THE DESIGN AND IMPLEMENTATION OF A FLEXIBLE MANUFACTURING SYSTEM FOR A SURFACE MOUNTING PRODUCTION LINE.

Mark Steven Chodos

A project report submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, 1990

DECLARATION

I declare that this project report is my own, unaided work. It is being submitted for the Degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

M.S. CHODOS

31st day of January 1990.

ACKNOWLEDGEMENTS

I would like to thank the following people for their dedicated and unselfish help and guidance:

M. Crouch (STC).

P.W. Van Der Molen (Teltech).

Prof. R. Adams (Wits).

My Father and Sister.

I would also like to thank the Management at Standard Telephones & Cables (STC), for allowing me to utilize their facilities and resources.

ABSTRACT

The viability of introducing a Surface Mount production line is chiefly determined by the reliability characteristics of the components being used. Surface Mount Technology (SMT) is entirely new and although related to traditional through-hole processes, requires different components, assembly techniques and design methods. The purpose of the literature survey is primarily to determine whether surface mount components meet today's industrial requirements with respect to their manufacturing reliability and availability. A brief review of the evolution of SMT is also presented. This study finds that the implementation of SMT should be given highest priority by manufacturing companies in order to maintain their share of the marketplace.

Surface Mount Technology embodies a totally new automated circuit assembly process, using a new generation of electronic components: surface mounted devices (SMDs). Smaller than conventional components, SMDs are placed onto the surface of the substrate. From this, the fundamental difference between SMD assembly and conventional through-hole component assembly arises; SMD component positioning is relative, not absolute.

When a through-hole component is inserted into a PCB, either the leads go through the holes or they don't. An SMD, however, is placed onto the substrate surface, it's position only relative to the solder lands, and placement accuracy is therefore influenced by variations in the substrate track pattern, component size, and placement machine accuracy.

Other factors influence the layout of SMD substrates. For example, will the board be a mixed-print (a combination of through-hole components and SMDs) or an all-SMD design? Will SMDs be placed on one side of the substrate or both? And there are process considerations like what type of machine will place the components and how will they be soldered?

This project describes in detail the processes involved in setting up an SMT facility. A simulation program was developed to verify the viability of these processes. The simulation program was also applied to an existing SMT facility and together with developed optimization software, attempted to identify and resolve some of the major problems. All this was achieved, and the extent to which simulation could be used as an efficient production tool, was highlighted.

CONTENTS PAGE

LITERATURE SURVEY

THEORETICAL SOLUTION

PRACTICAL IMPLEMENTATION

GLOSSARY

DIP	Dual-in-line-package
IC	Integrated Circuit
SMT	Surface Mount Technology
SMD	Surface Mount Device
SMC	Surface Mount Component
PCB	Printed Circuit Board
SOT	Small Outline Transistor
VSO	Very Small Outline
PLCC	Plastic Leaded Chip Carrier
EIA	Electronic Industries Association
JEDEC	Joint Electron Device Engineering Council
MLC	Multilayer Ceramic
MELF	Metal Electrode Face Bonding Component
LED	Light Emitting Diode
SOIC	Small-Outline Integrated Circuit
DRAM	Dynamic Random Access Memory
PQF	Plastic Quad Flatpack
COB	Chip-on-Board
TAB	Tape Automatic Bonding
TTL	Transistor-Transistor Logic

LITERATURE SURVEY

1. The status of surface mounting in the scope of today's manufacturing environment.

2. The manufacturing and reliability requirements of surface mount packages and devices.

1 THE STATUS OF SURFACE-MOUNTING IN THE SCOPE OF TODAY'S MANUFACTURING ENVIRONMENT.

1.1 Introduction

Printed circuit technology first gained widespread acceptance in the 1960s. At that time electronic circuitry was assembled from discrete components that required relatively few connections to the board. Early integrated circuits rarely employed more than 16 leads, and through-hole assembly technology (in which component leads were inserted and soldered into holes through the board) could easily accommodate the demands of the day. With the advent of the microprocessor and the growth of computer technology, circuit complexity has increased to the point that through-hole component mounting techniques are no longer adequate.

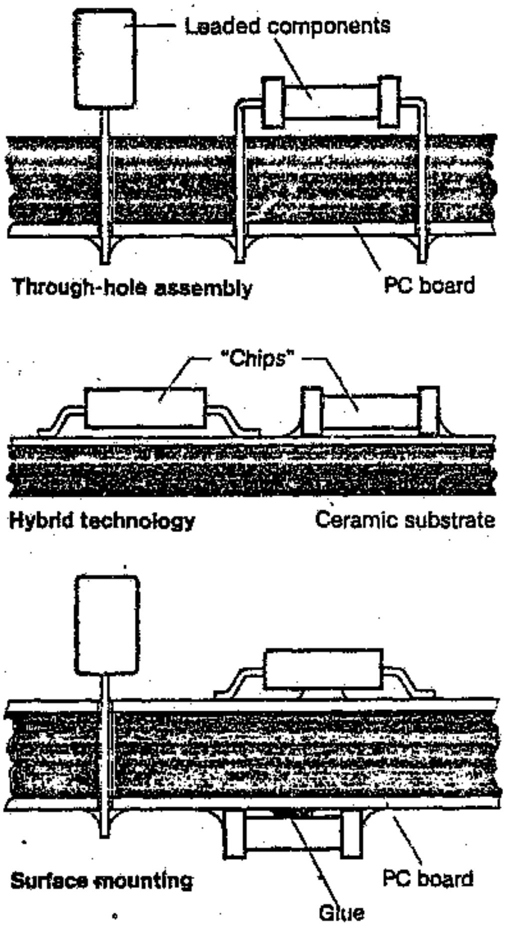
The trend in the semiconductor industry toward increasing circuit integration is being accompanied by advances in packaging techniques, and both are designed to increase the performance and to reduce the size, weight and cost of high package density electronic assemblies. Competitive pressures and advances toward miniaturization for virtually all system designs are some of the motivating factors influencing this revolution. Since the major burden of effective competition lies in significant cost savings, the use of surface mount technology (SMT) is indeed enticing.

The dual-in-line-package (DIP) was developed and became standardized at a time when integrated circuits (ICs) were relatively simple, requiring few Input/Output (I/O) connections. The DIP has proven to be a reliable package style that has more than adequately met the demands placed upon it. However, beyond this need for a functional and reliable component is an increasing need for better space efficiency and design flexibility which the DIP is often unable to meet.

As the complexity of the ICs increase and the number of leads increase accordingly, the size of the DIP quickly becomes unmanageable and inefficient. Large DIPs (requiring 40 pins or more) occupy excessive board area which reduces chip performance due to long connections from the internal silicon chip (die) to the external package-to-board terminations, see Figure 4.1. At higher lead counts, the utilization of the DIP begins to diminish the benefits of miniaturization in circuit integration because the packages are utilizing volumes much larger than the chips themselves.

1.2 Surface Mount Technology

Surface mount technology differs from the customary through-hole assembly of electrical devices by soldering the components directly to metallized pads or footprints on the surface of the printed circuit board (PCB). Comparably, conventional DIP technology consists of inserting the wire leads of components through plated holes which have been drilled in the board and connected to the solder pads by soldering on the reverse side (See Figure 1.1).



a. Through-hole assembly of leaded components
b. Surface mount devices on ceramic substrate
c. Leaded and surface mounted devices on the same PCB

FIGURE 1.1 :Stages of development in surface mounting

The solder joint of a surface mount component (SMC), therefore, serves as both an electrical and a physical connection to the PCB. Since surface mounting eliminates the need for hole-drilling to accommodate component leads, these smaller dimensioned components can be placed closer together and on both sides of the board. This contribution releases valuable board space for more component placement or simply reduces overall board size.

Surface mount components have been introduced as the solution to accommodating the size constraints and high performance capabilities of todays ICs. SMCs do have limitations of their own, however. Even though they have eliminated some of the problems associated with large conventional packages, they have not accomplished the task without introducing some new problems.

1.3 Development of SMT

Early applications of surface-mounting techniques were primarily in areas such as military electronics, but in recent years the consumer electronics portion of the industry has been reshaping the market. Surface mounting has been exploited by the Japanese with great success since the mid 1970s specifically in their consumer electronics markets, claiming significant cost savings and performance advantages over traditional packaging styles. The Japanese focus has been on discrete chip components and leaded (LSI) devices, directing that concentration toward effective packaging for thin calculators and watches.

The high volume users in the automotive industry have recently been joined by a second wave of users in the areas of telecommunications, instrumentation and peripherals. It is now apparent that a third wave is being composed of small-to-medium size users who are influenced by the space-savings reliability and lower assembly costs.

The vast growth in surface mounted components can be seen as the culmination of the following increases in assembly design:

(1) Initially, the general use of small leaded discrete components (resistors, capacitors and transistors) coupled with a few DIPS could achieve a packing density of 20 components per square inch of board space. The density was derived from close component placement and two-sided PCB application.

(2) The second evolution was achieved through the use of hybrid ceramic modules whereby the component density rose to approximately 50 components per square inch. This density was achieved by the direct attachment of the IC die to the hybrid substrate.

(3) State of the art density is accomplished by the utilization of a wide variety of surface mountable passive and active components which increases the range to hundreds of components per square inch of board area.

Other benefits associated with SMT include less parasitic capacitance, lead resistance and inductance (due to shorter lead lengths), plus reduced signal noise and crosstalk especially critical in some high-frequency and linear circuits.

From the viewpoint of manufacture, the particular suitability of SMDs to automated assembly is one of their most important criteria. In all their geometric forms (cuboid or cylindrical) and in all forms of component package (such as SOT, VSO, PLCC and flat pack) they can be placed automatically with a single application head. Since SMDs are only placed on the circuit substrate, no specific guide tools are required for inserting the component leads, in contrast to the automatic placement of leaded components.

The space around the component body which was previously required for the application tools is thus additionally available as a usable area for components. Particularly in the placement of small to medium-sized batches, SM technology results in new possibilities for flexible automation and hence a cost benefit in production. Flexible, automatic placement systems for surface mount devices are a significant element in achieving these objectives. The field of application for these automatic systems is considerably greater than for conventional automated lead insertion and extends from a single machines for single-point automation right up to complex system configurations in integrated production automation.

1.4 Present status of SMT internationally

A current profile of the international electronics industry indicates that the major equipment markets driving this technology are influenced by three basic factions: the military, the high-performance mainframes and consumer electronics. Figure 1.2 describes this relationships involved between the major forces motivating the industry toward SMT.

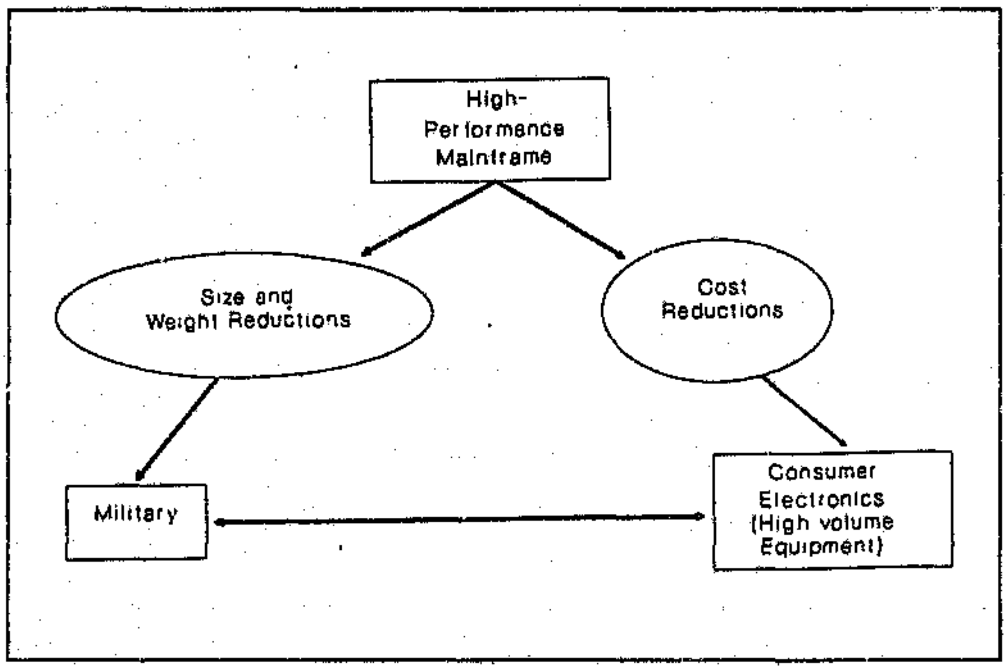


FIGURE 1.2: The major motivating forces behind SMT

Publications on SMT estimated that 30% of all electronic equipment manufactured in Japan during 1985 used SMT and project 50% usage by that country by 1990. In contrast, about 15% of US electronic products in 1985 incorporated SMT with only 30% usage projected by 1990[2]. With the Japanese industry still leading, the European industry could at least keep pace with the U.S. In 1986, surface mount devices worth 660 million US Dollars were purchased in Europe according to a study by Frost and Sulivan[5]

Figure 1.3a shows the increasing share of surface mount technology in the international market, and Figure 1.3b shows the growing number of components that are becoming available.

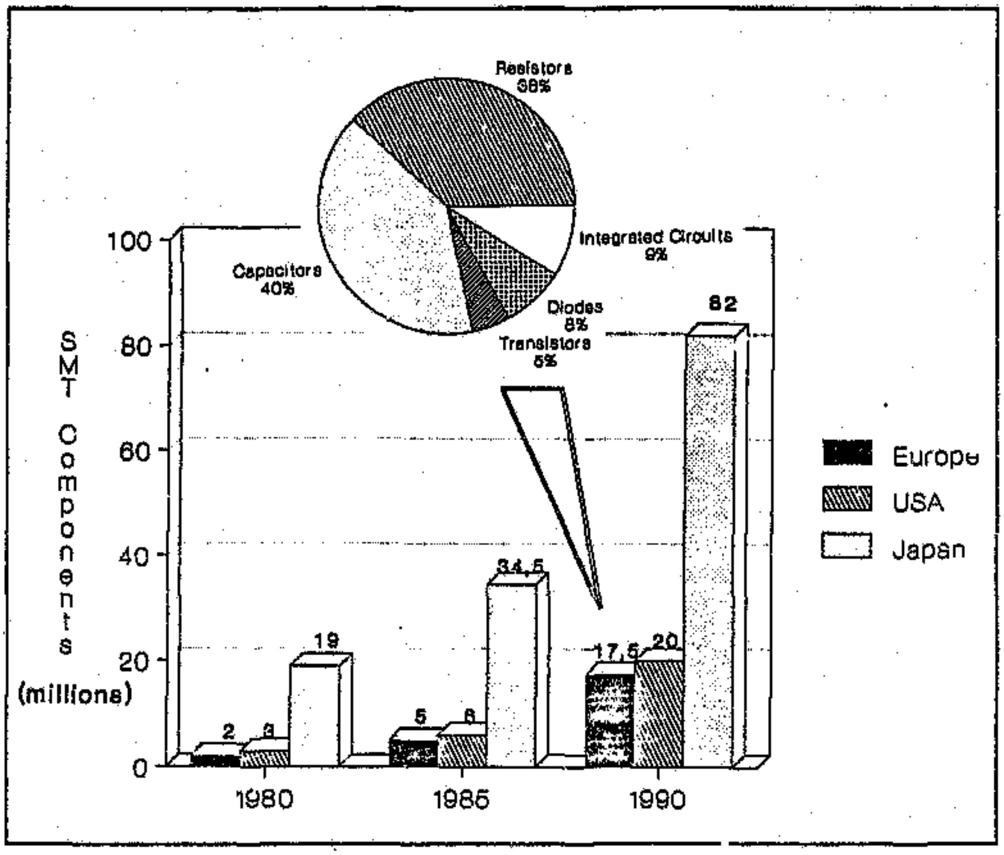


FIGURE 1.3a: Development of the SMD application from 1980 to 1990

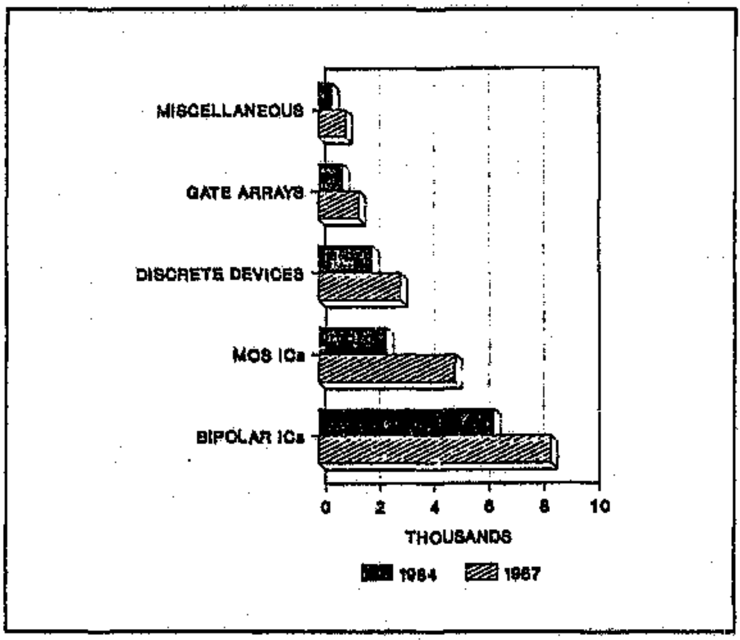


FIGURE 1.3b: SMT Component availability

1.5 SMT in South Africa

South African manufacturers trail their overseas counterparts in the introduction of SMT. While overseas electronics manufacturers use SMT on close to 50% of their PCB assemblies, South Africa fails to approach the 0,5% mark. So what has happened to the great SMT boom? Why has SMT not taken off in South Africa? Is a flip-over from through-hole to SMT inevitable?

Problems of component availability and package standardization have been one of the reasons given to explain the "slow" adoption of surface mount technology. On the other hand however, component suppliers from the US and Europe are claiming that the most-used functions are available and more could be put into surface mount packages, but the demand for them is low.

SMT systems, even at entry level, require a capital investment which is probably discouraging manufacturers who are still amortising the investment in conventional assembly equipment. Concern is also being expressed over the lead times required to implement SMT, particularly for smaller companies, which may have to implement non-automated production lines. This lead time has been put at two to three years.

Unless South African electronics manufacturers give serious consideration to introducing surface mount technology, the country's fledging industry could suffer a major setback as a result of a flip-over effect when SMT component prices plunge and through-hole devices increase in cost. Already in the last three years, surface mount resistors and capacitors have gone from being eight times as expensive as conventional components, to parity.

1.6 THE MANUFACTURING REQUIREMENTS OF SURFACE MOUNT PACKAGES AND COMPONENTS

1.6.1 Introduction

The success of any technology depends on a uniform understanding of expectations. Specifications are used to define the acceptability of the materials or services being procured and are an essential part of the agreement between the purchaser and the supplier.

Many organisations are active on a national or international basis in developing and promoting standards in all aspects of SMT. Such standards can be divided into three categories: those describing design requirements, those concerned with materials specifications, and those that address manufacturing process technology. The following are the major organisations active in the field of SMT:

- Institute for Interconnecting and Packaging Electronic Circuits (IPC)

- Electronic Industries Association (EIA)

- International Electrotechnical Commission (IEC)

The EIA has developed standards in many areas of electronic technology and of particular interest are their component standardization activities. One of their committees is The Parts Engineering Panel which develops standards for passive components and electromechanical devices. The Joint Electron Device Engineering Council (JEDEC) is another, which develops standards for semiconductor devices.

Appendix 1 describes in detail the characteristics of surface mount components, with a particular emphasis on information needed for proper selection of passive components, semiconductor and electomechanical devices.

While much of the scientific world has long since converted to the SI system of units, the electronics industry continues to employ a tangle of metric and English systems. Many components, for instance, are defined in English units: component lead pitches of 0,050 and 0,025 inches have become worldwide standards, while in many other cases, metric dimensions govern.

In an attempt to avoid confusion, units of length are expressed in both inches and millimetres throughout this document, with the industry-preferred unit being specified first.

Surface mount technology has emerged to optimize the high-density, high-speed integrated circuit performance of today's microelectronics. The mature manufacturing processes employed with conventional DIP componentry are taking a back seat to SMT, which is quickly becoming accepted industry-wide. By challenging the cost, size, weight, and performance characteristics of previous manufacturing methods, surface mounting is better meeting the demands and competitive pressures of advancing microcircuit miniaturization.

Although SMT is gaining momentum rapidly, one area must still be addressed for the technology to be exploited more fully, component standardization. In several areas the surface mount package styles offered by different vendors are not compatible, integrated circuit packaging being of particular concern. For example, both TEXAS Instruments (TI) and Signetics offer a broad line of TTL digital ICs. The Signetics product line, however, is exclusively in SOIC packages, while the TI line is offered only in 20-lead plastic chip carriers. At the present time, neither vendor can second-source the other.

Upon initial analysis, a major stumbling block appears to be the large initial capital investment and total commitment needed to venture into SMT. Further investigations, however, indicate that the total life cycle costs will be substantially lower in the long run; consequently, those who falter in the decision to accept SMT may very well be left behind.

Clearly, SMT poses a major challenge to the electronics industry generally. Component suppliers, equipment manufacturers and electronics manufacturing companies all have to make changes in, not just their technology, but also the management of their operations. SMT frequently requires fundametally different approaches to the management of production flow, stock control, quality control, etc. For many companies, the easy solution to these problems is to decline to be involved at present. The only logical conclusion of such an attitude is for such companies to see a decline in their competitiveness and marketplace, with ultimately disasterous results.

THEORETICAL SOLUTION

1. The design of a flexible manufacturing surface mount assembly facility.

3 SETTING UP PRODUCTION OF SURFACE MOUNT ASSEMBLIES

3.1 Introduction

Unlike the mature processes used to produce conventional printed circuit assemblies, surface mount technology is not yet well developed. Although certain aspects have recieved much attention throughout the industry, no standard overall process flow has yet emerged.

A well-designed facility should be capable of accomodating a variety of SMDs. These might include chip resistors, ceramic and electrolytic capacitors, inductors, transistors, diodes and integrated circuits in several package styles. It might also need to handle such varied substrate materials as epoxy-glass boards, ceramic thick films, flexible circuits and special purpose boards such as Teflon.

The facility should be analysed as an integrated operation: material flow, process technology, test strategy and compatibility with exsisting operations are some of the areas that must be addressed. Once the actual requirements have been specified, the task of selecting the appropriate technology can begin.

3.2 Assembly Procedure For SMT

Surface mount devices can be mounted on printed circuit boards either on the component (top) side or on the solder (bottom) side or on both sides of the board, with or without conventional through-hole components.

For SMDs mounted on the component side of the PCB, the standard process is to apply a solder paste (normally by means of a screen printing process) to the pads to which the device is to be soldered. The SMDs are then placed onto the solder paste; the "tackyness" or adhesive quality of the solder paste keeping the SMDs in place for the reflow soldering, where the board is heated to a temperature beyond the melting point of the solder paste. The solder is reflowed by one of several common methods, including infrared (IR), vapour phase, or thermal conduction. The boards then go through a cleaning step to remove residual flux and any stray solder balls that may be on the board.

For SMDs mounted on the solder side of the PCB, the standard process is to apply one or more dots of adhesive on the board where the body of the SMD will be located. The SMD is then placed with the body of the device displacing some of the adhesive; the "tackyness" of the adhesive keeps the placed components in position during the placing process.

When all the SMDs are placed, the adhesive is cured by means of an ultra violet light and/or heat, usually infrared. The cured adhesive is of sufficient strength for the board to go through a further automatic or manual insertion of conventional through-hole components, and then go through the wave soldering process, without the SMDs being dislodged from the board. In this approach, the SMDs are actually immersed in the molten solder during the process.

Both adhesive attach/wave solder and solder paste/reflow processes introduce operations that are new to most assembly facilities. Selecting the appropriate method for any given facility requires an understanding of the capabilities and limitations of these processes.

3.3 Assembly Procedures

Figure 3.1 shows the PCB assembly variations possible with SMDs: Assemblies exclusively with SMDs in the top row (a and b), mixed assemblies. i.e. SMDs combined with with leaded components in the middle (c and d), and mixed assembly consisting of dip-solderable components (on solder side) and non-dip-solderable components (on component side) in the last row (e).

