAR(2\text{–}5) \leftarrow DR(11\text{–}14)$, $CAR(0,1,6) \leftarrow 0$

TABLE 7-1 Symbols and Binary Code for Microinstruction Fields

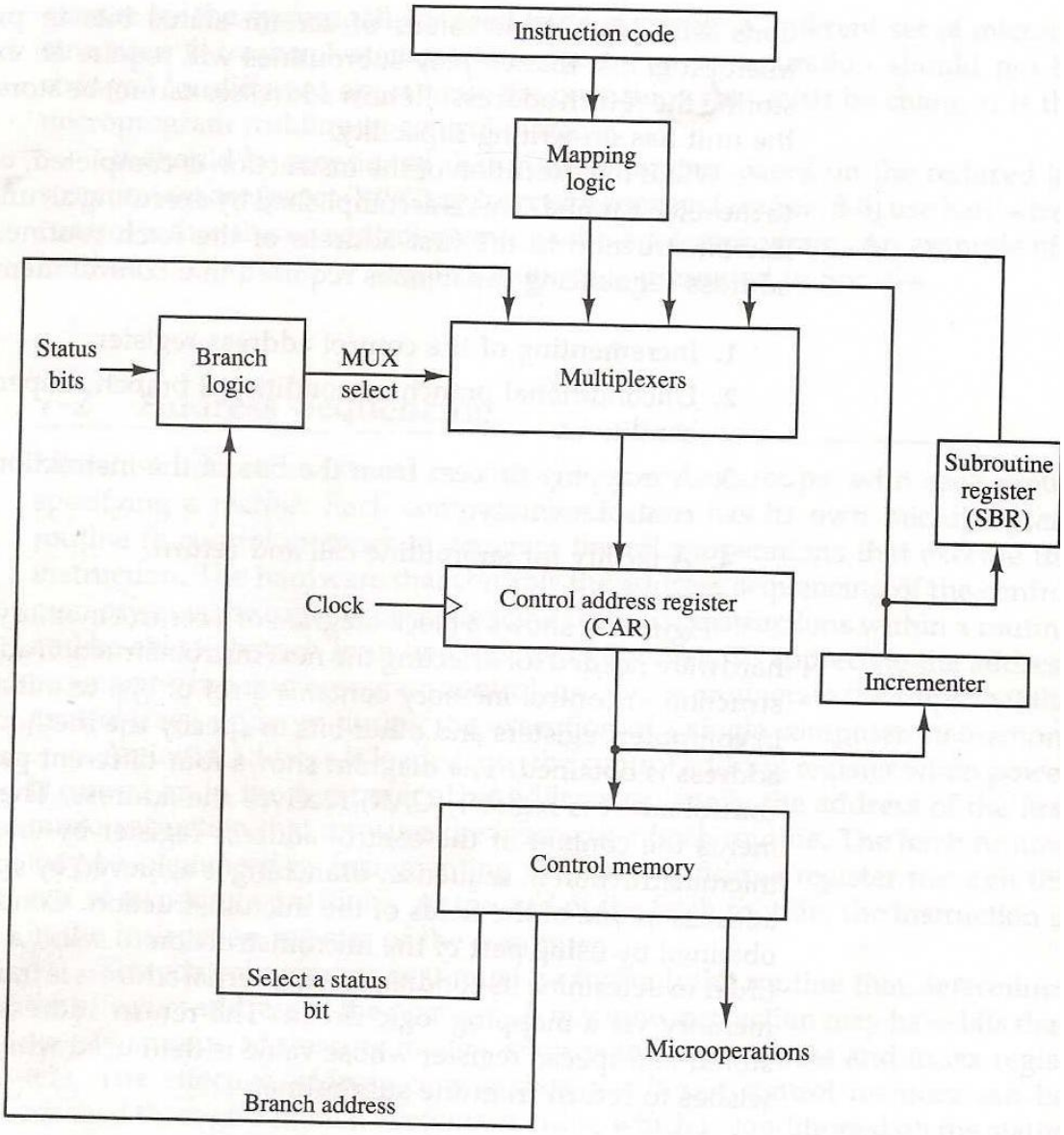


Figure 7-2 Selection of address for control memory.

5. The control memory system shown above uses a control memory of size 1024*32. The microinstruction has 3 fields as shown in the diagram. The micro operation field has 16 bits.
   a. How many bits are there in the branch address field and the select field?
   b. If there are 16 status bits in the system, how many bits of the branch logic are used to select a status bit?
   c. How many bits are left to select an input for the multiplexer?
6. The control memory shown above has 4096 words of 24 bits each.
   a. How many bits are there in the control address register?
   b. How many bits are there in each of the 4 inputs shown going into the multiplexer?
   c. What are the number of inputs in each multiplexer and how many multiplexers are needed?

