COMPUTER AIDED HEURISTICS FOR ASSEMBLY LINE BALANCING AND CAPACITY PLANNING

Thesis submitted in partial fulfillment of the requirements
for the award of the degree of

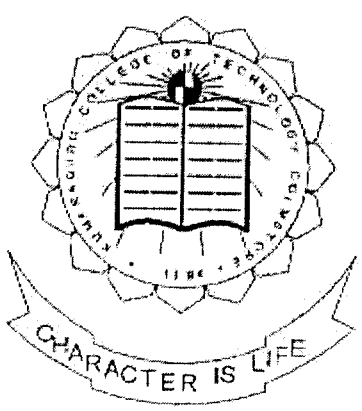
MASTER OF ENGINEERING IN MECHANICAL ENGINEERING
(INDUSTRIAL ENGINEERING)
of BHARATHIAR UNIVERSITY

By

N. KATHIRAVAN
(Reg. No.0037H0005)

Under the Guidance of

Mr.G. KARUPPUSAMY, M.E.,
Dept. of Mechanical Engineering



DEPARTMENT OF MECHANICAL ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University)
Coimbatore - 641 006

2000 - 2001

CERTIFICATE

Department of Mechanical Engineering

Certified that this is a bonafide report of thesis work done by

Mr. N. KATHIRAVAN

(Reg. No. 0037H0005)

at

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641006

During the year 2000 –2001

.....

Guide

Mr.G.KARUPPUSAMY

.....

Head of the department

Dr. A. MOHANAKRISHNAN

Department of Mechanical Engineering,
Kumaraguru College of Technology.

Place: Coimbatore
Date :

Submitted for viva – voce examination held at

Kumaraguru College of Technology on.....

.....
4/12/2001
Internal Examiner

.....
External Examiner

CERTIFICATE

This is to certify that this thesis work entitled **"COMPUTER AIDED HEURISTICS FOR ASSEMBLY LINE BALANCING AND CAPACITY PLANNING"** being submitted by **KATHIRAVAN .N** (Reg.N o.0037H0005) for the award of degree of **MASTER OF ENGINEERING IN MECHANICAL ENGINEERING (INDUSTRIAL ENGINEERING),** is a bonafide work carried under my guidance. The results embodied in this thesis have not been submitted to any Degree or Diploma

Mr.G. KARUPPUSAMY
Senior Lecturer
Dept. of Mechanical Engineering,
Kumaraguru College of Technology
Coimbatore.

Dedicated to

my

Beloved Parents

Certificate

ROOTS INDUSTRIES LIMITED

LICENCE BOSCH

27-Nov-2001

PROJECT WORK CERTIFICATE

This is to certify that **Mr.N.Kathiravan** M.E. student of Kumaraguru College of Technology, Coimbatore has done a project work on **"Computer Aided Heuristics For Assembly Line Balancing And Capacity Planning"** during the period of June- 2001 to Nov- 2001 in our organisation.

KAVIDASAN
AGM - Corporate HRD

TÜV CERT DIN EN ISO 9001: 1994 ISO/TS 16949: 1999 Zertifikat: 01 111 009578

R.K.G. INDUSTRIAL ESTATE,
GANAPATHY,
COIMBATORE - 641 006
INDIA

TEL : +91 - 422 - 532100
FAX : +91 - 422 - 532107
E-MAIL : info@roots-india.com
INTERNET : www.rootsindia.com

TÜV Rheinland

Acknowledgement

ACKNOWLEDGEMENT

I express my deep sense of gratitude to our management and to **Dr. K. K. Padmanabhan**, Principal, Kumaraguru College of technology, Coimbatore, who has given me the great opportunity to undergo this course.

I extend my sincere thanks to **Dr. A. Mohanakrishnan**, Professor and Head of the Department of Mechanical Engineering who has shown keen interest in my work and given valuable suggestions to complete my project, which leads to completion of my course.

The author is Grateful to **MR. G. Karuppusamy**, Senior Lecturer, Department of Mechanical Engineering, for his valuable guidance in all stages with more enthusiasm to make this project a grant success.

I extend my sincere thanks to **Mr. N. Gunasekaran**, Assistant Professor, Department of Mechanical Engineering, for his continues encouragement in bringing up this project.

The author is extremely grateful to the chairman and Managing Director of Roots Industries Ltd., **Sri. K.Ramaswamy**, who has given me permission and support in doing this project.

I gratefully express my thanks to **Mr. V. Jayakumar**, Senior Manager, Manufacturing Engineering for their appreciation and suggestions given during discussion at various stages of the project.

Finally, I would like to thank my friends for their kind help, which enabled me to make this venture more useful for Roots Industries Limited.

N. KATHIRAVAN

Synopsis

SYNOPSIS

Manufacturing sector plays a pivot role in improving the economy of any country. The success of the manufacturing sector mostly depends on productivity because productivity is the watchword for any organization. This project work is done at ROOTS INDUSTRIES LIMITED (RIL), COIMBATORE, which manufactures horns.

Due to globalization and because of hectic competition in this sector, RIL, Coimbatore is trying to introduce cost reduction strategies at shop floor level and productivity improvement and optimization techniques to compete in the market beyond 2000. Among various techniques available to improve productivity, Assembly Line Balancing (ALB) is a dominant technique for mass production industries like automobile manufacturing and consumable items manufacturing industries.

Assembly line is the best example of product layout, which consists of number of work elements, each having its own processing time. The time, which is given, as input for line balancing is the maximum station time, which is found by the Work Measurement Technique i.e., Stop Watch Time Study. The objective of ALB problem is to assign number of work elements to various workstations without violating their precedence relationship in such a way that the balancing line efficiency is maximized for a given volume of output. This is done using the computerized line balancing method COMSOAL (Computer Method Of Sequencing Operations For Assembly Lines) developed by Chrysler Corporation.

A package is developed using Visual Basic 6.0 as front end and MS-Access as back end for COMSOAL. This package avoids the diligence of the Time Study Man as it adds the advantages of computer.

The package developed is very much user-friendly and gives quick reports as and when the input is changed. This avoids the drawbacks of line balancing and capacity planning when done manually. Ultimately this project, concentrates on the development of a generalized package i.e., "COMPUTER AIDED HEURISTIC FOR ASSEMBLY LINE BALANCING AND CAPACITY PLANNING".

CONTENTS

Certificate		I
Acknowledgement		II
Synopsis		III
S.NO	**CHAPTER**	**PAGE NO**
1.	INTRODUCTION	
	1.1 Company profile	1
	1.2 Work study	2
	1.3 Assembly line balancing	3
2.	LITERATURE SURVEY	5
3.	PROBLEM DEFINITION	
	3.1 Introduction	8
	3.2 Problem definition	
	3.3 Objective of the Project	9
4	METHOD ANALYSIS	
	4.1 Work Measurement.	11
	4.2 Assembly line balancing	17
5.	ASSEMBLY PROCESS IN RIL	
	5.1Description	21
	5.2 Flowchart	23
6.	METHODOLOGY	
	6.1 COMSOAL Method	26
	6.2 Flowchart	28
	6.3 constraints and conditions	30
7.	SOFTWARE DESIGN	
	7.1 Introduction	31
	7.2 Database Design	
	7.3 Mode of Accession	37
8.	FORMS AND CODING	39
9.	RESULTS AND DISCUSSION	78
10.	CONCLUSION	79
	Bibliography	83

Introduction

1.INTRODUCTION

1.1 Company profile

Roots Industries Limited Industries was established in the year 1970 at Coimbatore. The company was the first to introduce the servo brake for light motor vehicle (1970-1972). This diversified to manufacture the indigenously high frequency wind tone Horns and later started the promotion of various pneumatic and electrical Horns since 1973.

With its new vibrating Horns taking the market by storm, the company currently holds the no.1 position in the Indian market. Roots Industries Limited is a leading supplier to Original Equipment manufacturers such as the Hindustan Motors, Toyota, TVS Suzuki, Benz, Uno, Maruthi, Telco, M&M, Kirlosar, Hero Honda, Bajaj, etc.

Roots Industries Limited is the first Horn manufacturing company in India to get ISO 9001certification in 1994 for electric Horn design and it is the first Horn manufacturing company in Asia to bag the QS- 9000 in 1998. Roots Industries Limited is the first company to get E certification from Europe for its products, and also to get VDA 6.1 and A 6052.

Roots Industries Limited is largest exporter of Horns in India, exporting Horns to Japan. In 1995 the company entered into technical collaboration with M/S.Robert Bosch, Spain to further enhance the technological competence.

Roots Industries Limited is the 11th largest Horn manufacturing company in the world. It is the only Indian Horn manufacturing company to satisfy the exact standards of the Japanese customers, 25 % of its export goes to Japan.

Roots Industries Limited is proud to have developed Horns successfully for Mercedes Benz in a record time of 3 months. It has full-fledged design, development and test center with CAD/CAM facilities. It established Metrology lab and it is equipped with the latest state of art, world-class instruments/equipments.

ARUNA AUTO CASTING is a unit of Roots Industries Limited, started in 1983. It is an aluminum and zinc pressure die casting unit with capacity of 500 tons per annum meeting the requirements of various local industries and exporting to Italy, Germany etc.

Poly Craft manufactures, which is also a unit of Roots Industries Limited. It manufactures Precision Molded Plastics Components. ROOTS MULTI CLEAN LTD is a joint venture company collaborated with M/S. Hakko Werke Gmbh &Co., Germany. It manufactures world-class industrial floor cleaning equipment.

1.2 Work Study

Work-study is the systematic examination of the methods of carrying on activities so as to improve the effective use of resources and to set up standards of performance for the activities being carried out.

Work study then aims at examining the way an activity is being carried out, simplifying or modifying the method of operation to reduce unnecessary or excess work, or the wasteful use of resources, and setting up a time standard for performing that activity. If work study results in cutting down the time of performing a certain activity by 20 percent, merely as a result of rearranging the sequence or simplifying the method of operation and without additional expenditure, then productivity will go up by a corresponding valve, that is by 20 percent.

Work-study is a more valuable tool. It is a means of raising the productivity of a plant or operating unit by the reorganization of work, a method that normally involves little or no capital expenditure on facilities and equipment. It is systematic. That ensures that no factor affecting the efficiency of an operation is overlooked, whether in analyzing the original practices or in developing the new, and that all the facts about that operation are available. It is the most accurate means yet evolved of setting standards of performance, on which the effective planning and control of production depends.

It can contribute to the improvement of safety and working conditions at work by exposing hazardous operations and developing safer methods of performing operations. The saving resulting from properly applied work start at once and continue as long as the operation continues in the improved form. It is relatively cheap and easy to apply.

1.2.1 TECHNIQUE OF WORK STUDY

The term "work study" embraces several technique, but in particular method study and work measurement.

Method study

Method study is the systematic recording, analysis and critical examination of existing and proposed ways of doing work and the development and application of easier and more effective methods

Method study is concerned with the reduction of the work content of the operation or job.

Work Measurement

Work measurement is the application of techniques designed to establish the work content of a specified task by determining the time required for carrying it out at a defined standard of performance by a qualified worker.

It is concerned with the investigation and reduction of the ineffective time and the subsequent establishment of time standards for the operation on the basis of the work content as established by method study.

1.3 Assembly Line Balancing (ALB)

Assembly line balancing consists of partitioning the work necessary to assemble a product(s) among different workstations of an assembly line. The objectives of assembly line balancing are to balance the workload across workstations so that no workstation has an excessively high or low workload. This helps in minimizing the idle time at workstations and maximizing the throughput from the line.

The line-balancing problem is to arrange the individual processing and assembly tasks at the workstations so that the total time required at each workstation is approximately the same. If the work elements can be grouped so that all the station times are exactly equal, we have perfect balance on the line and we can expect the production to flow smoothly.

Line balancing is an attempt to allocate equal amount of work to the various workstations along the line. The fundamental line-balancing problem is to assign a set of tasks to an ordered set of workstations, such that the precedence relations are satisfied and some performance measures are optimized.

When manufacturing activities are laid out as a line in a product layout it is important to ensure, as far as is possible, that the capacity of each facility on the line is approximately the same, to do otherwise would be inefficient and wasteful. Facilities should be planned therefore to provide a similar capacity. This may involve buying larger or smaller machine to suit.

When designing assembly lines, which involves many manual assembly tasks, it is possible to organize where and when tasks are done to maximize efficiency. For example the assembly of a product may require say twenty tasks, of differing standards times. This does not mean that twenty operators are required, each doing a single task. Doing each separately and in sequence would be very inefficient. There may be some flexibility in the sequence in which the tasks are done but much of the sequence will be fixed.

Literature Survey

of the SALB problem were contributed by Klein (1963); Patterson and Albratch (1975) and Talbot and Patterson (1984), who formulated it as a general integer program without binary variables. The formulation proposed by Patterson and Albratch (1975) significantly reduced the size of the problem formulation.

Dar-Ei and Rubinovitch (1979), proposes multiple solutions technique (MUST), which employs exhaustive enumeration to generate all solutions, or subset of them, for solving the type-II problem. As with MLAB (DAR-EL 1973), the type II problem is solved as a sequence of type I problems, the cycle time is fixed, and a hurdle number of workstations, . .is specified. If a feasible number of workstations cannot be found given. the cycle time is increased one time unit, and process is repeated. Starting with station one MUST generate all feasible task assignment.

Johnson (1983) solved ALB problem with some formulation irregularities. He solved problems having stochastic task processing times, allowing specific tasks to be assigned workstations, requirement of grouping tasks according to task skill levels and mixed model situations.

Consequently, researchers have concentrated primarily on developing heuristics. The immediate update first fit (IUFF) heuristic by Hackman et al (1988); depends on numerical sore functions that have been proposed in the literature.

Since then ALB has attracted the attention of both researchers and practitioners. Researchers have used exact or optimum techniques like dynamic programming and branch and bound techniques etc., to get optimum solutions and also heuristic technique like COMSOAL (computerized method of sequencing operations for assembly lines). It is developed at Chrysler Corporation and reported by Arcus in 1966[1]. Ranked Positional Weight (RPW) is developed by Helgeson And Brine, (1961).

While the RPW heuristic and other methods gives reasonably good solutions, it only gives a single solution. Often criteria other than minimum number of stations may have to be considered. It is therefore important to obtain several solutions. One way to achieve this is to use a randomized algorithm. In such an algorithm, several possible solutions can be generated by randomly selecting a feasible task from the available tasks instead of selecting the "best" task according

to some criterion. The Computerized Method For Sequencing Operations On Assembly Lines (COMSOAL) method uses this strategy. At each stage of the algorithm, the set of tasks that can be feasibly and which fit in the remaining time on the current work- station is determined. The task to be assigned is then randomly selected with each task having the same probability of being selected.

Problem Definition

3. PROBLEM DEFINITION

3.1 Introduction

The focused area of the project is the Horn assembly line of ROOTS IINDUSTRIES LIMITED. The company produces different types Horns for both the two wheelers as well as four wheelers. Roots Horn unit mainly do the Assembly Operations including both Sub-Assembly and Final Assembly operations.

There are as a whole 6 Assembly lines. In each line two different type of Horn assembling is going on. Those two-types are Screw type Horn and Riveting type Horn. Based on the requirement, the product is changed in the line. A continuously moving conveyor-line does the transfer of work from one station to another station. The company adopts Batch Type Production.

3.2 Problem definition

The problem identified in Roots Horn assembly line is, there are no standard times and the tasks are not equally distributed to workstations. So they are facing the following problems:

a. No proper workload distribution

Due to the improper allocation of tasks some station are heavily loaded and some stations are less loaded. This makes the worker in the heavily loaded station get more tired than the other people.

b. Lot of idle time

The idle time for the person who is working on the less loaded station is high, and he is also getting the same amount of wages like the person who is in the heavy loaded station.

c. No time standards

The standard time is not determined in the production line. So it is not possible to calculate the time required for manufacturing the required quantity of products and the production planning and control becomes inefficient.

d. High throughput time

Throughput time is defined as " the time taken for a job between the beginning of its first operation and the time it comes out of the last operation". This is high for the existing system because of the improper line balance.

e. No professional line balancing

Due to absence of professional assembly line balancing, it is not possible to distribute tasks evenly over the workstations, so that idle time of men and machines is not minimized.

3.3 OBJECTIVES OF THE PROJECT

This project concentrates on the following Objectives:

- Conducting Time Study
- Line Balancing of the Assembly line of Horns
- Developing a package for Computerized Line Balancing Method called COMSOAL. The package gives a ready report on Capacity planning and Precedence relationship, Line efficiency, Balancing Delay, Optimum no. of workstations etc.
- Graphical comparison of Assembly line before and after Balancing

3.3.1 NEED FOR SOFTWARE BASED SOLUTION

The manual balancing has some drawbacks

- If some change is required in current balancing in the form of change in output per shift or insertion or deletion of new activities, the whole balancing has to be done newly.
- Due to some natural constraints and constraint imposed by manual balancing a true homogeneous division of work is not possible.
- If some change is done in allowance or rating factor or Average Observed Time, the complete calculation has to be done to find the standard time, which takes more time when done manually.

The software based solution should

- Satisfy all the constraints to give balancing which would cause a homogeneous division of work.
- Perform the balancing on different balancing related parameters, apart from output per shift and the predecessor relation constraint.
- Facilitate insertion deletion and updation of activities and the database.
- Maintain separate database for different models, allowing creation and deletion of models also.
- Optimal solution can be generated in a relatively short time.

Method Analysis

4.METHOD ANALYSIS

4.1 WORK MEASUREMENT

Work measurement is the application of techniques designed to establish the time for a qualified worker to carry out a task at a defined rate of working. It is concerned with investigating, reducing and subsequently eliminating ineffective time, that is time during which on effective work is being performed, whatever the cause.

4.1.1 The use of Work measurement

- To compare the efficiency of alternative methods. Other conditions being equal, the method, which takes the least time, will be the best method.
- To balance the work of members of teams, in association with multiple – activity charts, so that each member has a task taking an equal time to perform.
- To determine in association with man – machine charts the number of machines an operative can run.

The time standards, once set, may then used:

- To provide information on which the planning and scheduling of production can be based, including the plant and labour requirements for carrying out the programmed of work and the utilization of available capacity.
- To provide information that can enable estimates to be made for tenders, selling prices and delivery dates.
- To set standards of machine utilization and labour performance which can be used for any of the above purposes and as a basis for incentive schemes.
- To provide information for labour – cost control and to enable standard costs to be fixed and maintained.

2.LITERATURE SURVEY

Prior works on Time study & Assembly Line Balancing

Time Study, originated by TAYLOR, was used mainly for determining time standards. International labour organization, "introduction to work study", Geneva, 1965 has given all the details pertaining to time study. Numerous research efforts have been directed towards the development of computer efficient approximation algorithms or heuristics and their subsequent implementation to solve large-scale ALB problems.