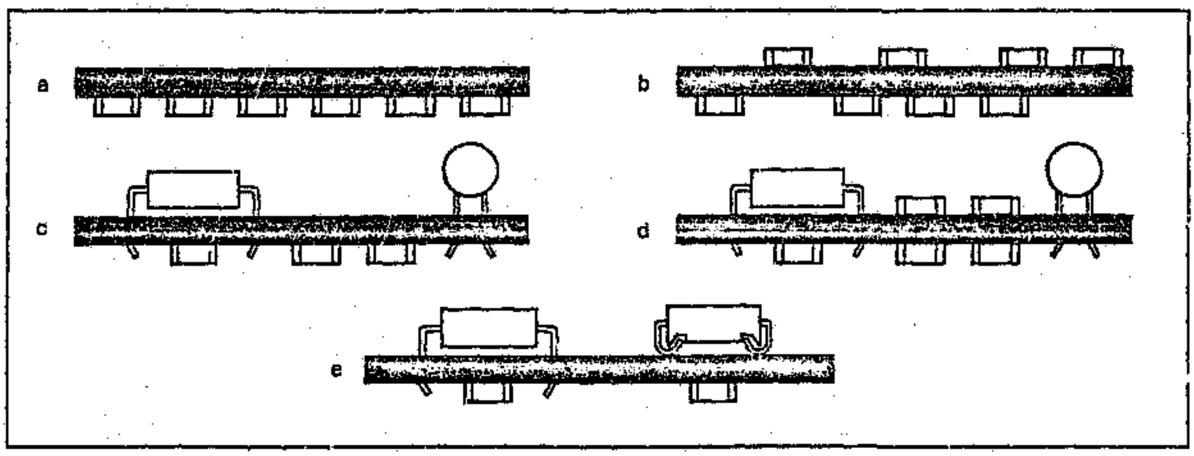


Figure 3.1 SMT Assembly variations

The ultimate manufacturing aim is a uniform mounting procedure with the exclusive use of SMDs. Figure 3.2 shows examples for totally surface mounted assemblies with reflow soldering (top) and wave soldering (bottom). However, very few surface mount boards consist entirely of surface mount components. Invariably, a few components will not be available, or if available, cost considerably more than their through-hole equivalents.

As a result, nearly all so-called SMT boards are actually mixed assemblies containing both surface mount and through-hole devices. Thus manufacturing of mixed boards is inevitable and there are two variations to this approach for SMDs placed on a single side of the board.

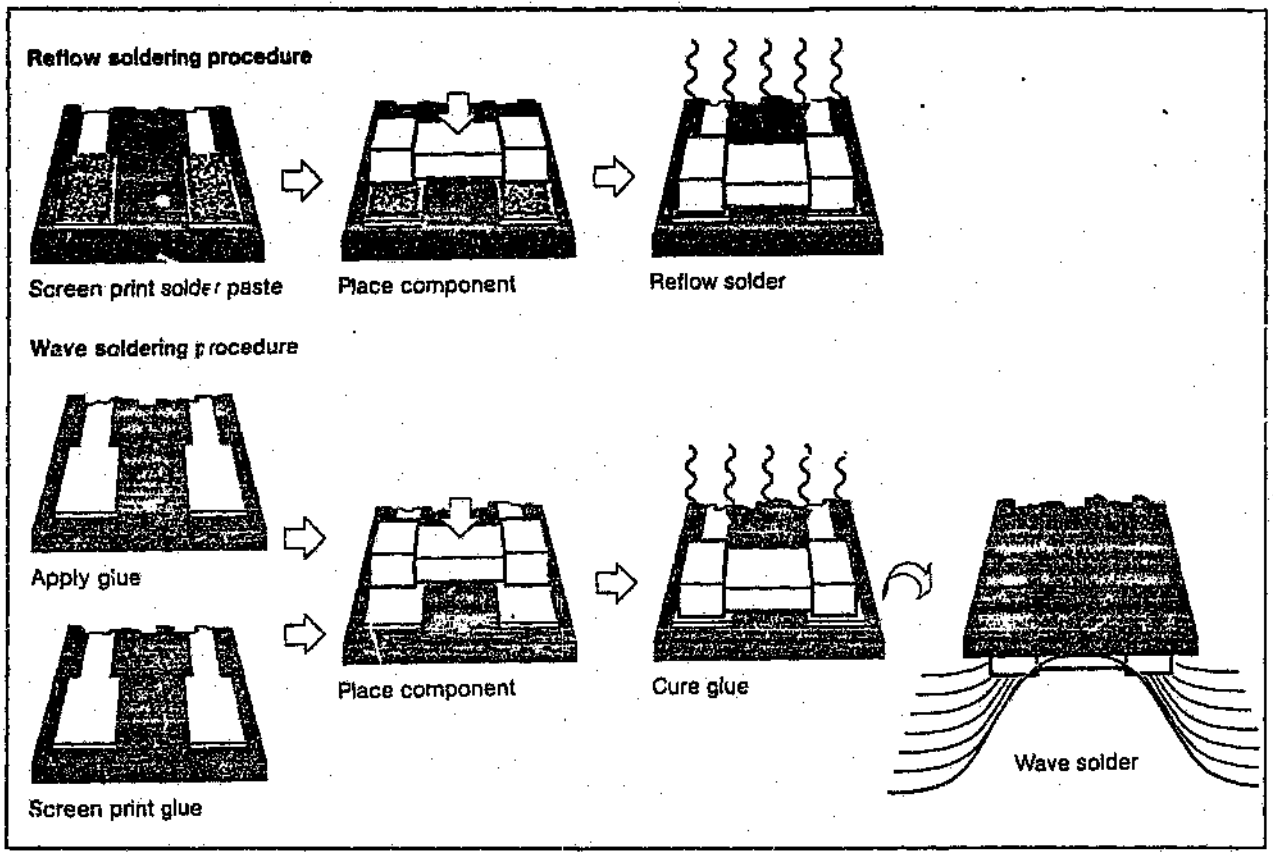
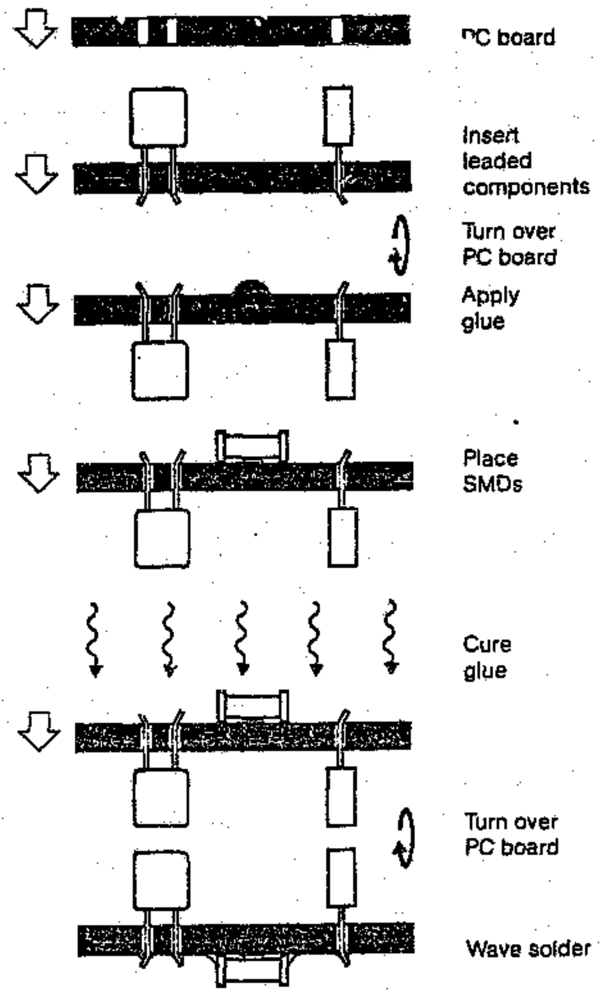


Figure 3.2 PCB exclusively with SMDs, reflow soldered or wave soldered

In mixed assemblies (See Figure 3.3) where the leaded components are placed first, the board must then be turned over and the glue applied for the SMDs. The screen printing method for applying the glue cannot be used as the pointed ends of the leaded components get in the way and an alternative method must be used. The SMDs are then placed, the glue cured and after a renewed turn over, the board is wave soldered.

The second variation differs from the first in so far as the glue is applied by screen printing at first and the production steps executed as illustrated in Figure 3.4. This procedure has the advantage that the glue can be applied by screen printing, however, because of the already mounted SMDs, vacant board space is required for the mounting tools of the leaded components and therefore high densities cannot be realized.



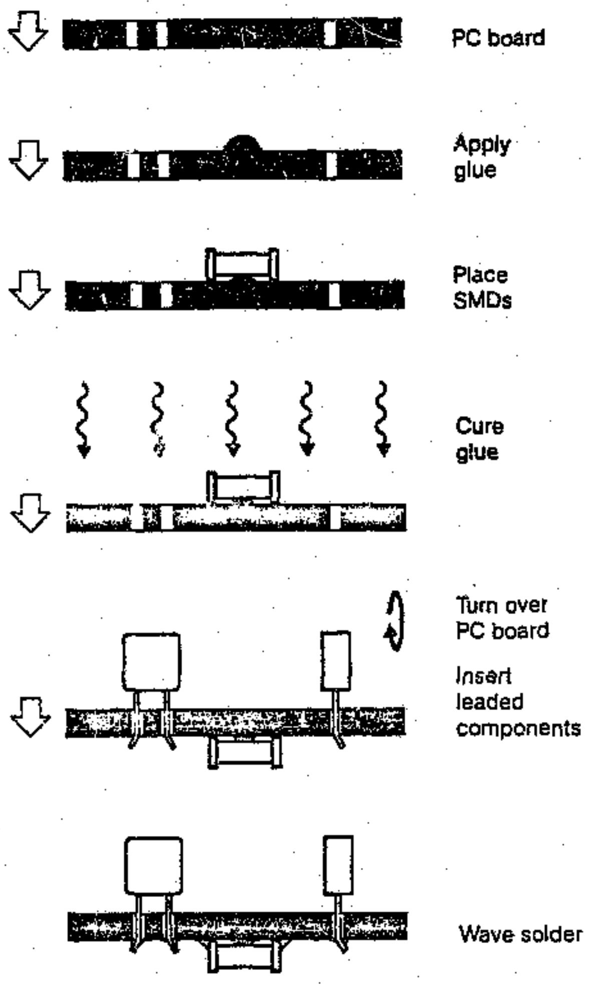


Figure 3.3 Mixed assembly of SMDs and leaded components.

Figure 3.4 Mixed assembly of SMDs and leaded components.

The procedure for double-sided SMD mounting is as follows:

- o Screen printing of solder paste
- o SMD placement
- o Reflow soldering
- o Insertion of leaded components
- o PCB turn over
- o Application of glue
- o Placement of SMDs on the reverse side
- o Curing the glue
- o PCB turn over
- o Mounting of components requiring special handling
- o Fluxing, wave soldering

Due to their inherent complexity, it is worth investigating the feasibility of a partitioned design before considering "mixed prints". Thus part of the circuit is made on an all-SMD PCB and the remainder on a conventional or mixed print PCB.

The very small dimensions of an all-SMD circuit often allow such a circuit to be repeated several times on a single substrate (See Section 4.2.1). This further increases production efficiency.

Figure 3.5 is a flow chart for the various assembly and soldering variants.

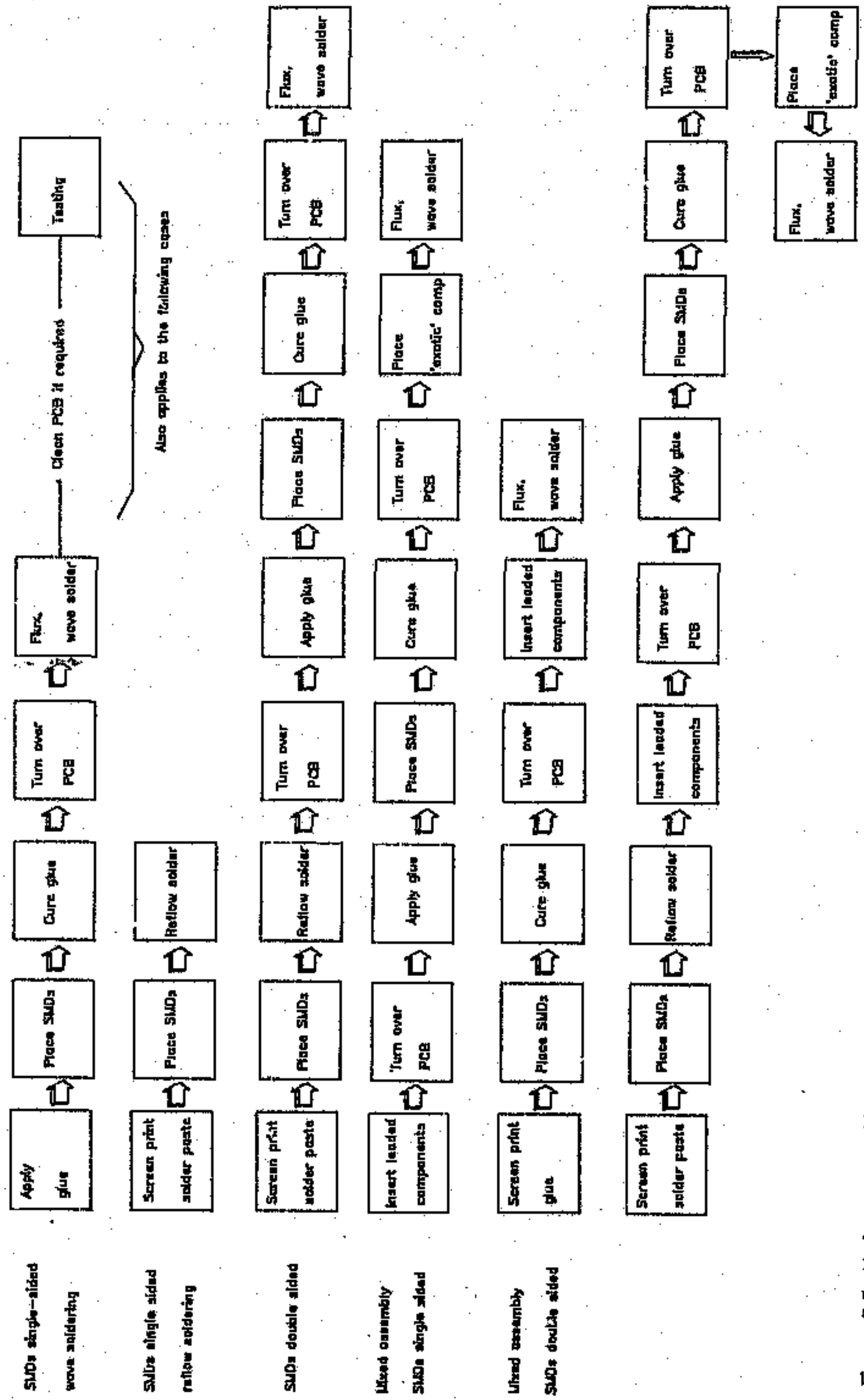


Figure 3.5 Various assembly and soldering variants.

4. DESIGN AND LAYOUT GUIDELINES

4.1 Introduction

The first and most important step towards SMT begins with the design of the PCB layout. At this stage it is already neccesary to consider how the PCB assembly will be manufactured, tested, repaired and maintained. Hence, for the layout of the PCB, the designer needs details on permissible component configurations on the circuit board, the minimum spacings between components, and on the size and shape of the pads on the PCB. These "layout rules" reflect the interdependence of the production steps in SMD technology.

Also, since SMDs can be mounted on both sides of the board, it takes considerable experience to draw up layout rules for economic production and they are only valid for a particular range of applications with it's specific requirements. In addition, the rules are subject to continual improvement. Also, the shape and arrangement of the pads or footprints is dependant on many parameters, which might be assesed quite differently by the users. It can therefore be expected that the same layout rules do not apply to all users.

Information in this section covers the SMT PCB layout guidelines and can be used to augment exsisting design guidelines for through-hole boards. It is also applicable for both manual and automatic assembly methods.

4.2 Printed Circuit Board Design

4.2.1 Board size and construction

PCBs are manufactured in panels of various sizes. Several recommended panel sizes have been identified by IPC and published in ANSI/IPC-D-322. The full panel is usually cut into subpanels for assembly processing. A typical panel and subpanel relationship is shown in **Figure 4.1.**

The 457 x 610mm (18 x 24in) panel is widely used, but after allowance for plating and processing, the maximum usable board dimensions for this panel are approximately 405 x 560mm. Thus lowest board costs are realized when the individual boar sizes are optimized for full panel utilization and this often forces the designer to deviate from the standards. The board size is also dependant on the capacity capabilities of the different processes; for example the placement machine may only cater for a maximum board size of 400 x 520mm, and in this case, non-standard board sizes may have to be produced.

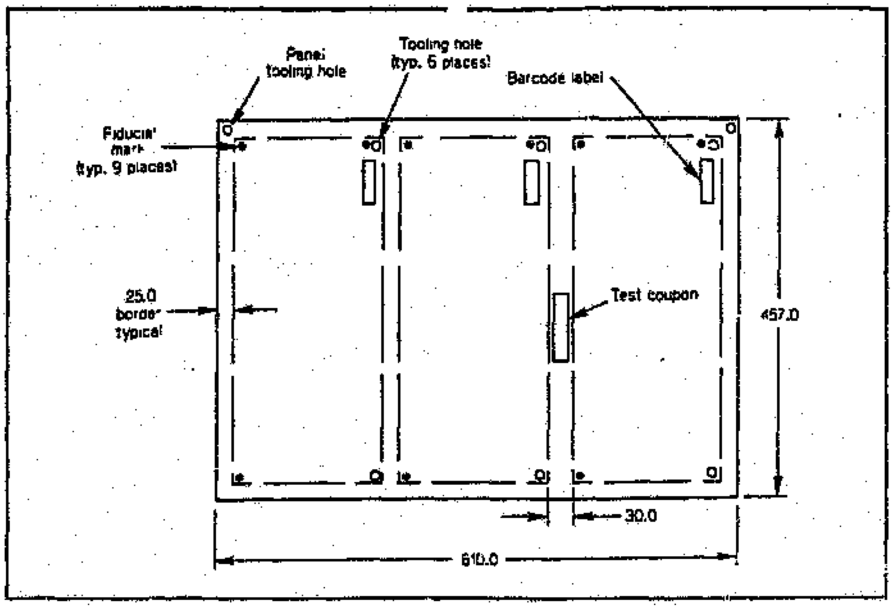


Figure 4.1 Typical third-panel dimensions for 475x610-mm full panel.

4.2.2 Tooling holes and Fiducial marks

Accurately located tooling holes at the corners of the board (subpanel) are essential features. They are frequently used to align the board at both the screen printing and the component placing operations. At least two tooling holes are required at opposite corners along the longest side of the board.

Because of accuracy limitations, tooling holes should not be used as the primary locating feature for fine-pitch chip carriers or PLCCs above 44 leads. Vision alignment is recommended for these devices and fiducial marks should be used. A minimum of three fiducial marks should be employed at three corners of the board as shown in Figure 4.10 above. The actual shape of the fiducials depends on the characteristics of the alignment system. Representative shapes are shown in Figure 4.2.

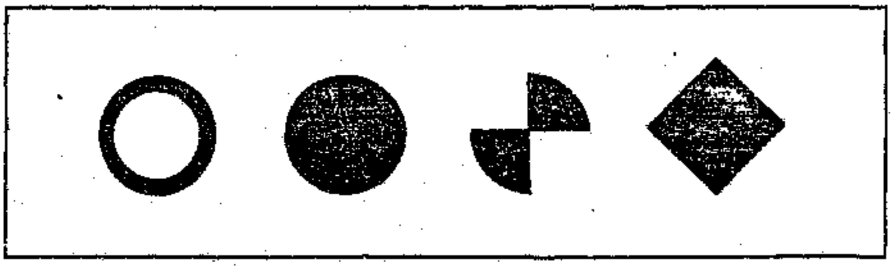


Figure 4.2 Shapes of fiducial marks used by optical alignment systems.

The best fiducial mark is one that has uniform brightness and high contrast compared to the background.

4.2.3 PCB Material

It must be determined by experiment or otherwise wether the planned PCB material is suitable for the application. A widely used material is epoxy-glass FR4. Other PCB materials vary from FR4, for example in the bonding strength of the copper layer and in the expansion coefficient.

4.2.4 PCB Coating

Coatings which remain on the PCB after manufacture are the protective coating and solder resist layer. The protective coating is a surface protection of the PCB against corrosion and mechanical damage.

With SMD technology, the solder resist layer performs the following functions:

- to reduce the solder lands to the required dimensions
- to prevent solder accumulation with wave soldering and to stop solder flowing away with reflow soldering
- to minimize the chance of short circuits through loose pieces of solder, e.g. solder balls
- to increase the insulation resistance between the tracks or between the tracks and the SMDs (insulation coating).

4.3 Factors Influencing The PCB Layout

4.3.1 Mounting pads

The term "pads" refers to the part of the copper layer on the PCB upon which the SMDs are to be soldered. A "footprint" is a set of pads matching the lead pattern of the component. The pad dimensions are determined by the solderable metallization area of the component, i.e. the solder terminals (See Figure 4.3).

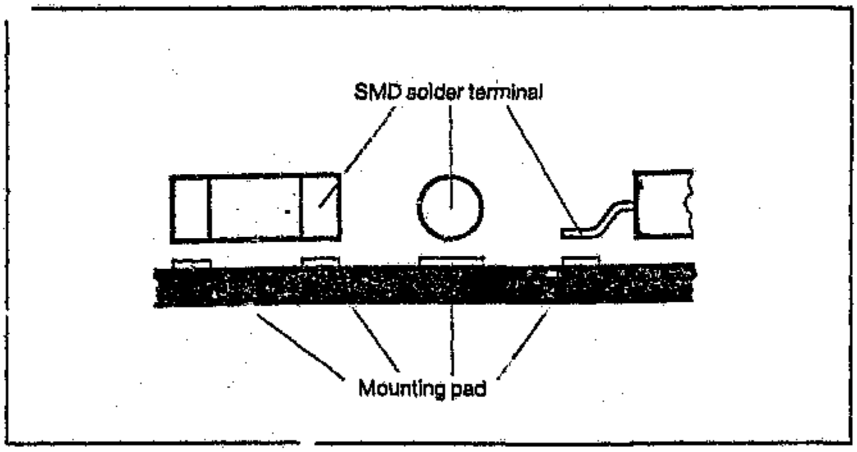


Figure 4.3 Minimum pad size not taking into account influencing factors.

Under ideal conditions, a good solder connection between a component and the PC board can be obtained with pad dimensions conforming to the SMD terminal area. However, various factors, e.g. the effect of tolerances, neccesitate the enlargement of the pad area (See Figure 4.4).

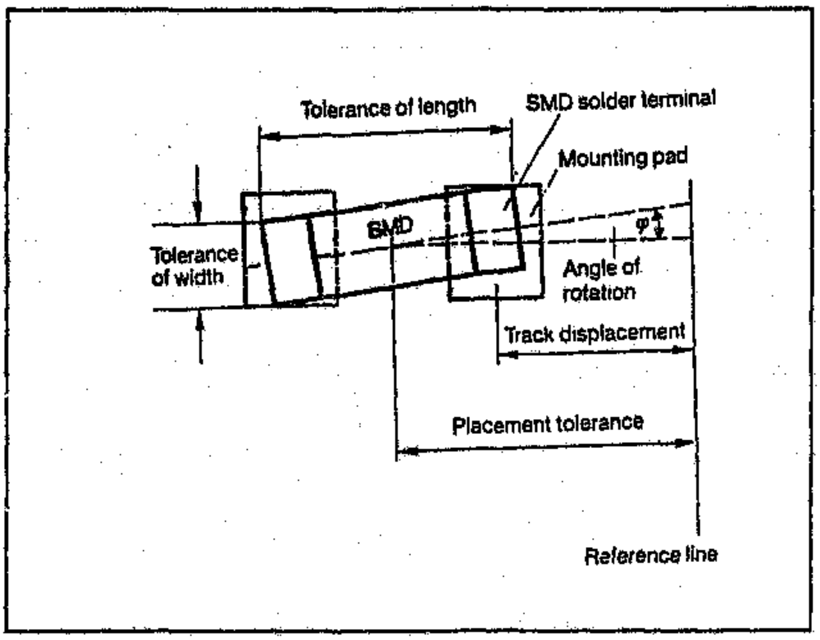


Figure 4.4 Pad size taking into account tolerances.

4.3.2 Components

Componet tolerances are taken into account by the use of pads larger than than those shown in Figure 4.3. From Figure 4.4 it can be seen that the component body as well as the length of the solder terminal determine the pad area.

Therefore, for example in Figure 4.5, upper part, the pads must be enlarged to the right and to the left, because the metallization on the underside of the SMD is almost zero. The solder joint can only be made by using the side areas. On the other hand it is not neccesary that the pads in the lower part of Figure 4.5 cover the entire metallized area of the SMD, because here a small pad area provides a proper solder joint.

