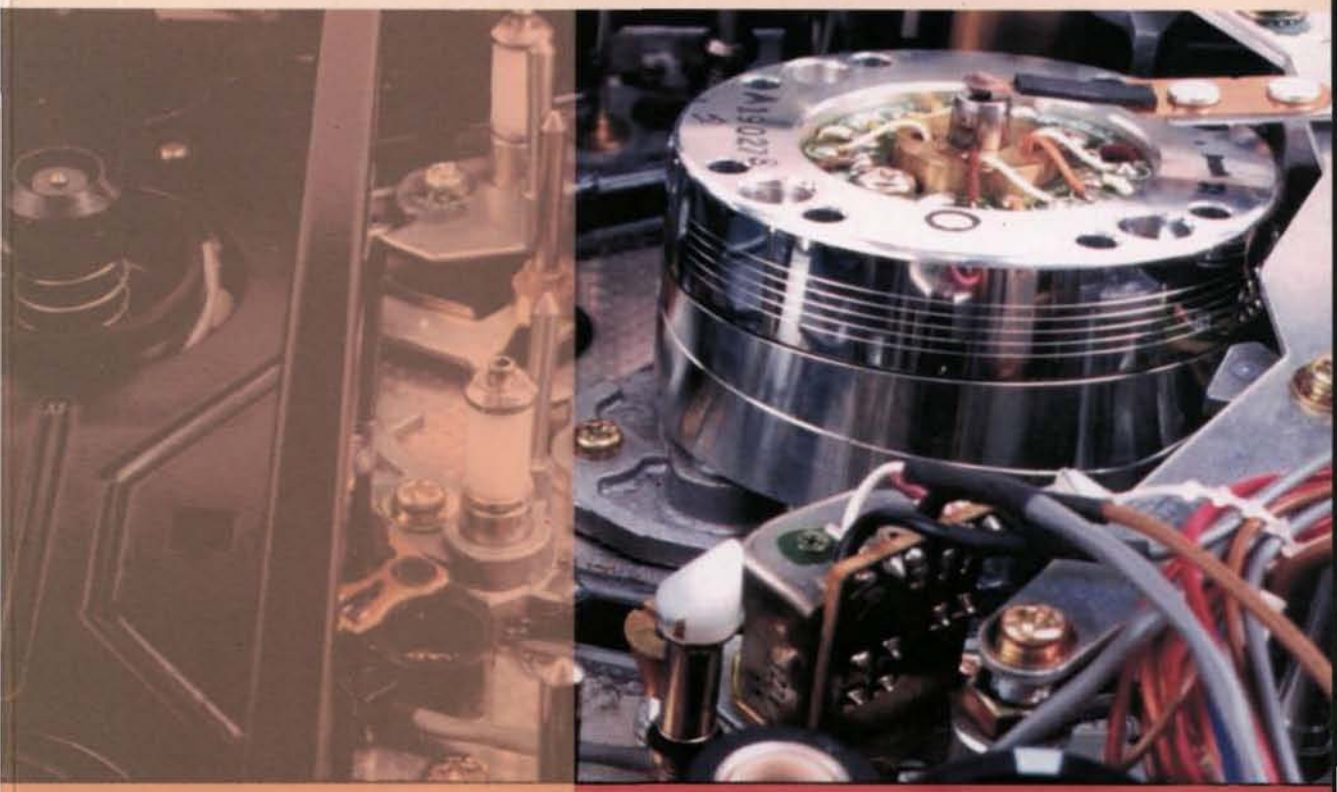
# COMPUTER ARCHITECTURE AND SYSTEMS DESIGN



J. Vaideeswaran

NEW AGE INTERNATIONAL PUBLISHERS

# COMPUTER ARCHITECTURE AND SYSTEMS DESIGN

**J. Vaideeswaran**
Electronics Department
MIT, Chromepet, Chennai

PUBLISHING FOR ONE WORLD

**NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS**

New Delhi • Bangalore • Chennai • Cochin • Guwahati • Hyderabad
Jalandhar • Kolkata • Lucknow • Mumbai • Ranchi

Visit us at www.newagepublishers.com

Published by New Age International (P) Ltd., Publishers



**ISBN (13) : 978-81-224-2496-6**

**PUBLISHING FOR ONE WORLD**

**NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS**
4835/24, Ansari Road, Daryaganj, New Delhi - 110002
Visit us at **www.newagepublishers.com**

*In memory of*
my uncle, Shri T.R. Subramanian

THIS PAGE IS
BLANK

# PREFACE

This century is indebted to the implementation of the Electronics technology which is considered to be a real revolution It was in 1950's electronic computing machines were born and living very much Today This only makes us think the enormous power of a computer Basically the term Architecture has been accepted with the inclusion of the microprocessors in the 1970 s Moreover in general, Architecture is an art and in particular, computer architecture is more concerned with the beneficiaries *i e*, applications area In its life-span the computers have learned a lot that it becomes a fitting subject of learning The architecture application domain started from data processing as a Robot meaning programmed machines and now grew to the highest status of programmable medium to suit varying models In the process each model by itself is a MODEL serving as examples and information base to others Many scientists and engineers have contributed a lot towards computing architectures

I in my experience as a teacher since 1982, take this occasion to write a Book on Computer Architecture and Systems Design ' The technology of processing elements in the realm of computer architectures process the input data and deliver some reasonable expected outputs Computer Architecture is defined to be the interaction and interrelation between the static hardware and dynamic software processing abilities The art of software engineering applied into the heart of computer hardware with the coordination of computer scientists is termed as Computer Architecture This text is intended primarily both for graduate and undergraduate levels for computer science and electrical engineering courses This is an attempt to interface the engineering and science curriculums

The book, in total, comprises of seven chapters In the first three chapters the Von Neumann machine of the single instruction single data sequential organisation is dealt in detail covering both central processing unit (CPU) and the primary memories Chapter 1 is intended for the number crunching activity in a meaningful fashion Chapter 2 dwells into internal organisation of CPUs more in general with the instruction capabilities Chapter 3 is dedicated primarily for primary memories closing with the secondary memories as well as input output devices Chapter 4 dictates much on a pipelined organisation for an interactive base besides summarizing some of the salient features of certain microprocessor chips

Chapter 5 on parallel processing and fault-tolerance touches on aspects of dataflow computing and multiprocessor configurations Chapter 6 on design methodology briefly touches on certain system concepts employing parallel processing and further explains transputers software reliability and some fault tolerant architectures

The reliability of systems is devoted to the fault tolerance of machine architectures, that is figured out on varying issues. The need for coprocessing is discussed in Chapter 7.

I deem it a great pleasure in thanking my wife Kanthimathi and daughter Pavithra for their encouragement while writing the book. I thank my Parents and the Almighty for the work herein reported

J. VAIDEESWARAN

# Contents

# CHAPTER 1

# NUMBER SYSTEMS AND ARITHMETIC LOGIC UNIT

## 1.1 Dialogue

From the time computers evolved long back in 1940's, it has revolutionised the communication activities around the universe as on date. The digital processing started only for off-line data as early as 1950's. This off-line machine is called a batch processing machine and COBOL (Common business oriented language), the communication language is used for data processing (just as a number crunching machine) since 1960's. In batch processing machines, the involvement of human being as the system operator has been reduced a lot with the evolving computer architectures. In such a design, throughput is the major consideration.

> ***Defn:*** *"THRUPUT is defined to be the complex combination of measured/ measurable output of a computer". Fortran (Formula Translation) is considered fit for scientific computing on batch machines.*

In 1971, Intel Corporation announced the first 4-bit microprocessor 4004 on a single integrated chip. This was indeed a revolution, i.e. to mean the beginning of man-machine interaction. In parallel, other manufacturers, viz., the Motorola, the Zillog corporation, the Texas Instruments, etc. came with innovative CPU architectures. With technology racing fast, it is seen the capability of central processing units grow multi-dimensionally to varying degrees. The period 1980's really can be called the golden age of interactive computers, men interrupting the machines. To cope up with the unit of time i.e. Seconds of the humans, the computers clock's unit has to be set to microseconds for counting any event. Thus, events contribute to a task, tasks forming an instruction and instructions making up a program in order to arrive at an expected and reasonable solution for a clearly stated problem. For an interactive machine, the Turn-around-Time is more a primary design parameter.

Communication, in general, emanates from the originating end and also ends with the point of origin finally. Fig. 1.1 depicts the IPO structure. With the importance of computer applications, the term User is felt in depth.

> ***Defns:*** *"User is one who logically, psychologically and arithmetically poses problem to a computer".*

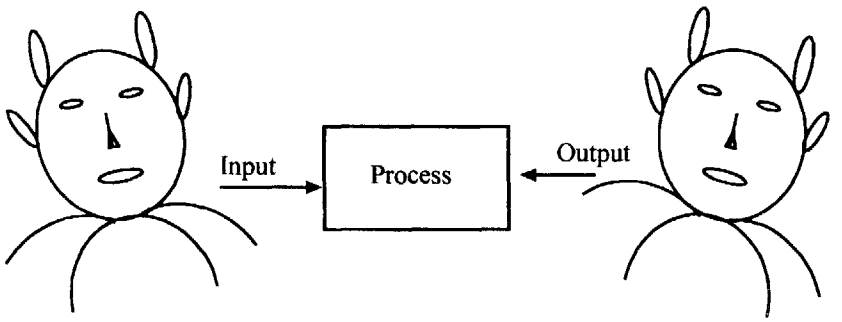



**Fig. 1.1** IPO Structure

*"User is defined as a single person, a group of people or a community of organisation benefited by the process in total:"*

***Defn:*** *TaT (Turn-around-Time) is the machine time lapsed from the program input to the problem solution as an On-line user is concerned on an interative environment".*

In this context, system engineering is a complex organisation of men, money and machines and through these integrated functions, an operational requirement is satisfied.

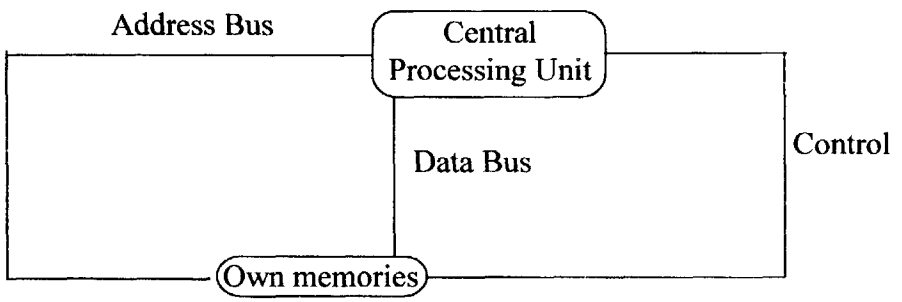



**Fig. 1.2(a)** CPU Architecture

Fig1.2(a) gives the schematic of the executor.

***Defn:*** *"Computer Architecture is defined to be the interaction and inter-relation between the static hardware and dynamic software processing abilities".*

Computers started assisting day to day life with varying radii of coverage, being built on a 2-valued logic, thanks to the identified dictionary of historical contributors from the

days of written history. English, probably the universally accepted language for visual global communications, is an approved language by computing scientists and engineers.

A remark at this point can be made, namely, "more often, THRUPUT and TaT never go together". So, today computer architectures converge to

1. Processing for communications and
2. Communications for computing in order to meet the busy business community at large and the active scientific community silently respectively.

*Defn: "The art of software engineering applied into the heart of computer hardware with the brain of software scientists can rightly be termed computer Architecture".*

Computer is a number crunching machine represented in binary logic, i.e., a sequence of 0's and 1's . With the CPU basically as a decider of two elementary binary symbols viz., 0 and 1, we shall pass onto Number Systems.

## 1.2 Weighted Number Systems

The universally accepted decimal number system uses the ten distinct symbols 0,1,2,3,4,5,6,7,8,9. In numerical computations the position of the symbol more reflects the value of a number. Thus, these are also known as positional number systems. The decimal number system has a base or radix of 10. Since all computations are carried using the two-valued binary logic the need arises to converge on binary systems.

***Defn.*** *" A number system with base or radix "r" has r different symbols including 0".*

Any number system with radix r has the r distinct symbols 0,1,2, ..... , r - 1 respectively. Thus, the binary number system employs the two binary digits 0 and 1. A binary digit is often called a *bit* in computer terminology.

The relationship between a number system with radix 'r' having the same capacity (of the universal decimal system) is given by $10^n = r^m$

(or) $$m/n = \frac{1}{\log_{10} r}$$

$$n \log 10 = m \log_{10} r$$

where *n* and *m* are the number of digits required for representing identical numbers in the two systems.

For eg. $10^3 = 2^m$

Therefore m = 3/.3010 = 10 bits.

Storage efficiency, $\eta_r$ of a number system with radix r can be defined with respect to decimal number system and is given by the equation

$$1/\eta_r = \frac{r.\,m}{10.\,n.} = \frac{r}{10\ \log_{10} r}$$

$\eta_r$ tells about the total number of symbols to be stored "with same information content", compared to decimal number system.

| r | 1 | 2 | 3 | 4 | 5 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|---|---|
| m/n | - | 3.32 | 2.10 | 1.66 | 1.43 | 1.11 | 1.00 | 0.93 |
| $1/\eta_r$ | - | 0.664 | 0.629 | 0.664 | 0.715 | 0.885 | 1.00 | 1.002 |

Maximum efficiency occurs for r = 3 , a *ternary* system.

Fig.1.2(b) shows input - output peripherals using different codes.

## Binary Number system

Any real number can be represented as

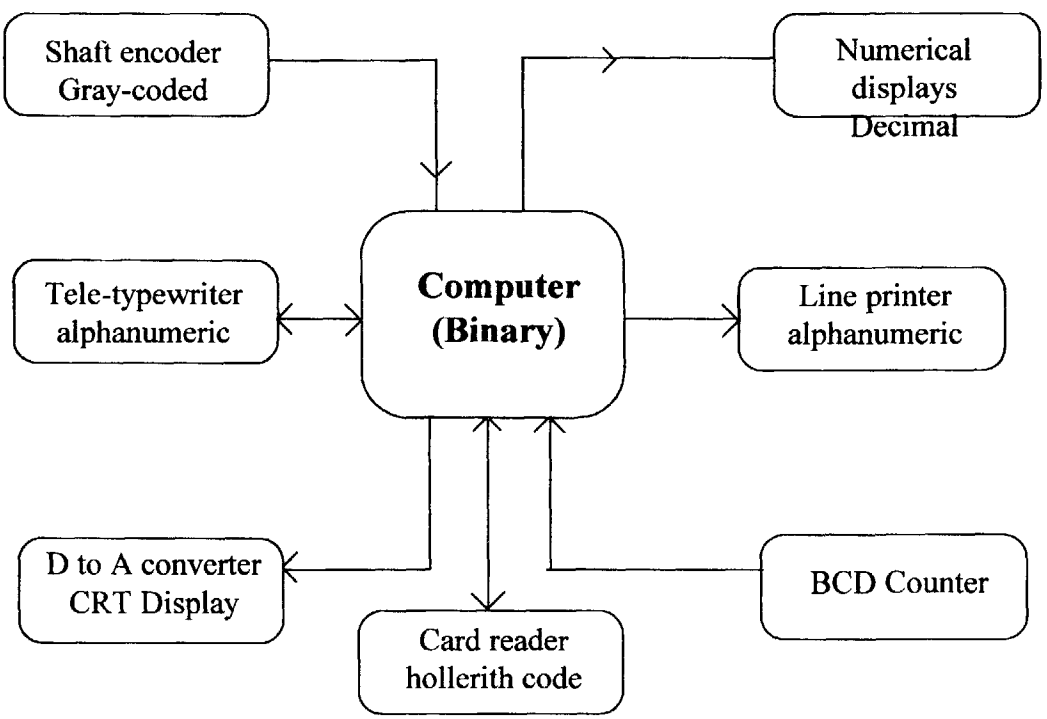



**Fig.1.2(b)** Peripheral equipments employing various codes

$i_{n-1}$ $i_{n-2}$ ..... $i_1$ $i_o$ . f1 f2 ..... fm

where the i bits symbolize the integer positions and the f bits denote the fractional component.

The representation assumes n integer positions and m fractional places to mean finally a decimal equivalent.

For example,

$$1011 = (1^* 2^0 + 1^* 2^1 + 0 * 2^2 + 1^* 2^3) = (11)_{10}$$

$$(0.11)_2 = 1 * 2^{-1} + 1^*2^{-2} = (0.75)_{10}$$

Thus by the previous generalised representation , any number in a number system with radix "r" can be converted to the equivalent decimal number by the formula.

$$\text{Decimal Equivalent} = (i_0 * r^0 + i_1 * r^1 + i_2 * r^2 + ..... + i_{n-1} * r^{(n-1)}) + (f1 * r^{-1} + f2 * r^{-2} + ... + fm * r^{-m})$$

which assumes n integer positions and m fractional places. The conversion from decimal to binary is shown with examples, separately for integer and fractional parts.

As a rule,

For a decimal integer to be converted to a number system with base 'r' , divide the given number by r and retain the remainders until you arrive at a quotient of 0, where every quotient is derived by successively dividing the previous quotient by "r" . The remainders taken in reverse order gives the integer equivalent in the number system with base "r".

1. $(10)_{10} = (1010)_2$

| | | |
|---|---|---|
| 2 | 10 | |
| 2 | 5 | – 0 |
| 2 | 2 | – 1 |
| 2 | 1 | – 0 |
| | 0 | – 1 |

2. $(0.75)_{10} = (0.11)_2$

$$0.75 * 2$$
$$1.50 * 2$$
$$1.00$$

Next, consider the fractional portion. Multiply the given decimal fraction by the base "r" which is of interest. Proceed multiplying the propagated fractional part only by 'r' until either you arrive at .00 fractional value or stop with 'm' fractional digits by doing only m times the above * process, where m denotes the stored arithmetic capacity of the number system "r".

***Defns:-*** *"A byte is a set of 8 bits and used more for memory standards".*
*"A nibble is a set of 4 bits".*

In terms of coding for ease of user handling, a set of 4 bits usually means an Hexadecimal digit (system with base 16), is normally acceptable in microprocessor based systems. Similarly, putting 3 bits together in all different ways ( here = 8) will generate an *octal code* having the unique symbols 0,1,2,3,4,5,6 and 7. The octal system is a standard in time shared unibus systems on an interactive environment for machine crunching and valid diagnostics. Table 1.1 depicts the encoding of octal digits.

**Table 1.1** Octal Coding

| *Basic Symbols* | *Binary equivalent* |
|---|---|
| 0 | 0 0 0 |
| 1 | 0 0 1 |
| 2 | 0 1 0 |
| 3 | 0 1 1 |
| 4 | 1 0 0 |
| 5 | 1 0 1 |
| 6 | 1 1 0 |
| 7 | 1 1 1 |

Table 1.2 shows the hexadecimal number system meaning more to byte crunching arithmetic processors. This type of encoding ( which is essentially a full encoder) fits best with straight binary processing.

## B.C.D Code

A much more practical and straight forward decimal system is accepted for real - time digital display systems. Also provision of independent decimal arithmetic is living in the area of data processing. Thus the IBM 360/370 computers having a

**Table 1.2** Hexadecimal Code

| *Symbols* | *4 bit code* |
|---|---|
| 0 | 0 0 0 0 |
| 1 | 0 0 0 1 |
| 2 | 0 0 1 0 |
| 3 | 0 0 1 1 |
| 4 | 0 1 0 0 |
| 5 | 0 1 0 1 |
| 6 | 0 1 1 0 |
| 7 | 0 1 1 1 |
| 8 | 1 0 0 0 |
| 9 | 1 0 0 1 |
| A | 1 0 1 0 |
| B | 1 0 1 1 |
| C | 1 1 0 0 |
| D | 1 1 0 1 |
| E | 1 1 1 0 |
| F | 1 1 1 1 |

separate decimal instruction set with COBOL language for data processing is a boon for throughput enhancement, on multiprogramming multiuser batch machines.

Table 1.3 gives the BCD code wherein the nibbles 1010 through 1111 are not utilised. Thus, this partial encoding method adds weightage in the final output in the domain of data base management systems as well.

**Table 1.3** BCD Code

| *Decimal Digits* | *Valid bit pattern* |
|---|---|
| 0 | 0 0 0 0 |
| 1 | 0 0 0 1 |
| 2 | 0 0 1 0 |
| 3 | 0 0 1 1 |
| 4 | 0 1 0 0 |
| 5 | 0 1 0 1 |
| 6 | 0 1 1 0 |
| 7 | 0 1 1 1 |
| 8 | 1 0 0 0 |
| 9 | 1 0 0 1 |

The excess - 3 code has been applied as intermediate processing stage for pure decimal arithmetic, which is obviously a self complementing code.

### BCD to seven segment LED code conversion

The visual representation of numeric - data in real-time applications calls for a device display. The seven segments is either directed by Hardware considerations such as common cathode or anode LED and the driving hardware. This is fixed design and also needs to generate a combinational logic for the chip implementation on circuit boards. This is costly and of course is fast.

Alternatively, the table look-up technique can be used effectively to drive the seven segments in real-time with decoding software of the BCD number dynamically. The continuous electronic weighing machines, the LED readout of petrol drawn at bunks, the digital count on a video cassette player serve as application examples wherein the seven segment display has become common.

Also the available combination of segments( $2^7 = 128$) can be used exhaustively if some coding can be effective and secured in communication systems in the topics of light ( Optical communication systems).

## 1.3 Non-weighted Codes

Gray or reflected codes: This is an unweighted code not suited to arithmetic operations. Table 1.4 gives the example for a 3 -bit gray code. Here the main concern is of all the 8

**Table 1.4** Gray code for 3 bits

| | |
|---|---|
| 1st symbol | 0 0 0 |
| 2nd symbol | 0 0 1 |
| 3rd symbol | 0 1 1 |
| 4th symbol | 0 1 0 |
| 5th symbol | 1 1 0 |
| 6th symbol | 1 1 1 |
| 7th symbol | 1 0 1 |
| 8th symbol | 1 0 0 |

Symbols are in sequence having the reflection property

possible codewords, the second set of 4 code words will be reflected mirror image of the 1st set of 4 code words but for only the most significant bit, with the additional property that when going from one codeword to the next in sequence there is only a single bit difference among the code words in sequence closing as a ring.

Probably one of the potential applications of gray code is organised in the area of Veitch - Karnaugh map construction and Quine-Mc clusky tabulation procedure for single and multiple output functions in combinational logic design for a better management based on the well-established Boolean logic minimization rules. The programmable logic arrays play a dominant role as hardware blocks for programmed logic implementations.

But, in general, the construction of the reflected code is left to the users for flexibility in the domain of encoding and decoding. These codes find applications in I/O devices, A to D converters and peripheral equipments.

Figure 1.3 shows the procedures for binary to Gray and Gray to binary conversion with examples.

### Binary to Gray

To get the gray equivalent: Repeat the msb; add the first two binary digits from left to right to obtain the next gray digit; continue adding each pair of adjacent bits to arrive at the gray equivalent.

### Gray to binary

Gray code is also called reflected binary code. Repeat the most significant digit; add diagonally as shown to get the next binary digit; continue adding respective diagonals to attain the remaining bits.

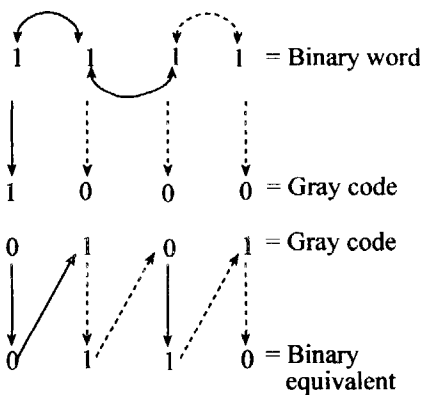



**Fig 1.3** Gray binary conversions

### Morse code

It is a ternary code with dot, dash and space. The space between dot and dashes in a single letter is one unit of time. The space between two letters is ( three time units). The space between two words is six units. Dash is three times the length of a dot. Still used in telegraphy for manual communicating ends, has the obvious decoding difficulties for automation.

For decimal applications, the 2 out of 5 Von Dieuren code is weighted except for decimal 'O' as shown in Table 1.5

The Morse code has almost completely been replaced by *Von Dieuren code* which uses 3 out of 7 positions filled with 1's ( binary code) meaning $7C_3 = 35$ possible code vectors.

**Table 1.5** Von Dieuren Code

| *Decimal Value* | *2 out of 5 code* |
|---|---|
| 0 | 00011 |
| 1 | 11000 |
| 2 | 10100 |
| 3 | 01100 |
| 4 | 10010 |
| 5 | 01010 |
| 6 | 00110 |
| 7 | 10001 |
| 8 | 01001 |
| 9 | 00101 |

As the use of the number crunching machines picked up ( 1970's ) with the third generation of operating systems, a need was felt to have an on-line user for the computer as a campanion. For well established reasons, the standard Keyboards have got adapted to all computer manufacturers inviting the use of the following standard coding schemes:-

*ASCII*: American standards committe on Information Interchange is a 8-bit code that includes 1 parity bit for interactive environments with flexible keyboard models.

*EBCDIC:* Extended Binary Coded Decimal Interchange code (*eb-si-dik*) is a standard 8-bit code ( byte orientation) followed by the IBM batch machines.

The standards of ASCII and EBCDIC has got established due to enhanced reliability with syntax oriented languages and a purposeful semantic based machine in spite of little redundancy, to match the memory specification of BYTE.

### *Huffman code*

This is a variable length code to represent data and text. This is not a uniformly spaced code. The length of a symbol is decided by the frequency of occurrence (usage) of that symbol, the more used having less number of bits. They should be uniquely decodable, meaning no code symbol is a prefix of any other code word in the domain of machine opcodes. This essentially is used for inherent data compression and also for an embedded pipelining at execute times. Huffman has given a systematic procedure for forming the code set.

Finally the process of encoding is a continuous affair in symbol representation and message passing which has a strong theoretical base with practical implementations for secure and reliable fault tolerant systems discussed in Chapter 5 Next we pass on to the processing elements, that is, the arithmetic logic unit.

## 1.4 Arithmetic Logics

Man started counting, proceeded to calculate and rose up to computing. A computer takes the reversal phase for its powerful processing job, viz. it calculates the task planning both statically and dynamically based on the events count in order to contribute to the Thru-put.

The universally accepted decimal number system is the expected arithmetic input and output of a computer. Though the electronic machine is purely based on binary logic ( ie. the two elementary symbols 0 and 1), it has to cater to varying arithmetic needs. The rule for binary addition and the equivalent binary number for the decimal numbers 0 to 9 is shown in Figure 1.4 (a) and Table 1.6 respectively. Figure 1.4(b) gives for binary subtraction.

The symbols A and B stand for the binary single bit operands. A can take a value either 0 or 1 and similarly B can assume any one of the values 0 or 1. Fig. 1.4(c) depicts

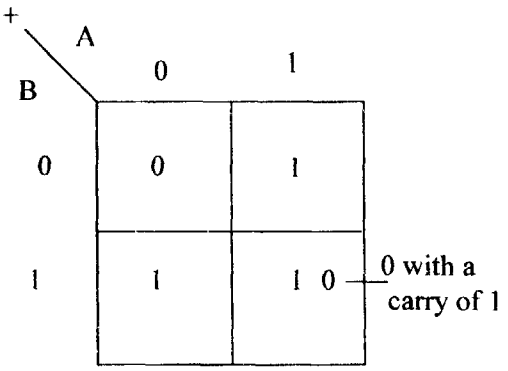



**Fig. 1.4(a)** One bit addition

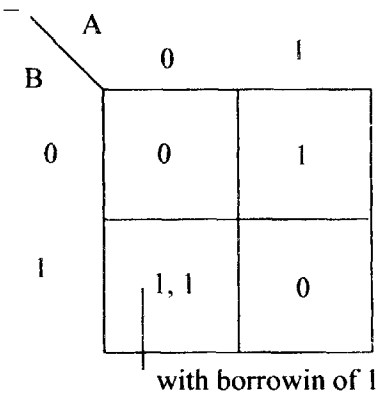



**Fig. 1.4(b)** One bit subtraction

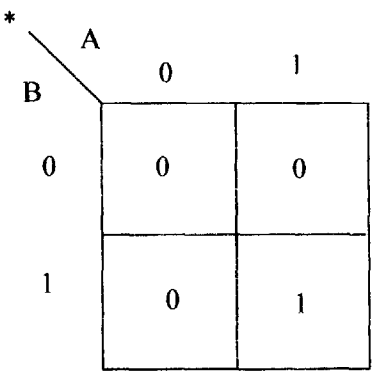



**Fig. 1.4(c)** Binary multiplication

**Table 1.6** Packed binary

| *Decimal Number* | *Equivalent binary* |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 1 0 |
| 3 | 1 1 |
| 4 | 1 0 0 |
| 5 | 1 0 1 |
| 6 | 1 1 0 |
| 7 | 1 1 1 |
| 8 | 1 0 0 0 |
| 9 | 1 0 0 1 |

the binary multiplication rule.

Charles Babbage considered to be the father of digital computers built up the analytical engine. The basic arithmetic operations can easily be realized by using the logic circuits first designed by Claude Shannon in 1930's following the invention of boolean algebra by George Boole in 1850's. Lady Lovelace is supposed to be the first programmer belonging to the 19th century A.D.

The major advantages of using binary arithmetic are:

i) Well isolated logic values 0 and 1 with a good noise margin;
ii) High-speed switching achievable: and
iii) An unambiguous and fast decision to match the control complexity in central processing units.

## Number Representation

The number representation in computers is much decided by the storage registers available in the CPU and also the wordlength of the processing element. Usually, the integers occupy length varying from 1 byte to no. of bytes based on the wordlength ( processing power available). Generally, most of the standards treat 8 bits equal to 1 byte and exceptionally, certain manufacturers take 6 bits as a byte for their range of products. Fig. 1.5(a) gives the straight binary representation for fixed point arithmetic assuming one sign bit for a 8 bit register storage. The maximum number is $+ ( 2^7 ) = + 127$ and the minimum value is $- ( 2^7 ) = - 128$ with 8 bits in the sign magnitude scheme.

In general, r's and ( r–1) 's complement are defined for a number system with base "r". Accordingly for representing negative numbers, one can identify the one's complement and two's complement schemes. Fig. 1.5(b) depicts the representation of + 0 and – 0 with 1's complement scheme. The sign bit, by convention, is 0 for positive number. Thus, the ambiguity and unwanted values like – 0 and + 0 creep in with the first two schemes.

Fig. 1.5(c) gives the 2's complement representation for -128 thus having a unique representation for zero besides using all the 8 bit patterns. Table 1.7 gives the expression for the minimum number, in an n - bit with one bit reserved for sign in all the schemes, whereas the maximum decimal value is $2^{(n-1)}$ -1 in all cases.

**Table 1.7** Accomodated minimum value with differing notations

| *Scheme* | *minimum number with n bits* |
|---|---|
| signed magnitude | $-(2^{n-1}-1)$ |
| One's complement | $-(2^{n-1}-1)$ |
| Two's complement | $-(2^{n-1})$ |

## Floating point numbers

In most of the compute-bound numerical problems, we have to handle real numbers,

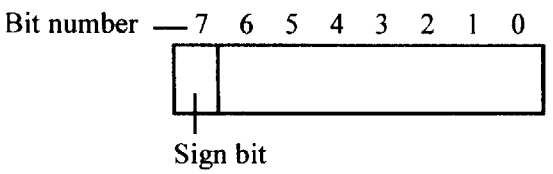



**Fig. 1.5(a)** Signed magnitude form for integer format

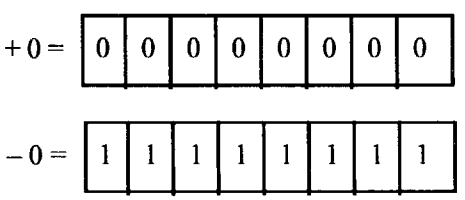



**Fig. 1.5(b)** One's complement scheme

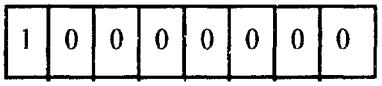



**Fig. 1.5(c)** 2's complement representing –128 in one byte

that is, one having also a fraction part, together accounting the sign.

For example

$(1101.01)_2 = (13.25)_{10}$

When it comes to representing the decimal fraction ( always < 1.0) , it is more likely we may not get the exact equivalent due to restriction of storage capacity and not converging necessitating either truncating or rounding off of the actual values. In accounting procedures, the rounding or truncation must be the last step in final solved arithmetic values. Thus generation and propagation of errors is really felt with the unique method of real number representation.

## Floating point notation

In order to avoid this sensitive handicap and also accommodate a fairly good range of values the floating point notation standard is followed.

**Examples are:**

$$7.3 \times 10^4 = 0.73 \times 10^5$$
$$.033 \times 10^7 = 0.33 \times 10^6$$
$$5.47 \times 10^{-11} = .547 \times 10^{-10}$$

In normalised mode, the number takes the form $m \times b^e$, where m, the *mantissa* (also called fraction) satisfies the relation $1/b \leq m < 1$, b denotes the base of the number system and "e" the *exponent* value (Characteristic).

**Examples are:**

$$0.11011 \times 2^{12}$$
$$0.100 \times 2^{-5}$$

An exception which cannot be normalized is a zero value, 0.0.

In typical IBM system, 24 bits are used for mantissa which includes 1 sign bit and the next 8 bits for the biased exponent. The constant which is added to the true exponent is determined (here, the constant is 128) so that the minimum possible integer is raised to zero. The biased value varies from 0 to 255.

The Range is dictated by the exponent value and accuracy is spelt by the Mantissa part.

Though languages like Pascal and COBOL ( for structured data) allow for integer and real arithmetic, *FORtran* has always dominated for floating point arithmetic accommodating the I, F and E formats besides giving more flexibility for user definitions in a dynamic way. FORtran is still treated as a special arithmetic tool for scientific and engineering problems.

For much more precision and accurate data IBM batch machine supports double precision arithmetic needing 64 bits for numeric data. In real time applications to accommodate standard practices, character strings are prefixed or suffixed to meet data processing as well as readability. As example, $ 1345 millions account tallying so many numbers in the respective denominations of currency in banks, statistical data for generating various types of graphs with attached graphics in interactive environments are worth to mention.

Especially for the process of addition and subtraction, the exponents has to be aligned which may often mean an approximate solution in critical application areas. Whereas, the floating point notation is comfortable with multiplication and division operations. All computers are well designed for establishing the hierarchy of operators in evaluation of arithmetic expressions in order to manage an accurate result.

So to fit to this domain of arithmetic processing, the stack architecture is an invited guest to host the ongoing processes.

Since the capability of a machine is application-bound, most of the minimal microprocessor based systems provide only for addition and subtraction only deriving the * and / operations by software for Real-time systems. Exhaustively, a computer may have different functional units independently for each operation to support pipelining as well as parallel processing trends.

Many libraries of functionally proved softwares reside with the system in the area of maths and graphics for dedicated users to enhance productivity. Users can include the necessary libraries only for their jobs, in a timeshared system like the UNIX interactive environment.

Though the central processing units have enormous power of number crunching, the intelligent method for a real optimized solution is only through *Algorithms* which necessarily have to be programmed to the capability of machine architectures.

The 2's complement is the radix complement system ( RC) in the case of binary logic. Especially for the process of subtraction, one can at best make use of the 2's complement representation which, of course, necessitates

(i) generating the 2's complement of the operands on the one hand which is easily realized using logic circuits, but at the same time,

(ii) the availability of adder circuits alone is sufficient which is a boon in the construction of the ALU best fitting to the Von Neumann architecture following the fundamental Flynn's configuration of SISD (single instruction single data).

Fig. 1.7 illustrates with examples, the procedure for *Two's complement arithmetic* to perform A + B for all possible cases.

The procedure clearly illustrates the applicability of only a positive zero (+0) for real arithmetic with computer systems.

Before further converging into arithmetic processing units in 1-7, we shall visit the powerful logical elements in the following section 1-5.

## 1·5 Boolean Algebra

Boolean Algebra ( Symbolic logic) is named after George Boole ( 1815 - 1864) who developed it for describing logical statements. This remained a subject of pure mathematics until 1938, when Claude Shannon applied this to switching circuits. Today it is the backbone of computer circuit analysis and design, a totally different application than Boole intended.

### Fundamental posulates

(i) A = + 3; b = + 3
A + B = 0, 011
0, 011
0, 110 = + 6

(ii) A = + 3 ; B = – 2
A + B = 3 + (–2) = 0, 011
(1,010) 1, 110
"Carry discarded" ~~1~~ 0, 001 = + 1

(iii) A = + 2 ; B = – 3
A + B = 2 + (–3) = 0, 010
1, 011 1, 101
"sign of result" 1, 111 magnitude in 2's complement form
= – 1.

(iv) A = 3 ; B = 3

A − B = A + (−B) = 3 + (−3)

0, 011

1, 101

Carry discarded ↓ 0, 000 = + 0

**Fig. 1.7** 2's complement procedure for add and subtract

$A \neq 0$ iff $A = 1$

$A \neq 1$ iff $A = 0$

*"A boolean variable (A) in the true form (a) or in the complemented form (a) is called a literal".*

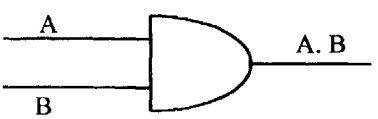
A boolean Algebra, is an algebra ( B; . , + , 1; 0, 1) consisting of a set S.

(which contains atleast two elements 0 and 1) together with the three operations, the AND ( Boolean product ) operation . , the OR ( Boolean sum) operation +, and the NOT ( complement) operation' defined on the set, such that for any element A, B and C of the set S, A.B, A+B, $A^1$ are in S.

SWITCH A SWITCH B

Switch closed = True = 1

Switch open = False = 0




| *A* | *B* | *A.B* |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

The AND operation produces the carry when two bits are added which forms part of adder circuits. Binary multiplication also involves the AND function.

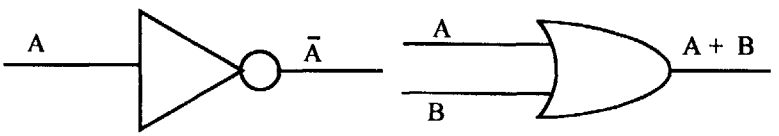
It is also called as ALL gate.

NOT function, or complement, is the simplest logicalfunction for the unique reason it has only one input. NOT inverts logic levels.

| *A* | $\overline{A}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

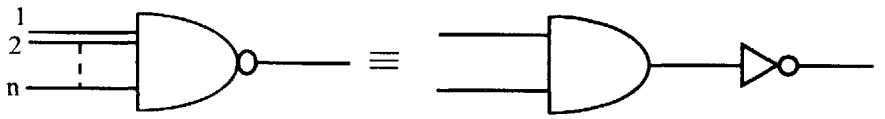
This *NOT function* is useful in generating 1's complement of a number.

The OR function can be described in terms of parallel controls. In this sense, it is also called Any gate. (inclusive - OR).



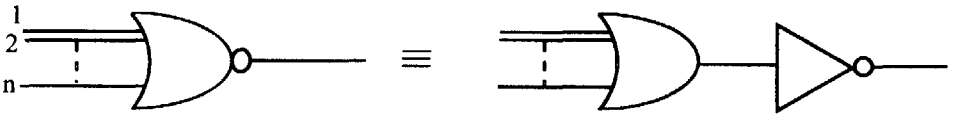

| *A* | *B* | *A + B* |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

A NAND gate is one, that contains AND logic followed by NOT gate.




n - input NAND

A NOR logic is one, that contains OR gate followed by an inverter gate.




n - input NOR gate

The NAND and NOR gates are called as universal Logic elements since they are easy for fabrication and also give a good true yield in experience, besides generating any combinational logic function. Fig.1.8 (a) depicts the basic logic gates discussed above.

Boolean algebra can be used to display the concept of the logic of sets by means of Venn diagrams. See Fig. 1.8(b). A universal set is a set containing every object satisfying certain properties.

Complement of a set A, designated as A contains those objects found in the universal set but not in set A. A null set is a set containing no objects. Set of postulates ( axioms) were suggested by *Huntington* in 1904.

The first property which drastically differs from the algebra of real numbers, and accounts for the special characteristics of switching algebra is the

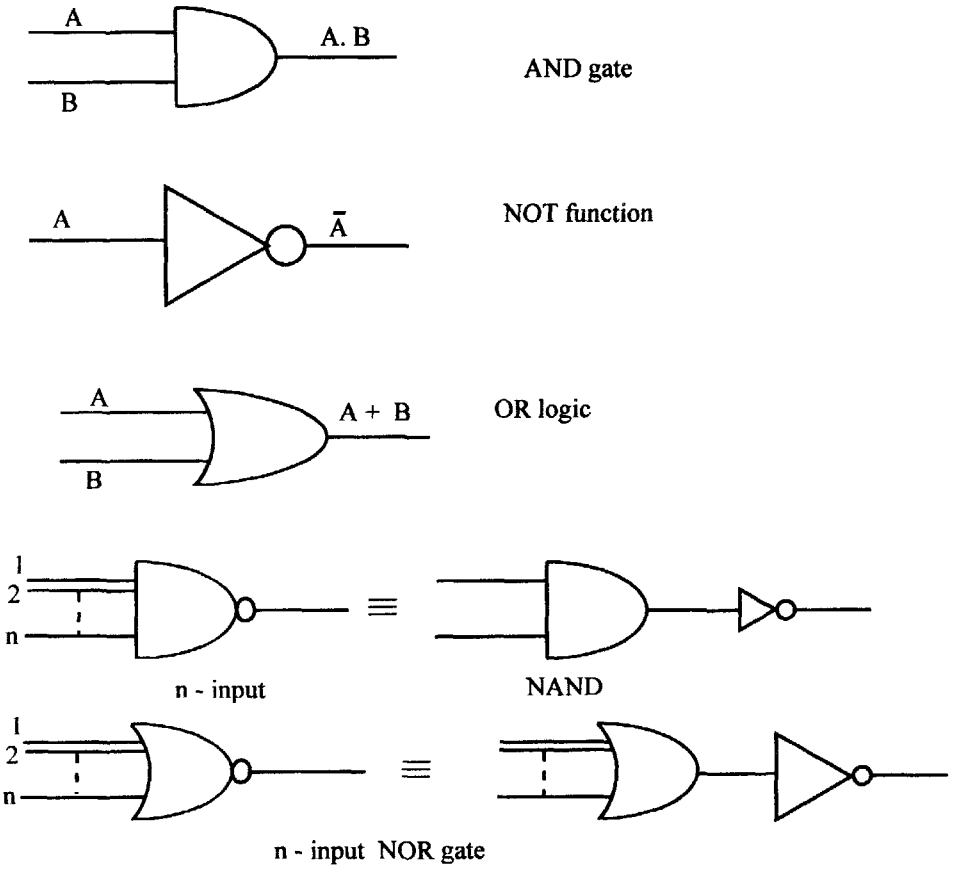



**Fig. 1.8 (a)** Basic logic gates

1. Idempotence Law.

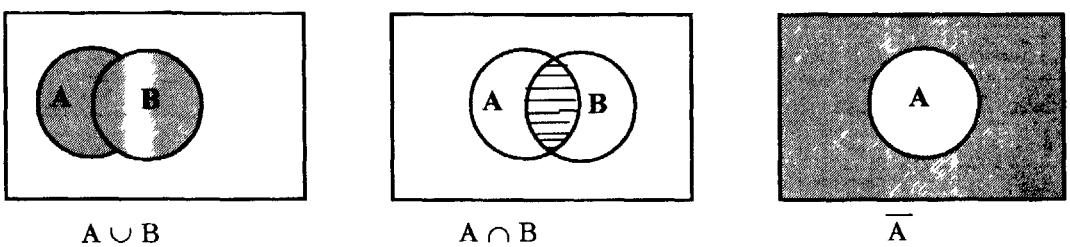



**Fig. 1.8 (b)** Venn diagrams

$$a.a = a$$
$$a + a = a$$

where a is two - valued variable.

2. There exists a unique multiplicative identity ( the one element) 1, such that for each $a \in s$,

$$a.1 = 1.a = a$$
$$a + 1 = 1$$

There exists a unique additive identity ( the zero element) $0 \in S$ such that for each $a \in S$,

$$a + 0 = 0 + a = a$$
$$a.0 = 0$$

3. The operations + and . are commutative.

$$a + b = b + a$$
$$a . b = b . a$$

4. Associative Law.

$$a + b + c = (a + b) + c = a + (b + c)$$
$$a . b . c = (a . b) .c = a . (b . c)$$

5. Each operation + and . is distributive over the other.

$$a. (b + c) = (a . b) + (a . c)$$
$$a + (b . c) = (a + b) . (a + c)$$

6. Complementation.
   For every element a in 'S', there exists a complementary element a or a' such that

$$a + a^1 = 1$$
$$a . a^1 = 0$$

7. Absorption Law.

$$a . (a + b) = a$$
$$a + (a . b) = a$$

   Law of cancellation does not hold good

8. Involution law.

$$(a^1)^1 = a$$

The above postulates exhibit a duality relation. Principle of duality can be stated as "every theorem which can be proved for boolean algebra is transformed into a second valid theorem if the operations ( + ) and ( . ) and the identity elements 0 and 1 are interchanged throughout."

From absorption law,
i.e. if $a \cup b = a \cup c$
then $b \neq c$ in general (law of cancellation).

Table 1.8 Summarises some boolean theorems.

"Every identity of boolean algebra has its dual". Positive and negative logic rise to a basic *duality* in all the identities. We can produce a dual identity by changing OR and AND and vice versa, changing 1s to 0s and 0s to 1s in the original expression. *DeMorgan's theorem* is a derivation of this property of duality.

## Complement of a function

"If in a boolean function the +'s are changed to .s and .s to +'s and every element is replaced by its complement then the resulting function and the original function are complements of each other".

***Defn:*** *"When two or more (finite number of switching variables) elements of a boolean algebra are connected by one or more of the three operators, + . ', the*

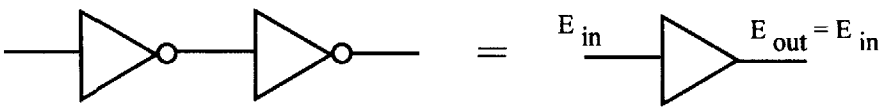
**Table 1.8 Laws of Tautology**

| | |
|---|---|
| Idempotent | X + X = X<br>X . X = X |
| Commutation | X + Y = Y + X<br>XY = YX |
| Association | (X + Y)+ Z = X + (Y+ Z)<br>(XY) Z = X (YZ) |
| Distribution | X + YZ = (X + Y)(X + Z)<br>X (Y + Z)= XY + XZ |
| Absorption | X + XY = X<br>X (X+Y) = X |
| Universe Class | X . 1 = X<br>X + 1 = 1 |
| Null Class | X . 0 = 0<br>X + 0 = X |
| Complementation | X + X' = 1<br>X X'= 0 |
| Contraposition | X = Y'<br>implies Y = X' |

| | | |
|---|---|---|
| Double negation (involution) | | X = X" = (X')' |
| Expansion | | XY + XY' = X |
| | | (X + Y) (X + Y') = X |
| **Laws of** | De Morgan | X + Y = X' Y' |
| | | XY = X' + Y' |
| | Reflection | X + X'Y = X + Y |
| | | X (X' + Y)= XY |
| | Transition | XY + YZ + X'Z = XY + X'Z |
| | | (X+Y)(Y+Z)(X'+Z) = (X+Y)(X'+Z) |
| | Transposition | XY +X'Z = (X+Z)(X'+Y) |
| | | (X+Y)(X'+Z) = XZ + X'Y |

*resulting expression is called a Boolean function".*

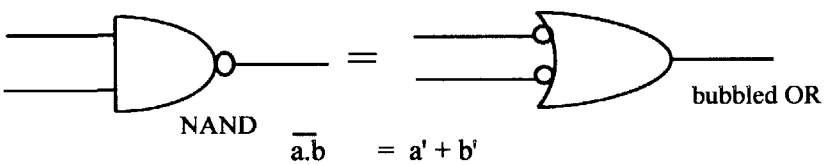
If two NOT gates are cascaded, we get a buffer or a noninverting amplifier. It is like an emitter follower used whenever it is necessary to increase the impedance level.



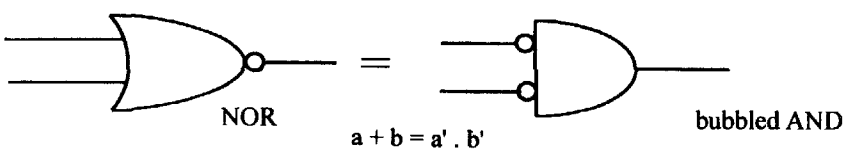

Augustus De Morgan was the first to acclaim Boole's great achievement.

**De Morgan's Theorem:**

(i) "The complement of a product equals the sum of the complements".




(ii) "The complement of a sum equals the product of the complements".



**The Consensus Theorem**

It is frequently used in the simplification of switching expressions (used to eliminate redundant literals).

$a\,b + a^1 c + b\,c = ab + a^1 c$
$(a+b)(a^1 + c)(b + c) = (a + b)(a^1 + c)$

*Proof*
$a\,b + a^1 c + bc = a\,b + a^1 c + bc \cdot 1$
$= a\,b + a^1 c + bc(a + a^1)$
$= a\,b(1 + c) + a^1 c(1 + b)$
$= a\,b + a^1 c$

Dual is:
$(a + b) \cdot (a^1 + c) \cdot (b + c) = (a + b) \cdot (a^1 + c)$

This enables us to convert an expression into an equivalent one with fewer literals, where, by literal we mean an appearance of a variable (in its true or complement form).

## 1·6 Function Minimization

Since the ALU is mainly constructed using the Logical circuits of AND OR and NOT, the minimization of Boolean functions is of primary importance which minimises the cost besides meeting the VLSI design and implementation for computing.

The application of various theorems and postulates discussed in 1.5 can serve this purpose. But with the current day technology of computer aided minimization, (involving PAL'S and PLA'S ), in what follows we will touch on (i) the Karnaugh map method (ii) the Quine- Mc Clusky tabulation method for combinational circuits design.

A function can be represented on a map. Veitch ( 1952) suggested such an idea and the same was reorganized by Karnaugh ( in 1953).

> ***Defn:*** *"The Boolean product of two or more literals, in general, is called a product term. (conjunction)".*

A variable cannot appear more than once in a *product term*.

*Example* $a\,a\,b = (aa)\,b = a.b$

$a\,a'\,b = (aa')\,b = 0.b = 0$

"The Boolean sum of two or more literals it called a sum term (*Disjunction*)".

"When a Boolean function appears as a sum of several product terms, it is said to be

expressed as a sum-of-products. The SP form is also known as the Disjunctive Normal Form (DNF)".

*Example* $f(A, B, C, D) = a^1 + b c^1 + cd$

"When a Boolean function appears as a product of several sum terms, it is said to be expressed as a product-of-sums. The PS form is also known as the conjunctive Normal Form. ( CNF)".

*Example* $f(A, B, C, D) = (a + b) . (b + c' + d)$

"A product term of n variables is called a minterm of n variables".

*Example* $a^1 b^1 c^1$, $abc^1$ are minterms of three variables.

Each minterm is designated $m_i$, where the subscript 'i' denotes the decimal value of the binary number.

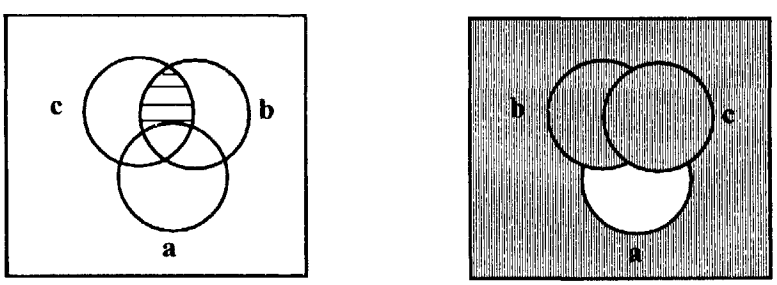
"A sum term of n variables is called a maxterm of n variables".

*Example* $(a^1 + b + c + d^1)$, $(a + b + c + d)$ are maxterms of (Each Maxterm is designated $M_j$ ).4 variables.

"The total number of all possible minterms of n variables is $2^n$". It is evident that $\overline{m}_i = M_i$ and $\overline{M}_j = m_i$.

For $n \leq 3$, a minterm can be represented on a Venn diagram.