Helgeson was the researcher to introduce the concept of ALB in 1954 and salveson first published it in its mathematical form in 1955 and suggested a linear programming solution. Since then, the topic of the line balancing has been of continuing interest to researchers and has some what paralleled the development of assembly lines in mass manufacturing.

Nemhauser and Gutijahr (1964) formulated the simple assembly line balancing (SALB) problem as a network and used the concept of finding the shortest route in a directed network to solve ALB problem. The major portion of the paper involves finding an efficient way to enumerate all feasible combinations called states. They construct the graph with the states and time as arc parameter, and use the shortest route technique to determine the station assignment.

Magazine and Web (1981) reported excellent results for type I line balancing problem with their branch and bound procedure. With their method each node in the solution tree corresponding to a feasible set of task assignment to a particular work station, where all the nodes at the same depth in the tree refer to the same station number. Hoffmann (1992) proposes a B&B approach to solve the ALB problem. He uses the concept of theoretical slack time in calculating bounds.

Scharge And Baker (1978) have proposed an efficient method for implementing dynamic programming approach through improved procedures for generating feasible subsets and for labeling. Integer programming (IP) formulation

Work measurement provides the basic information necessary for all the activities of organizing and controlling the work of the factory in which the time element plays a part.

4.1.2 The basic procedure

The following steps, which are necessary for the systematic carrying out of, work measurement.

- **Select** the work to be studied.
- **Record** all the relevant data relating to the circumstances in which the work is being done, the methods and the elements of activity in them.
- **Examine** the recorded data and the detailed breakdown critically to ensure that the most effective method and motions are being used and that unproductive and foreign elements are separated from productive elements.
- **Measure** the quantity of work involved in each element, in terms of time, using the appropriate work measurement technique.
- **Compile** the standard time for the operation, which in the case of stop- watch time study will include time allowances to cover relaxation, personal needs etc.
- **Define** precisely the series of activities and method of operation for which the time has been complied and issue the time as standard for the activities for the activities and methods specified.

4.1.3 The techniques of work measurement

The following are the principal techniques by which work measurement is carried out is shown in fig 1:

The choice of techniques depends on the level of detail desired and the nature of the work itself. Highly detailed, repetitive work usually calls for time study and predetermined motion –time data analysis. When work is done in conjunction with fixed processing time equipment, elemental

data are often used to reduce the need for direct observation. When work is infrequent or entails a long cycle time, work sampling is the tool of choice.

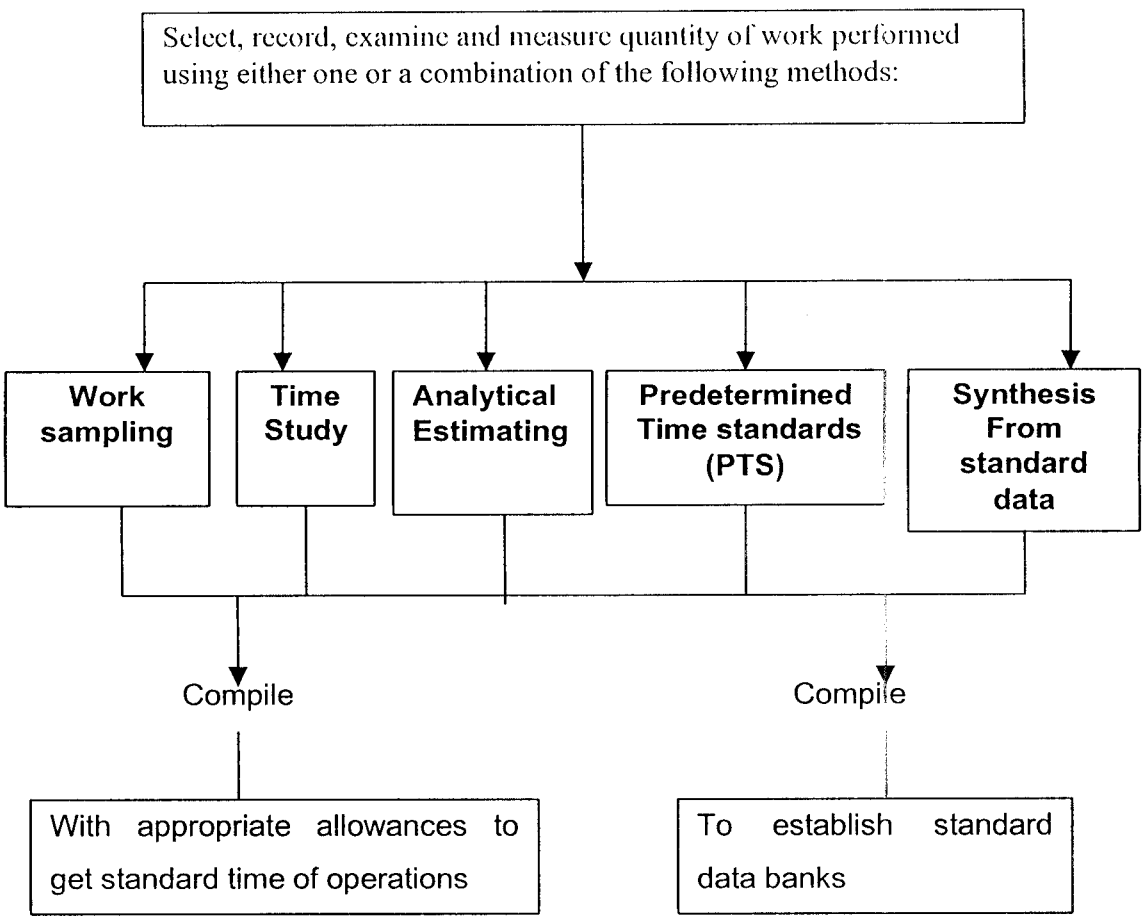


Fig 1.

4.1.4 Time Study

Time Study, originated by Taylor, was used mainly for determining time standards. One of the most important work measurement techniques namely time study.

Time study is a work measurement technique foe recording the times of performing a certain specific job or its elements carried out under specified conditions, and for analyzing the data so as to obtain the time necessary for an operator to carry it out at a defined rate of performance.

4.1.5 Time study equipment

If time studies are to be made, certain items of equipment are essential. Basic time study equipment consists of:

- A stop-watch
- A study board
- Time study forms

4.1.6 Steps in conducting Time Study

When the work to be measured has been selected, the time study usually consists of the following steps:

- Obtaining and recording all the information available about the job, the operator and surrounding conditions likely to affect the carrying out of the work.
- Recording a complete description of the method and breaking the operation into "elements".
- Measuring with a timing device (usually a stopwatch) and recording the time taken by the operator to perform each "element" of the operation.
- At the same time assessing the effective speed of the working of the operator relative to a predetermined "normal" speed.
- Converting the observed times to "normal times".
- Determining the allowances to be made over and above the normal time for the operation.
- Determining the "standard time" for the operation.

4.1.7 Terminology used in time study

I. Rating factor

Rating (also known as performance rating) is the mental comparison by a work-study man of the performance of an operator under observation with his own idea of a standard performance for a given method.

Normal performance (or pace) is the working rate of the average worker working under capable supervision but without the stimulus of an incentive wage-payment plan. This pace can easily be maintained day after day without undue physical or mental fatigue and is characterized by the fairly steady exertion reasonable effort.

II. Observed time

This is the actual time observed by using a stopwatch. The observed time of an operation is the total of the elemental times. The time study for the same job is conducted for a number of times. The average of the observed times is calculated.

III. Basic time

Basic time is the time for carrying out an element of work at standard rating, i.e.

$$\text{Basic time} = \frac{\text{Observed time} * \text{observed rating}}{\text{Standard rating}}$$

IV. Allowances

It is not possible for a worker to do job continuously without any break. There are many interruptions (stoppage of work) taking place. Extra time is added to the basic time to compensate this interruption. This extra time given is known as allowance.

Generally interruption occur to the following:

- Personal factors:

  Going for drinking water, toilet etc.

- Nature of work:

  Taking rest after hard work

- Other factors:

  Tool breakage, listening to supervisor etc.

Various types of allowances are:

- **Process allowances**

A process allowance is an allowance of time given to compensate for enforced idleness (which would otherwise cause loss of earning power) on the part of an operative due to the character of the process or operation on which he is employed.

- **Rest allowances**

The rest allowance is an addition to the normal time (usually calculated as a percentage) intended to provide the worker with an opportunity to recover from the physiological and psychological effects of expending energy in the performance of specified work under specified conditions and to allow attention to personal needs.

- **Fatigue allowances**

Fatigue is a physical and / or mental weariness, real or imaginary, existing in a person and adversely affecting his ability to perform work.

- **Contingency allowances**

A contingency allowance is a small allowance of time which may be include in a standard time to meet legitimate and expected items of work or delays, the precise measurement of which is uneconomical because of their infrequent or irregular occurrence.

- **Policy allowances**

A policy allowance is an increment, other than bonus increment, applied to standard time (or to some constituent part of it, e.g. work content) to provide a satisfactory level of earning for a specified level of performance under exceptional circumstances.

- **Special allowance**

In a shop, some activities take place occasionally. These activities will not be part of the production cycle. But these are necessary for production work. Examples of these activities are: -

1. Tool re-setting
2. Cleaning
3. Tool maintenance

- Select a primary rule by which tasks are to be assigned to workstations, and a secondary rule to break ties.
- Assign tasks, one at a time, to the first workstations until the sum of the task is equal to the cycle time, or no other tasks are feasible because of time or sequence restrictions. Repeat the process for workstation 2, workstation 3,and so on, until all tasks are assigned.
- Evaluate the efficiency of the balance derived using the formula

$$\text{Efficiency} = \frac{\text{Sum of task times (T)}}{\text{Actual number of workstations (N)*cycle time(c)}}$$

4.2.3 Terminology used in Assembly Line Balancing

I. Work station

A workstation is a location on the line layout where a certain group of elemental tasks are performed. The workstation may have one or more operations and may have one or more of the same equipment. Similarly, an operator can share by more than one workstation.

II. Element task or work element

This is defined as the smallest work unit beyond which further sub-division of work will not make much sense.

III. Station time

Based on the group of elemental tasks assigned to the workstation, the 'station time is the total amount of time required to perform that group of elemental tasks.

IV. Cycle time

This is the amount of time required to produce one unit of the finished product. This is also the total amount of time available at each workstation. The cycle time is different from the station times and has to be equal to or greater than any of the station times.

V. Total work content

It is the aggregate of all work elements to be done on the line. Let Twc be the time required for the total work content. Hence

Twc = sum of all elements time.

VI. Workstation process time

A workstation is a location along the floe line where the work is performed, either manually or by some automatic device. The work performed at the station consists of one or more of the individual work elements and the time required is the sum of the times of the work elements done at the station. Tsi is indicate the process time at station I of an n station on line. It should be clear that the sum of the station process times should equal the sum of the work element times.

VII. Line efficiency

Line efficiency is defined as total work content divided by total time taken by the product from the first workstation to the last workstation

$$\text{Line efficiency} = \frac{\text{Total work content}}{\text{No of station} * \text{max station time}}$$

VIII. Balance delay

Sometime also called balancing loss, this is a measure of the line inefficiency, which results from idle time due to imperfect allocation of work among stations. It is symbolized as D and can be computed for the line as follows:

$$D = nTc - Twc / nTc$$

Where,

n = total no of station

Tc = max station time

Tw = total work content

Assembly Process in RIL

5.ASSEMBLY PROCESS IN RIL

5.1Description:

The assembly of electric Horn is carried out in product type layout with dedicated assembly lines for different type of products. The study is conducted on assembly line of similar product. The assembly line layout is given in the figure 3.

Present Assembly Process in RIL

At present the electric Horn (**riveting type**) assembly is split into the following stages. A belt conveyor is used to move the subassembly between stages. This belt conveyor is shared on both sides of the conveyor by two assembly lines.The process flow chart of present assembly is given in the fig 4.

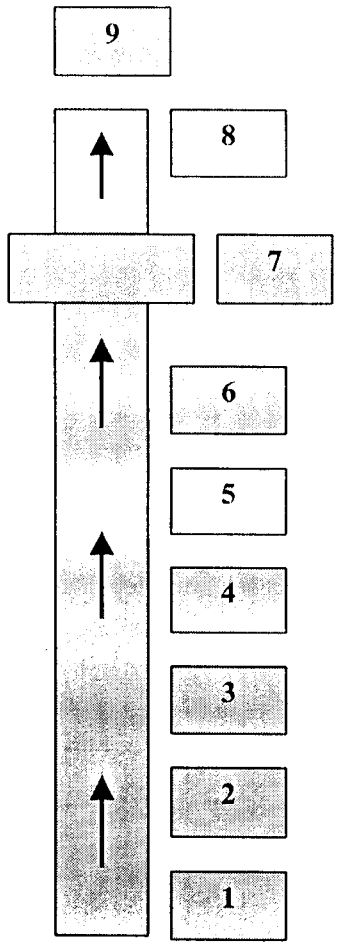
1.Coil Locking Stage:

In the coil locking stage, the housing with core assembly is located into the fixture and spool assembly is placed over it. Two guide pins are used to locate the housing and spool assembly. Two-hand operated push button switch is pressed to operate the fixture. The spool assembly is located into the main body-using fixture operated by pneumatic force. Then the subassembly is taken out by hand and terminal base assembly, point plate assembly inserted into the holes manually. Then the subassembly is placed on the conveyor and moved to the next stage.

2.Terminal Riveting Stage:

The subassembly of the previous stage is taken from the conveyor and point holder assembly, washer is assembled and the whole assembly is placed into the fixture. The hydraulic fixture is operated by hand and riveted.

Then the tuning screw is located on the tuning screw assembly fixture and the housing assembly after riveting from the hydraulic fixture is moved to the self-tapping fixture. Now the self-tapping fixture is operated by two hands operated Push button Switch. After the assembly of tuning screw is completed, the housing assembly is placed on the conveyor.



S.No	PROCESS NAME
1	COIL LOCKING
2	TERMINAL ASSEMBLY RIVETTING
3	DIAPHRAGM ASSEMBLY RIVETTING
4	PRE-CRIMPING & FINAL CRIMPING
5	AIR GAP ADJUSTING
6	MOUNTING BRACKET ASSEMBLY
7	HORN TESTING
8	TUNNING RANGE CHECKING & BATCH NO STAMPING
9	PACKING

Fig 3. HORN ASSEMBLY LINE LAYOUT

PROCESS FLOW CHART

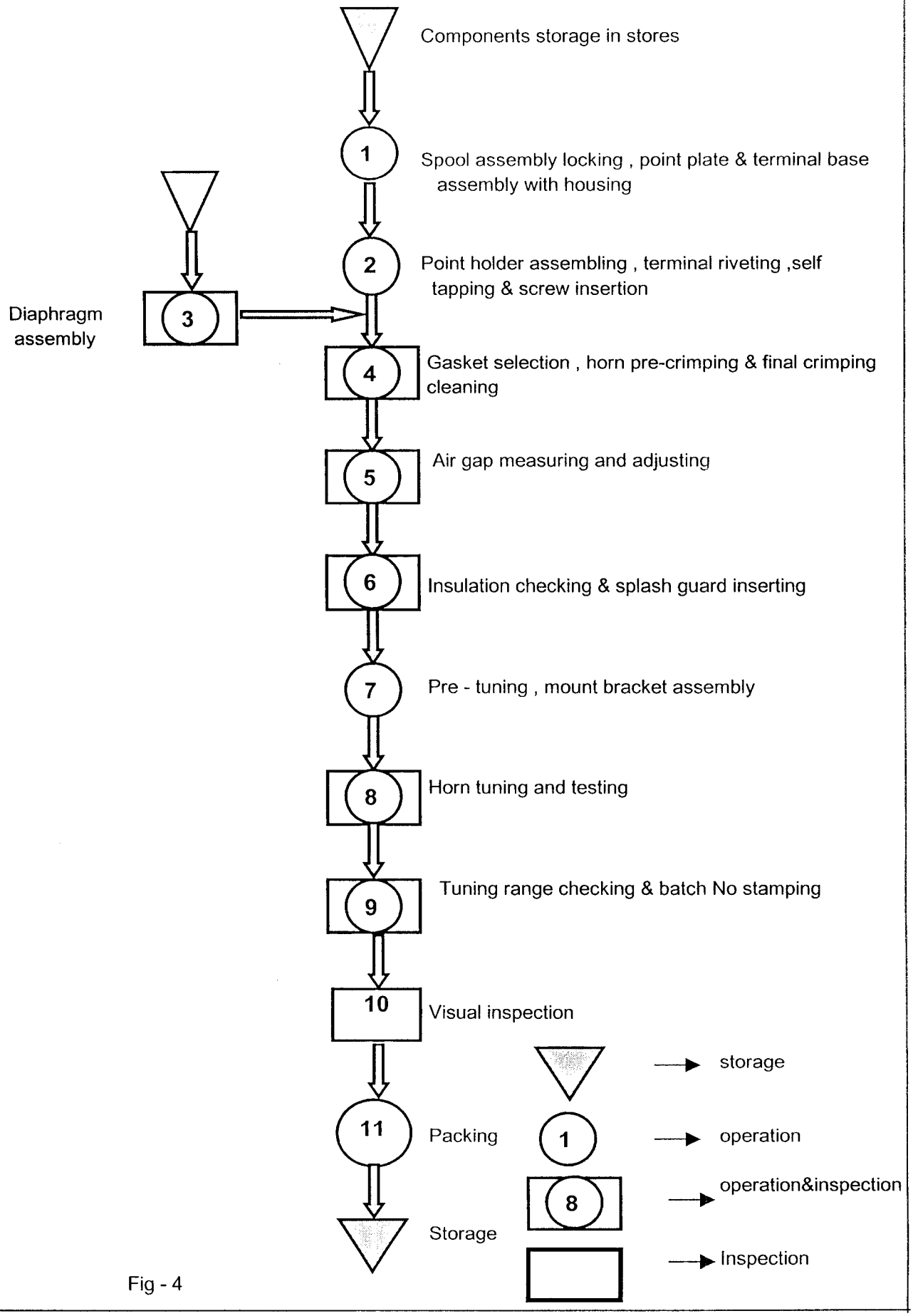


Fig - 4

3.Diaphragm Assembly:

The Armature rod, Spacer, Diaphragm, Spacer top, Tone disc and Washer are assembled in the given sequence and kept in the hydraulic riveting fixture. Now the riveting fixture is operated by two hands operated Push button Switch and riveted. Then the diaphragm assembly is moved to the inspection fixture and located. Then turning the handled operates the inspection fixture and the diaphragm assembly height is inspected. After inspection, the diaphragm assembly is moved to conveyor.