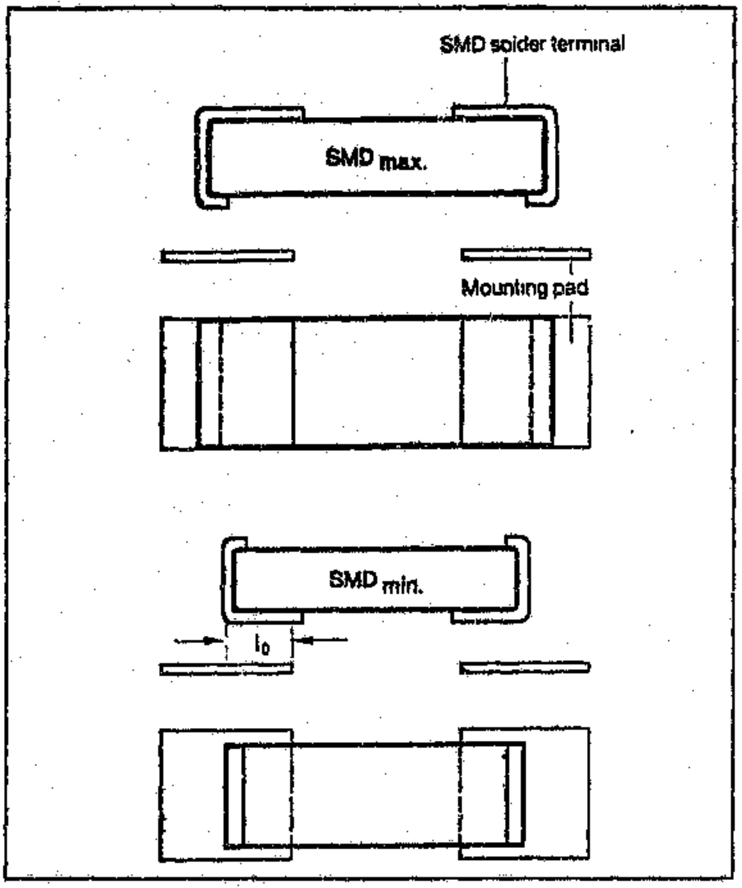


Figure 4.5 Influence of component tolerances on pad size.

4.3.3 Soldering components

Since pad sizes and the component positioning on the PCB also depend on the soldering method, the techniques to be used for soldering the components must be considered at the layout stage. With regard to soldering techniques SMDs can be divided into two groups:

o suitable for wave soldering and reflow soldering

o only suitable for reflow soldering.

In practice this means that SMDs intended for different soldering processes cannot be located on the same side of the PCB.

Wave soldering can be unsuitable for a number of reasons:

- High temperatures
- The packages are not sufficiently sealed to permit complete immersion in flux or solder
- The SMD terminals are closely spaced or are arranged inconveniently, so that short circuiting solder bridges may form (See Section 8).

Although reflow soldering is normally regarded as being suitable for SMDs, problems can occur in certain cases:

- -The tombstone effect, partly caused by the component (See SEction)
- -Insufficient wetting of the solder terminals at the relatively low reflow temperatures.

A detailed analysis of the wave and reflow soldering techniques can be found in Section X.

4.3.4 Via holes

Vias are used to connect tracks on different sides of the PCB. In SMD technology vias can be small in diameter (0.3 to 0.5mm). The via hole should not be located within or immediately adjacent to a mounting pad because it will draw solder into the hole and away from the joint (See Figure 4.6). The via should be be placed near the pad and joined with a thin track.

Via hole diameters should be made as large as possible to obtain highest board yields and lowest costs. For lowest cost, as-drilled diameters of 0.75mm or larger are recommended. The reliability of small holes can be improved by filling the holes with solder.

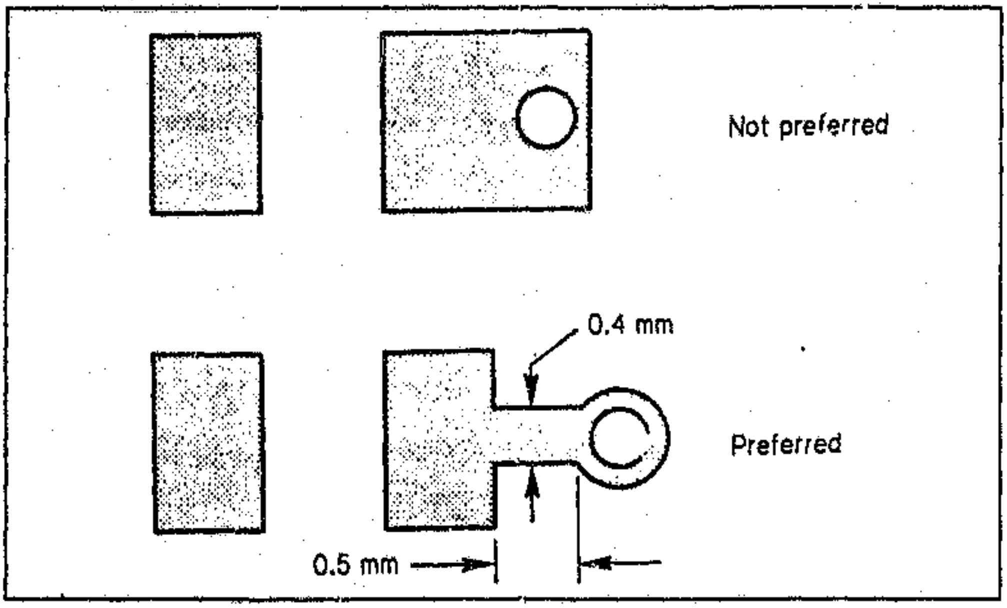


Figure 4.6 Preffered and non-preffered via hole placement.

4.3.5 Test points

As a rule of thumb, the costs for the correction of defects rise by a factor of 10 with each production stage. This shows how important it is to include a test strategy in the development of PCB assemblies (See Section 8).

There are two methods to be considered:

- Fuctional test
- In-circuit test

When considering testing, the assembly's packing density and the smallest grid spacing to be expected should be included right from the start.

The following recommendations are to ensure adequate testing:

- Seperate test points should be provided for all electrical nodes, if possible; minimum diameter 0.9mm, covered with solder. Extended pads, tracks and vias can also be used as test points. See **Figure 4.6**.
- With PCBs having components on both sides, all test points should be brought to the one side of the PCB, preferably the solder side, using vias.
- Miniature test pins for grid spacing 1/20 in. instead of the standard pins for 1/10in. should be avoided. This can be achieved by a staggered or fan-like arrangement of test points. See **Figure 4.7**.
- The test points must have a minimum distance from the pads and components, so that recesses in the bed-of-nails adapter for high components are far enough away from the test pin location holes.

It is not recommended that the test pin be applied directly to the SMD for the following reasons:

- Some SMDs cannot be directly contacted, e.g. PLCCs and MKT capacitors.
- The components, particularly the ceramic ones, can be damaged.
- Defective solder joints may appear acceptable due to the test pin's pressing force.

If the testability of an SMD assembly is considered right at the design stage, automatic testing of the SMD boards will present no serious difficulties.

There are two methods to be considered:

- **Fuctional test**
- **In-circuit test**

When considering testing, the assembly's packing density and the smallest grid spacing to be expected should be included right from the start.

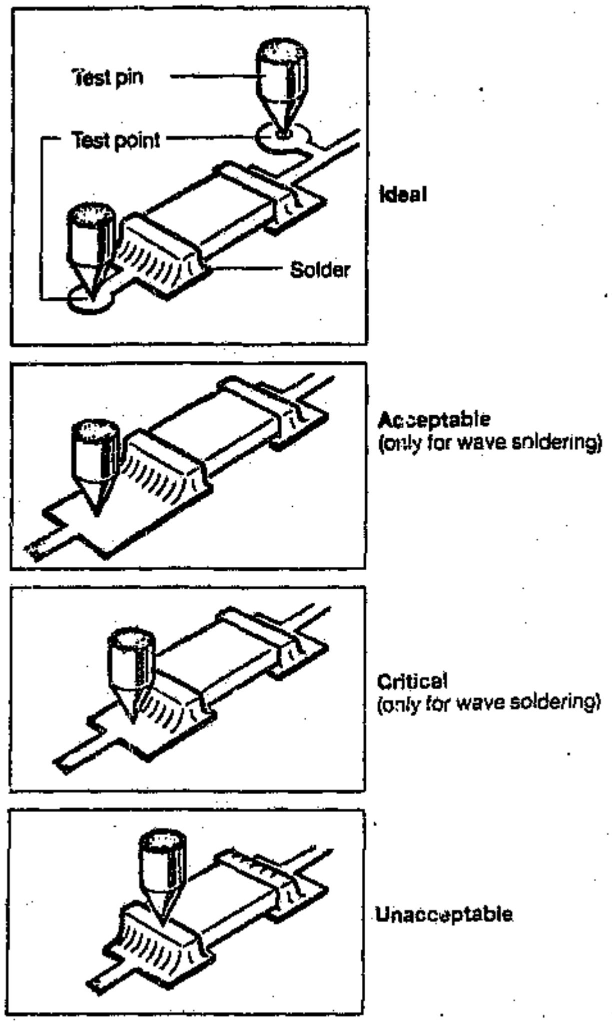
The following recommendations are to ensure adequate testing:

- Seperate test points should be provided for all electrical nodes, if possible; minimum diameter 0.9mm, covered with solder. Extended pads, tracks and vias can also be used as test points. See Figure 4.(
- With PCBs having components on both sides, all test points should be brought to the one side of the PCB, preferably the solder side, using vias.
- Miniature test pins for grid spacing 1/20 in. instead of the standard pins for 1/10in. should be avoided. This can be achieved by a staggered or fan-like arrangement of test points. See Figure 4.7.
- The test points must have a minimum distance from the pads and components, so that recesses in the bed-of-nails adapter for high components are far enough away from the test pin location holes.

It is not recommended that the test pin be applied directly to the SMD for the following reasons:

- Some SMDs cannot be directly contacted, e.g. PLCCs and MKT capacitors.
- The components, particularly the ceramic ones, can be damaged.
- Defective solder joints may appear acceptable due to the test pin's pressing force.

If the testability of an SMD assembly is considered right at the design stage, automatic testing of the SMD boards will present no serious difficulties.



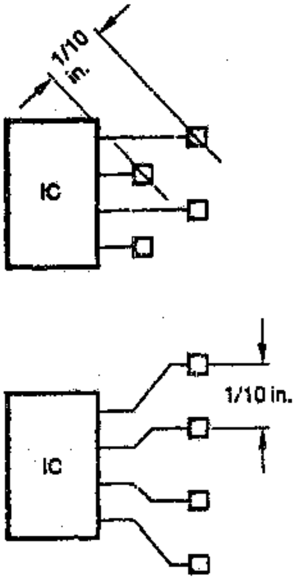


Figure 4.7

Figure 4.6 Test point locations.

4.3.6 Other important factors influencing PCB layout

The high packing density of SMD assemblies generally results in higher power dissipation per unit surface area or volume. Overheating must be avoided by an appropriate layout. With SMD technology there is no possibility of mounting resistors with a clearance from the PCB. However, the power can be divided between a number of resistors. As with through-hole assembly the thermal management can be improved by the use of larger and thicker copper lands. Here, however, more attention must be given to the separation of heat sources.

Another solution is the use of low-power semiconductor circuits, e.g. in CMOS technology. For special applications there are substrates with increased thermal conductivity, e.g. metal substrates or metal-plastic laminates.

However, since these are significantly more expensive than the usual PCB materials, such as FR4, their use is very restricted.

Surface mounting brings about considerable advantages for **RF technology**. Parasitic inductances and capacitance are reduced and the critical circuit parameters can be reproduced more easily. With regard to the circuit certain components should be closely spaced in order to keep the track inductances and capacitance as low as possible. As these demands collide with the requirements of soldering and testing, the pros and cons have to be carefully weighed up.

With **high voltages** a suitable separation from the voltage-carrying parts must be maintained. The relevant standards have to be observed in this respect.

The requirements on the bonding strength of the SMDs on the PC board depend on the particular application. With vibration high accelerations can occur in the case of resonance. The pad size must also be checked with regard to this critical case. By their enlargement both the bonding of the SMDs to the pads and the bonding of the pads to the PCB can be increased.

4.4 Arrangement Of SMDs And Tracks On The PCB

SMDs

The most favourable orientation of SMDs on the PCB is mainly determined by the components themselves and the soldering method used. While vapour phase soldering does not impose any restrictions, the best solder penetration with wave soldering is only achieved by mounting SO packages longitudinally to the direction of the wave.

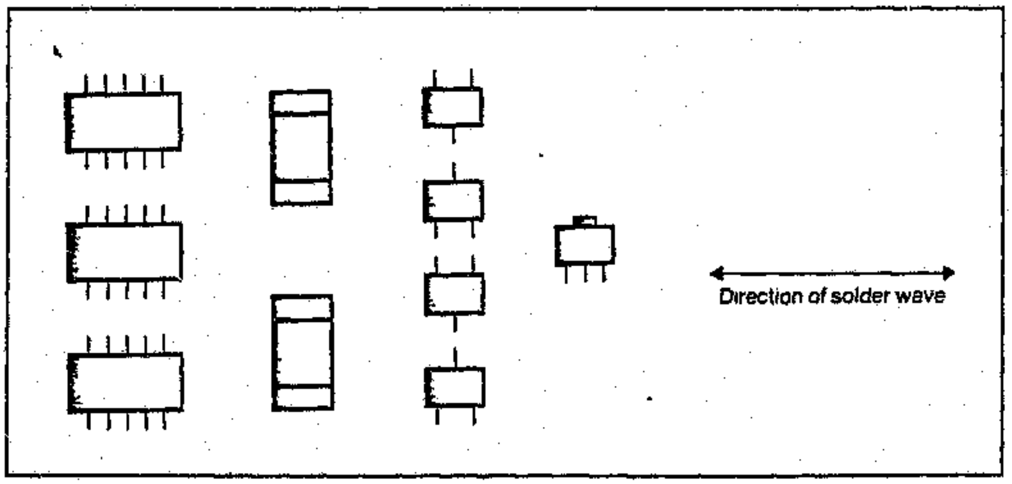


Figure 4.8 Prefered orientation for some SMDs withwave soldering.

Figure 4.8 gives some of the most favorable orientations for avoiding solder shadows and accumulations during wave soldering.

Unacceptable orientations for wave soldering of SMDs are shown in Figure 4.9. If the components cannot be orientated as recommended, with two-terminal components the lengthening of the pad in and against the direction of the movement of the PCB can provide an alternative.

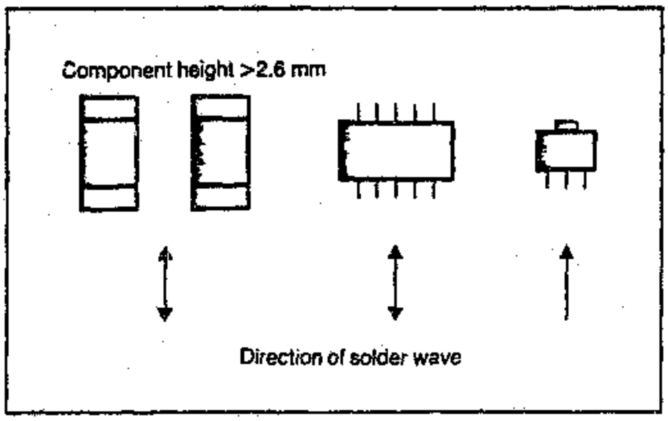


Figure 4.9 Unacceptable orientation with wave soldering.

With IR reflow soldering, the infrared radiation can be shadowed by the component. The problem of solder or IR shadow increases with the component height.

Tracks

Track or conductor widths and spacings for SMT boards are generally smaller than for through-hole technology. Highest board yield is obtained by providing generous margins between actual conductor geometries and the process minimums. Thus, small conductors should be employed only in those regions where they are actually needed.

Conductor routing can have a major impact on soldering yields. In general, anything that alters the symmetry of the land pattern is likely to increase the number of solder defects. For example, exposed conductors entering the component land pattern can act as solder thieves that draw solder away from the land. Steps must be taken to prevent this happening.

4.5 Component Land Patterns

The relatively loose tolerances associated with many surface mount components have made it difficult to define a single footprint or land pattern that works equally well for components at either end of the tolerance range.

A practical approach is to optimize land patterns for the centre of the range of sizes and accept slightly reduced yields with the components near the size extremes. Tightened component tolerances will eventually be neccesary to resolve this conflict.

For many component families it is convenient to represent land pattern geometries in terms of standardized formulae. These formulae can then be used to calculate pattern shapes for numerous components, including types for which no previous land patterns have been developed. **Appendix** 3 contains a detailed guide to using these formulae.

5. COMPONENT PLACEMENT

5.1 Introduction

The component-placement operation, often called "pick and place", consists of all steps neccesary to remove a component from its packaging materials and mount it onto the PCB. Because of the extreme accuracy required (± 0.2mm or less for many component types), automatic equipment is mandatory for all but the smallest production volumes. The basic placement sequence consists of the following eight steps:

a. **Board indexing:** the positioning of a new board onto the system.

b. **Board registration:** alignment of the board to the machines coordinate system.

c. **Component set-up:** presentation of components to pre-determined locations for pickup by the placement tool. Components are stored in special feeders connected to the machine.

d. **Component pickup** extraction of the components from the feeders in preparation for placement.

e. **Component verification:** electrical testing of components.

e. **Component centering:** alignment of the component to the machine's coordinate system.

f. **Component placement:** actual placement of the component onto the PC board.

g. **Board indexing:** removal of the fully loaded board from the machine.

A generalized component-placement machine is illustrated in Figure 5.1. The major components are as follows:

Base

This is the structure on which all other rem[illegible]e parts of the machine are mounted. It must be [illegible] enough to support the rest of the system without allowing excessive vibration, which would degrade placement accuracy.

Component feeders

Components can be presented to the system from a variety of shipping containers. Commonly available feeders accept components in magazines, tape and reel, waffle trays or bulk packaging. These are discussed in detail in Section 5.4.

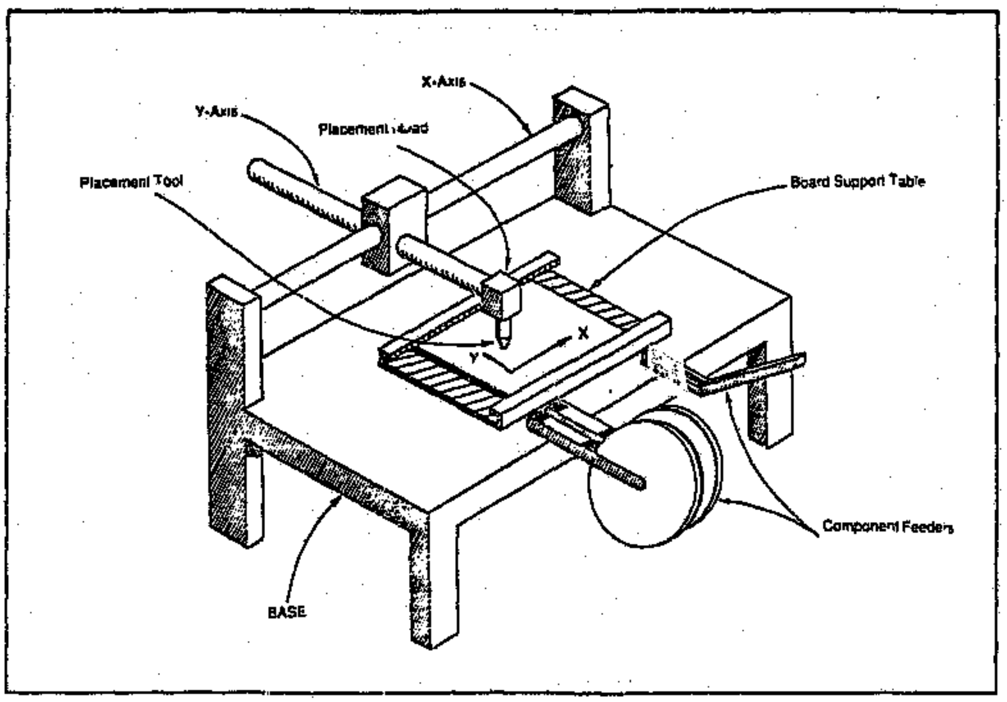


Figure 5.1 Generalized component placement machine.

Board support table

The board or conveyor automatically feeds a new board into the machine and removes the populated board for delivery to the next process step. It is usually adjustable on one side to accomodate different board sizes.

Placement head

The head contains everything necessary to pick components from the feeders and place them onto the PC board. It includes the placement tool, centering jaws (if used) and optional features such as adhesive dispenser, electrical verification fixturing and optical board alignment system.

Placement tool

The heart of the placement head is the tool used to pick components from the feeders and hold them securely during transport to the placement site. A single tool can usually only handle a limited range of part sizes. On machines that must place a wide variety of component types, tools must be changed either manually or under program control.

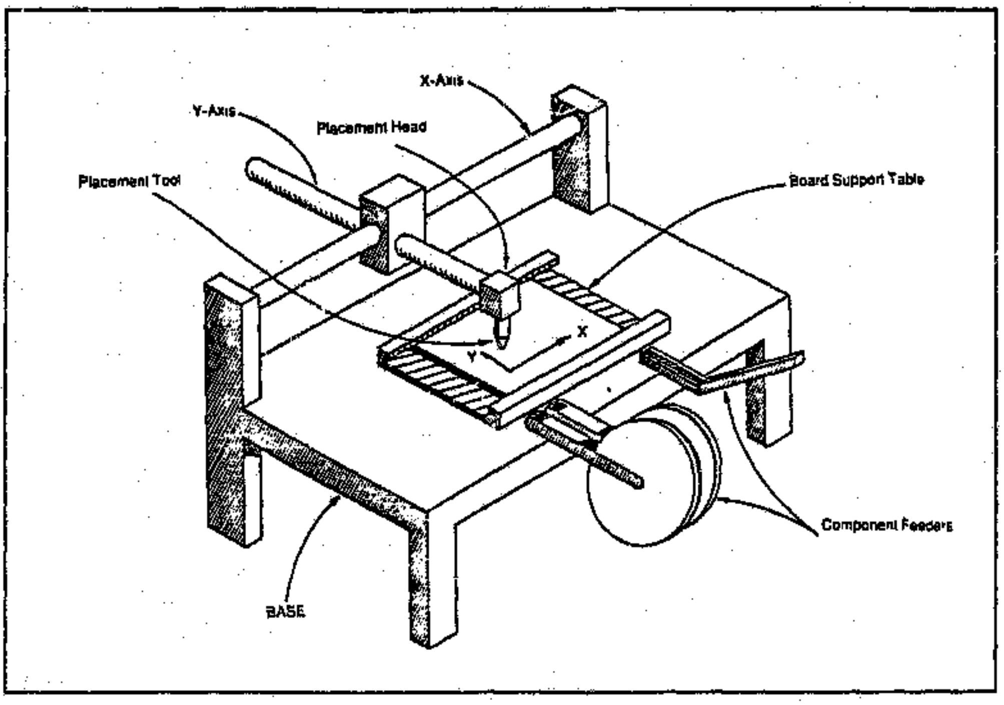


Figure 5.1 Generalized component placement machine.

Board support table

The board or conveyor automatically feeds a new board into the machine and removes the populated board for delivery to the next process step. It is usually adjustable on one side to accomodate different board sizes.

Placement head

The head contains everything necessary to pick components from the feeders and place them onto the PC board. It includes the placement tool, centering jaws (if used) and optional features such as adhesive dispenser, electrical verification fixturing and optical board alignment system.

Placement tool

The heart of the placement head is the tool used to pick components from the feeders and hold them securely during transport to the placement site. A single tool can usually only handle a limited range of part sizes. On machines that must place a wide variety of component types, tools must be changed either manually or under program control.

5.2 Equipment Characteristics

The three characteristics of primary importance in component placement equipment are:

- o accuracy
- o speed
- o flexibility

The accuracy of a machine determines the range of component types that it can place. Its speed sets the capacity of the line and determines how many machines are neccesary to meet anticipated product volumes. Its degree of flexibility determines wether one type of machine can handle all placement needs or wether more specialized machines are needed.

A detailed discussion of theses concepts can be found in Appendix 6.

Although the most obvious requirements are for accuracy, repeatability, reliability and high speed, the list of user requirements expands to include the following:

- automatic load/unload of PCBs
- interface to CAD/CAM
- sense missing or wrong components
- expansion to multihead system
- both "walk-through" and off-line programming
- PCBs up to 12 in. by 18 in.
- automatic epoxy dispense.

5.3 Equipment Categorisation

Categorizing modern pick-and-place systems is complex, specified either by the control techniques, method of operation or type of placement head. Control categories are the simplest, consisting of mechanically programmed, microprocessor controlled and computer controlled.

The method of operation categories consist of two groups: maually-loaded machines with placement rates of 4000 components per hour (cph), and fully automatic machines with placement rates between 9000 and 24000 cph. These two groups are further subdivided as to the number of placement heads, PCB X-Y positioning or head X-Y positioning, feeder X-Y positioning and means of PCB transport from load through placement to unload.

Machines having head X-Y movement and placement are considered to be most efficient primarily because the distance between the points of pick and place can be minimized. However, factors such as head travel speed, PCB size, component mix and feeder configuration will affect placement rate.