It is interesting to note that a *minterm* occupies occupies a minimum area on the *Venn diagram*, whereas a Maxterm occupies the maximum area. (Refer Fig.1.9)




$m_3 = a^1 b c$ $\qquad$ $m_4 = a^1 + b + c$

**Fig. 1.9** Venn diagram representation of $m_i$ and $M_j$

***Defn:*** *"When each of the terms of a Boolean function expressed either in the SP or PS form has all the variables in it, then the function is said to be expressed in the Canonical form".*

Here also by idempotence Law, no term can appear more than once. The canoanical SP form is called the disjunctive canonical form ( DCF) also known as *standard sum form*. The canonical PS form is called the conjunctive canonical form (CCF) also known as standard product form.

Examples: DCF: $f(A, B, C) = ab^1c + a^1b^1c + abc$

CCF: $f(A, B, C) = (a+b^1+c) \cdot (a^1+b^1+c^1) \cdot (a+b^1+c^1)$

*"A Boolean product term of K variables can be expressed as a sum of $2^{n-k}$ product terms of n variables. $(n > k)$".*

Lemma: A boolean product term of K variables can be expressed as a sum of 2 product terms of ( k + 1 ) variables.

***Corollary***:

The identity element 1 can be expressed as a sum of $2^n$ product terms of "n" variables.

***Proof***:

$1 = a + a^1$

Now both a and $a^1$ can be expressed as sum of $2^{n-1}$ product terms of 'n' variables.

Hence, 1 can be expressed as a sum of $( 2^{n-1} ) + 2^{n-1} )$ or $2^n$ number of product terms of 'n' variables.

**Theorem 2**

*"A Boolean sum term of K variables can be expressed as a product of $2^{n-k}$ $(n > k)$ sum terms of n variables".*

***Corollary***:

The identity element 0 can be expressed as a product of $2^n$ number of sum terms of 'n' variables.

**Theorem 3**

"Every Boolean function can be expressed in a canonical form".

One of the main objective of a logic designer is to realize a Boolean function with minimum cost.

A Karnaugh map is, actually, a modified form of a truth table in which the arrangement of the combinations is particularly convenient.

***Defn***: MSP (minimal SP form).

*(i) The two level AND - OR realization of the minimal sum of products form has a minimum number of AND gates.*

*(ii) NO AND gate can be replaced by an AND gate with fewer inputs.*

## VEITCH - KARNAUGH MAP

*V-K map* for n variables has $2^n$ cells, so that each cell correspond to one minterm. Theoretically, it is possible to draw a V-K map for any number of variables. In practice a map for more than 6 variables is seldom used, whereas 2,3 and 4 - variable maps are extensively used.

Fig.1.10 show the VK map for 2,3 and 4 variables.

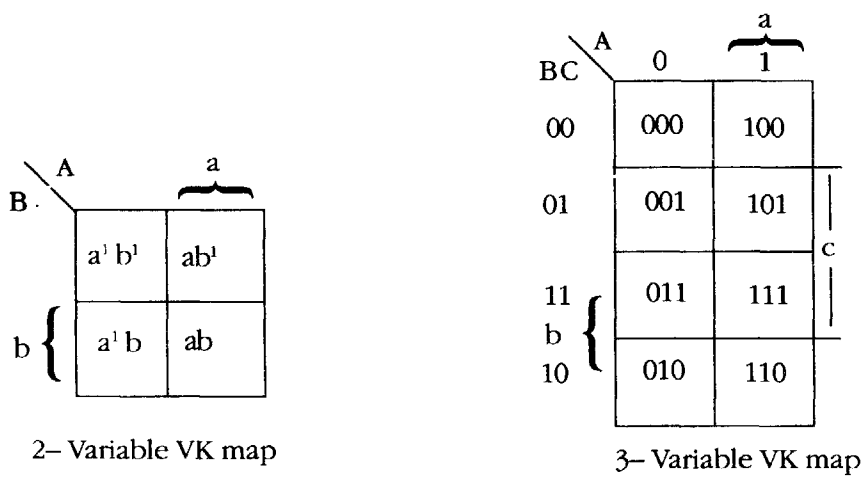



**Fig. 1.10** Veitch Karnaugh maps

In the 2 variable map of Fig. 1.10, A appears as a in the 1st column and as a in the second column.

Similarly B appear as b in the 1 st row and as b in the second row. Usually, columns and rows for only the true form of the variables are so marked, the rest are meant by implication. The 4 minterms $a^1 b^1$, $a^1 b$, $ab^1$ and ab are located as shown.

Each cell of the map can also be recognised by the decimal designation of the minterm it represents as shown in the 4 variable map of Fig.1.10.

4 - variable map has
$2^4 = 16$ cells

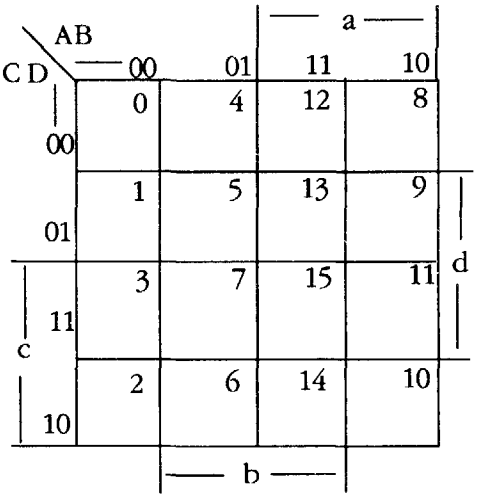



**Fig. 1.10** Veitch Karnaugh maps

*Defn: "Two rows/columns of a VK map are said to be adjacent to each other if their variable coordinates differ in one variable position only".*

If a boolean function is given in its *DCF* ( disjunctive canonical form), it can be represented on the map just by writing 1's on those cells for which the function is 1. (true).

Examples: Represent the following functions on a VK map.

(i) $f(A,B,C) = \cup(0,1,5,7)$

(ii) $f(A,B,C,D) = \sum_{m_i} 2,4,6,9,11,12,15$

(Refer Figure 1.11)

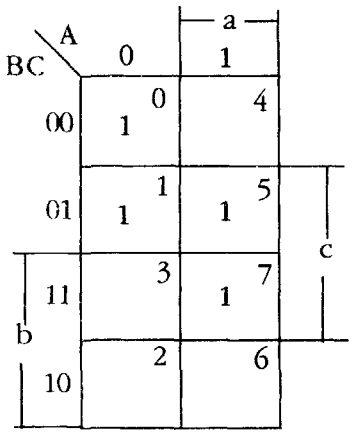
A cluster of four 1's represents a product term of two variables on a 4 variable map. Thus, a two variable product term can be plotted on a 4 variable map by a cluster of four 1's shared by two adjacent rows and two adjacent columns. The same can also mean a single row or a single column.

For minimizing a function plotted on a VK map: the following procedure is adopted.

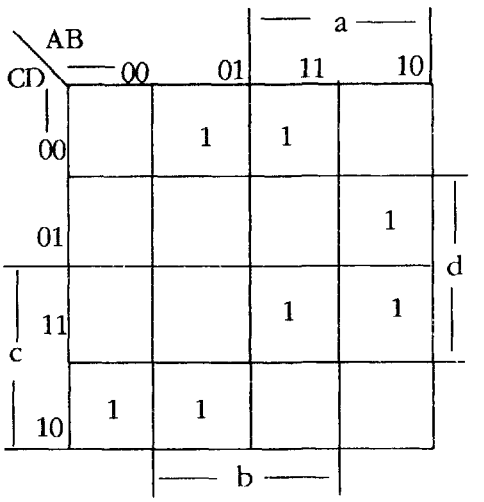
A pair (two 1's) eliminates
One variable;
A quad (four 1's) eliminates two variables:
An octet (eight 1's) eliminates three variables.

A set of $2^K$ 1's eliminates K variables from ( N and K are integers) an N variable function where, ( N >K). After construction of Karnaugh map, encircle the largest

**Solution**




f (A, B, C) = ∪ (0, 1, 5, 7)




f (A, B, C) = $\sum_{m_i} \Sigma(2, 4, 6, 9, 11, 12, 15)$

**Fig. 1.11** Representation of functions in VK map

group first. For example, the octets followed by the quads, then the pairs. In this way the full simplification results in an easier approach. While forming the group of 1's, the same set of 1's can be a subset of another group.

Inclusion relation:-

$b \geq a$ b covers a

For all a, $b \varepsilon s$,

$a \leq b$ iff $a.b = a$

$a + b = b$

*Defn: (1) A product term which implies a Boolean function is called an implicant of the given function.*

*(2) An implicant from which no literal can be removed without altering its implicant status is known as a prime implicant.*

The P Is have two specific characteristics:-

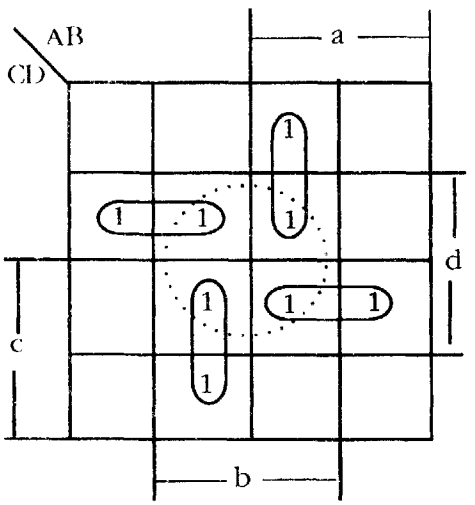
(i) They cannot be combined further to form simple products.
(ii) They are potential candidates in the final realization of the function, i.e the complete function is realizable in terms of the PIs alone.

1. Every implicant of a Boolean function implies a prime implicant of the function.
2. Every Boolean function can be expressed as a sum of prime implicants only.

***Defn:*** *When a cluster of 1's indicates a prime implicant on the map, each of the minterms represented by a 1 within the cluster is said to subsume the prime implicant, and the P.I. is said to cover each of these minterms.*

4. If among the minterms subsuming a prime implicant, there is atleast one which is covered by this and only this P.I., then the P.I. is an essential PI (E.P.I.)

5. If each of the minterms subsuming a P.I is covered by other E.P. Is, then the P.I. is a redundant P.I. ( RPI). as shown above (circled).




The terms $a^1 c^1 d$, $abc^1$, $acd$ & $a^1bc$ are all EPIs.

6. A P I., which is neither an essential nor a redundant P.I. is a selective P.I.(SPI).

***Defn:*** *When a Boolean function is expressed as a sum - of - products so that the total number of literals in the whole expression is minimum, the function is said to be expressed in its minimal sum-of-products (MSP) form.*

The MSP form of Boolean function,

(a) must contain all essential P.Is.

(b) must contain the subset with the smallest number of literals of the chain or chains of selective P.Is

(c) must not contain the Redundant P. Is.

Though it may be possible to factor a simplified sum of products equation and reduce the number of input gate leads, this is not normally done because it increases the overall propagation delay time and may cost more to implement the logic circuits with NAND gates. See examples.

e.g. It is desired to have a logic circuit to convert any 4 bit number in Gray code into equivalent BCD form. Express the binary digits.

| | BCD (BITS) |
|---|---|
| $G_3, G_2, G_1, G_0$ | $B_3, B_2, B_1, B_0$ |

as functions of the four gray code bits and minimize these four functions.

$B_0 = f_0\ (G_3, G_2, G_1, G_0)$

$B_1 = f_1\ (G_3, G_2, G_1, G_0)$

$B_2 = f_2\ (G_3, G_2, G_1, G_0)$

$B_3 = f_3\ (G_3, G_2, G_1, G_0)$

## The Tabulation Method

The V-K map method is very useful for functions of upto six variables.

(For programming done by a computing for functions of large number of variables, machine) the tabulation procedure, known also as the *Quine - MC Clusky* method of reduction is used.

Quine ( 1952); Mc CLUSKY (1956)

Mc Clusky improved ( reorganised the method) but was suggested by Quine.

### Examples

1. Simplify the function $f(A, B, C, D) = \sum_{m_i}(0, 2, 4, 6, 8) + \phi(10, 11, 12, 13, 14, 15)$

   Use Karnaugh map method. The function minimizes to

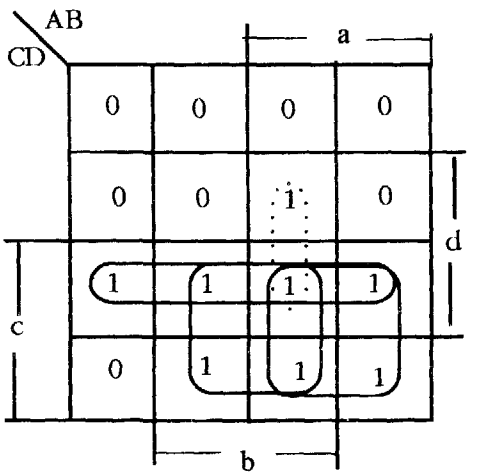
   $f(A, B, C, D) = a^1 d^1 + b^1 c^1 d^1$. With the use of don't cares,

   $f(A, B, C, D) = d^1$

The saving in the cost of design is obvious. Thus the don't care terms are useful in as much reducing the design cost.

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | ϕ | 1 |
| 01 | | | ϕ | |
| 11 | | | ϕ | ϕ |
| 10 | 1 | 1 | ϕ | ϕ |

2. What is the simplified boolean expression for the Karnaugh map shown below? What is the AND-OR network for the minimised function? If the equation is factored,what is the resulting logic circuit?




The minimal SP form is f (A, B, C, D)

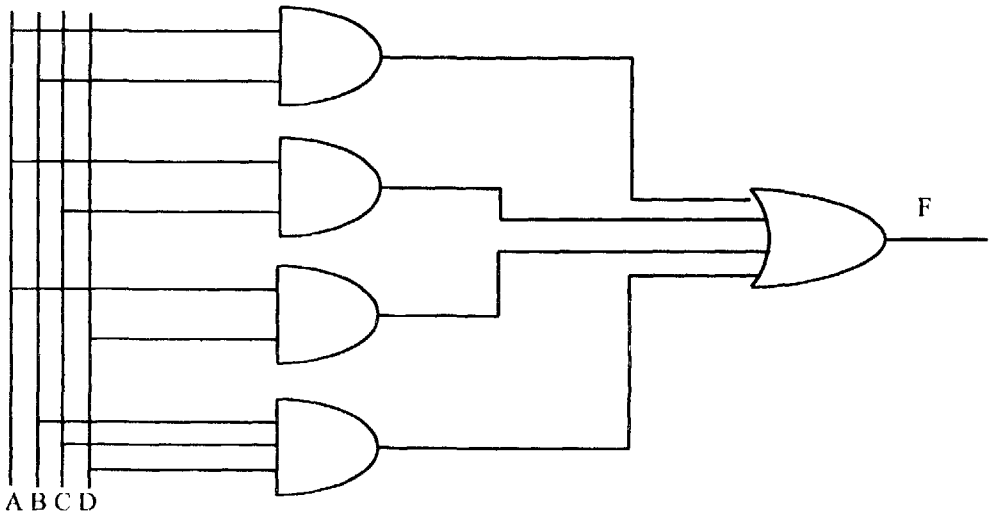
f(a, b, c, d) = ab + ac + ad + bcd { (logic circuit) } is on page 32

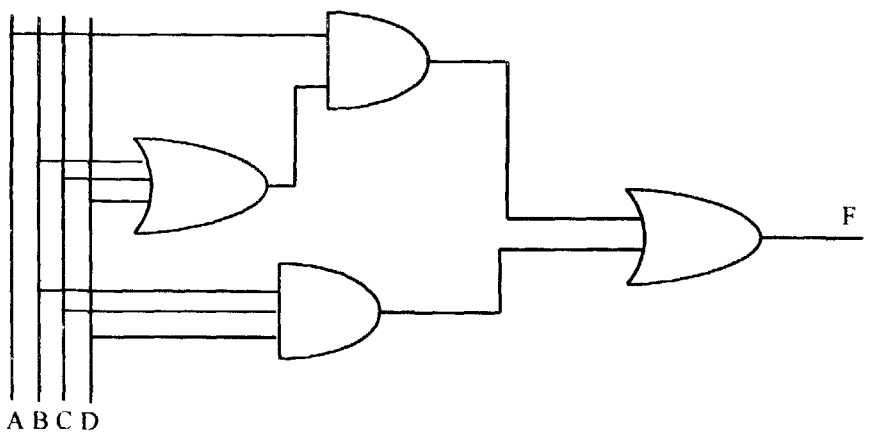
contains 13 input gate leads, has a smaller propagnation delay and can be converted to NAND - NAND network. Factored form is

f (A, B, C, D) = a ( b + c + d) + b c d { (logic circuit) } is on page 32

This has only 10 input gate leads.

It (Quine McClusky) is a specific step-by-step procedure that is guaranteed to produce a simplified standard - form expression for a function.It can be applied to problems with




many variables and has the advantages of being suitable for machine computation, first formulated by quine and later improved by McClusky Quine - Mc Clusky.

> ***Defn:*** *The (index) weight of a term is defined as the number of 1's in the binary designation of the term.*

Thus the weight of the minterm $m_6$ (0110) is 2. (written in binary form).

**Procedure**

(i) Arrange all minterms in groups, such that all terms in the same group have the same number of 1's in their binary representation. Start with the least number of 1's and continue with groups of increasing numbers of 1's.

The number of 1's in a term is called the "index" of that term. $S_a + S_a^1 = S$ (Combining theorem).

## Decimal Representation

Two minterms can be combined only if they differ by a power of 2, that is, only if the difference between their decimal codes is $2^i$ (0,1,8,9)

0 and 8 , or 1 and 9 can be combined.
$2^3\ 2^2\ 2^1\ 2^0$ they differ by $2^3 = 8$
A B C D 1,9 ( 8 )
0,8 ( 8 )

This process, which is recorded by placing the weight of the redundant variable in parantheses, e.g.1,9 (8), is nothing but a numerical way of describing the algebraic manipulation.

$a^1\ b^1\ c^1\ d^1 + a\ b^1\ c^1\ d^1 = b^1\ c^1\ d^1$.

Whenever a single literal is removed by the combination of two terms, the weights of the two terms must differ by only one.

So the entire procedure is, actually, a mechanised process for combining and reducing all adjacent pairs of terms. The unchecked terms are the P.Is of f, since each implies f and is not covered by any other term with fewer literals.

## Illustrations

$f(A, B, C, D) = \sum 0, 1, 2, 5, 7, 8, 9, 10, 13, 15$

minimise using Quine-Mcclusky technique

The tables and the prime implicants generated are given on the next page.

## Prime Implicant Chart

From this set we must select a minimal subset whose union is equivalent to f.

We go in for P.I. chart. The P.I Chart displays pictorially the covering relationships between the P.I.s and the minterms of the function. It consists of an array of U columns and V rows; U the no. of minterms for which the function takes on the value 1 and v the number of P.I.s. $i^{th}$ row Xs depict the minterms covered by the $i^{th}$ P.I.

The P.I chart is shown on next page.

**Table $T_0$**

| minterms | A | B | C | D |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 13 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 |

**Table $T_1$**

| | A | B | C | D |
|---|---|---|---|---|
| 0, 1 | 0 | 0 | 0 | – |
| 0, 2 | 0 | 0 | – | 0 |
| 0, 8 | – | 0 | 0 | 0 |
| 1, 5 | 0 | – | 0 | 1 |
| 1, 9 | – | 0 | 0 | 1 |
| 2, 10 | – | 0 | 1 | 0 |
| 8, 9 | 1 | 0 | 0 | – |
| 8, 10 | 1 | 0 | – | 0 |
| 5, 7 | 0 | 1 | – | 1 |
| 5, 13 | – | 1 | 0 | 1 |
| 9, 13 | 1 | – | 0 | 1 |
| 7,15 | – | 1 | 1 | 1 |
| 13, 15 | 1 | 1 | – | 1 |

**Table $T_2$**

| | | A | B | C | D | |
|---|---|---|---|---|---|---|
| 0,1 | 8,9 | – | 0 | 0 | – | = $P_1$ |
| 0,2 | 8,10 | – | 0 | – | 0 | = $P_2$ |
| 1,5 | 9,13 | – | – | 0 | 1 | = $P_3$ |
| 5,7 | 13,15 | – | 1 | – | 1 | = $P_4$ |

The terms labelled P1, P2, P3 and P4 in Table T2 cannot be further combined and therefore form the set of prime implicants of the given function.

| | 0 | 1 | 2 | 5 | 7 | 8 | 9 | 10 | 13 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|
| $P.I_1 = b^1c^1$ | x | x | | | | x | x | | | |
| $P.I_2 = b^1d^1$ | x | | x | | | x | | x | | |
| $P.I_3 = c^1d$ | | x | | x | | | x | | x | |
| $P.I_4 = bd$ | | | | x | x | | | | x | x |

Prime implicant chart for the function f (A, B, C, D)

$= \Sigma(0,1,2,5,7,8,9,10,13,15)$

The given four variable function above can be easily solved by Karnaugh map method. The reader is expected to verify the same. From the tick marks in Table $T_0$ the possibility of generating a term in table $T_j$ (where $j \leq n$), n being the number of variables of the function. In this problem, the degree of adjacency is between 2 and 3.

## P.I. Chart for function

If any column contains just a single x, the P.I. corresponding to the row in which this x appears is Essential, and consequently must be included in any irredundant expression. The x is circled and a check mark is placed next to the E.P.I., $P.I_2$. The row which corresponds to an E.P.I is referred to as an Essential row. Once, an essential P.I has been selected all the minterms it cover are checked off. $E.P.I._2$ covers 2, 10 also 0 & 8. Still, the minterms 1 and 9 are uncovered.

These minterms may be covered by either $P.I_1$, or $P.I_3$ and since both are expressed with the same number of literals, we obtain two minimal expressions for the function, viz.,

$f(A, B, C, D) = (b^1d^1 + bd + b^1c^1)$

(or) $= (b^1d^1 + bd + c^1 d)$

In P.I table, if there exists a column under which there is one and only X, then the row of this x is headed by an E.P.I.

Row $P.I_3$ dominated by $P.I_2$.

*Defn: In P.I table, if there is any row which has no* x *under any column, then the P.I. heading the row is a R.P.I. (redundant prime implicant).*

Don't care minterms need not be listed as column headings in the P.I. chart, since they do not have to be covered by the minimal expression.

e.g. f (A B C D E)

$\Sigma(13, 15, 17, 18, 19, 20, 21, 23, 25, 27, 29, 31)$

$+ \phi(1, 2, 12, 24)$

## The Adjacency Method - Biswas (1971)

In a method expounded by Biswas ( 1971) the need for P.I. tables is entirely eliminated.

Like the Q - M method, in this method also the function to be minimized is first expressed as a sum of minterms, and then the minterms are arranged as rows of a combination table in ascending order of their weights. The resulting table is designated as $T_0$. Successive tables are called $T_1, T_2$ etc. $T_1$ formed from $T_0$. $T_2$ formed from $T_1$. $T_n$ formed from Table $T_{n-1}$.

*Defn: Each time a term in Table To combines with another term in To to generate a term in $T_1$, an X is placed after each one of the two combining terms. After the formation of table $T_1$ has been completed, the number of Xs after a term in*

*Table $T_0$ is the degree of adjacency of the term.*

It can be easily verified that the *degree of adjacency* of a minterm of n variables lies between 0 and n, both values inclusive.

**Theorem 1**

If a minterm with a degree of adjacency X can generate a term in table $T_x$, then that term in table $T_x$ is an EPI.

Here all EPIs are detected in a Table $T_x$ by the application of the above theorem. The procedure for forming the tables is same as the Q–M technique.

**Summary**

One should recognize that the reflected code sequence chosen for the V-K maps is not unique. It is possible to draw a map and assign a binary reflected code sequence to the rows and columns differently. As long as the binary sequence chosen produces a change in only one bit between adjacent sequences, it will produce a valid and useful map.

"There exists an extension of the tabulation method for multiple - output circuits however, this method is too specialised and very tedious for human manipulation. It is of practical importance only if a computer program based on this method is available to the user".

The numerical algorithms for minimization involving heavy logical computations and decisions will fit in for computer aided design of ALUs.

A *multiplexer* is a combinational logic circuit which has N input lines anyone of which can be selected by m select pins, where $2^m = N$ and steered on to the output line. This many to one switching unit is useful in input device selection on a computer system besides also meeting the powerful on line process instrumentation. Apart from the above basic feature, the multiplexers can be also utilized in implementing switching functions. In general, for implementing a n variable function, a multiplexer with 1 out of $2^{(n-1)}$ inputs is required with or without extra logic gates. The variables for the select pins can be chosen in such a way that additional overhead can be avoided, mainly the inverter gates. If in the minimal sum of products form of a switching function at least one variable is missing or if at least one variable appears only in one of the literal forms, the need for an inverter can be obviated. For example,

$$f(A, B, C, D) = (7, 9, 11, 12, 13, 14, 15)$$

The MSP form is

$$f = bcd + ab + ad$$

since all variables appear only in the true form, any of the 3 variables out of 4 can be used as address inputs.

$$f(A, B, C, D) = (8, 10, 11, 12)$$
$$f(A, B, C, D) = a\,c^{1} + a\,b^{1}\,c$$

The variable A occurs only in the true form. Thus the variables, B,C,D are connected as address inputs of an eight input MUX.

A demultiplexer is a circuit which steers the input line to anyone of the output pins based on selected address. The MUX - de MUX combination can go a longway in distributed processing involving duplicating images (Virtual work environment) for reliable and effective machine communications.

## 1.7 Arithmetic Processing Units

### Begining era of computers

The analog computer was first developed in 1876 by English scientist William Thomson ( popularly known as Lord Kelvin) for solving differential equations. The first working model could only be developed in 1931 by Vannevar Bush at M.I.T.

The idea of *punched cards* wasd introduced by a Frenchman Joseph Mary, Jacquard in 1780 in the development of an automatic loom for weaving intricate designs in fabric. Later on he modified the machine in 1801 for controlling 12000 needles in programmed fashion which remains almost unchanged even today. In 1896 Herman Hollerith (an American Statistician) developed a manufacturing company for punched Card equipments which later became part of International Business Machines ( IBM), the leading computer manufacturer of the present day.

In 1854 Boole found a new way of thinking and used symbols instead of words to reach logical conclusions. Though originally intended for solving logic problems, Boolean algebra now finds its greatest use in the design of digital computers.

The first all electronic computer the Electronic Numerical Integrator and Calculator *ENIAC* was planned and built by J.P. Eckert and J.W.Mauchly at the Moore school of Engineering, Pennsylvania, in 1946. It contained 18000 thermionic valves (vacuum tubes) and 1500 relays and worked in decimal arithmetic. ENIAC was capable of carrying out a decimal addition in 200 μS and a multiplication in 2mS.

In 1946 the lectures "Theory and Techniques of Electronic computers" given by J.Presper Eckert and Jonh W.Mauchly, owed much of their inspiration to Dr.J.Von Neumann, who had prepared a report on the logic design and outline proposals for a new

computer. These proposals called for common storage of program and data in binary fashion. *EDVAC (Electronic discrete variable automatic computer)* had its program stored in the computer's memory, not depending on external sequencing. This was an important innovation, and a computer which stores its list of operators, or program, internally is called a stored-program computer. Storage was regarded a key issue.

In 1948, IBM built the first stored program computer using 13000 valves and 23000 relays (Selective sequence Electronic calculator). Storage hierarchy was provided with a high-speed flip-flop store, a relay store, and a backing store using 80 column paper tape. Another stored program computer EDSAC (*Electronic delay storage automatic computer*) using different concept was developed in 1949 at the University of Manchester in England.

The first commercial data processing computer was built by Sperry Rand in the year 1951 using semiconductor diodes and vacuum tubes. This was known as *UNIVAC (Universal automatic computer)*. By 1955, 15 such models were produced. (IBM 701-1953). There after IBM and many other manufacturers started building commercial computers and real computer age started. UNIVAC was used to process the data from the 1950 census. Mercury delay lines were used as the main store with backing store on magnetic tape. A primitive operating system permitted data transfer directly under program control. Provision was made for off-line data preparation and print-out.

Thereafter, the progress with respect to the physical equipment, the hardware, though spectacular, has been confined largely to the use of enhanced technology. Progress on the programming side, the software, also has been spectacular, involving the provision of a hierarchy of programming languages, the development of software aids to programming, so-called debug facilities, and finally, software-based operating systems, which permit the overall control of machines, large and small and their associated peripheral equipment.

The field of digital computers can be subdivided into (i) general purpose computer and (ii) special purpose computer. A general purpose computer like IBM Sytems 370 can perform almost unlimited variety of tasks in engineering, scientific and business areas with the help of programming. The special purpose computers on the other hand is designed for particular applications and has prewired programs in general. The examples being navigational computers in space vehicles, electronics switching system for telephone exchanges, a computer used for N/C of machine tools and process instrumentation control systems.

Digital computer characteristics are listed as follows:

(i) Versatality, (ii) Accuracy, (iii) Speed, (iv) Large Memory, (v) Ease of communication (vi) Automation (vii) Supporting Problem oriented languages and (viii) Simulation.

The ALU of most computers are capable of performing the operations of addition, subtraction, division, multiplication and logical operations. ALU's vary considerably in

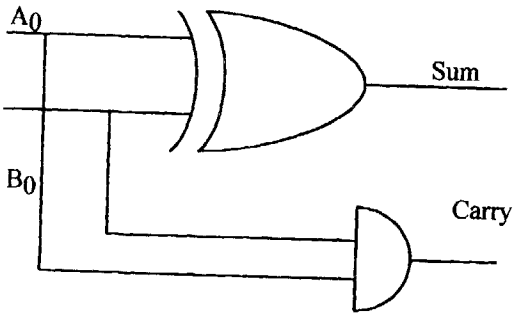
the number of different operations implemented. Most ALUs have a repertoire in the range of 16 to 256 commands. The main register of the arithmetic unit is known as accumulator and contains one of the numbers (operands) while performing arithmetic operations of two numbers.

The basic logic gates are often employed to achieve the arithmetic operations and the software of the machine controls the arithmetic logic units with the essential ingredients of Sequential logic circuits based on the flipflop terminology. The half adder is used to perform single bit addition. It produces the sum output and a carry as shown in Fig. 1.12. Fig. 1.13 shows the diagram of a full adder with the truth table. Fig. 1.14 gives the circuit of full addition using two half adders. Full adder essentially adds 1 bit position in practical situations, involving operands of 2 or more number of bits in length. The parallel 4 bit binary adder circuit is depicted in Fig. 1.15. The 8421 code is also known as the universal BCD Code (decimal number system). The BCD arithmetic can be performed by the adder unit shown in Fig.1.16 contributing to single decimal digit position.

Fig. 1.17 gives the truth table for half subtractor and the circuit. The full subtraction has to take into account 3 bits at any point of time, which can be realized as in Fig. 1.18. The vast applications of *ex-OR logic* include in giving a "1" output when the binary input string is of odd parity (has odd number of 1"s) and also can be employed as controlled inverters (NOT logic) on a 2-in-1 adder/subtractor circuit for sequential machines as shown in Fig. 1.19.

The circuitry required to perform arithmetic using decimal operands (BCD) is more complex than that needed for binary arithmetic.

A uniform method of propagating carries between adjacent positions is not possible since the weights Wi and Wi+1 of adjacent bits do not differ by constant factor. With n bits, We can have $2^n$ combinations with straight binary, whereas BCD restricts this to $10^{n4} \cong 2^{0840n}$. Thus numeric data requires more space in direct decimal arithmetic. Hardware conversions for B C D to straight binary is employed in floating point arithmetic. Note that microprocessor 8085 provides the DAA instruction for hexadecimal to decimal conversion. Fixed length op codes are preferred for easy decoding facility. The IBM batch machines have separate hardware units for decimal and straight binary arithmetic. The usual parallel adder is also called as ripple carry adder and the carries generated have to ripple through settling down before full propagation thus implying a delay in the addition process. To overcome this difficulty, faster arithmetic units employ the carry look ahead (CLA) process to speedup, discovered by Weinberger and Smith in 1956. The design was further refined in the stretch computer of 1960's. Fig. 1.20 just explains the $j^{th}$ stage full adder unit for the *carry-look ahead* principle. The minimum-delay adder is impractical because of size of the logic, and fan-in fan-out considerations. The CLA technique is used in groups of 4 bits and extended to sections of each 16 bits so that computational speedup can be achieved.

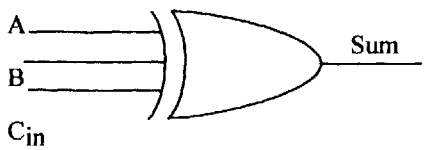



| $A_0$ | $B_0$ | Sum | Carry |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Sum $= \bar{a}_0.b_0 + \bar{a}_0.b_0 = A_0 \oplus B_0$

Carry $= a_0.\ b_0$

**Fig. 1.12** Half adder with truth table




| A | B | $C_{in}$ | Sum | $C_{out}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Sum = A ⊕ B ⊕ $C_{in}$

$C_{out}$ = (ab + b$C_{in}$ + $C_{in}$a)

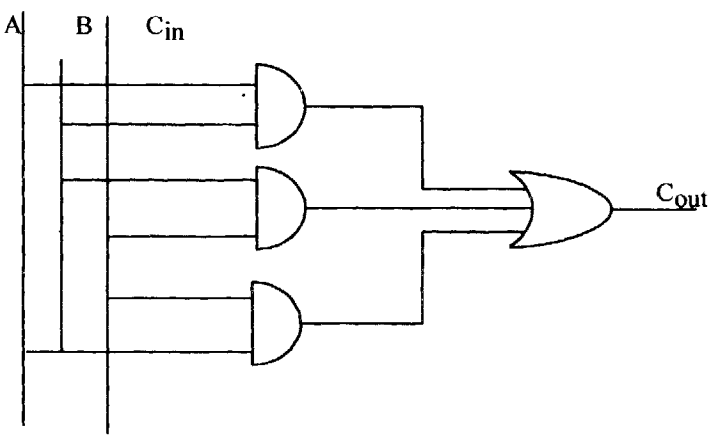



**Fig. 1.13** Full adder with truth table

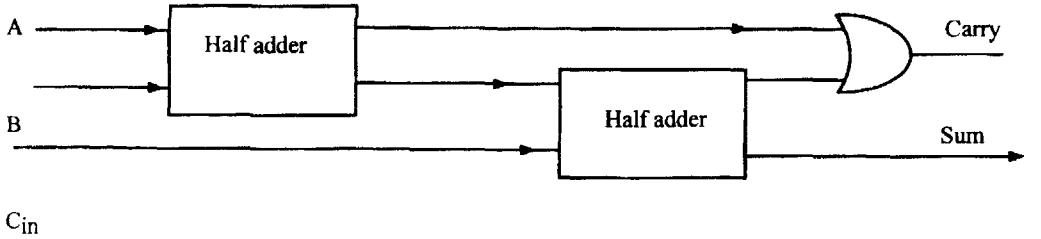



**Fig 1.14** Full adder using two half adders

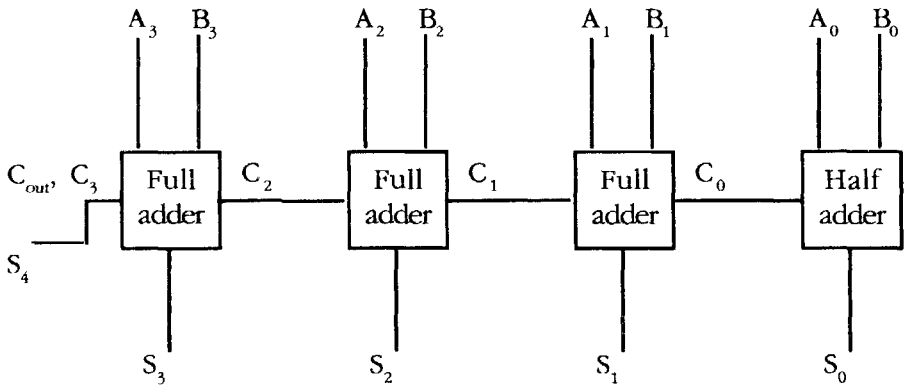



**Fig 1.15** Parallel 4 bit adder

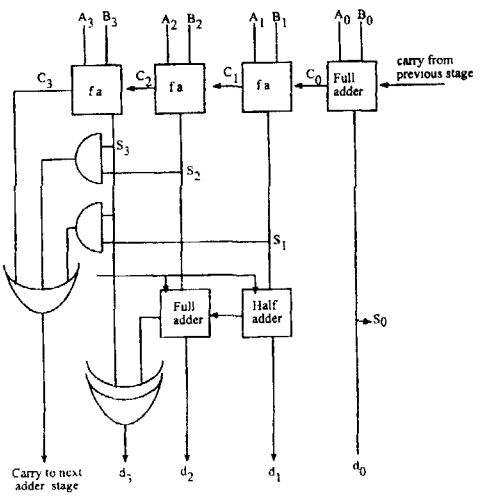



**Fig 1.16** BCD adder circuit

| $A_0$ | $B_0$ | $A_0 - B_0$ Difference | Borrow |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

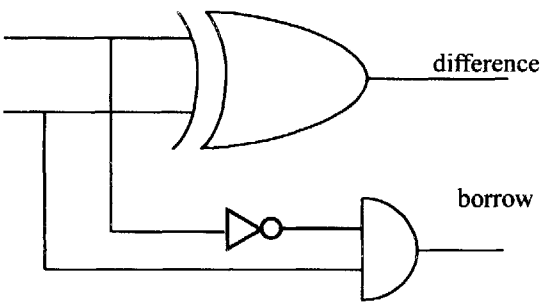



**Fig. 1.17** Half subtractor circuit and truth table

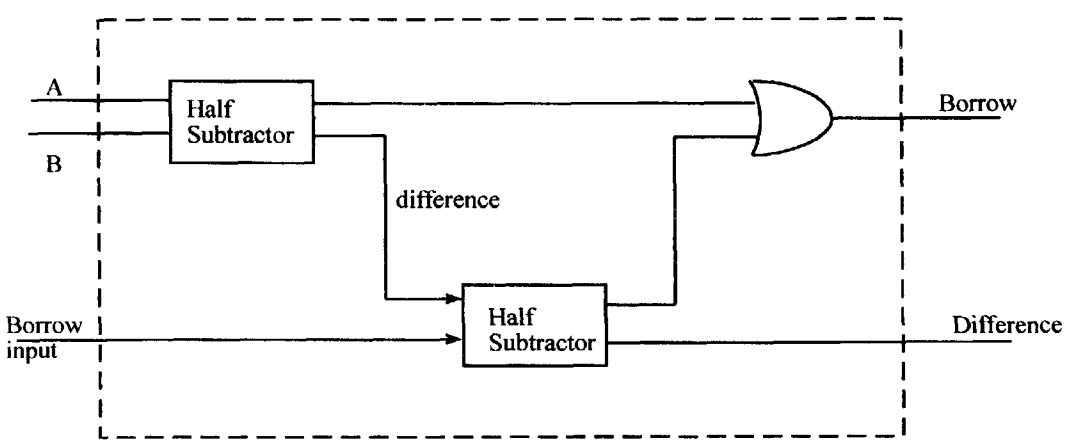



**Fig. 1.18** Full subtractor circuit

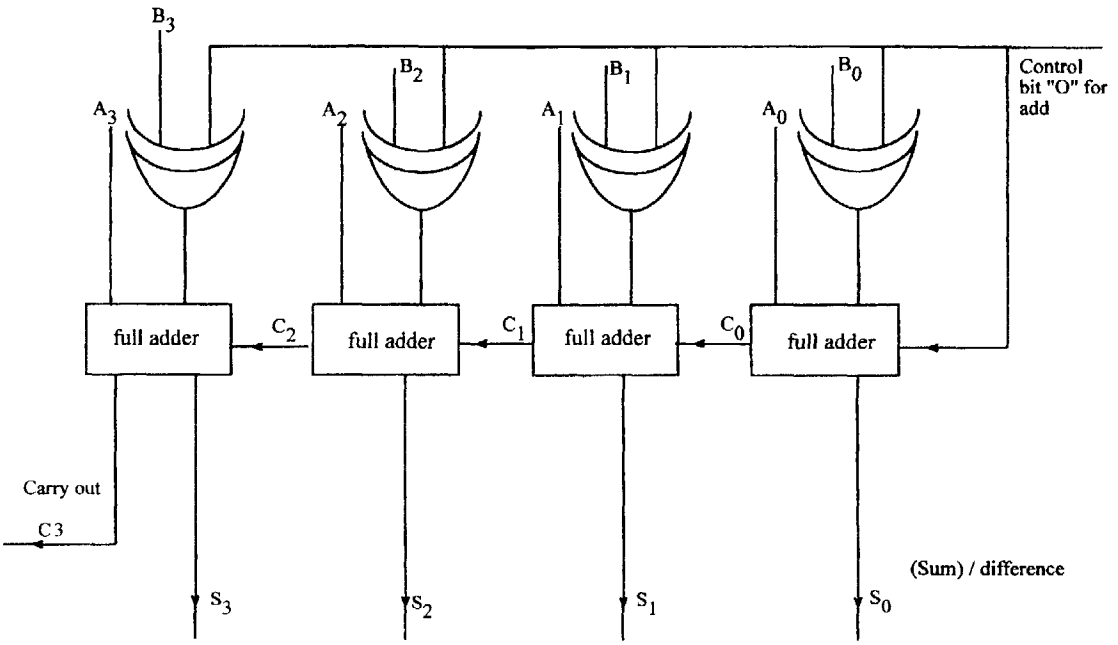



**Fig. 1.19** Straight binary adder/subtractor A+B / A–B

| $A_j$ | $B_j$ | $C_j$ | |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 1 | Cj+1 | (from prior bit) Propagate stage |
| 1 | 0 | Cj+1 | |
| 1 | 1 | 1 | = Generate stage |

**Fig. 1.20** Carry look ahead principle

For 64 bit addition, a 9:1 improvement in speed is reported with just 50% increase in cost. In C D C 6600 (60 bits wordlength), ALU contains groups of 3 bits combined in fours to make 5 sections. In IBM 370, 2 sections of four groups of 4 bits each contribute to the carry look ahead addition. (74181 is a 4 bit ALU and 74182 fits with it for CLA assistance in construction).

The other methods are partial carry, partial sum save techniques so that the units perform the arithmetic continuously with additional shift control and register store requirements on well designed synchronous sequential machines. Also this approach is a welcome factor on RISC architectures for memory bandwidth bottlenecks. By this method the *data rate* on the machine can be maintained in improving the system performance measurements.

The integer multiplication is a common activity and the 'a 1a russe' has given the dibble - dabble method as given in example shown in Fig 1.21. The doubling is just a leftshift operation whereas having gives rise to integer division discarding the remainders. The terms on the multiplicand columns corresponding to odd entries of multiplier column are added up to get the result.

| **Multiplier** | **Multiplicand** | |
|---|---|---|
| 45 | 19 | 19 |
| 22 | ~~38~~ | |
| 11 | 76 | 76 |
| 5 | 152 | 152 |
| 2 | ~~304~~ | |
| 1 | 608 | 608 |
| | | **855** |

**Fig. 1.21** Dibble - dabble method

The multiplication is usually performed by shift and add algorithm following the normal decimal arithmetic practice, inspecting everytime the multiplier bit status either 1 or 0. The product result can also be achieved by a repeated addition process choosing the smaller operand value as the counter value for number of additions to be performed. Still more splitting one of the operands may help a lot in software algorithms for multiply activity. Perhaps the MUL instruction of CPU 8086 is software driven. The fast super computers must essentially incorporate direct hardware multipliers to achieve the goals. Fig. 1.22 gives the block Schematic for the division process.

The divisor is stored in the X register and the dividend in ACC1 and ACC2 registers. The comparator compares ACC1 with (X). If (ACC 1) $\geq$ (X), '1' is stored in the least significant bit (l.s.b) of ACC 2. The content of X register is subtracted from (ACC 1) and the difference is stored back in ACC 1. Thereafter ACC 1 and ACC 2 are left shifted one bit with a shift pulse. The most significant bit (m.s.b) of ACC 2 occupies the l.s.b. position of ACC 1. In case (ACC 1) < (X), '0' is stored in the l.s.b. of ACC 2 and ACC 1, ACC 2 are left shifted by one bit. This process of comparism, subtraction and shift are

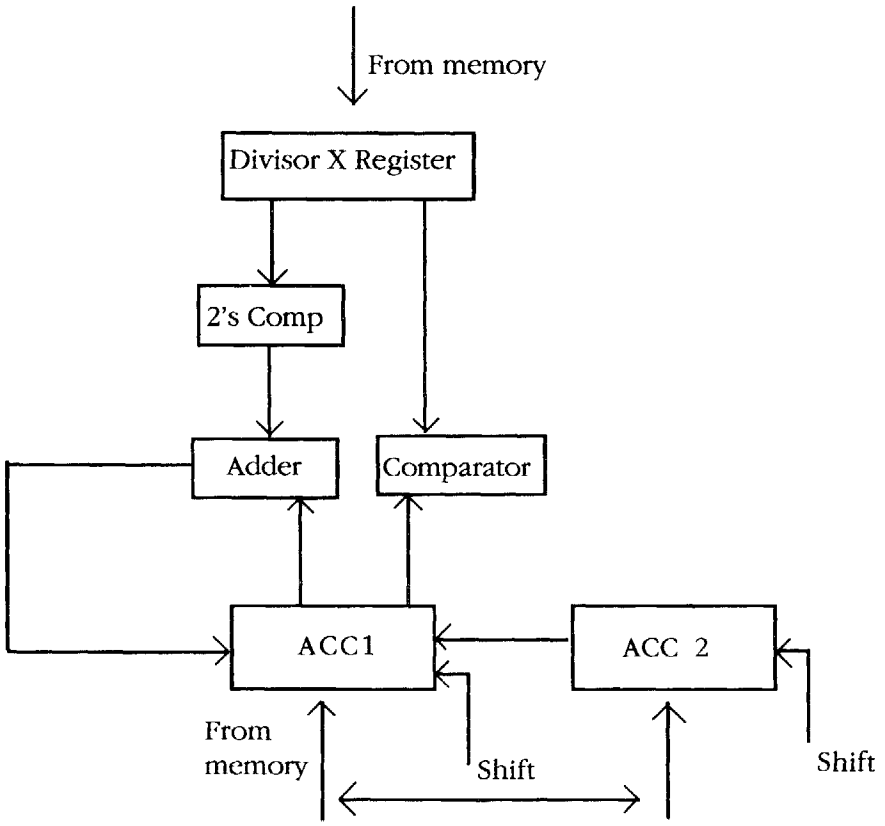



**Fig. 1.22** Divider Unit

continued till all bits of the dividend are compared. The content of ACC 2 is the quotient. The method of integer division is called as non-restoration type. Restoration type also gives the remainder value at the end of division process supporting floating point division. Normally in signed operations of multiplication and division, the sign bits are treated separately for ease of generating the final resultant values. Usually the computers support both the fixed point and floating point units to cater to the pipelined needs. Also these operations ( * and / ) can be realized by software instructions calling inturn an embedded subroutine on lesser compute bound machines.

The software libraries are supported by many a vendor for arithmetic functions involving algebraic, trigonometric and graphic solutions. For examples, math.h; graphics.c is a library towards this end in the "C" programming domains. While designing these

software utilities, the *portability* is considered a major issue besides the directly pottable CPUs. Arithmetic portability in ADA has received detailed attention, while floating point portability remains difficult and controversial despite the use of IEEE floating point standard.

In the bit slice processing area, serial addition units and implementation of *, / by not directly involving hardware units are a little towards improving the CPU storage though at the cost of less compute abilities in time domain is a way to bridge the bus bandwidth bottlenecks to maintain a smooth data flow at runtime. Introduction of energetic caches along with multiple functional units of this type is an indication of RISC units being deployed for parallel processing needs. Some of the computers support a lot of bit manipulation instructions for logical processing to meet the dual needs of parallel algorithms and expert systems in artificial intelligence. The operator precedence for evaluating boolean expressions is parentheses, NOT, AND and OR.

Encoder has $\leq 2^n$ input lines and produces n output signal bits. Octal to binary encoder has 8 inputs and produces a unique 3 bit octal code. Similarly a decoder is a combinational circuit that converts the binary data from n input lines to a maximum of $2^n$ unique output lines.

The arithmetic unit discussed so far are much dependent on the sequential circuits for their activities in real-time. Hence the sequential logic elements behave in a controllable fashion for the complex control unit governing the CPU runtime results at CPU clock speeds.

## 1.8 Sequential Machine

A *flipflop* is another name for bistable multivibrator and used as a bit storing element. The basic flipflops like set clear, JK are worth mentioning.

The characteristic equations of a SC flipflop are

$S\ C = 0$

$Q = (S + \overline{C}\ q)$ where Q is next state and q is present state

and S, C are excitation inputs.

The J, K allows also the inputs J=K=1 for a toggling state. The characteristic function is $Q = (J\ \overline{q} + \overline{k}\ q)$. The D flipflop is used as a data latch and T flipflop for frequency dividers. A set of n "T" flipflops delivers output of $f_{out} = f_{in} / (2^n)$, where $f_{in}$ is the frequency of the input clock pulses. In general, all flipflops are used under the supervision of a clockpulse (trigger) and find enormous applications in store registers, flag decisions and control propagations. Fig.1.23 gives the block schematic of a sequential machine.

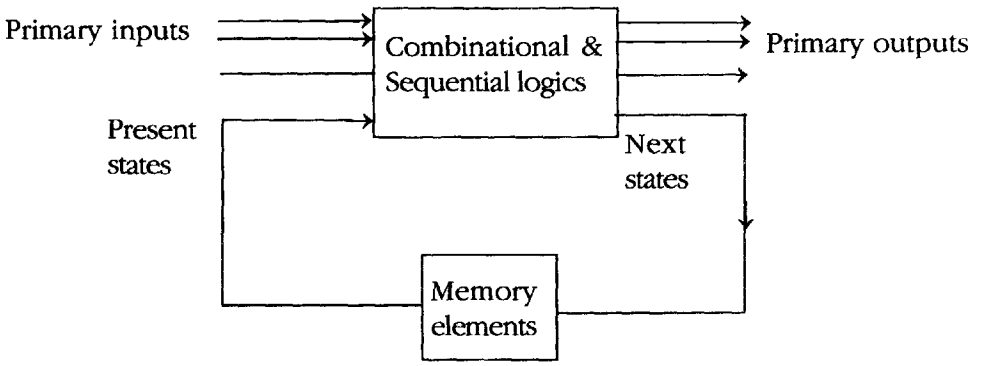



**Fig. 1.23** Sequential Machine

A sequential circuit that is based on an equal state time or a state time defined by external means is called a synchronous sequential circuit. A circuit whose state time depends solely upon the "Internal logic circuit delays" is called an asynchronous sequential circuit.

The SR latch is an asynchronous sequential circuit and the operational behaviour of the flipflop is depicted in Fig. 1.24 by its state transition diagram. Fig. 1.25 shows the excitation tables for different types of flipflops. A sequential circuit is an interconnection of flipflops and gates. This necessarily includes primary output points and inputs. Given a finite state machine with clearly defined states, it is possible to design and construct the circuit using several types of flipflops to meet the implementation constraints. This process is called circuit synthesis. A state diagram is an alternative approach for stating sequential circuits.

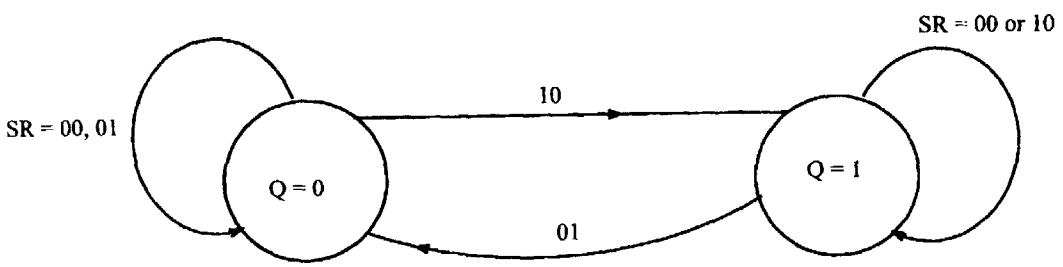



**Fig. 1.24** State transition diagram of SR flipflop

In addition, the combinational circuit may also contain external outputs. In such a situation, the boolean functions for the external outputs are derived from the state table by combinational circuit design methods.