4.Crimping:

The housing assembly and diaphragm assembly from the conveyor is moved to the crimping stage. The Crimp ring, Diaphragm assembly, Gasket and Housing assembly are assembled in the given sequence and kept in the pre-crimping fixture. Now the crimping fixture is operated by Two-hand operated Push button Switch and crimping is carried out. After the crimping operation, assembly is moved to the next stage.

5.Air Gap Setting:

The Horn assembly from the conveyor is taken out and kept in the air gap setting fixture and operated by Push button Switch. The Horn after air gap setting is taken out and inspected for required air gap. After Air gap setting, the Horn assembly is placed on the conveyor.

6.Mounting Bracket Assembly:

The Horn assembly from the conveyor is taken out and kept in the air gap setting fixture and two brackets are assembled to the Horn at required orientation using washer and nut. Pneumatic Nut Runner is operated by hand to assemble the bracket. The Horn is placed on the pre-tuning fixture and operated by the Push button Switch. Then after pre-tuning, the Horn is placed on the conveyor.

7.Tuning and Testing:

The Horn from the conveyor is taken out and fixed in the testing fixture. The wiring connections are given then checked for performance. If required, the Horn is tuned and then rechecked for performance. The values of sound pressure level, current and frequency are noted down and Horn is moved to the conveyor.

8. Water Proof Sealing & Batch no stamping

In this station, the horn housing is sealed with Ana-bond paste to make it water proof and also it is marked with Batch number, Year and Month of Assembly, Shift in which it is done. After Visual Inspection, Quality sticker is pasted for its acceptance i.e. OK on the housing. A paint solution is also applied on the tuning screw for identification when replacing the customers demand on damaged items.

9.Packing

Horns are packed in carton boxes and it is safely sealed in this station.

Methodology

6.METHODOLOGY

6.1 COMSOAL:

This acronym stands for" Computer Method Of Sequencing Operations For Assembly Lines". It is a method developed at Chrysler Corporation and reported by Arcus in 1996 [1].

While the RPW and other heuristics method give reasonably good solutions, it only gives a single solution. Often criteria other than minimum number of stations may have to be considered. It is therefore important to obtain several solutions. One way to achieve this is to use a randomized algorithm. In such an algorithm, several possible tasks instead of selecting the "Best" task according to some criterion. The Computerized Method for Sequencing Operations on Assembly Lines (COMSOAL) method uses this strategy. At each stage of the algorithm, the set of tasks that can be feasibly assigned and which fit in the remaining time on the current workstation is determined. The task to be assigned is then randomly selected with each task having the same probability of being selected.

6.1.1 Algorithm:

Step1: Construct list A, showing all work elements in one column and the total number of, elements that immediately precede each element in an adjacent column.

Step2: Construct list B, showing all elements from list A that have no immediate predecessors.

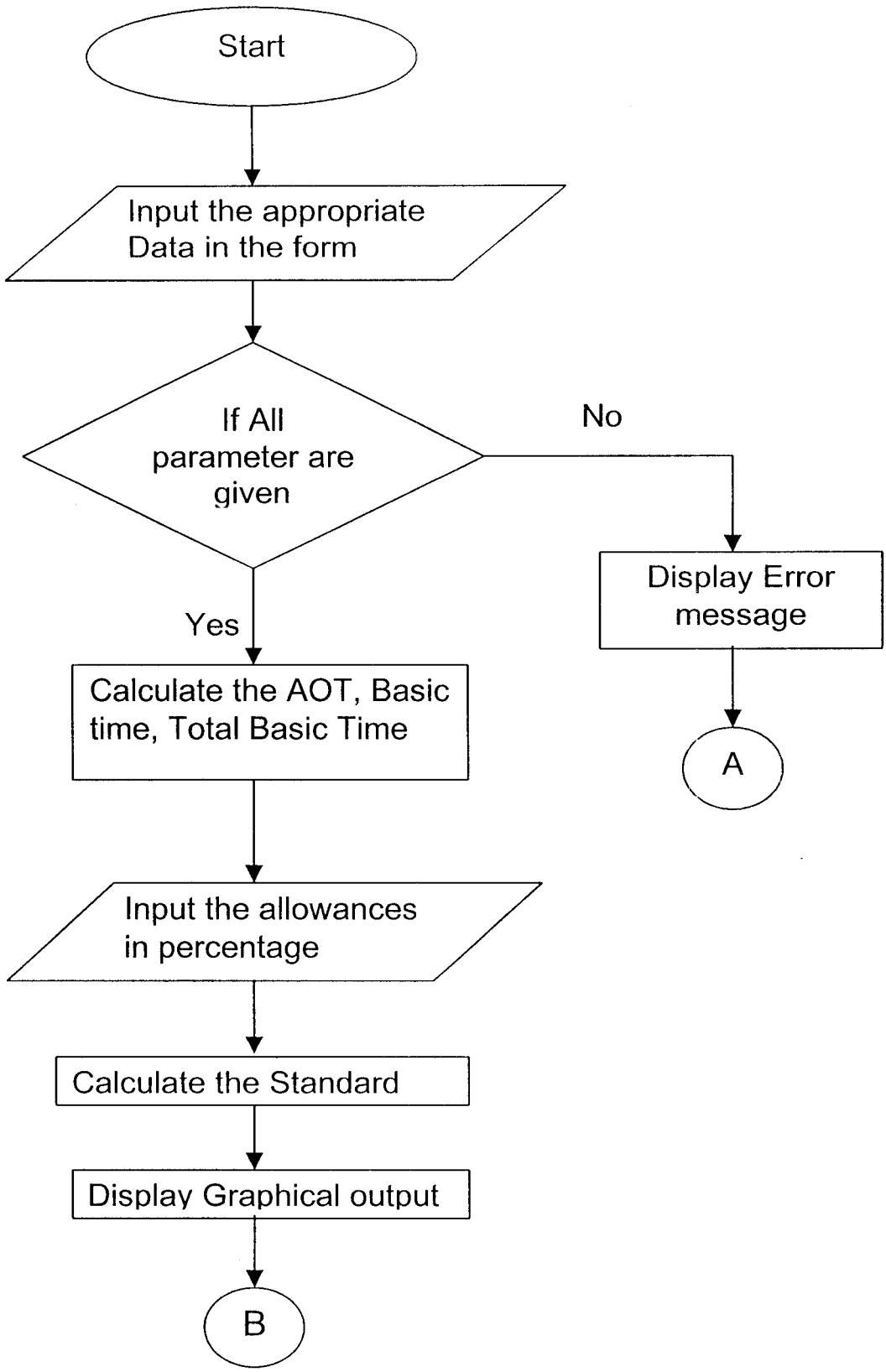
Step3: Select at random one of the elements from list B. The computer would be programmed to perform this random selection process. The only constraint is that the element selected must not cause the cycle T to be exceeded.

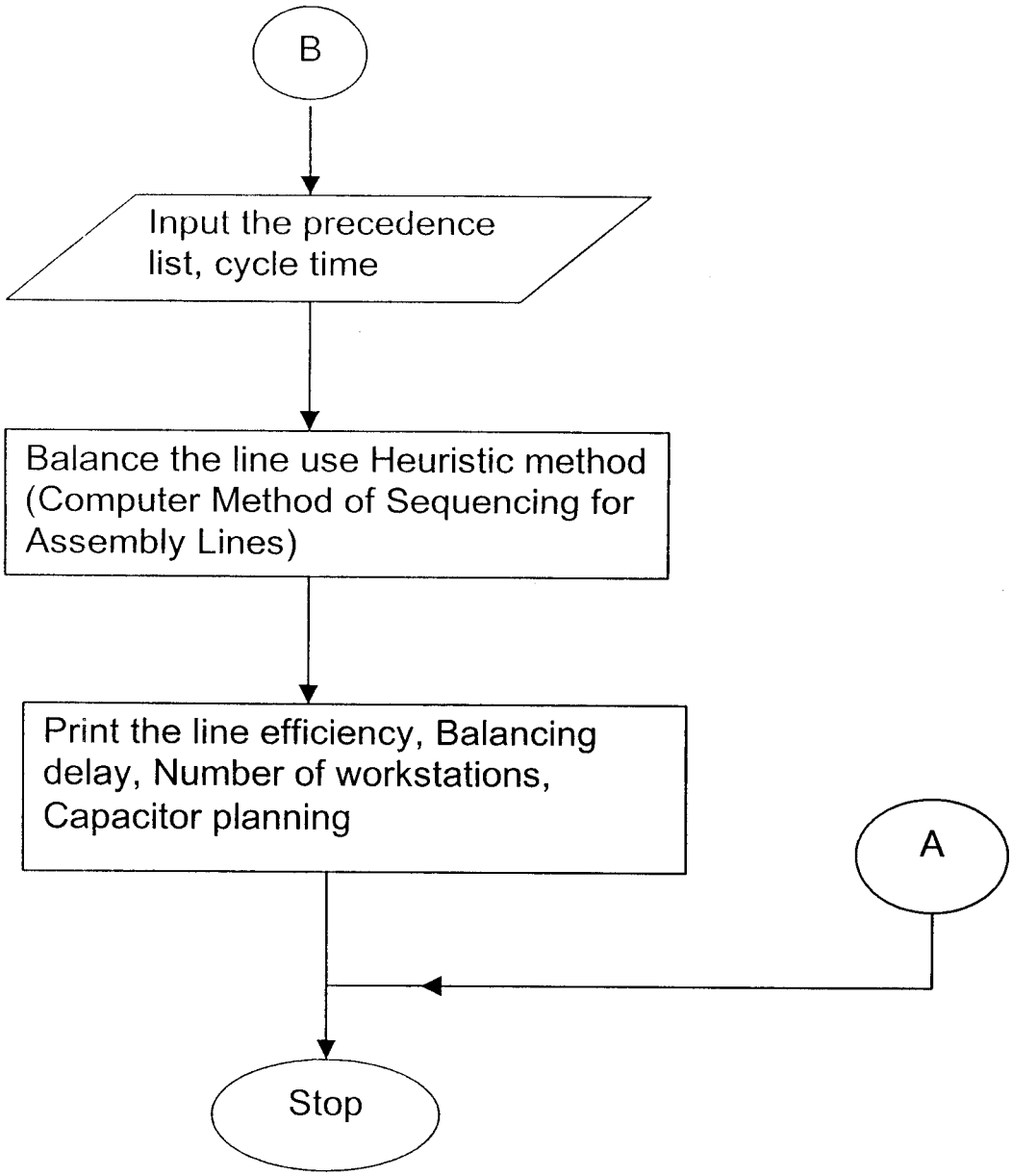
Step4: Eliminate the selected in step 3 from list A and B update both lists, if necessary. Updating may be needed because the selected element was probably an immediate predecessor for some other elements. Hence, there may be changes in the number of immediate predecessors for certain elements in list A; and there may now be some new elements having no immediate predecessors that should be added to list B.

Step5: Again select one of the elements from list B that is feasible for cycle time.

Step6: Repeat steps 4 and 5 until all elements have been allocated to stations within the T constraint.

Flowchart showing the main steps in software solution




B
Input the precedence list, cycle time
Balance the line use Heuristic method (Computer Method of Sequencing for Assembly Lines)
Print the line efficiency, Balancing delay, Number of workstations, Capacitor planning
A
Stop

6.3 CONSTRAINTS AND CONDITIONS

Measuring time with a stop watch

There are two methods of timing using a stopwatch. They are:

1. Fly back or snap back method.
2. Continuous or cumulative method.

In this project continuous method is followed. Here the stopwatch is started at the beginning of the first element. The watch runs continuously throughout the study. At the end of each element the watch readings are recorded on the study sheet. The time for each element is calculated by successive subtraction. The final reading of the stopwatch gives the total time. This is the observed time (O.T).

The constraints and conditions that are observed and how they are handled in the algorithm is given in the following paragraphs. Here the basic work elements are grouped into activity and this is used for balancing purpose as input.

I. Activity Time

This is used as basic input for assembly line balancing. The cycle restricts the activities being assigned; on activity time .it affects the balance delay also. In case, where activity time is greater than cycle time there are two cases:

a. If the activity can be done on subassembly remove it from the assembly line.
b. Allow the activity to be done by two persons in alteration with other activities to achieve utilization.

I. Predecessors

The activity can be assigned to a workstation only when all or its predecessors are covered before .It is the primary constraint .It restricts the number of activities available to a stage at that instant.

II. Cycle Time

These are the primary inputs of the balancing. When cycle time is given as the input it is used straight way for the balancing .It is the basic parameter of balancing. The cycle time is different from the station times and has to be equal to or greater than any of the station times.

Software Design

7.SOFTWARE DESIGN

7.1 Introduction

Manually balancing the line becomes a tedious process when there is any change in the task times as well as change in the requirement. Hence computerization is necessary to do balancing in a simple way. A software package is proposed, developed to resolve the problem.

This problem is done using Visual Basic 6 as front end and MS Access 2000 as back end. Visual Basic is one of the very useful and powerful front end created by Microsoft inc. it provides an easy user interface and an easy to develop environment for effective application. It has enhanced and rich OLE features and very good data handling capabilities. Visual Basic has lot of advantages like on line source code checking for syntax, rich graphical support, etc. it is a simple to create complex application in a faster way. It has advanced features like OLE automation, enhanced developed environment. Access is a build in Visual Basic data manager for Visual Basic.

An application is a collection of code and visual elements that work together as a single program. Developers can build and run applications within the development environment, while users usually run applications as executables files outside the development environment.

Creating an application in Visual Basic involves following steps like

- Designing user interface.
- Writing code
- Adding modules and supporting files
- Debugging
- Creating executables

7.2 Data base design

A database is a repository of collection of related data or facts. It arranges them in a specific structure. Data in a database is most commonly viewed in one or more two-dimensional tables, each consisting of columns and rows. The entire collection of related data in one table is referred to as a file or Table. Each row in a table represents a Record, which is a set of data for each database entry. Each table column

represents a Field, which groups each piece or item of data among the records into specific categories or types of data.

The data are stored in the back end i.e., MS Access 2000. Access is a build in Visual Basic data manager for Visual Basic. MS Access is very easy to use. For this software four tables were created namely,

For time study table

1. Master table
2. Master detailed table
3. Table (two... Eight)
4. Temporary table 1
5. Temporary table2

For assembly line balancing table

6. Line balance table
7. Table balance detailed
8. Temporary table
9. Temporary table1

1. Master database

This database is used to store the time study input table. It consists of 15 fields in this table.

S.No	Name of the field	Data type	Description
1	Mast ID	Auto number	Table identification
2	Mast Date	Date/time	Date
3	Mast RefNo	Number	Reference Number
4	Mast OptnNo	Text	Station number
5	Mast OptnName	Text	Operation name
6	Mast StudyNo	Text	Study number
7	Mast Line	Text	Line
8	Mast OptrNo	Text	Operator name
9	Mast TokenNo	Text	Token number
10	Mast Exp	Text	Experience
11	Mast Hf	Text	Product/part
12	Mast ScBy	Text	Studied by
13	Mast Calculate	Text	Calculate
14	Mast Rf	Number	Rating factor

15	Mast Allow	Number	Allowance

Table1. Time study form input table

2. Master detail database

This database store the input of element no, element description and activity time (no. Of cycles). There are 30 fields in this table.

S.No	Name of the field	Data type	Description
1	Det ID	Auto number	Table identification
2	Iden	Text	Calculated by Row/Column wise
3	Ele NO	Text	Element number
4	Ele des	Text	Element description
5	Mast ID	Number	Master identification column
6	A1..... A25	Number	Activities time input table

Table2. Time study form input table

3. Table (Two) database

This database store the observed times (i.e., Cycle timings). This table contains 27 fields.

S.No	Name of the field	Data type	Description
1	B ID	Auto number	Table identification
2	Mast ID	Number	Master identification column
3	B1... B25	Number	Activities time input table

Table 3. Time study form input table

Similarly to table three, four, five, six, seven and eight.

4. Temporary table 1

This database is used to generate the report. (I.e., printing purpose). It consists of 17 fields.

S.No	Name of the field	Data type	Description
1	T ID	Auto number	Table identification
2	Tdate	Date/time	Date
3	Sta No	Number	Station number
4	Operation	Text	Operation name
5	Study No	Text	Study number
6	Line	Text	Line
7	Product/part	Text	Product/part
8	Operator	Text	Operator
9	Token No	Text	Token number
10	Exp	Text	Experience
11	Study BY	Text	Study by
12	Rf	Number	Rating factor
13	TBT	Number	Total basic time
14	Allow	Number	Allowance
15	Allow Amt	Number	Allowance value
16	St	Number	Standard time
17	Refno	Number	Reference number

Table 4. Time study form output table

5. Temporary table 2

This database also is used to generate the report. (I.e., printing purpose). It consists of 6 fields.

S.No	Name of the field	Data type	Description
1	T1 ID	Auto number	Table identification
2	Ele NO	Text	Element number
3	Ele des	Text	Element description
4	AOT	Number	Average observed time

5	BT	Number	Basic time
6	Ref.No	Number	Reference number

Table 5. Time study form output table

Assembly line balancing tables

6. Line balance table

This database stores the input data for assembly line balancing table. There are 6 fields in this field.

S.No	Name of the field	Data type	Description
1	L ID	Auto number	Table identification
2	L RefNO	Number	Reference number
3.	L Date	Date/time	Date
4	L Product	Text	Product / part name
5	L Cycle	Number	Cycle time
6	L Shift	Number	Shift

Table 6. Activity input table

7. Line balance detailed table

This database store the input of station no, element no, element description, basic time, immediate predecessor etc (i.e. is the activity input sheet) it consists of 7 fields.

S.No	Name of the field	Data type	Description
1	Det ID	Auto number	Identification
2	L ID	Number	Table identification
3	Det Sta	Text	Station number
4	Det EleNO	Number	Element number
5	Det EleDes	Text	Element description
6	Det BT	Number	Basic time
7	Det Immpre	Text	Immediate predecessor

Table 7. Activity input table

8. Temporary table

This database also is used to generate the report. (I.e., printing purpose). It consists of 8 fields.

S.No	Name of the field	Data type	Description
1	Temp ID	Auto number	Identification
2	Product	Text	Product / part name
3	Nosta	Number	No .of stations
4	Line Eff	Number	Line efficiency
5	Bal Delay	Number	Balancing delay
6	Shift	Number	Shift
7	Pro cap	Number	Production capacity
8	Cycle	Number	Cycle time

Table 8. Summary table

9. Temporary table 1

This database also is used to generate the report activity balanced form. (I.e., printing purpose). It consists of 7 fields.