7.

7-11. Using Table 7-1, give the 9-bit microoperation field for the following micro-operations:

a. $AC \leftarrow AC + 1, \quad DR \leftarrow DR + 1$

b. $PC \leftarrow PC + 1, \quad DR \leftarrow M[AR]$

c. $DR \leftarrow AC, \quad AC \leftarrow DR$

8.

7-12. Using Table 7-1, convert the following symbolic microoperations to register transfer statements and to binary.

a. READ, INCPC

b. ACTDR, DRTAC

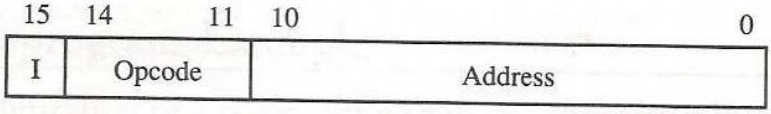
c. ARTPC, DRTAC, WRITE

9. Which control unit you will prefer? Tell the reasons for your decision.

10.

BSA	D_5	$M[AR] \leftarrow PC, \quad PC \leftarrow AR + 1$
ISZ		

Figure 7-5 Computer instructions.



(a) Instruction format

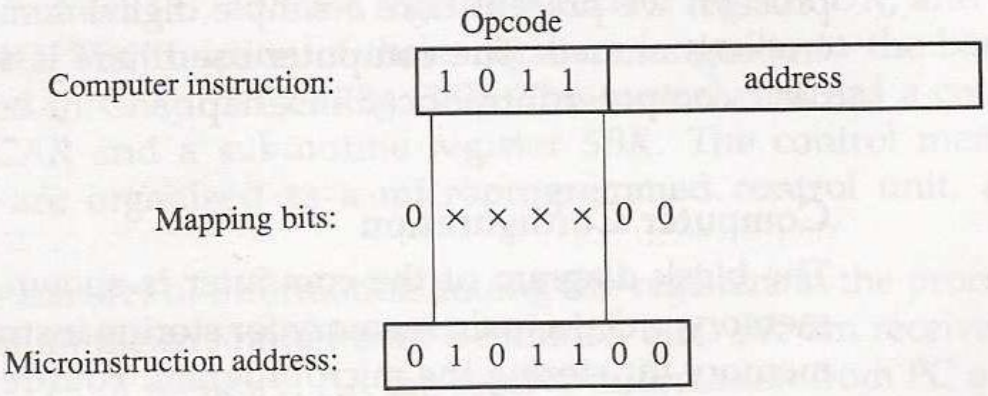
The micro operation for BSA instruction is given above and the microinstruction format is also given above. Write the symbolic micro program routines for the BSA instruction.

c. Essay Type Questions:

1. Explain about the micro programmed control organization
2. Explain the differences between hardwired control and micro programmed Control.
3. Define following terms, micro operations, microinstruction, micro program, micro code.
4. Explainthevarietyoftechniquesavailableforsequencingofmicroinstructions basedontheformatoftheaddressinformationinthemicroinstruction.
5. ExplaintheOrganization ofthecontrolunittoallowconditionalbranchinginthe micro program
6. Explain how the mapping from an instruction code to a micro instruction address can be done by means of ROM.
7. Draw the block diagram of a micro programmed sequencer and explain its operation.
8. Explain t h e b a s i c organizationof a m i c r o p r o g r a m m e d control u n i t a n d t h e generationofcontrolsignalsusingmicroprogram.

d. Problems

Figure 7-3 Mapping from instruction code to microinstruction address.



1. Using the mapping procedure given above, give the first microinstruction address for the following operation code: (a) 0010; (b) 1011; (c) 1111.
2. Explain how the mapping from an instruction code to microinstruction address can be done by means of a ROM. What is the advantage of this method compared to the method shown in the figure above?.
3. **7-14.** The following is a symbolic microprogram for an instruction in the computer defined in Sec. 7-3.

```
        ORG 40
        NOP         S    JMP     FETCH
        NOP         Z    JMP     FETCH
        NOP         I    CALL    INDRCT
        ARTPC       U    JMP     FETCH
```

   a. Specify the operation performed when the instruction is executed.
   b. Convert the four microinstructions into their equivalent binary form.

e. Case study<As per required Number> N.A.

3.4.4 Skill Building Exercises/Assignments

Draw a neat chart of micro program sequencer.