Counters are easily achieved by making use of JK flipflops. All flipflops in a synchronous counter are clocked at the same time whereas the ripple counter is an asynchronous one. A chain of n flipflops will count upto $(2^n - 1)$ and then reset to 0 0 . . . 0 at the $2^n$th pulse and is referred to as modulus $2^n$ counter. The design of sequential logic reliability depends much on the clock pulse rate and width with reference to the circuits they trigger to ensure racefree operation. To meet the same objective, the master slave flipflops offer isolation while sampling and also buffering for a good data flow in computing. The other capacity of registers, that is shifting, is much utilized for arithmetic operations and logical processing at run time.

A sequential machine is said to be of the MOORE type if its output function is dependent only on the states prevailing.

**J K Flipflop**

| | | Input JK | | | |
|---|---|---|---|---|---|
| | | 00 | 01 | 10 | 11 |
| State | 0 | 0 | 0 | 1 | 1 |
| Q | 1 | 1 | 0 | 1 | 0 |

$Q(t+1) = J(t)\,\overline{Q}(t) + \overline{K}(t)\,Q(t)$

**SR Flipflop**

| | | Input JK | | |
|---|---|---|---|---|
| | | 00 | 01 | 10 |
| State | 0 | 0 | 0 | 1 |
| Q | 1 | 1 | 0 | 1 |

i.e. SR = 1 1 is *forbidden.*

$Q(t+1) = \overline{R}(t)\,[S(t) + Q(t)]$

**D Flipflop**

| | | Input D | |
|---|---|---|---|
| | | 0 | 1 |
| State | 0 | 0 | 1 |
| Q | 1 | 0 | 1 |

$Q(t+1) = D(t)$

**T Flipflop**

| T | Q(t) | Q(t+1) |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 1 |

**Fig. 1.25** Excitation tables

Every Mealy machine can be converted into a Moore type and vice-versa. According to Turing, two machine systems can be said to be *equivalent* if any type of problem can be solved on either of the machines, barring only the storage and time complexities involved. In automata theory, any CPU is defined to be a finite state machine which is amicable for computer aided design and simulation areas. In-circuit-emulation requires extra attachments (both hardware and software) for simulations like the microcomputer development systems for enhancing the powerful features of an existing one.

State reduction algorithms are concerned with procedures for reducing the number of states in a state table while keeping the external input - output requirements unchanged. An unpredictable effect in reducing the number of flipflops is that they may require more combinational logic circuits. State assignment procedures are concerned with methods for assigning binary values to states in order to reduce the cost of the combinational circuit.

A synchronous sequential circuit is one in which the memory contents change only at discrete instants after the occurrence of a synchronizing pulse. In asynchronous circuits the change in memory state variables and inputs are restricted to one bit at a time to avoid races and buzzer cycles. A state table with binary assignment is called a transition table. Sequential logic synthesis requires problem formulation, state reduction and assignment and circuit selection for final implementation.

**State equivalence**: Two states, $q_i$ and $q_j$ of a sequential machine are said to be distinguishable if and only if there exists at least one finite input sequence which when applied to the machine results in different output sequences.

In the area of sequential logic minimization, the techniques of sequential state assignment and minimization are open problems even today. This answers the instruction set size of a CPU is always 60% of its full capacity for further improvements by innovative research areas like microprogramming, additional vector processors (coprocessors for compute bound solutions) and intelligent compilers besides effective programming trends. The power of the powerful CPU architectures is taken up in the subsequent Chapter 2.

## Keywords

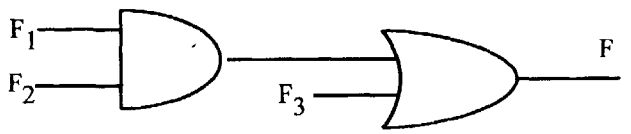
Thruput, user, turn-around-time (TaT), architecture; radix, bit, ternary, byte, octal code; Von Dieuren code, ASCII, eb-si-dik, Huffman code; mantissa, exponent, FORtran, algorithms, 2's complement arithmetic, literal, NOT function, Huntington, duality, De Morgan's theorem, product term, disjunction, minterm, Venn diagram, canonical form, standard sum form, V-K map, DCF, prime implicant, Quine McClusky, weight, degree of adjacency, multiplexer, punched cards, ENIAC, EDVAC, UNIVAC, Ex-OR logic, carry-look ahead, data rate, portability; flipflop, equivalent.

## PROBLEMS

1. Convert to the base required.
   $(71)_8 = (?)_{16}$
   $(35.5)10 = (?)_{BCD}$
   $(35.5)10 = (?)_2$
   $(730.4)_8 = (?)_{16}$
2. Mention the merits of numeric data representation in straight binary form.
3. Write a program in 8085 assembly language to add two 16 - bit unsigned BCD numbers. Assume the operands are in BC and DE register pairs. The result must be stored in BC . ( higher order register in the register pair contains higher order digits of operands).
4. Write detailed notes on ASCII and excess - 3 codes.
5. Write a note on gray codes. Comment on the minimum distance for a code to be "t" error correcting.
6. An 8 bit binary code contains a parity check bit. How many code words are now available that are not used?
7. "Operands first, operations next"
   How does the phrase fit to COBOL language? Explain the COMPUTE verb of COBOL.
8. If $f_1(A,B,C,D) = \Sigma(0,1,2,3,8,9,10,11,12)$
   $f_2(A,B,C,D) = \Sigma(0,1,2,9,10) + \phi(11,15)$
   find $f_1 \cap f_2$ and $f_1 \cup f_2$
9. State and prove De Morgan's laws.
10. What is the significance of two level implementation of logical functions in digital systems?
11. Write down the equations of any two 2-input logic gates with their truth tables and symbols.
12. Three switching functions are expressed as below:

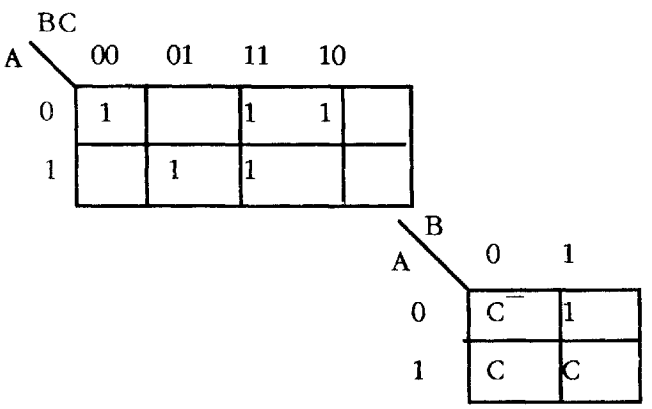
    $f_1(A,B,C,D) = \Sigma(0,1,2,3,5,12)$
    $f_2(A,B,C,D) = \Sigma(0,1,2,10,13,14,15)$
    $f_3(A,B,C,D) = \Sigma(2,4,5,8)$

Express the function "f" realized by the circuit shown below as the sum of minterms ( in decimal notation).




13. The karnaugh map of a function f(A,B,C) is as shown and the reduced form of the map is shown on the right hand side, in which the variable "c" is entered in the map itself. Discuss the methodology by which the reduced map has been derived.
14. How is a demultiplexer different from a decoder?
15. Define the terms - weight of a binary string and essential prime implicant.
16. Minimize the function
    $f(X_1, X_2, X_3, X_4) = \Sigma(0,1,5,6,7,11)$ Using Karnaugh map method. What is a don't care term?
17. Write a note on Canonical forms for combinational function representation.
18. Write a clear note on the adjacency method expounded by Biswas.
19. Minimize the following by karnaugh map procedure.

    (a) $f(x, y, z) = \bar{z}x + z\bar{y}\bar{x} + zy\bar{x}$
    (b) $f(A,B,C,D) = (b^1 c^1 d^1 + b^1 c^1 d + ab^1 c + a^1 b c^1)$.
20. Using quine Mc Clusky method, simplify the following in SOP form
    $f(A,B,C,D) = \pi M_j (1,4,6,7,10,12,14) + d(2,13)$ where d represents don't care conditions.



21. Implement using a 8 – to –1 Multiplexer, the function
    $f(A,B,C,D) = \Sigma(0,1,2,3,5,6,8,10,15)$. Use A,B,C as select inputs to the MUX.
22. Using logic gates, implement a half-subtractor.
23. Implement a logic circuit for $A \geq B$ where A and B are 2 bit data.
24. What is the need for decimal conversions in switching systems?
25. What is ripple carry addition? Explain the advantages of serial arithmetic on a single user system.
26. How is the problem of memory bandwidth alleviated by the use of counter registers on innovative machine architectures ?
27. Draw the schematic of adder/subtractor unit and explain how software selects the desired arithmetic operation.
28. What are carry propogate and generate stages? Explain anyone method for faster addition.
29. Attach the usefulness of bit slice processors for compute time reliability with single user serial organizations.
30. Write a straight line program that computes the determinant of a 3 x 3 matrix, given the nine scalar elements as input.
31. Correlate the terms accuracy and range of numbers in the exponent mantissa format.
32. Write note on hardware multipliers with master slave flipflops.
33. Mention few applications of integer arithmetic.
34. Show that there can be no mantissa overflow after a multiplication operation.
35. Double-precision floating point numeric data uses 53 bits for mantissa that includes a sign bit and 11 bits are used for biased exponent. Calculate the maximum number of decimal digits that can be stored by this representation and the maximum and minimum values of the exponent taking proper constant as bias.
36. Using a 2 input multiplexer, implement only the sum output of a full adder.
37. Construct a four input multiplexer making use of two-input multiplexers. Mention the advantages of the same in processor control design.
38. Generate the distance and weight distribution matrices for the standard Hamming (7,4) code.
39. What is state equivalence and attach the above phrase for fail safe sequential circuits.
40. Using only JK flipflops, construct mod– 4 down counter.

41. Explain the significance of uninterrupted power supplies in supercomputers.
42. Bring out the specific advantages of counters as controlling elements in computational activity.
43. Design a mod – 5 counter assuming the state transitions using anyone type of flipflops.
44. Write on the current status of state assignment and reduction techniques in sequential machines.

# CHAPTER 2

# CENTRAL PROCESSING UNIT - POWER OF ARCHITECTURES

## 2.1 Levels of Abstraction

A central processor unit essentially is made up of the arithmetic logic unit which can do processing to varying degrees based on the architectural constraints. This powerful unit is termed the system controller in the dual sense it controls itself to arrive at a solution to the program input and is being controlled by the data available in primary memory, often called the mother-board. The central processor is also called the *microprocessor* with its birth in 1971. The computer system designed with nil programmability are called programmed machines. The examples are hand-held calculators, business desk-top calculating machines and identified process instrumentation systems mapping to the Applied specific Integrated circuits design group.

But since the manipulation and interpretation on data varies from each user, the computers really have to become programmable catering to flexibility in applications. Though every processing element is a number cruncher, each has its own machine language. The major task of Language support is a primary parameter as any architecture is concerned. Fig. 2-1(a) presents the different levels of language organisation. From the figure, the assembly Level is supposed to be the special area of system engineers who are to be thorough with the language of the particular central processing unit in terms of its geography and the powerful system routines (like MACROS, pseudo - programs and micro - routines) besides the basic instruction - set the processor supports at the assembly level for really tailorable events. With this exhaustive knowledge of the P.E. (processing element) configuration, the Assembler has been written for many a system, which is in essence, a true translator, which implies only a lot of calculations and computations are done before arriving at the Objective, i.e., the .obj files. Whereas any mistakes (called as bugs) arising after translation is fully attributed to the logical and psychological mind on the . ASM source file.

> ***Defn.*** *" An assembler translates the stated assembly code into machine code in a readable form".*

But with the evolving architectures touching the phase of computer networks and portability, the need has come up for Cross - assemblers in distributed and parallel processing environments.

> *"A Cross - Assembler is one which translates the assembly program of One machine to the object code of a second machine where the machines can be said to be equivalent if not identical".*

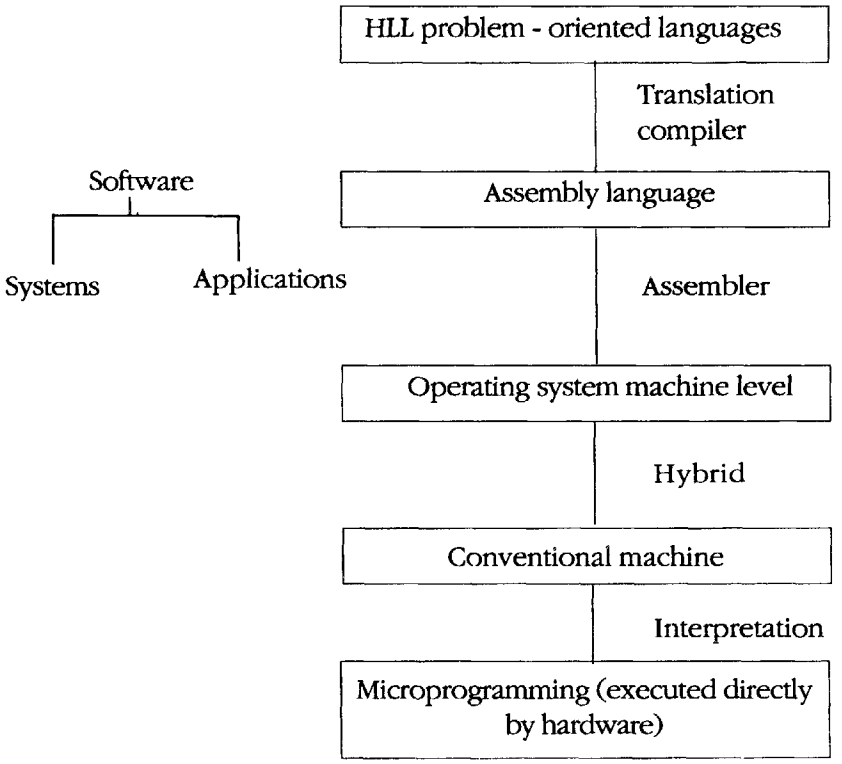



**Fig. 2.1 (a)** Levels on computer architecture

Cross-assembling across processors in the parallel processing domain needs a heavy weightage on *portability* issues to cater the portable tasks. The complexity of computers grew much with the birth and growth of operating systems. A system can be said to be operating if the operating system is alive. Thus, the architectures have to maintain a lot of its own manageable data before being well capable of managing the User data. Thus rightly in the domain of software engineering, program maintenance is left to the USER (programmer, in general), and problem accounting is left to the processing element.

The High level language is more user friendly and English like, and each language has been designed for specifc application areas like voluminous data processing in

commercial circles, scientific computing needs and finally boiling to packages for interactive and command-based languages. The languages at this level are strictly syntax-driven serving more a general semantic base. Table 2-1 (a) depicts merits of HLL VS assembly level. Table 2-1 (b) lists some assembly level models. Now computer systems have captured the voluminous area of desk-top computing in daily applications to microprocessor-based process control domain covering both programmed needs and intelligent bases. The machine level, is of course, the straight mother-tongue as machines are concerned for number crunching which is more of one-to-one nature.

## 2·2 Powerful Parameters

**Table 2.1 (a)** HLL vs ALL

| *High level languages* | *Assembly level Languages* |
|---|---|
| Machine independent and user-friendly | One to one and for systems engineers |
| Less programmer efforts | Very fast execution meeting both time and storage complexities |
| Provides standardised application packages as utilities and 100% commercialised | Best fits the ASICs area with lesser word length machines examples being microcontrollers |

**Table 2.1 (b)** Assembly language machine models

| *Computer model* | *Assembly language* |
|---|---|
| TDC – 316 | BAL – 316 basic assembly |
| TDC – 312 | ECOBAL |
| ICL – 1900 | PLAN programming language |
| Honeywell | EASY easy efficient assembly |
| IBM – 1401 | AUTO - Coder |
| IBM – 360 | Assembler |

With the conceived idea of stored program concept and sequential processing owing to John Von Neumann and with the birth of VLSI in 1970's with the inclusion of microprocessors, it is felt essential to study the power of computers so that the desirable features can be embedded into a machine in order to meet both the interactive command base and demand oriented applications. The power of a CPU is governed by the following list of parameters which will be discussed one-by-one.

(i) The clock of the system;
(ii) The wordlength of the processor;
(iii) The external Buses;
(iv) The inbuilt computing power;
(v) CPU Storage optimization;
(vi) Data transfer schemes - priority of;
(vii) The instruction set complexity; and
(viii) Cache memory and associative linking.

**Clock**

In real-time and in the fast world, our activities are governed by the universal clock which is based on the elementary unit of time, Second. Whereas the microprocessors began their generation from a clock frequency of 2 to 3 MHz and propagated to date, to a clock speed of 50 to 60 MHz in a span of two decades. Thus, the *clock*-speed is the first and foremost factor dictating the THRUPUT on a system which is composed of multi and parallel events.

**Word Length**

A microprocessor like Intel 8085 is a 8-bit CPU which can at one stroke do arithmetic and logical computing on 8 bits of data. In the realm of architectures, the range of wordlengths from 1 bit to 64 bits is a tremendous growth regarding the number crunching factor in a matter of 2 decades. See Fig. 2-1(b). Applicationwise, the process-control areas (dedicated) are suited for a wordlength of 8 bits, 16 bits category accommodating the PC

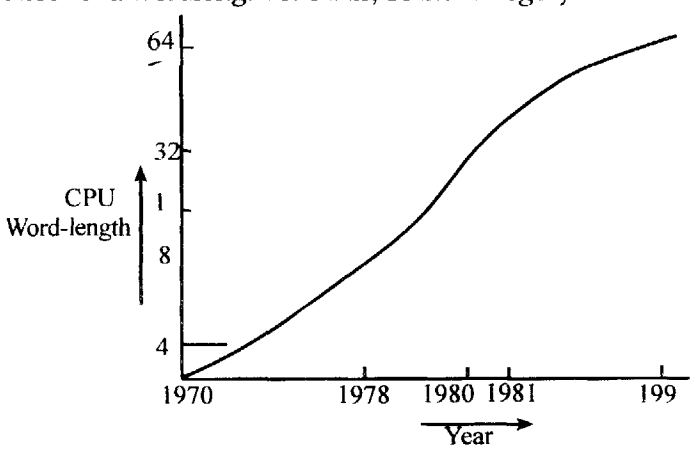



**Fig. 2.1(b)** Growth of word lengths.

community (pioneered by IBM corporation), 32 bit machines catering to timesharing and multi-user platforms to the powerful giants of > 64 bits for the super computing categories.

It is, no doubt, this factor of *wordlength*, governs both the power of a machine and the cost involved in installation of such a system. It has been accepted and more become a standard to designate a set of 8 bits as a Byte.

**External Buses**

Most of the initial systems employed multiplexing of address and data pins because of the highly sequential nature of instruction flow for a program-meet. But with the inclusion of parallel programs, viz., user-written, user called and self-resident embedded softwares, it has become essential to have separate address and data paths to meet constraints of Turn around Time and Thruput as the machine architect is concerned. For example, the IBM/360 supports decimal arithmetic as well floating -point arithmetic which obviously has to go in parallel on batch machines to cater to processor utilization. In this respect, *Relocation* is a facility which adds to the embedded power of the central processor.

This means being able to move a machine code program about in memory, the addresses still being valid. More precisely, it means being able to defer a decision on where-in the program is to be loaded until loading time (that is, not at assembly or compilation time). In order to achieve this, instead of addresses being absolute (that is, refering to a specific location in memory) they must be relative to something that will be adjusted at load time (when the program is loaded into memory just prior to execution). This is achieved based on the mode of specification of primary memory locations, viz., like direct or base addressing. Perhaps a Loader is used to accomplish the task of loading any program into main memory, usually from backing-store. Addressing memory locations is left to the instruction set of a computer which is more attached to the wordlength. Thus, computers with a relatively short wordlength have to employ a more complex addressing mechanism requiring multiple accesses to memory just to fetch the instruction.

For the CPU, the data operated (i.e., operands) are available either in registers or attached storage (Primary and cache memories) and the operators themselves, more often, reside within the CPU. Thus, the number of Buses and width of each bus play a crucial role in computer operations.

**The in-built computing power**

The bipolar as well as unipolar devices suit to build up a computing element covering both logic and arithmetic. The speed of operation of the operator is the primary concern governing the power of a central processing unit besides the desirable property of embedding more storage to enhance the throughput by reducing the number of seeks to primary memories for the current and active data. Some essential parameters for different logic families is given in Table 2-2.

**Table 2.2** Performance data for IC types

| *Logic type* | *Power per gate, mW* | *Propagation delay, nsec* | *Fanout* | *Noise immunity* |
|---|---|---|---|---|
| RTL | 10 | 25 | 5 | Fair |
| DTL | *10* | *30* | *5* | *Good* |
| TTL | 15 | 10 | 10 | Good |
| ECL | 30 | 3 | 20 | Very good |
| COS/MOS | 0.005 μW | 25 | 20 | Very good |

Because of the very major advantage of power of the unipolar devices, the MOS technology has become a promising candidate for VLSI design area in capturing higher power built within a single CPU. The CPU integration strength will directly reflect the instruction capabilites of a machine. Based on the processing abilities, the CPUs are said to follow the CISC (complex instruction set computer) or a RISC ( reduced instruction set computer) machine. Fig.2-2 (a) gives the block diagram of a reduced indtruction set machine. It has to be agreed that more CPU - in - storage is costlier than in general the memory capacity to compute into powerful compute - bound applications. Thus, one type of computation is only reflected by the main problem orientation.

The evolution of a CPU towards reduced instruction set computing (RISC) needs to be viewed in the context of application specific integrated circuits (ASICs). The CPU has to be designed having as few functional units as possible in terms of processor design. The ALU set is implemented using adders, shift registers and combinational logic because the hardware ALU operates at a very high speed of just a few nanoseconds compared to the data rate. For example, the VLSI chip Motorola 68,000 having about 68,000 transistors on chip using NMOS technology operating at a clock of 8 MHz performs at a minimum instruction time of 500 nanoseconds to a maximum instruction time of 21.25 microseconds.

In the RISC unit shown in the Figure 2.2 (a), many internal registers serve as general purpose, program pointers and save areas. Thus, a relatively high amount of feeding can occur enhancing the inherent parallelism feature embedded in a CPU.

By using a proper pipeline -design, many complex instructions can be achieved by a few instructions of the RISC machine, which involves a higher degree of co-operation among the machine architects and compiler writers in the use of compilers to optimise object code performance.

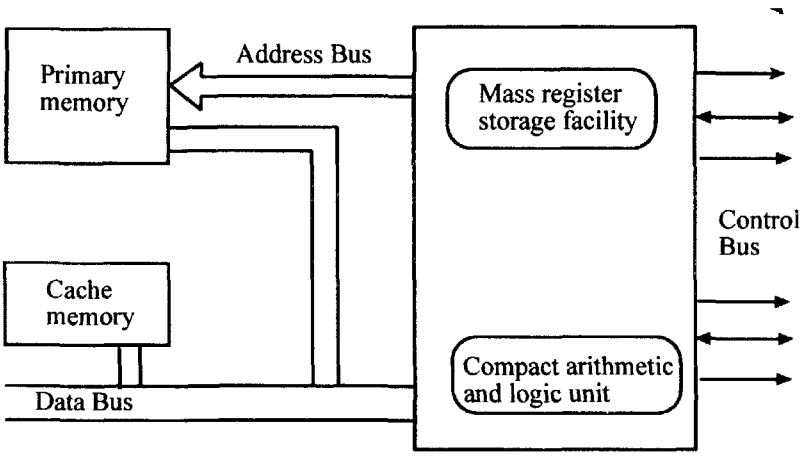



**Fig. 2.2(a)** RISC Architecture

The main attributes like few instruction types and addressing modes, fixed and easily decoded instruction formats, fast single-cycle execution and hardwired control will accelerate thruput, justifying the need for large CPU-in storage. In addition, cache memories will be an added feature in the area of RISCs in database management systems, converging either towards the applications of information technology or to the buidling up of expert systems in real-time applications towards an intelligent base. This feature will provide more of a parallel processing environment towards a shared database, which is the demand of today.

External to the CPU, we can have separate address pins and data pins for parallel flow of information in the current age of information technology incorporating a *memory mapped I/O* feature in the area of multi -user multi-tasking applications. The only limitation of RISC is its lack of utility in the area of scientific computing.

Examples include RISC arch. developed by Stanford University. DEC, SILICON GRAPHICS, SONY, Military Avionics Program in U.S.A are using. Specific computer RISC microprocessor uses less number of addressing (Memory) Modes, Program manipulation instructions, and stronger memory system diagnostics.

A noteworthy feature in the computer industry is the inclusion of engineering work-stations in the market of PCs, midrange systems and even mainframes.

An overwhelming number of work stations use RISC technology. Newer systems are built around 32 bit conventional architecture, e.g. being, Intel's 80386/ 80486 as well RISC architectures such as Sun Micro Systems, Sparc and MIPS computer System.

With the present day trend in multitasking activities, the in-built computing power can be optimised by intelligent timesharing designs on a single CPU platform and a systematic method of memory organisation and synchronization can be attempted by the use of semaphores and dataflow networking. The CPU power is also monitored by the ability of the system supervisor with respect to operating systems, for an effective throughput and improved Turn-around-time as the data transfer schemes configuration is concerned. Today, CPUs are available with a clock rate from 2 MHZ to 40 MHZ which does reflect the type of application environment.

**CPU Storage Optimization**

The need of increased storage within a CPU is attributed to two different situations, viz.,

To manage with a simple instruction set in arriving at solutions with higher speeds;

To match to the computing speeds that is available of the CPU elements, more so, in the domain of pipeline and parallel processing.

Though, no doubt, the CPU storage available is an asset, the return can only be felt by intelligent programming co-ordination activity. This statement applies to both multi user interactive environment and giant batch processing machines for high Scientific computing. But, in essence, the additional overheads, both in terms of Hardware and Software is not ruled out in order to optimize the available use of resources at any point to time. For example, the Intel 8086 maintains a 6-byte instruction queue apart from its rich register storage towards optimization. For compute-bound problems a numerical co-oprocessor 8087 and I/O processor may be added along with the prime 8086 CPU to speed up the activities. 8086 has a 20 bit address bus and the data-bus width is 16 bits. Modular programming which calls for memory segmentation in using high level languages is a desirable feature in chip designs. The peak computing power is often stated in so many mega flops (million floating point operations per second) or lips (Logical inferences per second). This lips is basically decided by maintaining a close-look on the flag activities of the various CPU flags of a stated problem.

**Data Transfer Schemes - Priority of:**

Every computer system employs the following data transfer methods:

PROGRAMMED DATA TRANSFER

Most of the .obj and .exe files are executed using this method. Essentially many instructions are just data movement across the working sections when the CPU is active, and about 80 to 90% of the computer time is allotted for this. The basic reason behind this is, more often, the processing elements are sequential by nature. This is a synchronous scheme with a predictable output. Algorithms assist in improving by manifolds the *thruput* factor of any available machine.

### Interrupt Driven I/O

Sometimes, it is required that interrupting programs are attended to and the main program is resumed. Also on a multiuser platform, the wanted data of a terminal is immediately communicated in the usage of operating system commands (which are permanently self-resident programs that have the highest priority of attention). Most of the process control and electronic instrumentation programs are interruptive in nature as interactive programs. At the Apex, the computer networks are either used for interprocess communication or just for data routing, which employ a higher degree of interrupt driven I/O mechanisms.

### DMA (Direct Memory Access)

This is also called cycle-stealing transfer, because, in essence the external buses are used by the channel requesting the DMA operation. More of the flush-in and flush-out activities need the DMA scheme not only to meet their own ends but also pleasing the CPU with a higher performance figure. Fig 2-2 (b) gives the block diagram for DMA scheme. The use of the interrupt driven and DMA methods will more reflect the character of a computer system with its prolonged use during its lifetime.

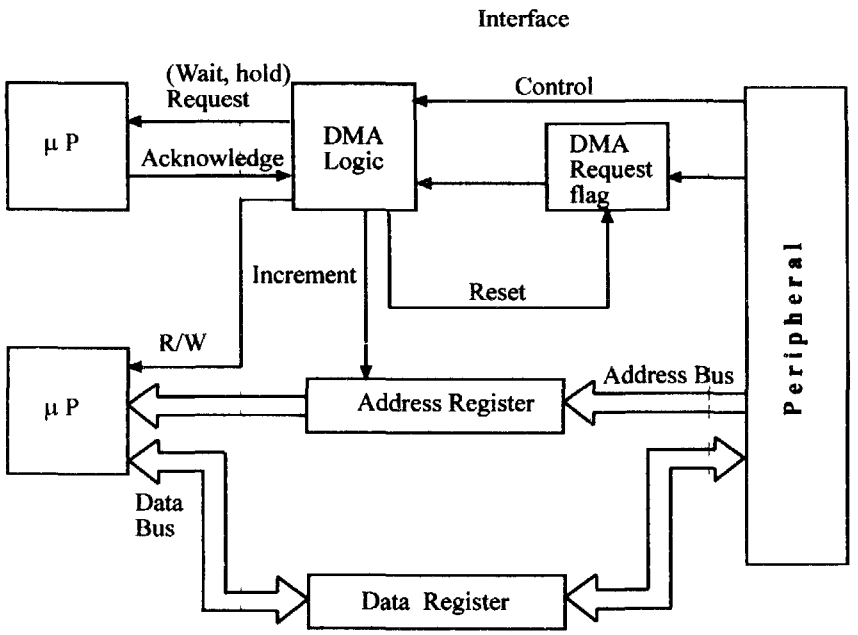



**Fig. 2.2 (b)** DMA block schematic

## Instruction set complexity

The power of a system depends mainly on the total number of instructions it supports and more qualitatively the various addressing schemes the processor may employ. The different addressing modes are listed and explained one by one.

### Direct Addressing

Fig. 2-3 (a) addresses this type of accessing mechanism. In essence, every instruction has an *Opcode* (Operation code) and the accessing of Operands is implied by both the instruction mnemonic and the operands themselves. Here, the operand address is a part

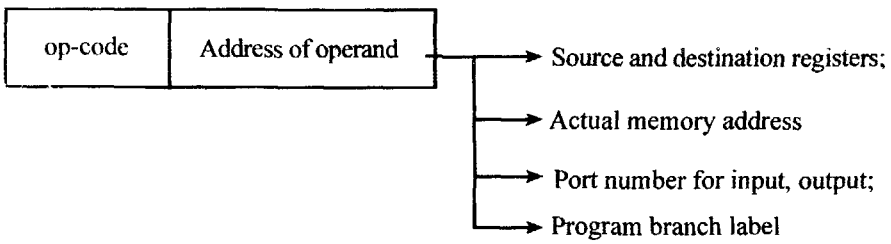



**Fig. 2.3 (a)** Direct addressing

of the instruction itself and is explicitly stated. Examples are moving data between CPU registers, loading data between memory locations and registers where again the address of memory is directly specified. Also, branch instructions to a direct address falls under this category. IN and OUT instructions are a special category of this group. The instruction complexity is often dictated by varying attributes like instruction types, length of instruction and the instruction time. To summarise the features of the direct addressing method, the following points are noteworthy:-

It is the fastest method (less instructions time); Address is a part of the instruction; Simple and easy for implementation.

### Immediate Addressing

Fig.2-3 (b) shows the general format of this scheme. Here the operands or data is a part of the instruction, that is, the data is stored consecutively in memory locations.

Initializing count registers with a definite integer for perfect program loops, comparing the real variable with a fixed set point value as in decision making applied to data base systems and critical quantified values in process instrumentation systems serve as illustrative examples of this scheme. The points worth mention of this method are, it is simple to understand and decode; is also fast to implement and more often fits to be used with the direct addressing mode combination.

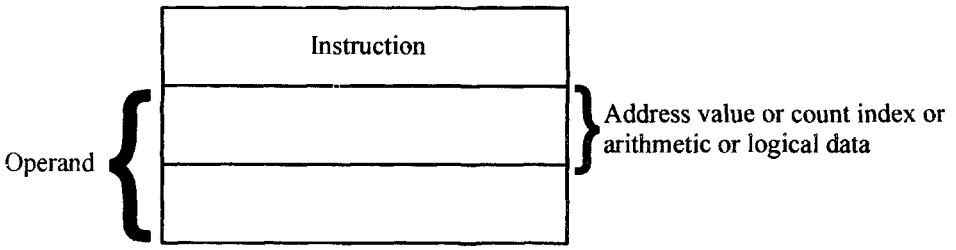



**Fig. 2.3 (b)** Immediate addressing

INDIRECT ADDRESSING

The address part of the instruction just links to another directory (like registers or another memory address) whose contents directly points to the data under use. See Fig.2-3 (c). Thus, the effective address calculation needs more fetches and slows down the process. The following points highlight the characteristic views of this mode, viz., it is a slower method of access; is highly flexible for data routing and is often utilized with addressing schemes like index and base register accessing.

| opcode | operand address routed |
|---|---|

Example: MOV M, B of 8085 CPU.

[(H) (L)] ← (B)

HL register pair serves as memory pointer

**Fig. 2.3 (c)** Indirect addressing

STACK ADDRESSING

Stack grows physically downwards. Fig. 2-3 (d) gives the stack property. In most of the conventional sequential machines, either system or user *stacks* are accessed by this method. Here the contents of a specific register, called, stack pointer, is used to point to the stack data. Automatic storing of the subroutine return addresses, saving of essential register data for reuse are some routine applications of the scheme. Apart from this, at the peak level the computers follow a stack architecture for a data flow in compute bound activities subjected to SIMD domain too. This type is also referred to as 0 - address instructions.

The features are – Specific instruction mnemonics like PUSH, POP identify this mechanism; system indirect calls attended to at appropriate links and with more number of stack pointers fits well for data switching in the multiuser timeshared machines.

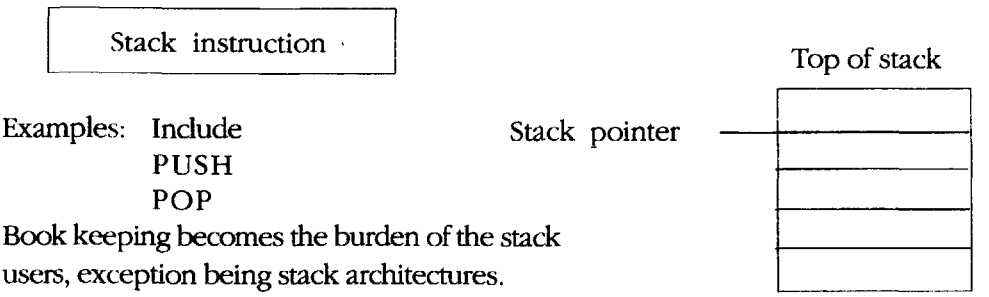



**Fig. 2.3 (d)** Stack addressing

### Implied Addressing

The machine control instructions at assembly level, the system calls at shell level with respect to various operating systems inviting the special interrupt driven and DMA transfer schemes are the embedded properties of computer system.

### Index and Base Register Addressing

Here, some registers specifically assigned the status of Base and index registers are used to assist in data fetching and data parallelism. These methods of addressing give a quantum of values to classify a machine as complex instruction type besides having more application with pipelined actions.

### Relative Addressing

When a program is in execution, the *program counter* register keeps track of the current instruction in use for a total control activity. This relative addressing method can go a long way with PC relative addressing for dynamic multitasking activities suited best for data flow computations for independent and interdependent calculations. Examples lie in distributed data processing in the realm of module linking towards a parallel processing environment.

The above listed addressing mechanisms used in a varying mix of combinations will reflect the instruction set complexity and power of a system.

It is by experience gained, it has been observed that certain code portion of a program is repeatedly used as in do loops, which may include also the system function calls which can be used either iteratively or recursively in an arithmetic bound situation to converge to the solution. Also when the problem is one of data processing of file handling, the operands ( the data) have to be fetched continuously at discrete intervals of time. Under the above situations, the speed of memory will help a long way in enhancing the thruput of the system. In this context, the very high speed memory unit called the *Cache memory*

### Cache Memory and Associative Linking

Fig. 2-4 depicts the instruction time distribution. It is often noticed that the fetch cycle occupies almost all the time whereas execution takes place instantaneously at one clock pulse. The value of the instruction times depends on length of instruction, type of instruction and also at times the inherent delay of synchronising (leading to wait states).

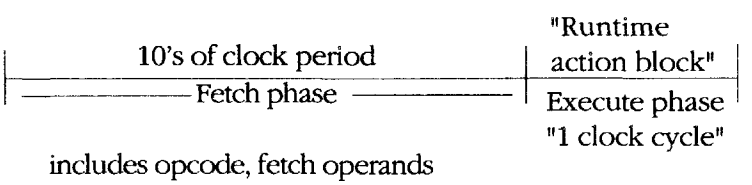



**Fig. 2.4** Instruction cycle

( more often pronounced as CASH memory) has come to stay balancing with the CPU speed so that a good data rate is maintained in the realm of managing good operating systems.

Systems with high performance requirements often use a high-speed buffer, or cache memory between the CPU and the primary semiconductor memories. Instruction look-ahead is mandatory for powerful processors; cache memory will provide this capability as well as a fast scratchpad area for operands. The access time of less than 100n sec. is common for caches compared to 0.5 to 1 μ sec. of the primary memories. While the job is running, the contribution due to cache memory linking are determined by various factors like cache size, the addressing mechanism and replacement policies. The utility of a cache system is often dictated by the hit ratio "h" defined as

$$h = \frac{\text{number of references to the cache}}{\text{total number of memory references}} \times 100$$

for most of the reading operations done by the CPU.

Caches of size 8 to 16 K bytes with block sizes of 64 to 128 bytes and a set associative mapping scheme will yield good hit ratios of over 95% in the realm of maintenance of virtual memory management systems which employ paging and segmentation. The list of above attributes discussed will reflect the general power of a machine.

## 2·3 Language Support Environment

The systems support is decided by the nature and number of programming language environments it can withstand. This calls for secure systems without incurring severe penalties in execution time or storage space. In this context, a batch processing machine

suits to the above points to attract diverging users in an interleaved timing fashion. But all interactive machines are proficient systems by converging to one variety of programming environment in achieving the best out of distributed processing. An operating system, in general, consists of the kernel, the shell and utilities.

For compiler construction is a difficult one which can be made rigid by using formal syntax. Notation that is used in describing the syntax ( Backus - Naur Formalism ( BNF) is a well-known *meta-language* for specifying concrete syntax) or semantics of a language is termed a meta language. Dangling references and uninitialized locations are common insecurities because they are difficult to prevent by language design and their detection by processors can be quite difficult and expensive. Even in the domain of database applications, no standardisation is easily possible to mean a specific language environment at the user's disposal mainly because of the broad band ( range) of specifications to be met in real-time.

So, obviously, it is the burden of the programmer who has to select the language to communicate and sensitize the problem application . This stimulates a rapid and ever increasing demand of software exports with reference to experts in the computing market around the globe.

Thus, precisely, the CPU power will to a greater extent also reflect the consumer selectivity part for architectural designers, because the stated facts of 2-2 and 2-3 have to go in parallel with the guidance of experts and consultant teams. Fig.2-5 gives the trends in computing and communication needs as CPU power is concerned. Fig.2-6 depicts the potential growth and need with reference to software and hardware during the lifetime of computers. With more of interactive and online computer usage, the emphasis is on developing methods for formally specifying programs and verifying that they meet their specifications.This factor has led to developing more reliable programs which reduce the

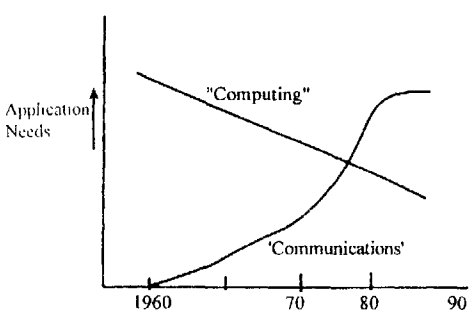



**Fig. 2.5** Trends in computing

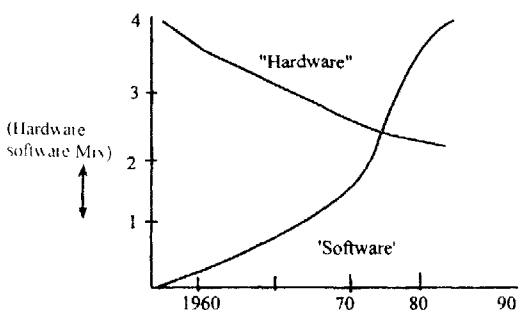



**Fig. 2.6** Potential growth

need for testing and debugging. Formal semantics is a tool which can help language designers achieve their objectives. Program decomposition and reuse of precompiled modules are sought after in differing language environments. Already the packages have captured the customer market with the natural competition coping to the factors like user ease and system friendliness. Working at Assembly language level will truthfully generate good system software in an effective manner. *Parsing* methods are compared by the criteria like generality, ease of writing a grammar, ease of debugging a grammar, Error detection, space requirements, Time requirements and Ease of application of semantics.

STATIC STORAGE ALLOCATION

Here, it is necessary to be able to decide at compile time exactly where each data object will reside at runtime.

In this case, dynamic arrays and variable - length strings are disallowed.

WATF IV is a one - pass compiler. A system or language which describes another language is known as a metalanguage. The metalanguage used to teach German at most universities is English, while the metalanguage used to teach english is english.

Efficiency is important after, not before, reliability.

**N F A**

Non Deterministic finite-state automata, are similar to D F A except that there may be more than one possible state that is attainable from a given state for the same input character.

It is often easier to recover from context-sensitive errors than from context- free errors.

Two methods of parsing are Top - down and Bottom - up schemes.

In general, data-processing applications, do not require a very dynamic run-time environment , and therefore a language like COBOL is often used.

Burroughs 6700, however, support dynamic storage allocation with machine-level instructions. The apparent overhead in dynamic storage addressing is significantly reduced in these machines.

*COMPACTION* works by actually moving blocks of data, etc., from one location in memory to another so as to collect all the free blocks into one large block. An intermediate source form is an internal form of a program created by the compiler while translating the program from a HLL to assembly-level or machine-level code.

e.g Register allocation. Compilers which produce a machine-independent intermediate source form are more portable than those which do not.

A program is adaptable if it can be readily customized to meet several user and system requirements.

Semantic Analysis, Code Optimization and machine dependent adoption are based mainly on the ability in tapping the capabilities of a CPU structure.

P D P - 11 is a 16 bit minicomputer (1970's).
M C 68000 has more addressing modes and registers.

COMPILER - COMPILERS

Compiler - Compilers, also called as compiler generators are used as a tool in this domain. Fig.2-7 shows a translator writing system.

Function : Scanning
Parsing or
code generation

Good lexical analyzers and parsers can now be automatically produced for many programming languages. More work, however, on good error- handling strategies remains to be done.

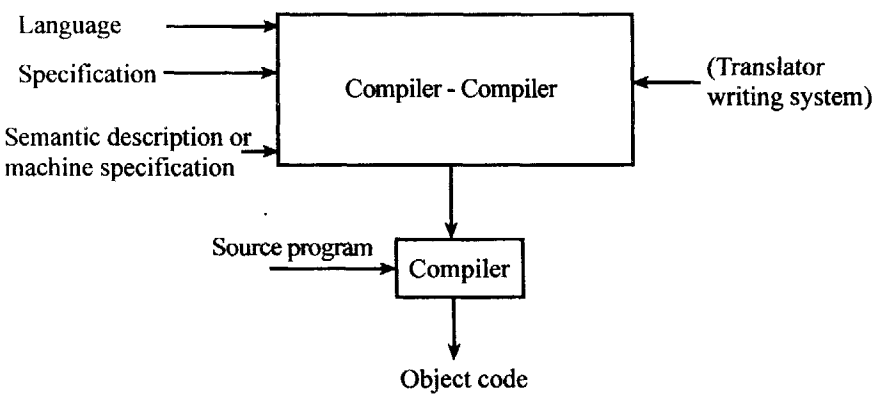



**Fig. 2.7** Compiler - compilers

With the arrival of VLSI technology, the question that often arises concerns what machine should be built to implement a given language. *Compiler - Compilers* will play an important role in answering this question. The CPU resource utility can well be evaluated by making use of standard software metrics which can go a long way in absorbing the power of CPU architectures.

## 2·4 Working Configurations

According to M.J. FLYNN ( 1966), the working of computers fall into the following classifications:

Single instruction single data SISD ;
Single instruction multiple data SIMD ;
Multiple instruction single data MISD ; and
Multiple instruction multiple data MIMD.

In order to achieve computing rates of 100 M flops with adequate precision, the super computers architecture have to depart from the strict von Neumann concept. Hence the decomposition of computations into tasks for distributed processing shall become mandatory, which invites the present concepts and theories on parallel processing trends. Some major attributes like *pipelining*, Vectorization, array processing and algorithms will play a continuing role in the evolving strategies in architectural designs. There are already example systems living to suit the different attributes on date. The above listed varying parameters is discussed in Chapters 4 and 5 of the book. Multiple CPU systems of same nature or differing architectures will find a place in coordinating processes and distributed computing environments to meet the desired turn-around-times.

Presently dedicated microprocessor based instrumentation and real-time systems are evolving due to their reliability and affordable cost. Though the inherent power of CPUS are responsible for system performance, the careful authority of memories (operating system software, in specific) play the dominant role in accelerating the Throughput of a system. More often, it is remarked that memory cost is the machine cost. The organisation of memories is a key factor in radiating the CPU power for user feelings which is taken up in the following chapter 3 on Memory Organisation.

### Keywords

Microprocessor, assembler, portability; Clock, wordlength, relocation, RISC, memory mapped I/O, thruput, opcode, stack, program counter, cache memory, metalanguage, parsing, compaction, compiler-compiler; pipelining.

## PROBLEMS

1. Distinguish debugging by means of breakpoints with single step operation on assembler level.
2. Correlate the usage of high level language programs and assembly language programs from the eyes of varying users for ease of use.
3. What is a Cross - Assembler?
4. What is in-circuit emulation for language portability on microcomputer development systems?
5. Comment on the length of instruction cycles and clock speed for CPU designs.
6. Write on the use of stacks at assembly language level as an architect.
7. What are the attributes of an instruction format?
8. Devise an algorithm for binary to BCD conversion.
9. Bring out the differences between I/O program-controlled transfer and DMA transfer.
10. Explain instruction word and data word formats.
11. Explain immediate and indirect addressing modes with assembly level instructions.
12. Attach the usefulness of bit slice processors for compute-time reliability with single user serial organisations.
13. What is a RISC machine?
14. What is "LABEL" declaration in programs? How does this help in relocation?
15. Is stack addressing mode often referred as zero address instruction? If so, why?
16. Explain the use of parsing tables in the recognition of syntax.
17. Write notes on error detection and recovery phase of a compiler.
18. Write notes on catching the syntax versus semantic errors at compute time, valuing the measurements as a software engineer.
19. Mention and explain some dynamic debugging tools for program interaction.
20. Explain the role of multiprocessors highspeed computing. Also distinguish loosely coupled and tightly coupled multiprocessor systems.
21. Mention the role played by demultiplexer in interrupt I/O systems. What is parallel I/O activity?
22. Enumerate the factors contributing to the Turn around time on a distributed processing bench.
23. Account for the usage of integer arithmetic towards inaccuracies in real accounting applications.

# CHAPTER - 3

## MEMORY ORGANISATION

The stored program concept is on the Zenith assuming a widely varying parameters like memory speeds, store reliability, computational complexity, well standardised application bases and cost-effectiveness.

Whereas the technology helps to meet the speed and reliability aspects, the domain of intelligent *algorithms* for system storage management and user algorithms towards program maintenance in the perspectives of software engineering will certainly reflect the ways of improving memory efficiency on a generalised sense. It is often said that memory cost is the machine cost; thus the memory size the system supports is a major criterion besides the slowly refined virtual memory management towards good I/O management policies in real-time.

Under the general context the memory is distributed all over a computer installation, whereas the dedicated end-usage confined to low compute needs and flexible process control areas dictate much on a hardware biased design with memory locatable *microcontrollers* in the Application specific Integrated Circuits group (ASICs).

Let us begin giving credit to the electronics technology since its inception in 1940's and then categorically discuss on the storage capabilities.

### 3 · 1 Semi Conductors and Memory Technology

Electronics is the study of the behaviour of charged particles called electrons at differing environments. Let us start stating a few terms in nuclear physics.

An *electron* is an elementary particle having a rest mass of $9.1091 \times 10^{-31}$ kg, 1/1836 that of a hydrogen atom, and bearing a negative electric charge of $1.6021 \times 10^{-19}$ coulombs ($4.80298 \times 10^{-10}$ ESU).

Electrostatic unit, ESU of charge called statcoulomb is that quantity of electricity which will repel an equal quantity, 1 cm distant from it in vacuum, with a force of 1 dyne.

Radius of electron is $2.81777 \times 10^{-15}$ metres. A hole may be regarded as a mobile vacancy with a positive electronic charge and a positive mass; it is thus mathematically equivalent to a positron.

A free electron is one which has been detached from its atomic orbit.

*Electron - volt*, eV is the unit of energy used in nuclear physics. The increase in energy or the work done on an electron when passing through a potential rise of 1 volt.

1 eV = $1.6021 \times 10^{-19}$ joules

1 MeV = $10^6$ eV;

1 GeV = $10^9$ eV.

Valency is the combining power of an atom of an element, the number of hydrogen atoms which an atom will combine with or replace; for example, the valency of oxygen in water, $H_20$, is 2.

*Semiconductor* is an electrical conductor whose resistance decreases with rising temperature and presence of impurities, in contrast to normal metallic conductors for which the reverse is true. The main semiconductors include Germanium, silicon, Selenium and Lead - Telluride. Pure or intrinsic semiconductors have negligible conductivity and the conduction has little flexibility; the process of adding impurity atoms of adjacent valency groups is called *Doping* and the extrinsic semiconductors so obtained become valuable.

Germanium and Silicon are tetravalent. When pentavalent impurity atoms such as phosphorous, antimony, arsenic or bismuth are used in doping will result in n-type semiconductor. The trivalent impurities include boron, gallium, indium and aluminium.

As temperature of a semiconductor increases the mobilities $\mu_n$ and $\mu_p$ decrease, but the decrease in mobility is more than offset by the increase in the carrier densities n and p and the resistivity decreases with increase in temperature. Thus, the semiconductors are also used in temperature measuring devices.

The compounds of Germanium are rare. It is a brittle white metal and used in transistors. Silicon is a nonmetal similar to carbon in its chemical properties. This occurs in two allotropic forms, viz., a brown amorphous powder and dark grey crystals. This is used in alloys and in the form of silicates in glass. Silicones are also widely used. Some of the properties of basic semiconductor materials.

| | *Germanium* | *Silicon* |
|---|---|---|
| Valency | 4 | 4 |
| Atomic number | 32 | 14 |
| Atomic weight | 72.59 | 28.086 |
| Specific gravity | 5.35 | 2.42 |
| Melting point | 937.4°C | 1420°C |

A p–n semiconductor junction is formed when there is a change along the length of a crystal from one type of impurity to the other. At a p - n junction, an internal electric field is created between the charged impurity ions of the two types. This field is sufficient to prevent the drift of electrons from the n side to the p side of the junction, and the drift of holes in the opposite direction.

The *cut-in voltage* $E_\gamma$ of p - n diode for Germanium is 0.1 to 0.2 volt whereas for silicon it is 0.6 to 0.7 volt. The larger delay in switching characteristics of silicon accounts for germanium being used in faster digital circuits. For reasons of more power dissipation capabilities and ease of fabrication, invariably silicon is used in memory constructions. N - type devices are faster because electrons mobility is greater than hole mobility.

**Table 3.1** Compares the operational features of the versatile transistor technology.

| *Conventional Junction Transistor - Bipolar* | *Field-effect transistor - Unipolar* |
|---|---|
| Input is forward biased so low Rin | High Rin |
| Current operated device | Voltage controlled |
| Finite offset voltage | No offset voltage. |
| Used in CPU storage | High fanout, capacity and higher Less |
| memory capacity | packing density. |

Certain typical parameters for standard IC families are charted in Table 3.2.

**Table 3.2** Typical parameters for standard I C families

| *Logic family* | *Fan-out* | *power dissipation (mw )* | *propagation ns delay* | *Noise V margin* |
|---|---|---|---|---|
| Standard TTL | 10 | 10 | 10 | 0.4 |
| Schottky | 10 | 22 | 3 | 0.4 |
| Lowpower schottkyTTL | 20 | 2 | 10 | 0.4 |
| E C L | 25 | 25 to55 | 2-15 | 0.2 |
| C M O S | 50 | 0.1 | 25 | 3 |

E C L non saturated logic finds applications in high speed computers while C M O S is preferred in microprocessors since its inception in 1970's.