S.No	Name of the field	Data type	Description
1	Temp ID	Auto number	Identification
2	T sta	Text	Total no. Of station
3	T eleNo	Number	Element number
4	T Ele Des	Text	Element description
5	T BT	Number	Basic time
6	T Sta time	Number	Station time
7	T Rst time	Number	Remaining station time

Table 8. Balanced Activity table

7.3 Mode of Accession

In this software first form is the login form. It will ask the user to enter the **Username** and **Password**. After login correctly the main form, which contains the start button, will be displayed. The start button is placed in the left corner of the main form which when clicked will display the **POPUP** menu. The POPUP menu contains the following modules:

- Time study
- Assembly line balancing
- MS Project

If the user wants to work on any of the above modules he have to click the corresponding modules, which is displayed on the POPUP menu. The time study & assembly line balancing module contains the following six controls

1. Text box
2. Label box
3. Combo box
4. MS HFLEX Grid (Microsoft Hierarchical Flex Grid)
5. Command button &
6. List box

If the user wish to work with time study module he have to click the **New** button in the time study form. After clicking the new button, a new record that contains the new reference number, current date etc. will be displayed. If the user wants to go for first record he can use the **Top** button. In a similar way he can use the **Bottom** button to go for last record. The **Next** and **Previous** buttons used for seeing the successive and previous record from the current record respectively. The **Save** button is used for saving the entire operations that are done upto that time.

If the user wants to change or modify anything he can use **Edit** button. The **Delete** button is used for deleting a record entirely from the database after getting the confirmation from the user. The **Add** & **Graph** buttons is used for adding the records and plotting the graphs between various workstations Vs Standard times respectively.

Using **Print** button the user can take the hard copy of various results that are obtained from this software. The **Close** button is used for closing the

current module. The **Search** button especially used for searching a particular record among the displayed records. The **Balance** button is used for switching over to assembly line balancing module from the time study module. Hence it acts as a link between the two modules.

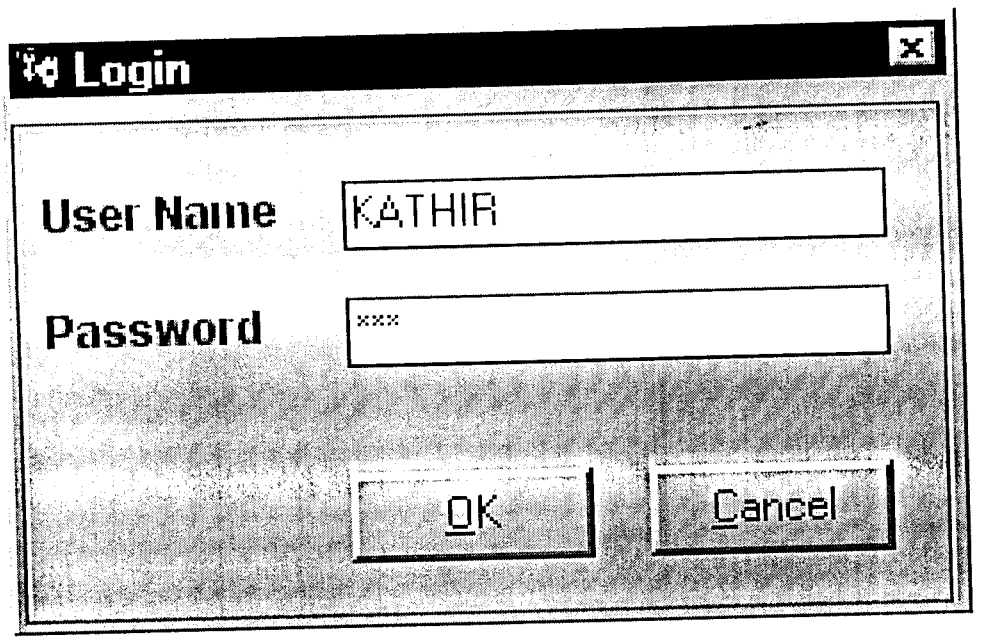
The user can work in a similar way for assembly line balancing module as the buttons used in this module performs the same function as one that is used in time study. In addition to the above-mentioned buttons this module also has the specific buttons that are relevant for this module. The **Graph** button in this module used for plotting the graphs between station numbers Vs time in minutes for both before and after line balancing. The **Summary** button as the name suggests displays the brief results of all the modules.

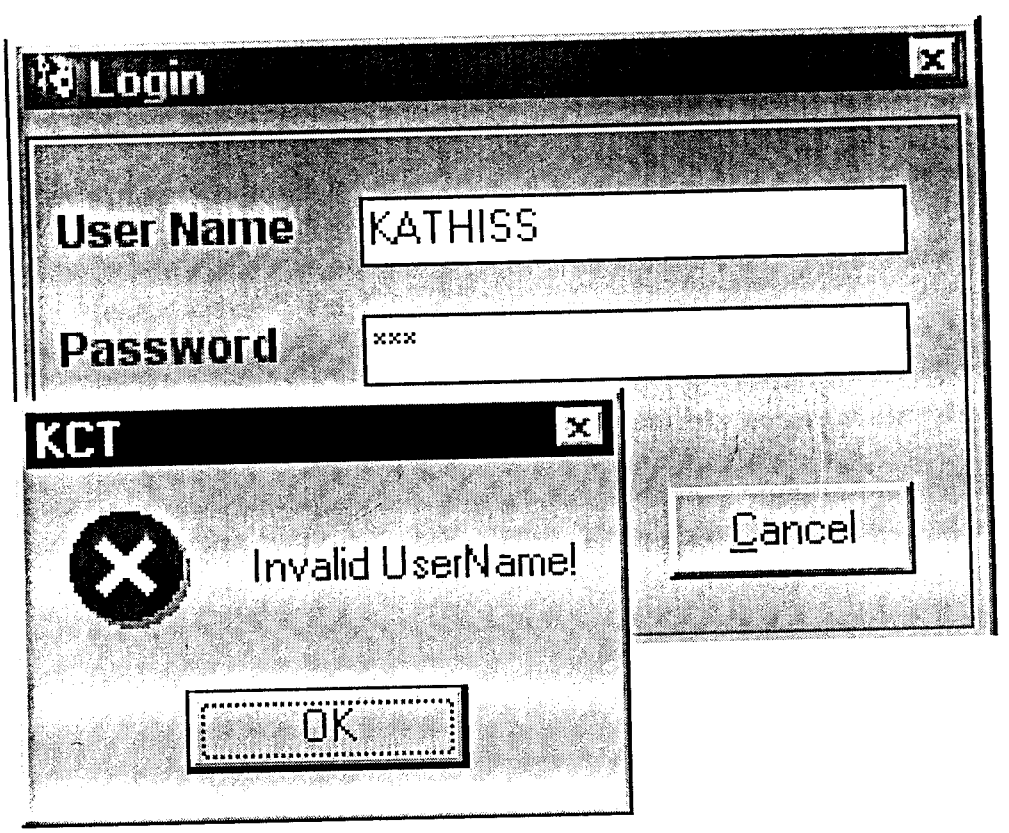
The user can also use the **ESC** button in the keyboard for calculations or adding or deleting the rows when the control is in MS HFLEX Grid.

MS Project is a exclusive software which is used widely for plotting Gantt charts, PERT charts, Resource graphs etc. So this software is used for visual representation of networks with their precedence relationship between the element number and sequence of operations.

After working with all the modules, the user can come out of this software by clicking the **Exit** button, which is placed in the POPUP menu.

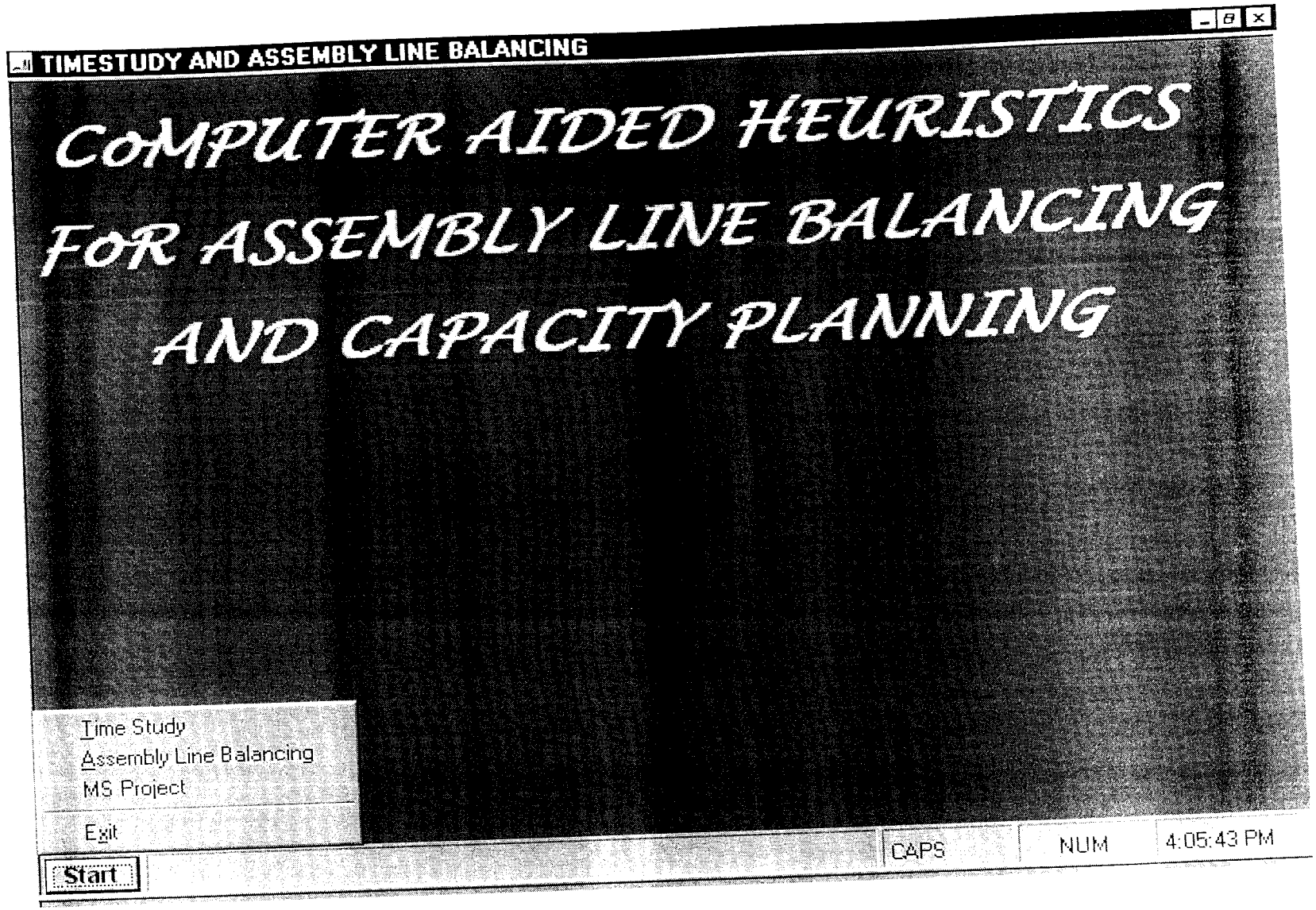
Forms and Coding


Login
User Name
KATHIR
Password
***
OK
Cancel

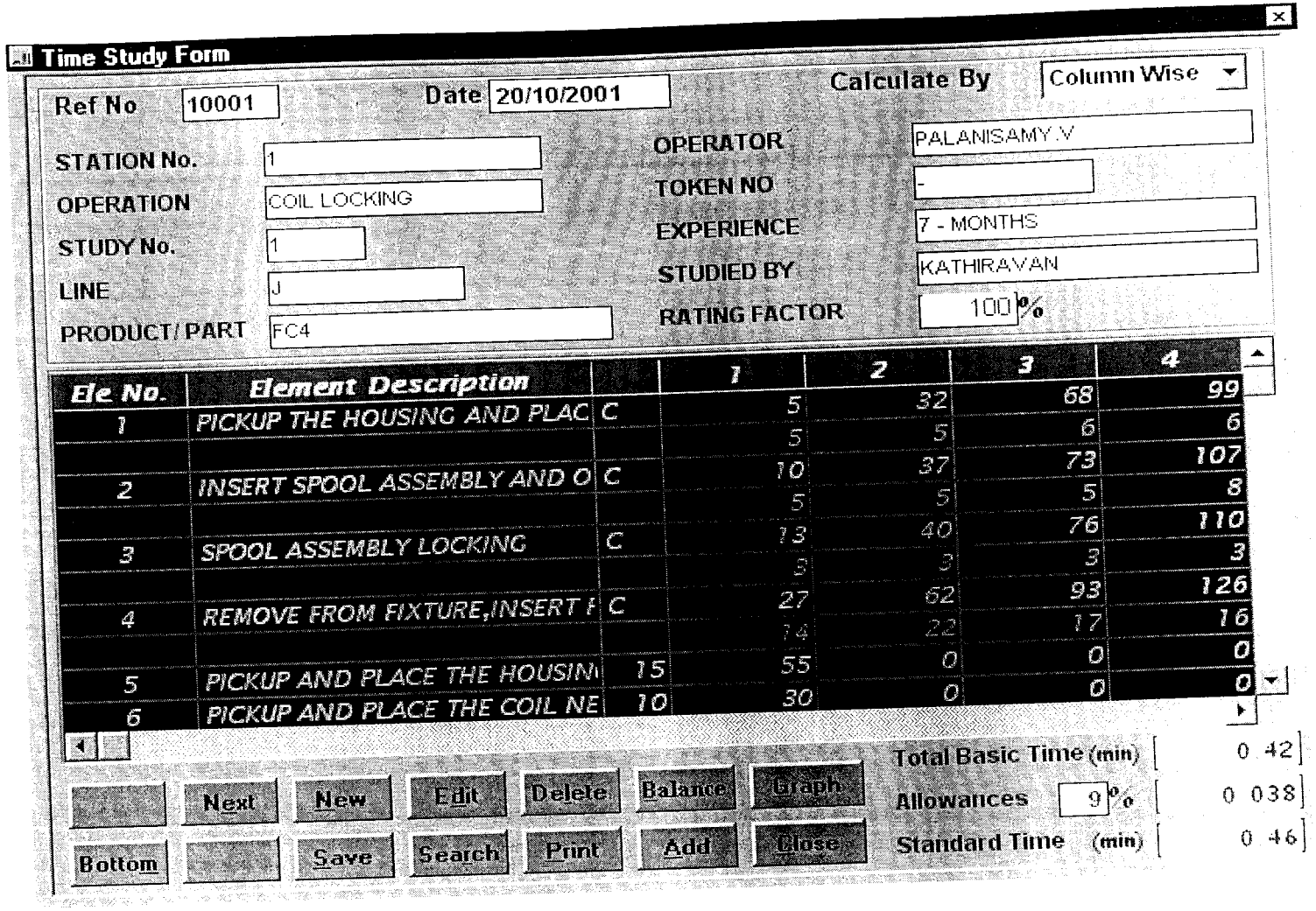

Login
User Name
KATHISS
Password
***
KCT
Invalid UserName!
OK
Cancel

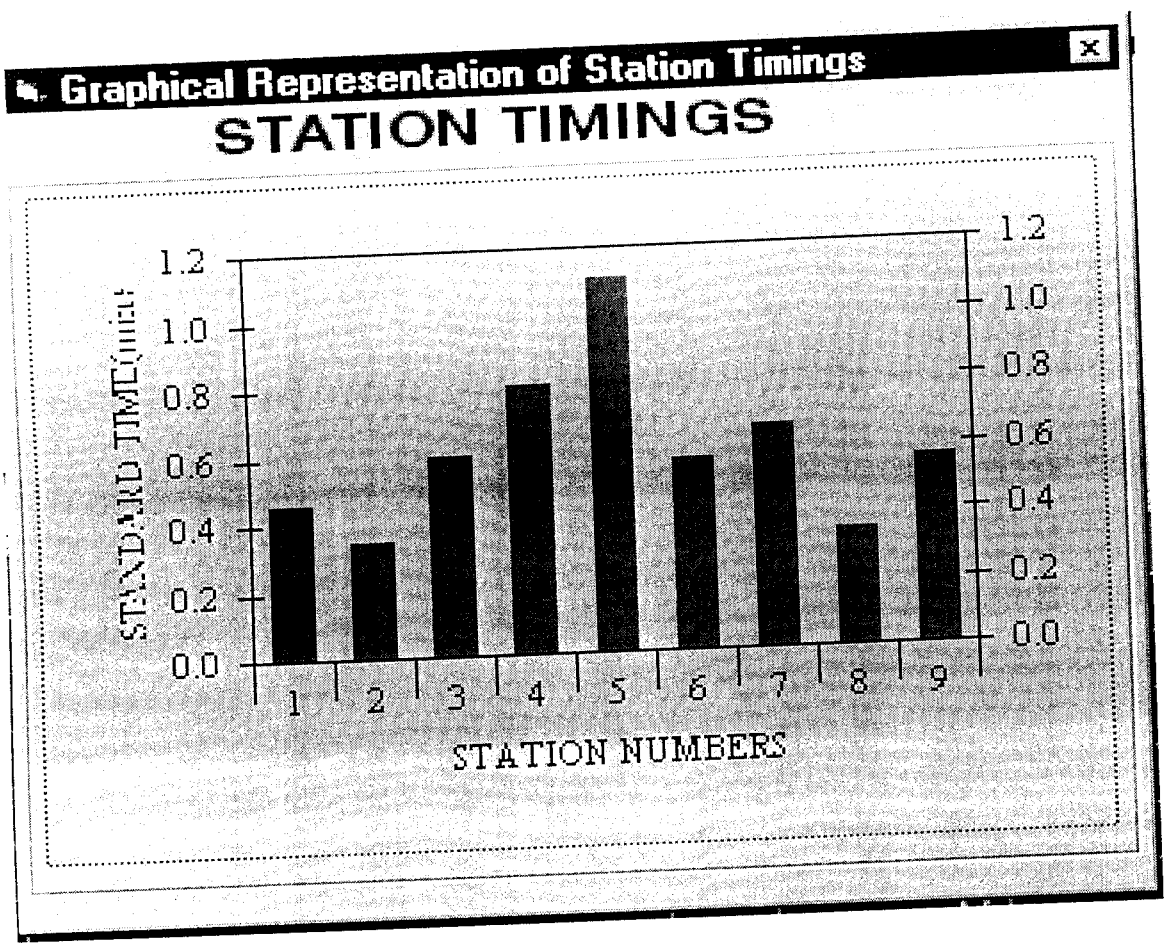
```
Private Sub cmdOK_Click()
If txtUsrName.Text = "Kathir" Or txtUsrName.Text = "KATHIR" Or txtUsrName.Text =
                                                                    "kathir" Then
   If txtPassword.Text="Kct" Or txtPassword.Text="KCT" Or txtPassword.Text="kct" Then
        Unload frmLogin
           frmMDI.Show
   Else
           MsgBox "Invalid Password!", vbCritical, "KCT"
           txtPassword.SelStart = 0
           txtPassword.SelLength = Len(txtPassword.Text)
           txtPassword.SetFocus
   End If
Else
   MsgBox "Invalid UserName!", vbCritical, "KCT"
   txtUsrName.SelStart = 0
   txtUsrName.SelLength = Len(txtUsrName.Text)
   txtUsrName.SetFocus
End If
End Sub


Private Sub cmdCancel_Click()
End
End Sub


Private Sub txtPassword_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
   If txtPassword.Text="Kct" Or txtPassword.Text="KCT" Or txtPassword.Text="kct" Then
           cmdOK.Value = True
   Else
           MsgBox "Invalid Password!", vbCritical, "KCT"
           txtPassword.SelStart = 0
           txtPassword.SelLength = Len(txtPassword.Text)
           txtPassword.SetFocus
   End If
End If
End Sub
```

```
Private Sub txtUsrName_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
If txtUsrName.Text = "Kathir" Or txtUsrName.Text = "KATHIR" Or txtUsrName.Text =
                                                                    "kathir" Then
   txtPassword.SetFocus
Else
   MsgBox "Invalid UserName!", vbCritical, "KCT"
   txtUsrName.SelStart = 0
   txtUsrName.SelLength = Len(txtUsrName.Text)
   txtUsrName.SetFocus
End If
End If
End Sub
```