Write a report on the functioning of plcs

3.4.5 Previous Questions (Asked by JNTUK from the concerned Unit)

4. JNTUK QUESTIONS

1. Discussthebasicorganizationofamicroprogrammedcontrolunitandthegenerationof controlsignalsusingmicroprogram

2. Explaintheconceptofmicroprogrammedcontrolunit.

3.
(a)Explainthevarietyoftechniquesavailableforsequencingofmicroinstructions basedontheformatoftheaddressinformationinthemicroinstruction.

(b)CompareandContrasthardwiredcontrolunitwithmicroprogrammedcontrolunit.

4. (a)Hardwiredcontrolunitisfasterthanmicro programmedcontrolunit?Justify thisstatement.

(b)Briefly explainthebasic organizationofamicro programmedcontrolunitandthe generationofcontrolsignalsusingmicroprogram

5 a) Explaintheexecutionofmicroinstructionswithaneatdiagram. [5]

b) Brieflydescribethedesignofahardwiredcontrolunit. [10]

6. Explainthedesignofmicro-programmedcontrolunitindetail. [15]

7 a) ExplaintheOrganization ofthecontrolunittoallowconditionalbranchinginthe microprogram.

b) Whatishardwiredcontrol?Howisitdifferentfrommicroprogrammedcontrol?

8.Explain the basic organization of a micro programmed control unit and the generationofcontrolsignalsusingmicroprogram.

9. a)Givethetypicalhorizontalandverticalmicroinstructionformats.

b)Describehowmicroinstructionsarearrangedincontrolmemoryandhowthey areinterpreted.

10.a)Clearlydistinguishbetween

i. Packed/Unpackedmicroinstructions ii. Hard/Softmicroprogramming

b)Listandbrieflyexplainapplicationsofmicroprogramming.

11.a)Whatarethedesigngoalsforadesignerwhiledecidingahardwiredormicro programmedCUforaCPU.

b)Writeshortnoteson microinstructionsequencing.

12. a)
Definethefollowing:

i. Microoperation

ii. Microinstruction

iii.Microprogram

iv. Controlmemory.

b) Explaintheselectionofaddressforcontrolmemory?

13.a)GivetheCircuitDiagramforMicroProgramSequencefora ControlMemory.

b)Whatisthedifferencebetweena MicroprocessorandaMicroProgram? [8M+8M]

14.a)DiscussindetailaboutAddressSequencing.

b)WritetheSymbolicMicroProgramforBranchandSaveInstructions. [8M+8M]

15.a) DiscussindetailaboutthedecodingofMicroOperationfields.

b) Showhowa9-bitMicrooperationfieldinMicroinstructioncanbedividedintosubfields to specify46 Microoperations?HowmanyMicrooperationscanbespecifiedinone Microinstruction? [8M+8M]

16.a) WhatistheDifferencebetweenaMicroprocessorandaMicroprogram?Isitpossibleto designaMicroprocessorwithoutaMicroprogram?

b)WhatisthefunctionofControlUnit?ExplainthedifferencebetweenHardwiredcontrol andMicroprogrammedcontrol. [8M+8M]

17 a) Explain briefly about the micro programmed control organization. [7]
b) Briefly discuss about the design of the control unit[8]
18. a) Discuss about the process organization in computer.
b) Discuss about the register organization in computer.
19. a) Define following terms, micro operations, microinstruction, micro program, micro code.
b) Explain the differences between hardwired control and micro programmed control.
20. a) Draw the block diagram of microprogram control organization and explain.
b) Discuss how selection of address for control memory.

4.4.1 GATE Questions (Where relevant)

N.A.

4.4.2 Interview questions (which are frequently asked in a Technical round-Placements)

1. What is micro operation and Micro program?.
2. What is the use of control memory in a microprogrammed control?
3. What is meant by hardwired control?
4. What are the advantages of micro programmed control unit?
5. What are the functions of a micro program sequencer?

4.4.3 Real-Word (Live) Examples/Case studies wherever applicable

- Micro programmed State Machine design.
- A lot of machine controls with sequential operations can be implemented with control memory. Ex: Automatic machine tools with sequential control

4.4.4 Suggested "Expert Guest Lectures" (both from in and outside of the campus)

4.4.5 Literature references of Relevant NPTEL Videos/Web/You Tube videos etc.

http://nptel.ac.in/video.php?subjectId=106102062

For the following Topics: Processor Design Micro programmed Control (33:11)

4.4.6 Any Lab requirements; if so link it to Lab Lesson Plan.

N.A.

4.4.7 Reference Text Books / with Journals Chapters etc.

Text Book:

M. Moris Mano (2006), Computer System Architecture, 3rd edition, Pearson/PHI, India.