The fully automatic category of pick-and-place systems has, by definition, automatic PCB loading, transport and unloading. Most are multi-head systems featuring high placement rates. Common features include individual or tandem-head X-Y control and programmable feeder positioning.

Equipment can be designed to sequentially place individual components. The PC board is positioned below the pick-and-place head using a computer controlled X-Y moving table.(See Figure 5.2a)

Simultaneously placement places SMDs in a single operation. (See Figure 5.2b). A placement station with a number of pick-and-place heads, takes an array of SMDs from the packaging medium and simultaneously places them on the PC board. Simultaneous placement systems offer short cycle times and high throughput, but sequential systems provide higher flexibility.

A variation of sequential placement is known as in-line placement. (See Figure 5.2c). In this mode, a board progresses through a series of individual placement heads, each of which places only a single component. As with simultaneous placement systems, in-line systems are used when high production volumes must be achieved.

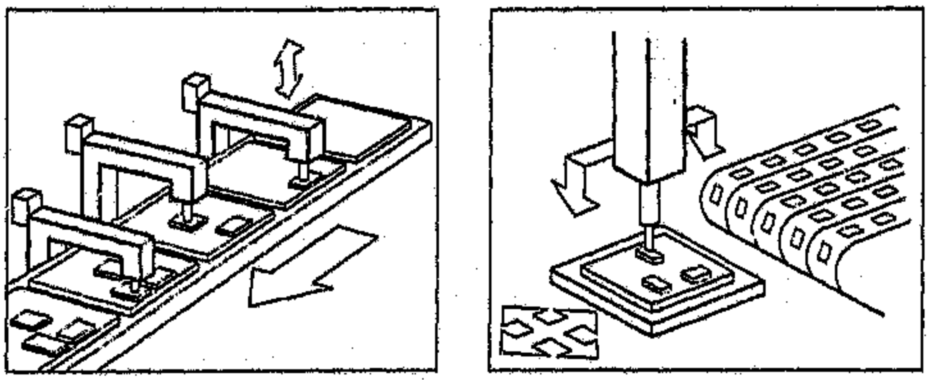


Fig. 3(a) In-line placement.

Fig. 3(b) Sequential placement.

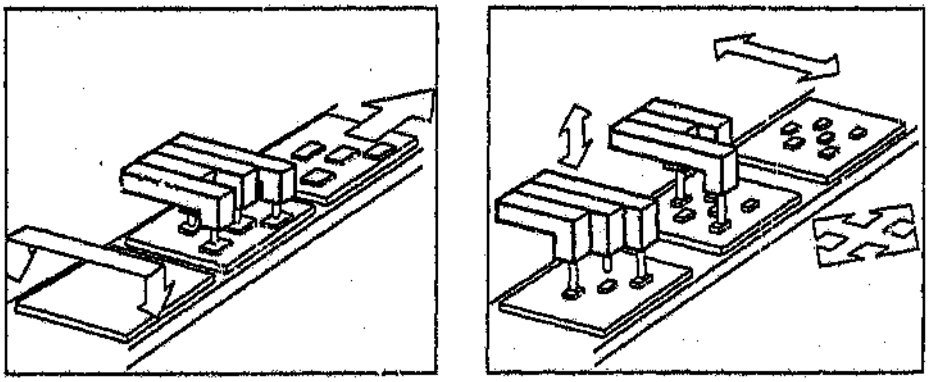


Fig. 3(c) Simultaneous placement.

Fig. 3(d) Sequential/simultaneous placement.

Figure 5.2 Placement options for "pick and place" machines.

5.4 Component Feeding Systems

The success of the component-placement operation depends on reliable presentation of components. Components that are skewed, inverted or otherwise misorientated cannot readily be picked from their feeders. Operator intervention may be neccesary to clear the resulting feeder jams. In some cases, the machine will not detect the error and will attempt to complete the placement. Such defects may not be found until the end of the manufacturing process, when repair is much more difficult and costly.

Component feeding must be viewed as a system consisting of the component shipping container as well as the actual mechanical feeder. If the componnent cannot easily be extracted from the shipping container, or if it moves so freely as to become misoriented, the placement machine will not be able to compensate. A variety of component shipping containers have been developed. Those most commonly used in SMT are:

o tape and reel
o magazine (stick)
o bulk
o matrix tray

These are discussed in Appendix 10

6. SOLDERING IN SURFACE MOUNT TECHNOLOGY

6.1 Introduction

The size and complexity of electronic assemblies is steadily increasing, which also implies a growing number of solder joints. Today, some hundred solder joints per PCB are not uncommon. Consequently, an economic soldering process can only be achieved by automatic soldering equipment. In addition, exacting demands are made on quality and reliability.

Surface mount with it's special requirements places new demands on the solder joint. Essentially, SMD technology means automated production of PCB assemblies. Minimization of manual work is mandatory. Rework (repairs), however, constitutes a major part of manual work and has to be reduced reduced in SMD to an extent up to now uncommon. The aim is yields of 95% (Q 95) i.e. at least 95% of all finished PCB assemblies must be faultless. In order to achieve this percentage, soldering techniques have to feature extremely low failure rates (10..20dpm).

In conventional PC board production, a number of soldering methods and special soldering equipment were developed which are now used in a modified form for surface mounting. However, the choice of the "correct" soldering technique has not become easier. Each procedure is suited to typical applications and the benefits and drawbacks have to be carefully evaluated. With an increasing variety of components to be soldered and with a growing number of special requirements, such as packing density, it will be necessary to use more than one soldering process.

To clearly understand the impact that the soldering process has on the PCB and components, one must understand some of the soldering fundamentals. This has been detailed in Appendix 4.

6.2 Solder Profiles of SMDs

The solder profiles of different devices are presented in Appendix 8. The page order of the profile section has been arranged to be similar to the frequency with which the devices are used in industry. Profiles are described as:

Unacceptable - none or too little solder

Acceptable (min) - minimum acceptable amount of solder present to make a successful joint.

Acceptable (max) - maximum amount of solder that would normally be acceptable.

Excessive - too much solder has been used, but it does not necessarily mean the joint is defective. It does however indicate that process parameters should be adjusted.

7. AUTOMATIC SOLDERING TECHNIQUES

7.1 Introduction

There is no general answer to the frequently asked question of which soldering technique and which equipment is the most suitable for surface mounting. Each technique and each type of equipment qualifies for specific applications and is consequently less suitable for others. Therefore, the soldering methods and equipment have to be carefully selected, taking into consideration all factors involved in the assembly.

7.2 Summary

Automatic soldering can be divided into bath soldering and reflow soldering techniques.

o Bath soldering

- Conventional wave soldering
- Dual wave soldering
- Drag soldering
- Dip soldering

o Reflow soldering

In reflow soldering all kinds of heat transfer can be used; in practice the different methods are frequently combined.

- Radiation
- Conduction
- Convection
- Condensation

The heat can be input

- from all directions (simultaneously)
- from above (local, sequential or over the surface simultaneous)
- from below (over the surface, simultaneous)

The most common reflow soldering techniques are listed in Table 7.1

7.3 Wave Soldering

Conventional wave soldering equipment can be used for the soldering of surface mount devices, but with increased circuit density and the resulting reduction of space between conductors, non-wetting and solder bridging can become a problem.

7. AUTOMATIC SOLDERING TECHNIQUES

7.1 Introduction

There is no general answer to the frequently asked question of which soldering technique and which equipment is the most suitable for surface mounting. Each technique and each type of equipment qualifies for specific applications and is consequently less suitable for others. Therefore, the soldering methods and equipment have to be carefully selected, taking into consideration all factors involved in the assembly.

7.2 Summary

Automatic soldering can be divided into bath soldering and reflow soldering techniques.

o Bath soldering

- Conventional wave soldering
- Dual wave soldering
- Drag soldering
- Dip soldering

o Reflow soldering

In reflow soldering all kinds of heat transfer can be used; in practice the different methods are frequently combined.

- Radiation
- Conduction
- Convection
- Condensation

The heat can be input

- from all directions (simultaneously)
- from above (local, sequential or over the surface simultaneous)
- from below (over the surface, simultaneous)

The most common reflow soldering techniques are listed in Table 7.1

7.3 Wave Soldering

Conventional wave soldering equipment can be used for the soldering of surface mount devices, but with increased circuit density and the resulting reduction of space between conductors, non-wetting and solder bridging can become a problem.

Table 7.1 Reflow soldering options.

Heat input / Heat transfer	Heat input from below	Heat input from above		Heat input from all directions
	over surface, simultaneous	local, sequential	over surface, simultaneous	simultaneous
Radiation	IR radiation	Laser beam	IR radiation	IR radiation
Condensation	–	–	–	Vapor phase
Conduction	Hot plate Hot belt	Thermode (stamp)	Thermode (bracket)	–
Convection	–	Gas nozzle	Hot air Hot gas	Hot air, circulating air Hot gas

The dual wave method was developed expressly for the soldering of surface mount components and to overcome these problems. The principle of the dual solder wave is that contact with the first wave, usually a turbulent wave with high vertical velocity, consumes or drives out most of the flux and ensures good solder contact with both edges of the device. The second, laminar wave, then completes the formation of the solder fillet and removes bridges.

In the wave soldering process, the PCB is transported over a fountain of continuously flowing molten solder, as shown in Figure 7.1. The height of the conveyor is adjusted so that as the board passes over the wave, its entire underside is washed by the solder. In the case of through-hole components, the solder wets the protruding component leads and is drawn up into the plated through-holes. Surface mount components are soldered by adhesively attaching them to the underside of the board and immersing them directly in the solder wave.

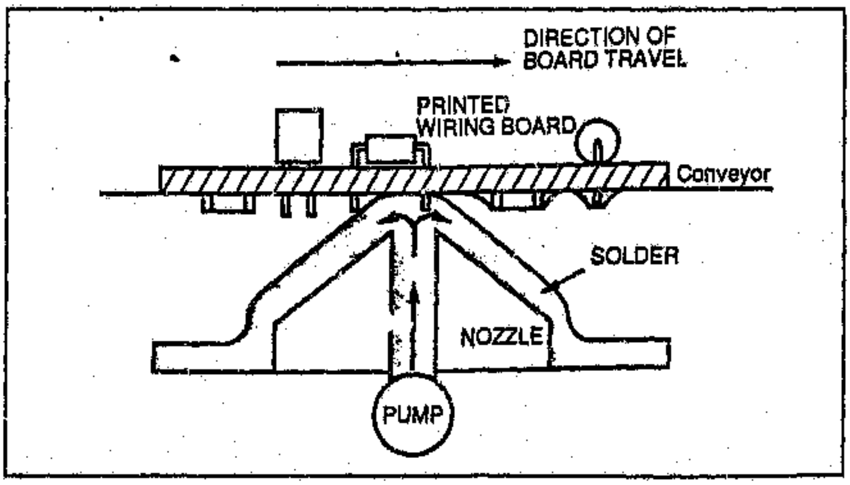


Figure 7.1 Priciple of wave soldering.

The complete wave soldering process consists of three steps which are normally performed within a single machine, as illustrated in **Figure 7.2**. The steps are:

- o Flux application
- o Board preheat
- o Wave soldering

In a typical dual-wave soldering process, the preparation of the substrate is followed by the application of flux; usually based on a resin dissolved in an organic solvent such as isopropanol.

This assures good wetting of the soldering surfaces. (See **Appendix 3** for a detailed description of fluxes)

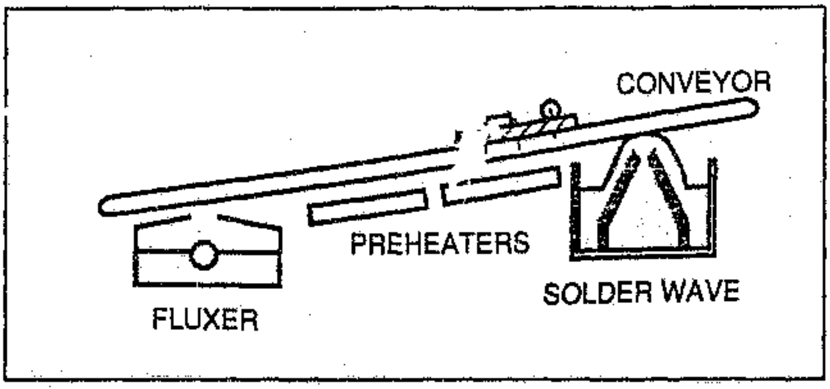


Figure 7.2 Schematic of complete wave soldering system.

Next, the board is preheated. This serves several purposes. First, it reduces the flux to the required viscosity, and second, it heats the PCB and components to a predetermined temperature to reduce thermal shock and promote faster wetting. Pre-heating also minimises the time spent at soldering temperature and so prevents dissolution of component metallization (sometimes known as leaching).

A variety of pre-heating systems are available, using forced hot air, quartz lamps, or heated panels. During the pre-heat time, usually lasting about 10 seconds, the temperature of the PCB is raised to approximately 80°C. These values are approximate and will depend largely upon the composition of the substrate and the nature of the components. (See detailed description in **Appendix 8**)

After pre-heating, the underside of the substrate is brought into contact with the molten solder, first with the turbulent wave and then with the laminar wave.

Experience has shown that a contact angle with the solder waves of approximately 7° gives optimum results. **Figure 7.3** shows the division of the solder stream into a first and second wave. (See **Appendix 7** for detailed description of wave soldering)

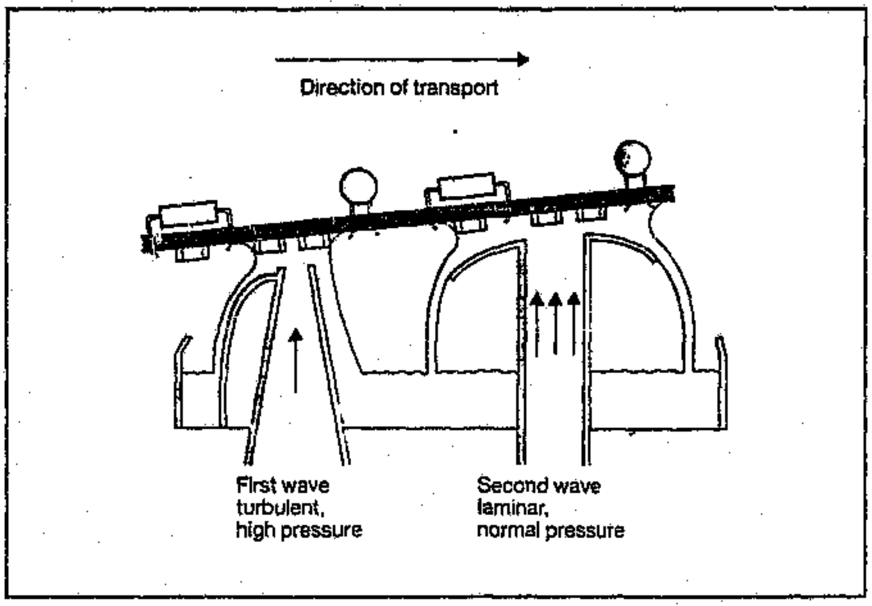


Figure 7.3 Division of solder stream into a first and second wave.

After the second and final soldering step, the PCB assembly is allowed to cool down in free air.

7.4 Reflow Soldering

Unlike wave soldering, the reflow soldering process requires the deposition of measured quantities of solder directly onto the PCB pad. Components are then placed and the final interconnection is formed by fusing or reflowing the solder by elevating the temperature above the melting point of the solder, at which time it wets to the metallic surfaces. When the temperature is lowered, the solder solidifies, completing the joint.

Reflow soldering offers several benefits. A primary advantage over wave soldering is the fact that solder is applied only where it is needed. This is especially important for components that would be damaged if immersed in molten solder. Another benefit is that the temperature profile seen by the assembly can be more precisely controlled. SMDs sensitive to thermal shock are more compatible with a reflow process than a wave process.

7.4.1 Temperature Profile

To prevent component damage during exposure to high temperatures, the temperature profile during reflow must be carefully controlled. An ideal profile is shown in Figure 7.4.

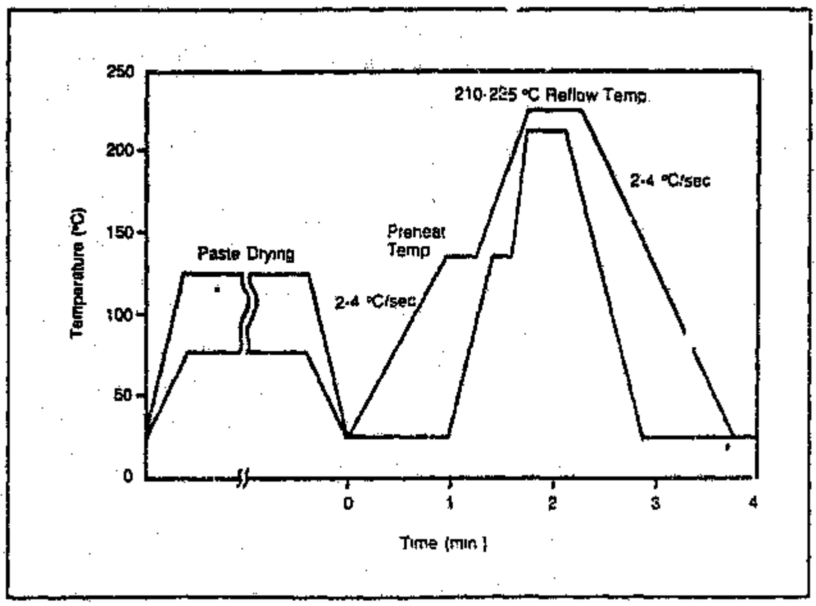


Figure 7.4 Idealized reflow temperature profile.

It consists of a slow rise to an intermediate temperature (the preheat zone), a relatively quick excursion above the liquidus temperature (just long enough to ensure complete melting of all solder), and a gradual cooling back to ambient.

The specific nature of the profile can vary depending how it is measured. The profile recorded by a bare thermocouple may be vastly different from that recorded recorded by a similar thermocouple affixed to the PCB. The latter method is preferable using more than one thermocouple at different locations on the board.

An often overlooked fact is that the shape of the reflow temperature profile is more important than the method used to achieve this profile. Vapour phase, infrared, thermal conduction and thermal convection have all been successfully used to solder surface mount assemblies. Most problems attributed to particular processes are a result of large deviations from the ideal profile rather than an inherent defect in the technique.

7.4.2 Preheating

Adequate preheating is important for several reasons. Components such as ceramic capacitors are sensitive to thermal shock and can be damaged by too rapid a temperature rise. Rapidly increasing temperatures also disturb the thermal equilibrium of the assembly and cause small components to reach the liquidus temperature more quickly than the larger PCB. When this occurs, the molten solder will preferentially wet the component lead and may flow along the lead and away from the joint.

This "wicking" phenomenom is observed primarily on J-lead components, such as PLCCs (see Section X (first sec on components)). Finally, preheating is neccesary to achieve proper flux activation. Inadequate preheat time and temperature will prevent the flux from removing surface oxides at the joint prior to reflow.

7.4.3 Paste Drying

Solder paste contains volatile solvents that improve screenability and extend the working life of the paste. If these solvents remain in the paste during reflow, they will boil violently and spatter solder balls across the surface of the board. To prevent spattering, the paste must be thoroughly dried.

The optimum drying profile depends on the nature of the volatile elements contained in the paste. These in turn are determined by the desired screening properties and working life of the paste. Pastes with low boiling-point solvents may require only 5-10 minutes of drying time at 60-80°C. Higher boiling-point solvents may need to be dried for as long as an hour at temperatures as high as 125-150°C. The paste manufacturer should be consulted for specific recommendations.

The drying process should always be distinguished from the preheat portion of the reflow process. Drying time and temperature must be selected on the basis of the composition of the solder paste. The preheat profile is governed by the need to minimize thermal shock and maintain thermal equilibrium at all points on the assembly. It is rare that a single process profile can be employed to simultaneously achieve both results.

7.4.4 Reflow Alternatives

Several techniques for reflow soldering are in common use. Those most frequently employed in production of SMT applications are:

- o thermal conduction
- o vapour phase
- o infrared
- o laser
- o thermal convection

One of the oldest reflow techniques is the application of heat via thermal conduction. In its simplest form this consists of nothing more than a hot soldering iron applied to previously deposited solder. The most elaborate equipment of this type employs conveyors to automatically transport the boards across a temperature controlled hot plate and through a cooling section to allow the solder to solidify.

Reflow by thermal convection is [illegible] method which uses heated air as the thermal [illegible] medium. An ordinary resistance-heated ove[illegible] simplest example of a convection reflow syste[illegible]

In condensation soldering, common[illegible] vapour phase reflow, boards are immersed in [illegible]urated vapor of an inert liquid whose boiling point [illegible] higher than the melting temperature of the solder. A typical reflow temperature is 215°C. Since the entire process occurs in an inert environment, there is little potential for oxidation of the molten solder.

Solder reflow by direct infrared (IR) radiation is a more recent development. For these ovens, 90% or more of the heat is transferred through direct radiation of near-visible IR energy. A special form of direct radiation is laser soldering. In this approach, the energy from a high-power laser is directed onto the joint. Reflow can occur in a fraction of a second, but the process is relatively slow because each joint must be soldered individually.

A detailed look at each of the above mentioned methods can be found in Appendix 5.

7.4.5 Comparison Of Reflow Techniques

Table 7.2 compares the attributes of the various reflow processes. Most mass soldering is performed either with a vapour phase or reflow system.

Reflow method	Advantages	Disadvantages
Vapour phase	Insensitive to variations in board thermal mass. Uniform temperature across entire board. Well controlled maximum temp. Insensitive to variations in board geometry.	High capital equipment cost. High operating cost due to expense of vapour phase fluid Fluid decomposition generates toxic by-products. Little control over temperature profile
Infrared	Precise control of reflow temperature profile. Moderate capital equipment cost. Moderate operating cost. Improved heating dynamics compared to vapor phase.	Equipment parameters must be readjusted for boards of different thermal mass. Possible to overheat board if the equipment is misadjusted. Heating rate depends on colour of surface IR energy is blocked by tall components.
Laser	Heat is confined to joint region. Compatible with temperature-sensitive devices.	Low throughput due to each joint having to be reflowed individually. Very high equipment cost.
Conductive belt	Low equipment cost Simple operation. Moderate control over temperature profile.	Incompatible with organic PC boards. Small maximum board size.

Table 7.2: Comparison of reflow methods.

7.5 Post-Solder Cleaning

The quality and reliability of a PCB assembly is a strong function of its initial cleanliness level. Inorganic residues such as flux activators (See flux SEction) cause a reduction in insulation resistance and an increase in current leakage. In combination with atmospheric moisture, they promote corrosion of metallic surfaces. Organic residues, such as rosin, greases or oils form an insulating film that can prevent proper electrical contact between mating surfaces of connectors, switches or relays. For all but the most benign operating environments, these contaminants must be thoroughly removed.

Cleaning processes can be divided into two basic categories: aqueous and solvent. Aqueous processes use water as the primary cleaning fluid, while solvent processes use chlorinated or fluorinated hydrocarbon liquids.

For PC boards, the major contaminant is flux residue from the soldering operation, so the choice of cleaning method depends largely on the type of flux used. Rosin-based and synthetic fluxes are best removed in a solvent cleaning process. Aqueous processes are employed when water-soluble organic acid fluxes have been used.

Of primary importance is compatibility with the components to be cleaned. Some types of plastics and marking inks are dissolved by the more aggresive chlorinated solvents, while certain other components are not suitable for water cleaning.

Another concern is the ability to comply with local environmental regulations. Solvent processes release potentially harmful chlorofluorocarbons into the atmosphere, while aqueous processes discharge heavy metals into the local water supply.

8. INSPECTION AND TESTING STRATEGIES AND PROCEDURES FOR SMD ASSEMBLIES

8.1 Introduction

To avoid the prospect of building large quantities of defective products, immediate feedback on process performance is essential. By understanding the current behaviour of the process, problems can be anticipated and corrective action taken before defects arise.

Within the surface mount factory, there are two primary techniques for obtaining process feedback. The first is by monitoring physical characteristics, such as solder bath temperatures, conveyor speeds, and the like. The second is by examination of the end product through physical inspection and electrical testing. Both approaches are essential in a well-run factory.

Inspection and testing address two different aspects of production performance. The inspection process focuses on physical variations that could lead to mechanical reliability problems. Although often associated with simple visual inspection, several other approaches can be employed. These include machine vision, X-ray, and infrared techniques.

Electrical testing confirms that the assembly is electrically functional by comparing the PCB circuit performance against predetermined limits. The two most common approaches are the functional test, which tests the operational characterisitics of the whole assembly, and the in-circuit test, which measures the performance of each component on an individual basis.

Niether inspection or test, by itself, can identify all defects that are likely to occur. Many electrical defects are not manifest as obvious physical anomalies and so would escape detection at a physical inspection point. Conversely, many reliability problems do not show up as immediate electrical problems but are readily apparent when visually inspected. Well run factories make intellegent use of both techniques to efficiently collect the optimum amount of data.