TIMESTUDY AND ASSEMBLY LINE BALANCING
COMPUTER AIDED HEURISTICS
FOR ASSEMBLY LINE BALANCING
AND CAPACITY PLANNING
Time Study
Assembly Line Balancing
MS Project
Exit
Start
CAPS
NUM
4:05:43 PM

```
Private Sub cmdStart_Click()
        PopupMenu mnuMain, , 10, 6500
End Sub

Private Sub mnuALB_Click()
        frmLine.Show
End Sub

Private Sub mnuExit_Click()
        Dim Msg, Style, Title, Help, Ctxt, Response, MyString
        Msg = "Would you like to exit from Timestudy Software ?"   ' Define message.
        Style = vbYesNo + vbCritical + vbDefaultButton2   ' Define buttons.
        Title = "Time Study"   ' Define title.
        Response = MsgBox(Msg, Style, Title)
If Response = vbYes Then   ' User chose Yes.
  End
Else
   cmdStart.SetFocus
End If
End Sub

Private Sub mnuTS_Click()
frmTime.Show
End Sub.
Private Sub Timer1_Timer()
StatusBar1.Panels(5) = Time
End Sub
```


Time Study Form
Ref No 10001
Date 20/10/2001
Calculate By Column Wise
STATION No. 1
OPERATION COIL LOCKING
STUDY No. 1
LINE J
PRODUCT/ PART FC4
OPERATOR PALANISAMY.V
TOKEN NO -
EXPERIENCE 7 - MONTHS
STUDIED BY KATHIRAVAN
RATING FACTOR 100 %
Ele No. | Element Description | | 1 | 2 | 3 | 4
1 | PICKUP THE HOUSING AND PLAC | C | 5 | 32 | 68 | 99
| | | 5 | 5 | 6 | 6
| | | 10 | 37 | 73 | 107
2 | INSERT SPOOL ASSEMBLY AND O | C | 5 | 5 | 5 | 8
| | | 13 | 40 | 76 | 110
3 | SPOOL ASSEMBLY LOCKING | C | 3 | 3 | 3 | 3
| | | 27 | 62 | 93 | 126
4 | REMOVE FROM FIXTURE,INSERT F | C | 14 | 22 | 17 | 16
| | | 55 | 0 | 0 | 0
5 | PICKUP AND PLACE THE HOUSIN | 15 | 30 | 0 | 0 | 0
6 | PICKUP AND PLACE THE COIL NE | 10
Next
New
Edit
Delete
Balance
Graph
Bottom
Save
Search
Print
Add
Close
Total Basic Time (min) 0 42
Allowances 9 % 0 038
Standard Time (min) 0 46


Graphical Representation of Station Timings
STATION TIMINGS
STANDARD TIME(HRS)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
9
STATION NUMBERS

```
Private Sub cmdTop_Click()
        Call Mo("Top", 2)
        Cal.Enabled = False
        If Edit = False Then
                mnuAdd.Enabled = False
                mnuDel.Enabled = False
        End If
End Sub

Private Sub cmdBottom_Click()
        Call Mo("Bottom", 2)
        Cal.Enabled = False
        If Edit = False Then
                mnuAdd.Enabled = False
                mnuDel.Enabled = False
        End If
End Sub

Private Sub cmdNext_Click()
        Call Mo("Next", 2)
        Cal.Enabled = False
        If Edit = False Then
                mnuAdd.Enabled = False
                mnuDel.Enabled = False
        End If
End Sub

Private Sub cmdPrevious_Click()
        Call Mo("Pre", 2)
        Cal.Enabled = False
        If Edit = False Then
                mnuAdd.Enabled = False
                mnuDel.Enabled = False
        End If
End Sub

Private Sub cmdNew_Click()
    cboCalculate.Locked = False
    Neww = True
    Cal.Enabled = True
    mnuAdd.Enabled = True
    mnuDel.Enabled = True
    cmdSave.Enabled = True
    cmdPrint.Enabled = False
    cmdAdd.Enabled = False
    cmdDelete.Enabled = False
```

```
cmdSearch.Enabled = False
cmdBal.Enabled = False
cmdEdit.Enabled = False
cmdTop.Enabled = False
cmdBottom.Enabled = False
cmdNext.Enabled = False
cmdPrevious.Enabled = False
txtID.Text = ""
mskDate.Text = Format(Date, "dd/mm/yyyy")
txtRefNo.Text = ""
txtOptnNo.Text = ""
txtOptnName.Text = ""
txtStudyNo.Text = ""
txtLine.Text = ""
txtOptrName.Text = ""
txtTokenNo.Text = ""
txtExp.Text = ""
txtHF.Text = ""
txtSCB.Text = ""
txtPer.Text = ""
txtAllow.Text = ""
txtBT.Text = ""
txtST.Text = ""
cboCalculate.Text = cboCalculate.List(0)
txtRF.Text = ""
Call clsTime.NewRef
txtRefNo.Text = clsTime.Mast_RefNo
HFGTime.Clear
HFGTime.Rows = 2
HFGTime.Cols = 213
HFGTime.ColWidth(0) = 1200
HFGTime.ColWidth(1) = 3000
HFGTime.ColWidth(2) = 500
HFGTime.ColWidth(HFGTime.Cols - 1) = 0
HFGTime.ColWidth(HFGTime.Cols - 2) = 0
HFGTime.ColWidth(HFGTime.Cols - 3) = 0
HFGTime.ColWidth(HFGTime.Cols - 4) = 0
HFGTime.ColWidth(HFGTime.Cols - 5) = 0
HFGTime.ColWidth(HFGTime.Cols - 6) = 0
HFGTime.ColWidth(HFGTime.Cols - 7) = 0
HFGTime.ColWidth(HFGTime.Cols - 8) = 0
HFGTime.Row = 0
HFGTime.Col = 0
HFGTime.Text = "Ele No."
HFGTime.Col = 1
HFGTime.Text = "Element Description"
```

```
HFGTime.Col = HFGTime.Cols - 10
HFGTime.Text = "AOT(Mts)"
HFGTime.Col = HFGTime.Cols - 9
HFGTime.Text = "BT(Mts)"
HFGTime.Row = 0
HFGTime.Col = 2
HFGTime.Text = ""
If cboCalculate.Text = "Column Wise" Then
   HFGTime.TextMatrix(1, 2) = "C"
ElseIf cboCalculate.Text = "Row Wise" Then
   HFGTime.TextMatrix(1, 2) = "R"
End If
For intCol = 3 To HFGTime.Cols - 11
   HFGTime.Col = intCol
   HFGTime.Text = intCol - 2
Next intCol
For intCol = 0 To HFGTime.Cols - 3
   HFGTime.Row = 0
   HFGTime.Col = intCol
   HFGTime.CellAlignment = flexAlignCenterCenter
 Next intCol
cmdClose.Caption = "&Cancel"
mskDate.SetFocus
End Sub

Private Sub cmdSave_Click()
        cmdSave.Enabled = False
        cmdPrint.Enabled = True
        cmdAdd.Enabled = True
        cmdDelete.Enabled = True
        cmdSearch.Enabled = True
        cmdBal.Enabled = True
        cmdNew.Enabled = True
        cmdTop.Enabled = False
        cmdBottom.Enabled = False
        cmdNext.Enabled = False
        cmdPrevious.Enabled = False
        Set clsTime = New clsTimeStudy
        Screen.MousePointer = vbHourglass
        clsTime.Mast_ID = Val(txtID.Text)
        clsTime.Mast_Date = mskDate.Text
        clsTime.Mast_RefNo = txtRefNo.Text
        clsTime.Mast_OptnNo = txtOptnNo.Text
        clsTime.Mast_OptnName = txtOptnName.Text
        clsTime.Mast_StudyNo = txtStudyNo.Text
        clsTime.Mast_Line = txtLine.Text
```

```
        clsTime.Mast_OptrName = txtOptrName.Text
        clsTime.Mast_TokenNo = txtTokenNo.Text
        clsTime.Mast_Exp = txtExp.Text
        clsTime.Mast_HF = txtHF.Text
        clsTime.Mast_SCB = txtSCB.Text
        clsTime.Mast_Calculate = cboCalculate.Text
        clsTime.Mast_RF = Val(txtRF.Text)
        clsTime.Mast_Allow = Val(txtPer.Text)
                Call clsTime.SaveData(txtID.Text)
                Call clsTime.Sel(txtID.Text)
        txtID.Text = clsTime.mvartxtTest
        Call clsTime.SaveGrid(frmTime.HFGTime, txtID.Text)
        Edit = False
                Call Form_Load
        frmTime.cmdTop.Enabled = True
        frmTime.cmdBottom.Enabled = True
        frmTime.cmdNext.Enabled = True
        frmTime.cmdPrint.Enabled = True
        frmTime.cmdPrevious.Enabled = True
        frmTime.cmdDelete.Enabled = True
        frmTime.cmdEdit.Enabled = True
        frmTime.cmdClose.Caption = "&Close"
        frmTime.cmdNew.Enabled = True
        frmTime.cmdSearch.Enabled = True
        Screen.MousePointer = vbDefault
        Cal.Enabled = False
        mnuAdd.Enabled = False
        mnuDel.Enabled = False
End Sub

Private Sub cmdEdit_Click()
        cboCalculate.Locked = False
        cmdSave.Enabled = True
        cmdPrint.Enabled = False
        cmdAdd.Enabled = False
        cmdDelete.Enabled = False
        cmdSearch.Enabled = False
        cmdBal.Enabled = False
        cmdNew.Enabled = False
        Edit = True
        Cal.Enabled = True
        mnuAdd.Enabled = True
        mnuDel.Enabled = True
          Call Mo("Bottom", 2)
        cmdClose.Caption = "&Cancel"
        mskDate.SetFocus
```

```
End Sub

Private Sub cmdSearch_Click()
        HFGSearch.Visible = True
        HFGSearch.Cols = 3
        HFGSearch.Rows = 2
        Call clsTime.Search(HFGSearch)
        HFGSearch.ColWidth(0) = 1000
        HFGSearch.ColWidth(1) = 3000
        HFGSearch.ColWidth(2) = 0
        HFGSearch.Row = 0
        HFGSearch.ColSel = 2
        Cal.Enabled = False
        mnuAdd.Enabled = False
        mnuDel.Enabled = False
        HFGSearch.SetFocus
  End Sub


  Private Sub cmdDelete_Click()
        Dim Msg, Style, Title, Response
        Msg = "Are you sure you want to delete Reference No : " & txtRefNo.Text & " ?"
        'Style = vbYesNo + vbQuestion + vbDefaultButton2
        Title = "Time Study"
        Response = MsgBox(Msg, Style, Title)
        If Response = vbYes Then   ' User chose Yes.
           Set clsTime = New clsTimeStudy
           Call clsTime.Delete(Val(txtID.Text))
           Call Form_Load
        End If
        Cal.Enabled = False
        mnuAdd.Enabled = False
        mnuDel.Enabled = False
   End Sub


   Private Sub cmdPrint_Click()
   Cal.Enabled = False
   mnuAdd.Enabled = False
   mnuDel.Enabled = False


   Call clsTime.SavePrint1(mskDate.Text, Val(txtOptnNo.Text), txtOptnName.Text,
                           txtStudyNo.Text, txtLine.Text, txtHF.Text, txtOptrName.Text,
```

```
                    txtTokenNo.Text, txtExp.Text, txtSCB.Text, Val(txtRF.Text),
                    Val(txtBT.Text), Val(txtPer.Text), Val(txtAllow.Text),
                    Val(txtST.Text), Val(txtRefNo.Text))
Call clsTime.SavePrint2(HFGTime, Val(txtRefNo.Text))
    If EnvRepts.rscmdStudy.State = adStateClosed Then
        EnvRepts.rscmdStudy.Open
    End If
    EnvRepts.rscmdStudy.Requery (1)
    RptTime.Show
End Sub

Private Sub cmdBal_Click()
    Cal.Enabled = False
    mnuAdd.Enabled = False
    mnuDel.Enabled = False
    frmLine.Show
End Sub

Private Sub cmdAdd_Click()
    Dim i As Integer
    Dim Row As Integer
    Dim Rows  As Integer
    Dim Opt As Boolean
    If Val(txtRefNo.Text) > 0 Then
        If Len(txtOptnNo.Text) <> 0 Then
            If Add = False Then
                frmLine.cmdNew.Value = True
                Add = True
            End If
            Row = frmLine.LineB.Rows - 1
    For i = 1 To frmLine.LineB.Rows - 1
      If frmLine.LineB.TextMatrix(i, 0) = txtOptnNo.Text Then
            Opt = True
      End If
    Next i
    For i = 1 To HFGTime.Rows - 1
    If HFGTime.TextMatrix(i, 0) <> "Nil" And HFGTime.TextMatrix(i, 0) <> "NIL" And
                                        HFGTime.TextMatrix(i, 0) <> "nil" Then
      If HFGTime.TextMatrix(i, 2) <> "" And Opt = False Then
        frmLine.LineB.TextMatrix(Row, 0) = txtOptnNo.Text
        frmLine.LineB.TextMatrix(Row, 1) = HFGTime.TextMatrix(i, 0)
        frmLine.LineB.TextMatrix(Row, 2) = HFGTime.TextMatrix(i, 1)
        frmLine.txtProduct.Text = frmTime.txtHF.Text
        If HFGTime.TextMatrix(i, 2) = "C" Or HFGTime.TextMatrix(i, 2) = "R" Then
     If HFGTime.TextMatrix(i + 1, HFGTime.Cols - 9) <> "" Then
        frmLine.LineB.TextMatrix(Row, 3)=HFGTime.TextMatrix(i+1,HFGTime.Cols - 9)
```

```
                End If
            ElseIf HFGTime.TextMatrix(i, HFGTime.Cols - 9) <> "" Then
              frmLine.LineB.TextMatrix(Row, 3) = HFGTime.TextMatrix(i, HFGTime.Cols - 9)
          End If
            frmLine.LineB.Rows = frmLine.LineB.Rows + 1
            Row = Row + 1
        End If
    End If
    Next i
    If Opt = True Then
       MsgBox "Already Selected.", vbExclamation, "ALBS : Message"
    ElseIf Opt = False Then
          HFGTemp.TextMatrix(HFGTemp.Rows - 1, 0) = txtOptnNo.Text
          HFGTemp.TextMatrix(HFGTemp.Rows - 1, 1) = txtST.Text
          HFGTemp.TextMatrix(HFGTemp.Rows - 1, 2) = txtRefNo.Text
          HFGTemp.Rows = HFGTemp.Rows + 1
          List1.AddItem txtRefNo.Text
    End If
    For i = 1 To frmLine.LineB.Cols - 1
       frmLine.LineB.ColAlignmentFixed = flexAlignCenterCenter
       frmLine.LineB.ColAlignment = flexAlignLeftCenter
    Next i
       frmLine.LineB.ColAlignment(0) = flexAlignCenterCenter
       frmLine.LineB.ColAlignment(1) = flexAlignCenterCenter
       frmLine.LineB.ColAlignment(3) = flexAlignRightCenter
       frmLine.LineB.MergeCells = 1
       frmLine.LineB.MergeCol(0) = True
    Else
       MsgBox "Station No. Should not be blank.", vbCritical, "Time Study"
    End If
    Else
       MsgBox "Reference No. Should not be blank (or) 0.", vbCritical, "Time Study"
    End If
    End Sub

    Private Sub cmdGraph_Click()
          Dim Column As Integer
          Dim Row, i As Integer
          Cal.Enabled = False
          mnuAdd.Enabled = False
          mnuDel.Enabled = False
          If List1.ListCount > 0 Then
            With frmTGraph.MSChart1
              .chartType = VtChChartType2dBar
              .ColumnCount = 1
              .RowCount = HFGTemp.Rows - 1
```

```
            For Column = 1 To 1
               i = 0
               For Row = 1 To HFGTemp.Rows - 1
                  .Column = Column
                  .Row = Row
                  .RowLabel = "" & HFGTemp.TextMatrix(i, 0) & ""
                  .Data = HFGTemp.TextMatrix(i, 1)
                  i = i + 1
               Next Row
            Next Column
         End With
         frmTGraph.Show
      Else
      MsgBox "This operation need you want to add some values.", vbCritical,"Time Study"
      End If
End Sub

Private Sub cmdClose_Click()
      If cmdClose.Caption = "&Close" Then
         Unload frmTime
      ElseIf cmdClose.Caption = "&Cancel" Then
         cmdClose.Caption = "&Close"
         Edit = False
         Neww = False
         cmdSave.Enabled = False
         cmdPrint.Enabled = True
         cmdAdd.Enabled = True
         cmdDelete.Enabled = True
         cmdSearch.Enabled = True
         cmdBal.Enabled = True
         cmdNew.Enabled = True
         cmdEdit.Enabled = True
         cmdTop.Enabled = False
         cmdBottom.Enabled = False
         cmdNext.Enabled = False
         cmdPrevious.Enabled = False
         Call Form_Load
      End If
End Sub



Private Sub txtSel_KeyPress(KeyAscii As Integer)
   If KeyAscii = 13 Then
```