Georg OHM (1787 - 1854) is responsible for all electrical measurements thanks to the Ohm's Law.

"The ratio of the potential difference between the ends of a conductor and the current flowing in the conductor is constant", given by $E/I \alpha R$ ohms.

Subsequently, Gustav Robert *Kirchhoff* (1824 - 1887) established his well proved electrical theorems, namely, Kirchhoff's voltage law. "The sum of the voltages around a closed loop is Zero."

Kirchhoff's current law:

*"The sum of the currents entering a node equals the sum of currents leaving that node".*

The above axioms find a great deal of efforts in designing integrated circuits in V L S I area. Reduced cost and size, improved reliability and speed of operation, and increased packing density are among the technological advancements catering to the modern memory-based digital systems.

### 3 ·2 Primary or Main Memory Unit

Besides the precious and limited processor memory also called *registers*, the adjacent working memories during a process are termed primary memories. The abilities of C P U registers include storage, counting and shift operations. They are constructed of very high speed logic to meet processor - memory bandwidths on computing platforms. The registers are more often labelled for dedicated activities while a few may incorporate the flexibility for programmability on well designed programmed machines.

Since more C P U in storage demands for larger space on the Integrated chip and high number of connectivity shells with more on-process book- keeping, the arithmetic logic units may become underutilized in processor management, a key function for any operating system. In this context, the variety of semiconductor memories and advantageous fast access stores like cache memories for real immediate process needs have come to stay on single stand-alone systems of larger wordlenths. In early systems, the ferrite core served as main store as in the T D C - 312, a 12 bit machine which confined a byte to be made of six bits. The main merit of these batch processing single job machines was the nonvolatility offered by core memories for a safe storage under power failures. But the standard *byte* refers to a set of 8 bits in present day memory organisations. The cores are the only magnetic memories employed earlier as main stores.

The 1 bit storage unit consits of a small toroidal (ring - shaped )piece of magnetic material (ceramic). Minimum dimension for typical ferrite cores are 0.5 mm outside diameter with a central hole of 0.4 mm and thickness of 0.15 mm. Switching speed increases with reduction in core size.

Fig.3-1. depicts the core with the square-loop characteristics. A substantial current $I_m$ of the order of several mA is required to cause switching. The TDC –312 used main memory of size 4 K words where 1 word = 12 bits. The maximum memory size is known to be 128 K of word lengths reaching upto 32 bits or more with core storage.

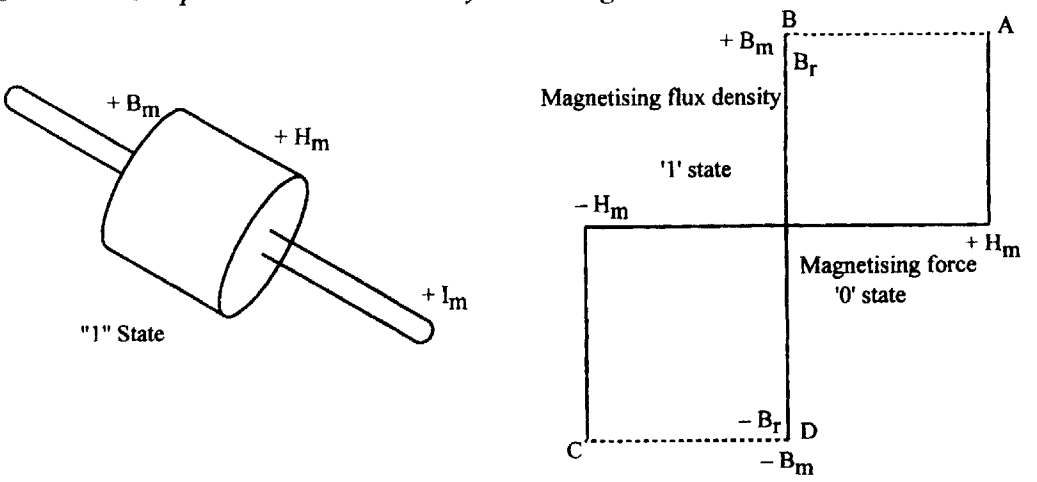



**Fig. 3.1** Square loop characteristics

The core memory organisation can be categorised as follows:- Linear select and conincident current memory. This is *word organised* two-dimensional store. Fig. 3-2. shows the 2-D organisation.

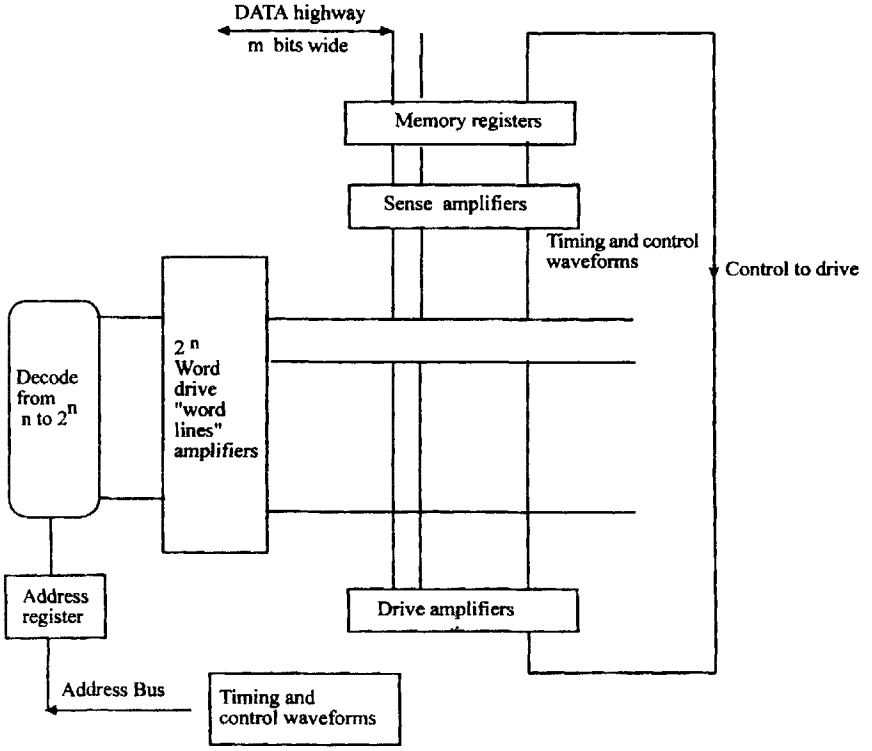



**Fig. 3.2** Linear 2D select organisation

The sensing or reading memory is of destructive type and more generally the word has to be rewritten after every read operation. Similarly the write operation on selected cores is followed by the verification of the data register contents for faithful systems. Three wires are used across each core thus making 2D memories faster. But it requires $2^n$ word drive amplifiers and m bit drivers (sense amplfiers).

Coincident current memory Three-*D organisation*:-

This is also called bit organised space. The store is arranged as a set of planes equal to m (word size of machine).

This uses m Sense wires and m inhibit drivers embedding each core. The core-size becomes large and reduces the achievable speed in memory transactions.

For N, the number of memory locations assumed to be exactly perfect square, this method needs $2\sqrt{N}$ drivers for rows and columns preserved to be a square matrix.

For illustration, a 64 x 8 word organisation in 2D becomes 8 x 8 x 8 bit organised in 3D, thus reducing the associated drive circuitry very much for large values of N.

2 1/2 D memory technique :-

This is employed for fast operation with large size memory. One core plane is used for each bit of the word and coincident current technique is employed as in 3 D. But each plane requires separate x drivers. The core arrays need not be square. The y column drivers are shared by all the core planes. M sense wires are used, one per core array. Thus, this meets the dual effects of 2 D and 3 D with homogeneous construction. The maximum access time is product of clock period and loop bit size.

**Example**: A system with frequency of 20 M H Z has a memory density of 8 K. What is ATmax?

$$\text{AT max} = \frac{1}{20 \times 10^6} \times 8192 = 409.6\ \mu s.$$

It there are 8 planes, then

$$\text{AT max} = \frac{409.6}{8} = 51.2\ \mu s.$$

Corresponding to the 21/2 D technique.

Further trends for magnetic core as main stores take different forms like list below:
Thin film memory; superconductive or cryogenic memory, plated - wire type and magnetic bubbles.

Magnetic film memories have very thin flat layer of nickel iron alloy as the memory element. Along one geometrical axis magnetization is easy. They can be arranged in 2-D form.

*Cryogenic* memory has large packing density of the order of 15,000 cells per square inch and also fast switching speed. Some metals have very low resistance ($\cong 0\ \Omega$) at very low temperatures. The switching from super conductive to conductive state is used to designate the 1 and 0 binary values. Coincident current selection is provided for organising data on this. But it requires a very low temperature container (near $4^0$K).

Plated wires are similar to thin film memory. Magnetization is easy along the wire circumference and hard along the axis. 2–D scheme is preferred.

Magnetic bubbles are static devices which has the property that data can be recorded by the polarity of small zones and the zones get shifted along a path by applied field (shift register). It offers high density, reliability at low cost.

With the advent of microprocessors in 1970 the semiconductors have taken over the main memory functions. It holds the current program in execution besides accommodating the operating system, compilers/Assemblers which occupy a significant fraction of the total random access store with a high activity ratio. The semiconductor or electronic memories are all considered to be of nondestructive nature, random access type and volatile but for the ROM (read only memory) category.

A memory is said to be volatile when its contents are lost with power loss. The access time considered to be one of the primary parameters for a good system *throughput* and turn around time is fastest for main memories and is *random* meaning the time required to *access* any portion or part of the store is identical. The foremost factor considered is the clock of a C P U on a synchronous system where all the activities of processing is governed by the system clock. Thus, today C P Us with clocks ranging from 1 M H Z to 60 M H Z is not uncommon. Crystal clocks are preferred and employed for they are very accurate and precise to establish a total control over the operating system. The improvement in clock speed also implies a good feed rate possible to the arithmetic logic units.

A computer system is considered to be strictly a sequential machine involving both synchronous ans asynchronous activities which calls for electronic clocks in binary systems. In this sense, a clock is an astable multivibrator (relaxation Oscillator) implemented using electronic devices and circuits. In general, power dissipation of a logic circuit is defined as the supply power required for the gate to operate with a 50% duty cycle at a specific frequency (varies from few μw to 50mw/gate). Fig.3.3.gives the circuit for an *astable* multivibrator.

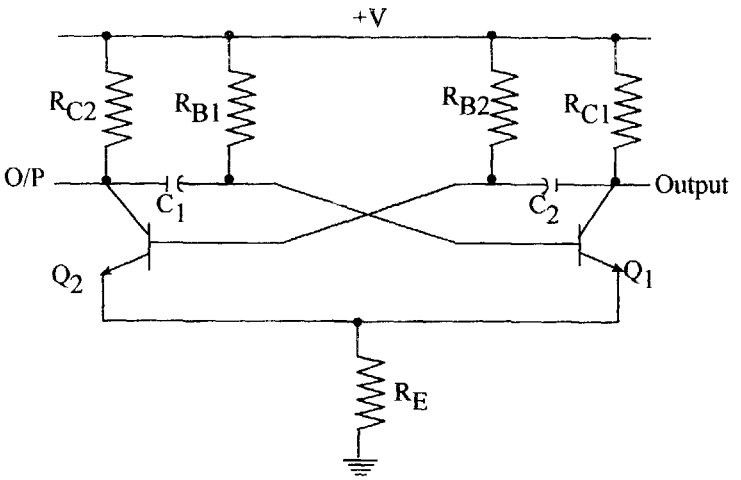


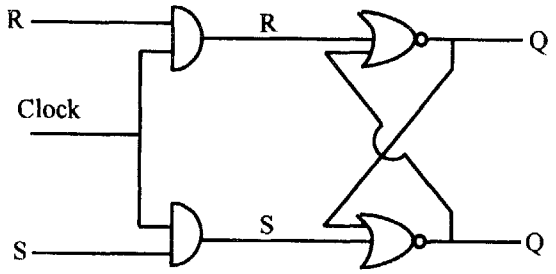

**Fig. 3.3** Circuit of astable multivibrator

The ON and OFF times can be made equal by using $R B_1 = R B_2 \cong R_B$ ; and $C_1 = C_2 \cong C$, the time period of the square wave is $T \cong 1.4$ RB.C. Moreover, the circuit operates with $Q_1$ and $Q_2$ switching in opposite states thus providing also the inverted clock for use in master-slave flipflops devised to provide buffering and avoid racing in sequential machines.

A *monostable* multivibrator or a one-shot is stable in one of the states but unstable in the other state. The IC 555 is one which can be employed both as an astable or monostable multivibrator with varying duty cycles over a wide range of frequencies. Usually the square wave is differentiated by a simple R C network to generate triggering pulse (as clocks) to operate various types of flipflops in synchronous machines. For C P U activities, synchronous design is simple at the cost of the idle time of processing elements. In the domain of parallel processing for pipelining activities on a single sequential machine, the asynchronous operation can be well achieved with careful design for idling C P U. This sort of a pipelined approach also calls for cache memory used as slave-store in the powerful I/O activities concerned of the central processing elements.

A bistable multivibrator is stable in both the'1' and '0' states which becomes an elementary cell in a primary semiconductor memory. The circuit behaviour strictly follows the Newton's first law of thermodynamics, namely, "every body tends to remain in its state until and unless an external force attempts to change the state under supervision". Thus the bistable circuit is more often known by the term "*flipflop*" which are usually clocked for their applications. Before delving into memory organisations, the various types of flipflops are explained to advantage.

A bit is a binary digit which can be either 0 or 1. A byte is considered to be a set of 8 bits while a nibble is made up of 4 bits. Fig.3-4. shows the circuit for a clocked RS flipflop.




Excitation Table

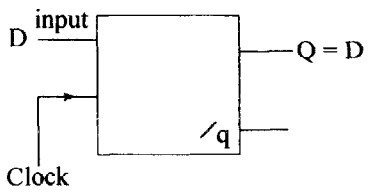
| S | R | Q n+1 |
|---|---|---|
| 0 | 0 | Qn |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

**S.R ≠ 1**

S = R = 1 is not allowed

**Fig. 3.4** Circuit of a clocked RS flipflop

The flipflop provides its status both in true and complemented form Q and $\overline{Q}$, the excitation S = R = 1 is never possible by the definition of a bistable multivibrator. Moreover, observe the change in the next state Qn + 1 only with the application of a clock pulse. Fig.3-5(a) shows the block of a D filpflop which is a derived one from the S R flipflop.



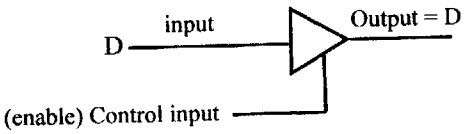




**Fig. 3.5(a)** D flipflop **(b)** Buffer tristate logic

Here Q = D is the characteristic equation and just the R input is inverted and fed to the S input of a S R flipflop. This is known as voltage follower and used in buffering. With tristate logic as shown in Fig 3.5 (b), this D-type latch acts as a temporary storage for more of the interrupt driven information. These latches are much valuable in microprocessor based systems for the interacting I/O actions.

**Excitation Table**

| J | K | $Q_{n+1}$ |
|---|---|---|
| 0 | 0 | $Q_n$ |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | $\overline{Q_n}$ |

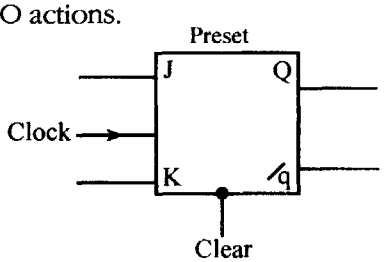



**Fig. 3.6** JK flipflop with excitation table

The more versatile JK flipflop allows the forbidden combination of S = R = 1 of a S R flipflop making J K meaningful for register storage within a microprocessor. Fig.3.6 shows the block symbol of JK flipflop with the excitation table.

More often, the flipflops are provided with clearing options and preset to any desired state as shown. The master/slave concept is often used for buffering and to avoid racing in sequential circuits. The JK flipflops are used in counters and shift registers for rotating displays. A register is supposed to consist of 'm' flipflops where m denotes the CPU word length.

The registers and counters with varying degrees of abilities are constructed by careful design using the sequential logic circuits to make the storage well organized. The T flipflop is one which accepts a clock of frequency, fin and provides an output waveform (square) with frequency $f_{OUT}$ = fin/2, which can be employed for timing control in sequential systems. In general, n "T" flipflops in cascade will provide an output frequency $f_{OUT} = f_{in} / 2^n$.

Thus the various types of flipflops provide a sitting status for the main store elements and their dynamic usage in complex architectures. In addition, the P S W (program status word) dictates and reflects the status of the current user program being executed on the machine by means of powerful Flags. Only certain flags can be preset at user's will like the carry flag of the 8085 microprocessor. The other *flags* include Zero, Auxiliary carry, parity and sign. 8086 has nine flags in total. Also the processor status is indicated by the process effects or defects more on a batch system by indicating if the machine is in Fetch, Execute, Halt or other states for the system operator to interact on a regular basis as of a supervising robot. No doubt, the computer is a faithful worker as long as the system behaves well like meeting the Asimov's law of robotics. Hence, modern computer systems must incorporate detecting tampering of machines by any means (example software viruses, etc) and make alarms and shut down the process for preventive maintenance to qualify to be a secure system done to safety. The sequential state minimisation is further conveyed in chapter 1. and fault - tolerance addressed in chapter 5. Fig.3.7 shows the J.K. flipflops used as mod - 8 ripple counter.

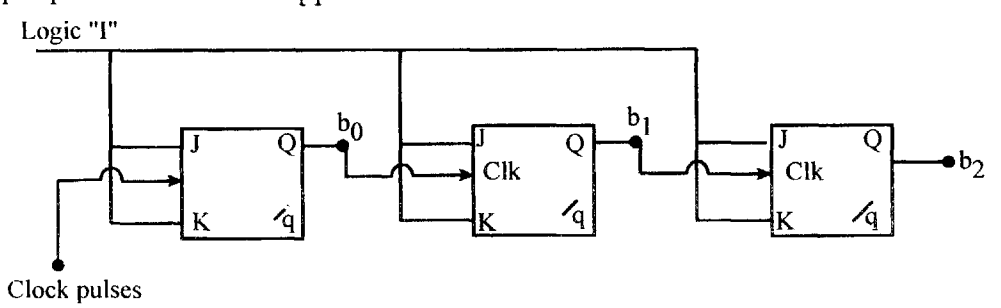



**Fig. 3.7** JK flipflops used as mod-8 ripple counter

Just using the complemented output $\overline{Q}$ to clocks will imply a down counter. Any counter to m states, mod - m counter, needs n flipflops for realization, where, ($2^{n-1} < m \leq 2^n$).

Most of the registers and counters are also available as M S I chips. Multiplexers and Demultiplexers again play a vital role in addressing the total STORES of a computer system when access can be fruitfully achieved.

A *decoder* takes a bit pattern as input and selects only one of several output address lines at any point of time serving as a hardware unit during semiconductor memory access. Instruction cycle decoding and machine cycles encoding is a regular procedure at runtimes to present useful expected outputs of a computer which otherwise is impossible in real-time. The amazing real-time areas where computers have carried a name include online Reservation systems, data base management for information retrieval, computations for space exploration and image transmissions in global communications and computer networks.

A *random access memory* ( RAM ) is a Read/ Write memory which forms the major portion of the addressable memory space. It is volatile; the read out is nondestructive. All electronic memories follow the random access method. Presently these are byte organised and only the instruction takes care of the number of bytes accessed on every instruction. Fig.3.8. gives the byte organized nature of electronic memories.

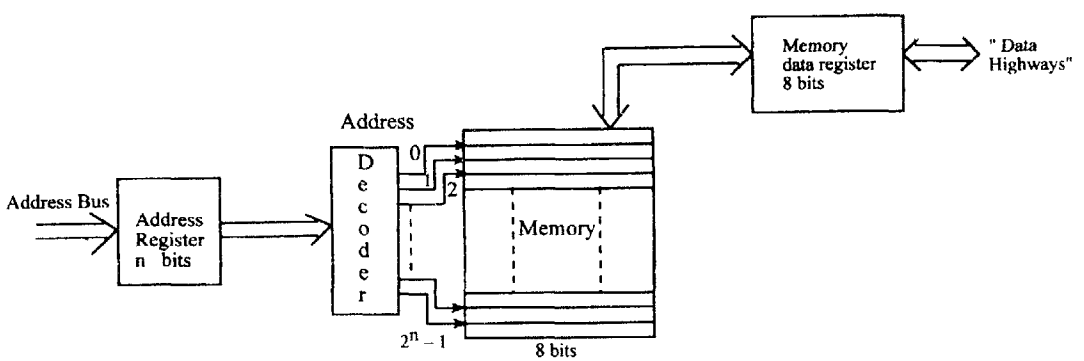



**Fig. 3.8** Electronic memories

The memory chip must have provision for read/write control, data input lines, sense data outputs, and address input pins to access anyone location of the chip capacity at any point of time as shown.

A 12 to $2^{12}$ line decoder is far expensive than a 6 to $2^6$ line decoder. This calls for differing 3-D, $2^{1/2}$D organisations as well as increasing the component counts and reducing and with each location storing 16 bits is called a "16-bit 4096-word" memory, or in the vocabulary of computer trade a "4K 16-bit"memory. 8085 C P U can address to a maximum of 64 K bytes of memory, whereas 8086 with 20-bit address bus accommodates 1 mega bytes of memory. 8086 also uses segment registers and partitions the memory user space into 4 x 64 K bytes partitions for parallel data bus, $AD_7$ - $AD_0$ meeting both needs. At times, some of the memory addresses are used for secondary and end termials (I/O units) which is called *memory-mapped I/O*. This concept may also be useful in process instrumentation area. The 8085 C P U uses I/O mapped I/O with IO/$\overline{M}$ pin indicating the address redirection. But today with reliable and fast switching areas of communications, Bit slice processors and bit organisation of memory become irresistable for error control.

The bipolar RAM has less access time of the order of 10 ns but it needs large area and has high power dissipation.

The MOS RAM has access time of the order of 1 μs but needs less area and has low power dissipation. Fast memories are used in cache stores whereas the MOS technology fits the usual program store. Gold doping to reduce transistor recovery time as well as schottky diode clamping to prevent transistor saturation are used with TTL configuations to increase speed. ECL is a non saturated logic offering very high speed at a higher cost/ bit. Table.3.3 depicts the factors associated with various memory technologies.

**Table 3.3** Memory performance data.

| | *MOS* | *n- channel* | | |
|---|---|---|---|---|
| | *p-channel* | *High Speed* | *High density* | *Bipolar* |
| Number of bits/chip | 4096 | 4096 | 16,384 | 2048 |
| Access time,ns | 300 to 400 | 50 | 150 | 20 |
| Power dissipation mw/bit | 0.2 | 0.1 | 0.02 | 0.25 |

Fig. 3.9 shows a static MOSRAM cell.

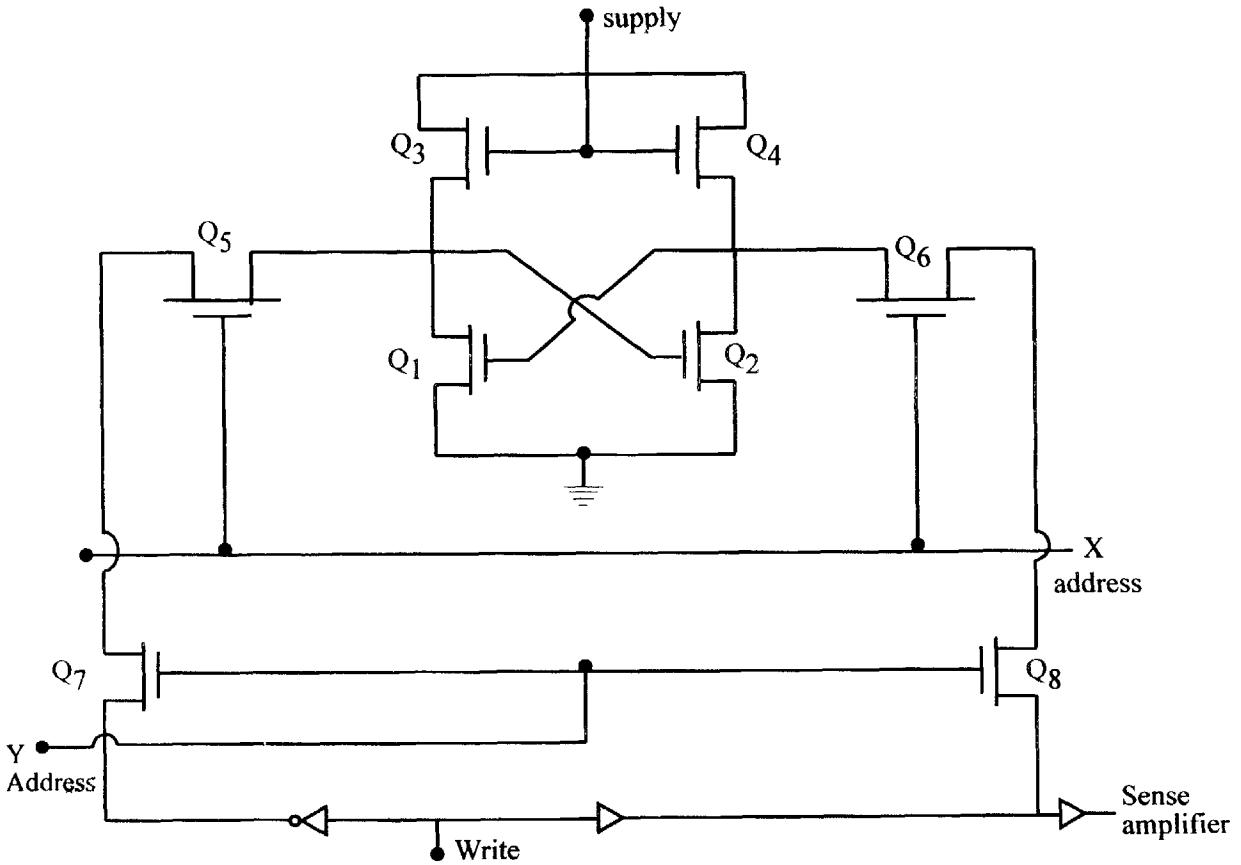



**Fig. 3.9** Static MOS RAM cell.

$Q_1$ and $Q_2$ form the flipflop whi ch is the basis of the storage cell. $Q_3$ and $Q_4$ sense as pull up resistors for transistors $Q_1$ and $Q_2$.

If coincident memory current selection is used, a part of the address is decoded to choose a particular row (x address). The X address line controls $Q_5$ And $Q_6$ of all cells in the appropriate row. The remainder of the address is decoded to determine the particular column (Y address) that is to be read. The output from decoder is applied to the y address line and controls $Q_7$ and $Q_8$ of the appropriate column of cells. If $Q_5$ through $Q_8$ are turned ON, a current will flow in one and only one of the two bit lines (X, Y select).

A sense amplifier is provided to convert the current signal into a voltage level. The state of the flipflop ($Q_1$ and $Q_2$) is not altered by the R E A D operation. In order to perform the W R I T E operation, the proper cell must be addressed ($Q_5$ through $Q_8$ turned O N ) and the appropriate voltage levels must be applied to the bit lines by the W

R I T E amplifiers. This sets the desired state of the flipflop. Static M O S memories consume less power and have high packing densities.
Dynamic MOS memory cell is shown in Fig.3.10

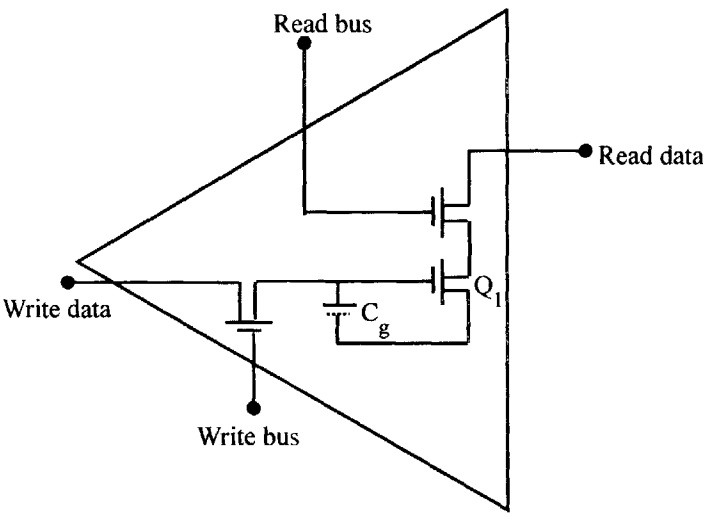
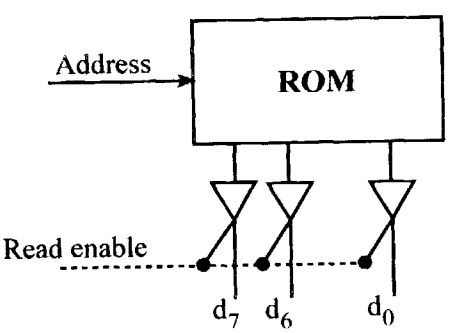



**Fig. 3.10** Dynamic M O S memory cell

In *dynamic memories* restricted to M O S technology, storage takes place as charge on gate substrate capacitance. Refresh is required once every few ms, and consumes lesser power. This also offers high speed relative to static cells. Extra timing and logic circuitry is needed for refresh cycles. To write into the cell requires holding the write bus at a low logic level; then a low level at the write data input charges the gate capacitance; stores a 1 in the cell. With the write bus held low, and a high logic (Vcc) at the write data input discharges the gate capacitance. To read from the cell requires holding the read bus input at a low logic level; if the gate capacitance is charged, the read data line goes to + Vss. It offers store capacity of about 16,000 cells/IC.

The next type of powerful semiconductor memory is ROM (read only) which is utilized for storing monitor programs, Table look-up entries, File management data of disk stores, sub routines, dedicated software for microprocessor based instrumentation systems and so on. The ROM is essentially nonvolatile and frequently restricted to MOS technology. Microprogrammed ROM's, at times meet the complex arithmetic operations in *RISC* architectures meaning more in-storage of CPU being utilised successfully. Fig.3-11 shows the block for a 1 K byte ROM.



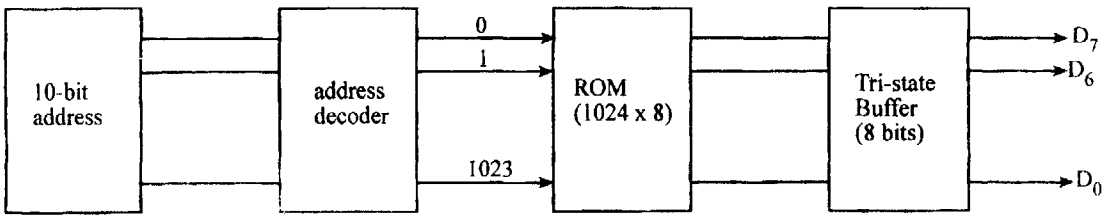



**Fig. 3.11** Block diagram of 1KB ROM

At times also diode ROMs are employed in routine advertisements and displays. A more flexible approach is allowing people to program the ROM which insists on PROM that makes use of fusible links which can be broken when a high current is passed through it. Initially every bit is set to 1.

An *EPROM* (Erasable programmable ROM) is a chip, wherein the fusible links can be regenerated with exposure to ultraviolet light so that it can be reprogrammed. But even to change a few cells, the whole chip must be erased.

EAROM s (electrically alterable) give the advantage of making even a few changes due to mistakes done earlier while programming. These are the various types of read only store which improves the user volume by leaps and bounds to the electronic stores.

### Cache Memory

The speed offered by these chips (cache memory) are compatible with the processor elements. It just takes few 10's of ns for accessing the cache area. The size of cache is low compared to the primary RAM area and this often serves as a copy (image) of the main store as well an output buffer in meeting the memory and processor bandwidths. The timing differences is governed by the adequate equation $t_p < t_m < t_d$, where $t_p$ is the time spent on processing units, $t_m$ is primary memory access time and $t_d$ is the devices interrupting times. With this variable ratio of $t_p : t_m : t_d$, today there exists multiuser terminals on a single C P U machine like the UNIX environment, more often catering to uniprogramming (one language) environment at any particular point of time. Also on batch processing machines, the CACHEs play a crucial role of voluminous data processing in meeting the desired throughputs of machines.

The major applications of *caches* include, storing the current portions of program in use on a selective mode with a sensitive approach; accommodating the executable subprograms; storing input data like arrays and data files; in search applications the relevant files with good virtual memory management and also CPU output buffering activities.

Every time an address is given out of the CPU, the availability is first checked in the Cache area; if available it implies a Cache Hit. Otherwise it goes usually with the main store seeking for the data. Today systems have been well designed to achieve cache hit ratios of more than 90%.

$$\text{Cache } \textit{Hit ratio} = \frac{\text{Total no. of cache hits}}{\text{Total no. of memory references}}$$

This portion is also employed as content addressable memory (CAM). Here, only certain bits are selectively mapped for faster decisions. In CAM, read operation follows a MATCH operation. Thus, matched words are obtained as read output. Complex operations like less than, greater than, next lower, next higher are also performed with CAMs. The *PERFORM* and SEARCH ALL verbs of COBOL remind this feature. SORT and *MERGE* utilities on data files again implies the cache area fitting to both program and data files.

This is also referred in the literature as content addressable memory. The function of the operating systems become much more effective on process management with the inclusion of cache memories. It has become compulsory of every interactive on-line machine (including a PC) to include the powerful cache storage. We can comment only saying that cache is a cash transaction while the usual RAMs are cheque driven in terms of access speeds.

Similarly, a microprogrammed control unit utilizes a ROM to store binary control information. Mask programmable ROMs are also used for combinational logic circuits realizations. ASCII to EBCDIC code converters use ROMs for portability across machines and compatibility for printing data outputs. The visual display using CRTs more often accommodate upto 4 k bytes of semiconductor memory with EBAMs technology to cater to the ever interactive DMA environment on interrupted machines. The front end processor acting as a channel gathers input from terminals and routing results back to users will relieve the main CPU in order to concentrate on the real computing tasks ahead of it. Thus the main CPU has to deal with one interrupt, that of the block transfers. By inclusion of the front end (I/O) processors and a good blend of electronic memories, the best performance output can be achieved in computer applications.

**Programmable logic array**

In the realm of combinational logic designs where the number of don't care conditions is excessive, it is more economical to use LSI chip *PLA*.

A PLA is similar to ROM in concept ; however, it does not provide full decoding of the variables and does not generate all the minterms as in the ROM. In the PLA, the decoder is replaced by a group of AND gates, each of which can be programmed to generate a product term of the input variables.

The AND and OR gates inside the PLA are initially fabricated with links among them. The specific boolean funtions are implemented in SOP form by opening appropriate links and leaving the desired connections. Fig. 3.12 gives the PLA block diagram.

It consists of n inputs, m outputs, K product terms, and m sum terms. The product terms constitute a group of K AND gates and the sum terms constitute a group of m OR gates. Links are inserted between all n inputs and their complement values to each of the AND gates. Links are also provided between the outputs of the AND gates and the inputs of the OR gates. Another set of links in the output inverters allows the output function to be generated either in the AND-OR form or in the AND - OR - INVERT form.

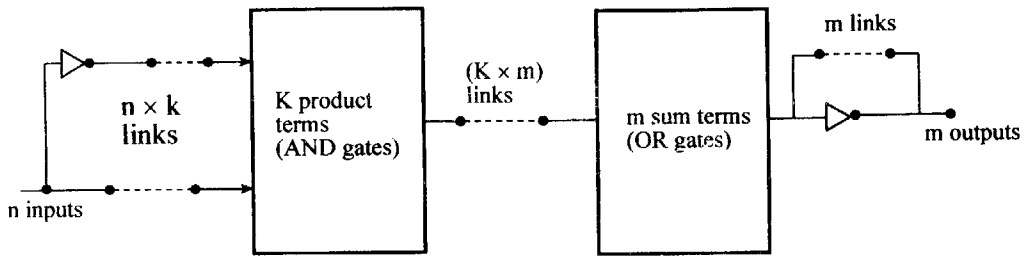



**Fig. 3.12** PLA Block Diagram

The size of the PLA is specified by the number of inputs, the number of product terms and the number of outputs (the number of sum terms is = the number of outputs).

A typical PLA has 16 inputs, 48 product terms and 8 outputs (TTL IC type 82S100). The number of programmed links is ((2n x K) + (k x m) + m) whereas that of a ROM is ($2^n$ x m).

With the chip mentioned, (32 * 48) + (48 * 8) + 8 = 1928 links.

Field PLA can be programmed by the user by means of certain recommended procedures. PALs are referred to as programmable gate arrays. A Typical PAL may have 10 inputs, 8 outputs and two product terms per output. A PAL differs from a PLA that the number of terms per output is fixed, imposing more constraints on the designer. Some PALs are available with feedback and others with registered output and feedback. A PAL having 8 inputs and 8 outputs of which 6 are registered, with all outputs fedback uses two 1 of 8 multiplexers and five NOR gates.

In arithmetic for data representation, use of the E format not only accommodates a wide range of values but also simplifies the computational requirement by effective storage utilization and optimization of machine resources. The algorithmics area for *arithmetic*

again give a big profit on compute base environments. In numerical methods involving lot of iterations which has to converge either based on error set limits or on time bound criterion, the cache memories are a boon to application engineers. Thus computer science is a domain that can be augmented and assisted in many a problem by giant mathematicians. Thus machines in future will really meet the concept of number crunching activities. Fig.3.13 shows memory arrangement of a computing system.

The cost of a machine is very much dependent on the main memory size it supports. Starting with 4 K of RAM, today computers can support mega to gigabytes of this valuable storage besides catering to bulk amount of secondary or auxiliary data-points. With disk operating systems, the hard disks have begun to occupy a more or less prominent place in real computers that can accept a variety of Languages.

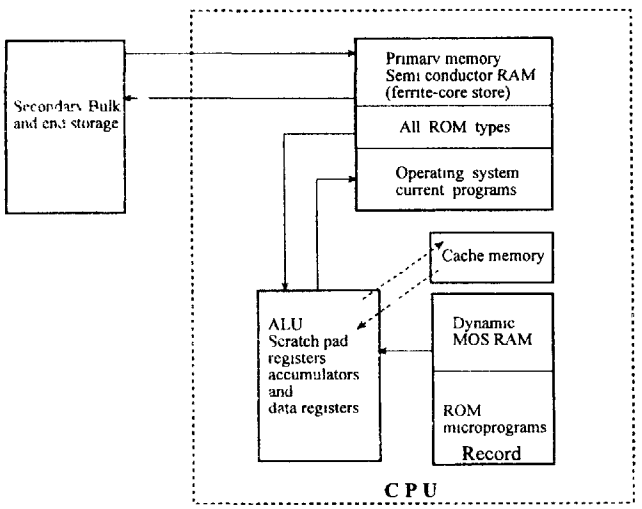



**Fig.3.13** Memory arrangement of a computing system.

### 3.3 Secondary and Auxiliary Storage

With the cyclic access store, large amounts of storage are involved and many a time are retrieved in blocks containing many words. Digital magnetic memories are confined to store medium like magnetic cassette and cartridge tapes, Magnetic Drum and Diskettes. The main parameter distinguishing them is the type of access, either serial or sequential. With these storing elements, a very large volume of data can be stored indefinitely due to the non-volatile nature of this auxiliary media. The system programs, compilers, packages, various editors, diagnostic and debugging tools including the operating systems reside on these disk and tape mountings on large installations, besides the ever growing file space for the user community. Thus rightly, this place could be considered the memory place for the fuel as well as food for the turn-around-time of any job. *Turn-around-time* is the of

a source file to the generation of the desired output results. Even the spooling is an activity which requires this mass medium for a batch printout at a later time on multiuser platforms.

## Magnetic Tape Memory

These are used as external storage devices and as high speed I/O system. They provide serial access. The total body of information passes a reading device in serial fashion, and the average retrieval time is half the total time necessary for a complete pass of data recordings. The home entertainment units require high fidelity amplifier and Supersonic bias used to achieve linearity, with multimedia trends. The computer tapes store data as 1 and 0 using return to zero (RZ) and NRZ (non return to Zero) techniques. It stores individual alphanumeric characters across the tape in 8-bit ASCII code, one track to establish uniform parity in reliable systems. The batch IBM machines use the standard 8 bit *EBCDIC* (pronounced as eb-si-dik) code for data processing. The computer tapes are generally wider and longer than tapes of entertainment units typically 2400 to 3600 feet long and 1/2 to 1" in width. Plastic tapes coated with metal oxide on one side is used for durability. 20 million bits of information can be stored on a simple magnetic tape unit. Data is read/written simultaneously on all tracks but recorded serially in each track. Tape speed is about 100 inches/ second with a capacity of 1200 characters/inch. Fig.3.14 shows the tape drive mechanism. R/W heads can be single gap or dual gap heads. The information from store registers is written on the tape surface by write head after amplification. The tape moves and the recorded information comes below the Read - head. Thus, the information so recorded is immediately read out, amplified and then applied to a register known as Readout register. The contents of ROR and computer O/P register are compared with comparator circuits. If different, an alarm singnal is given and tape drive stops, otherwise recording continues.

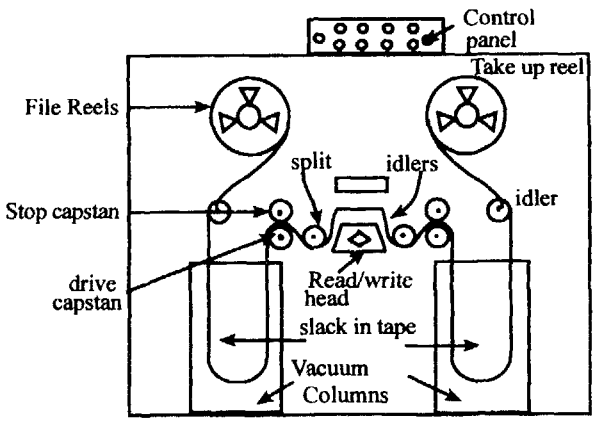



Fig. 3.14 Depicts locating magnetic tape records.

The address of the record is first read in a read OUT register (ROR) through the read head and amplifier. The desired address in address register is compared with ROR. If same, a signal is sent out to the AND gate which enables the AND gate to store the record in computer primary memory. Similarly the output from computer can be written to the tapes by write amplifiers while recording. The tape drive incorporates the start, stop, locate tape functions.

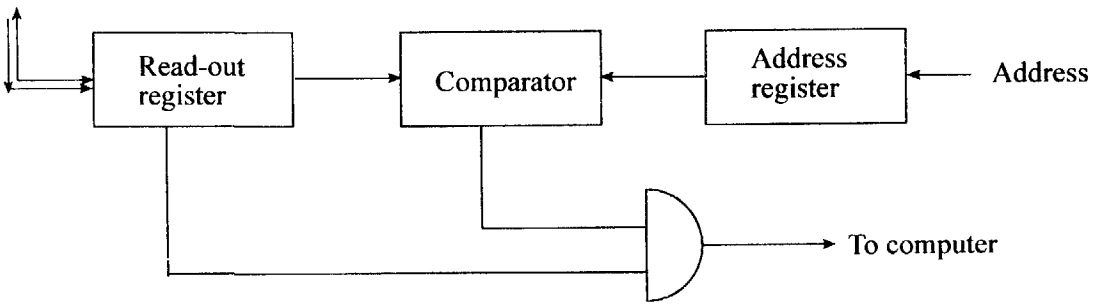



**Fig. 3.15(a)** Circuit for locating magnetic tape records

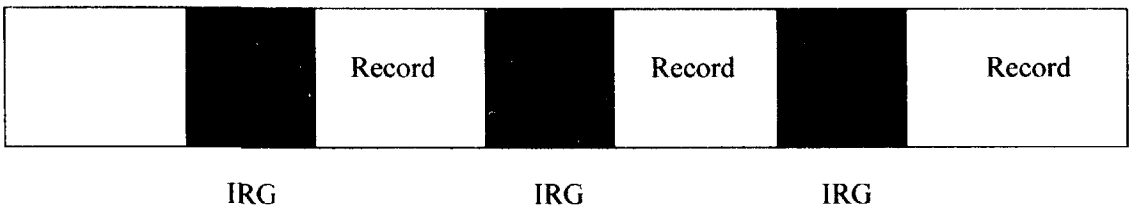



**Fig. 3.15 (b)** Record formating

The record length is variable determined by wordsize of computer. There is a uniform inter record gap (IRG). Records are grouped together into files, files separated by file gaps, as in Fig.3.15(b). Blanks are provided on tape to indicate the start and end of files.

This medium being of serial access type, is a slow device. Also these involve a considerable amount of electromechanical engineering, are somewhat inflexible and regarded as less attractive than discs. Neverthless, they serve as an essential medium for distributed data storage and off-line information for real records.

### Magnetic Drum memory

These were among the first devices to provide a relatively inexpensive means of storing information and have reasonably short access times. Fig.3.16(a) shows a magnetic drum. It consists basically of a rotating cylinder coated with a thin layer of magnetic material which possess a hystersis loop similar to that of material used in magnetic cores.

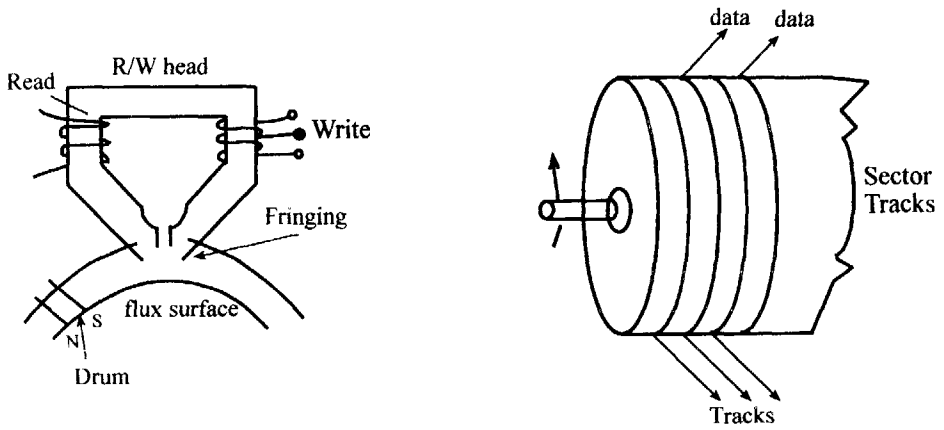



**Fig. 3.16(a)** Magnetic drum

A number of recording heads are mounted along the surface of the drum. These are used to read and write information from the surface of the drum by sensing or magnetizing small areas as shown. As the drum rotates, a small area continually passes under each of the heads. This area is known as a track. Each track is divided into number of cells each of which can store 1 bit. Generally, one of the tracks is used to provide the timing for the drum. The drum is addressed by a track number and sector number in its address field as in Flg.3.16(b).

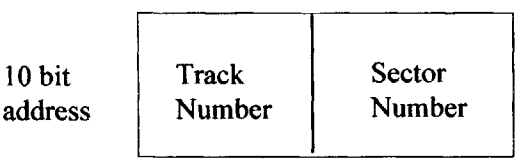



**Fig. 3.16(b)** Track number and sector number

The size of the metallic drum is typically 12" in diameter and 8" in length. The outer surface is divided into number of tracks or channels. 20 tracks per inch is a common measure. Some drums have multiple R/W heads per track for parallel data transfer. The drum is rotated at a constant speed of 1200 to 15000 rpm, average speed being 3600 rpm. The binary data in the form of magnetization can be stored in the tracks at various locations. Each track can store a large number of bits, 200bits/track inch of the circumference of a drum is a typical figure. Thus more than $10^6$ bits can be placed on a 12" dia, 8" long drum. This capacity is obtained at a low cost. The selection circuit for the R/W head is simpler because of ease of drum addressing. Considering a rotational speed of 10,000 rpm, Time for 1 revolution = $\frac{1}{10{,}000}$ x 60 sec= 6 msec. The average A acess time = 3 m sec.

Usually larger drums have slower speed of rotation while small drums have large speeds of rotation.

**Example:** A certain magnetic drum is 12 inches long and 12" in diameter. It has in total 200 tracks, with a recording density of 500 bits/inch. Calculate the drum capacity in bytes. If the speed of rotation is 3000 rpm, calculate the average access time.

The bits / track = 22 / 7 × 12 × 500 = 18, 840 bits.

$$\text{The drum capacity in bytes} = 18,840 \times \frac{200}{8} = 4,71,000 \text{ bytes}$$

A speed of 3000 rpm = 50 revolutions per second.

$$\text{Time for 1 rotation} = \frac{1}{50} \text{ sec.} = 20 \text{ ms.}$$

The average access time = 1/2 (20) = 10 ms.

Drum memory can be organized in bit serial form where all the bits of a word are in the same track; in parallel organization, tracks put together represent a word and it has a fast data rate only at the expense of more R/W heads working simultaneously. When dealing with BCD arithmetic data and eb-si-dik codes a combination of serial and parallel data representation has to be followed.

Thus, this memory provides a high data rate though it takes fairly large access times. The recording methods can be return to zero (RZ), wherein the magnetization is in two directions for 1 and 0 bits but the flux on the magnetic surface within a track always returns to the reference value between two adjacent bits of information. In *NRZ* technique, the direction of magnetization changes whenever a "1" bit value is stored and this offers higher packing density. One track is used as origin or for indexing purpose for indexed sequential files in database managements.

**Magnetic Disc Memory**

Disc stores are used as I/O for large computer systems. This has the features of cyclic access and nonvolatility. The major classifications of disks are Hard and floppy diskettes. The Personal computers use hard disks which is a fixed store unit. The RSX - 11M of DEC computers are based on Hard disk units. The Unix also utilizes the disk storage for the many terminals it supports. Exchangeable Hard disk units provide the facility for removal and replacement of the disks when a very large volume of data is thought of. With all disc systems a coating of magnetic material (ferrite) is applied to the surface of the disc platter. Writing and reading of data is performed by a recording/playback head positioned above the disc. The head is moved radially to select one of a number of concentrict racks. Each head comprises a high-permeability split-ring core wound with a coil of fine wire. Fig.3.17 shows the disc pack.

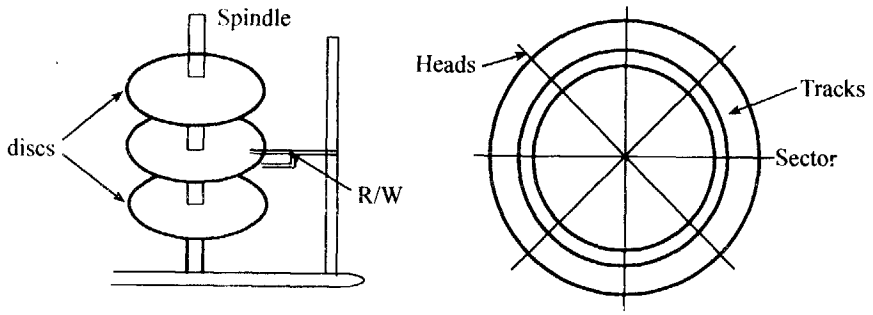



**Fig. 3.17** Disk pack

The unit may also have a number of discs arranged vertically on a spindle. The time taken for interchanging disk packs is about 1 minute with batch systems. Both sides of the disc are used for storing data. Memory cycle time is the total time to read from the memory together with the restoration time in the case of destructive readouts. Data is stored on tracks. Each surface contains about 250 tracks. Disc diameter varies between 1 to 3 feet and are rotated at a speed of about 1800 rpm. The average access time is about 50 ms to 250 ms. The outer most track serves as the index track and a timing track gives clockpulses for synchronization purpose. *NRZ* type of recording is preferable. Development of non-contacting heads with magnetic surface will improve the speed as well the life of disks. Formatting a disc is usual procedure to start using the memorywhich has a geographical layout of the medium as well the allocation space for various files of data and programs.

**Floppy Disc Memory**

Today every computer user has to take the floppy packs for his own workspace and datafiles concerning his own records. Thus this is a removable disc system similar to magnetic tape.