Several systems are currently available to the manufacturer, including short-circuit testers, in-circuit testers, in-circuit analysers and functional testers. **Figure 8.1** shows a bar-chart giving a comparison of percent fault detection and programing time for various automatic test equipment (ATE).

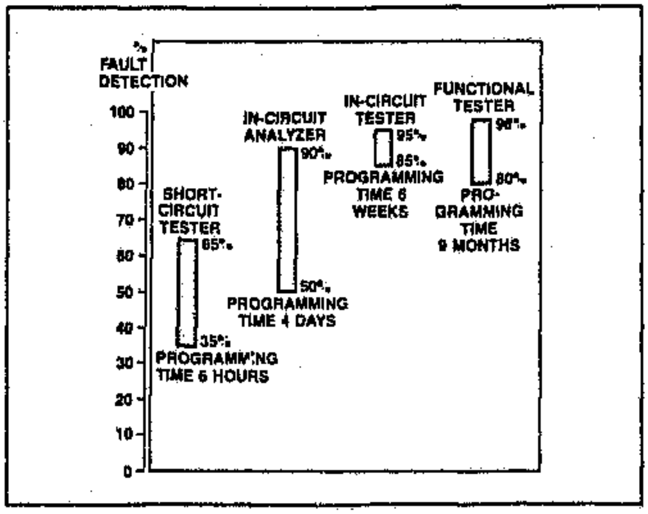


Figure 8.1 Comparison of % fault detection and programming time for various ATE systems.

8.2 In-Process Inspection

The decision of how and where to inspect must be based on economics. The cost of inspecting must be weighed against the potential cost of not identifying defects until a later process step. Those processes with greatest overall impact on the finished product are the ones most likely to benefit from immediate feedback. The most common inspection points are illustrated in Figure 8.2.

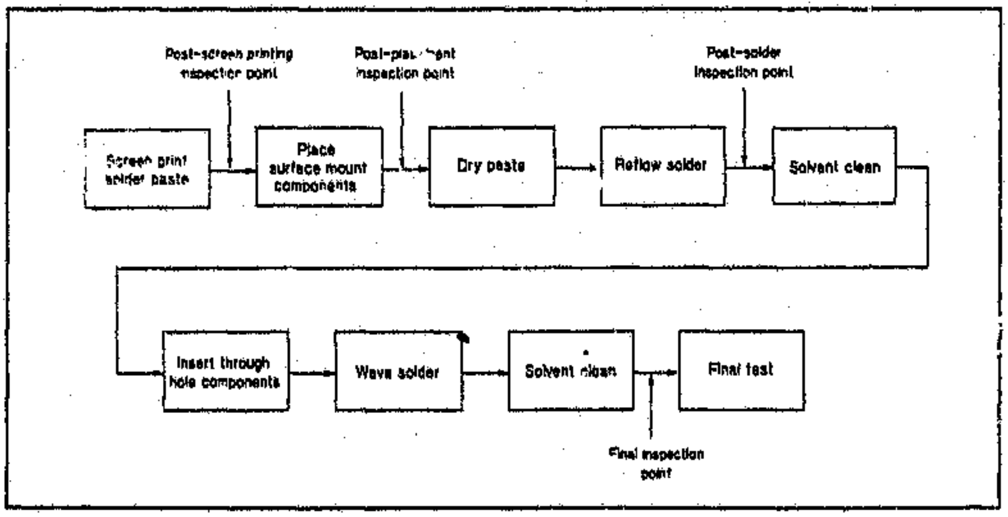


Figure 8.2 Typical in-process inspection points.

In order of priority, they are:

- Final inspection
- Post-solder inspection
- Post-placement inspection

- Post-screen printing inspection.

The details of each of the above are given in **Appendix 2**

8.3 Functional Testing

In this method, the performance of the entire circuit is analyzed on a system that simulates the intended operating environment. All electrical contact is normally made through the PCB edge connector. A series of input signals serves as stimuli, and the output responses are measured and compared against acceptable limits. The result is a pass/fail indication of board performance.

One disadvantage common to all forms of functional testing is a limited ability to diagnose failures. It is usually possible to trace a problem only to a particular region of the circuit, rather than to a specific component. Some functional testers have no diagnostic capability at all, but merely indicate wether the board has passed or failed the test. Another concern is with the time required to write the test program for a new board. Complex assemblies may require anywhere from a few weeks to as long as nine months for program development.

8.4 In-Circuit Testing

In-circuit testing does not check the performance of the circuit as a whole but rather of each component individually. The board is accessed by a series of probes (called a bed-of-nails fixture) that make contact with all circuit nodes. Test signals are applied across succesive combinations of nodes and the responses measured. Defects can ordinarily be traced to a specific component or small cluster components.

Digital components are tested by applying a series of test patterns, called test vectors, to the component inputs. The outputs are measured and compared to the expected truth table for the device. Many digital in-circuit testers can function at clock rates similar to those encountered in actual operation.

8.5 Anticipated Fault Profile

The measure of success for a test technique is how well it detects the types of faults most likely to occur. The distribution of anticipated faults, called a fault profile, varies somewhat depending on the assembly process employed. The precise impact of the various faults depend on the specifics of the board design and the manufacturing facility. For many manufacturers, a fault profile similar to that, in **Figure 8.3** has been found to representative.

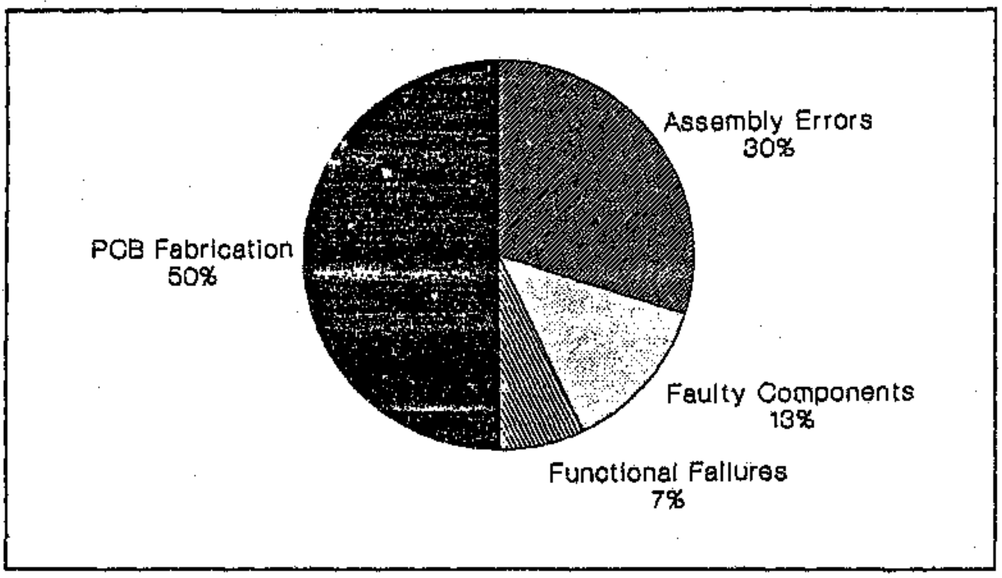


Figure 8.3 Typical fault profiles for PC boards.

Defects from the printed circuit board production and soldering processes account for for about half of all defects. Assembly and wiring faults account for another 30%, while defective components are the cause about 13% of the time. Functional failures account for only 7% of all defects.

8.6 Defects in Soldering SMDs

It is recognised that certain defects are more common to specific soldering techniques. These can be categorised into those which the solder is preplaced and then heat is applied, and those where the application of solder and heat are simultaneous. In the latter category are wave soldering, hand soldering and most rework techniques. For the former, the solder is applied as a plating, paste or preform, and heat is applied by one of several methods including vapour phase, hot belt and infra red to reflow the solder.

Each technique is prone to produce certain classes of defects; the reflow methods give rise to misalignment whereas bridging and shadowing are defects more often found when using wave soldering. A brief description of each type of defect is covered in the following sections.

8.6.1 Solderability

Soldering is a process which depends upon the formation of a mettalurgical bond between the component and the substrate (or PC board). The formation of these bonds depends critically upon the ability of the solder to wet both the leads and the pads (See Section XXX). There are many reasons for poor wetting including excessive growth of the intermetallic layer, dirt, inappropriate flux, and diffusion problems. See Figure 8.4

8.6.2 Misalignments

Misalignments of components with respect to their pads are associated mainly with reflow techniques. On reflowing, the component swims in a pool of molten solder, and is finally positioned only when the solder solidifies. Misalignment is caused by different surface tension forces acting at each pad resulting in rotational movement. Another type of misalignment occurs when one end of the device lifts from the pad, this is sometimes reffered to as 'Drawbridging', 'Tombstoning' or the 'Manhattan Effect'.
See Figure 8.5

The two main factors promoting misalignment are:

Improper design of component pads; these must be of the correct size to achieve good joints, and minimise component movement during soldering.

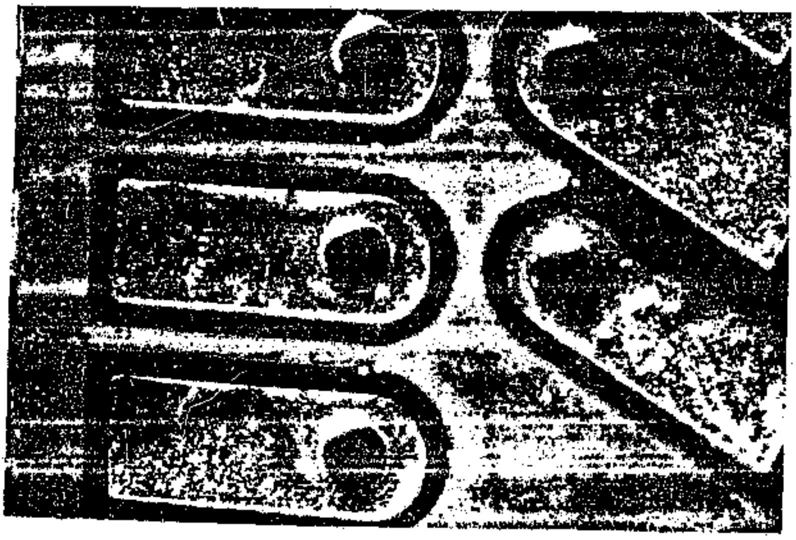
Differential heating will cause one end to melt before the other and produce component lifting.

8.6.3 Short Circuits/Bridging

This type of defect occurs particularly with wave soldering. If the wave conditions are not correctly adjusted and the component design has closely spaced leads, then a bridge can occur between two or more leads. See Figure 8.6

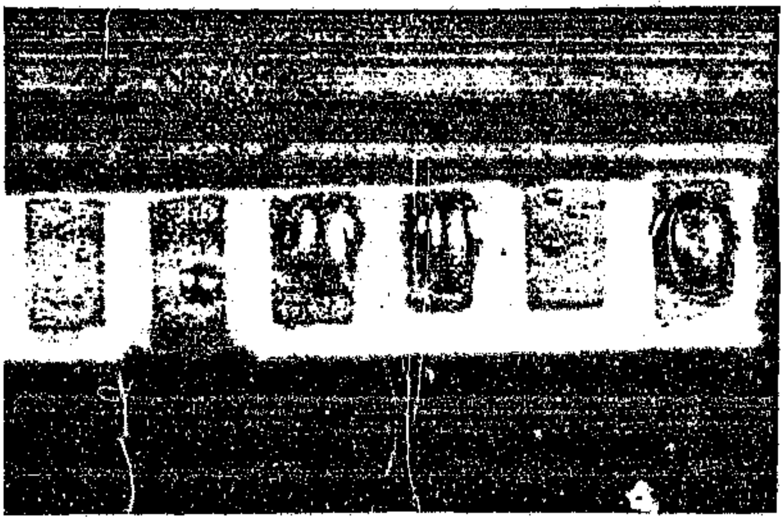
8.6.4 Shadowing and Skips

Shadowing is caused by components obstructing the wave path.Wave soldering can produce these defects and it is neccesary to orientate the surface mount device correctly with respect to the direction of travel of the PC board over the solder wave. See Figure 8.7



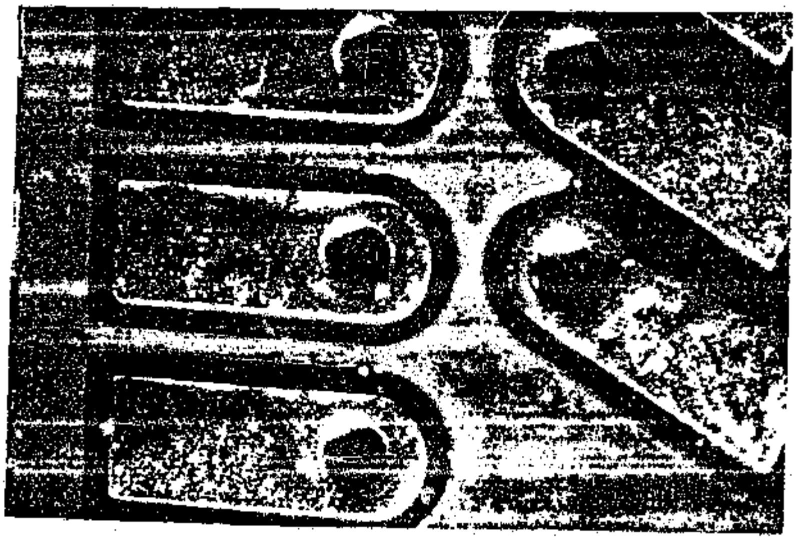
a) Substrate exhibiting dewetting.

b) Non-wetting of one pad exposing the copper underlay.



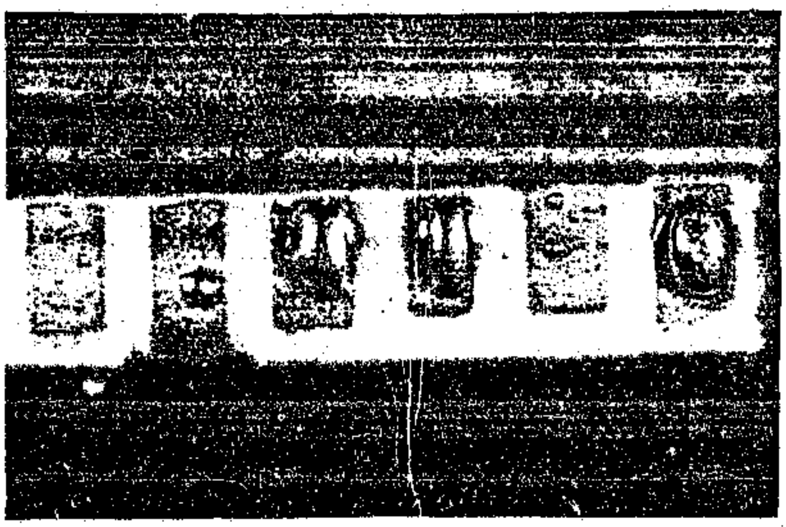
c) Non-wetting of metallisation.

Figure 8.4 Defects associated with the solderability of the substrate.



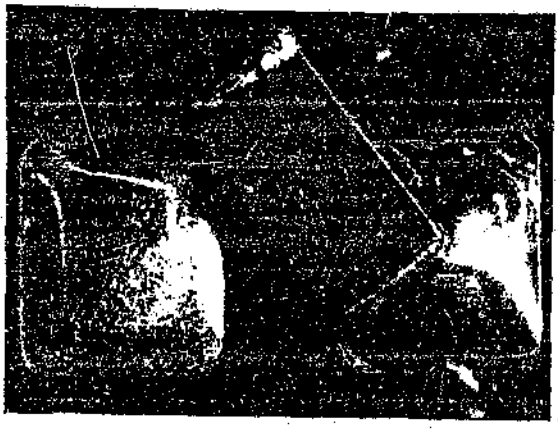
a) Substrate exhibiting dewetting.

b) Non-wetting of one pad exposing the copper underlay.

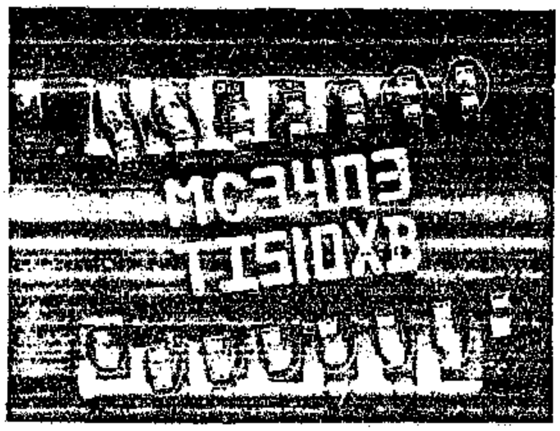


c) Non-wetting of metallisation.

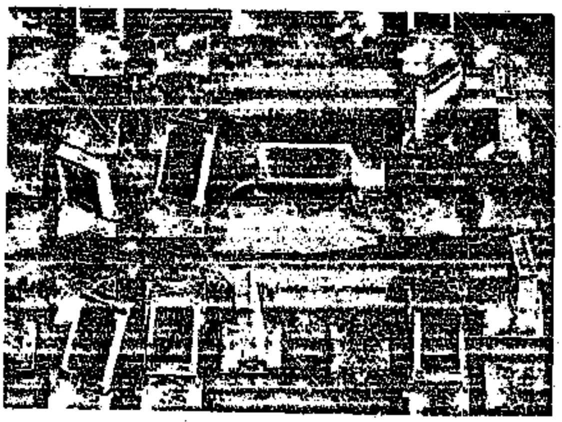
Figure 8.4 Defects associated with the solderability of the substrate.



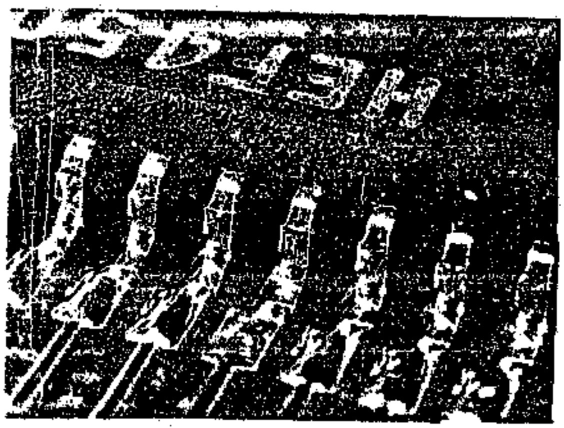
a) Rotational misalignment of a chip component.



b) Rotational misalignment of a SOIC.

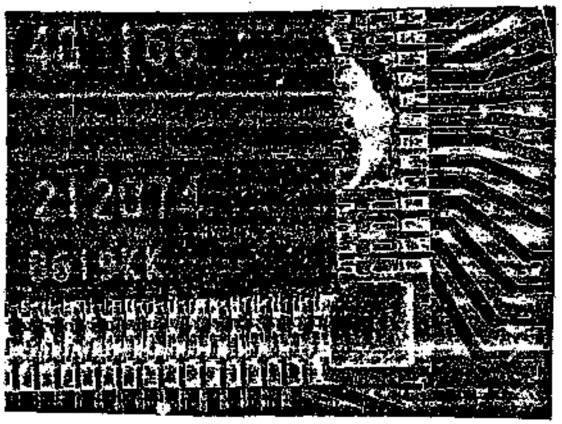


c) Uplifting of chip components (Manhattan effect).



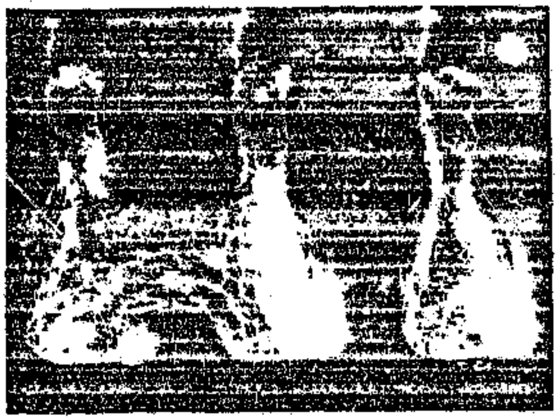
d) Translational misalignment of a SOIC causing a mismatch between lead and pad.

Figure 8.5 Defects associated with misalignment of components.

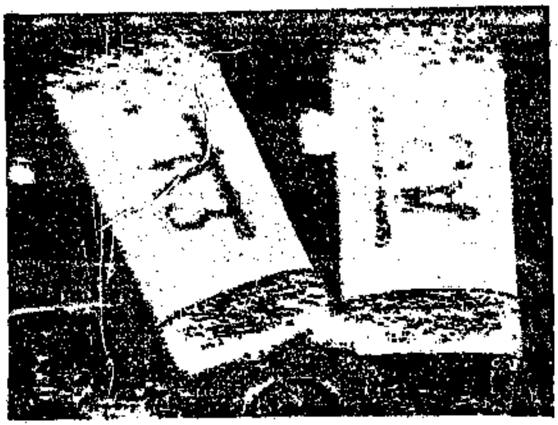


a) Bridging of quad pack leads.

b) Bridging of SOIC leads.

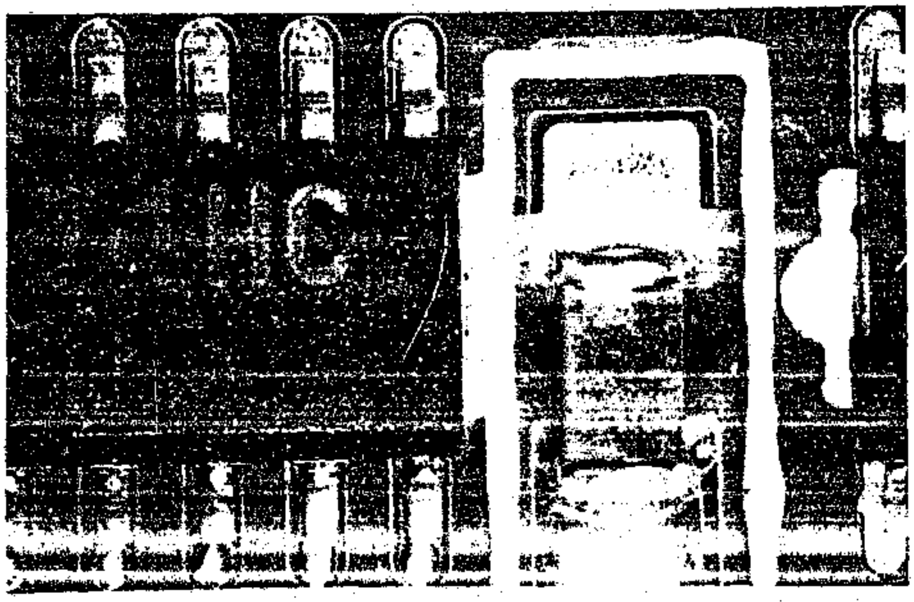


c) Bridging on a LCCC.

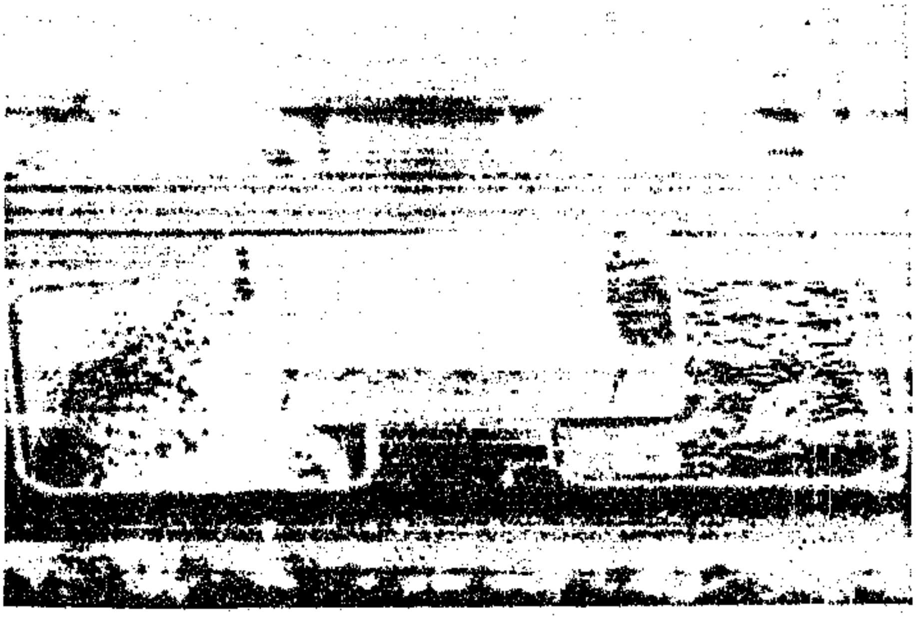


d) Bridging of two chip capacitors causing a short circuit.