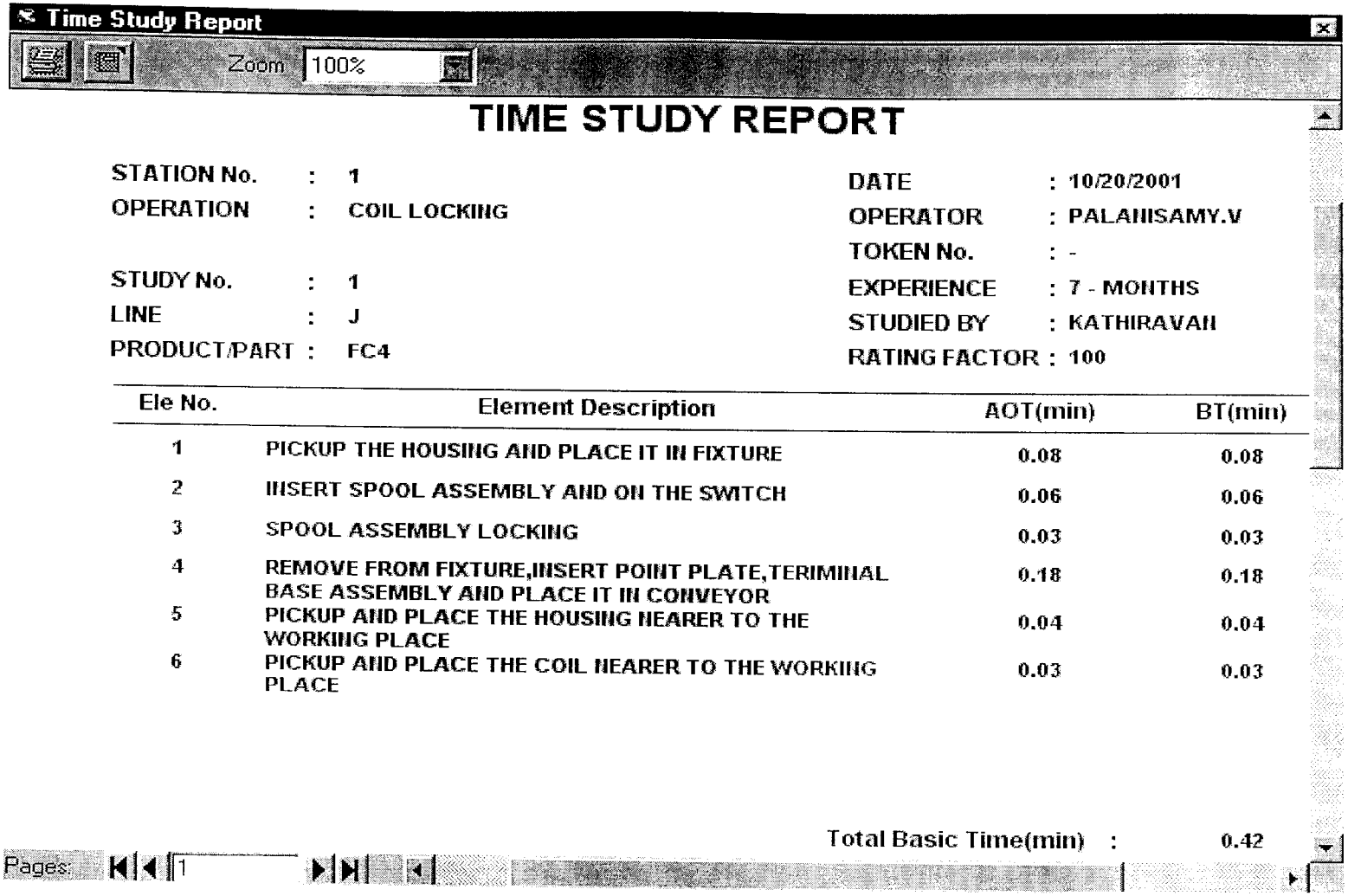
```
        Dim Srow As Integer
        Srow = HFGTime.Row
        HFGTime.Text = txtSel.Text
        If HFGTime.Col < 3 Then
           HFGTime.ColAlignment(HFGTime.Col) = flexAlignLeftCenter
        ElseIf HFGTime.Col >= 3 Then
           HFGTime.ColAlignment(HFGTime.Col) = flexAlignRightCenter
        End If
           HFGTime.ColAlignment(0) = flexAlignCenterCenter
        txtSel.Visible = False
        HFGTime.SetFocus
        KeyAscii = 0
   End If
End Sub

Private Sub HFGTime_KeyPress(KeyAscii As Integer)
  If KeyAscii = 13 Then
      If HFGTime.MouseRow < 0 Or HFGTime.MouseCol < 0 Then
         txtSel.Visible = False
      Exit Sub
      End If
    If HFGTime.Col >= 0 Then
      txtSel.Visible = True
      txtSel.Text = ""
      txtSel.Visible = True
      txtSel.Top = HFGTime.Top + HFGTime.CellTop
      txtSel.Left = HFGTime.Left + HFGTime.CellLeft
      txtSel.Width = HFGTime.CellWidth - 10
      txtSel.Height = HFGTime.CellHeight - 10
      txtSel.Text = HFGTime.Text
      txtSel.Visible = True
      txtSel.SetFocus
    End If
  End If
  If KeyAscii = 27 Then
     PopupMenu mnuFlex, , 240, 4900
  End If
End Sub



Private Sub cmdCalculate_Click()
Dim Srow As Integer
```

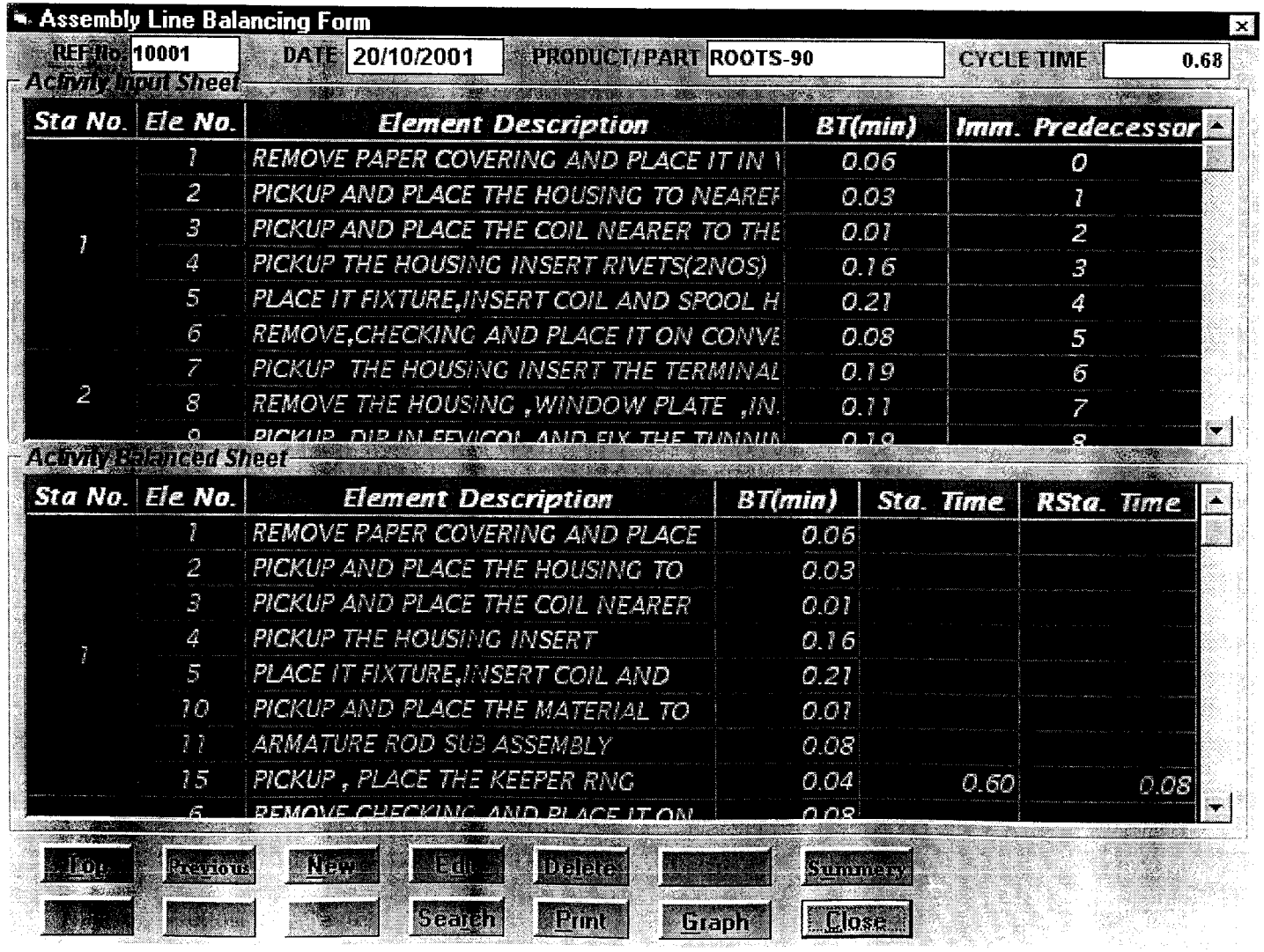
```
Dim SCol As Integer
Dim PreVal As Double
Dim i As Integer
On Error GoTo EndD:
If txtRefNo.Text <> "" And txtRefNo.Text <> 0 Then
Screen.MousePointer = vbHourglass
If cboCalculate.Text = "Column Wise" Then
For Srow = 1 To HFGTime.Rows - 1
   If HFGTime.TextMatrix(Srow, 2) = "C" Then
      HFGTime.Rows = HFGTime.Rows + 1
   End If
Next Srow
For SCol = 3 To 200 + 2
   HFGTime.Col = SCol
 If SCol = 3 Then
   For Srow = 1 To HFGTime.Rows - 1
      HFGTime.Row = Srow
   If HFGTime.TextMatrix(Srow, 2) = "C" Then
         HFGTime.RowPosition(HFGTime.Rows - 1) = HFGTime.Row + 1
      If Srow = 1 Then
         HFGTime.TextMatrix(Srow + 1, 3) = HFGTime.TextMatrix(Srow, 3)
         HFGTime.TextMatrix(Srow + 1, 2) = ""
      ElseIf Srow > 1 Then
         HFGTime.TextMatrix(Srow + 1, 3) = Val(HFGTime.TextMatrix(Srow, 3)) -
                                           Val(HFGTime.TextMatrix(Srow - 2, 3))
         HFGTime.TextMatrix(Srow + 1, 2) = ""
      End If
   End If
   Next Srow
 ElseIf SCol > 3 Then
   For i = 1 To HFGTime.Rows - 1
      If HFGTime.TextMatrix(i, 2) = "C" Then
         If HFGTime.TextMatrix(i, SCol - 1) <> "" Then
            PreVal = HFGTime.TextMatrix(i, SCol - 1)
         End If
      End If
   Next i
   For Srow = 1 To HFGTime.Rows - 1
      HFGTime.Row = Srow
      If Srow = 2 And HFGTime.TextMatrix(Srow, 2) = "" Then
         If HFGTime.TextMatrix(Srow - 1, SCol) <> "" And Val(HFGTime.TextMatrix(Srow - 1,
                                                            SCol)) <> 0 Then
           HFGTime.TextMatrix(Srow,SCol)=Val(HFGTime.TextMatrix(Srow-1,SCol)) - PreVal
         Else
            HFGTime.TextMatrix(Srow, SCol) = 0
         End If
```

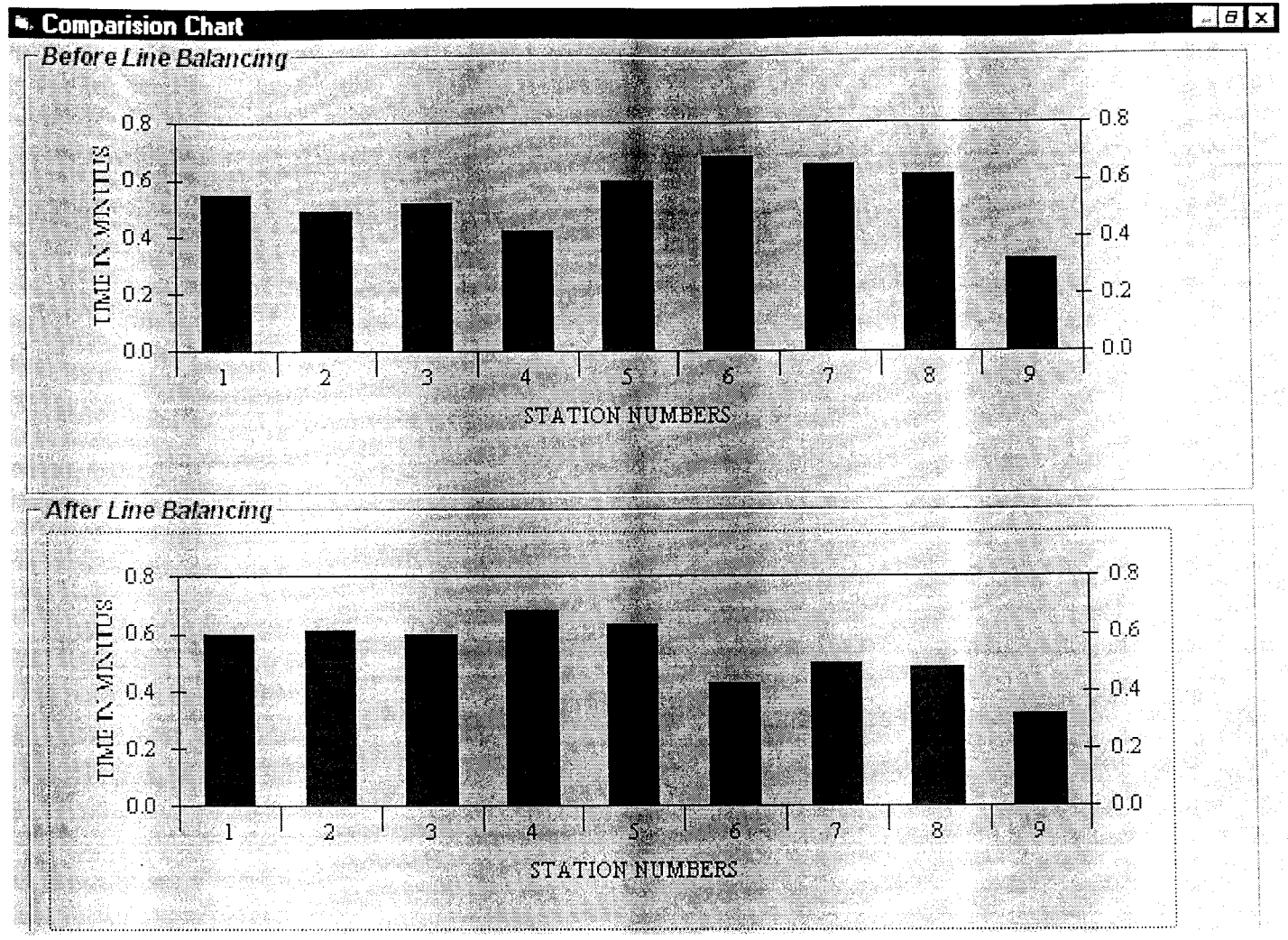
```
      ElseIf Srow > 2 And HFGTime.TextMatrix(Srow, 2) = "" Then
        HFGTime.TextMatrix(Srow, SCol) = Val(HFGTime.TextMatrix(Srow - 1, SCol)) -
                                         Val(HFGTime.TextMatrix(Srow - 3, SCol))
      End If
    Next Srow
End If
Next SCol
For Srow = 1 To HFGTime.Rows - 1
   For SCol = 3 To HFGTime.Cols - 11
      If HFGTime.TextMatrix(Srow, 2) = "" And Val(HFGTime.TextMatrix(Srow,SCol))>0 Then
         Summ = Summ + Val(HFGTime.TextMatrix(Srow, SCol))
         Countt = Countt + 1
      End If
      If HFGTime.TextMatrix(Srow, 0)<>"" And Val(HFGTime.TextMatrix(Srow,SCol))>0 Then
         Summ = Summ + Val(HFGTime.TextMatrix(Srow, SCol))
         Countt = Countt + 1
      End If
   Next SCol
      If HFGTime.TextMatrix(Srow, 2) = "" Then
       If HFGTime.TextMatrix(Srow, 3) <> "" And HFGTime.TextMatrix(Srow - 1,3)<>"" Then
         HFGTime.TextMatrix(Srow,HFGTime.Cols-10) = Format(Summ/(Countt*100), "0.00")
         HFGTime.TextMatrix(Srow, HFGTime.Cols - 9) =
      Format((Val(HFGTime.TextMatrix(Srow,HFGTime.Cols-10)))*Val(txtRF.Text)/100,0.00")
       End If
      End If
      If HFGTime.TextMatrix(Srow, 0) <> "" And Val(HFGTime.TextMatrix(Srow, 2)) > 0 Then
       If HFGTime.TextMatrix(Srow, 3) <> "" Then
         HFGTime.TextMatrix(Srow, 2) = Val(HFGTime.TextMatrix(Srow, 2))
         HFGTime.TextMatrix(Srow, HFGTime.Cols - 10) = Format(Summ /
(HFGTime.TextMatrix(Srow, 2) * Countt * 100), "0.00")
         HFGTime.TextMatrix(Srow, HFGTime.Cols - 9) =
     Format((Val(HFGTime.TextMatrix(Srow,HFGTime.Cols-10)))*Val(txtRF.Text)/100,"0.00")
       End If
      End If
      Summ = 0
      Countt = 0
Next Srow
txtBT.Text = 0
       For Srow = 1 To HFGTime.Rows - 1
          txtBT.Text = Val(txtBT.Text) + Val(HFGTime.TextMatrix(Srow, HFGTime.Cols - 9))
       Next Srow
       If txtPer.Text <> "" Then
           Call txtPer_KeyPress(13)
       End If
For Srow = 1 To HFGTime.Rows - 1
 For SCol = 3 To 200 + 2
```

```
If HFGTime.TextMatrix(Srow, 2) = "" Then
      HFGTime.Row = Srow
      HFGTime.Col = SCol
      HFGTime.CellForeColor = &HFFFF00
End If
Next SCol
Next Srow
ElseIf cboCalculate.Text = "Row Wise" Then
For Srow = 1 To HFGTime.Rows - 1
   If HFGTime.TextMatrix(Srow, 2) = "R" Then
      HFGTime.Rows = HFGTime.Rows + 1
   End If
Next Srow
End If
For Srow = 1 To HFGTime.Rows - 1
   HFGTime.Row = Srow
   If HFGTime.TextMatrix(Srow, 2) = "R" Then
      HFGTime.RowPosition(HFGTime.Rows - 1) = HFGTime.Row + 1
         For SCol = 3 To 200 + 2
            HFGTime.Col = SCol
         If SCol = 3 Then
             HFGTime.TextMatrix(Srow + 1, SCol) = HFGTime.TextMatrix(Srow, SCol)
         ElseIf SCol > 3 And HFGTime.TextMatrix(Srow, SCol) <> "" Then
             If HFGTime.TextMatrix(Srow, SCol) <> 0 Then
                HFGTime.TextMatrix(Srow + 1, SCol) = Val(HFGTime.TextMatrix(Srow,
                                                    SCol)) - Val(HFGTime.TextMatrix(Srow, SCol - 1))
             End If
         End If
         Next SCol
    End If
Next Srow
  Summ = 0
  Countt = 0
For Srow = 1 To HFGTime.Rows - 1
   For SCol = 3 To 200 + 2
   If HFGTime.TextMatrix(Srow, SCol) = "" Then
      HFGTime.TextMatrix(Srow, SCol) = 0
   End If
If HFGTime.TextMatrix(Srow, 2) = "" And Val(HFGTime.TextMatrix(Srow, SCol)) > 0 Then
         Summ = Summ + Val(HFGTime.TextMatrix(Srow, SCol))
         Countt = Countt + 1
   End If
If HFGTime.TextMatrix(Srow, 0) <> "" And Val(HFGTime.TextMatrix(Srow, SCol)) > 0 Then
         Summ = Summ + Val(HFGTime.TextMatrix(Srow, SCol))
         Countt = Countt + 1
   End If
```

```
    If HFGTime.TextMatrix(Srow, 2) = "" Then
       HFGTime.Row = Srow
       HFGTime.Col = SCol
       HFGTime.CellForeColor = &HFFFF00
    End If
Next SCol
       If HFGTime.TextMatrix(Srow, 2) = "" Then
         If HFGTime.TextMatrix(Srow, 3) <> "" And HFGTime.TextMatrix(Srow - 1, 3) <> ""
Then
          HFGTime.TextMatrix(Srow,HFGTime.Cols-10)=Format(Summ/(Countt * 100), "0.00")
          HFGTime.TextMatrix(Srow, HFGTime.Cols - 9) =
     Format((Val(HFGTime.TextMatrix(Srow,HFGTime.Cols-10)))*Val(txtRF.Text)/100,"0.00")
         End If
       End If
       If HFGTime.TextMatrix(Srow, 0) <> "" And Val(HFGTime.TextMatrix(Srow, 2)) > 0 Then
         If HFGTime.TextMatrix(Srow, 3) <> "" Then
           HFGTime.TextMatrix(Srow, 2) = Val(HFGTime.TextMatrix(Srow, 2))
          HFGTime.TextMatrix(Srow, HFGTime.Cols - 10) = Format(Summ /
                                              (HFGTime.TextMatrix(Srow, 2) * Countt * 100), "0.00")
          HFGTime.TextMatrix(Srow, HFGTime.Cols - 9) =
     Format((Val(HFGTime.TextMatrix(Srow,HFGTime.Cols-10)))*Val(txtRF.Text)/100,"0.00")
         End If
Summ = 0
      Countt = 0
Next Srow
txtBT.Text = 0
'For Srow = 1 To HFGTime.Rows - 1
          txtBT.Text=Format(Val(txtBT.Text)+Val(HFGTime.TextMatrix(Srow,HFGTime.Cols-
                                                                                  9)), "0.00")
Next Srow
If txtPer.Text <> "" Then
    Call txtPer_KeyPress(13)
End If
Screen.MousePointer = vbNormal
End If
cmdSave.Enabled = True
If Neww = True Then
   MsgBox "All Activites are Calculated by " & cboCalculate.Text & ".", vbExclamation
   Neww = False
End If
Exit Sub
EndD:
   MsgBox "Unexpected Error!", vbCritical, "Time Study"
End Sub
```