Capacities range from 256 k bytes to 1 Mbytes data. This is made of plastic material, 8" in diameter and is coated with ferrite of about 0.003" in thickness. It is always kept inside a plastic cover thus increasing the mechanical stability. Cleaning and lubrication is done with the cover. There is a hole at the centre through which the spindle of the drive unit rotate the disc. The 8" floppy has 73 data tracks. Each track has 26 sectors with 64 words/sector. The speed is 360 rpm. The average access time ranges from 200 ms to 500 ms. The tracks are commonly divided into sections called sectors, sector implying the minimum amount of data which can be transfered. Disks that are permanently attached to the unit assembly are called hard disks. A disk drive with removable disks is called a floppy disk more commonly used with personal computers. Now 5 1/4" and 3 1/2" miniature disks of high densities and fast access have come to stay. There is a notch provision for write protect facility on user sensitive diskettes. These play the dual role of

portability and pottability across machines. One has to be careful with the use of foreign disk memory units that they have to be inspected and checked thoroughly that they don't harm any part of the computer memories in terms of *VIRUS*. The slow I/O devices is discussed in the following topic on End input - output medium.

### 3.4 End Input/ Output Medium

Most of the noise in computer installation is due to the writing and reading actions of I/O units. Fig.3.18 shows some of the typical I/O for direct user data interaction.

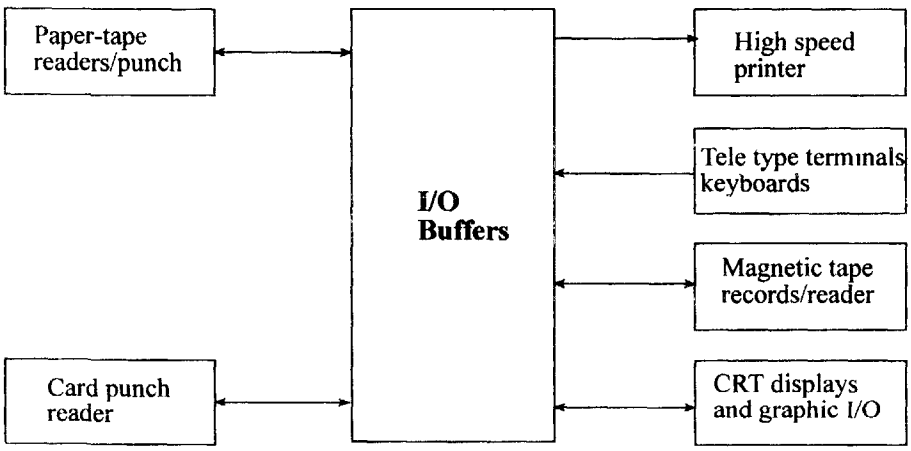



**Fig. 3.18** Typical I/O arrangement

The most widely used medium for data entry is the punched card. A hole punched on the card represents the binary value 1 and the hollerith code is used for data processing. Typical card size has dimensions of 7 3/8" long , 3 1/2" width and 0.007" thick. Each card has 80 vertical columns and there are 12 horizontal rows. Most of the program inputs used this medium and the languages like FOR tran and COBOL adopted their own card formats for usage. This still is a powerful I/O unit with mainly the batch processing machines. Each card is called a unit record. The card punch also prints the characters punched into a card on the face of the card itself. Card reading operation is shown in Fig.3.19.

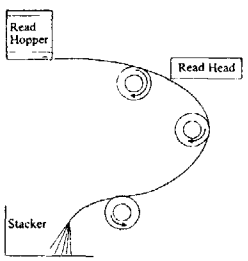



**Fig. 3.19** Card reading operation

The cards are stacked in the read hopper and are drawn from it one at a time. Each card passes under the read heads which are either brushes or photocells. The data read are stored in flipflops which form the input storage register. The cards then pass over the rollers and are placed in the stacker. There is quite often a second read head which reads the data a second time to provide a validity check on the reading process. These are quite inexpensive offering a slow speed of 100 to 1000 cards per minute. In early machines, the job control cards have to be fed alongwith the program deck when operating systems were in their infancy. Card punching units are available for a faster store of the output data from the computer system. These also serve as permanent back-up store on selective data securitysystems besides employed in real-time categories. This card device is treated more today as an OFF -line store. The ASCII and EBCDIC are the usual alphanumeric codes applied with these units.

The paper tape is continuous reel of storage like a magnetic tape which is also used as input/output data mainly in dedicated slow real time processes like numerical control of machine tools. The coding and reading becomes embedded onto such systems.

Also printers are used for outputting with varying speeds (character at a time to line at a time types) belong to the electromechanical category of I/O medium.

### Computer Printouts

The throughput of a Computer is measured by the printed output produced during frequent intervals of time period. The speed and quality of printing are essential components today, for *printers* are the *primary output* devices used to prepare documents in human-readable form (hard copy).

Impact printers use the familiar typewriter approach of hammering a typeface against paper and inked ribbon, while *non-impact printers* use thermal, electrostatic, Chemical and inkjet technologies. In what follows, different printers are discussed

### Character Printers

These print only one character at a time. Letter-quality printers use a daisy wheel as a printwheel font.

Each petal of the daisy wheel has a character embossed on it. A motor spins the wheel at a rapid rate. When the desired character spins to the correct position, a print hammer strikes it to produce the output. The speed ranges from 10 to 50 characters per second.

### Dot Matrix Printers

These print each character as a pattern of dots (9 x 7 matrix). Seven rows with nine needles in each.

The shape of each character, i.e., the dot pattern is obtained from information held electronically in the printer. The speed ranges from 40 to 250 characters per second used with micro-computer systems. The flexibility of character shaping permits many special character, different sizes of print and to print graphics such as charts and graphs. Dot matrix printers will become auxiliary devices while the laser printers are flooding - in for rough working and extending the useful lifetime of primary printers.

**Line Printers**

They are impact printers used with most medium and large computer systems for producing high volume paper output. Speed of printing varies from 300 to 2,500 lines per minute. The line length can be either 120 characters or 144 characters. Drum and chain printers are the widely used line printers.

**Inkjet Printers**

These are nonimpact character printers based on a relatively new technology. They print characters by spraying small drops of ink onto paper. Special type of ink having a high iron content is used. Droplets of ink are electrically charged after leaving a nozzle. The droplets are then guided to the proper position on the paper by electrically charged deflection plates. Some models allow for colour printing. These nonimpact printers cannot produce multiple copies of a document.

**Page Printers**

These are very high speed nonimpact printers which can produce documents at speeds of over 20,000 lines per minute. Electronics, Xerography and lasers have made these high volume systems possible. They employ electrophotographic techniques and cost very high. These are fit for real-time printing applications like desk top publishing (DTP) and professional publishers for automation. The corporate applications, DTP for design preparation and page layouts, the advertising agencies and graphic designers imaging call for CAD/CAM techniques. Personal computers promise the peripheral market today. The lasers accommodate range from 300 dpi to 600 dpi.

**LED Printers**

The LED print head consists of 2,496 led elements arranged in a line, driver ICs, lens arrays and frame. When the print head receives the image data signals, the LED elements emit light beams. These light beams are focused on the surface of the OPC ( organic photoconductive) drum through a lens array. As the drum rotates, a latent electrical image is formed on the drum surface where the led light beam strikes. Next, the toner is deposited on the exposed area of the drum. The paper gets the image from the drum. After this, the paper is passed through the fuser. In this way the toner is fused into the paper by heat and pressure and finally the printed page pops out. LED technology enables a multi-functional product which has printer, fascimile and copier functions all together with smaller size than the conventional laser printer. For real good quality of

colour reproduction thermal transfer and dye sublimation is the answer and for an occasional touch of colour with text, inkjet is an affordable solution. Printers thus discussed become a sharable resource on a multi user system and *spooling* is often one of the important device management task of the present day operating systems more of a batch processing nature.

High speed magnetic tape recorders will be much faster than a high-speed printer which is treated as an ON-line medium on well governed process management systems. Final printing is done on usual line printer which interprets the recorded tape. But today with visual mediums like the video display unit and data stored on files pagewise, the fastest *LASER printers* are used for quality as well in critical applications involving image processing and desk-top publishing areas.

A *teletypewriter* TTY is used as a remote terminal connected to a large scale general purpose computer via telephone lines. Two distinct audio frequencies visualize the binary data. An acoustic tone coupler does the translation. This TTY console consists of basic keyboard and a printer output unit. Most modern TTYs use an eight hole punched paper-tape.

Today with the growing trends in personal computers and multitasking systems the keyboard more like a *typewriter* board has a host of function keys, control characters besides software mode editors in the real realm of interactive user systems. Interactive machines involve varying sets of users. Thus, the users on interactive machines include the end person using the data base, a person managing a package use, programmers on their language shells, to system analysts and controllers restricted and dedicated with console operations for achieving interactive targets with the robot computer. The cathode-ray tube terminal has become a popular interacting medium. This is the fastest sitting strength of a computer centre involving the optical terminology. The CRT output display is driven by vertical and horizontal sweep circuits similar to that used in Television set and oscilloscopes.

Scanning is done several 1000 times/second. Thus the CRT terminal allows quick display of a file or record of informations. This device also provides for buffering to about 2K memory. The transmission of data across devices with CRT are generally represented in bauds. The pixels on the screen also provide a form of optical character recognition for standard inputs to the essential computing requirements of pattern recognition, etc.in the area of digital image processing with efficient algorithms. It is estimated that 30 to 50% of installation cost today is met in data preparation support mediums, more so essentially in dedicated computer aided *graphic* devices for varying applications like machine design, civil engg. and information technology areas. Dedicated systems for CAD applications make use of graphic workstations. Thus for an interactive domain the very fast and easy to use graphic devices have gathered momentum on specific

areas. The CGA (colour graphic display adaptor ) and extended graphics adaptor (EGAs) have shown flexibility and provides an artistic look for ease of use.

CRT is a glass enclosed tube having phosphor coated screen where the focussing of electron beam takes place for visibility. Raster scans copes up for animation and dynamic data processing. Liquid crystal displays and flat screens have low power consumption and higher capabilities. The main graphic devices used include light pens, touch screens, Joysticks, Mouse and digitizers. These graphic medium require dedicated software packages or/and software libraries in a language shell for high compute problems and colour graphics animations. The discussion of these graphic devices is left to the reader as an exercise.

The graphics applications have been approved by management professionals, design engineers, SQL - based (structured query Language ) data base systems and computer aided flexible manufacturing systems. Nevertheless the cost factor of graphic workstations are always accountable. Thus, the input output stores play a vital role with today's technological innovations. In order to convey the needs of input output devices to the CPU, two types of channels known as byte multiplexer and block multiplexer channels making use of interrupt driven mechanism and *DMA* (direct memory access) respectively are often employed in computing systems. For most of the OS shell commands which directly operate on available executable files, are either satisfied or triggered by direct lines (interfaces) for each of the terminal I/O and the Uni BUS (shared) concept is applicable only for virtual memory management systems. Having discussed on I/O units, it is important to realize that all data communicated across the system are digital (binary in nature) in the overall process. But realtime systems are analog types, involving nonelectrical or electrical signals. Thus, transducers in the context of computers essentially give rise to an electrical signal (current or voltage) which are considered as analog signals derived from the strength and features of the parameters involved. These must be converted to binary form (often referred as digital) suitable for computing processes during their stay with computers. The analog to digital converters and the digital to analog converters are considered to be the two EYES of the system. In the following the digital to analog converters is explained.

In direct ADCs (analog to digital converters), the analogue signal is directly converted into digital output without any intermediate conversion step, as in the case of successive approximation type. These are characterized by a short conversion time and are used in instrumentation systems like the data logger. The other types will be mentioned while scanning on ADC s. Many direct ADCs use DAC in the feedback path. Also digital outputs of data acquisition systems must be converted to analog voltage to actuate servomotors, potentiometers etc. A DAC produces an output voltage which is proportional to the input digital number.

Generally D/A conversion is achieved by summing weighted currents at the input of an operational amplifier.

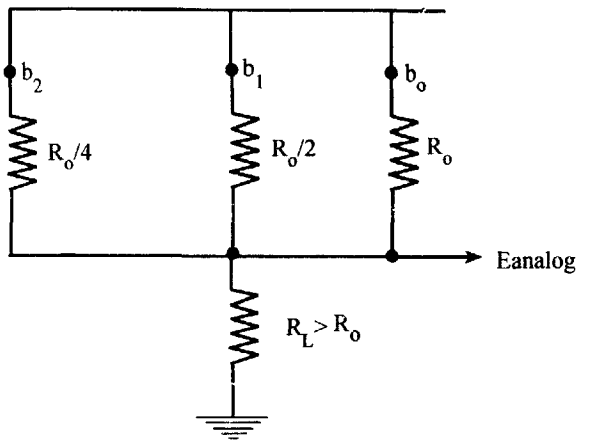



**Fig. 3.20** Weighted resistors D/A circuit

As shown in Fig.3.20 the passive resistors form the divider network. A 3-bit DAC is depicted. $R_L$ the load is large enough such that it does not load the divider network. In practical applications a 12 bit DAC means the need for a wide range of resistor values. If the most significant bit ($2^{11}$) has a resistor of $10^4$ ohms, then the least significant bit would require a value equal to $2^{11} * 10^4$ ohms.

The accuracy and stability of the DAC is dependent on the absolute accuracy of the resistances used and their ability to track each other with temperature, especially when the range of values required for good *resolution* is very large. Conversion rate is relatively slow owing to the high input impedances and speed limitations of the voltage switches.

$$\text{Eanalog} = \text{EReference} \sum_{i=0}^{n-1} \frac{a_i * 2^i}{(2^n - 1)} \quad \text{for "n" input bits.}$$

The least significant bit has a weightage of $\frac{1}{(2^n - 1)}$ .The full scale output voltage achievable is equal to E Reference when all bits are high "1".

For example, for the 3 bit D A C, the maximum O/P

$$= \text{E Reference} \frac{(2^0 + 2^1 + 2^2)}{(2^3 - 1)} = \text{E Reference}$$

Thus the precise resolution becomes $\frac{1}{2^n - 1}$ * E reference volts.

High accuracy calls for very accurate resistor values which means high cost. A well controlled accurate reference power supply able to source/sink considerable currents is again needed. Offset in amplifiers must be quite small and response time is mainly determined by the slew rate of the summing amplifier. The switches shall be required to have a low ON resistance. More often, a D A C is an integral part of analog to digital conversion methods.

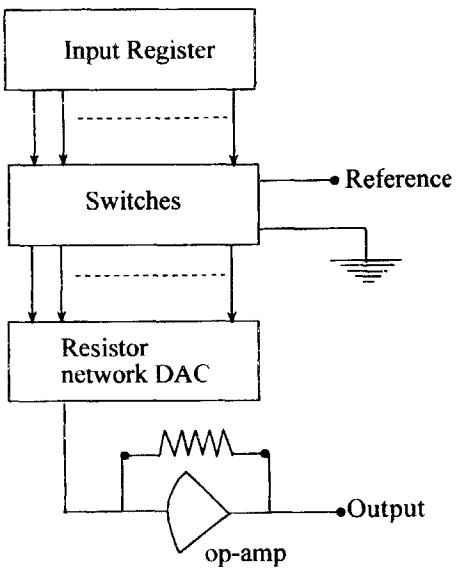



**Fig. 3.21** DAC Block Diagram

Fig. 3.22 gives the circuit for a practical 4 bit D to A binary ladder constructed using only two resistance values R and 2R. Here the output voltage is a properly weighted sum of the input bits.

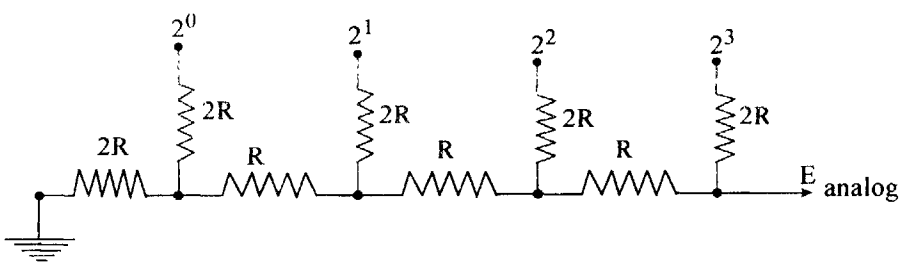



**Fig. 3.22** 4-bit binary ladder

From theVenin's theorem, the total resistance looking from any node back toward the terminating resistor or out toward the digital input is 2R. This is true regardless of whether the digital inputs are at ground or E Reference, because the internal impedance of an ideal voltage source is 0 ohm.

The contribution from the different bits are 1/2 for the most significant bit (MSB),1/ $2^2$ = 1/4 for the second MSB, . . . . . so on and $1/2^n$ for the $n^{th}$ bit, i.e., the LSB (least significant bit). Thus the minimum measurable signal is $1/2^n$ times E Reference for n input bits on a practical DAC. The maximum measurement value is (1/2, 1/4, 1/8, 1/16)

$$\frac{2^{n-1}}{2^n} * E_{Reference}, \quad \frac{8+4+2+1}{16} = \frac{15}{16} * E_{Reference}$$

where n is the number of bits reflecting the resolution. So we can say better the resolution, accurate the calibration. A high performance linear operational amplifier can be employed to tap good analog outputs. Fig.3.23 gives the practical circuit of a 4 bit DAC.

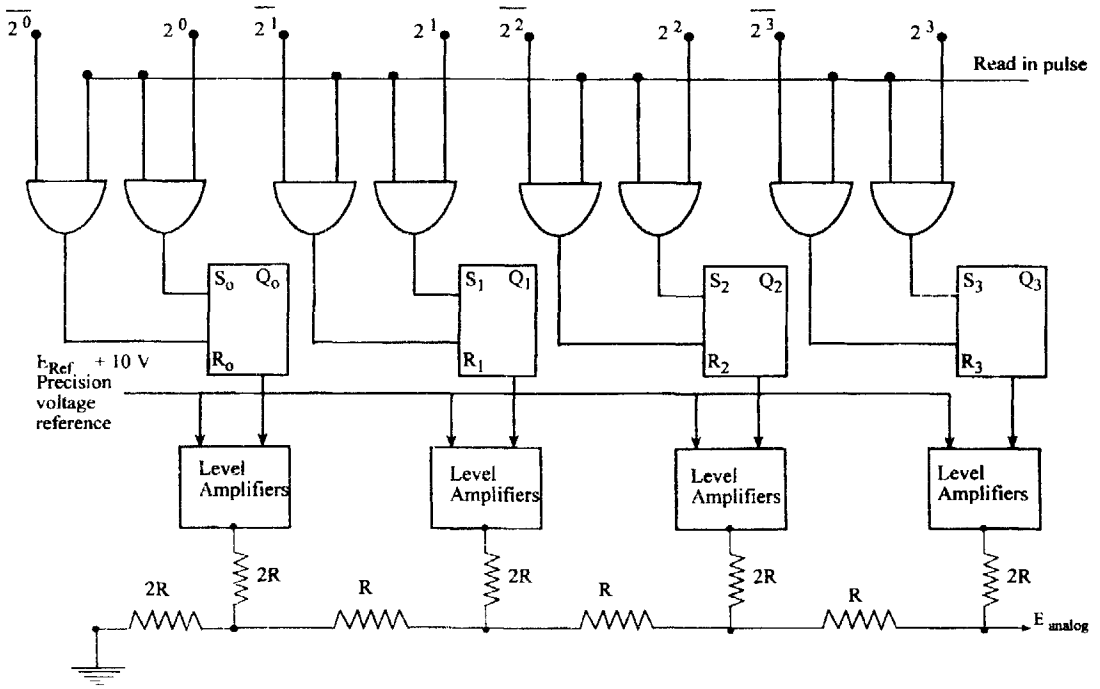



**Fig. 3.23** Circuit of 4 bit DAC

The level amplifiers work in such a way that when the flipflop status is High, the amplifier output is EReference = +10 volts, otherwise 0 volt. The R value can be conveniently chosen between 5KΩ to 50 KΩ which lends to manufacture by thin-film deposition techniques that improve the temperature tracking capabilities; thin film nickel-chromium R-2R ladder networks, featuring tracking with temperature to better than 1 part per million/°C are commercially available.

The D A C is used at the output of the computer and if the application pertains to a process control system involving many parameters, the signal output must be held between sampling periods using sample and hold amplifiers as in Fig.3.24.

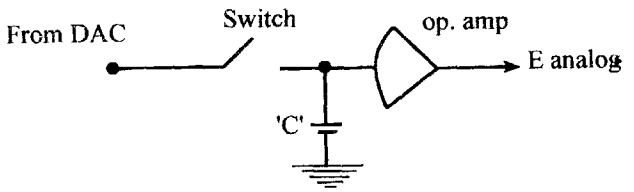



**Fig. 3.24** Sample and hold amplifier

The sampling theorem is used in practice to determine minimum sampling speeds. If the sampling rate in a system exceeds twice the maximum signal frequency, the original signal can be reconstructed at the destination with vanishingly small distortion without any sampling errors.

As in figure 3.24, when the switch is closed, the capacitor charges to the DAC output voltage. When the switch is opened, the capacitor holds the voltage level until the next sampling period. The operational amplifier provides a large input impedance so as not to discharge the capacitor appreciably and at the same time offers gain to drive external circuits.

Linearity or relative accuracy is the maximum deviation of the output compared with the desired output. The speed of DAC refers to the time necessary for analog output to settle down within desired accuracy. Resolution defines the smallest increments in signals that can be discerned.

Fig. 3.25 gives DAC with a current output. Binary - Weighted currents are generated using active sources and the output is obtained by summing them on a common output bus.

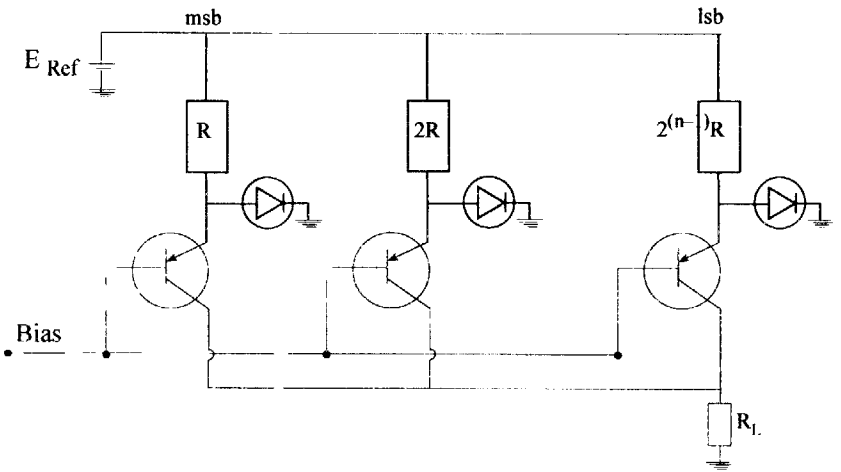



**Fig. 3.25** DAC with a current output

The errors involved in this type of conversion usually limits its accuracy to about 12 bits. A full scale output current of 10 mA is ideal, since a 1 volt output would develop across a load resistance of 100Ω, an approximate impedance level for high-speed coaxial cable operation. Multiplier type DAC is one which is specially designed to accept a varying reference voltage.

The output magnitude is directly proportional to the product of the reference voltage and the digital input. The reference can be a bipolar or AC source and its performance depends mainly on the type of electronic switches used. For example, FET switches are most suitable for the bipolar reference mode.

M D A Cs are extensively used in programmable test equipments where the amplitude of a varying or DC signal can be controlled by a digital computer. They are also used for graphic displays since they can perform direct multiplications such as R sinθ which is required for conversion of polar to rectangular coordinates in microseconds.

DACs in Integrated form are used in peak detectors, voltmeters, stepper motor drive and programmable systems.

## Analog to digital conversion

In indirect ADCs, an analog signal is converted into an intermediate form which is driven to digital output. The dual slope converter is one of integrating type.

The analog input is converted to a proportional time interval known as voltage to time conversion. T'hen by counting, the digital output is achieved which offer inherently high accuracy with simple circuitry. These are slower and very commonly used in digital instruments like digital voltmeters.

Direct ADCs are characterized by a short conversion time finding applications in real-time instrumentation and process controls. The various methods of direct ADCs include simultaneous or parallel conversion, counter ramp type, successive approximation method, servo and delta modulation ADC. The process of approximating the input value is called *quantizing.*

In analog to digital conversions, resolution is quantified as the change in analog input voltage for obtaining a 1 bit (1sb) change in output.

Resolution = $\frac{E}{2^n - 1}$ where E is maximum swing in input and "n" is the number of output bits Drift refers to the change in the output for the circuit parameters with time and temperature. Very low drift is achieved at high cost and complexity. The speed of A/D converter is measured by the time to perform one conversion or the time between successive conversions at the highest possible rate. Some of the advantages achieved for digital display of measured data in instrumentation systems are:- allows practically unlimited precision; offers very slow ageing rate; noise effect on system accuracy can be reduced considerably.

Analog techniques are preferable in certain applications wherein a peak or null has to be obtained (setpoints) and small variations in certain parameters are to be recorded or displayed continuously (for example, the speedometer of a moving vehicle on road). The simultaneous ADC is shown in Fig. 3.26.

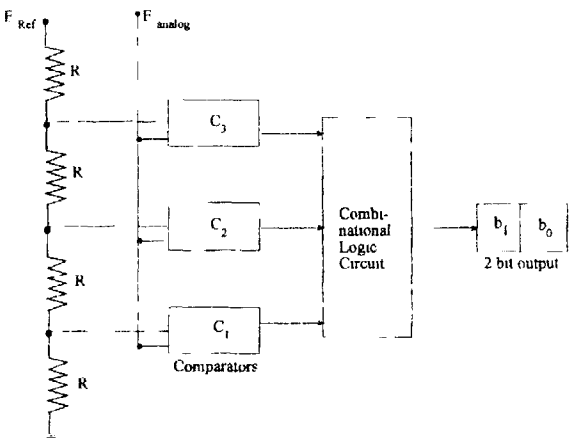



**Fig. 3.26** Simultaneous ADC

This is the fastest method. The reference voltage to each comparator is fed using a precise voltage source and potential divider network. The input Eanalog is fed to another input line of the op. amp. comparators. Thus the circuit shown provides only a 2 bit output with 3 comparators. In general, a n bit output requires $(2^n - 1)$ comparator units which is costly but finds application in supercomputers, real-time ON-line adaptive control systems, space research and simulation domains.

Nevertheless the more general types of ADC are the counter-ramp and simultaneous approximation procedures. Fig. 3.27(a) gives the counter type of analog to digital converter.

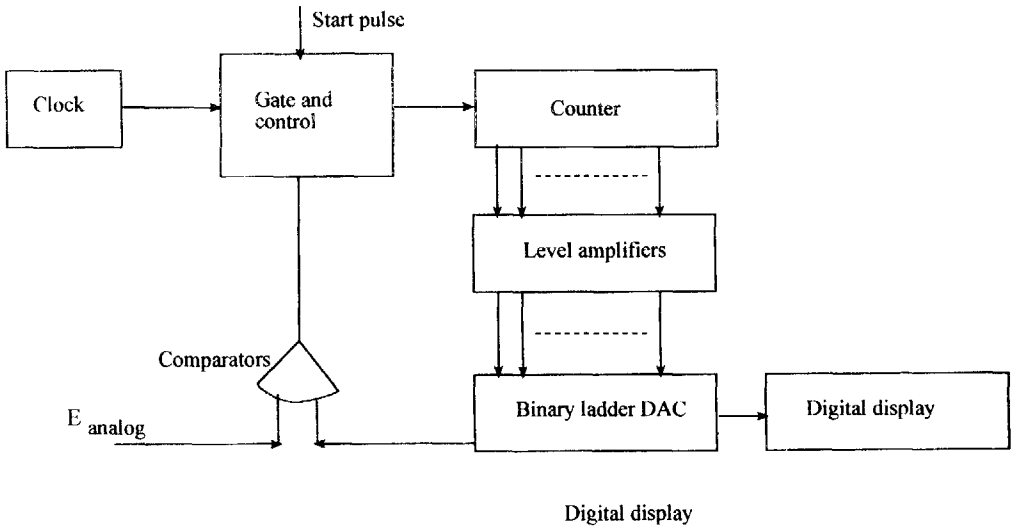



**Fig. 3.27 (a)** Counter-ramp type A to D

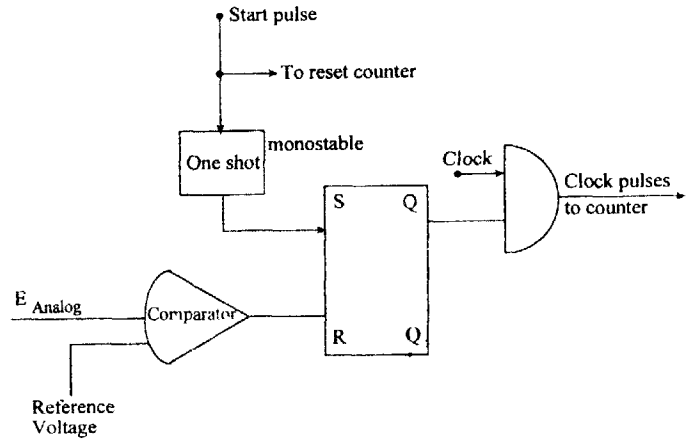



**Fig. 3.27 (b)** Control logic of Fig. 3.27 (a)

The variable reference is achieved by a counter and DAC arrangement which is continuously scanned by the comparator with E analog and the output is displayed when the conversion is over on digits. The control logic is shown in Fig.3.27 (b) which employs a flipflop and a monostable control to stop the counter at the appropriate instant. The speed of ADC is governed by the clock period and also the *counter* size. Continuous A/D requires the use of an up/down counter and additional control.

**Example**: Calculate the assured maximum conversion rate of the ADC if the output is 8 bits and the clock frequency is 500 KHZ. The number of counts = $2^8 = 256$.

$$\text{Time for 1 count} = \frac{1}{500 \times 10^3} = 2\mu S.$$

Time for 256 counts = 512 μS.

$$\text{The average conversion time} = \frac{512}{2} = 256\ \mu S.$$

$$\text{Assured maximum Speed} = \frac{1}{.512 * 10^{-3}}$$

= 1953 conversions/second.

**Successive approximation method**

This involves the process of approximating the analog voltage by trying one bit at a time beginning with the most significant bit. This is a relatively optimum approach giving fast conversions with critical design of circuits. Also more of the practical ADC chips employ the successive approximation type especially with microprocessor based systems. Speeds as high as 100 ns/bit can be achieved. The accuracy depends on the stability of the E reference, the level converters, the D to A network and the comparator. Fig. 3.28 gives the block schematic of this method of ADC.

To start with, a "start conversion pulse" clears the output register and turns ON its msb so that the D A C gives half of a full-scale deflection. This output is compared with the analogue signal, and if E analog < DAC reading, comparator turns off the msb of the output register via the control logic unit. And if E analog > DAC, the msb in output register is left ON. The next clock pulse turns ON the next significant bit in the output register via the shift register. The process is repeated n times for an bit converter until the 1sb has been compared. Then the clock stops, to be restarted only by the next "start conversion pulse".

The"status" line changes state to indicate the contents of the output register now constitute a valid conversion. The analog input must not change during the conversion

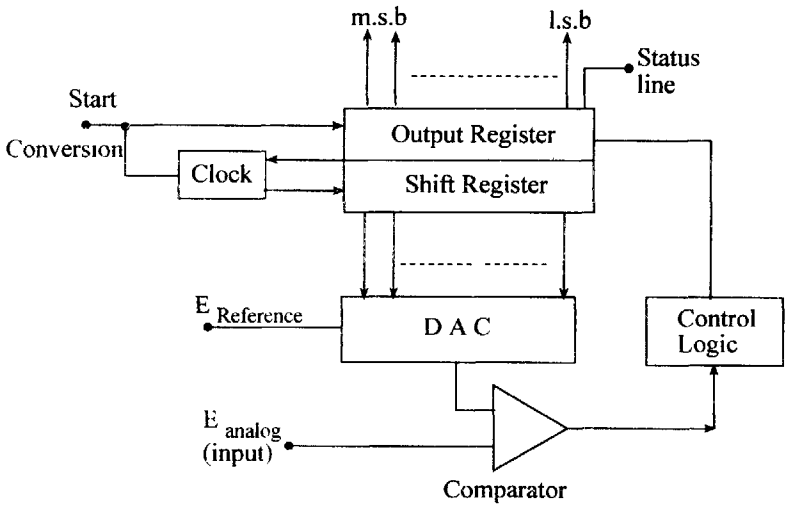



**Fig.3.28** Block schematic of ADC.

interval and if it does, a sample-Hold device is used to hold the input signal value constant during the conversion interval and the "status" signal is used to release the"HOLD" mode to again sample at the end of conversion. Assuming a 10 bit converter and a 1 MHz clock, the conversion time for 1 sample = $10 \times 10^{6}$ S = 10 μS.

Differential linearity is a measure of A to D converter performance which reflects the variation in voltage-step size which causes the converter to change from one state to the next and is usually expressed as a percent of the average step size.

The counter and continuous type ADCs possess better differential linearity than successive approximation type. Up-down counters are employed in continuous conversion systems offering better speeds. In isolated analog control instrumentation, the indirect ADCs using voltage to time conversion technique are often used for real-time robots.

In pulse and digital circuitry, the high impurity p–n junction devices like tunnel diodes made of germanium or gallium arsenide are used. Ga As provides a high $I_p/I_v$ ratio (Peak to valley current). For computers, devices with $I_p$ in the range of 1 to 100 mA are common. *Tunnel diode* offers high speed, low noise, low power and environmental immunity.

### Opto electronic devices

In a photodiode, reverse-biased p-n junction is illuminated and the current varies linearly with the light flux. It is the fastest photo detector coping with nS switching rates.

The reverse current is in the order of μA for photodiodes. Photo transistors are applicable for real computing where the switching time is of the order of μS and the output current mA range.

Solar cell is unbiased because it is used as an energy source. It is physically larger and is more heavily doped than a photodiode.

### L E D

The energy released when an electron falls from the conduction into the valence band appears in the form of infrared radiation. The o/p light power increases with injected current input. *Light emitting diodes* give off light when forward - biased. GaAs for infrared radiation, GaAsP for red or yellow light and GaP for red or green light is the nomenclature of usage. Red has a wave frequency of 4 ($10^{14}$) Hz, green 6($10^{14}$) Hz whereas blue has 8($10^{14}$)Hz.

Photon energy E = h f,

where h= planck's constant = ($6.63 * 10^{-34}$)

E = energy in joules;

f = frequency in Hz.

Colour of radiation depends on frequency and brightness on number of photons received/second.

LEDs are used in display systems and they do (led arrays) supply input power to LASERs or entering information into optical memories. Lasers are seldom used in advanced colour televisions and holograms for 3-D imaging.

### 3.5 Futuristic Memory Techniques

In the use of electronic memory the octal buffers/ drivers with tristate outputs improve the performance figures of serial transceivers also increasing the density of printed circuit board in interfacing area. Quadraple bus transceivers for asynchronous two way communication between data buses with propagation delay just of the order of 14 to 24 ns is already available in market with full TTL and CMOS compatibility. CMOS ROMs from 16K to 64K bytes organized in 8 banks already assist in memory interleaving on time shared machine with access times of 350 to 450 nS. FAST logics have become important in microprocessor systems. The fabrication components to support the devices area is detailed below: FAST is a trademark of Fairchild Camera and instrument Corporation.

The small geometries associated with oxide-isolation makes complex functions possible with very high-speed and low power characteristics. The speed of devices are fixed primarily by the node capacitance of the device junctions. Using the smallest

practical resistor size creates the current schottky family of high-speed logic. Input loading and short-circuit protection are the major difficulties in implementing high-speed circuits. Thus mainly in critical areas like super -computing, schottky TTL has found an impact besides the usual emitter-coupled logic. They operate on $F_T$, upto a few GHZ.

The importance of using high-speed bipolar circuits in conjunction with relatively slow MOS memories has called for SEC-DED codes Hardware.

They generate 6 - bit and 7 - bit checkwords from 16 or 32 bit words. Each check word is stored along with a corresponding dataword, during the memory write cycle. During the memory READ cycle, the 22 or 39 bit word retrieved is analysed and SEC-DED facility is achieved as well the gross-error condition (all 1's or 0's) of unidirectional errors is detected.

The input capacitance of 256 K NMOS DRAM is of the order of 8 pF. The maximum size of the memory that can be driven by one controller is set by the maximum output capacitance that can be safely driven by any output - 256pF. This dictates the effective fanout of the controller to 32 and the maximum memory size to 1 M byte. The oxide - isolated fast TTL has addressed octal functions for buffers & transceivers, error control and dualbus data transfers in high- performance microprocessors.

Multiport memories are also called shared memories by more processors. The most common configuration is the dual-port memory with 2 processors sharing the same array.

Applications range from simple message passing between processors in a loosely - coupled environment to multiple processors executing from a common program memory in tightly - coupled systems. (e.g.with C R T controllers of shared memory). The display memory must be accessed by the controller to refresh the display and by the microprocessor to update and manipulate data on the screen.

Static RAMs are preferred over dynamic RAMs because the latter require refresh circuitry which adds another port to the array, and also decrease maximum throughput rates in fast systems. The dynamic RAMs offer reduced power consumption and large storage capacity in a single memory chip. Optical memories employ a 64 kilobytes dynamic RAM for image processing which posesses enough resolution to be applied to robots (ultrasonic vision upto 10 metres). Dynamic microprogramming calls for occasional writable control stores. More often multi-port memories are small in Size (density). The multiport memories include the control logic for the status of use to the users, may be, besides providing priority.

A micron Eye camera employing an *OPTIC RAM* (64 kilobits of dynamic RAM) has been developed by micron technology. It contains a printed circuit board with suitable

interfacing circuitry so that it can be interfaced to a microcomputer. With this type of sensors, bit pattern of images are stored in the computer memory. It has enough resolution to be applied to robots, as in ultrasonic vision accommodating upto 10 meters with a resolution of 3mm.

## Optical Systems

The optical system where the input and output are essentially 2-D spatial patterns are immune to electromagnetic interference, a desirable feature in optical interconnections. Liquid crystal light Valve (*LCLV*) act as optical storage cells equivalent to electronic primary memory; whereas high density secondary memory being implemented by optical disks and holographic stores.

**Table 3.4** Summarizes some factors of optical storage systems.

| *Type* | *Size (cm)* | *capacity (m byte)* | *Transfer rate (mb/S)* | *Access time (ms.)* | *Usage* |
|---|---|---|---|---|---|
| Philips Videodisk | 30 | 2500 | 5-10 | 100-500 | Write once |
| CD- ROM | 12 | 540 | 0.15 | 50-1500 | Read Only |
| Toshiba magnetooptic | 13.3 | 268 | 2.62 | 100 | Read Write |

Optical reading of the magnetically stored data is based on the Faraday effect. Optical writing of magnetic information involves thermomagnetic effects. Erasable magnetooptic disks are now at advanced stages of development. In holographic memory for read only on-line data processing, the hologram is reconstructed for electronic processing. From literature, the optical memories are susceptible to only unidirectional errors which gives an edge of reliable computing over the electronic counterparts. It is fairly easier to construct codes for information reliability for this kind of errors with lesser redundancy. Fiberoptics are already in use in Local area networks (to connect computers to peripherals) providing a noise-free communication. Optical computers promise to the area of SIMD machine of processing 2-D images, biomedical image processing being a typical candidate. It can be hoped in future, the new technology of optical computing may meet the pattern recognition and neural computing ends in a narrow origin.

The performance figures of any type of memory can be realized only by a detailed study of how much storage and how long the same is used in discrete or continuous modes. This also leads to problems in virtual memory management for good process designs. Memory interleaving in Hardware amounts to account for *reliability* in systems which can be further brushed up by using error control mechanisms..

Magetic bubbles and charge-coupled devices promise to be the stores for bit slice architectures. The holograph memories serve as data storage for bulk data with light

technology in Image processing. Already the optical memories serve well as offline banks for pattern recognition and data communications. Optical memories are more susceptible to *unidirectional* class of errors better suited for reliable computing area.

### 3.6 Memory based Architectures

The stack addressing machines have very little complexity for access coding and applied in a narrow range of problems like single language domain, process control of machine tools, etc with reprogrammable features. The various methods of main memory access will decide the processor architectures. The memory organisations of interleaving and bit slice process lend themselves for sensitive data processing domains. The paging and memory segmentation of virtual systems attract many users converging on MIMD and SIMD benches creating the cohesion and security problems. On-line productive environments altogether call for graphics and device usage in all engineering integrated manufacturing. Beyond all, the super-computers have to address still the associative memory concepts, cache hits, multibus protocols. *dataflow* and pipelining. In conclusion, the memory must be construed not as empty set but as an intelligent informative tool for real file maintenance in a broad base of applications.

## KEYWORDS

Algorithm, microcontroller; Electron, electron-volt. semiconductor, Doping, cut-in voltage, Kirchhoff; Register, byte, Word organisation, 3-D organisation. Cryogenic, throughput, random access, Astable, monostable, flipflop. flag, decoder, random access memory, memory-mapped I/O, dynamic memory, RISC, EPROM, Cache, Hit ratio, Perform, Merge, PLA, arithmetic; Turn-around-time, EBCDIC, NRZ, VIRUS; printers, primary output, non-impact printers, spooling, Laser printer, teletypewriter, graphic, DMA, resolution, Quantization, Counter, tunnel diode, light emitting diode, Optic RAM, LCLV, Reliability, Unidirectional, dataflow.

## PROBLEMS

1. Define wordlength of central processor.
2. Correlate the machine language to the constituents of a CPU.
3. What is stored program execution?
4. Quantify an instruction cycle.
5. Why should unconditional jump instructions be avoided in preparing program inputs?
6. Mention the various types of semiconductor memory technologies and bring out the characteristics of each.
7. What are the major points taken into consideration while selecting the CMOS chips?
8. Define the terms Fan-out, and noise immunity of a logic gate. What is emitter coupled logic?
9. Sketch the organization of registers in Intel 8085. What is the advantage of pairing registers?
10. Compare register and stack addressing modes. Distinguish between clock, machine and instruction cycles. With timing diagram explain the fetch cycle for 8085.
11. Define THRUPUT of yourself as a programmer.
12. What is the significance of E format in FORtran language for floating point arithmetic in system optimizations?
13. How do Cache memories help in improving program runtimes?
14. How many address pins are required to access 4K memory locations?
15. How do the flags help a microprocessor for program data flow and for self-testing towards reflecting the abilities of a computer?
16. Mention the merits of hexadecimal code in minimal micropocessor sytems.
17. How is a programmable I/O port superior to a simple 8212 I/O latch?
18. How a processor status word differ from the program status word on a UNIX bench?
19. Mention the merits of crystal oscillator usage for microprocessor clocks.
20. Explain the purpose of various segment registers of 8086 microprocessor.
21. (a) When is the need for dynamic electronic memories ?

    (b) Give example of anyone staticRAM chip and explain its constructional details and usage.

22. Explain anyone procedure for internal sorting.
23. Comment on the register storage requirement of floating point arithmetic activity.
24. Attach a note on D flipflops for buffering.
25. When are EPROM s desirable in micro computer development systems?
26. A digital computer has a memory unit at 8192 words, 36 bits/word. The instruction code format consists of 5 bits for operation code and 13 bits for address part. Two instructions are packed in one memory word and a 36 bit instruction register is available in control unit. Formulate the FETCH and EXECUTE cycles for the computer.
27. What are the different types of mapping procedures in the organisation of Cache memory? Explain in detail.
28. Mention the speciality of stack segments at execute time and correlate the same with Cache usage.
29. Correlate the terms accuracy and range of numbers in the exponent mantissa format for enhancing the architectural capabilities of machines.
30. Express the effective address for the following schemes. Direct, Indirect and pre-indexed indirect.
31. Give the differences between virtual memory and Cache memory and discuss the implementation of associative memory.
32. Explain DRAM refreshing. How is refreshing done?
33. With a neat sketch explain the basic working of a memory chip.
34. Compare the features and characteristics of PROM and EPROM.
35. What is absolute addressing? Write an exhaustive note on immediate addressing mode.
36. The access time of a Cache is 10 ns and that of main memory is 1000 ns. It is estimated that 80% of the memory requests are for read and the remaining 20% for write. The hit ratio for read accesses only is 0.9. A write-through procedure is used. (i) What is the average access time of the system considering only memory read cycles? (ii) What is the average access time of the system for both read and write requests? (iii) What is the hit ratio taking into consideration the write cycles?
37. Explain the techniques used to avoid racing in sequential circuits.
38. Define the roles played by the program counter and flag status registers of a central processing unit.

39. A 2-D RAM has N addressable locations. Show that the number of address drivers is minimum if and only if the memory is Organized as a square matrix.

40. The address of a terminal connected to a data communication processor consists of two letters of the alphabet or a letter followed by one of the 10 numerals. How many different addresses can be formulated.

41. How many 128 x 8 RAM chips are needed to provide a memory capacity of 2048 bytes? How many lines of the address bus must be used to access 4096 bytes of memory?

42. Differentiate bit and byte organised memories for systems reliability.

43. Write clear notes on the capabilities of JK flipflops in memory design.

44. What is cache miss rate and how is it helpful on intelligent search applications.

45. Write a note on resolution of monitors as I/O support.

46. Describe any one output device suitable for use with a mainframe computer.

47. Describe any one memory management scheme with suitable block diagram for a modern computer system.

48. Mention the applications of clippers and clampers in pulse control circuits.

49. A ROM has the following timing parameters:

    Maximum address to valid data output delay = 30 nsec.
    Maximum chip select to valid data output delay = 20 nsec.
    Minimum Data Hold time
    (After address change or after chip deselect) = 10 nsec.

    Assume that, the chip is selected using one of the address lines, and, the data set up time is negligible.

    What is the maximum rate at which a CPU can continuously read data from this ROM? (show your calculations step-by-step).

50. In a two-level virtual memory, the memory access time for main memory $t_{A1} = 10^{-8}$ sec, and the memory access time for the secondary memory, $t_{A2} = 10^{-3}$ sec. What must be the hit ratio, H such that the access efficiency is within 80 percent of its maximum value?

51. (a) For the synchronous counter shown in figure below, write the truth table of $Q_0$, $Q_1$ and $Q_2$ after each pulse, starting from $Q_0 = Q_1 = Q_2 = 0$, and determine the counting sequence and also the modulus of the counter.

    (b) Design a 3-bit counter using D-flipflops such that not more than one flipflop changes state between any two consecutive states.

52. A microprocessor is capable of addressing 1 megabyte of memory with a 20-bit address bus. The system to be designed requires 256 K bytes of RAM, 256 K bytes of EPROM, 16 I/O devices (memory mapped I/O) and 1 K byte of EERAM (electrically erasable RAM).

    (a) Design a memory map (to reduce decoding logic) and show the decoding logic if the components available are:-

    | Type | Size | Speed |
    |---|---|---|
    | RAM | 64 K x 8 | 140 n sec |
    | EPROM | 256K x 8 | 150 n sec |
    | EERAM | 256 x 8 | 500 n sec - read, 3m sec -write. |

    (b) The microprocessor is operating at 12.5 MHZ and provides time equivalent to two clock cycles for memory read and write. Assuming control signals similar to 8085, design the extra logic required for interfacing EERAM.

53. When is it desirable to store programs and data in separate memories on a time-shared multiuser system?

54. Discuss how the speed compatibility between a central processor and a memory device is ascertained.

55. Explain the need for auxiliary memory devices. How are they different from main memory and from other peripheral devices?

56. A certain moving arm disk - storage device has the following specifications: number of tracks per surface = 404, track storage capacity = 13030 bytes, disk speed = 3600 rpm, Average seek time = 30 m secs. Estimate the average latency, the disk storage capacity and the data transfer rate.

57. Explain the terms seek time and latency with serial access stores.

58. Compare Assemblers with compilers towards speedup in generating. obj code.

59. Elaborate the distinct differences of an encoder from a multiplexer.

60. What are the uses of a demultiplexer?

61. What is binary search of an organised file and what is the time complexity for retrieval?

62. Write a note on the usefulness of user stacks to good operating systems.

63. What is racing in sequential processes and exhaust on the methods to arrest the same.

64. Can a disk be used with bad sectors? If so, how and how long? If not, how is the disk replacement carried out?

65. Mention some essential and desirable hints for pottability and portability respectively of floppy disks.

66. Why is databus bidirectional?
67. Distinguish a floating BUS from a fetched BUS for a UNIBUS architecture.
68. Mention few applications of priority encoder in interrupt driven I/O for communications.
69. What is the purpose of monitor ROM with reference to minimal I/O in microprocessor systems?
70. Give a clear Cost/Trade analysis supporting packages in the ASICs area for endusers.
71. Bring out the superior meaures for light emitting diodes in display systems compared to the seven-segment displays.
72. What is lexicographic sort and how are alphanumerics compared in a character set?
73. Mention a few distinct advantage of a CPU incorporating the serial communication facility in addition to the usual bus strategy with I/O systems architecture.
74. Assuming a 8 bit DAC, compute the resolution of a binary ladder with + 10 volts treated as "HIGH" and 0 volt as "LOW" logic levels.
75. Explain crisply the easeness of 'C' language for graphic displays. Also remark how colour can act as a third dimension for computer vision?
76. What do you understand by I/O processor?
77. Bring out the differences between isolated I/O and memory - mapped I/O.
78. Describe the analog to digital conversion technique employed in a digital voltmeter.
79. Compare the two basic types of D/A converters using resistors.
80. An ADC which will encode pressure data is required. The input signal is 666.6 mV/psi. If a resolution of 0.5 psi is required, find the number of bits in the A to D converter. The reference is 10 volts. Also find the maximum measurable pressure.
81. What are the disadvantages in software simulation of analog to digital conversions for analog input readings?
82. Explain the operation of a schmitt trigger circuit for pulsing activity in digital systems. Define duty cycle of astable multivibrator.
83. How are analog to digital conversion methods evaluated? Explain anyone ADC technique.
84. Write a note on multiplier type DACs.
85. How are utility programs different from systems programs and also how they influence the software objectives?

86. Write a well detailed note on modularization criterion for structured programming methodology.
87. Explain the direct memory access for distributed processing in loosely coupled networks. Also comment on CMOS suitability for high speed computers.
88. State a few specific examples involving arithmetic inviting the use of loop-up tables.
89. Explain the operations on queue and stack data structures.
90. What is a DEqueue structure and contrast the same with exchange instruction.
91. What are the advantages delivered by using bit/memory chip organisation?
92. Clearly distinguish memory-mapped I/O from I/O - mapped I/O architectures.
93. What are the desirable features in addressability for a multiuser system with identical I/O units? Also highlight on the interrupt driven transfers for this interactive environment.
94. What is the purpose of password concept on multiprogrammer domains? Highlight the same with the idea of secure systems.
95. Bring out the need for subroutines and Macros.
96. How are function subprograms decided to be included in computing systems? How are they consumed on a multiuser system?
97. If a system has two stack pointers, comment on the use of one stack exclusively for the user as a bonus in computer organisations.
98. Why computer architects provide upward compatibility in their products? What is simulation in the realm of revolving architectures?
99. Write notes on (a) arithmetic pipelining in the case of scarcity of compute elements as in RISC machines and (b) for representing numeric data with CPU in-storage scarcity.
100. Comment on instruction complexity biased with respect to instruction lengths.
101. Mention the speciality of JK flipflops in up/down counting for CPU delays in reconfigurable systems as a test measure.
102. Write clear notes on arithmetic algorithms for computations beyond the capability of a machine. Highlight ( spell out) your approach in simple words with specific examples.
103. Mention the objectives of VLSI testing for digital semiconductor memories.
104. Why should there be separate address and data buses? Mention their merits on memory shared architectures.

105. What was the basic feature of the UNIX operating system from the eyes of Language designers?

106. What is the need for computing - algorithms?

107. What is segmentation? Compare it with paging. What is page-fault?

108. What do you mean by stack - based machine architecture?

109. Explain the technique of designing parallel memories for loosely and tightly coupled systems.

110. What are the databases and data structures used in each phase of compilation?

111. Write notes on storage allocation in FORTRAN, with special reference to COMMON and EQUIVALENCE declarations.

112. Write a note on Amdahl effect.

113. Explain anyone DOS and UNIX shell commands as an on-line user.

114. Write a detailed note on CMOS - TTL interfacing for digital systems.

115. Compare the activities following an hardware interrupt from that of a software interruption.

116. What are the specific advantages of memory segmentation along with cache memory in application domains?

117. A queue Q containing n items and an empty stack S are given. It is required to transfer all the items from the queue to the stack, so that the items at the front of the queue is on the top of the stack, and the order of all the other items is preserved. Show how this can be done in O ( n) time using only a constant amount of additional storage. Note that the only operations which can be performed on the queue and stack are Delete, Insert, push and pop. Do not assume any implementation of the queue or stack.