Figure 8.6 Defects associated with bridging of components.



a) Shadowing preventing soldering of one side of the component.



b) Absence of solder from one of the pads due to a skip.

Figure 8.7 Defects associated with shadowing and skips.

PRACTICAL IMPLEMENTATION

1. Software development of simulation program to verify theoretical implementation.

2. Software development of database and optimization programs to allow the simulation program to be applied to an actual surface mount facility.

9 SOFTWARE DEVELOPMENT

9.1 Introduction

During the initial stages of the practical implementation of this dissertation, the SMT line at STC was not yet completely installed . Therefore a software simulation model was developed to verify the theoretical proposals. During the period of the software development, the SMT line was commissioned and test runs started. It was decided to use the simulation model to improve the utilization of some of the machines on the line, in particular the Pick and Place machine. This, however, called for an additional optimization and database software program to be written to allow realistic results to be obtained.

This section discusses the development of both the simulation and optimization programs.

9.2 Overall Problem Statement

The SMT plant at the STC premises is shown in Figure 9.1. The line consists of the following workstations:-

- 1 Silkscreen machine
- 2 Gravity Platforms
- 2 Auto Magazine Unloaders/Loaders
- 1 Pick n Place machine consisting of 2 placement Heads and 2 Extended Input Modules
- 1 In-Line Reflow Oven
- 2 Inspection Positions
- 3 Conveyor Belts
- 1 Repair and Test bay

The simulation model was developed to enable management to answer the following questions about the SMT facility in order to meet production objectives:

1) What is the expected system throughput?

2) How many of the machines, conveyors, operators and queues are needed to keep the facility producing at the required rate?

3) How will random equipment failures affect the facility?

4) What is the impact of the variation in process yields in the facility?

5) How will the introduction of another product into the production schedule affect the facility?

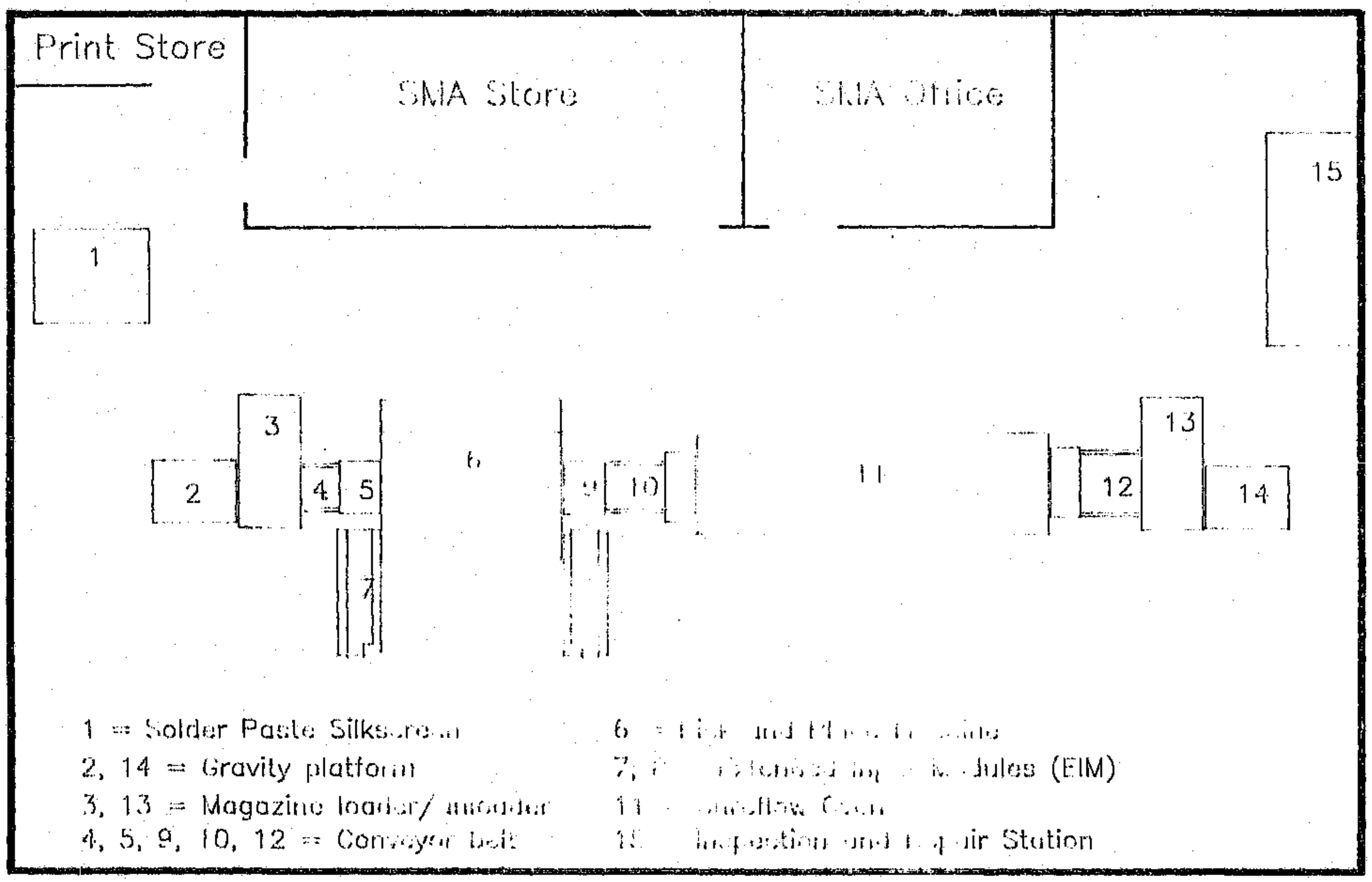


Figure 9.1 Surface Mount Facility Layout

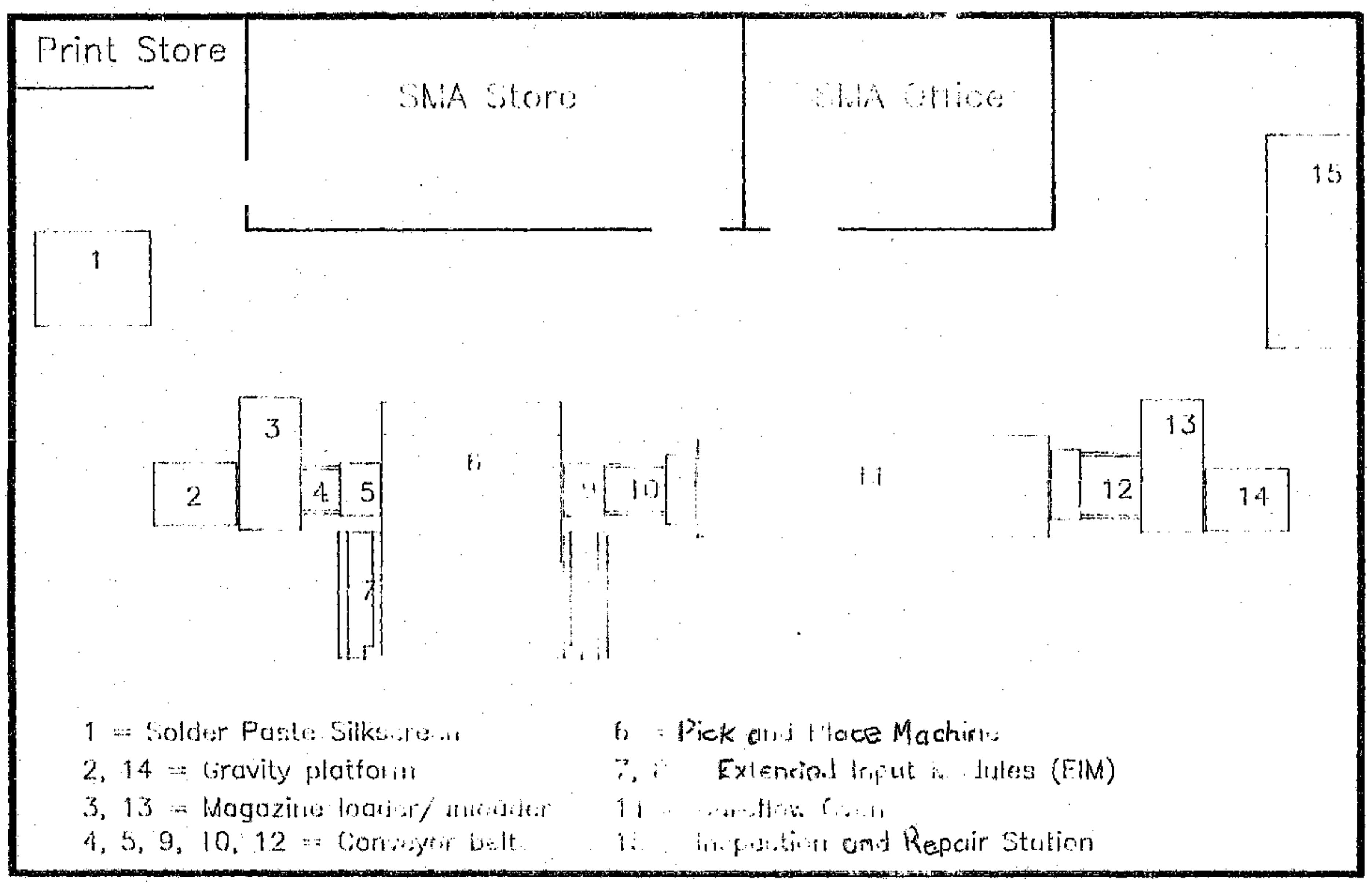


Figure 9.1 Surface Mount Facility Layout

A major problem which is highlighted by the use of the simulation program, is that as long as the two heads of the placement machine are not working equal times(i.e. they are not placing the same number of components), stoppages are caused and boards begin to queue at the input of the machine. Thus, the optimization of component feeder placement around the two heads of the placement machine also requires attention.

9.3 Problem Solution

To provide management with the information it requires, a simulation model was developed incorporating real-time graphical animation and statistical analysis using the Simscript II simulation language.

A database was created to optimize the Pick n Place component feeder placement. Using a convenient graphical and menu-driven technique factory layout information and production parameters can be entered into the database. Two completely different methods are used to determine the best result and the exact feeder placement positions for each component.

Once the user has accepted the optimization results, which can be modified manually, the database is linked to the simulation program to allow for an exchange of relevant data.

Based on the information retrieved from the simulation run, the user has the option to modify one or more of the initial parameters and rerun the model. In this way, various scenarios can be tested before the actual implementation of the SMT production line.

A graphical representation of the interaction between the user and the software modules is shown in **Figure 9.2**.

The details of both the optimization and simulation programs are presented in the following sections.

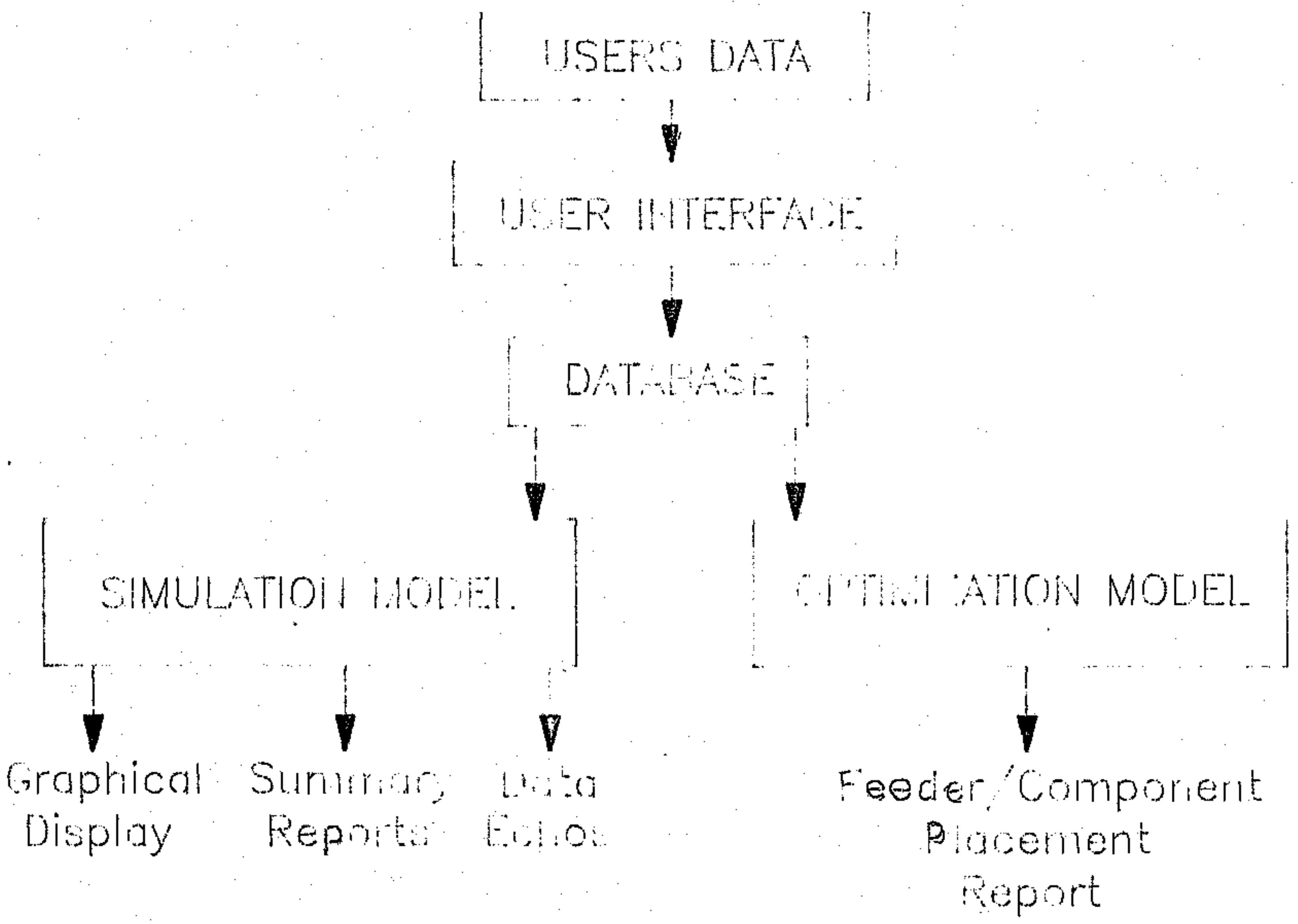


Figure 9.2 Software Module Interaction

10 THE SIMULATION PROGRAM

10.1 Why Simulation?

Simulation is the process of developing a mathematical model that describes the functions of a real-world situation called a system. By collecting data on the system's response under various conditions, it is possible to learn how the real system may perform without having to attempt costly experiments with the actual equipment.[2]

Within an operating facility, management must react to a rapidly changing environment to meet production objectives. Decisions on work order release, scheduling and staffing must be made in light of new orders, equipment availability, absenteeism and other factors. Simulation has significantly improved the predictability, and therefore reduced the risk of control and scheduling decisions. In the area of production systems, simulation can be used to identify design flaws and operating problems, verify proposed layouts and permit management to examine how well a proposed design will meet the production objectives.

The main advantage of simulation is that it permits controlled experimentation on a system with no disruption of the actual process. This is ideally suited to a flexible manufacturing environment (FME) in that insights into the functioning of such a complex system may be made before any hardware or software is specified.

However, the results of simulation studies are often presented only as pages of values which fail to communicate the understanding gained. Premature action may be taken based on invalid assumptions, or hidden modeling errors. Conversely valid simulation results may not be quickly appreciated.

The best way to represent a dynamic system is often graphically. Animated graphics show clearly the operation of the simulated system, and graphic results are easily evaluated; system operation is better understood, and decision makers have more confidence in the simulation results.

A simulation project incorporates the following activities[3]:

Project formulation and software selection
(Approx 25 - 30%)

Data and information collection
(Approx 20 - 30%)

Statistical modeling of system randomness such as machine breakdowns, validation of the model, and the statistical design and analysis of the simulation runs [1]
(Approx 10 - 15%)

"Coding" or writing the software and graphical interactions.
(Approx 30 - 40%)

10.2 Selection of Simulation Software

The selection of software was limited by cost and it was decided to use a package call Simscript II which was already in use at the University. The package provided good modeling flexibility, contained a wide variety of statistical capabilities and also allowed for graphical animation to be interfaced with the model. However, ease of model development, output report generation and fast model execution were lacking.

10.3 The Simulation Model

10.3.1 Simulation terminology

The following terminology is fundamental to an understanding of simulation and its capabilities.

A **SYSTEM** is the collection of all the elements of a process or area to be analysed. The components of a system are divided into two categories:

PERMANENT items - the physical components always present in the system.

Transient **ENTITIES** - the items that flow through or are being processed by the system.

Many of the permanent items can be considered to be limited **RESOURCES** that are necessary to perform a step in the process. Resources can be described by a set of physical conditions called **STATES**. A machine or worker can be in a **BUSY** state (working on a part), an **IDLE** state (waiting for work) or possibly **BLOCKED** because of a breakdown.

Other permanent parts of a system can also be described by a series of states. A warehouse or buffer area for stock can be EMPTY, FULL or have some intermediatory number of parts stored.

A simulation model records state changes by using a series of variables, sometimes referred to as **STATE VARIABLES**, that are constantly updated over the simulation time.

In general, we define a **PROCESS** to be a time-ordered sequence of interrelated events separated by passages of time, which describes the entire experience of an entity as it flows through a system.

Types of processes: By determining how often the state of the system changes, it can be classified as either **DISCRETE** or **CONTINUOUS**. Discrete systems only change states at specific, countable moments in time. Each time a discrete system has a state change, it is called an **EVENT**. Such a model is called a Discrete Event Simulation Model and is the type which we will be dealing with.

There are two types of animations; post-processed and real-time animations. A post-processed animation takes the output of a simulation run, and graphically displays what happened

The SMT simulation model uses Simscripts real-time, interactive animation system, which allows the viewing of the animation **WHILE** the simulation is running.

10.3.2 Software development

The simulation model was developed by dividing the model building process into two sections or frameworks.

1] **Model Frame** - used to describe how the entities flow through the system. This includes:
  - processing steps
  - decision/control logic
  - machine failures
  - material handling

2] **Experiment Frame** - contains the parameters for a specific condition or scenario of the system being modelled.

  - number of machines
  - specific processing times
  - part routings
  - statistical distributions
  - output information to collect

By separating this information, the user can test the same system under various experimental conditions, without changing the model frame.

10.4 The Simscript Program

The Surface Mount facility that is being simulated is shown in **Figure** 9.1. There are in total 11 resources (or in this case workstations and conveyors) which have some form of impact on the total throughput of the line. Thus all of these are treated with the same priority, for example a conveyor belt which is moving too slowly will have a similar effect on the throughput as a magazine loader which is loading too slowly etc.

The input data requirements for the simulation are definable and fall into each of four areas, namely:

- Workstation characteristics
- Materials handling characteristics
- Job (parts) characteristics
- Production characteristics

Each <u>workstation resource</u> has the following characteristics attributed to it:-

* the types and numbers of each machine
* the types and numbers of the machine queues
* the queue serving protocol
* transfer times between machines and storage points
* failure consequences for each machine and the system
* tool storage at each machine
* tool sharing possibilities
* tool handling and set-up times

The <u>materials handling</u> resources have the following attributes:-

* type of system (conveyors)
* dimensional layout of machines, load stations, queues and buffers
* speed, direction and stopping points for conveyors
* failure rates and repair times for materials handling systems

<u>Jobs</u> going through the system are represented by permanent entities with the following attributes:

* complete manufacturing sequences for each job
* all feasible workstations for each operation on each job
* the working or task time for each operation
* machining and tool changing times for each feasible process

- required tools
- all feasible machining sequences
- type and number of fixtures for each set-up
- times for set-up
- frequency, duration and location of part inspections

Thus each type of job has a unique routing sequence which determines which workstations it requires for completion and how long it will require those specific workstations for.

The overall system or production characteristics include:-

- the number of shifts
- scheduled maintenance
- part batches and number per batch
- manpower requirements
- production scheduling

10.4.1 The program structure

This section gives a brief decription of the subprograms which comprise the Simscript simulation program.

The program begins with a **PREAMBLE**, whose statements are declarative in nature. (i.e., the preamble contains no executable statements). The preamble is used to define all the building blocks for the model, such as processes, resources and other structures. The preamble is also used to define the global variables, the basic unit of time for the simulation clock and the desired measures of system performance (e.g., statistical variables).

The **MAIN** program is where the execution of the program takes place. The simulation actually begins with the execution of the **START SIMULATION** statement.

There is a process routine for each process defined in the preamble; the routine and corresponding process having exactly the same name. The routine describes the entire experience of a process entity as it moves through its corresponding process.

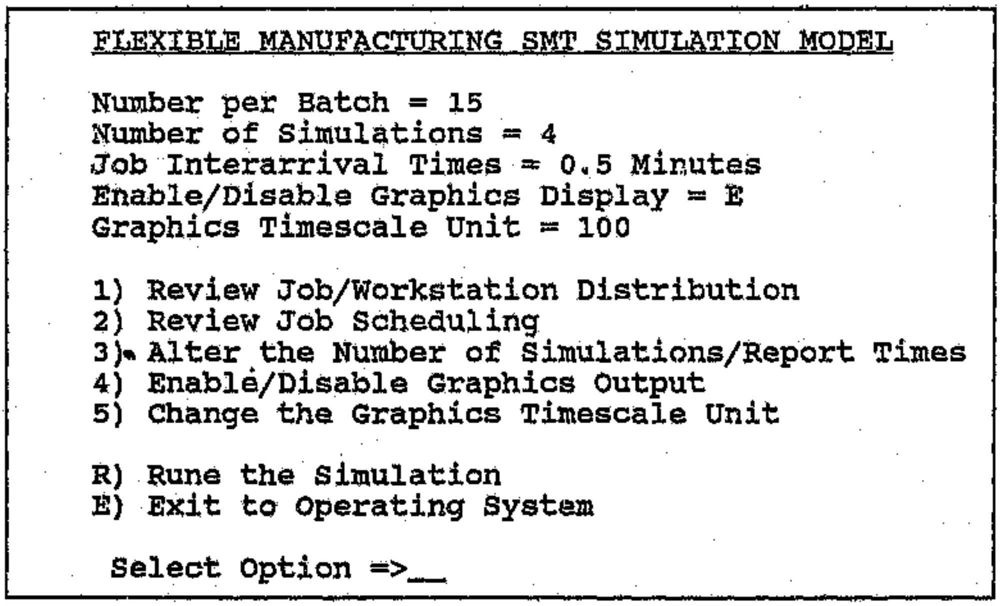
Both static and dynamic graphical icons have been created. These are simply screen images of program entities, for example, a job moving from process to process. Changes in their position or appearance correspond to changes in the system being simulated.

Simscript has no inherent measurement units. Spaces are thus defined in a model-orientated manner (in real-world coordinate). These are transformed into coordinates suitable for graphics display through viewing transformations.

Each graphical entity displayed on the screen has its own graphics routine which is activated every time the entity changes state.

The program begins by reading the system parameters from the database into its own state variables. All permanent and temporary entities are then created and initialized. The graphic drivers and their corresponding display routines are initialized and the main menu displayed (See Figure 10.1). The user then has the option to alter any of the experimental frame parameters.

Once the **RUN** option has been selected, the graphics for the simulation are displayed on the screen. The user also has the option to disable the graphics drivers and then only the simulation time and important status information will be displayed on the screen.



```
FLEXIBLE MANUFACTURING SMT SIMULATION MODEL

Number per Batch = 15
Number of Simulations = 4
Job Interarrival Times = 0.5 Minutes
Enable/Disable Graphics Display = E
Graphics Timescale Unit = 100

1) Review Job/Workstation Distribution
2) Review Job Scheduling
3) Alter the Number of Simulations/Report Times
4) Enable/Disable Graphics Output
5) Change the Graphics Timescale Unit

R) Rune the Simulation
E) Exit to Operating System

 Select Option =>__
```

Figure 10.1 Simulation Model main menu.

The simulation begins by initiating all the resources (machines) to the idle state and activates the first batch of jobs, each job being seperated by a uniformly distributed inter-arrival time. As mentioned before, each job type has a unique routing sequence associated to it, and it now goes through that sequence in the following way:-

A **REQUEST** is made to the first workstation in its production schedule. If the workstation has any free resources (i.e., if any machines at that workstation are idle), then the request is accepted and the job is **WORKED** on for the required amount of time. On completion of the work time, the resource is **RELINQUISHED** and the job requests the following workstation according to the schedule. If on request, the workstation does not have any idle resources, the job is put into the workstation **QUEUE** and waits until one of the resources is relinquished.

An important point to note is that the Simscript software is a multitasking environment. Therefore any number of jobs can be worked on at the same time independent of all others. Also, throughout the simulation, the statistical variables are being recalculated and updated.