Time Study Report
Zoom 100%
TIME STUDY REPORT
STATION No. : 1
OPERATION : COIL LOCKING
STUDY No. : 1
LINE : J
PRODUCT/PART : FC4
DATE : 10/20/2001
OPERATOR : PALANISAMY.V
TOKEN No. : -
EXPERIENCE : 7 - MONTHS
STUDIED BY : KATHIRAVAN
RATING FACTOR : 100
Ele No. | Element Description | AOT(min) | BT(min)
1 | PICKUP THE HOUSING AND PLACE IT IN FIXTURE | 0.08 | 0.08
2 | IINSERT SPOOL ASSEMBLY AND ON THE SWITCH | 0.06 | 0.06
3 | SPOOL ASSEMBLY LOCKING | 0.03 | 0.03
4 | REMOVE FROM FIXTURE,INSERT POINT PLATE,TERIMINAL BASE ASSEMBLY AND PLACE IT IN CONVEYOR | 0.18 | 0.18
5 | PICKUP AND PLACE THE HOUSING NEARER TO THE WORKING PLACE | 0.04 | 0.04
6 | PICKUP AND PLACE THE COIL NEARER TO THE WORKING PLACE | 0.03 | 0.03
Total Basic Time(min) : 0.42
Pages: 1


Assembly Line Balancing Form
REF No 10001
DATE 20/10/2001
PRODUCT/PART ROOTS-90
CYCLE TIME 0.68
Activity Input Sheet
Sta No. | Ele No. | Element Description | BT(min) | Imm. Predecessor
1 | 1 | REMOVE PAPER COVERING AND PLACE IT IN | 0.06 | 0
1 | 2 | PICKUP AND PLACE THE HOUSING TO NEAREF | 0.03 | 1
1 | 3 | PICKUP AND PLACE THE COIL NEARER TO THE | 0.01 | 2
1 | 4 | PICKUP THE HOUSING INSERT RIVETS(2NOS) | 0.16 | 3
1 | 5 | PLACE IT FIXTURE,INSERT COIL AND SPOOL H | 0.21 | 4
1 | 6 | REMOVE,CHECKING AND PLACE IT ON CONVE | 0.08 | 5
2 | 7 | PICKUP THE HOUSING INSERT THE TERMINAL | 0.19 | 6
2 | 8 | REMOVE THE HOUSING ,WINDOW PLATE ,IN. | 0.11 | 7
Activity Balanced Sheet
Sta No. | Ele No. | Element Description | BT(min) | Sta. Time | RSta. Time
1 | 1 | REMOVE PAPER COVERING AND PLACE | 0.06
1 | 2 | PICKUP AND PLACE THE HOUSING TO | 0.03
1 | 3 | PICKUP AND PLACE THE COIL NEARER | 0.01
1 | 4 | PICKUP THE HOUSING INSERT | 0.16
1 | 5 | PLACE IT FIXTURE,INSERT COIL AND | 0.21
1 | 10 | PICKUP AND PLACE THE MATERIAL TO | 0.01
1 | 11 | ARMATURE ROD SUB ASSEMBLY | 0.08
1 | 15 | PICKUP , PLACE THE KEEPER RNG | 0.04 | 0.60 | 0.08
Top
Previous
New
Edit
Delete
Summery
Next
Search
Print
Graph
Close


Comparision Chart
Before Line Balancing
TIME IN MINTUS
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
9
STATION NUMBERS
After Line Balancing
TIME IN MINTUS
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
9
STATION NUMBERS

```
Private Sub Form_Load()
Set clsALB = New clsALBS
frmLine.LineB.FormatString = "Sta No.|^Ele No.|^ Element Description |^ BT(min)
                              |^Imm. Predecessor|^      |^       "
frmLine.HFGBal.FormatString = "Sta No.|^Ele No.|^ Element Description  |^
                               BT(min)  |^ Sta.Time |^ RSta. Time "
Call Mo("Bottom", 1)
        LineB.ColWidth(LineB.Cols - 1) = 0
        LineB.ColWidth(LineB.Cols - 2) = 0
        cmdBalance.Enabled = False
        cmdSave.Enabled = False
End Sub


Private Sub cmdNew_Click()
   Neww = True
   cmdBalance.Enabled = True
   cmdSave.Enabled = True
   cmdPrint.Enabled = False
   cmdDelete.Enabled = False
   cmdSearch.Enabled = False
   cmdEdit.Enabled = False
   cmdTop.Enabled = False
   cmdBottom.Enabled = False
   cmdNext.Enabled = False
   cmdPrevious.Enabled = False
   txtID.Text = ""
   mskDate.Text = Format(Date, "dd/mm/yyyy")
   txtRefNo.Text = ""
   txtProduct.Text = ""
   txtCycleTime.Text = ""
   txtMaxtime.Text = ""
   txtTottime.Text = ""
   txtNoStation.Text = ""
   txtLineEff.Text = ""
   txtBalDeley.Text = ""
   txtProCap.Text = ""
   txtShift.Text = ""
   Call clsALB.NewRef
   txtRefNo.Text = clsALB.L_RefNo
   LineB.Refresh
   LineB.Redraw = True
   LineB.Clear
   LineB.Rows = 2
   HFGBal.Clear
   HFGBal.Refresh
```

```
    HFGBal.Redraw = True
    HFGBal.Rows = 2
    HFGTemp1.Clear
    HFGTemp1.Rows = 2
    frmLine.LineB.FormatString = "Sta No.|^Ele No.|^ Element Description |^
                                  BT(min)|^Imm. Predecessor "
    frmLine.HFGBal.FormatString = "Sta No.|^Ele No.|^Element Description|^
                                   BT(min) |^ Sta.Time |^ RSta. Time "
    LineB.ColWidth(LineB.Cols - 1) = 0
    LineB.ColWidth(LineB.Cols - 2) = 0
    If cmdClose.Caption = "&Close" Then
       cmdClose.Caption = "&Cancel"
    End If
    cmdSave.Enabled = True
End Sub


Private Sub cmdSave_Click()
        cmdSave.Enabled = False
        cmdPrint.Enabled = True
        cmdDelete.Enabled = True
        cmdSearch.Enabled = True
        cmdNew.Enabled = True
        cmdTop.Enabled = False
        cmdBottom.Enabled = False
        cmdNext.Enabled = False
        cmdPrevious.Enabled = False
        Set clsALB = New clsALBS
        Screen.MousePointer = vbHourglass
        clsALB.L_ID = Val(txtID.Text)
        clsALB.L_Date = mskDate.Text
        clsALB.L_RefNo = txtRefNo.Text
        clsALB.L_Product = txtProduct.Text
        clsALB.L_Cycle = txtCycleTime.Text
        clsALB.L_Shift = Val(txtShift.Text)
        Call clsALB.SaveData(Val(txtID.Text))
        Call clsALB.Sel(txtID.Text)
        txtID.Text = clsALB.mvartxtTest
        Call clsALB.SaveGrid(frmLine.LineB, txtID.Text)
        Call Form_Load
        frmTime.cmdTop.Enabled = True
        frmTime.cmdBottom.Enabled = True
        frmTime.cmdNext.Enabled = True
        frmTime.cmdPrint.Enabled = True
        frmTime.cmdPrevious.Enabled = True
        frmTime.cmdDelete.Enabled = True
```

```
        frmTime.cmdEdit.Enabled = True
        frmTime.cmdClose.Caption = "&Close"
        frmTime.cmdNew.Enabled = True
        frmTime.cmdSearch.Enabled = True
        Screen.MousePointer = vbDefault
        Edit = False
        Neww = False
End Sub


Private Sub cmdBalance_Click()
cmdBalance.Enabled = False
Dim i As Integer
Dim str As String
Dim count As Integer
Dim Cycle As Double
Dim Station As Integer
Dim X As Integer
Dim Z As Integer
Dim Max1 As Double
On Error GoTo EndD:
If txtRefNo.Text <> "" And txtRefNo.Text <> 0 Then
If Edit = False Or Neww = True Or Edit = True Then
If Val(txtCycleTime.Text) > 0 Then
Max1 = 0
Cycle = Val(txtCycleTime.Text)
Station = 1
count = 1
'Take Immediate Predecessors is equal to `0`
For i = 1 To LineB.Rows - 1
   If LineB.TextMatrix(i, 4) = "" And LineB.TextMatrix(i, 1) <> "" And LineB.TextMatrix(i, 2)
<> "" Then
      MsgBox "The " & i & " Row  4  Column should not be blank."
      Exit Sub
   End If
   If Val(LineB.TextMatrix(i, 3)) > Max1 Then
      Max1 = LineB.TextMatrix(i, 3)
   End If
   If LineB.TextMatrix(i, 0) = "" And LineB.TextMatrix(i, 1) = "" And LineB.TextMatrix(i, 2) =
"" And LineB.TextMatrix(i, 3) = "" Then
      frmLine.LineB.RemoveItem (i)
   End If
Next i
If Val(txtCycleTime.Text) > Max1 Then
Screen.MousePointer = vbHourglass
For i = 1 To LineB.Rows - 1
```

```
      If LineB.TextMatrix(i, 4) = 0 Then
         If count = 1 Then
            str = str + CStr(LineB.TextMatrix(i, 1))
            count = count + 1
         ElseIf count > 1 Then
            str = str + "," + CStr(LineB.TextMatrix(i, 1))
         End If
      End If
   Next i
For Z = 1 To LineB.Rows - 1
HFGTemp1.TextMatrix(Z, 0) = str
Dim k As Integer
Dim s As String
Dim Arr(10) As Integer
Dim Count1 As Integer
Dim j As Integer
For k = 1 To 10
   Arr(k) = 0
Next k
Count1 = 1
a:
         If InStr(1, str, ",") Then
           For k = 1 To Len(str)
              s = Mid(str, k, 1)
              If s = "," Then
                Arr(Count1) = Val(Left(str, k - 1))
                Count1 = Count1 + 1
                str = Mid(str, k + 1, Len(str))
                GoTo a:
              End If
           Next k
          Else
           Arr(Count1) = Val(str)
         End If
'Checking for Task from which does not exceeds the cycle time
Dim C As Integer
Dim Rn As Integer
Dim Str1 As String
Dim St As String
Dim Temp(10) As Integer
Dim Chk As Boolean
Dim P As Integer
Dim Exceeds As Integer
Y:
Chk = False
Exceeds = 0
```

```
    C = 1
    For k = 1 To 10
     If Arr(k) <> 0 Then
       For j = 1 To LineB.Rows - 1
         If LineB.TextMatrix(j, 1) = Arr(k) Then
           If Val(LineB.TextMatrix(j, 3)) <= Cycle Then
            Exceeds = Exceeds + 1
             Chk = True
             If C = 1 Then
               Str1 = Str1 + CStr(Arr(k))
               Temp(C) = Arr(k)
               C = C + 1
             ElseIf C > 1 Then
               Str1 = Str1 + "," + CStr(Arr(k))
               Temp(C) = Arr(k)
               C = C + 1
             End If
           End If
         End If
       Next j
     End If
    Next k
If Exceeds = 0 Then
    Station = Station + 1
    Cycle = Val(txtCycleTime.Text)
    GoTo Y:
End If
C:
    If C > 1 Then
        Rn = Rnd * (C - 1)
        If Rn = 0 Then
         GoTo C:
        End If
    Else
        Rn = Rnd * C
        If Rn = 0 Then
          GoTo C:
        End If
     End If
Dim AT As Double
   For k = 1 To LineB.Rows - 1
     If LineB.TextMatrix(k, 1) = Temp(Rn) Then
       AT = LineB.TextMatrix(k, 3)
     End If
   Next k
   HFGTemp1.TextMatrix(Z, 1) = Str1
```

```
    HFGTemp1.TextMatrix(Z, 2) = Temp(Rn)
    HFGTemp1.TextMatrix(Z, 3) = Station
    HFGTemp1.TextMatrix(Z, 4) = AT
    HFGTemp1.TextMatrix(Z, 5) = Cycle - AT
    HFGTemp1.TextMatrix(Z, 5) = Format(HFGTemp1.TextMatrix(Z, 5), "0.00")
    Cycle = Cycle - AT
'Checking Predessor equal to Assigned value
        Dim Str2 As String
        Dim Temp1(10) As Integer
        Dim a As Integer
        Dim Cou As Integer
        Dim Cou1 As Integer
        Dim Chk1 As Boolean
        Dim Chk2 As Boolean
        Dim Chk3 As Integer
        Dim Str3 As String
        Dim Ct As Integer
        Ct = 1
        a = 1
        str = ""
        For X = 1 To LineB.Rows - 1
        For k = 1 To 10
           Temp1(k) = 0
        Next k
        Chk3 = 0
        Str2 = LineB.TextMatrix(X, 4)
T:
        If InStr(1, Str2, ",") Then
              For k = 1 To Len(Str2)
                 s = Mid(Str2, k, 1)
                 If s = "," Then
                   Temp1(a) = Val(Left(Str2, k - 1))
                   a = a + 1
                   Str2 = Mid(Str2, k + 1, Len(Str2))
                   If InStr(1, Str2, ",") Then
                       GoTo T:
                   Else
                      Temp1(a) = Val(Right(Str2, k + 1))
                   End If
                 End If
              Next k
             For k = 1 To 10
              If Temp1(k) <> 0 Then
                 If Temp1(k) = Temp(Rn) Then
                    Chk1 = True
                    Exit For
```

```
    End If
  End If
Next k
If Chk1 = True Then
  For k = 1 To 10
    If Temp1(k) <> 0 Then
      Cou = Cou + 1
      For j = 1 To HFGTemp1.Rows - 1
        If HFGTemp1.TextMatrix(j, 2) = Temp1(k) Then
          Cou1 = Cou1 + 1
        End If
      Next j
    End If
  Next k
Else
  GoTo b:
End If
  If Cou > 0 And Cou1 > 0 And Cou = Cou1 Then
    Ct = 1
    For j = 1 To 10
      If Arr(j) <> 0 Then
        Ct = Ct + 1
        If Arr(j) = Temp(Rn) Then
          Arr(j) = LineB.TextMatrix(X, 1)
        End If
      End If
    Next j
    For j = 1 To 10
      If Arr(j) <> 0 Then
        If Arr(j) = LineB.TextMatrix(X, 1) Then
          Chk3 = 1
        End If
      End If
    Next j
    If Chk3 = 0 Then
      Arr(Ct + 1) = LineB.TextMatrix(X, 1)
    End If
  Else
    For j = 1 To 10
      If Arr(j) <> 0 Then
        If Arr(j) = Temp(Rn) Then
          Arr(j) = 0
        End If
      End If
    Next j
  End If
```

```
        Else
            If LineB.TextMatrix(X, 4) = Temp(Rn) Then
            Ct = 1
               For k = 1 To 10
                  If Arr(k) <> 0 Then
                  Ct = Ct + 1
                     If Arr(k) = Temp(Rn) Then
                        Arr(k) = LineB.TextMatrix(X, 1)
                     End If
                  End If
               Next k
               For k = 1 To 10
               If Arr(k) <> 0 Then
                  If Arr(k) = LineB.TextMatrix(X, 1) Then
                     Chk3 = 1
                  End If
               End If
               Next k
               If Chk3 = 0 Then
                  Arr(Ct + 1) = LineB.TextMatrix(X, 1)
               End If
            End If
         End If
        Chk1 = False
b:
        Cou = 0
        Cou1 = 0
        Next X
        count = 1
        str = ""
        Str1 = ""
         For k = 1 To 10
          If Arr(k) <> 0 Then
             If count = 1 Then
                str = str + CStr(Arr(k))
                count = count + 1
             ElseIf count > 1 Then
                str = str + "," + CStr(Arr(k))
             End If
          End If
         Next k
         HFGTemp1.Rows = HFGTemp1.Rows + 1
'       MsgBox "String  " & str & ""
    Next Z
    HFGBal.Rows = LineB.Rows
    For i = 1 To HFGBal.Rows - 1
```