118. Clearly outline the structural organisation towards memory optimization in a distributed data processing COBOL interactive environment towards computerization.

119. Write clear notes on photodiode and Light emitting diodes.

# CHAPTER 4

# SINGLE CPU MACHINES AND PIPELINING

A program comprises of data and instructions that must be presented to the machine by means of a language. If the language happens to be command - based, then the system just executes the executable files already permanently residing on the system. Otherwise the critical aspects of compilation, interpretation, Assembling become the main task of the CPU in order to generate the .Obj files which can be linked further for execution on the machine. Strictly the above sequential organisation is mandatory for stored programs as well as programs being stored. Thus, the stored program concept is obeyed perfectly on a powerful batch processing machine. Whereas on an interactive environment where users spend their time with the machine on-line, have to be more alert on the knowledge of both command and communication aspects of programming languages. On a time shared system, the interactions of user activities are still further monitored by the active operating system. Thus, it becomes clear that a CPU has to cater to the varying categories of system configuration and more likely that the CPUs are identified for one type of workbench. With the computer networks coming up in future, the total system may have to take care of all types of workstations which calls for real implementation of parallel processing character.

*Pipelining* is an inherent hereditary character of CPUs for optimizing the use of valuable resources in terms of processing, storage and device components. In this context the transition from 8 to 16 bit processors can be considered a revolution in evolving CPUs.

## 4.1 PIPELINES AT EXECUTE TIME

In this section the CPU activity shall be analysed at system level. Each program consists of a set of instructions ( .exe files) to be executed for reaching the solution outputs. The CPU contains a high strength of arithmetic and logic elements with adequate storage facility. So, first and foremost, the instruction pipelining in terms of data preparation is a continuous activity with single CPU machines. Consider the fig.4-1 which depicts the flow of an *instruction cycle*. This involves both the external and internal activities in terms of data fetching. Even in a simple sequential machine the blocks (2) and (3) can go in parallel, besides allowing (1) to happen for instruction lookahead. In fact, the CPU 8086 maintains a 6 byte queue inside the microprocessor. In essence, the instructions sensitized fit best at any point of time to be simultaneously attended to. The machine cycle encoding reflects the micro-operations which have to be carried out in parallel to

denote an instruction time. Microprogramming is an attractive approach towards a complete program look-ahead view in order to accommodate relocating abilities in the domain of machine capability. The additional components of help to utilize the instruction pipelining capabilities are a fairly large cache store, separate address and data buses, good stack orientation and Algorithms and data structures. It is the user responsibility as the last point is concerned, but nevertheless, is the only controllable quantity with professional programmers on a machine for deterministic outputs.

The effect of this pipelining is definitely to increase the throughput of the system at runtimes. The quantification can be done as follows:-

Assuming a program p contains n instructions, $i_1$, $i_2$ .....,$i_n$ each having instruction times of $t_1$, $t_2$, ....., $t_n$, the ideal program time on a single CPU which doesn't incorporate any pipelining, will be,

$$= \sum_{i=1}^{n} t_i * P_i = ( t_1.P_1 + t_2.P_2 + .....+ t_n.p_n).$$

where Pi is the number of times the same instruction is used in the program. This gives the worst case measure. But with the embedded parallelism and the major additional factors mentioned above can vary the program time to as low as 20% this value, which means on an average, a 5 fold increase on the Thruput with single CPU architectures belonging to a one-to-one category ( examples being personal computers).

But, today, time has become a precious quantity that no user will be able to spend more than 4% of his time on-line. Also, secondly, it becomes essential to achieve targets within specified time limits, for a working group. These have created the need of a time shared system where also users really can spend useful time with competitive spirit. Thus in case of timesharing machines, the CPU is essentially shared and the other resources, i.e. Cache stores, software memory and user modules are sharable. So the instruction pipelining gains much more credit in providing utility services on a network where distributed computing takes shape.

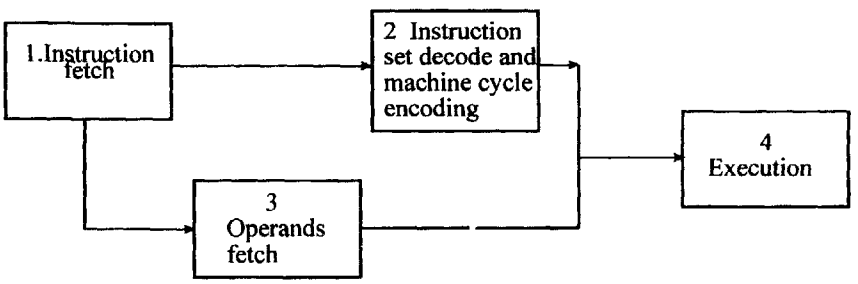



**Fig. 4.1** Instruction flow

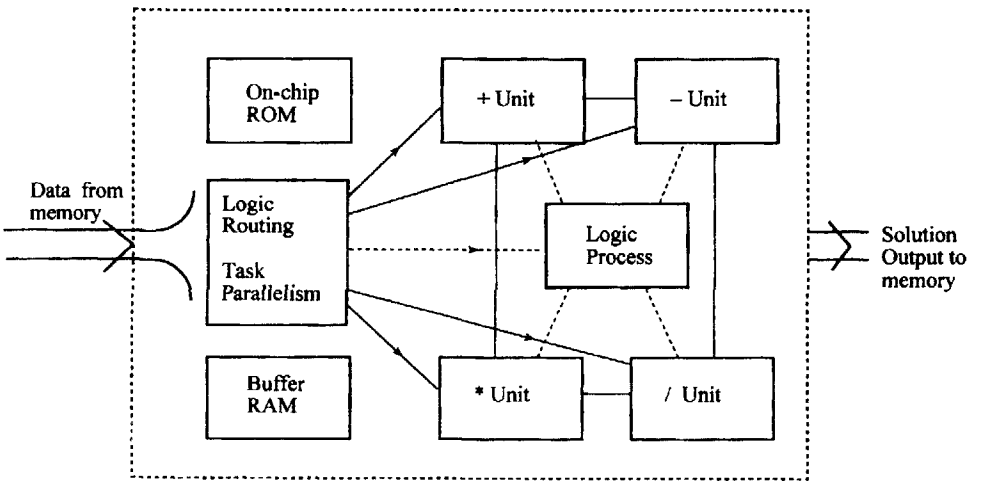



**Fig. 4.2** Compute pipeline architecture

Fig.4.2 represents the computing pipelining (both arithmetic and logical) for *speedup* and decision - making categories which also can be embedded with a CPU. As shown in fig.4-2, assuming independent blocks for each arithmetic operation and a powerful parallel logic routing are available, the instructions decoded are directly mapped on to a static microprogram available on a ROM used dynamically for generating and controlling the powerful control activity. Apart from the registers specific to a CPU, the buffer RAM provides a cushion space for visitors (both operands and operator scheduling) for a smooth flow on a well pipelined architecture. This pipelining of compute units at execute time offer the following merits: viz.,

- good vectorization ratio (data parallelism, operands);
- processor utility is improved;
- good user algorithms for arithmetic pipelines can be invaded and added on to the compute strength;
- Computer aided minimization of computer design algorithms can be implemented said to follow S I M D topology ; and in parallel,
- fast I/O activities can be performed.

The achieving of the above list of crucical parameters by way of pipelining calls for:

i) the number of pipe segments

ii) the length of each pipe, i.e. , time spent on a subprocess

iii) adequate fast cache storage for instruction look- ahead and

iv) good virtual memory management schemes for time sharing machines on a multiuser uni program environment.

It is clear from above, the inherent capacity of pipelined computers look and aspire for :

i) Heavy calls on the data processing and compute - bound users for a batch process.

ii) An interactive environment for group users in distributed computing angle on co-operative workbenches.

Thus, the design of the operating system is important to meet the dual demands of turn around Time and throughput which more often never go together. Hence there is need for a communication processor ( which the IBM calls as I/O channels), where the dynamic job scheduling activity is apparently present for an already available static pipelined hardware. Thus, there is a lot of scope for the system engineers (Hardware implementation) to gain a lot from Research and Development in the testing phase towards effectively applying the technology grounds.

## 4.2 Sample Architectures of Single CPU Machines

*Defn*: *"Computer Architecture can be defined as the inter relation and interaction between the static hardware and the dynamic sitting software facility embedded onto a system with the applications group counting the thruput of the machine in their own faculties/ communities"*. Fig.(4-3)(a) gives the 8085 geographical layout as user hardware design is concerned . In general, a CPU, on its own, is just a soul without the body. Hence it is highly important that CPU, whatever is connected external to the 8085 CPU makes the system set. "Like X + X' = 1 ". Essentially, the 8085 has a wordlength of 8 bits which takes the bidirectional data bus $AD7 - AD_0$ and has an addressing capacity of 64 K bytes of memory ( 1 K byte = 1024 bytes) and device addressing of upto 256 devices. Being a sequential processor, the following features are note worthy:-

1. The 8085 CPU makes use of *multiplexed* address/ data *bus*;
2. Employs an I/O mapped I/O with dedicated input output instructions for process instrumentation;
3. Provides serial communication;
4. Maintains hardware interrupt levels and priority is followed;
5. Has software interrupt instructions;
6. Provides direct memory access with an additional DMA chip ( 8257) for fast and large I/O transfers;
7. Has a clock speed of 2 to 3 MHZ;
8. Instruction length of 1,2 and 3 bytes;
9. Accommodates simple and easy addressing modes;
10. Includes powerful stack instructions for embedded pipelining;

11. Giving user flexibility in I /O design for selective applications:
12. A powerful Logical instruction set;
13. An effective assembler towards microprocessor development system:
14. A monitor ROM resides for machine level decoding with basic 8085 systems.

In side the CPU, with reference to Fig.4.3 (b)

1. The microprocessor has a good register set A,B,C,D,E,H,L;
2. the active accumulator register more often serving as space for input operand as well as delivered result;
3. Set of FIVE flags indicating the process status for powerful actions;
4. The PC (program counter) register as an instruction pointer for a program in execution;
5. A stack pointer to keep track of both systems stack and user stack area;
6. Temporary registers W and Z for scratchpad work in an opaque manner;
7. A powerful ALU with limited arithmetic capability;
8. Address/ data buffers and a Multiplexer to reflect the inherent parallelism of the machine.
9. The instruction register (8 bits) for usual Op code fetching employs the same width for ease of decoding and encoding operations.

Hence out of the 8 bit family of microprocessors, the 8085 can be put to optimum use with the particular chosen environment.

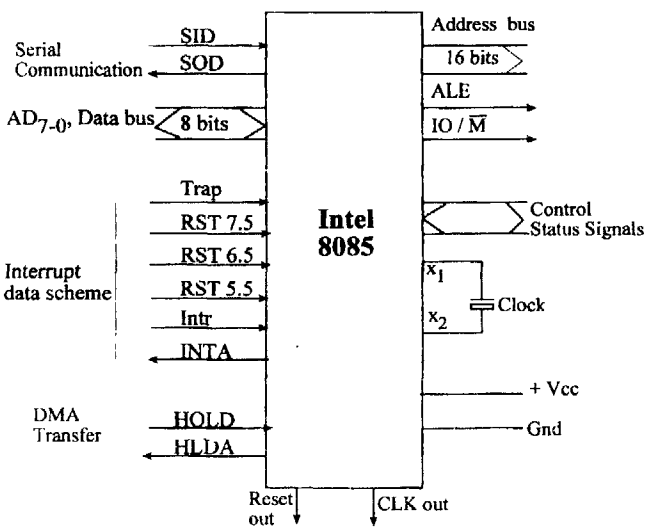



**Fig. 4.3 (a)** CPU 8085 outer

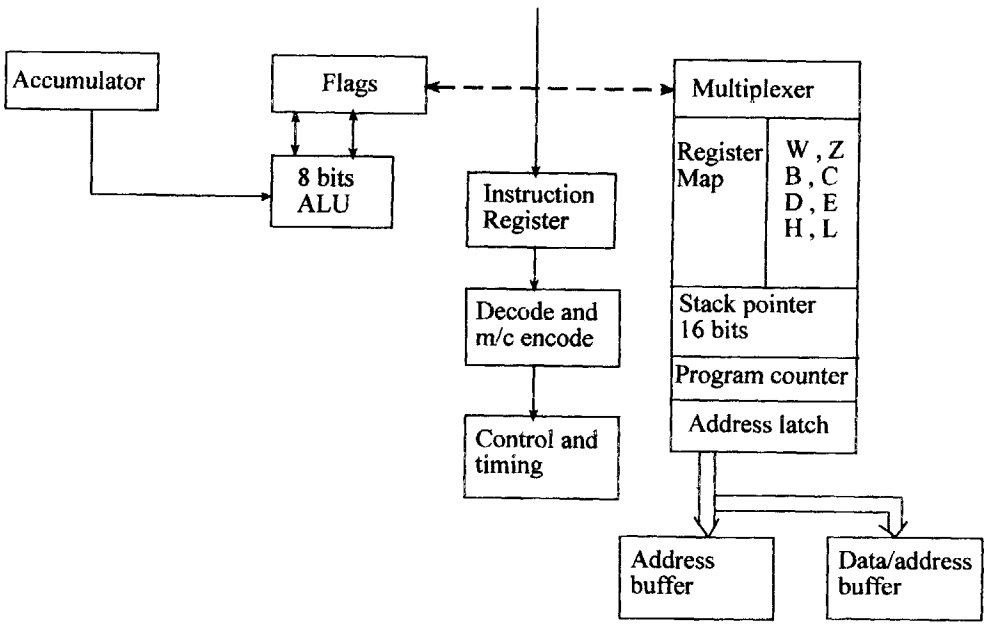



**Fig. 4.3 (b)** CPU 8085 in

## 16 bit Microprocessors

The 8086 from Intel is discussed in what follows:-
Fig. 4-4 (a) depicts the external pin outs of 8086.

This is upward compatible to 8085, meeting the software set of 8085 for byte operands besides supporting 16 bit operations because of the $\overline{\text{BHE}}$ signal as in Fig.4-4(a). This can address upto 1024 kilo (1 mega) bytes of memory (using multiplexed data address bus) with a larger amount of internal registers meant for memory management at runtime for a higher throughput. This includes 9 flags for process status.

It supports a good mix of addressing schemes with the flavour of maintaining a 6 byte ( instruction) queue within the CPU as shown in Fig.4.4(b). It has direct multiply instructions to support arithmetic. Additionally 8086 supports string variables subjected to character data. Also the IN, OUT instructions provide the I/O mapped I/O configuration.

The design of the system itself incorporates the bus interface function and the powerful execute logic. It embeds in itself an instruction pipelining which essentially is a link to data concurrency.

The 8086 accommodates a good exception handling facility supporting,

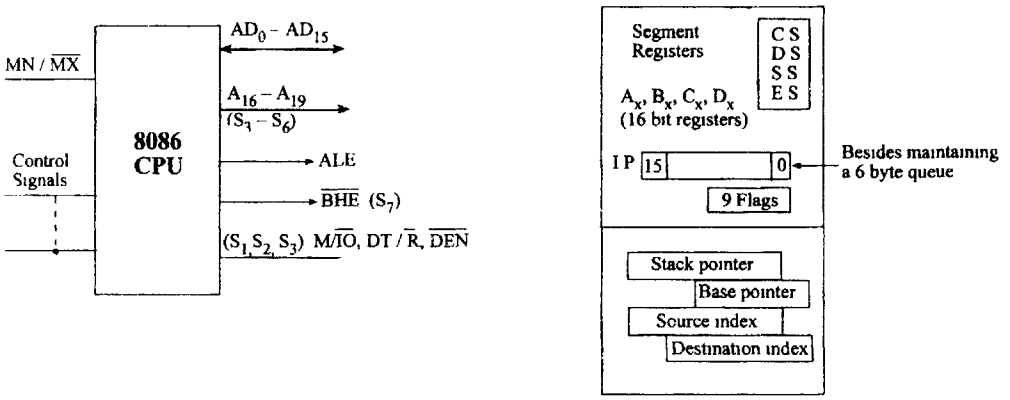



**Fig. 4.4 (a)** External configuration of 8086 CPU & **(b)** Features for pipelining

External interrupts;
Internal interrupts:
Divide by zero; and
*Single step* (tracing at instruction level);

It runs under the supervision of a CLOCK speed of 4 MHz to 8 MHz. These features and enhanced versions from Intel made the 16 bit processors fit for also the personal computer class.

The LOCK instruction can be employed when 8086 is used in MAX mode, which allows the bus arbitration unit to grant system bus to processors also at the same time denying bus requests from other processors. See Fig. 4-4 (c).

The main application of this facility is in distributed digital instrumentation systems where the response time is a critical parameter to be considered for an effective on-line real-time process. This exclusively touches the domain of ASICs ( applications specific integrated circuit group). CPU 8086 is used in MIN or MAX mode

Most programs have a good deal of scalar code ( about 20%) on a sequential Von Neumann machine Thus, good performance cannot be obtained unless the scalar operations are speeded up in addition to Vectorization.

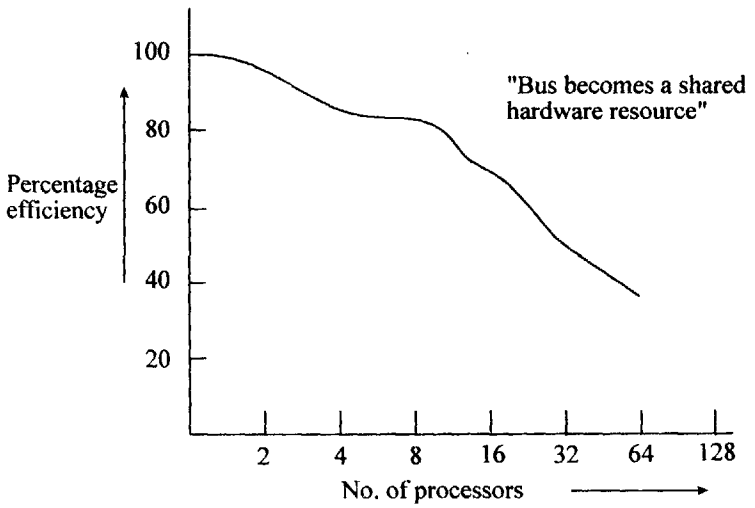



**Fig. 4.4 (c)** Convolution efficiency for different network sizes

This is commonly known as Amdahl's law stated as:

*" When a CPU has two distinct modes of operation , a high speed mode and a low speed mode, the overall speed is dominated by the low speed mode unless the low speed mode can be totally eliminated".*

Zilog Z -80 Microprocessor features:-

- Uses N-MOS technology
- Clock-speed 2 . 5MHz to 6 MHz
- Separate 16 address and 8 data lines.
- Has only two interrupt lines and has 158 basic instructions
- Operates on + 5V power supply.
- Provides more addressing modes having index registers.

  LDIR , (Load, increment and repeat)
  DJNZ, (Decrement B and jump if non zero)
  CPIR , (Compare increment and repeat)

- Includes bit manipulation in registers and memory.
- Good set of I/O instructions.
- Z80 is supported by parallel I/O, the clock timer, DMA and serial I/O chips.

Fig. 4.5(a) and (b) depicts pin-outs and in-contents of CPU Z-80 respectively. Motorola

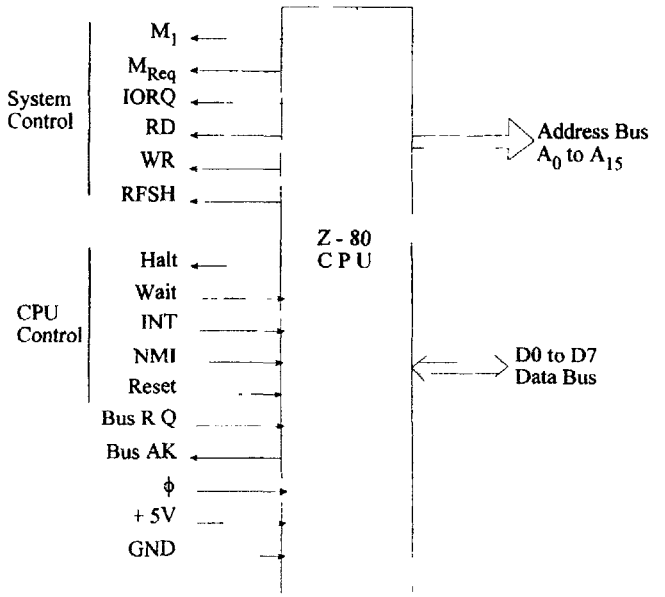



**Fig. 4.5(a)** Z-80 CPU Pinouts

conducts extensive reliability tests to qualify devices, to evaluate process and material changes and to accumulate generic performance data. The results of these tests provide the basis for production decisions and the generation of reliability reports for customer use.

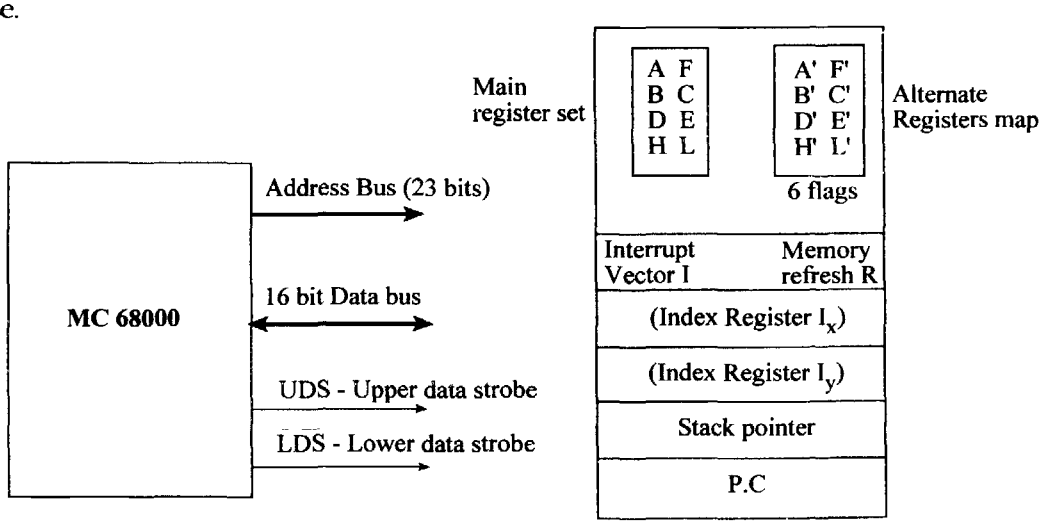



**Fig. 4.5(b)** CPU Z-80 registers

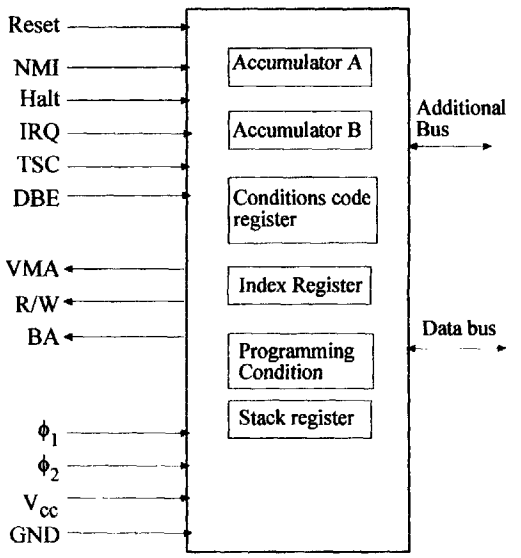



**Fig. 4.6** Motorola MC 6800 CPU

Reliability testing performed by Motorola Mos μp division during the last decade has produced excellent results.

Fig. 4.6 depicts the Motorola MC 6800 microprocessor.

Features of 6800 include:

- 72 basic instructions ;
- Make extensive use of memory refrencing;
- Employs memory mapped I/O;
- Clocks at 1 MHz and uses separate address, data buses.
- Employs N - MOS technology.

The address output lines for the 6800 microprocessor can sink 2 mA in logical 0 state and source 150 μA in logical 1 state. The VMA signal generated from the 6800 microprocessor is called a valid memory address. All system transfers are treated as memory transfers. Thus, 6800 follows memory mapped input /output. The 6800 CPU employs $\phi_2$ clock signal in the generation of the system control bus. Activity with the system takes place on the falling edge of $\phi_2$ clock (1 μ sec). In order to slow down the 6800 microprocessor to access memory or i/o devices, the $\phi$z pulsewidth is stretched for a required length of time to access a particular memory location. The pulsewidth can be extended to a maximum of 4.5 microseconds. By slowing down the microprocessor, we can allow enough time for the input to interface to slower memories without adding any hardware.

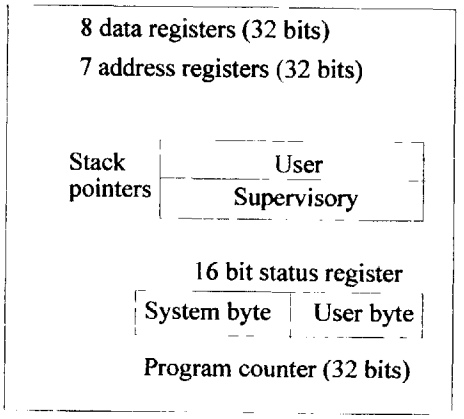



**Fig. 4.7** Motorola 68000 power

MC 6801 is an 8-bit single-chip microcomputer unit (MCU) which significantly enhances the capabilities of the M 6800 Family of parts.

It includes an upgraded M6800 MPU with upward-source and *Object-Code* compatibility . Execution times of key instructions have been improved and several new instructions have been added including unsigned multiply. ( 64 K byte address space). It is TTL compatible and requires one +5 - volt power supply.

On-chip resources include 2048 bytes of ROM, 128 bytes of RAM, a serial communications interface. Parallel I/O and a three-function programmable timer. Software of 6801 includes 8 bit multiply, five flags with PC relative addressing, internal clock with divide by 4 output of maximum 8 MHZ and parallel I/O facility.

An EPROM version of the MC 6801 , the MC 68701 micro computer is available for systems development.

MC 68000 has a 16 bit data bus and 23 bit address bus. Words begin only at even byte addresses. Memory can be viewed logically as linear array of 16 megabytes.

It has a powerful registers set as shown (32 bits) in Fig 4-7 with compact addressing modes including Relative addressing, with a separate user stack pointer. Provides three interrupt levels of HW priority ( $IPL_0$ to $IPL_2$).

This 68000 has a powerful ALU ( 32 bits) with more of a sequential approach.

This with M68451 (memory management unit) can be used to implement multiprogramming features.

Particular attention is required to the problem of resource allocation on a multi-user system. Resource sharing is one aspect on single CPU machines which must be done precisely. In 68000, the *semaphore* concept can be utilized by making use of the TAS instruction. In addition, this CPU provides a certain protection to users.

With these 16 bit microprocessors , it is indeed possible to employ more CPUs in order to increase and improve the throughput with resource sharing concept.

AM 29000 from Advanced Micro devices is a powerful 32 bit μp said to *RISC*.

It has

- 64 global registers ( 32 bits wide)
- 128 local registers.

Any of the 192 registers can be used in instructions. It takes 8 bits to specify 1 register address. All instructions are 32 bits long (with more often a single byte opcode).

- Some are stack registers.
- Ease of programming lies with the programmer to optimize the program size and time as per demand.

**Microprogramming**

The store registers within a central processing unit possess varying capabilities. Some of the desirable chara cteristics of register operations include shift, count, clear and load. The data that is already available as machine code in the main semiconductor memory constitute the program with the macro instructions at the machine level contributing to the subsequent microoperations.

Most of the high level language statements get executed by a majority of data movement involving registers, stacks, memory segments with a strict synchronous, control activity governed by the system clock. Each macroinstruction comprises of one or more machine cycles which inturn are made up of clock periods. Data movement between memory and registers employ the conventional memory address register (MAR) and memory data registers (MDR). This type of data transfers can also be conditional specified by a Boolean variable. Normally the letters constitute a register name and comma separates two microoperations. Arrow denotes the data movement clearly indicating the source and destination values. The symbols used to denote registers and some microoperations is given in Table 4-1.

Table 4-1

| *Symbol* | *Description* | *Examples* |
|---|---|---|
| Letters and /OR numerals | denotes a register | MAR, B, A, R2 |
| Parenttheses ( ) | denotes contents/ partial contents of a register | R2(0- 7) (A), R3(L) |
| Arrow | denotes data movement | (B) ⟵ (A) |
| Comma, | separates micro instructions | R2 ⟵ R1, B ⟵ { (H) (L)} |

At runtime, each machine instruction is decoded and the machine cycles are decided to score up the program counter. These machine cycles can be done wholly serially or a certain amount of overlapping (pipelining) is possible by the inherent control architecture to optimize the processor resource. Normally fixed length opcodes are preferred and the instruction decoding is done according to the nature of the instruction lengths and machining abilities of the processor instruction set. This also leads to the deterministic approach of static microprogramming and fixed finite state machines.

*Dynamic RAMs possess the destructive readout Characteristic.*

Writing the sequence of micro instructions for machine level instructions is known as *microprogramming*. This makes the hardware design to be simple. In microprogrammed design, new instructions can be implemented by writing new sets of microinstructions, which permits *emulation*.

As examples, the 16 bit processors Intel 8086 and Motorola 68000 employ microprogramming.

Some of the arithmetic microoperations are listed below:

$$(A) \longleftarrow (A) + (B)$$

Contents of A plus B transferred to A

$$A \longleftarrow \overline{A}$$

Contents of register A is 1's complemented

$$(B) \longleftarrow (B) + 1$$

B register contents is incremented by 1.

Complement carry CMC }
Set carry flag STC }
Some instructions capable of operating an processor flags.

Other types of instructions might include logical operations, stack operations, special input/output instructions, machine control operators and the communication instructions embedding priority control ( like the SIM, RIM of 8085). With 2 variables, $\{(2^2)^2=16\}$ microinstructions are possible. Some of the logic operations implemented by a multiplexer (MUX) is shown in Fig.4-8. The MUX serves as selecting either arithmetic or logical instruction and also for the register select control. Similarly, a *demultiplexer* is used for outputting activity on a select basis with segmented memory configurations.

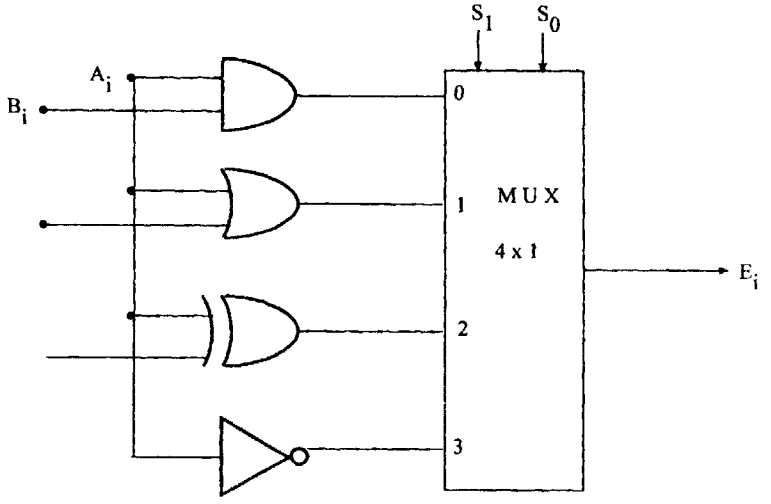



**Fig. 4.8** Logic operations using MUX

The type and methods of operating procedures involve queue, stack and dequeue structures besides the enormous ever increasing capacities of process registers.

The stored machine language program is composed of opcodes and operands as basic building blocks, which may have to cater to different addressing machanisms and varying concurrent events in constituting the instruction set. The user support to hardware interrupts on an operating system is a critical design factor for interactive on-line benches. The microprogramming area must also cater to good amount of buffer storage for *fault-tolerance.*

The microprogramming must desirably support assembler environment with good error diagnostics of a symbolic program presented as input. The microprogrammer has to meet the task of building up essential macro-calls and the embedded subroutine linkages wherever possible to mean a better static software.

The multitasking multiuser machines are often interruptable at the command level and the same has to be catered well in data base management systems and distributed compute-bound processor systems. They have to take care of collision avoidance, deadlock prevention and system security measures.

Fixed design machines of the RISC type often employ hardwired control. Whereas in microprogramming, a writable control memory possesses the flexibiliy of choosing the instructions of a computer dynamically. However, most microprogrammed systems use a ROM for the control memory because it is cheaper and faster than a RAM and also to prevent the occasional user from changing the system finiteness.

### 4.3 Operating Systems Overview

A microcomputer is one which has additional processor memory on-chip of a busy processing environment. Thus the growth of operating systems has realized powerful single CPU architectures like, CP/M for 8 bit. µps; MS-DOS for IBM PCs and the UNIX with 32-bit machines for multitasking activities.

In essence, the software tools designed to aid the µC user in performing a variety of tasks is called an operating system (OS). The view of OS is given in Fig. 4-9.

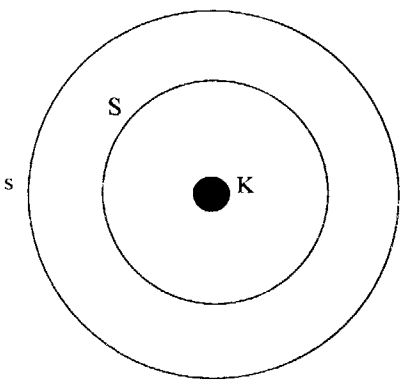



**Fig. 4.9** Operating system view

Men can only be calculative due to the inherent facts of limited storage availability and still very less memory for arithmetic processing.

As the user needs increase both in terms of memory requirements and computational power, one necessarily has to have a programmable machine which is essentially termed a computer. The system in operation governed by the operating system view is given in Fig.4-9.

K is the Kernel , S is the system shell and it is the integration between K and S that throws open the system's ability as a programmable machine to the user community. The space coverage between K and S is only used as a Resource Manager software and the space between S and s invites the manufacturers capacity to deliver standard packages, tools, libraries, compilers, editors, etc. to meet the diverging needs of a USER.

The need for a Language for communication as an on -line user of a machine as a serious concern has been felt very much and computer systems i.e., IBM PCs to multiuser environment like UNIX machines have become an acceptable standard in organisations. The Architectural capture during the time till now has gained more emphasis on communications thus leaving the computing needs to a limited scientific user community.

All these machines are SISD (a sequence of instructions and a sequence of data) stored in memory and operated upon sequentially to get the solutions is the method followed on John Von Neumann architectures which is a standard today as shown schematically in Fig. 4-10.

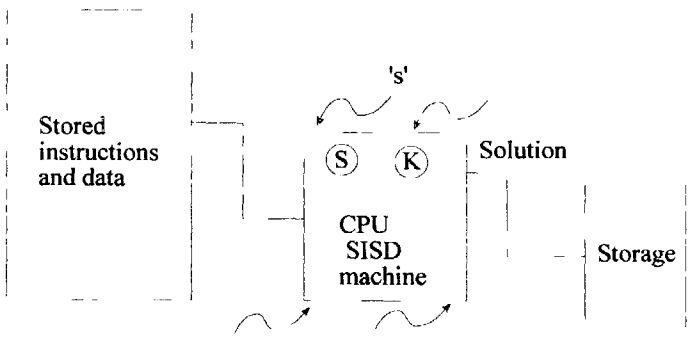



**Fig. 4.10** Von Neumann machine

An off-line system doesn't require the user while it performs the process as dictated by the input example being like IBM/360 batch processing systems.

The trend of other manufacturing giants like HP, APPLE is towards centralization or dechannelization of a multiuser environment in the application Base.

One of the hallmarks of UNIX is the fact that the user interface manager ( called SHELL) is a program like any other programme and can be replaced.

The O S consists of the Kernel, the shell and utilities.

Read and Write are complementary operations in an interactive environment which goes on highly sequentially and silently.

More and more PCs now bridge the gap by using high-speed SRAM (static RAM) chips to cache traffic between the CPU and DRAMs. A typical 486 or pentium system might have 256 KB of SRAM cache. But SRAM is much costlier than DRAM, and boosting cache beyond 256 KB yields a diminishing rate of return.

## Enhanced DRAM

By integrating a small SRAM cache with a fast core of otherwise generic DRAM. Each EDRAM chip has 2 kb of 15- nanosecond SRAM and 4 Mb of 35-ns DRAM.

The first pentium systems ( 1993) didn't take full advantage of the new *CPU (80486)* 66 MHZ.

Old designs are being adapted for use with faster processors. To boost critical I/O speed, engineers are using new bus designs and storage sub systems. Equally important, there are harnessing techniques such as redundancy, error corection, and predictive diagnostics to give 80 X 86 systems the reliability and manageability typical of hosts.

User functions such as voice annotation of documents or spoken attachments to E-mail messages must be provided through software, and adding music functions requires additional hardware - either a dedicated synthesizer chip or a programmable digital signal processor. The pentium provides a 64-bit memory interface and 32 bit I/O bus for load balancing. The pentium will be the platform of choice for doing things like dual processor multi-processing for reliability in the information age.

Tandem network is a configuration for standby redundancy in terms of multiprocessors for *fault tolerance.*

The need for high availability leads to things like redundancy, ECC ( error - checking and correction ) memory, disk arrays and predictive diagnostics which drive the system architecture.

On-chip performance-monitoring registers maintain statistical information on cache hits, bus transactions and average processor waits. The pentium's FRC ( functional redundancy - checking) mode lets two chips run side-by-side and compare their results in real time. Unfortunately , FRC mode doesn't indicate a resolution in case of an error, so it serves to flag but not solve data Integrity failures.

An operating system may have to meet the dual needs with the ability to support more than one user and the capacity to let one user simultaneously execute several tasks. If the particular OS meets the above factors, it is termed a multi-user multi-tasking system.

MS DOS is a single user OS, which is very powerful as a single CPU architecture is concerned. A multiuser system should incorporate certain definite features like user identity, system statistics, user protection (*security*), and provide better opportunity for on-line communication among the group of men evolving on a software project by means of a good I/O management protocol.

As OS may reside on a disk, or in ROM or both. Usually the static software strength sits on ROM and the other systems software (major portion) remains on disk to be used on demand both from the users and the microcomputer.

Some major constituents of an operating system include:-

- Command interpreter
- File Manager
- Assemblers
- Linker -/Loader
- Text editor
- Compilers of various high level languages
- Certain utilities, mainly for I/O activities and to support modular programming.

Currently the multitasking OS supports *window manager*. This is affiliated to the fast and powerful graphics activity on a communication base.

The window manager permits–

- opening a new window on screen;
- switching across windows( using mouses, lightpens etc.)

Activating a window. Some commercially available WMS (windowing mechanisms) are from suntools, X-windows and PC tools which essentially form an ingredient of real-time on-line user pipelining on a single CPU machine for better and artistic outputs. This is supported by the CAD/CAE applications area of scientific users as well as the very dynamic and fluctuating commercial circles. On -line query is an important tool which suits for reliable and active database management systems.

With the ever increasing complexity of software management on single CPU machines, a lot have come up as diagnostic tools in the hardware area mainly for system development and partly for service sectors of a powerful single CPU marketing associates.

For the ease of instruction decoding, it is preferable to have *fixed length op-codes* with single CPU machines which is followed by many a microprocessor family.

### 4.4 Software Strengths:

Transistors, integrated circuits, and VLSI have resulted in dramatic decreases in hardware costs. On the other hand, software is labour-intensive, and personnel costs are constantly increasing. See Fig.4.11. Similarly, software maintenance is an increasing portion of software cost because with passing time more software accumulates. The typical life span for a product is 1 to 3 years in development and 5 to 15 years in use (maintenance).

Software maintenance involves enhancing the capabilities of the product, adapting the product to new processing environments, and correcting bugs. Typical distributions of maintenance effort are 60, 20, 20% respectively.

Modern programming languages provide a range of features for software development activity. The factors include separate compilation, user-defined data types, flexible scope rules, and concurrency mechanisms . Usually, for a single CPU machine the structured programming methodology has been agreed as a major factor to decide its capability. Pascal is a suitable tool towards software training methodology and COBOL, a well-structured language is the strength to data processing activities.

Ada is named after Lady Ada Lovelace, during 1900s,. known as the world's first computer programmer. This was developed for "embedded" computer systems. This provides computation, communication and control functions of a larger system which has to meet concurrency constraints and interrupt-driven real time processing. Strong type checking improves the quality of a program by catching errors in the declaration and use of data entities through a combination of compile time, load-time and execution time checks. Any feature must provide security to the system with rigid programmer flexibility

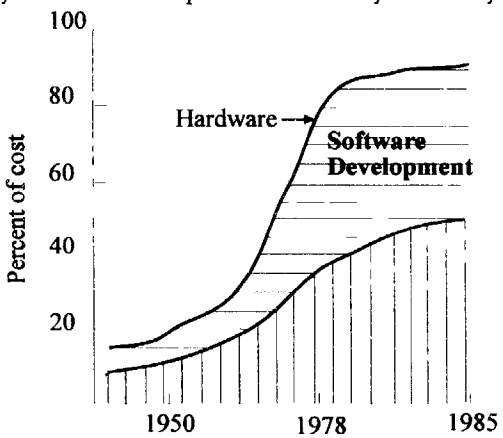



**Fig. 4.11** Changing hardware - Software cost ratio

Lisp is not suited to numerical applications nor is BASIC fit for list processing. The fundamental data structure in APL is the one-dimensional Vector array. FORtran permits mixed mode operations between similar data types.

### Separate compilation

The ability to develop modules and subsystems, to compile and store them in libraries, and to have the loader automatically retrieve library units and link them into the object code of a system provides a powerful abstraction mechanism.

FORtran, to some extent, and Ada support the separate compiling aspects. Only the storage requirement is high but ensures speedy and safe operation of a product development on a uniprogramming multiuser single CPU machine. Among the user-defined data, the Record type is a primary feature supporting file processing.

*Concurrency Mechanisms*

Two or more segments of a program can be executing concurrently if the effect of executing the segments is independent of the order in which they are executed. These program segments are referred to as tasks. On a single CPU machine, *code segments*, can be interleaved to achieve parallelism by a pipelined approach.

Two fundamental problems in concurrent programming are synchronization of tasks so that information can be transferred between tasks, and prevention of simultaneous updating of data that are accessible to more than one task. These problems are referred to as the synchronization and the mutual exclusion.

When variables are shared, multiple processes have access to a common region of memory. The use of semaphores, critical regions and monitors are used to arrest the problems that may arise otherwise.

In distributed processing environments, the asynchronous message passing is accomplished by use of buffers in concurrent tasks.

For synchronous message passing architectures, Ada calls its as a rendezvous; as exemplified in CSP. *Deadlock* is the worst outcome of concurrent systems which may prove the system as insecure. Here concurrency introduced by parallelism in software must take the machine architecture as a serious constraint and a sincere approach.

*Petri Nets* invented by Carl Petri in 1960 at the university of Bonn, West Germany is a graphical representation technique for concurrent events.

Petri Nets are state oriented notations for specifying parallelism of operations. They are used to specify synchronization and mutual exclusion among concurrent operations. This technique overcomes the limitations of finite state mechanisms for specifying parallelism.

**Unit testing:**

The performance of a CPU is derived by the individual programmer independently doing the unit testing:

Coding and debugging = Unit testing.

A program unit is usually small enough that the programmer can test it in greater detail (self-test) than will be possible when the units are integrated into an evolving software product. The routine types of tests include functional, performance and stress besides the exceptional structure tests.

Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. This is called as "White box" or "glass box" testing.

This concerns with test coverage criterion for enhancing software reliability by detecting missing path errors, computational errors, and domain errors.

*Debugging* is the process of isolating and correcting the causes of known errors. Commonly employed techniques include induction, deduction and back tracking. More often logical errors are the most common mistakes which lead to wrong branchings or abrupt terminations (processor Halt).

Hence the dual problems of verification and validation is a continuing task in a software development programme. Process audits examine applied software engineering techniques and tools in practice for product support which can improve quality and productivity. The software strength really is reflected by embedded parallelism in machine regarding processor-memory interactions, system software and applied utility software.

## 4·5 Microcomputer Development Systems

With the advent of microprocessors ( CPU on a chip) . it has become literally possible to simulate any type of process to a desired level of accuracy. Still the wordlength and the instruction set of the processor play a dominant role to compete in qualifying to be the primary component of a development system. Thus microcomputer based development systems really call for desired level of expectations as the architecture is concerned, and more often they are to be of enormous capabilities as of a general purpose machine. The other ingredients like system software, language translation, hardware support, good and fast emulation techniques and dedicated real-time control applications add dimensions to the geographical geometry of the system. Varying designs in EPROM programmers attach intimacy in the realm of graphical user interfaces and complex adaptible instrumentation control systems involving fuzzy logics and neural computing. Thus from the above listed points, it becomes quite apparent that a very few systems can cater to this expert system involving *reengineering* of engineered software, strong dynamic microprogrammes availability , and a good fault-tolerant system with a definite finite state automata.

Some of the major application areas of MDS (Microcomputer Development Systems) are:

A large number of assembler environment for different machine level language users, an equally good amount of debugging and diagnostic tools, and a strong emulation team for portability and pottability reasons in distributed processes. The debugger provides commands to disassemble object code into assembly language mnemonics, to trace program execution one step at a time, to insert breakpoints into user programs and examine and modify CPU registers.

In addition, it is expected of the prospective MDS to rise to the data-base management

systems for in-house tailorable software development for information presentation and intelligence representation in the competitive export markets towards customer satisfaction. Centralised data-bases and *computer networks* are crucial issues which attract attention from other related items like economy, security and authenticity in the global telecommunications scenario.

Instruction sequencing and interpretation used to activate the control signals cause concern in the microprogramming arena. The state table approach with delay elements incorporation is normally followed on CPU designs. The sequence-counter method perform well for RISC machines, for many digital circuits are designed to perform a relatively small number of actions repeatedly.

Microprogramming calls for control data to be stored in a ROM or RAM called a control memory. Each machine instruction is executed by a microprogram that acts as a real-time interpreter for the instruction. *Emulation* is possible when working with two similar machine level languages. More knowledge of the machine is required and calls for micro-assembly languages so that executable microcodes can be stored directly on the control store. M.V. Wilkes ( in 1951) proposed the microprogram design in which each bit of the control field corresponds to a dintinct control line. Though naturally fixed length opcodes are preferable in machine instruction codes, the varying microinstruction length is attributed to the following factors, viz., the degree of parallelism required at the microinstruction level; the way in which the control information is represented or encoded and the way in which the next microinstruction address is specified. Computers with writable control memory are often considered to be dynamically microprogrammable. The cost of control memory is measured by two parameters: its width w, which is the number of bits per microinstruction; and height H, which is the number of microinstructions it stores. Another chaining of programs call for nanoprogramming (used in 68000 series of microprocessors) thus giving a higher flexibility in hardware control. System macros and macro calls are powerful microprograms utilized to support the machine activities. Today microprogrammable machines with EPROM programmers and emulators are helpful as end control units in process instrumentation and distributed computer networking areas.

In striving for faster machines, designers have constantly migrated functions from software to microcode, , or from microcode to hardware. For example, by adding an instruction that requires an extra level of decoding logic, a machine's entire instruction set can be slowed. MIPS ( Microprocessor without interlocked pipe stages) of standford, a word addressed machine, is a RISC one of the research machines. The notion of moving functionality from run time to compile time ( compiler technology ) is a novel computer RISC design. *MC 68020* is a product of 60 men years of design effort. This chip is delivered using high density complementary metal oxide semiconductor process technology packaged in a 114 - pin grid array offering a clock-rate of 16.67 MHz, with

maximum of 1.75 watt power dissipation. The Acorn RISC machine (ARM) of cambridge, England has a 32-bit data bus and 26 bit address bus. All instructions are 32 bits in length and has sixteen general purpose registers. Supervisor and interrupt modes have access to 9 registers unavailable to the user mode programs. These registers are used to minimise the interrupt latency and context switching times as well as provide support for simulating a DMA channel. Computer architectures have become complex enough that it is often difficult to analyse program behaviour in the absense of a set of benchmarks to guide that analysis. One area where very high performance figures are required is in Artificial intelligence (A.I.). The performance rating of AI machines is measured in logical inferences per second ( Lips) . At times an array of ten thousand transputers could potentially deliver a similar performance figure.

### 4.6 Conclusion

Thus the pipelining method may include all the pipe steps in the voluminous sequential process. If certain pipe's are optional, then pipeline control becomes a serious problem because of tight synchronization requirement among the stages. When program loop segments are encountered it is desirable to provide a high-speed memory for these sensitive data in order to produce a sensible throughput on single CPU machines.

Some examples of pipelining include application of more similar functional modules as on a SIMD spectrum, a good architectural blend ever supporting the *data flow machines* (which requires computational split-up to the greatest degree because of hiearchical arithmetic execution priorities followed by languages ) for primarily matching the bus and bandwidth bottlenecks, and to the realm of distributed processing real applications all mainly aimed at a short Turn-around-time ( better service) and thus enhancing the available thruput of systems. The other major concern is the domain of algorithmics which may play a dominant role of sitting software strengths besides the dynamic professional software engineers implementing the difficult strategies said to the growing needs of parallel processing architectures. The future trend is more aimed at self verification in fault tolerant computing and automatic error control and recovery mechanisms in communication networks.

Thus, in principle with the system clock running high, a single CPU pipelined machine can provide a better turn around time with or without virtual memory support.

**KEYWORDS**

Pipelining, Instruction cycle, Speed-up, Architecture, Multiplexed bus, single-step, object code, semaphore, RISC, micropgrogramming, emulation, demultiplexer, fault-tolerance, 80486 CPU, fault tolerance, security, window-manager, fixed length opcode, concurrency, code segment, deadlock, Petrinet, debugging, re-engineering,

## PROBLEMS

1. What is pipelining? Discuss its classification with respect to parallel processing
2. Explain the pipelined version of a floating point adder.
3. Write on specific advantages of counters as controlling elements in computational activity.
4. Write a machine language program selecting anyone microprocessor to generate clock pulses of time period equal to 1 ms with a duty cycle of 25%.
5. What are the merits of using multiplexed address-data bus on single CPU system?
6. What is a bit slice processor?
7. Explain in detail, the organisation of any 8 bit microprocessor.
8. What is the purpose of instructions for disabling interrupts in I/O management?
9. Highlight on the SIM , RIM instructions of 8085 CPU for use in process control instrumentation measurements.
10. Bring out the essential differences of 8085 from that of Z80 microprocessor.
11. Explain the importance of segment registers towards memory segmentation.
12. Differentiate the user and supervisory states of Motorola 68000.
13. Write notes on microcomputer development systems.
14. Compare and contrast the use of fixed length and variable length opcodes towards microprogramming.
15. A number of computers have micro-instructions containing an "emit" field in which the microprogrammer can place an arbitrary constant for use as an immediate operand. Give some general reasons for including an emit field in microinstructions.
16. Explain the use and implications of wait states of microprocessor instructions.
17. Account for the use of fault tolerant microcomputers in process-control instrumentation. What is data acquisition?
18. Correlate the implications of the inclusion of cache memory and process segmentation towards the benefit of pipelining.
19. Distinguish between Moore and Mealy machines.
20. What do you mean by computer Architecture?
21. Distinguish the technique of defining Macros from user spelt subroutines.
22. Write a note on error - detecting codes for system reliability.
23. Explain the architecture of anyone RISC model you have come across.
24. Mention the advantages of 2's complement arithmetic towards pipelining in computer construction .

25. Name and explain some of the performance measures of application software on single CPU timeshared system.
26. What factors influence the selection of modular programming in distributed environment?
27. Scan on cognitive, augmentive and notational tools towards quality and productivity factors in software management Circles.
28. Write on Petri Net modeling to concurrent programming.
29. Comment on infix and post/prefix notations for computer arithmetic in data flow computing.
30. Give the equation for the cyclomatic number thanks to Mccabe. What is structured programming?

# CHAPTER 5

# PARALLEL PROCESSING AND FAULT-TOLERANCE

## 5.1 Process Networking

With most of the sequential computers adopting the John Von Neumann stored program concept, the organisation is of the SISD (single instruction single data) scheme. To improve the processor utility, the pipelining and extra CPU- in storage shall definitely produce a smooth flow. But the need for shorter program times of a single job and a higher throughput in multitasking on a single CPU bench called for *virtual memory management* and resource sharing techniques. Still with the identification of sensitive program spots for vectorization, the scalar type of instruction can never be eliminated of a program. Thus, the *dataflow* approach towards parallel activities on a processor unit containing multiple functional units repeated many in number ( like SIMD ) will definitely pave the way for real speedup on enhanced pipelined approach. This technique of reducing the Turn arround times of jobs involves a new concept in data flowing across the machine. The data flow model of this kind is explained in section 5-2.