Once the batch of jobs is completed, a final report detailing all aspects of the job and the workstations is written to a file, a copy of which is shown in **Figure 10.2.**

The user should never base decisions on the results of only one run, since being a stochastic simulation, there is no reliable way of determining the accuracy of the results. Therefore, a report is generated after each run to allow for a comparison of the results.

A listing of the program source code and a diskette containing all the files is given in **Appendix 13.**

```
                SIMULATION RESULTS - FIRST RUN
                ------------------------------

Total running Time =    1.89 Hours

Simulation Run Time =    1.89 Hours

Job Report:
----------

            Average Total
Job Type    Delay in Queue    Number Completed
--------    --------------    ----------------


    1            .37                 10


Station Report:
--------------
             Average Num        Average Delay    Prop Time    Prop Time    Prop Ti
Station      in queue           in Queue         Working      Idle         Blocked
-------      ----------         -------------    ---------    ---------    -------

   1             0.                  .000           .001         .999        -.00

   2             0.                 -.000           .006         .994        -.00

   3             0.                  .000           .009         .991        -.00

   4             0.                  .000           .010         .990        -.00

   5             1.611               .304           .626         .374        0.

   6              .331               .062           .700         .300        0.

   7             0.                 -.000           .010         .990        -.00

   8             0.                 -.000           .010         .990         .00

   9             0.                  .000           .010         .990        -.00

  10             0.                 -.000           .009         .991         .00
```

Figure 10.2 Simulation output report after first run is completed.

10.5 Conclusions

The simulation program is capable of simulating any resource i.e. machine or conveyor belt, in the surface mount facility. Machines can be simulated and results obtained detailing utilization time, queue time for jobs waiting to use the machine and breakdown or setup stoppages. A graphical interface was developed to allow events to be viewed in real-time, on the screen. The flow of a batch of jobs through the system can be traced graphically and graphs showing jobs queueing at machines, displayed on request by the user. The users interaction with the program is all through a mouse-driven menu system, allowing for simple yet powerful functionality.

Although the simulation program was initially developed to verify the theoretical design of the facility before any specific machinery was decided on, by the time it was completed, the machinery has already been purchased and the facility was being commissioned. The simulation was then applied to the existing facility to calculate possible throughput etc, and it was found that the Pick n Place machine had a overiding effect on the balance of the SMT line. The Pick n Place machine was, in fact, a major bottleneck.

Even though the next part this project falls out of the scope of the dissertation, it was included because sufficient analysis of the facility could not be achieved using the simulation program without rectifying the main bottleneck of the facility, namely the Pick n Place machine.

11 THE DATABASE AND OPTIMIZATION PROGRAMS

11.1 Introduction

Once the simulation program had been completed and tested on the SMT line, results proved that if the Pick n Place machine was not set up correctly, it caused throughput on the line to decrease dramatically. However to understand the complexity of setting up this machine, a closer look at its workings is required.

11.2 The Universal Omniplace 4621B System

The Universal Pick n Place machine is shown in **Figure 11.1**. The machine has two identical placement heads which pick components from the feeders using a nozzle and place them on the PCB. The placement head requires different nozzles to pick different components and the machine has a facility whereby up to eight nozzles may be utilized by both heads independently.

The front feeder positions are called base feeders and consist of 20 feeder positions each. The feeders on the two Extended Input Modules are called EIM feeders and there are 43 positions on each. The placement heads can only pick components from the base feeders.

If a component is on the EIM feeders, then a shuttle with two heads only picks the component from the feeder and travels with it to the edge of the EIM. The placement head then picks it off the shuttle and places it on the board.

Note that the shuttle only has the choice of two nozzles during machine operation. Therefore, the type of components placed on the EIM feeders are limited by the type of nozzles used by the shuttle.

An important factor to consider at this stage is that of placement speed. A component picked and placed from a base feeder is quicker than one from an EIM feeder (on average, about 0.40 seconds per component). It was initially assumed that a component picked from the first EIM position (that position nearest the placement head) would be substantially faster than one picked from the last position. This was later disproved after exhaustive tests were carried out on the machine.(Refer to **Figure 11.2a** and b). But we see that it would be preferable to have as many as possible of the components on base positions.

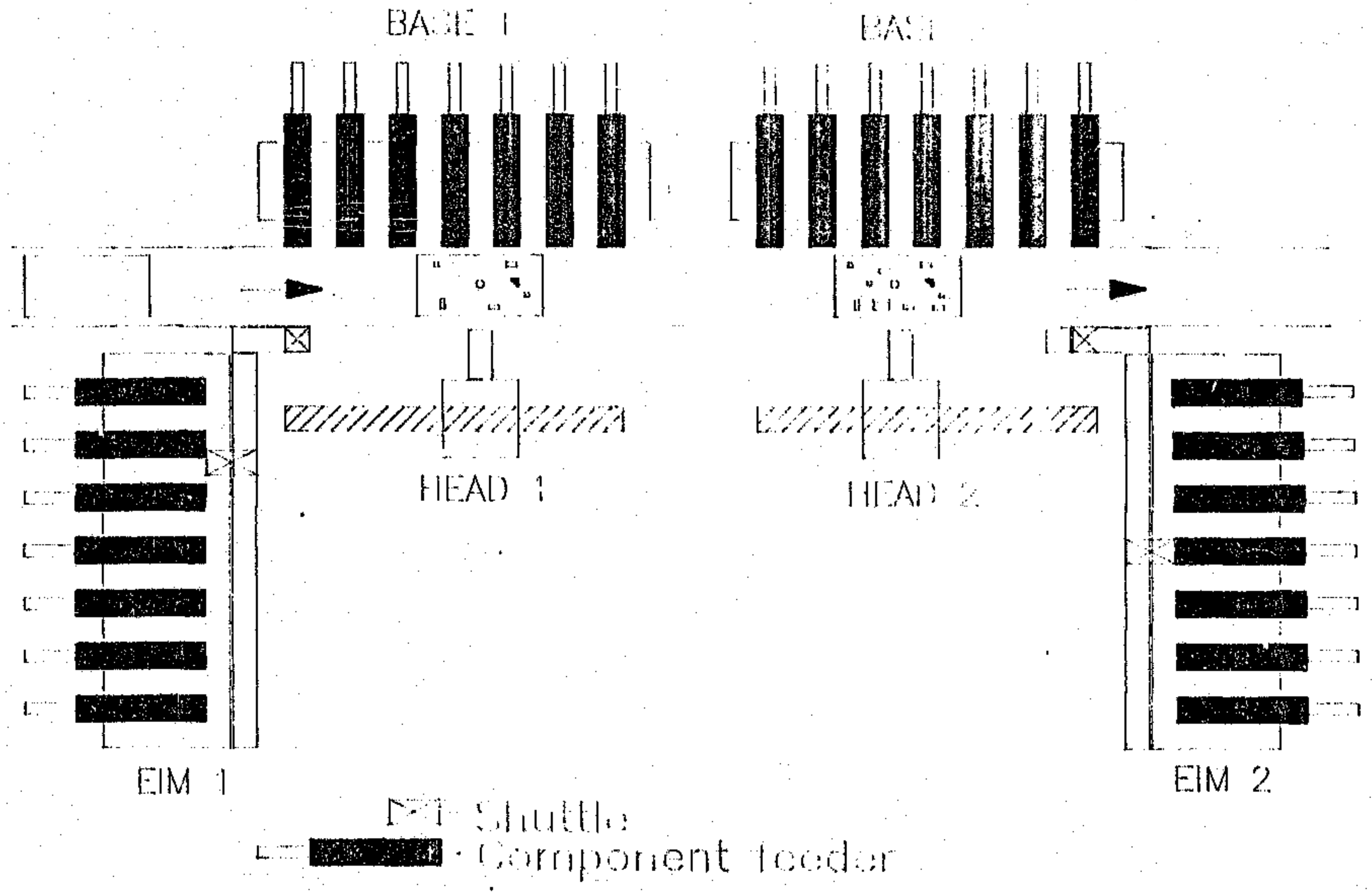


Figure 11.1 Universal Pick and Place Machine

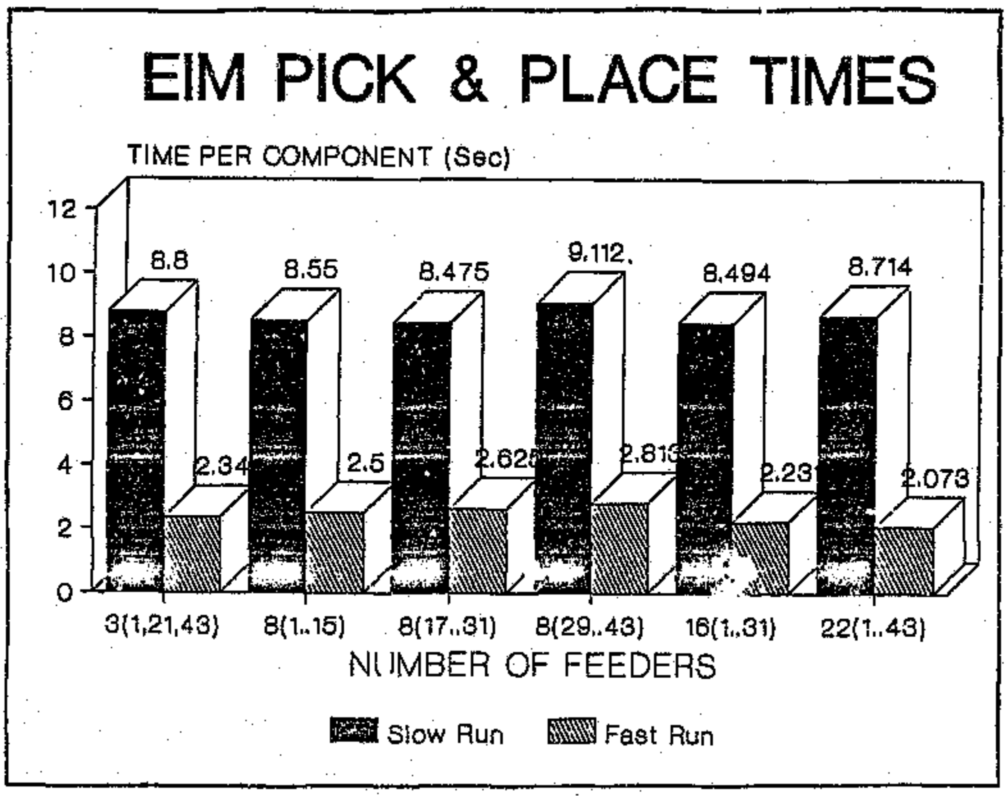


Figure 11.2a EIM pick & place times. Feeder placements are given in brackets.

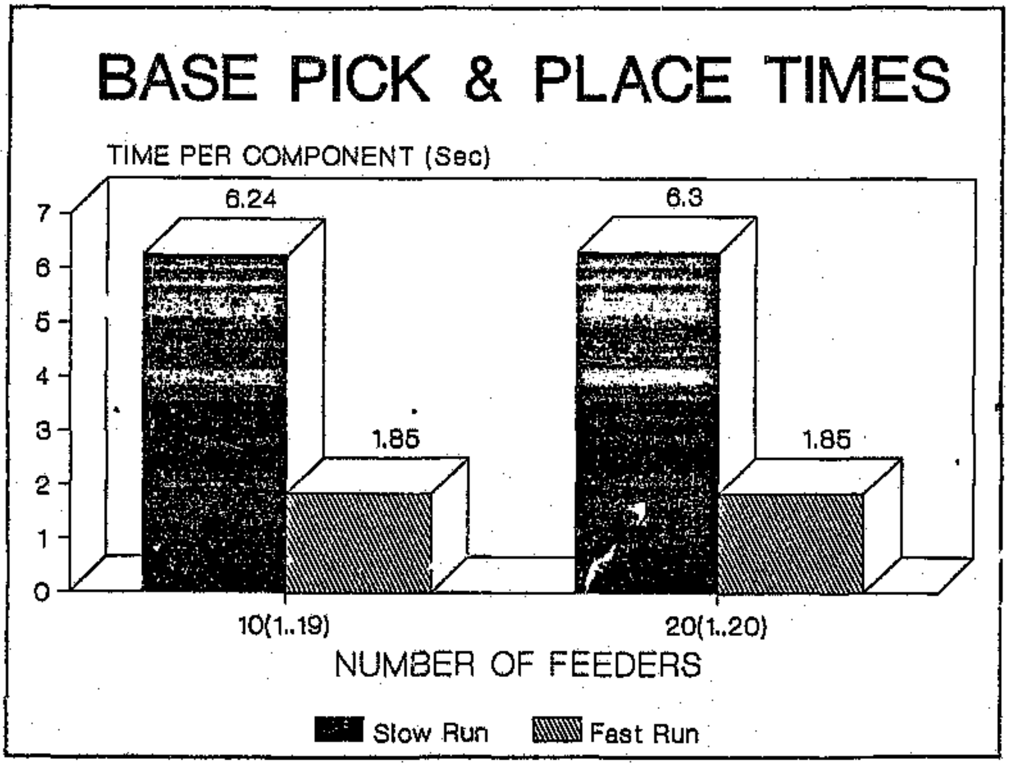


Figure 11.2a BASE pick & place times. Feeder placements are given in brackets.

Another important factor to be considered is verification. This is the identification of a component to ensure that it is the correct element being placed. Certain components, for example resistors and capacitors, are verifiable and it would obviously be desirable to have as many components verified as possible. There are two possible methods of verification.

The first is to install a special verifier at the side of the machine. The placement head would then pick a component and would have to take it to get verified before placing it. This is obviously undesirable, since the extra time needed to verify every component would add a big overhead to the normal working time.

The second method is to place a verifier in the EIM shuttle which could then verify the component "on-the-fly" or while it is taking it to the placement head. This method is obviously more desirable but the selection of components that can be placed on the EIM feeders is going to be further restricted to only verifiable ones.

A further complication to the feeder placement problem is that placement head 1 can be used to apply the adhesive to a board as well as place components. Thus, if we have a board which requires components to be placed on the solder (or bottom) side of the board, adhesive first has to be applied to the board before components can be placed. If the adhesive application can be done within sufficient time, then head 1 could also be expected to place some of the components. Tests were conducted to calculate adhesive application times and the results are given in Appendix 14.

Because both placement heads can be used to place components, the optimum situation would be for both heads to work equal times thus causing no delays. However, if for example head 2 works longer than head 1, the completed board at head 1 cannot be transferred to head 2 until it has finished it's present board and transferred that to the next station.

The simulation therefore became pointless if the Pick n Place machine was not set up correctly and it was decided to develop an optimization program to solve this problem.

11.3 Software Development

It is very important for the user to have quick and easy access to both the simulation program parameters and the optimization program results, to allow modifications to be made and the programs to be rerun.

Therefore a database was designed that caters for both programs in one environment. It consists of modules which can be modified independently. The database is an integrated program which automatically updates changes made in one module in all other related modules.

The optimization program was also written within this database environment because the nature of the database's storing and memory methods facilitated the neccesary computations.

11.3.1 The database program

Using the database program, the user may modify or edit an exisiting database or save the present database to a file and start a new one.

In order to edit an exsisting database, the following funtions are available to the user:
(See Figure 11.3)

- Initialization
- Workstations
- Job Setup and Optimization
- Run Times and Reports
- Quit

Initialization

This module enables the user to change the experimental frame parameters (See 10.3.2) which include:

- Total number of workstations
- Job inter-arrival time
- Number of jobs
- Number per batch
- Number of tasks
- Total number of simulations
- Number of components on the Pick n Place machine

On most occasions the user's changes will be automatically updated in other modules. However, if, for example, the user changes the number of components on the Pick n Place machine, the components module would be updated with the new number of components but the user is still required to enter the relevant details regarding these new components.

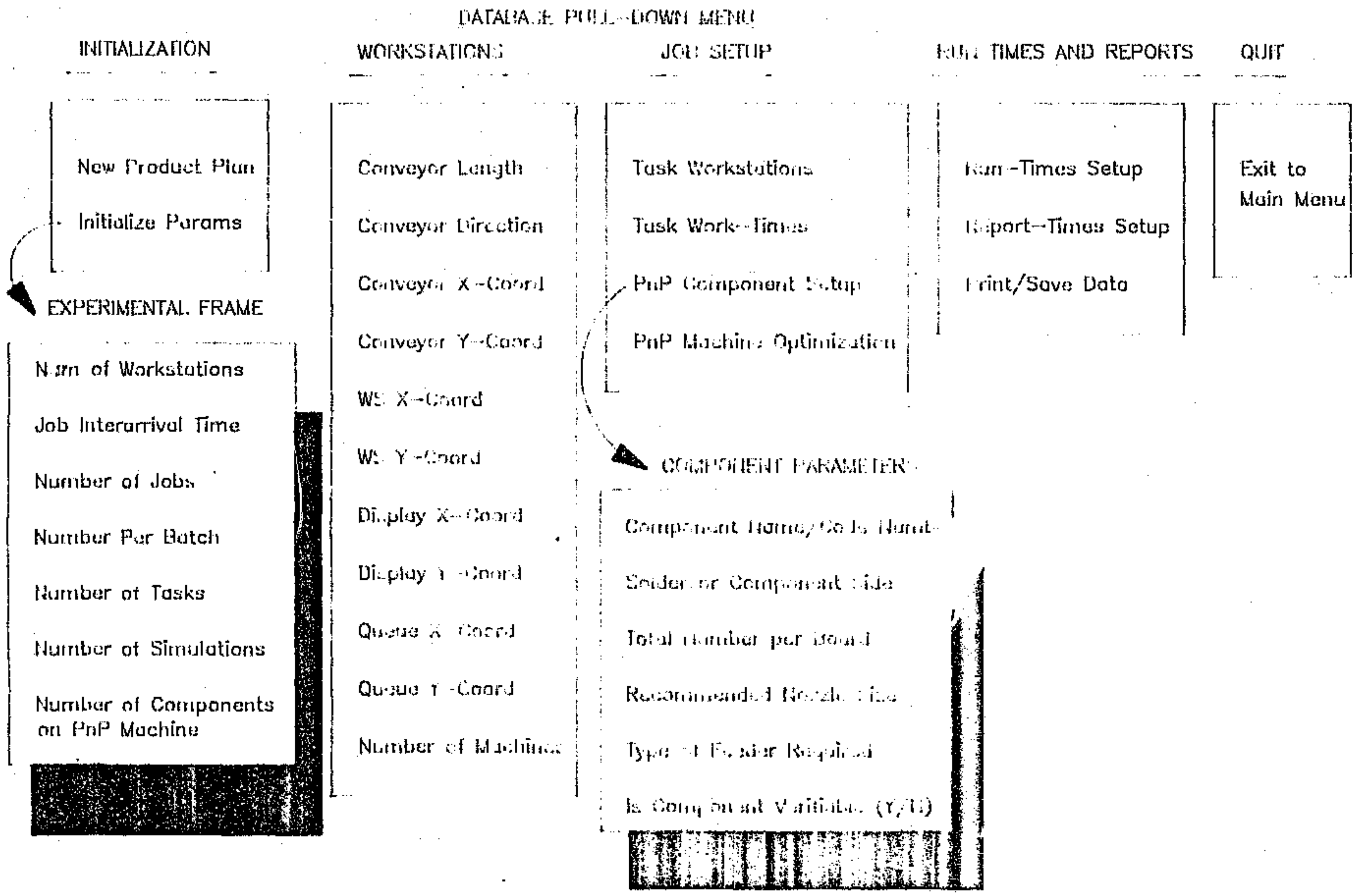


Figure 11.3 Database Pull-Down Menus

Workstations

In this module the user inputs all relevant details concerning the layout of the plant. Some of the choices include:-

- Conveyor length
- Conveyor direction
- Workstation X and Y coordinates
- The number of machines at each workstation

Job Setup and Optimization

The first two options in this module are:-

- Task workstations
- Task work times

Here the user defines the job routing schedule and working times at each workstation.

The next two options are:-

- PnP component setup
- PnP machine optimization

Both these options are discussed in detail in Section 11.3.2. However we see here how the optimization program is considered a module of the database and can therefore access all the relevant information directly from the database.

Run Times and Reports

The information in this module is purely for simulation purposes and allows the user to define starting times for the simulation runs (up to 6 runs are allowed). Apart from the reports that are automatically generated at the end of each simulation run, the user can define times when he would like additional reports to be generated. This kind of report would give the details of all previous simulations run up until that point in time.

Quit

This option closes all the modified files, releases all activated windows and takes the user back to the main database menu.

11.3.2 The optimization program

The optimization program uses two methods of optimization. The program can either be based on spreading the load between the two placement heads according to the number of components per board or by grouping similar nozzles together and then optimizing their feeder placement positions. The method used to produce the optimum result will depend on the type of component configuration for the PC board.

Each type of component that is used on the Pick n Place machine is entered into the component database with the following attributes:-

Component Type X
NAME OR CODE NUMBER
NUMBER PER BOARD
TOPSIDE/BOTTOMSIDE
NOZZLE SIZE REQUIRED
FEEDER TYPE REQUIRED
VERIFIABLE (Y/N)

If, for example, we have eight components as follows:-

Comp	Num Per Bd	Nozzle Size	Verifiable
1	85	0.0420	N
2	45	0.2080	Y
3	35	0.0580	N
4	80	0.0420	N
5	68	0.1090	Y
6	60	0.1090	Y
7	45	0.0580	N
8	69	0.2080	Y

Table 1 - Component database example 1.

The first method sorts the component database on the NOZZLE SIZE field, i.e. all components with the same nozzle size are grouped together. The program then optimizes by first checking if the component is verifiable. If it is, then it places it on an EIM in such a way so as to keep the two EIM's balanced. If both EIM's have their full quota of components (i.e., two different types of nozzles each), or if the component is not verifiable, it is then placed on a base position in the same way.

The second method sorts the component database on the NUMBER PER BOARD field i.e, sorts the components in a descending order of components per board. The program then optimizes by sequentially placing the components around the machine based on whether they are verifiable or not. This method does have its drawbacks, however. Although it usually gives the better optimization results than the first, it does this at the expense of placing some of the verifiable components on the base feeders thus assuming that optimization is more important than verfication. It would be up to the user to decided if those components on the base could do without verification.

The reason why two different methods are necessary, is because of the fact that we have so many parameters with which to work including component count, verification, EIM nozzle limitation etc. In some cases the first method works better than the second and visa versa. This point is clarified by the following example.

If the configuration in Table 1 is fed into the computer and optimized the following results would be obtained:-

Method 1

	Head1	Head2
Total No. of Component Types on Base	2	2
Total No. of Component Types on EIM	2	2

Optimization Percentage* = 82.8%

Method 2

	Head1	Head2
Total No. of Component Types on Base	2	2
Total No. of Component Types on EIM	2	2

Optimization Percentage* = 99.7%

* optimization percentage = $\frac{\text{total working time head x}}{\text{total working time head y}}$

We see that although they both place an even number of components around the machine, the second configuration is definitely the more desirable.

However, if we change the component configuration around as in Table 2 and make all the components verifiable, we get the following results:

Comp	Num Per Bd	Nozzle Size	Verifiable
1	85	0.0420	Y
2	45	0.2080	Y
3	35	0.0580	Y
4	80	0.0420	Y
5	68	0.1090	Y
6	60	0.1090	Y
7	45	0.0580	Y
8	69	0.2080	Y

Table 2 Component database example 2.

Method 1

	Head1	Head2
Total No. of Component Types on Base	1	1
Total No. of Component Types on EIM	2	4

Optimization Percentage* = 91.3%

Method 2

	Head1	Head2
Total No. of Component Types on Base	1	1
Total No. of Component Types on EIM	4	2

Optimization Percentage* = 43.6%

* optimization percentage = $\frac{\text{total working time head x}}{\text{total working time head y}}$

The above results justify the need for both methods.

If any of the above two methods do not produce results over and above 95% optimization AND if they have placed some components on the base feeders, then an additional optimization program called REOPT has been written. This program takes the results of the initial optimization program and attempts to re-optimize only the base placement feeders, thereby increasing the overall optimization percentage. If, for example, Head 1's total placement time is larger than Head 2's total placement time, the program attempts to transfer one or more of the component feeders from Base 1 to Base 2 until the placement time difference is at it's minimum. However, this optimization is only possible if Head1 has components on its Base feeders.

After any of the optimization routines have been run, the user has the option to print out the results of the last run to allow for a comparison of the results at a later stage. The report also gives the exact feeder placements for each component neede by the user to set up the machine. An example of an optimization report is shown in Figure 11.4.

```
DESCRIPTION              [illegible]
----------------------------------------------------------------------------------------



SOT 23                       10     1          1          B       0.0420    0.31
CHIP CAPS 0805..1505         10     1          3          B       0.0420    0.31
CHIP CAPS 0805..1505         10     1          2          B       0.0420    0.31
CHIP CAPS 1206..1505         10     2          2          B       0.0530    0.31


                                         HEAD1        HEAD2
                                         -----        -----

 TOTAL NO. OF COMPONENTS ON BASE =         3            1
 TOTAL NO. OF COMPONENTS ON EIM =          0            0

 TOTAL NUMBER OF COMPONENTS =  40

 TOTAL PLACEMENT TIME FOR HEAD1 =   0.31     Minutes

 TOTAL PLACEMENT TIME FOR HEAD2 =   0.31     Minute

 OPTIMIZATION PERCENTAGE =  33.3    %
```

Figure 11.4 Output report from optimization program detailing component and feeder placement around the Pick and Place machine.