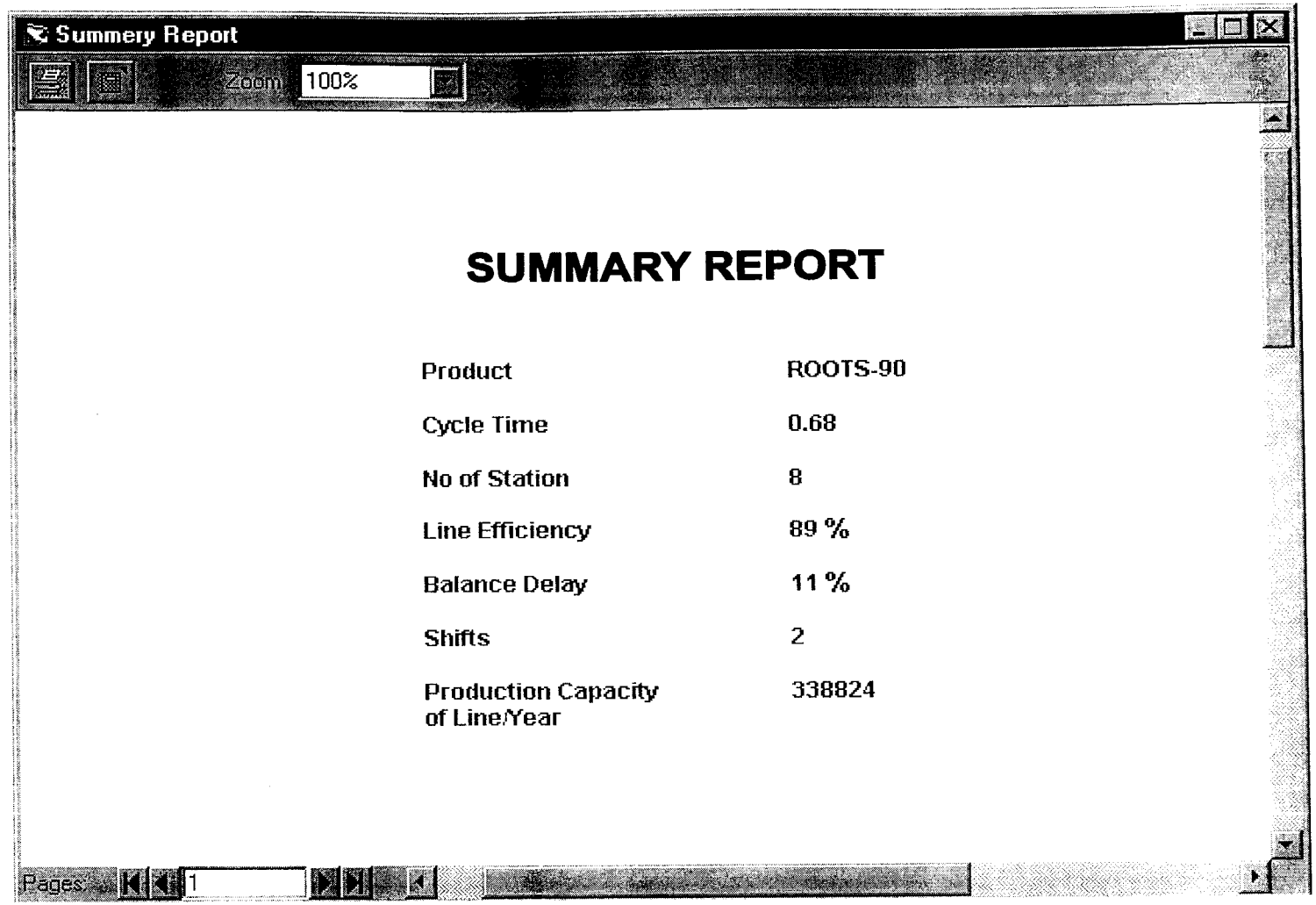
```
    HFGBal.TextMatrix(i, 0) = HFGTemp1.TextMatrix(i, 3)
    HFGBal.TextMatrix(i, 1) = HFGTemp1.TextMatrix(i, 2)
    HFGBal.TextMatrix(i, 3) = HFGTemp1.TextMatrix(i, 4)
Next i
For i = 1 To HFGBal.Rows - 1
    For j = 1 To LineB.Rows - 1
      If HFGBal.TextMatrix(i, 1) = LineB.TextMatrix(j, 1) Then
         HFGBal.TextMatrix(i, 2) = LineB.TextMatrix(j, 2)
      End If
    Next j
 Next i
 frmLine.HFGBal.MergeCells = 1
 frmLine.HFGBal.MergeCol(0) = True
 frmLine.HFGBal.ColAlignmentFixed(0) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignmentFixed(1) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignmentFixed(2) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignmentFixed(3) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignmentFixed(4) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignmentFixed(5) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignment(0) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignment(1) = flexAlignCenterCenter
 frmLine.HFGBal.ColAlignment(2) = flexAlignLeftCenter
 frmLine.HFGBal.ColAlignment(3) = flexAlignRightCenter
 frmLine.HFGBal.ColAlignment(4) = flexAlignRightCenter
 frmLine.HFGBal.ColAlignment(5) = flexAlignRightCenter
 For i = 1 To HFGTemp1.Rows - 1
    If HFGTemp1.TextMatrix(i, 0) = "" And HFGTemp1.TextMatrix(i, 1) = "" And
 HFGTemp1.TextMatrix(i, 2) = "" Then
       HFGTemp1.RemoveItem (i)
    End If
 Next i
 HFGBal.Col = 2
 HFGBal.ColSel = 0
 HFGBal.Sort = flexSortGenericAscending
 Dim Sta As Integer
 Dim Tot As Double
 Dim TotTime As Double
 Dim Max As Double
 Dim RST As Double
 Dim Line As Integer
 Tot = 0
 For i = 1 To HFGBal.Rows - 1
    HFGBal.Row = i
    If HFGBal.TextMatrix(HFGBal.Row, 3) <> "" Then
       TotTime = TotTime + HFGBal.TextMatrix(HFGBal.Row, 3)
    End If
```

```
If HFGBal.Row = 1 Then
   Tot = Tot + HFGBal.TextMatrix(HFGBal.Row, 3)
   Max = Tot
ElseIf HFGBal.Row > 1 And HFGBal.TextMatrix(HFGBal.Row, 0) = HFGBal.TextMatrix(i -
1, 0) Then
   Tot = Tot + HFGBal.TextMatrix(HFGBal.Row, 3)
   If Tot > Max Then
      Max = Tot
   End If
ElseIf HFGBal.Row > 1 And HFGBal.TextMatrix(HFGBal.Row, 0) <> HFGBal.TextMatrix(i -
                                                                    1, 0) Then
HFGBal.TextMatrix(HFGBal.Row - 1, 4) = Format(Tot, "0.00")
HFGBal.TextMatrix(HFGBal.Row-1,5)=Format(Val(txtCycleTime.Text)-Tot, "0.00")
   Tot = 0
   Tot = HFGBal.TextMatrix(HFGBal.Row, 3)
   If Tot > Max Then
      Max = Tot
   End If
End If
If HFGBal.Row = HFGBal.Rows - 1 Then
   HFGBal.TextMatrix(HFGBal.Row, 4) = Format(Tot, "0.00")
   HFGBal.TextMatrix(HFGBal.Row, 5) = Format(Val(txtCycleTime.Text) - Tot, "0.00")
End If
Next i
   frmLine.txtMaxtime.Text = Format(Max, "0.00")
   frmLine.txtTottime.Text = TotTime
   frmLine.txtNoStation = HFGBal.TextMatrix(HFGBal.Rows - 1, 0)
   frmLine.txtLineEff = Format(Val(txtTottime.Text) / (Val(txtNoStation.Text) *
Val(txtCycleTime.Text)), "0.00")
   Line = Val(frmLine.txtLineEff.Text) * 100
   frmLine.txtLineEff = Line
   frmLine.txtBalDeley = 100 - Val(txtLineEff.Text)
   Screen.MousePointer = vbNormal
Else
   MsgBox "Invalid Cycle Time ! ", vbCritical, "ALBS : Message"
   txtCycleTime.SetFocus
   Exit Sub
End If
Else
   MsgBox "Error       :  Missing Cycle Time ! " & vbCrLf & " " & vbCrLf & "Expected :
Cycle Time ", vbCritical, "ALBS : Message"
   txtCycleTime.SetFocus
   Exit Sub
End If
Tot = 0
For i = 1 To LineB.Rows - 1
```

```
   LineB.Row = i
If LineB.Row = 1 Then
   Tot = Tot + LineB.TextMatrix(LineB.Row, 3)
   Max = Tot
ElseIf LineB.Row > 1 And LineB.TextMatrix(LineB.Row, 0) = LineB.TextMatrix(i - 1, 0) Then
   Tot = Tot + LineB.TextMatrix(LineB.Row, 3)
ElseIf LineB.Row > 1 And LineB.TextMatrix(LineB.Row, 0) <> LineB.TextMatrix(i - 1, 0)
Then
   LineB.TextMatrix(LineB.Row - 1, LineB.Cols - 1) = Format(Tot, "0.00")
   Tot = 0
   Tot = LineB.TextMatrix(LineB.Row, 3)
End If
If LineB.Row = LineB.Rows - 1 Then
   LineB.TextMatrix(LineB.Row, LineB.Cols - 1) = Format(Tot, "0.00")
End If
Next i
Dim ProCap As Double
Dim ProCap1 As Integer
   ProCap = (Val(txtShift.Text) * 8 * 60 * 300) / Val(txtMaxtime.Text) * 0.8
   txtProCap = Round(ProCap)
End If
End If
Exit Sub
EndD:
   MsgBox "Unexpected Error!", vbCritical, "Time Study"
End Sub


Private Sub cmdChart_Click()
Dim Column As Integer
Dim Row, i As Integer
Dim intRow As Integer
intRow = 0
If HFGBal.TextMatrix(1, 0) <> "" And HFGBal.TextMatrix(1, 1) <> "" Then
  With frmLGraph.MSChart1
    .chartType = VtChChartType2dBar
    .ColumnCount = 1
    For Column = 1 To 1
    i = 1
    For Row = 1 To LineB.Rows - 1
        If LineB.TextMatrix(Row, 6) <> "" Then
           intRow = intRow + 1
        End If
    Next Row
        .RowCount = intRow
      For intRow = 1 To LineB.Rows - 1
```

```
            If LineB.TextMatrix(intRow, 6) <> "" Then
               .Column = Column
               .Row = i
               .RowLabel = "" & LineB.TextMatrix(intRow, 0) & ""
               .Data = LineB.TextMatrix(intRow, 6)
               i = i + 1
            End If
          Next intRow
        Next Column
      End With
intRow = 0
      With frmLGraph.MSChart2
        .chartType = VtChChartType2dBar
        .ColumnCount = 1
        For Column = 1 To 1
        i = 1
        For Row = 1 To HFGBal.Rows - 1
            If HFGBal.TextMatrix(Row, 4) <> "" Then
               intRow = intRow + 1
            End If
        Next Row
            .RowCount = intRow
          For intRow = 1 To LineB.Rows - 1
            If HFGBal.TextMatrix(intRow, 4) <> "" Then
               .Column = Column
               .Row = i
               .RowLabel = "" & HFGBal.TextMatrix(intRow, 0) & ""
               .Data = HFGBal.TextMatrix(intRow, 4)
               i = i + 1
            End If
          Next intRow
        Next Column
      End With
frmLGraph.Show
Else
    MsgBox "This operation need the Balanced Sheet should not be blank.", vbCritical,
              "Assembly Line Balancing"
End If
End Sub

Private Sub cmdSummery_Click()
If HFGBal.Rows > 3 Then
fraSummery.Visible = True
txtMaxtime.SetFocus
Else
MsgBox "Balanced sheet should not be blank.", vbInformation, "Assembly Line Balancing"
```

```
End If
End Sub


Private Sub Form_Unload(Cancel As Integer)
frmTime.List1.Clear
frmTime.Add = False
frmTime.HFGTemp.Clear
frmTime.HFGTemp.Rows = 1
If frmTime.Visible = False Then
    frmMDI.cmdStart.SetFocus
End If
End Sub


Private Sub txtBal_KeyPress(KeyAscii As Integer)
  If KeyAscii = 13 Then
        Dim Srow As Integer
        Srow = LineB.Row
        LineB.Text = txtBal.Text
        If LineB.Col < 3 Then
           LineB.ColAlignment(LineB.Col) = flexAlignLeftCenter
        ElseIf LineB.Col = 3 Then
           LineB.ColAlignment(LineB.Col) = flexAlignRightCenter
           LineB.TextMatrix(LineB.Row, LineB.Col) = Format(LineB.TextMatrix(LineB.Row,
                                                    LineB.Col), "0.00")
        End If
           LineB.ColAlignment(0) = flexAlignCenterCenter
        txtBal.Visible = False
        LineB.SetFocus
        KeyAscii = 0
    End If
End Sub


Private Sub LineB_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
      If LineB.MouseRow < 0 Or LineB.MouseCol < 0 Then
         txtBal.Visible = False
      Exit Sub
      End If
   If LineB.Col >= 0 Then
      txtBal.Visible = True
      txtBal.Text = ""
      txtBal.Visible = True
      txtBal.Top = LineB.Top + LineB.CellTop
```


Summery Report
Zoom
100%
SUMMARY REPORT
Product
ROOTS-90
Cycle Time
0.68
No of Station
8
Line Efficiency
89 %
Balance Delay
11 %
Shifts
2
Production Capacity of Line/Year
338824
Pages:
1

Result and Discussion

Conclusion

10.CONCLUSION

10.1 Major Conclusions

Based on the project done i.e., computer aided heuristic for assembly line balancing of electric horn assembly using work study and COMSOAL (computer method of sequencing for assembly lines),the following conclusions are drawn.

- A thorough study of the product and assembly process has given an in-depth knowledge of product and processes.
- Standard time for each workstation gives a clear idea to reduce the ineffective time.
- As all phases of time study and assembly line balancing have been carried out systematically, this project will serve as bench marking for future ALB projects.
- After line balancing, no. of workstations and no. of workers can be reduced, which will help in increasing the profit of RIL.
- A total reduction of two operators in the assembly line out of ten operators for the present system will reduce the product cost.

10.2 Report on Method Improvements, and Suggestions:

'I'- Assembly Line

Station no: 1

1. Coil deburring is done by using soldering iron before coil locking, time taken is 0.1 Mins is to be avoided.
2. To handle housing and coil, containers are required in coil locking station.
3. One unusable fixture is found nearer of the coil-locking machine. It disturbs the operator of the workstation.
4. Placing of excess materials and finished horns near the stations is to be avoided.

Station no: 2

1. In terminal riveting station sub assembly, inserting terminal assembly in housing time is more. It is suggested to avoid terminal block with rivets and terminals.
2. Additional set of window plate is required to assemble and rivet the terminal assembly.
3. Rubber sheet is to be provided to place the assembly to avoid scratch on the housing.
4. "ON" switch is to be provided close to the operator to avoid unnecessary movement.

Station no: 3

1. At the diaphragm assembly riveting station, after riveting the height is checked and the same is written on diaphragm. It is not referred any where in the line. This may be eliminated or sample checking is enough.
2. "ON" switch is to be provided closer to the operator to avoid unnecessary movement.

Station no: 4

1. Keeper ring and housing are placed over a working table tray. Rubber

sheet over the tray is required to avoid scratches on the housing.

2. Removed of sub assembly by a screwdriver is to avoid.

Station no: 5

1. Working fixture is placed at the Center of the working table and trays are arranged accordingly (Xerox copy of this arrangement is enclosed).
2. Dust and oil found in this station is to be avoided.
3. Two numbers of mounting brackets are assembled. Feasibility study is to be done to eliminate one no of mounting bracket.

Station no: 6

1. Two single leads are inserted horn for testing. It may be possible to provide single socket to reduce insertion time.

J ASSEMBLY LINE

1. Terminal riveting in 2nd station, the crack is found due to material problem. To avoid this the material is approved by end user (94 Nos. terminal crack is found out of the 175 Nos. of terminal riveted).
2. The working area of terminal riveting station is not sufficient
3. The passage between the "J" and "K" line is not sufficient since the sides of stations place all the working materials.
4. Unused table is found nearer the locking station.

GENERAL

1. 11 rollers of the conveyor in "I" line is not working.
2. Sliding rack can use bottom space of the conveyor.
3. Packing area is not cleanly maintained.
4. Over the working table (painting and packing) dust, paint marks are found. Feasibility study is to be done to provide stainless steel or rubber sheet over the working table.
5. Carton box or tray with horns is placed one over the other. The top carton box or tray is in open condition and this leads to dust formation of the horns. This can be avoided by placing a polythene sheet over the carton box or tray.

Bibliography

BIBLIOGRAPHY

Websites:

1. www.prenhall.com/weiss_dswin/html/one.html
2. www.graph.ms.org/cmt/neworlean95/talks/wao3.1.html
3. www.cen.unic.edu/course/ie261/notes/alb/alb.com

BOOKS

1. George Kanawaty, "Introduction to work study", Universal Publishing Corporation, Bombay (1995).
2. International labour organization, "introduction to work study", Geneva, 1965.
3. Mikell P. Groover, " Automation, Production Systems And Computer –Integrated Manufacturing", Prentice Hall Of India Pvt Ltd, New Delhi (1999).
4. S N Chary, "Production And Operations Management", Tat McGraw-Hill Publishing Company Limited, New Delhi (1999).
5. Richard B. Chase, Nicholas J. Aquilano, "production and operations management", McGraw-hill, New York (1995).
6. Panneerselvam. R., "Production And Operations Management", Prentice Hall, New Delhi (1999).
7. Stewart M. Lowry, Harold B. Maynard, G.J. Stegemerten, "Time And Motion Study", McGraw-hill book company, Inc, New York and London.
8. Evangelous Petrotsos, " Mastering Visual Basic 6", BPB Publications, New Delhi (1998).
9. Roger Jennings, "Roger Jennings' Database Developer's Guide With Visual Basic 6", Techmedia, New Delhi (1999).
10. Steven holzner, "visual basic 6 programming, black book", dreamtech press, New Delhi (2000).
11. Karl Schwartz, "Microsoft Access 97", Tata McGraw Hill Publishing Company Limited, New Delhi (1998).