In situations where the targeted turn around times can only be achieved at the expense of multiple processing units (CPUs) of the same or varying structures, the MIMD system domain has to cater to form a tightly - coupled compute system meeting the dual aspects of dataflow and distributed processing. This type of environment needs parallel language constructs, semaphores for collision detection towards resource sharing and multibus protocols with a good connectivity base. Already the transputers technology has achieved momentum in this particular area of parallel machine towards the super computing ends. In most of the establishments like companies, workshops, academic institutions etc. the real time data processing has to be done in a distributed way with a functional approach of each mode. This essentially calls for computer networks which are highly localized or channelized for operations and a LCS ( Loosely coupled system) approach for message passing, information sharing and decision making results in powerful database management meaning a well maintained system that shall not go *insecure*. The distance also plays a major role in determining the type of network suitable like local area ( LAN), wide area ( WAN) and long haul networks. The aim is not to focus on the design aspect of computer network architecture which otherwise by itself is a subject of concern.

A *graph* G ( V, E) consists of "n" vertices ( nodes) of computers and "m" edges (links), where n and m are integers. The value that n and m are assigned in a network will decide the connection topology like star, Ring, Tree, complete, intersecting or irregular.

A *network* can be defined to be composed of a set of autonomous computers which very much co-operate during the processes for co existence. "In any graph, the number of vertices of odd degree is even". A graph is completely connected if the number of edges 'm' is atleast equal to n(n–1)/2 where, 'n' is the total number of vertices. This implies that there is a direct link across any two nodes. But in real-time applications like Telephone networks, one cannot assure complete connectivity and only linking junctions will provide a way for real traffic control and a good grade of service. This paves the way for Routing in networks and related issues - like message and packet switching attempts. Dijkstra's shortest path finder, Floyd's method and other interesting and evolving strategies of planning for effective *algorithms* towards problem optimizations with good performance figures in the realm of software engineering is a vast open area for researchers in their respective objectives.

One example is the use of SQL ( structured query languages) in database management which is at present taking competitive spirit on the network layers. Pattern matching and image identification is an innovative application of graph theory which has already found a place in medical diagnosis, chemical analysis and in forensic sciences. The Powerful use of graphics workstations across the globe have assisted a lot in weather forecasting and the television entertainment spheres MIMD (multiple instruction multiple data) draws good amount of attraction among architects.

## 5.2 Dataflow Mechanisms

The floating point arithmetic with the mantissa exponent notation (employs carry save approach) serves a good blend of *arithmetic pipeline* for matching memory bandwidths and thus helping in smooth dataflow. Fig.5.1 gives the usual instruction pipeline on SISD computers. MIMD machines capable of executing several independent programs simultaneously are aimed at throughput reliability, flexibility and availability. These employ multiport memories with time shared buses and cross- bar switching.

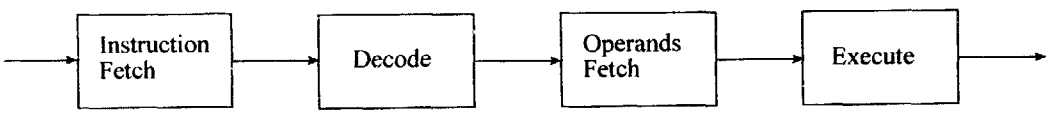



**Fig. 5.1** Instructions overlapped fashion

Fig.5.2 depicts the control versus speedup on MIMD protocols. It is inferred that small number of fast processors are a solution to the objectives rather than large number of slower processors which only increases the synchronization overheads. The program segments may be visited many a time by the same operand set and collision must be avoided by a fine process strategy. In multiprocessing, the minimum time that must elapse between the initiation of any two processes is called the Latency.

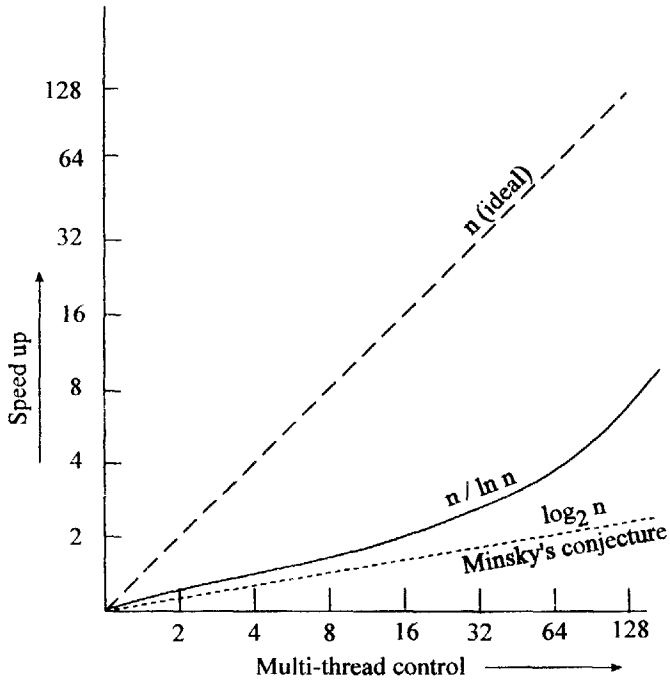



**Fig. 5.2** MIMD Speed-up

Pipelined multiprocessor system is state of the art design in parallel processing. These Von neumann models are rather termed as control flow machines.

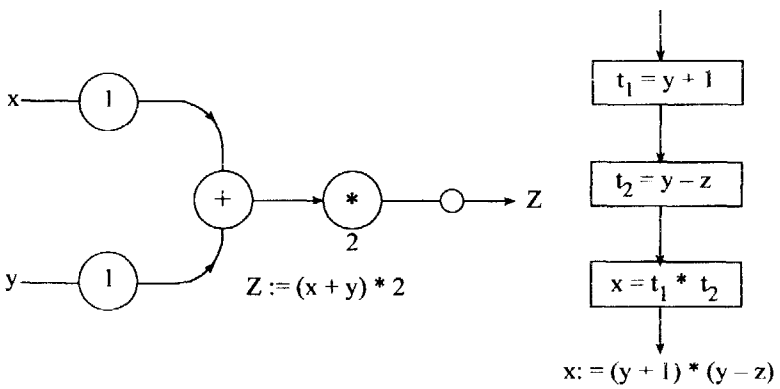



**Fig. 5.3** Data flow graph

The dataflow concept is being applied for maximising parallelism and instruction initiation depends on data availability. Program instruction can be represented by dataflow graphs as shown in Fig.5.3 example.

The data tokens are direct store places and the control tokens are instruction activators. The processes y + 1 and y – z can go in parallel to prepare for the final result of x = ( $t_1$ * $t_2$) as in Fig.5.3 on a dataflow approach. The basic dataflow machine of Fig - 5 . 4. has gone into VLSI microelectronics area.

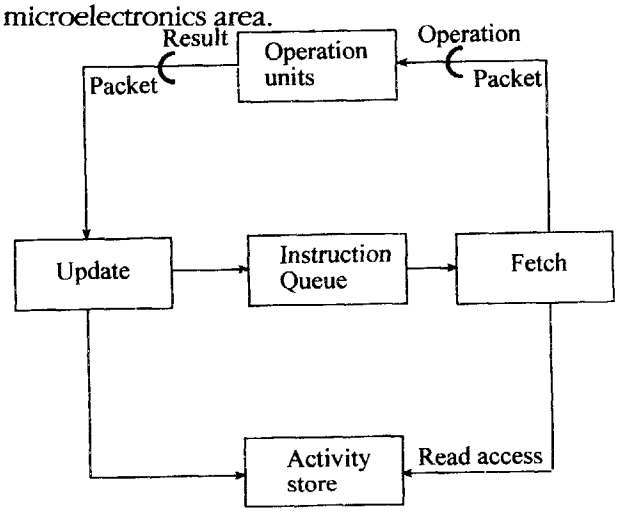



**Fig. 5.4** Basic dataflow mechanism

ADT (abstract data type) is a mathematical model together with various operations defined on the model.

In a static dataflow model, only one token is allowed to exist on any arc at any given time, otherwise the successive sets of tokens cannot be distinguished. The Jack Denis group at MIT and Manchester ARVIND's machine configured a dynamic dataflow approach employ tagged tokens using a pipelined ring structure as in Fig.5.5.

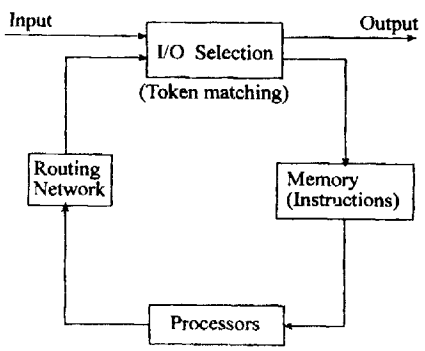



**Fig. 5.5** Ring structured data flow organisation

The ID ( Irwin Df) design VAL ( Value algorithmic language) is compatible with the use of dependance graphs for compilation on conventional computers.

The major design issues in Data Flow Computing (DFC):

Language implementation (one to one assignment rule); Decomposition of programs difficult; developing intelligent data driven mechanisms; efficient handling of complex data structures like arrays; memory hierarchy and allocation; software supports for compilation; and performance evaluation.

**Advantages**

DFC assumes highly concurrent operations, matching with VLSI , VHSIC technology and increasing programming productivity for vectorization.

**Shortcomings:**

Data driven at instruction level causes excessive pipeline overhead per instruction which may destroy the benefits of parallelism. The data flow programs tend to waste memory space for the increased code length due to the single assignment rule and the excessive copying of data arrays. The packet switching becomes cost prohibitive with many processors.

Few of the experimental df machines use a tree structured architecture with each computing element supported by offspring elements to aid compilation, simulation and performance measurement programmes.

As a design alternative for dfc, the dependance driven approach is suggested by Motooka et al ( 1981) and Gajski et al ( 1982) which aims to raise the parallelism level for compound function level at runtime. This includes array operations, linear recurrence and pipeline loops.

The multilevel event driven approach shown in Fig.5.6 is given by Hwang and Su where an event is a logical activity defined at job level. Today's supercomputers expect the *dataflow* as self-servers which may be a viable approach to improve computer's performance and satisfy the aspiring clients. More of the compute-bound problems and real-time processing call for the same sequence of instructions acting on different datasets. A good example is multiple single output functions minimization needing array processing in the realm of computer aided minimization involving multiple functional units with their own *local memories* working under the Central supervisory control unit. Another example is in real time transactions (railway reservation system) which has a shared database memory and concurrent handling of datafiles in the tightly coupled demand based secure systems. Voluminous dataprocessing towards *sorting* and search activities again involve divide and conquer rule for optimization. Many numerical solutions for image-processing involving fourier transforms, various types of recursion and iterative methods repeated

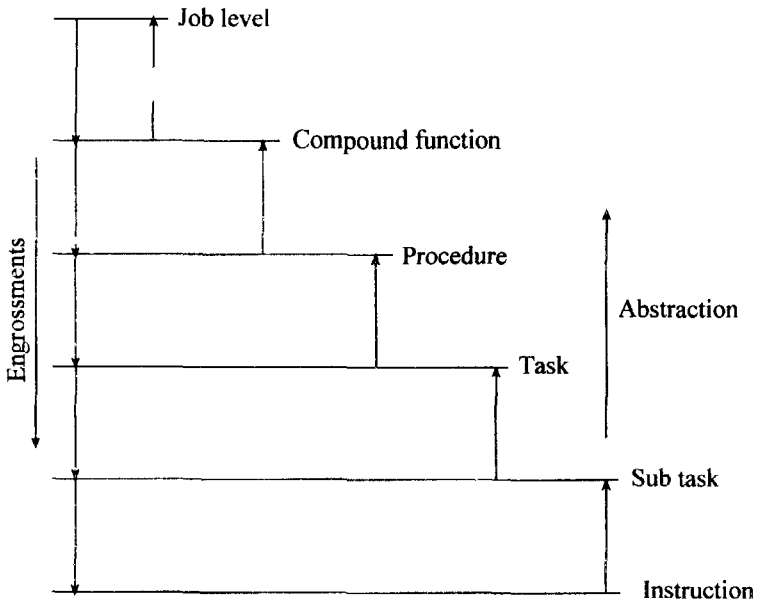



**Fig. 5.6** Machine with structured control hardware

on varied data values to arrive at appropriate solutions and models involving statistical techniques for simulation and performance criterions again call for a parallel processing workbench. The most sought after data processing is a combination of SIMD and SISD, on these specific and unique problem areas. This trend towards SIMD is also known as vectorization for increased thruput towards speedup. The scaling up of functional processing elements with an embedded flavour for parallel working is a selective domain of computing architectures.

The dataflow diagrams assist for the MIMD actions on a scheduled tight processor for effective processor utilization. The real *MIMD* implies a multiprocessor ( of the same or different types ) configuration attached to the transputers domain and the distributed networks. The rare combine of MISD ( multiple instructions single data) finds more fitting to applications like medical diagnosis, logic processing with prolog and Lisp languages paves the convergence to expert systems for quality and quantifiable decisions. The parallel processing is also realized employing the associative memory concept in dedicated search applications with little hardware overhead (match registers and comparator circuits). The higher power dissipation and pin/circuit ratio are the major difficulties encountered. Arithmetic pipelining towards computational tasks with concurrency will achieve a speed of 100 Mflops with reasonable precision on super computers like the CDC 6600. The unix operating system satisfied the cream of pipelined approach for interactive machines.

## 5.3 SIMD Class

A single control unit fetches and decodes instructions. The processing elements operate synchronously but their local memories have different contents. The distinct category of pipeline, i.e., vector or array processors is attributed to factors like : the complexity of the control unit, the processing power and addressing methods of the P.E.s and the interconnection among the processing elements. In array processors, the P.Es communicate with their neighbours through a network switch and are suited for vector processing. The design complexity is felt to cover a wider range of applications. Fig.5.7. shows the example of a SIMD sub system, Illiac IV which is primarily meant for matrix manipulation and solving partial differential equations. More often, since this is not meant for scalar mode, the basic software is hosted by a satellite computer and *Illiac IV* becomes a powerful special attachment as shown in Fig.5.8.

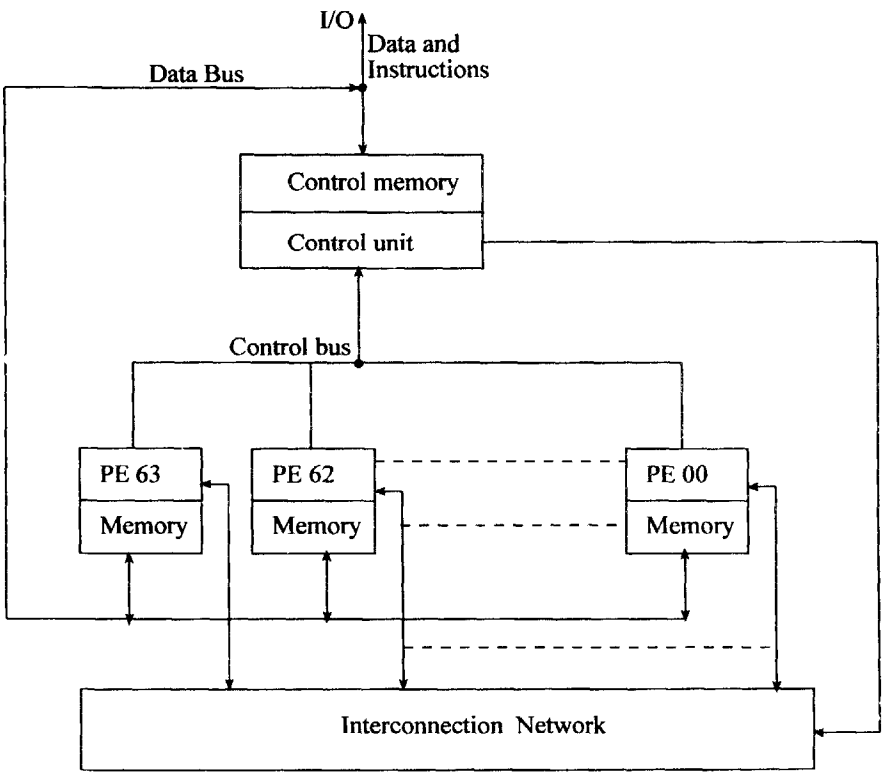



**Fig. 5.7** Illiac IV for SIMD

Glypnir ( an AlGOL based language ) and Fortran based CFD serve as language domains. The difficulties in producing efficient compiled code have slowed new language developments and the burden of designing efficient algorithms with the implied knowledge of the CPU architecture is left to the user. "Turn around time is defined to be the time

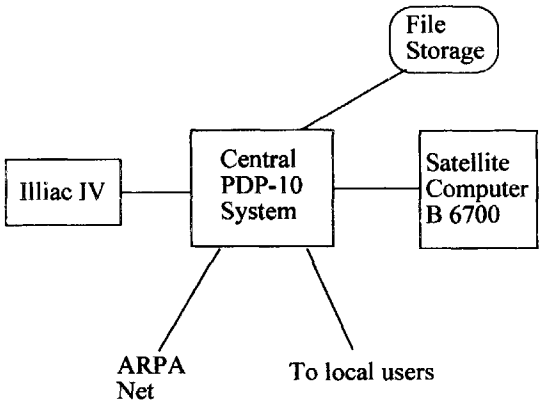



**Fig. 5.8** Illiac IV attachment

lapsed for a task to be completed from the time it is initiated in the area of parallel processing", which varies over a broader range of values. *Handler* has proposed a scheme for representing a computer. Each system is denoted by a triplet : C = ( K, D, W) where, K is the number of control units, D is the number of ALU's controlled by each unit and W is the i = unit length of the entities managed by the D's.

As example systems,

IBM system / 370 = ( 1, 1, 32),

C D C 6600 = ( 1, 1, 60),

IBM/ 360 dual processor = ( 2, 1, 32),

Illiac IV = ( 1, 64, 64, ) and

C. mmp = ( 16, 1, 16).

The goal of scheduling algorithms is to keep a maximal rate of initiations while avoiding collisions. *Latency*, which is the interval of time between successive initiations can be used as a performance measure contributing to throughput rate. I /O processors ( called communication processors) can be a desirable component on a tightly coupled system of interactive users in addition to database management and sharing strategies with distributed processing trends. DSP algorithms need hardware multipliers and adders in arithmetic processors. Negative numbers are handled by fractional two's complement representation.

A functionally divided multiprocessor system is designed primarily as a network of tasks which communicate through data objects resident in common memory. *RISC* architectures for matching I/O bandwidth towards MIMD invite process networking and dataflow. Dataflow machines involving a synchronous and stochastic processing support parallel environments including neural computing. The use of semaphores and preemption

techniques can provide deadlock avoidance in multiprocessing trends. Highly parallel processing platforms may use multibus concept with reliable memories. The major isues of parallel systems involve the algorithms design for potential concurrency, inclusion of language features to explicity express parallelism, the synchronization problem among cooperating and competing processors and scheduling and load balancing techniques. The transfer speed is determined largely by the number of software synchronization points. The ratio of the bit rates for the three methods, programmed I/O : Interupt controlled I/O : direct memory access is approximately 1 : 2 : 4.5.

The use of bit slice microprocessors offer higher speed, ease of expansion and flexibility. A programmable logic array is functionally equivalent to two-level AND-OR logic gates. Mask PLA is programmed once and for all by the manufacturer.

Petrinet is a tool that provides a systematic, concise and easily comprehensible description of the sequential system shown in Fig.5.9. Live and safe *petrinets* that are not state machines are used to model systems presenting parallelism.

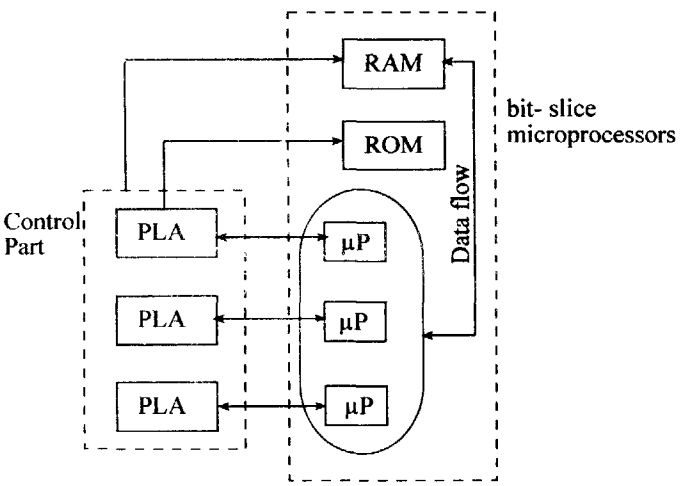



**Fig. 5.9** A structured system

## 5.4 Multiprocessing Epilogue

### Microprogramming

With the birth and growth of microprocessor technology, the microprogramming strategy started springing up for flexible systems. *Microprogramming* is a systematic and orderly approach of a CPU control unit introduced by Wilkes around 1950's. Each machine instruction is broken into microprograms. ROMs, RAM's and PLAs are utilized in microprogram designs. Parallel commands could be found in a microinstruction which consumes the instruction fastly.

Horizontal microprograms are of small lengths, easy to decode and results in lengthier microprograms. Vertical microprogramming calls for concurrency and process optimization methods but establishes simple coding schemes, whereas potential conflicts in bus transfers and in data dependent microperations must be handled with horizontal technique. Hardwired control units are faster at the expense of dedicated circuitry. Typical applications of microprogramming include emulation, high level language direct execution and tuning of architectures. Emulation is defined as the interpretation of an instruction set different from the native one. DEC VAX 11/780 has both a native instruction set and a PDP-11 emulated set. If the host is faster, the implementation of guest gets speeded up. EULER ( a subset of ALGOL 60) is a direct execution language for IBM 360/30. Direct execution is useful when the micro engine is flexible and the high level language has easier feature extraction. Static microprogramming more fits with mirroprocessor based process instrumentation systems.

Tuning of architectures lends oneself to dynamic microprogramming which has to account for instructions usage monitoring, performance efficiency and deciding to modify the native instruction set on evolving and strong processors.

Thus, microprogramming leads to essentially language implementations for problem - Oriented sectors. Also the single assignment rule of dataflow computations extends help for concurrency and parallelism with microoperations giving the job of resource conflict detection and process scheduling from the operating system viewpoint. The extreme step is high level language architectures for direct execution with bottom up machine design strategy. Stack architectures fit well with specific numerical control production process applications using LISP language and also for enhancing the arithmetic capacity of existing systems which otherwise are indicated by overflow errors.

CRAY - 1 is about two and a half times faster than that a Amdahl 470 V/6 ( in 1978) having a 29 nsec cycle time, owing to employing concurrent activation of twelve tightly coupled functional units by vertical microprogramming and pipelining. The I/O section consists of 24 channels, of which 12 are for output. The instruction formats are similar to CDC 6600 and this in addtion includes vector and backup registers presenting interacting challenges to compiler writers. The I/O devices are those tranducers which allow the outside world to communicate with processor memory. A to D and digital to analog converters are more desirable in process control instrumentation. R S 232 C is used with modems in local area networks for serial communication. The attractive candidates of Read only store ( ROS) are executable function call programs (math and graphic libraries) besides monitors and supervisors. For mass production of computers with ROS, thick film process in IC fabrication provides cost saving and increased reliability.

Ethernet is a standard local area network connecting terminals and computers within 100 meters to 10 kilometers range at a speed of 1 to 10 megabits per second. The topology is that of an unrooted tree. There is a single path between any two nodes of the network

which facilitates easy collision detection meaning a way for aborting transmissions. Error checking is met by *CRC* (Cyclic redundancy check) in which the choice of generator polynomial is crucial.

Current memories include error correction and detecting schemes. Local networks have provisions with hardware interfaces and software protocol to continue operating in the presence of errors towards fault tolerance. The low cost of introducing extraneous bits for error control makes MTBF of semiconductor memories improve by ten times. For example , the IBM 370 , CRAY - 1 and the DEC VAX 11/780 have 8 bit ECC's appended to 64 bit words for SEC - DED and detect 70% of multiple (larger than 2 bits) errors.

The electronic beam addressed memories are used for real-time slow switching. Random addressing of a bit ( pixel) is possible and block addressing is done by a TV - like Scan. Reading is partially destructive and refreshing is necessary every minute. These EBAMs promise to be fast compared to charge coupled devices and useful in extended core storage.

The Gorden research conferences, the American Chemical Society and many universities sponsor symposia on adhesion and adhesives. Thermal shock is an environment test performed to emphasize differences in expansion coefficicients for components of the packaging system in VLSI area. The 64 pin MC 68000 ( 16 bit microprocessor) is offered in a head spreader package (68000 G) . Besides holding the chip in place, the die bonding adhesives must conduct heat from the chip to the heat sink. All Organic *adhesives* release water vapour in hermetically sealed packages. The released water vapour must be less than 15000 ppm. to avoid chip deterioration. In todays life, failure of adhesives could cause computers to stop functioning, cities to blackout or missiles to misfire.

Having discussed on multiprocessing trends, we finally touch on the aspect of system reliability in section 5-5.

## 5.5 Fault Tolerance in Computers

The performance valuation of a computing system is called for when the system is not underutilized ; i.e., when the system cannot go insecure and when input efforts need not be increased or the output profitability shall not be impaired. In 1936, A.M. Turing has shown that most computers are logically equivalent to each other since they can compute the same wide class of computable functions, conditional only on the character sets they can accept and print and on the storage size.

System evaluation is of interest to architects, system programmers and users. Manual operations like tape-disk mounting, printer paper bursting and console interaction can have significant effects on overall performance. Thruput is a commonly used measure for

batch systems and is expressed in units of job done per minute. The lack of sensitivity to order of completion makes thruput less important for fast response interactive benches. Statistical and probabilistic methods will become important in computer systems as they serve more users in many unpredictable ways. Performance results, whether they are obtained by measurements , simulation or analysis are meaningful only with respect to the choices. Hardware monitors help in measuring resource performance metrics. The main attributes of microprogram speed improvement calls for reduction in instruction fetch times and more effective use of registers and fast storage units.

A few universities have taken the lead in producing fast-compile compilers. WATFOR Fortran compilers ( University of Waterloo, Canada) and PL/C, a PL/1 compiler at Cornell University support excellent diagnostic aids besides very fast compilation. Dan Ingalls and Don Knuth of standford University developed a software tool called execution time profile monitor with FORtran source input to study the program performance metrics. Automatic program generators from problem specification has definitely improved the programmer on line time to utilize the machine investments which are rather leased because of unaffordable cost and demand. REMAPT is a language for describing parts in numerical controlled machine tools. Execution profile analyzer helps system programmers to identify codes suitable for microprogramming. Multiplexed information and computing service has been successfully utilised in a predominantly timesharing environment at M I T.

*Fault-intolerance* is aimed in eliminating the sources of faults as much as possible whereas fault tolerance involves redundancy to provide a required level of service despite faults having occurred or being present. Reliability and availability dictate on the system dependability factor. High availability systems keep downtime to a minimum value and maintainability is affiliated to software engineering area.

Hardware redundancy is used as standby for continuous operations. Tandem computer is a system of high availability for commercial transactions. Triple modular redundancy is a concept for success voting on fault-tolerant instrumentation domain.

The software reliability is of major concern on the global markets. The effect of 25% increase in problem complexity results in a 100% increase in programming complexity. Software reliability is the probability that the software will execute for a particular period of time without a failure, weighted by the cost to the user of each failure encountered. Fault avoidance and tolerance are associated with software reliability. The Bell system's TSPs ( telephone traffic service position system) employs fault correction method and has a stringent availability requirement that downtime cannot exceed 2 hours in 40 years. Users must generate good algorithms and well defined data structrues to account for real concurrency with conventional languages. Blackbox approach of information hiding can be followed in software development to tackle the glassbox tests and improve safe communication skills.

Quantitative evalution of software quality proposed by *B.W. Boehm* in 1976 dictated 60 quality metrics. In 1977, Walters and MCcall reduced quality factors to 11 candidates. Accuracy in program error prediction is a major problem in quality control of a large-scale software system.

A *fault* is a damage, defect, or deviation from the normal state of the computing system on which tasks execute. Fault detection is concerned with detecting the manifestation of a fault by some means other than program execution, while error detection deals with detecting those errors in program execution induced by fault (s). Fault location is possible only by 100% error detection. Totally self-checking circuits are used to detect faults concurrently with normal operation. The purpose of error detecting codes is simply to detect the presence of errors whose non recognition could be harmful.

A code C is capable of detecting all unidirectional errors if the codewords are unordered. A *syndrome* is a binary word computed by the decoder and used in the decision as to which codeword was transmitted. Bose and Rao have shown that constant weight codes with minimum distance 2t + 2 are t – error correcting / all unidirectional error detecting codes. In general a distance K code will detect upto ( K – 1) errors. In future, code distances are expected to have a vital share in data communication.

The minimum distance of a code D(c) is the minimum hamming distance between two distinct codewords. Residue codes are well known arithmetic error detecting codes, while Berger codes are optimal systematic AUED codes. Error correcting codes can be classified according to the number of erroneous bits that can be corrected/detected. The error control schemes include ARQ (Automatic repeat request) and (forward error correction) FEC methods. ARQ scheme is preferred in data communication networks. A hybrid method combining ARQ and FEC will improve error control as well increase thruput for bus bottlenecks. Coding theory, currently a subject of research owes a practical floor in the frontiers of computer science towards reliable and secure machines.

To conclude, parallel processing issues are evolving around the revolving computer architectures in the problem spaces.

## KEYWORDS

Virtual memory management, dataflow, insecure, graph, network, algorithm, arithmetic pipeline, dataflow, local memory, sorting, MIMD, Illiac IV, Handler, latency, RISC, petrinets, microprogramming, CRC, adhesives, fault intolerance, Boehm, fault, syndrome.

## PROBLEMS

1. When is a graph said to be completely connected?
2. What is congestion? Explain how the congestion can be avoided in networks.
3. Compare and contrast synchronous and asynchronous buses from the viewpoints of data bandwidth, interface circuit cost and reliability.
4. When is serial communication called for? Explain the 8286 transceivers for half-duplex communications with an interface schematic.
5. Differentiate between errors, faults and failures.
6. Clearly distinguish error detection and repairability of a system fault.
7. Write notes on testability and fault coverage in VLSI designs.
8. What are static dataflow machines?
9. Explain anyone type of pipelining for concurrency with an objective approach.
10. How the SIMD configuration helps the ASIC's group?
11. What is multiprocessing on a MIMD machine?
12. Mention the merits of horizontal microprogramming.
13. Define fault detection.
14. Write an extensive notes on coding theory for secure systems.
15. Explain any project features featuring the parallel processing need of your own experience on an implementable task.

# CHAPTER - 6

# DESIGN METHODOLOGY AND EXAMPLE SYSTEMS

## 6.1 PARALLEL PROCESS FACTORS

Parallel processing can be attributed to the following factors:

### Pipelining

It is a technique of decomposing a sequential process into suboperations, with each subprocess being executed in a special dedicated segment that operates concurrently with all other segments (arithmetic pipelines and instruction pipeline). RISC uses an efficient instruction pipeline. Data dependency is tackled by compiler support on a RISC machine, for proper subtask scheduling.

### Vector processing

For scientific computing, the vector instructions capability has to be tapped with a proper way of pilelined approach (on a multiple processor environment) to respond for real-time applications. Speedup is more often achieved with super-computers.

### Array processors

This is achieved by deploying more functional units basically to support SIMD with pipelines besides meeting fault tolerance area with modular redundancy. Already *SIMD* benches offer good service as special attachments on general computers.

In multiprocessing environment, cache coherence problems arise because of the need to share writable data. In hardware solutions, the cache controller is specifically designed to monitor bus requests from all CPUs and IOPs, referred to as a snoopy cache controller.

Today the computing architectures are aimed at reliability, availability and serviceability for they have gained high degree of confidence limits in user applications. Some of the fault tolerant machines include ESS (Electronic Switching System), SIFT of NASA for commercial Aircrafts, system R of the IBM san Jose Research Laboratory for database management and the PLURIBUS of the ARPA (Advanced Research Projects Agency) computer network in real-time needs. The complexity of many software systems

have become unmanageable, and the natural consequences are delays and unreliability. The consequences of software errors and failures are normally more serious for on-line systems than in batch processing machines. Thus software must be consistent, robust and fail-safe to account for a fault-free computer.

*Spoofing* is an active endeavour in which the offender induces the system to provide the desired information while bugging is a passive activity in which the offender must await user communications and can only steal what the users transmit. Hardware reliability calls for good modular designs, testing for fault-coverage in the VLSI sphere, whereas the system reliability has to concentrate on software redundancy techniques like N-version programming, good module coupling and cohesions and a thorough self-diagnostics and a range of debugging tools to mean a real worker as an augmentive tool besides the ever improving knowledge power of the irreplaceable human experts on intelligent platforms. The present decade may concur with natural language processing and neural networking as architectural blends.

Parallel and multi processing appeared since the 1980's to innovate newer systems. The Cray-1 and Cyber - 205 use *Vectorizing compilers*. Examples of multiprocessor systems include Univac 1100/80, Fujitsu M382, the IBM 370/168 MP, and the Cray X-MP. A high degree of pipelining and multiprocessing is greatly emphasized in commercial supercomputers. In what follows, the principles of multiple processes and some example computer structures have been touched upon to depict the architectural trends well over the past two decades.

## 6.2 Parallel Processing Motives

Research and development of multiprocessor systems are aimed at improving throughput, *reliability*, flexibility and *availability*. The speedup that can be achieved by a parallel machine with n identical processors working concurrently on a single problem is at most n times faster than a single processor whereas thruput is more defined on a batch process. In practice, the actual speedup varies from $\log_2$ n to an upper bound n/ln n. More often the speed achievable is dictated by the chosen problem and the supportive algorithms. Computer systems have improved in differing phases like batch processing, multiprogramming, time-sharing and multiprocessing. Varying amounts of parallel processing mechanisms have captured the uniprocessor Von Neumann trends. The techniques may include multiplicity of functional units, parallelism and pipelining within the CPU, Overlapped CPU and I/O operations and balancing of subsystem bandwidths. Parallel processing applications in various spheres call for basic scientific research in the fields of data base *maintenance* systems to supercomputing targets.

**Memory Subsystems**

Block - structured programs yield a high degree of modularity as found with structured languages like pascal, C and Algol. The modules are compiled to produce machine codes in a logical space which may be loaded, linked and executed. The set of logically related contiguous data elements which are produced is commonly called as *segment*. Segments are allowed to have variable sizes unlike pages. The method to map virtual address into a physical address space and segments sharing is a critical design factor. *Burroughs B5500* uses segmentation techniques. Each system process has a Segment Table (ST), pointed to by a segment table base register (STBR) with the total process being active. The STBR permits the relocation of the ST, a segment itself, which is in main memory with the active programs. The *page-fault* rate function is minimised by dynamic dataflow computing, good memory allocation strategies and cache memory maps. Prefetching with swap facility meets time- sharing configuration.

Since terminals are relatively slow devices which often interact, serial data transmission assures reliable communication. *INTEL 8089* input output processor provides parallel data transfer used with microcomputers. Distributed data processes involve good amount of memory management on shared/sharable information. Multiprocessors are classified by the way their memory is organized. A multiprocessor system with common shared memory is called a shared-memory or tightly coupled multiprocessor. Loosely coupled or distributed memory systems have their own private local memories and prove fruitful when the interaction between tasks is minimum. Bus being a constraint, various inter-connection methods like time-sharing, multiport memory, crossbar switching and *hypercube* are used with loosely coupled processing elements.

### 6.3 Pipeline Mechanism and Machines

The efficient utilisation of processor elements during pipelining and the effective control on program parameters play a significant role in reflecting the response-times. The number of tasks that can be completed by a pipeline per unit time is called its throughput. The efficiency of a linear pipeline is measured by the percentage of busy time-space spans over the total time-space span, which equals the sum of all busy and idle time-space spans. An intelligent compiler must be developed to detect the *concurrency* among vector instructions so that parallelism can be installed which otherwise is lost by use of conventional languges. It is also desirable in CISC (Complex instruction set computer) to use high-level programming languages with rich parallel constructs on vector processors and multiprocess protocols. *Algorithms* do play a dominant role in process optimization.

The attached array processors include AP-120B (FPS - 164), and the IBM 3838. Vector computers include the early systems Star-100, TI-ASC having a speed of 30 to 50 million operations per second, and the further improved systems like Cray-1, Cyber-205 and

VP-200. The Cray-1 is not a "stand-alone" computer. A front -end host computer serves as the system manager. The computation section using a 12.5 ns clock period has a 64-bit wordlength and plenty of register storage. The memory section has interleaved banks each of 65,536 words and also incorporates the *SECDED* (single error correction and double error detection) facility for safe communications. The I/O section have 12 input and output channels also meeting priority resolvers.

The efficicncy of a computer depends heavily on the inherent parallelism in the programs it executes. The system designers and programmers share the responsibility of exploring the embedded parallelism on pipelined machines.

### Multiple Processors Programming

Multiprocessing supports concurrent operations among co-operative processes in spite of shared resources. The carnegie Mellon Cm, Cyber-170 and PDP-10 are some model examples of *multiprocessing* architecture. The *symmetric* systems facilitate error recovery in case of failure, by techniques like N-Version programming and backtracking. *Recoverability*, however, is not synonymous with reliability. The inherent redundancy on a multiprocessor improves the fault tolerant character of a system. The connectivity of a multiprocessor organisation is decided by whether the nodes are loosely or tightly coupled and the adjacency matrix. The time shared bus configuration assumes the least hardware cost and fit for uniprocesses. The *crossbar switch* means a tight connectivity and calls for reliability measures. Multiport memories allow distributed memory management.

The presence of private caches in a multiprocessor necessarily introduces cache coherence problems that may result in data inconsistency. Non homogeneous processing elements with differing functionality call for software resource managers in a multiprocessor system. Program control structures aid the programmer to write efficient parallel constructs. The high degree of concurrency in a multiprocessor can increase the complexity of fault-handling,especially in the recovery step. The use of transputers for scientific computing employ concurrency among the coordinating process elements. The data base distribution for a lightly-coupled multiprocessor environment calls for good system design besides an embedded physiques of the machine. Scaling up the activities and *speedup* are contrasting attributes to be met by effective algorithmics, of the respective machines.

### Multiprocess Control

With a high degree of concurrency in multiprocessors, *deadlocks* will arise when members of a group of processes which hold resoures are blocked indefinitely from access to resources held by other processes within the group. A *deadlock* is balanced by one or more of the following reasons, namely mutual exclusion, non preemption, awaiting and circular waits.

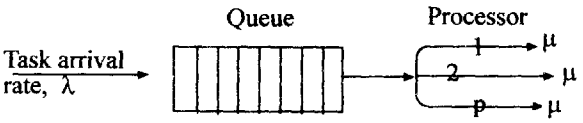



**Fig. 6.1** SIMD Queue Model

Consider a system consisting of P identical processing elements and a single infinite queue to which processes arrive. The mean processing time of processes on each processor is $1/\mu$ and the mean interarrival time of processes to the system is $1/\lambda$. Figure 6 -1 shows the model.

The utilization of the processors is $\rho = \frac{u}{p}$, where u is the traffic intensity given by

$$u = \frac{\lambda}{\mu}$$

The number of parallel tasks, the pipelength of each task and the vectorization factors play a dominant role on MIMD (multiple instruction stream multiple data stream) multiprocessing bench along with balanced system bandwidths and *compiler-compilers* for pottability to augment the parallel processing scenario.

## 6.4 VLSI ARCHITECTURE

The constraints of power dissipation , I/O pincount, long communication delays are prominent figures in VLSI. Properly designed parallel structures that need to communicate only with their nearest neighbours will gain the most from very-large-scale integration. The delay in crossing a chip on *polysilicon*, one of the three primary interconnect layers on an NMOS chip, can be 10 to 50 times the delay of an individual gate. *WSI (Wafer Scale Integrated)* implementation of highly parallel computing structures demand high yield on the wafer. The wafer is structured so that the presence of faulty Processing Units is masked off and only functional ones are used. Many practical problems of testing, routing around a faulty PE, power consumption, synchronization, and packaging are open problems even to date. Feature extraction and pattern classification are initial candidates for possible VLSI implementation. The eigen-vector approaches to feature

selection and Bayes Quadratic discriminant functions, must be realizable with VLSI hardware. *VLSI computing* fits with applications in image processing, syntactic pattern recognition, pictorial queries and database systems gaining reliability.

Packet switching networks for dataflow multiprocessors and *Wafer Scale integration* of the switch lattice are worth mentions of the parallel processing architectures.

## 6.5 Certain System Concepts

### Illiac IV System Concepts

The Illiac IV project was started in the Computer Science Department at the University of Illinois under the principles of parallel operation to achieve a computational rate of $10^9$ instructions/second. The system employed 256 processors operating simultaneously under a central control.

The logical design of Illiac IV is patterned after that of the *Solomon* computers. In this design, a single master control unit (CU) sends instructions to a sizable number of independent processing elements (PEs) and transmits addresses to individual memory units associated with these PEs. The design of Illiac IV contained four CUs, each of which controlled a 64- ALU array processor. Each PE has its own 2048 word 64-bit memory called a PE memory which can be accessed in no longer than 350 ns. The I/O is handled by a host B6500 computer system. The operating system inclusive of assemblers and compilers resides in B6500.

Illiac IV was indeed used as a network resource by the ARPA network system. The instruction set included a good number of arithmetic and logical functions. The B6500 supervises a $10^{12}$ -bit write -once read-only laser memory developed by the Precision Intrument Company. The beam from an argon laser records binary data by burning microscopic holes in a thin film of metal coated on a strip of polyester sheet, which is carried by a rotating drum. The read and record rate is four million bits per second. A projected use of this memory will allow the user to "dump" large quantities of programs and data into this storage medium for leisurely review at a later time; hard copy output can optionally be made from files within the laser memory.

### Super Computer Architecture Example

The Texas Instruments advanced scientific computer (*TI-ASC*) is a highly modular system offering a wide spectrum of computing power and configurability. It has also a peripheral processor and supports on-line bulk storage. A significant feature of the central processor hardware is an operand look-ahead capability. Data communications are controlled by a data concentrator which, in turn, interfaces to the memory control unit through a channel control device. TI-980, the data concentrator, is a general-purpose computer with upto 64K 16-bit words of memory and a one microsecond cycle time .

The operating system provides for buffering, reformating, routing, protocol handling, error control and recovery procedures and system control messages. Standard types of magnetic tape drives, card equipment, and printers are interfaced to the ASC. The memory control unit acting asynchronously is the speciality that can chase with technology race.

**Staran Array Processor**

The associative or content-addressed memory could acquire in a single memory access any data from memory. The STARAN S is an associative *array processor* consisting a symbolic assembler called APPLE and a set of supervisor, utility, debug, diagnostic and subroutine library program packages. Actual applications have been in real -time sensor related surveillance and control systems. In air traffic control application, a two-array STARAN S-500 was interfaced via leased telephone lines with the output of the FAA ARSR long range radar at Suitland, Maryland. Digitised radar and beacon reports for all air traffic within a 55 mile radius of philadelphia were transmitted to STARAN in real -time. A processing function for locating specific character strings (such as place names) in textual information was developed for STARAN that ran 100 times faster on a Sigma 5 conventional machine. The large scale data management architects are certainly the beneficiaries of the associative mapping.

The Goodyear Aerospace STARAN and the Parallel Element processing Ensemble (PEPE), built around an associative memory (AM) instead of using the conventional random access memory belong to the SIMD class. AM is content addressable, also allowing parallel access of multiple memory words. This allows for use in searching data bases besides the image processing and pattern recognition domains.

The processor utility and complexity is often dedicated to the control it supports. The simple one-by-one instruction execution to the wait/await forms of dataflow computing dictate the control architectures besides the other SIMD and *MIMD* benches.

**The MPP System**

In 1979, the massively parallel processor for image processing (*MPP*) was brought out by Goodyear Aerospace containing 16384 microprocessors to operate in parallel to process satellite images. Multiple -SIMD systems are dedicated for massively large arithmetic and logical processing. The high-level languges like Tranqual, Glypnir are extensions to Illiac -IV machine towards array processing.

One of the first projects aimed to dissipate the RISC (reduced instruction set computer) architecture advantages was conducted at the University of California, Berkeley. The Berkeley RISC I is 32-bit integrated circuit CPU that supports 32-bit address and either 8-, 16-, or 32-bit data. It has a total of 31 instructions and a 32-bit instruction format containing a 8-bit fixed length opcode. It has 138 registers. The strategy employed

is more hardwired control and a rich register set with compact addressing modes allows these machines to be fit for distributed parallel processing ends. Thus, multiprocessor scheme of this type shall allow desirable features on a server-client architecture in order to meet continuing thruputs on the continuous system operation.

In universal high-speed machines it is necessary to have a large -capacity fast-access main store. Though it is technically feasible to make the one -level memory hierarchy, it is more economical to provide a core store and drum combination with vitual memory support. High-speed computing splits the program store into four separate stacks and extracting many instructions in a single cycle despite the fast 2 μ sec. machine cycle time. The memory segmentation of the Intel processors is similar to the usage of multiple stacks.

**Burroughs' B 6500 Stack Mechanism**

Burroughs B6500/B7500 use the *stack architecture* mechanism to cope up with multiple processes and support languages ALGOL, COBOL, and FORTRAN. Some salient features of these systems are dynamic storage allocation, re-entrant programming, recursive procedure facilities, a tree structured stack organisation, memory protection and an efficient interrupt system. The software has to cope up with good book-keeping of the stacks used dynamically under program execution and meet the tangible events of data flow and instruction catches. The addressing environment of the program is maintained by hardware. The stack orientation of the supervisor at runtime thus establishes an opening for multiple processes composed of complex program modules to be run smoothly.

**VAX-11 Extension to DEC Family**

*VAX-11* is the Virtual Address extention of PDP - 11 architecture. It extends the 16-bit virtually addressable pattern of PDP - 11 to 32 bits, giving an address capacity of 4.3 gigabytes. The high level language compilers accept the same source languages as the equivalent PDP - 11 Compilers and execution of compiled programs gives the same results. It's environment is real -time operations with a Unibus tradition. Actually VAX - 11 is quite stack oriented, and although it is not optimally encoded for the purpose, can easily be used as a pure stack architecture if desired. It employs sixteen 32-bit general registers which are used for both fixed and floating point operands. Besides the program and stack pointers, the frame pointer (FP) and argument pointer (AP) need special mention. The VAX-11 procedure calling convention builds a data structure on the stack called a stack frame. FP contains the address of this structure. AP contains the address of the data structure depicting the argument list. The T bit of the PSW which when set forces a trap at the end of each instruction. This trap is useful for program debugging and analysis purposes. The software is supported by a good mix of addressing schemes. The virtual address space of 4.3 gigabytes supports timesharing jobs and priority scheduling. The CPU is a microprogrammed processor which implements the native and compatibility

mode instruction sets, the memory management, and the interrupt and exception mechanisms. Thus, the system fits to the class of *MAXI computers* of a uniprocessor type. Any mix of four Unibus or Massbus adaptors provide for attaching to peripheral buses that are not compatible with the VAX-11 /780 processor/memory.

### B 5000 Design Issues

A hardware-free notation was used (machine- independent language) in *B 5000* processor with symbol manipulative capabilities like ALGOL. Speed of compilation and program debugging were improved in order to reduce the problem time. Program syntax should permit an almost mechanical translation from source languages into efficient machine code as is the case with COBOL. The parallel processing can be facilitated for a multiprogramming domain. The character mode of usage allows list structures as employed in information processing with interpreter languages. Each program word contains 4 syllables equal to 48-bits of store. It follows a stack mechanism for addressing capacity for a reliable amount of data transfers. It employs program reference tables and polish notation to ease the objects code generation. B 5000 also uses nesting of subroutines and dynamic storage allocation policy.

### PDP-11 Evolution

Sets of computing systems exhibiting architectural similarity form a family. The evolving technology accounts much for the categorisation besides the program compatibility features. A family ususally is planned to span a wide cost and performance range which have to cope up with software Reuse, engineering and interchangeable peripherals The *PDP-11* of Digital Equipment Corporation, a minicomputer, is an evolved system for range The weaknesses like limited address ability, small number of working registers. nonprovision of stack facilities, and limited I/O processing were improved upon to build the PDP-11 general -purpose computer. The system is primarily intended for scientific computations because of addition of Six 64-bit registers for floating point arithmetic. It follows a Unibus architecture to accommodate the performance utility. A *read only memory* can be well used for reentrant codes but definitely unfit for an interactive I/O as used extensively in display processors and signal processing. Despite the deficiencies, the system could support a multiprogram environment meeting both COBOL and FORTRAN users. It operates under RSX - 11 M, a real time system for project groups. Thus a good amount of information science has gone into PDP systems range coverage. The machine manufacturers went in either to manage constant cost with increasing functionally or decrease the cost to maintain constant functionality on the growth path. PLAs ( Programmable Logic Arrays) and Microprogramming ( firmware concept) have been deployed actively in making the system stay.

### Cray - 1 System

A maxicomputer is a largest machine that can be built in a given technology at a given time. The *CRAY-1* supports fast floating point operations ( 20 to 50 million flops) and makes use of an optimised Fortran compiler for vector processing. The system is equipped with 12 i/o channels, 16 memory banks, 12 functional units and more than 4 K - bytes of register storage. Though the machine is of CISC category, the instructions are expressed in either one or two 16-bit parcels. The arithmetic logic instructions occupy a 7-bit opcode.

With a good capability of pipelining, the system indicates interrupt conditions by the use of a 9-bit flag register. The interrupt conditions are: normal exit, error exit, i/o interrupt , uncorrected memory error, program range error, operand range error, floating-point overflow, real-time clock interrupt, and console interrupt. Floating - point numbers is composed of a 49-bit signed magnitude fraction and a 15 - bit biased exponent while integer arithmetic is performed in 24-bit or 64-bit 2's complement form. The addressing scheme is highly flexible for array operations. The cray operating system is a batch type supporting 63 jobs in a multiprogramming environment. Cray Assembler language has the features of a powerful macro assembler, an overlay loader, a full range of utilities including a text editor, and some debug aids. Front-end computers can be attached to any of the i/o channels without affecting the CRAY's performance. Cache memory, dynamic microprogram mapping and optimum algorithms shall go a longer way in aiding aspiring *Cray users*.

### Symbol System

The SYMBOL language is directly implemented in hardware and thus uses less main memory for "System software". Also less virtual memory transfer time is needed. Source programs are special forms of string fields. They are variable length ASCII character strings with delimiters defining length and type. This is the outcome of a major developmental effort in increasing the hardware functional abilities (Memory management). The translator TR accepts SYMBOL and produces a reverse polish object string by means of direct hardware compilation. Fairchild built *SYMBOL computer* during 1964-70 that provided hardware support for interactive compilation. A study of a modern computer installation and its users as a total "system" reveals where and how the computing cost is divided. Consultants from IOWA State University did such a study. The objective of data processing is to solve problems where the "user with a problem" is the input and the "Answer" is the output. It is assumed that the user has his problem well defined and has the data available but the data is not yet programmed. The conversion of his problem to a computable language and the debugging necessary for correct execution is included in the total cost of operating an installation. The SYMBOL hardware has been engineered for good reliability and at the same time easy maintenance.

### IBM System/38

*IBM system/38* is an object - oriented machine for data base maintenance with major attributes like programming independence, authorization ability and dynamic microprogramming. This employs a lot of pointer types to smoothen the data - flow and permits user to declare the object rights. The processor and I/O units have access to the main storage and a multilevel, queue driven task control is achieved with microcodes. The system, in essense, supports modular programs with structured approach attracting the expert data base designers.

### Personal computing systems : Alto

Personal computers are often targeted at a particular application area, such as scientific calculation, education, business, or entertainment. By personal computing we mean a low-cost computer structure that is dedicated to a single user. The xerox Palo Alto Research Center ( PARC) *Alto* is a high performance personal computer by way of its 3 M bit/sec communication facility on Ethernet systems. The system supports 64 K words ( 1 word = 16 bits) of semiconductor memory. Applications include document production, interactive programming, animation, simulation and playing music. BCPL is a typeless implementation language that has much in comon with its well - known descendant, C, employed extensively to build Alto software. The *BCPL emulator* provides a vectored interrupt system implemented entirely in microcode, reason for servers being slowed down on the early Alto PCs. A design for a communication system must anticipate the need for standard communication protocols in addition to standards for the physical transmission media. The Alto was designed at a time when experience with protocols was limited.

*Microprogramming* is a form of emulation wherein one ISP (instruction set processor of a computer) is used to interpret a target ISP. Microprogramming permits an orderly approach to control design. The microprogram is easy to debug and maintain. It makes the control easy to check via coding techniques. (e.g. parity and Hamming code). One can implement complex ISPs through microprogramming. A two-way switch, controlled by a special flipflop called a conditional flip-flop, is inserted between control matrices towards construction of good micro-programmes for sensible instructions like repetitive loops more prominent in RISC machines besides multiuser operating systems.In a parallel computer with an asynchronous arithmetical unit every gate requires only one kind of control wave-form and the timing of that waveform is not critical. But for a serial machine of the von-neumann type requires many control signals which also include a decoding tree of even the external pinpoints that slows down the process.

## 6.6 Bit Sliced Microprocessor

An example of a microprogrammable system based on the AM 2910 sequencer and the Am 2901 bit slice ALUs is targeted to the *PDP-8 ISP*, the main feature being clarity of

implementation, while increasing the design complexity. The DEC PDP-11 employing the Unibus concept is a CISC type microprogrammed to align data and control paths using cache stores. With fast schottky TTL and a good pipeline provides the best parallelism to meet performance targets. A desk-top computer can be used as a simple calculator ( programmed machine) at any time during the entry or execution of a program. The wider display modes allow the generation of entirely arbitrary patterns on the CRT screen through the use of a *graphics raster*. The HP 9845 A having a high degree of physical integration provides CRT - thermal Printer dump. This was made possible by providing the ability to use the contents of the 16 K cache memory as a source of data to drive the internal thermal printer. That printer has a thermal printhead with 560 uniformly spaced Print resistors. The Graphics Dump produces a dot-for--dot image of the CRT's graphics mode display on the printer.

**Intel Microprocessors evolution**

Microcomputers have revolutionized design concepts in countless applications. The *VLSI* can meet the desirable properties like high gate- to - pin ratio, regular cell structure and high-volume applications. The 8080 of Intel had to use separate clock and controler chips to form a processing unit. With 8085 which sprung up in the year 1976, gave the least component count as well power requirement meeting many real-time interrupt applications controllable by RIM, SIM instructions. The 8086 chip in 1978 could provide for memory segments, varying register types and an addressing capacity of one mega bytes. The system based on 8086 CPU has a wider instruction set with nine flags slowly shifting over to the personal computer benches with evolutions in supporting a number of high level languages for portability. But a careful look at the Intel processors definitely convey the best abilities of the assembly languages to suit better for dedicated business processes and real-time industrial control applications. MOS technology is characterized by the parameters like propagation delay of gates, speed- power product measured in picojoules, gate density and cost.

*Electron beam lithography* will make possible the scaling down of structures to micron and submicron sizes. The influential areas for microcomputer software includes diagonostic tools, specialized logic analysers and hardware emulation tools. High level language support by microcomputer architects will make system programming efficient. The twenty first century is likely to see the birth of high-level language directly executable machines for user-friendliness and throughput increase.

## 6.7 Transputer Technology

As parallel processing has become affordable because of the availabaility of cost-effective processors, the need for parallel programming constructs have become inevitable. Modula-2 and *ADA* have added features to aid concurrency on a timeshared unibus computer architecture.

Transputer introduced in 1985 combines high speed computing with high speed communication with its own built-in on chip memory, floating point processor, timer and serial communication data links for direct communication to other transputers. OCCAM is a language specially designed for parallel processing using Transputers as processing elements. The small number of registers, together with the simplicity of the instruction set enables the processor to have relatively simple ( and fast) data paths and control logic. Stack addressing ( zero address) is followed implicity in covering the arithmetic and logical operations. The close mapping between the occam process model and the transputer architecture allows for an efficient implementation of concurrent parallel processes. The scheduler maintains different queues including priority processes. The possible scheduling points are defined by the time slice period and a set of jump instructions allowing a rescheduling.Standard Inmos Links provide full-duplex synchronised communication between transputers opening venues for high performance concurrent systems in a single processor system or network architectures. The communicating speeds range from 5 to 20 M bits/Sec. The features of a few *transputers* is shown in the following table , 6-1.

**Table 6-1**

| *Transputer No.* | *Size of Micro-computer* | *On-chip RAM* | *Maximum Throughput* |
|---|---|---|---|
| IMS T212 | 16 bit CMOS with four links | 2 K bytes | 10 MIPS |
| IMS T222 | 16-bit CMOS with four links | 4 K bytes | 20 MIPs |
| IMS T414 | 32-bit CMOS with four links | 2 k bytes | 10 MIPs |
| IMS T425 | 32-bit CMOS with graphics support | 4 K bytes | 30 MIPs (30 MHz clock) |
| IMS T800 | 32-bit CMOS with 64 bit floating point unit | 4 K bytes | 3.3 Mega flops at 30 MHz |

IN MOS T9000 is a 32-bit CMOS microprocessor having a 64-bit floating point unit, 16 k bytes of cache memory, a communication processor, 4 high bandwidth ( 20 M bytes/sec) full duplex serial communication links and 2 control links. Its off chip message routing is supported by low latency, high bandwidth, and dynamic routing switch IMS C104. TMS C40 is a digital signal processing parallel processor for real time applications. It has a 32 bit processor with 40-50 n sec. instruction cycle and throughput of 320 M bytes/sec. It also supports on chip DMA for 6 channels with external /internal clocking.

The major applications of transputer technology include super-computing, process control and image processing.

An occam program consists of a number of parallel processes, which can be run by a single processor or if fast execution is required, the program can be run by several processors,. *Occam* processes are built from three primitive events, namely, assignment, input and output.

Examples:

```
X: =y + z
Channel ? Var
adc ? temp , where
variable "temp" is input from channel "adc1."
channel 1 var
dac? cont - Var
where variable "cont - var" is output to
channel 'dac1".
```

The primitive processes are combined to form constructs like SEQ, PAR, ALT and ALT with priority. The control structures include WHILE and IF statements. PAR avoids deadlock by allowing concurrency. Every variable and channel that are used must be declared with a data type.

The configuration description must include each processor in the network, the type of the processor, and the link interconnection between the processors. The processor T800 with four bidirectional links can be associated with 8 unidirectional channels. Multiplexers and demultiplexers play a key role in data acquisition softwares.

Parallel C is another popular concurrent programming language for a transputer system. It provides mechanisms to dynamically create new concurrent threads of execution within a task. Each thread has its own stack allocated by its creator, but shares its code, static data and heap space with other threads in the same task. Semaphore functions in the run time library are provided to control access to shared data and channels. This is relatively weak in compile time checks as compared to OCCAM. The transputer network can be carefully designed towards SIMD algorithms and still crucial MIMD image processing domains. PARAM based on T-800 is a parallel processing system developed at C DAC , Pune.

The advanced plant automation and control system ( APACS) developed by Electronics Research and development corporation, Trivandrum is based on transputers. The prototpe model has undergone field trial at RCF Bombay in the ammonium Bicarbonate plant. The results show an increase in production with quality improvement and reduction in plant shutdown time. Certain applications such as flight testing require high speed acquisition of data (about 2000 I./O s every 10 m sec. is communicated).

Xerox corporation's *Ethernet* is universally adopted as a defacto standard for LANs. Today, most major computer networks( e.g. DEC net, HP's Advance Net, and Novell) are commonly based on IEEE 802.3 ethernet standards. While the physical layer provides only a raw bit stream service, the data link layer attempts to make the physical link reliable and provides the means to activate, maintain and deactivate the link. It also provides services like error detection and control to the higher layers.

Transport layer and Application layer are exploited by implementing typical software routines that perform the network functions like File transfer, chat, network management and error management. The two competing protocols for Ethernet are TCP/IP and DECNET. Since virtually all UNIX systems support TCP/IP it has been chosen to implement the TCP/IP at the transputer end making communication with X windows possible.

The transputer represents a novel approach to designing VLSI microprocessor systems. Inmos principal goal is to provide a family of high performance programmable VLSI components containing high bandwidth communication links for the creation of concurrent multiprocessor systems. IMS T424 has a simple *microcoded CPU* and 2 k bytes of on chip memory. It has only six registers. The A.B and C registers form an evaluation stack. The workspace pointer points to local variables at run time. The next instruction pointer is similar to program counter. The operand register is used to form instruction operands. It has only one instruction format being 1 byte long. The system designer is still posed with the decisions concerning which processes are most effective when operating concurrently, how concurrent processes should be distributed over transputers and when those processes should communicate with one another.

### 6.8 Super computers

The advances in technology by way of better devices offer only a low speedup. *Gallium arsenide* is employed in Super computers. A fault tolerant system halts an entire pipeline even if one pipesegment fails. The expected clock of 2 nano second period will aid supercomputers.FORtran fits well as a language offering good vecrotization ratio, besides being supported by efficient translators.

The Amdahl's law states,

> *When a computer has two distinct modes of operation, a high speed mode and a low speed mode ( example, MAX/MIN of 8086 CPU ), the overall speed is dominated by the low speed mode.*

A 3D hypercube contains 8 compute elements ( $CE_s$). A general hypercube structure of dimension d has $n = 2^d$ processors and the number of links is n*d/2. The hypercube interconnection of CEs is employed with commercial multiprocessing and has a maximum distance equal to d. Convex is a shared memory multiprocessor made by Convex Computer Corporation, U.S.A. CDAC's PARAM comprises of 256 $CE_s$ each with 4 Mbytes of main memory. This is a message passing multicomputer system under PARAS environment. The augmentive tools are used in process management, debugging and profiling. Benchmarks are becoming available today to evaluate supercomputers performance.

Ab-initio methods are used to study chemical properties of various clinically active drug molecules, where experimentation is difficult. Cyclophosphomide, one of the most widely used anti cancer and immuno - suppressive agents acts on a wide variety of tumours and lukemias. Typical *matrices* in computational chemistry are *sparse* with about 1% non zero elements. The algorithms can tremendously speed up the turn around-time in this case.

## 6.9 Operating System Review

An operating system makes good sense in acting as senses of the hardware constituting the Physical Machine. The routine tasks of the Operating System include job setup and change overs, context switching and time sharing by jobs, batch processes manuring towards good thruputs.

The division of jobs like processor, memory and device management by the Operating Systems ( O.S) was primarily aimed at improving the system performance besides the distributed data maintenance aspects and constructing of good editors. The interruptions during any process grows higher from a unibus to Multibus architectures and the interruption class dictates the salient features of the respective O.S. based on the end application. These interactive bases on a distributed process are expected to cater to the reliability and maintenance requirements. The PC-DOS is an example for single user O.S. while batch systems cater to voluminous inputs on a Queue basis. Presently, the real-time operating systems grow in size and complexity for the on-line user category, wherein the dual needs of accuracy and speed have to be met. UniX is one example of time sharing multi-user environment of a unibus category.

The *hard disk* device gave the impetus for a self-resident monitor with large supervisory support and an opening for file management systems in todays information technology race. Job-scheduling and device management are primary concerns on server client modelling, by the clever use of dispatchers. Memory management protocols become a challenging sphere from scientific computing at one end to physical data base crashes in the knowledge bases. Buffering is a continuous activity for all Input-Output activities, starting from the use of shell command to a big real-time process like a robot with programmable automation benches. The real-time distributed operating system ( RT DOS) has been developed to provide a hierarchical set of functions that provide applications processes with system management services, control and network services for a wide range of real-time processing requirments.

## 6.10 Parallel & distributed Computing

The problems involving array processors belong to the SIMD category of parallel processes ( example being computer synthesis of hardware design). The shared memory in message passing tasks has to accommodate structured data flow and lesser pipelengths to cope up parallel computing.

The Vector Processor *Intel 80860* having a 64 -bit word length uses RISC topology with onchip cache memory on UNIX operating system and is fit for floating point arithmetic speed up. The performance evaluation of parallel computing is a research domain that has to account for evolution of parallel program constructs, optimization with structural systems ( more application-specific) and the availability of distributed operating systems in the real MIMD category. Distributed computer systems represent an evolutionary path from early operating systems. Resource networks such as arpanet is a begining to distributed computing. The mainframes would operate in usual fashion and link into the network to send or receive messages based on usage. The transmission lines were leased lines of high speed and controlled by the network interface control processors. Routing and security aspects add strength in computer communications. Reliable transmissions involve CRC checks, coding and retransmission schemes. The evolution of languages for distributed processes is one of object-orientation.

### 6.11 Systems software

The term firmware is used for the software stored in read only storage devices. Utility programs can serve in their capacity for program development, debugging and documentation. *Linkers* with good relocation abilities and fast loaders are desirable features of the computing hardware. If a compiler runs on a computer for which it produces the object code, then it is known as a self or resident compiler. The tools towards debugging are simulators, logic analyzers and trace routines with memory dump facility. Ada is used in the control of multiple concurrent processes and embeds the best features of pascal, Algol and PL/1.

An assembler produces object codes for programs supplied in assembly lanaguage of a machine. A two-pass assembler collects all labels during the first pass and assigns addresses to the labels counting their positions from the starting address. On the second pass the assembler produces the machine code for each instruction and assigns address to each. Certain directives used under assembler environment is discussed in the next paragraph. Standard directives are commonly used to define constants, data areas, and reserve blocks of memory.

**Examples:**

**Label**: EQU expression. At assembly time the expression is evaluated and assigned to label and an entry to that effect is made in the symbol table. A *label* can be defined by an EQU directive only once in a program.

**Date**: SET OA(H) Constants are defined by the SET directive and can be changed anywhere to another value by a reassignment.

ORG 0100 (H) The first byte of object code will be assigned to 0100 (H) and all succeeding bytes will be ocupied by higher addresses until another ORG statement is encountered. The END directive makes the assembler to terminate the present pass.

**Message**: DB 'This is a message' is a valid string definition, DS is used to reserve data storage locations as specified in the define storage directive. Conditional Assembly is allowed by the IF and ENDIF words. EPROM programmers are available on dedicated systems and an MDS ( Microcomputer development system) costs more, especially, to support high level conventional languages with a rich and complex debugging and diagnostic softwares. Programs written with HLL (High level languages) are easy to maintain, portable and often occupy more memory for execution. Pascal and LISP serve as interpretable language for machine tool in sophisticated electronic systems.

## 6.12 Software Reliability

The complexity of many software systems has become unmanageable, and the natural consequences are delays and unreliability. A program must react to errors. It must indicate regarding errors or malfunctions without shutting the entire system down. Software correctness is concerned with the consistency of a program and its specification . A program is correct, if it meets its specification; otherwise it is incorrect. When considering correctness it is not asked whether the specification corresponds to the intentions of the user. A program that can be extended to tasks other than for which it has been designed and developed is a better approach in programming houses. In software development, the programming time is given a major weightage for it has to meet the essential constraints of speed, reliability and ease of use.

Reliability is ( at least) a binary relation in the product space of software and users, possibly a ternary one in the space of software and users and time. System reliability must take into account the influence of hardware errors and incorrect inputs. A software system is said to be *robust* when the consequences of an input or hardware error related to a given application are inversely proportional to the probability of the appearance of that error in that application. The cost to user "c" is a measure of the error effect.

$C \simeq 1/p$, where p is error probability. Errors which are expected to occur frequently should only have a minor effect on the given application . A list of critical error effects and of probable error causes must be available . It is a better practice of recording critical as well as noncritical errors towards performance valuation of systems.

Reconfiguration is followed with the occurrence of permanent errors. Though software testing helps out in combating permanent faults, the elimination of other errors by software means may not be satisfactory due to prohibition of testing costs for quality pass. Normally permanent faults and failures are fit enough for repeatability and only transient errors are difficult to catch. *EXDAMS* ( Extendable Debugging and Monitoring system) meets the twin requirements of repeatability and observability useful with real-time systems.

On - line debuggers provide the programmers ample facility to control and observe program flows. The facilities include setting and deleting of break- points, error traps, interrupts simulation and performance reports generation. Any modification that is required to be performed on a system after acceptance to enhance a particular software status can be called software maintenance. The supervision effort required in determining development progress can be comparable with the development effort.

## 6.13 Text Processes

The aims of a text processing system is to organise and control large volumes of text so that they can be used for the efficient communication of information. A text processing system consists of a number of text sources, a means of storing text, a means of extracting information from the stored text and a group of users.

It is likely that in time every business will find a need for a computerised text processing system, just as now it has need for a typewriter. Privacy and security are essential ingredients of a text processor usage. The functions of text data processing involves document retrieval, data retrieval and question answering. Text analysis of differing kinds will be pronounced as the associated objectives. Because of the voluminous amount of data growth, the following factors are worth to be tackled by a text processor namely, optimisation enhancement and innovation.

The quality of documentation is a good indicator to the quality of the software. Good documentation should be precise, to the point, and complete. Perhaps, good management towards reliable software throughout the product life-cycle shall make the package easily marketable and hence reach standards. Line, string and cursor oriented editors assist for software developmental activities. With the present day keyboards, a large number of keys are added on to increase and improve the editing functions available of a system.

A software error is present when the software does not meet the end- user's reasonable expectations. The best way to dramatically reduce software costs is to reduce the maintenance and testing costs. Fault avoidance is concerned with detection and removal of errors at each translation step.

The concept of fault detection are accepted, as on date, for application software projects. The *PRIME system* is a Virtual memory multiprocessing system developed at the University of California at Berkeley wherein data security is a primary goal. From the eyes of language designers, the language must not encourage obscurity and in effect, decreases the programmer's opportunities to make mistakes.

Assertion statements at runtime help for debugging as well as modification. The simon monitor is a production tool developed by Mitre Corporation used during the

programming process. The one language formula for a development team is important for portability at one end and environment preferences at the other extreme. Language compilers must provide exhaustive checks for syntax errors and reduce semantic gap. The most common method of proving programs is the informal method of inductive assertions; proofs on non-numerical programs are much more difficult than proofs of numerical programs. The difficulty of the proof is proportional to the richness and complexity of the programming language. Computer architecture has an indirect yet important effect on software reliability for the hardware reliability is often ascertained with the help of software programs. Explicit program specification by self-documentation ensures avoidance of program misinterpretation.

## 6.14 Data Communication

The word Robot was first used in 1921 by Karel capek, czech playwright, novelist and essayist. The recovery from errors, which is fairly straightforward on basic data communication links becomes highly complex in a distributed processing environment. The use of distributed processes typically permits a simpler program structure and more computational power in most applications. Error correcting codes are sometimes used for data transmission especially when the channel is simplex. Security and privacy govern the performance of a network. Some of the Bus structures is listed in Table . 6-2.

Table 6-2

| *Type* | *Originator* |
|---|---|
| IBM PC Bus | IBM |
| Multibus | Intel |
| VME Bus | Motorola |
| I EEE 488 Bus | Hewlett-Packard |
| Q- Bus ( Unibus) | Digital Equipment Corporation. |

It is worth mention of the data rate for channels thanks to Nyquist and Shannon. H. Nyquist, in 1924, derived an equation expressing the maximum data rate for a finite bandwidth noiseless channel.

If the signal consists of V discrete levels, *Nyquist's* theorem states maximum data rate = 2 H $\log_2$ V bits/sec. where'H' is the bandwidth. For example, a noiseless 3 KHz channel cannot transmit binary signals at a rate exceeding 6000 bits/sec.

According to *Claude Shannon* ( 1948) , the maximum data rate of a noisy channel whose bandwidth is H Hertz and signal to noise ratio is S/N, is given by

$$\text{Maximum number of bits/sec} = H \log_2(1 + S/N)$$

If H = 3000 Hertz at 30 dB, then speed $\leq$30,000 bits/sec.

## 6.15 Epilogue

Both the methods of hardwired and microprogrammed control employ counters, sequencing registers and combinational logic in generating the control configuration on a computing system. The hardwired approach has been identified for high speed scientific computing and reliable networks. Delays can be well managed with this topology.

The microprogramming splits each macroinstruction to well-defined control operations with good design from the microprogrammers. Microinstruction lengths can vary depending on concurrency in microoperations, the coding of control data and microinstructions sequencing. Multiplexers are often employed in steering the micro - control fields besides governing the functional attribute in SIMD(Single instruction multiple data) as well pipelining controls.

The CPU pins also dictate quite a good amount of control complexity for interactive on-line as well batch computers. The system commands operated at user nodes activate the interface hardware units to a greater extent and also they occupy a fair portion of memory for directly executable architecture based on package environments.

File transfers often utilise the direct memory access mechanisms and network servers must deploy more number of interrupt control pins on the mother-board in improving the control complexity. Whereas good processor resource management is an embedded property of RISC work stations, the cache memories add weightage to account for retaining the system speeds. Redundancy approaches by standby or other means in hardware helps in fault detection and reduces system failure.

## 6.16 Future Directives

The architectures can be language-based but not biased on multiprogramming domains for multiuser environments. Front end translators must cope up to support many a programming language for systems design and distributed processing. Faster servers and secure data communications shall pave the way on business data processing application. A huge amount of cache partitions and user memory segments with a good job scheduling policy will go a long way to assist both on-line and off-line users and optimise user resources on multiuser scenario. Program measurement tools and good diagonostic debugger shall increase the productivity in meeting standards ( ISO 9000's). The effective fault tolerance measures continuously evolve in the software scenario to make machines perform better. Future *compilers* has to face the challenges like layout of data to reduce memory hierarchy and communications overhead and exploitation of parallelism. Ignorance of I/O will lead to wasted performance as CPU's get faster. The future demands for I/O include better aolgorithms and organisations and more caching in a struggle to keep pace. Latency and throughput never go together.

## 6.17 Fault-Tolerance features

Earlier fault - tolerant computers were limited to military aerospace and telephony applications where system failures meant significant economic impact and loss of life. But today with the increased momentum in all areas, fault-tolerant techniques provide relief and a longer service under harsh usage and environment .Toleration of transient faults is felt essential on microprocessor based instrumentation systems. Permanent hardware faults can never be tolerated for on-line real time use applications. Large mainframe manufacturers like Amdahl and IBM use redundancy to improve user reliability as well assisting field service personnel in fault isolation; Minicomputers have gone in for Hamming error - correcting code on memory and special LSI chips like cyclic redundancy code encoder/decoders. The effect of defects can be overcome through the use of temporal redundancy (Repeated calculations) or spatial redundancy ( Extra hardware or software). Error detection is a specific technique employed to cover hardware faults even at user level predominant for business data processes.

The range of fault tolerance techniques adopted vary in response to the different reliability requirements imposed on different systems. Co-operation between humans and computer system takes on differing forms, depending on the extent to which the computer's software is sophisticated enough to be able to perform what we might call "Cognitive processing". Factors leading to economics of scale for computers often include several dimensions. The same software can be used on many models. Sales and maintenance personnel can service a wide range of equipment. Flexible manufacturing automation leads to adaptability.

In commercial aircraft, computers are usually used to provide a variety of services such as navigation, semiautomatic landings, flight control and stability augmentation. The crew are always available to provide manual backup if computer failure occurs. Software implemented fault tolerance (*SIFT*) machine employs hardware redundancy for critical tasks. Executive software is responsible for implementing the NMR ( Modular redundancy) structure; since design faults in the executive may lead to system failures, SIFT designers intend to give a rigorous mathematical proof of its correctness. Some of the fault-tolerant systems are brought out and discussed of their specific features.

## 6.18 System R

System R is an experimental relational database management system designed and implemented at the IBM san Jose Research Laboratory. Relational data interface is provided for high level features in data retrieval, manipulation and definition as well as mechanisms to assist in the provision of fault tolerance at the user level. It is expected that the information entrusted to the system is not lost or corrupted. Thus, fault tolerance has been adopted as a means of providing reliable storage in many data-base systems, particularly for the purpose of providing tolerance to failures of the hardware units on

which the data are stored, and against system stoppages which could leave the data-base in an inconsistent state.

## 6.19 Example Systems

### 6-19-1 The Star Computer

The presented work is a continuous investigation of faualt-tolerant computing conducted at the jet propusion laboratory during the period 1961-70. The star ( Self-Testing and repairing computer) system employs balanced mixture of coding, monitoring, standby redundancy, replication with voting, component redundancy and repetition in order to attain hardware controlled self-repair and protection against all types of faults. The standard computer is supplemented with one or more spares of each subsystem. The spares are put into operation to recover from permanent faults. The principal methods of error detection and recovery are as follows.

All machine words ( data and instructions) are encoded in error detecting codes and fault detection occurs concurrently with the execution of the programs. The arithmetic logic unit function is subdivided by replaceable functional units with more of a centralized control that allows simple fault location procedures. Special hardware takes care of error recovery besides software augmenting the memory damage. Program segment repetition helps out transient fault recovery. Monitoring circuits help error detection for synchronization in error control for data communications.

The CPU overall is divided into various processing ensembles providing a good pipeline design. Logical faults result either in word error or control errors. The opcode formation employs 2-out-of-4 code. The Test and Repair processor (*TARP*) monitors the STAR operation. The replacement of faulty functional units is commanded by the TARP vote and is implemented by power switching and strong isolation is provided for catastrophic failures.

The interested reader may refer to Avizienis et. al ( 1971). SCAP ( the Star computer assembly program) is the first module of STAR software. SCAL is the assembly language used. The software comprises of an assembler, a loader and a functional simulator. The wordlength is 32 bits . Data sharing with processes of a read-write capacity allows automatic maintenance information from the space craft telemetry system.

### 6-19-2Electronic Switching Systems (ESS)

The stored program control of Bell systems has been developed since 1953 for varying exchange capacities. The best hardware is deployed for switching circuits and a duplicate ( image) processor runs side by side serving as a checker as well a standby to avoid severe downtimes. It is approximately no more than few minutes in a year.

The reliability in software &hardware is achieved by redundant packages. Automatic error correction for data routing during communication alleviates channel noise recovery problems. The trained maintenance personnel shall be an asset at main distribution frames ( exchange side) for on-line maintenance without affecting the on-line service. Good documentation directions and concise manuals shall help the maintenance staff for the DO's so that procedureal errors can be minimised.

The system Architecture is shown in Fig. 6-2.

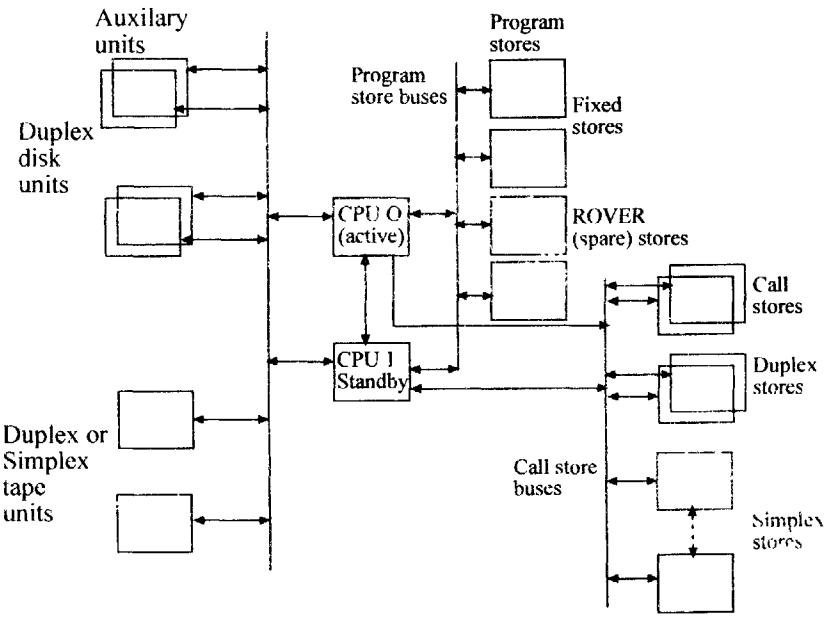



**Fig. 6.2** ESS 1 A System organisation

The reliability and availability requirements for the ESS systems are extremely stringent: downtime of the total system is not supposed to exceed 2 hours over a 40 year period, with not more than 0.02% of calls being handled incorrectly. Continuous commercial operation commenced in 1965 with the No. 1 ESS, a system supporting large telephone offices of upto 65000 lines and capable of handling 100,000 calls per hour. The processor being out of service cannot exceed 2 minutes per year, which operates continuously.

Extensive redundancy for hardware faults is called for. Buses duplicated. CPUS duplicated. The timings (CONTROL) operate synchronously for both CPUs. The information necessary for processing and routing telephone calls is held in the call store complex which is also constructed from core stores.

The random access memory in the system is divided into protected and unprotected areas. This division is enforced by the active CPU , which contains the mapping registers defining these areas, and which provides different instructions to write to the different areas. The mapping registers are software controlled via other ( special ) write instructions. The protected area contains the part of the memory that are not duplicated ( e.g the program stores)., as well as those locations which affect the operation of the system (e.g. the internal registers of the CPUs).

All of the auxiliary units operate autonomously, competing with the CPUs for access to the core-stores. The call processing programs are divided into two classes, deferable and non-deferable. The deferable programs are those for which data is already in the system , and these programs are not therefore critically time dependent. The non-deferrable programs (eg. input/output) are those which must be executed according to a strict schedule and are activated by a clock interrupt. For example, the program that detects and receives dial pulses has to be run at regular intervals.

The program store ( PS) is read only memory ( ROM) containing the call processing, maintenance, and administration programs besides the long-term translation and system parameters. The call store contains the transient data related to telephone calls in progress.

The Mean-Time - To-Failure (*MTTF*) a measure of availability, is

MTTF = $\mu/2\lambda^2$

Where $\mu$ is the repair rate and $\lambda$ is the failure rate.

System downtime is madeup of:

| | |
|---|---|
| Hardware unreliability | - 0.4 Minutes per year |
| Software unreliability | - 0.3 Minutes per year |
| Procedural faults | - 0.6 Minutes per year |
| Fault tolerance deficiencies | - 0.7 Minutes per year. |

Procedual faults arise from manual interactions with the operation of the ESS.

Tolerance of hardware faults is achieved by a combination of hardware and software techniques. In the main, error detection is implemented in hardware ( eg. operation of the active CPU is checked against that of the stand-by CPU by matching circuits) with exceptions being signalled by means of an interrupt mechanism to invoke the fault treatment programs which form a major component of the fault tolerance techniques. Fault tolerance deficiencies, which are expected to be the major cause of system down - time, concern the deficiencies in these programs. Tolerance of software faults is limited to attempts at maintaining the consistency of the data base. The programs that perform these error detection and recovery actions are referred to as the audit programs. Today, microprocessor based (EPABX) switching for office and intra-inter communications is picking up with architectural blends.

### 6-19-3The Tandem 16

A fault-tolerant computing system:

The increasing need for businesses to go on-line is stimulating a requirement for cost effective computer systems having continuous availability as in data base transaction processing. A power supply failure in the I/O bus switch or a single integrated circuit(IC) package failure in any I/O controller on the I/O channel emanating from the I/O bus switch will cause the entire system to fail. The hardware expansion are typically dwarfed in magnitude by the software changes needed when applications are to be geographically changed. The Tandem 16 uses dual-ported I/O controllers with a Dynabus and a DC power distribution system. The on-line maintenance aspects were key factors in the design of the physical packaging of the system.

The processor includes a 16-bit CPU, main memory, the I/O channel and control, employing schottky TTL circuitry. The CPU is a microprogrammed processor consisting of a bank of 8 registers which can be utilised as general purpose registers, as a LIFO register stack, or for indexing - and an ALU ( arithmetic logic unit.) It also has CPU stack management registers and scratch pad stores and miscellaneous flags for the use of the microprogrammer.

Microprograms are organised in 512-word sectors of 32-bits in Read only memories. The address space for the microprogram is 2K words. The microprocessor has a 100 ns cycle time and is a two stage pipelined microprocessor. The software has 123 machine instructions each having a length of 2 bytes. Interrupt levels of upto 16 are provided with the I/O system. Main memory is organised in physical pages of 1 K words of 16 bits/ word upto 256k words may be attached to a processor. This includes additionally 6 check bits/word to provide single error correction and double error detection. Memory is logically divided into 4 address spaces each of 64k words. The lowest level language provided on the Tandem 16 system is T/TaL, a high-level , block-structured, ALGOL-like language which provides structures to get at the more efficient machine instructions. The basic program unit is the PROCEDURE.

Each process in the system has unique identifier or "processid" in the form: < cpu #, process # >,which allows it to be referenced on a system-wide basis. This leads to the next abstraction, the message system, # which provides a processor-independent. failure-tolerant method for interprocess commmunication. A memory management process residing in each processor where pages are brought in on a demand basis and pages to overlay are selected on a "least recently used" basis.

The heart of the tandem 16 I/O system is the I/O channel. All I/O is done on a direct memory access basis. The greatest fear that an on-line system user has is that "the data base is down". To meet this critical phase, Tandem provides automatic mirroring of

data bases. The disc controller uses a *Fire code* for burst error control. It can correct 11 bit bursts in the controller's buffer before transmission to the channel. The Tandem 16 power is met by a 5V interruptable section, a 5 volt uninterruptable section and a 12-15 volt uninterruptable section. The power supply provides over voltage, over current, and over temperature protection. The system provides on-line maintenance.The operating system *Guardian*, provides a failure - tolerant system. As many as sixteen processors, each with upto 512 K bytes of memory, may be connected into one system. Each processor may also have upto 256 I/O devices connected to it. Error detection is provided on all communication paths and error correction is provided within each processor's memory. Systems with between two and ten procesors, have been installed and are running on-line applications.

**6-19-4PLURIBUS**

Is a multiprocessor system designed to serve as an interface message processor (IMP) of the Advanced Research projects Agency (*ARPA*) computer *network*. Essentially, the IMPs act as communications processor, providing a message transmission facility between nodes of the network implemented as a store-and-forward packet switching system. They also perform packet receipt, routing and retransmission as well as connecting host systems to the rest of the network.

FAULT TOLERANCE: if a sending IMP does not receive an acknowledgement after the transmission of a packet, then retransmission Via another IMP and communications line ( if one exists) can be attempted. Thus the reliability requirements for an IMP place emphasis on availability , rather than fault-free operation. Occasional losses of a packet or message, or short periods of down-time, are considered to be acceptable.

The pluribus system places the major responsibility for recovery from failures on the software. The system may be characterized as a symmetric, tightly coupled multiprocessor and is highly modular. A processor failure merely causes it to run a little slower. In the pluribus, the first detection of a fault is usually through failure of an embedded check in the main program, and frequently that is all that is required to initiate a correct recovery procedure.

The software reliability mechanisms for a pluribus system are coordinated by a small operating system called *STAGE* which performs the management of the system configuration and the recovery functions. The overall aim is to maintain a current map of the available hardware and software resources. The tools and diagnostics are well enough defined and documented for repair purposes. Some application architectures to pluribus include message systems, real-time signal processing, reservation systems employing time shared configuration and process control.

## 6.20 Conclusion

The future systems are more tuned for software support environment for dedicated applications imparting also portable programs. Automatic program generation, performance valuation programs and monitors give added strength for productivity. The new emerging topics of artificial intelligence, neural networks and expert systems have to await the discovery of fresh and biased languages besides good graphic tools to take on a different scenario. Thus reliable design of systems shall pave the way to good productivity on software workhouses.

## KEYWORDS

SIMD, Spoofing, Vectorizing compilers, reliability, availability, maintenance, segment, Burroughs B 5500, page-fault, Intel 8089, hypercube, concurrency, algorithms, SECDED, multiprocessing, symmetric, recoverability, crossbar switch, speedup, deadlock, compiler-compilers, polysilicon, WSI (Wafer scale integration).

VLSI computing; packet switching networks, Solomon, TI-ASC, array processor, MIMD, MPP, stack architecture, VAX-11, MAXI computer, B 5000, PDP-11, read only memory, CRAY-1, Cray users, symbol computer, IBM system/38, Alto, BCPL emulator, Microprogramming, PDP-8 ISP, graphics raster, VLSI, electron beam lithography, ADA, transputers, Occam, Ethernet, microcoded CPU, Gallium Arsenide, Sparse matrix, hard disk, Intel 80860, linkers, label, robust, EXDAMS, Prime system, Nyquist, Claude Shannon, compiler, SIFT, TARP, MTTF, fire code, guardian, ARPA network, stage.

## PROBLEMS

1. Write about differing factors that assist parallel processing.
2. Define multi processing giving the context under which the above term is pronounced.
3. Give the constructional features of MPP for image processing.
4. Write on the evolutions of Digital Equipment Corporation's PDP series of minicomputers.
5. Write a detailed note on Intel's microprocessors evolution.
6. Differentiate bit and byte organised memories for system reliability. Discuss the salient features of CRAY machines.

7. Explain clearly how transputers contribute to multiprocessing environment.
8. State Amdahl's law.
9. What are the merits of data compression in data communications? Explain the Huffman coding scheme in achieving the same.
10. Write short notes on SIFT system architecture towards fault-tolerance.
11. What is software robustness? Mention the need for subroutine calls in program writing.
12. Explain the desirable features of systems software factors that shall contribute to supercomputing area.
13. Explain digital data communication system with a block diagram and also Shannon's coding theorem for a noisy channel.
14. With a block diagram, explain the operation of Electronic switching system organisaiton.
15. Discuss the architecture of Tandem-16 fault-tolerant system towards system reliability.
16. Comment on the system maintainability with respect to the pluribus multiprocessing environment.
17. Discuss on the hardware/software trade-offs for the following applications:
    a. Artificial intelligence and expert systems.
    b. Real time control systems for robotics and computer aided production processes in mechanical engineering circles.

# CHAPTER - 7

# CO- PROCESSORS SCENARIO

## 7-1 INTRODUCTION

The increase of the hardware functionality circuits with the pipeline strategy necessarily paved the way for CISC (complex instruction set computer) systems. The RISC (reduced instruction set category) specially increased the processor in-storage and met the constraints of parallel processing with a synchronized approach on microprogrammed machines. In either case, obviously, there is an implication of far below the expected computational power. The application needs call for additional processors of similar or different nature of organisations to contribute for co-operative processes in order to obtain optimum performance figures.

The coprocessors can be an I/O (input-output) processor or an arithmetic processor depending on the application demands. Thus, computers today employing 2 or more processors of same or different nature can be referred to as a coprocessing environment on well co-ordinated operating sytems. The above statement indicates an altogether different approach as compared to the SIMD (single instruction multiple data) or the CISC machines. The communication processor assists data flow on batch machines whereas arithmetic coprocessors meet the turn around time of dedicated computing tasks. The idea of *co-processing* avoids the bus bottleneck problems, improves the data band width utility and innovates newer confuguration management techniques. The data base management system of an interactive nature has to take care of deadlocks and livelocks. A critical region in a program refers to a section of code that is executed with exclusive access to the shared data. *Livelock* is a situation that results in no advancement in the computing process. Software engineering for parallelism orients itself to avoid deadlock and livelock; to prevent unwanted race conditions; to restrict the creation of too many parallel constructs; and to detect program termination.

In parallel programming environment, more emphasis must be placed on writing code that is correct because of tremendous difficulty which may creep in at debugging time.

Programme organisation in terms of pure mathematical functions enhances possibilities for correctness and automatic compiler checking.

The chapter is further organised as follows: The trends of VLSI needs followed by networking configuration examples. The 8087 coprocessor abilities are touched upon. The TMS 320 family of digital signal processors having a specific orientation is discussed later.

The fault-tolerant computing aspects to CMOS domain is presented and concludes with local area network.

## 7-2 Hardware Replacement Fact

The two basic hardware schemes for implementing software functions are Read Only Memories and the hard-wired logic gates using MSI (Medium scale integration) circuits or the field programmed PLAs (programmable logic arrays). See fig. 7-1.

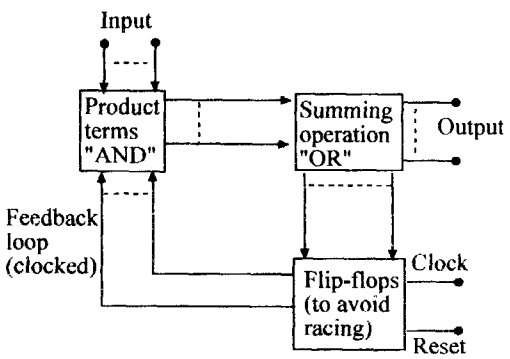



**Fig. 7.1** High speed PLA using Schottky TTL or ECL

Programmable logic arrays are suitable for sorting, Fast fourier transforms (FFT), and floating-point arithmetic. Intersil IM 5200 or signetics 82S100 are used in random logic functions to interface the external LSI unit or bulk-storage memory (disk or tape) to the microprocessors. Research on computer architecture has been stimulated by the research on software engineering innovations in VLSI technology. The hardware must support functionality for portability that is an important feature for parallel architecture such as data flow machines.

Existing super-computers used techniques such as pipelined processing (TIASC and CDC Star - 100), vector processing (CRay-1 and cyber-205) , and array processing (Illiac-IV, BSPand MPP) to provide high performance and achieve procesing rates of hundreds of magaflops.

## 7.3 Sample Systems

### BBN Butterfly parallel processor:

Initially developed as a high-speed network switch, the Butterfly has been used in a variety of applications like fluid dynamics, image understanding and data communications. Each processor node consists of : Motorola MC 68020 CPU and MC 68881

Floating point coprocessor; 1to 4 Mbytes of local memory; A process node controller (PNC). An I/O bus; A butterfly switching interface. The number of switching elements for an n-processor system increases at a rate of n $\log_4 n$ which is significantly better than the $n^2$ complexity of a crossbar switch. The development environment (compilers, libraries, etc.) reside on the host. Programs for the Butterfly are written on the front-end- computer, either a DEC VAX or SUN work station, running under UNIX 4.2 BSD. It is as of now a single user system of conventional CISC architectures. *OCCAM* is an interesting language, that shall evolve to support massively parallel architectures, particularly architectures without global storage as of the transputer domain.

### IPSC ( Intel personal super computer )

### Hypercube

For any *hypercube*, if d is the dimension of the cube, it has d nearest neighbours and $2^d$ nodes implies 16 nodes.

d = 4

The average distance between any two nodes is d/2, and the maximum distance is d.

I/O (input-output) sub system design attains great significance in multimedia applications. Research has invaded techniques of designing I/O systems that can handle real-time demands of *multimedia computing.*

The IBM 360 third generation systems are mainly employed for batch inputs on a single language environment at any point of time.In order to achieve the targeted throughput, they employ communication (input-output) processors which are often called as channels for data flow. Byte-multiplexing and block multiplexing using direct memory access method are used inter-wovenly on this bench for resource utility and portability issues.

## 7.4 8087 Coprocessor

Intel 8087 is a math co-processor used along with 8086/8088 microprocessor to enhance the computing abilities of the system. The programmer of 8086 enjoys the privilege of the 8087 coprocessor capabilities with only the instruction set of the 8086 multiprocessing domain. The parallelism is accounted for by pipelining and memory

segmentation tactics thus ensuring fast arithmetic and a good throughput. This configuration could be more called as co-operative processing to meet the bus bottlenecks in compute organisations. The *Intel 8087*, a numeric processing unit, works in conjunction with either an Intel 8086 or 8088 microprocessor in order to support floating-point notations. The 8087 is a coprocessor to enhance the computing capacity of 80 x 86 systems. The (proposed) IEEE Binary Floating-point standard and the 8087 numeric processor are very closely related. One important numeric application is graphics. Robotics also require rapid responses involving matrix multiplications. Speech recognition, image processing and echo cancellation in telecommunications require moderate accuracy but very fast arithmetic. The 8087 can perform operations on 16-, 32- and 64- bit integer data and can also accommodate an 80- bit decimal format consisting of a sign bit and 18 decimal digits. There are seven unused bits with a sign bit. COBOL language standard requires only 18 decimal digits (the predominant language in data processing applications). Real numbers are represented by making use of a sign bit, followed by the exponent field, followed by the significand field. The sign bit specifies the sign of the significand. The value of a floating-point number is

$$(-1)^{sgn} * S * 2^{E}$$

where sgn is the value of the sign of the sign bit, S is the value of the significand (also called mantissa), and E is the value of the exponent (also called characteristic). The coprocessor 8087 contains the same number of address/data pins as 8086 for usage. The coprocessing approach in a way aims at pipelined parallelism where a CISC (complex instruction set computer) has to achieve the same by more space complexity. The 8087 has eight arithmetic registers, each 80 bits wide. These registers are used as a stack and environment contains a 3 bit field ST ( stack top) that indicates the chosen register. The assembler mnemonics for all 8087 instructions start with an F (for floating point) so they can be readily discerned from 8086 instructions. Though apparently stack architecture is compact for accessing, the book-keeping becomes a crucial issue with programmers. 8087 does support special arithmetic instructions like square root, partial tangent, etc. Trignonmetric, logarithmic and exponential functions are supported to its ability. With the available programming skills, FORTRAN compilers for the 8087 have intrinsic functions to perform load and store the control words. Intel's new pentium processor supports both fast computing and database applications. These possess continued software compatibility with high performance figures.

The T.90 series, first-ever wireless supercomputers (formerly code-name Triton) of CRAY systems carry 1 to 32 processors and provide upto 60 billion calculations per second of peak computing power.

## 7.5 Digital signal Processing

Digital Signal Processiong (DSP) involves the representation, transmission, and

manipulation of signals using numerical techniques and digital processors. Digital communications and computing offer better reliability and efficiency as compared to their analog counterparts in the field of signal processing. Thus the very fast compute capabilities of the TMS 320 series of processors with software support for emulation and simulation has invaded widely varying applications. They include speech processing, telecommunications, defence research, bio-medical engineering domains and graphic work stations.

The TMS 320 products are 16/32-bit single-chip micro-computers applying the array processing concept with a tremendous I/O (input-output) strength. The TMS 32020 Macro assembler allows TMS 32010 source code to be executed for upward-compatibility and has a compact instruction set of 109 that allows ease of software development. The TMS 32020 is fabricated in a 4μ NMOS technology and has a chip area of 119k square mil. It is produced in a 68-pin grid array package and has a typical power consumption of 1.2 w. The maximum clock frequency is 20.5 MHZ for an instruction rate of five million instructions per second. The TMS 320 oc 25 offers faster instruction time of 100 ns.produced using CMOS version. The development tools range from very inexpensive evaluation modules, assembler/linkers, and software simulators.

Some common digital signal processing routines and interface circuits are frequently used. For example, the same structure of a digital filter used for audio signal processing may also be used for a modem in data communications. A Fast Fourier Transform (FFT) routine can be used for analysing signals both in instrumentation and speech coding.

Digital filters can meet tight specifications on magnitude and phase characteristics and eliminate voltage drift, temperature drift, and noise problems associated with analog filter components. The two methods used are finite impulse response (FIR) and infinite impulse response (IIR) filters with the TMS 320 family of digital signal processors.

A high-speed numeric processor, such as the TMS 32020 digital signal processor, may serve as a coprocessor with a slower yet capable host in a computer system. The TMS 32020 is capable of performing numeric functions, such as a multiply-accumulate, in a single cycle(200ns). Other 16-bit processors such as the Motorola MC 68000 have other qualities such as " Supervisor and user modes" which endorse them to be host processors. The applications of the MC 68000 - TMS 32020 interface include speech processing, spectrum analysis and graphics library. The design of a full-duplex 2400-bit per sec *vocoder* implementing an LPC (Linear predictive coding) algorithm in real-time making use of TMS 32010 is discussed in published proceedings.

The ADSP – 2100 family of processors are programmable single – chip microcomputers optimized for digital signal processing (DSP) and other high - speed numeric processing applications. They support serial interfaces to be used with personal

computers. The ADSP 2181 is noteworthy to be mentioned for it has gained entry on laboratory workbenches. The program memory is organised as 24 bits wordlength. The data address generators (DAG) provide memory addresses when memory data is transfered to or from the input /output registers. With two independent DAGs, the processor can generate two addresses simultaneously for dual oparand fetches. ADSP cards are now coming up for in-circuit emulation for portable environments. In addithion, the ADSP helps for micro computer diagnostics development system especially to run - time events (debuggers). The C compiler reads ANSI C source and outputs ADSP language mnemonics that is ready to be assembled. It also supports inline assembler code. Signal processors demand fast and flexible arithmetic, extended dynamic range, hardware circular buffers as used in filter algorithms and zero - overhead in looping and branching operations.

## 7.6 CMOS AND FAULT - TOLERANCE

Complementary Metal Oxide Semiconductor (CMOS) technology has become popular because of its low-power requirement and high density. The complexity of testing increases with circuit density. Testing in the context of digital systems is defined to be the process by which a defect can be exposed. Permanent faults (stuck-at) are desirable candidates for design for testability during the manufacturing process. Error-detecting codes are widely employed in fault-tolerant computer systems. In particular, the inputs and outputs of a self-checking circuit are assumed to be encoded with a suitable error-detecting code. The choice of the code depends on what errors are most likely. Programmable logic arrays (*PLAs*) are desirable to be thoroughly tested for they offer flexibility of combinational circuit synthesis. At times, by software means the failures of a system are detectable, achieving importance in fail-safe systems. Several factors contribute to reliability measures in software engineering.

## 7.7 LOCAL AREA NETWORKS (LAN)

Computers have become affordable due to the LAN development that allows sharing of peripherals and costly resources, that is, information. The growth of LAN is clearly an indication of the improved practices in digital communications. Creating backups, security provision and establishment of standards are the potential problems for network users. The data communications tactics make extensive use of graph theory for network connectivity and routing algorithms. Thus, the co-processors may take a dominant role in the present age of information technology in all of its forms. The information provided of this bookwork hope to reach many aspiring computer learners.

## KEYWORDS

Coprocessing, Livelock; Occam, Hypercube, multimedia computing; Intel 8087; VOCODER; PLAs.

## PROBLEMS

1. Explain what do you mean by a coprocessor in atleast two ways.
2. Write, in brief, the butterfly parallel processor.
3. Describe, in detail, the co-processor 8087 for 80 x 86 systems.
4. Explain the importance of FIR and IIR filters for digital signal processing applications.
5. Discuss elaborately the Design for test of VLSI towards testability and fault-coverage.
6. What is a computer network?
7. Write notes on TCP/IP high level protocol for LANs environment.

## SUGGESTED FOR FURTHER READING


Andrew Holck and Wallace Anderson,"A single-processor LPC vocoder", ICASSP 184, proceedings, San Diego, CA, March 19-21, 1984.

Baer, Jean-Loup, "Computer Systems Architecture". Computer Science Press, Inc., 1980.

Balzers, R.M., "EXDAMS, Extendable Debugging and Monitoring System", Proceedings of the spring Joint Computer Conference (AFIPS Press, 1969) pp. 567-80.

Conte, G. and Del Corso, D., "Multi-microprocessor systems for real-time applications", Dordrecht, the Netherlands; D.Reidel, 1985.

David J.Kuck., "Structure of Computers and Computations", John Wiley, NAVS 1978.

Dias, D.M., and Jump, J.R., "Packet switching interconnection networks for modular systems", IEEE Comp., Dec. 1981 b, pp.43-53.

Ellingson, O.E., "Computer program and change control", Record of 1973 IEEE symposium on computer software reliability., 1973, pp.82-89.

Feitelson Dror G., "Optical Computing - A survey for Computer Scientists", The MIT Press Cambridge, 1988.

Hedlund, K.S., "Wafer scale integration of parallel processors", Ph.D Thesis, Comp. Science Dept., purdue univ. Ind., 1982.

Kim M.Y., "Synchronised Disk Interleaving", IEEE Trans. on compute., Vol.35, No.11, Nov.1986, pp.978-988.

Palmer J, "The Intel 8087 Numeric Data Processor", proceedings of the seventh annual international ACM symposium on computer architecture, 1980.

Palmer John F., and Stephen P.Morse, "The 8087 Primer" John Wiley & Sons., Inc., 1984.

Slotnick, D.L., Borck, W.C., and McReynolds, R.C., "The Solomon Computer", Proc. of AFIPS Fall joint Comp. conf., Wash.D.C., 1962, pp.97-107.

Wakerly J., "Error detecting codes, Self checking circuits and applications", Elsevier North - Holland, Inc., New York 1978.

Willis Neil, "Computer Architecture and Communications", Paradigm Publishing Ltd., 1986.

THIS PAGE IS
BLANK

# Index