11.4 Conclusions

The optimization program not only overcomes many of the problems highlighted by the simulation program, but is to this day aiding the machine operator tremendously in setting up the machine. To manually calculate component feeder placement for the Pick n Place machine previously incurred enormous human error which was only identifiable once the machine was set-up and running. At present, the user enters all the component information into a database which, in turn, generates a detailed feeder placement report.

The integration of the database and simulation programs, allows the user to make small changes to the system parameters and view the results almost simultaneously.

12 CONCLUSIONS AND RECOMMENDATIONS

12.1 Summary and Conclusions

A surface mount facility should be capable of flexibility with respect to component variation, board size, soldering processes etc, but without a complete set of standards, this is not possible. However, to wait for a unified set of standards could be just as disasterous from a market-entry point of view. SMT is new and undeveloped and the learning curve is therefore that much greater. The risks are therefore a lot greater, but companies can reap large rewards for persistance.

Due to the high degree of flexibility required for such a facility, and the number of options available to achieve this, no specific theoretical methodology or equipment was recommended throughout this project. However, a detailed description of component availibility, reliability and availability of standards is presented. Also. a detailed description of each process in the SMT facility is also presented. In this way, the reader has been given all the possible options that are available, and can make his choice depending on what type of facility he requires.

A simulation program was developed to simulate the viability of specific macninery working as a complete production line. Results obtained could determine the individual effect a machine was having on the line as a whole. This machine could then either be replaced or modified until a suitable result was obtained.

The simulation program succesfully identified the major bottleneck in the SMT facilitiy, namely the Pick n Place machine. Together with the optimization software, an integrated system was developed which allows for real-time simulation of the facility and allows the user to try out different variations without actually incurring any wastage of materials and/or production time.

12.2 Recommendations for future work

On many occasions, a Materials Requirement Package (MRP) having been setup up at a factory, has failed to produce the "promised" results. The reason for this, is that results could only be verified once the system had been installed and was operational for some months, thus causing finacial and inventory losses.

What is needed is the incorporation of the simulation program with an (MRP) program which would provide a foresight into the viability setting up the operation in the actual factory.

The simulation program would be capable of simulating the entire plant and would provide the production manager with the ability to not only have a forward scheduling ability but also to verify that it its a viable one i.e. the simulation program would take into account breakdowns, stoppages, work in progress, finite queue sizes etc. If, according to the simulation program, the production schedule could not meet it's requirements, the MRP program could be easily modified and the simulation rerun, or an alternative method chosen.

Because of its inherent flexibility, the simulation model could easily be adapted from one facility to another within a short time span.

The justification for this proposal is simply "Costs". Thousands of Rands could be saved by analysing the situation before actually implementing it.

Appendix 1. The Manufacturing and Reliability Requirements of Surface Mount Components and Packages

A1.1 Passive Components

A passive component is one which does not contain a source of energy. The chief types of passive components in electronics are capacitors, resistors and inductors.

The common nomenclature defining capacitor and resistor sizes employs a four-digit code. The first two digits describe the component length (in hundredths of an inch), and the last two refer to its width:

12 06

0,12 in nominal length — 12 — 06 — 0,06 in nominal width

A1.2 Capacitors

The two predominant classes of surface mount capacitors are the ceramic and the tantalum dielectric families. A range of multilayer ceramic capacitor (MLC) chips account for the vast bulk of surface mount capacitors, however the capacitance values available are more limited. By and large MLCs now meet present-day needs without compromise to high-density or high-tolerance requirements.

A1.2.1 Multilayer Ceramic capacitors

Ceramic capacitors were one of the first surface mount components to be widely used. As a result, a high degree of standardization has been achieved worldwide.

Physical characteristics

The internal construction of a MLC is shown in **Figure A1.1.** It consists of alternating layers of dielectric and electrode materials printed consecutively and co-fired at a temperature of 1000-4000°C. The dielectric is usually a barium titanate composite, and the electrodes are platinum-silver films. Alternate electrodes are connected to opposite end terminations to form a set of parallel-plate capacitors. To prevent dissolution during soldering, a thin barrier layer is applied over it. The barrier consists of a metal, such as nickel or copper, that has a low solubility in tin-lead solder. Finally, a tin coating is applied over the the barrier metal to provide a highly solderable surface.

Electrical characteristics

Electrical performance characteristics of ceramic capacitors depend on the nature of the dielectric material employed. Materials are usually classified according to the definition in EIA-198.

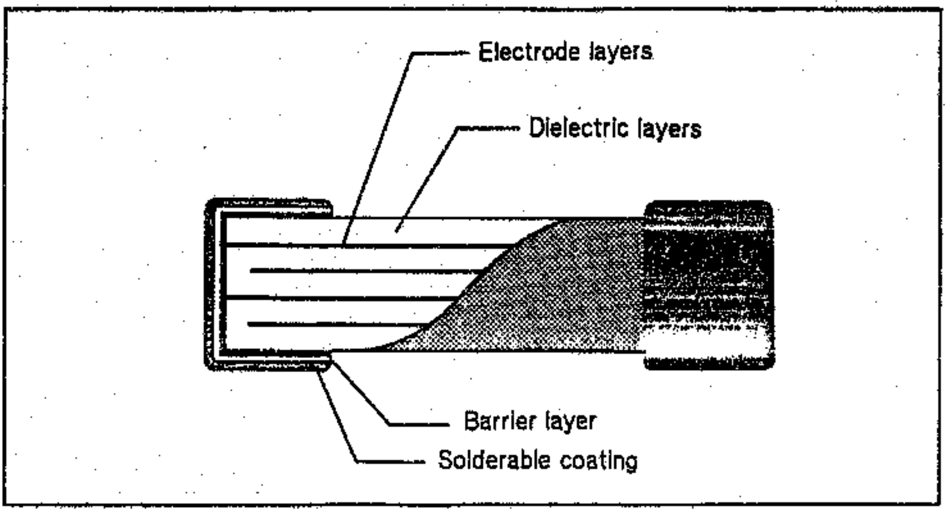


Figure A1.1 Internal construction of ceramic capacitor.

The three most common dielectric classes are COG (formerly NPO), X7R, and Z5U(3). The COG dielectric offers the highest degree of temperature stability due to its low dielectric constant. The X7R dielectric has a higher dielectric constant and is preferred for many general purpose applications.

The Z5U is used in applications requiring high capacitance and small physical volume due to its high dielectric constant. The approximate physical sizes of COG, X7R, and Z5U capacitors are shown in **Figure A1.2.**

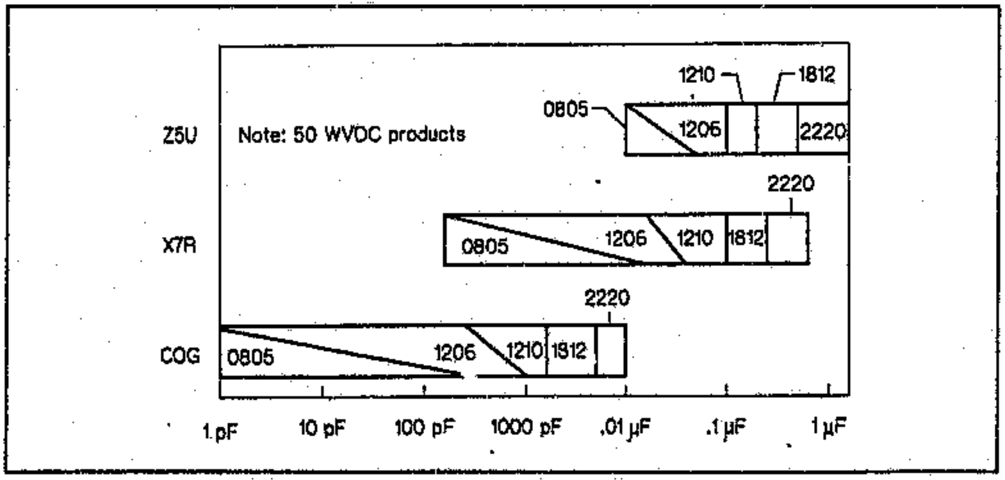


Figure A1.2 Approximate capacitor ranges for various ceramic capacitor packages.

A1.1.2 Tantalum electrolytic capacitors

The demand for higher capacitance values is usually met with tantalum electrolytic capacitors. Because of their small physical sizes, these capacitors are primarily suited for small-signal and low-voltage applications.

Physical characteristics

The post-molded plastic body construction shown in **Figure A1.3** has gained general acceptance, and because of its repeatable body dimensions and flat top surface, this package is ideally suited for automatic assembly. Its non-hermetic construction makes it an attractive low-cost solution for commercial and general industrial applications, but less suitable for high reliability use.

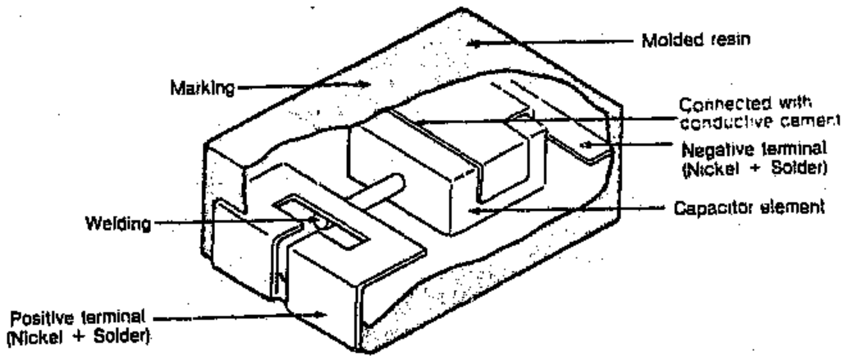


Figure A1.3 Construction of post-molded tantalum capacitor.

Important features in this design are: a) solid electrolyte capacitor element; b) post molded resin package body; and c) external leads soldered or welded to the capacitor element. The leads fold under the capacitor body to provide a measure of compliance without significantly increasing the overall package dimensions.

Electrical characteristics

The relationship between size and capacitance value for various voltage ratings is given in Table 1. As with most capacitors, best reliability is obtained when the applied voltage is considerably less than the maximum rating.

Some SMT manufacturers recommend that the applied voltage not exceed 50-60% of the rated working voltage. Design improvements currently in development may eventually eliminate the need for this large derating factor.

Table 1. EIA Standard Capacitance package sizes.

Capacitance (pF)	Working Voltage (volts)					
	4	6	10	25	35	50
0,1				3216	3216	
0,15					3216	3528
0,22					3216	3258
0,33					3216	3528
0,47				3216	3528	6032
0,68					3528	6032
1,0				3528	6032	
1,5		3216	3528	6032	7243	
2,2	3216			6032	7243	
3,3	3216			6032	7243	
4,7		3528		7243		
6,8	3528		7243			
10,0	3528		6032	7243		
22,0	6032					
33,0	6032		7243			
47,0		7243				
68,0	7243					

A1.2 Resistors

Just like capacitors, surface mounted resistor chips are available in both rectangular and cylindrical formulations, each of which has its own resistance, power, and tolerance ratings. The most common are thick film, thin film, bulk metal foil, and wirewound. Thick film chips, which are the resistor counterparts to MLCs, represent the bulk of the surface mount market for individual resistor chips.

A1.2.1 Rectangular chip resistors

Physical characteristics

Most rectangular chip resistors are based on thick film process technology. As shown in Figure A1.4, the active element consists of a thick paste, which has been screened and fired over a ceramic base. The base material is 96% alumina (a composite of 96% aluminium oxide Al_2O_3, and 4% other oxides). The resistive element is ruthenium oxide or similar paste. To protect the resistor a glass layer is fired over it.

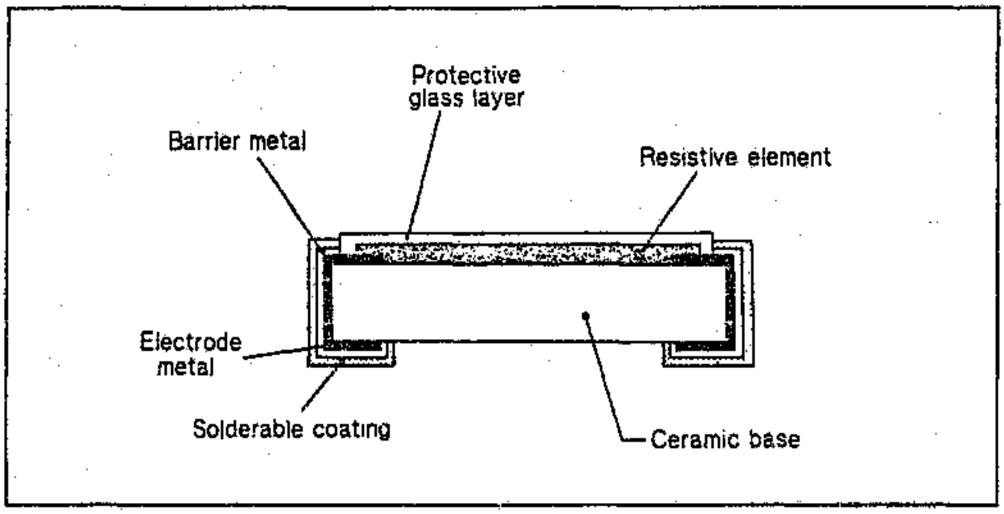


Figure A1.4 Construction of thick film rectangular chip resistor.

Electrical contact is made from the ends of the device. A thick film of conductive material is fired to form the electrodes. As with chip capacitors, the end terminations are protected with a nickel or copper barrier layer to prevent dissolution of the precious metal electrode during soldering. Finally, a tin-lead coating is applied over the barrier metal to provide a highly solderable surface.

Electrical characteristics

The most popular member of the standard family is the 1206 resistor(3). It is rated for 0,125W power dissipation or 200 V DC maximum continuous voltage. The 0805 and 1210 meet the lower and higher power requirements respectively. Recommended power derating characteristics for this family are presented in Figure A1.5. These products are commonly available in resistance values from 10 ohms to 1 megaohm and in several tolerance ranges ranging from 0,1% through to 20%.

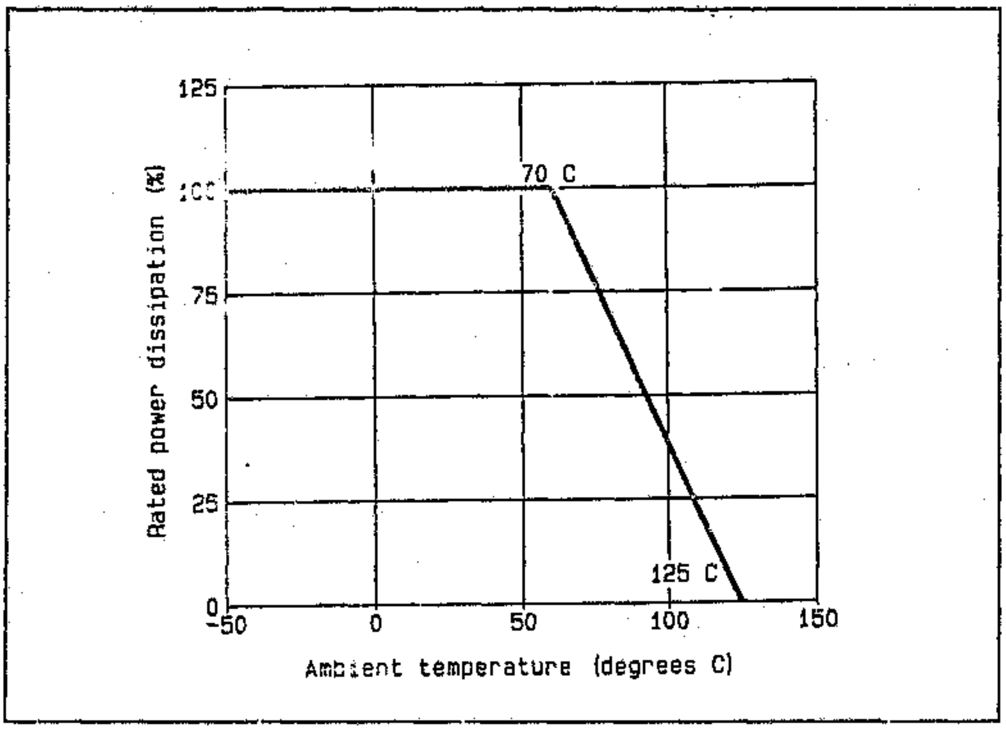


Figure A1.5 Power derating characteristics for thick film chip resistors.

A1.2.2 Cylindrical or MELF (Metal Electrode Face) resistor.

Physical characteristics

The MELF resistor shown in Figure A1.6 is adapted from conventional through-hole resistor technology. Rather than attaching leads for insertion mounting, the electrodes are metallized and solder-coated for surface attachment. The most commonly used MELFs are of either carbon film or metal film construction(3). Carbon film resistors are low-cost products intended for general purpose use. The somewhat more expensive metal film products offer tighter tolerances, improved temperature stability, and better noise performance.

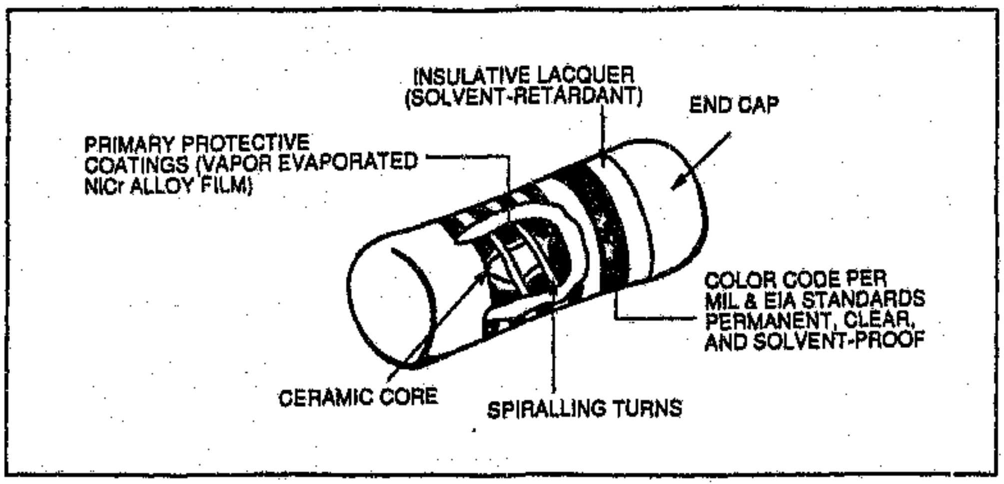


Figure A1.6 Construction of MELF resistor.

Both products employ a similar physical construction. The resistive element (chromium film or nickel-chromium alloy) is vapor-deposited onto a ceramic core. Metal end caps are welded to the base to form electrodes. An insulating, solvent-resistant resin coating is applied over the element, and colour coding bands are applied.

Electrical characteristics

Carbon film resistors are typically available in values from 1 ohm to 2 megaohms. Tolerances of 5% are easily produced. Metal film resistors range in value from at least 10 ohms to 1 megaohm with 1% or better tolerance. Several power ratings are presently available from 0,125W/200V to 1W dissipation.

A1.3 Inductors

Surface mount inductors are available in either wirebound or multilayer construction. Several levels of performance can be obtained, but the variety of product offerings does not yet approach that of through-hole technology. Although the EIA is considering the issue, there are as yet no officially recognized standard sizes for surface mount inductors.

A1.3.1 Wirewound inductors

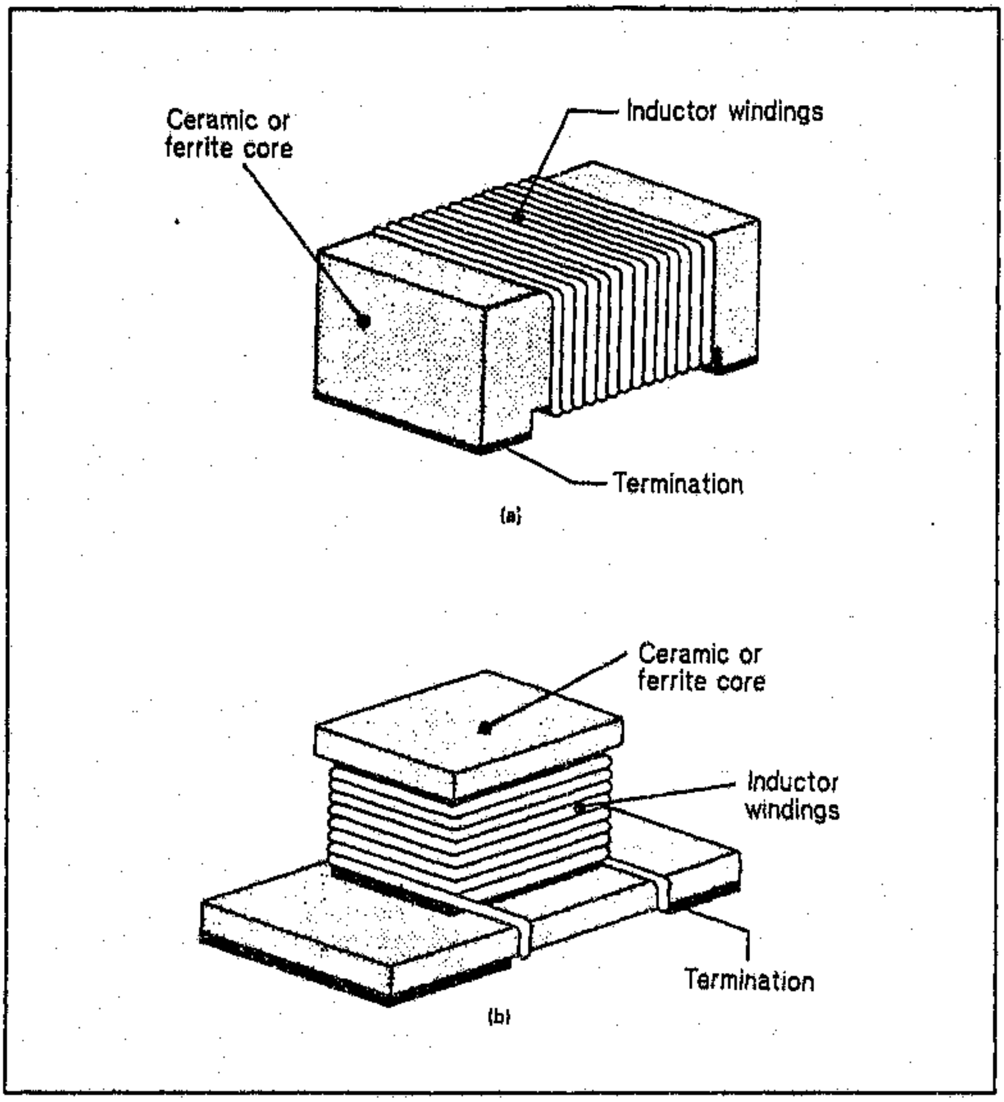
Wirewound inductors consist of a number of turns of fine wire wrapped around a core material. Low inductance values employ a ceramic core, while larger values need the higher permeability of a ferrite core. As shown in **Figure** A1.7, the windings may be orientated either vertically or horizontally.

A1.3.2 Multilayer inductors

Multilayer inductors are similar in construction to multilayer capacitors. They typically consist of alternating layers of ceramic or ferrite paste and conductor paste in a monolithic configuration. Inductance values extend from about 0,01H to 200H in body sizes identical to the 1206 and 1210 ceramic capacitors.

Multilayer inductors tend to have somewhat higher self-resonant frequencies, higher series resistances, and lower Q-factors than similar wirewound inductors. Common available tolerances are 5% and 10%, but with no industry standards available.

Other important passives now suitable for SMT (which can survive virtually any vapor-phase or infrared-soldering techniques) include LEDs, DIN41612 and other connection systems, crystals and oscillators, push button and toggle switches.



a. Vertical windings, b. Horizontal windings.

Figure A1.7 :Construction of wirewound inductors.

A1.4 Semiconductor Devices

A1.4.1 Introduction

Surface mount semiconductor devices (transistors, diodes, integrated circuits, and related products) employ the same silicon die as used in their through-hole counterparts. Any change in overall device reliability is strictly due to the difference in package format.

The lack of standardization of package style appears to be limiting the growth potential of this technology. Though there is an eminent need to define package parameters consistent with the philosophy of surface mount techniques, there are some basic requirements which any packaging style must reflect.

First, in the case of ICs, the package is responsible for providing the electrical connection between the internal die and the exterior circuit path. A path must also be provided for heat dissipation. The lack of a good path for the heat to escape creates a potentially hazardous operating situation. The package must also be able to provide an interior environment compatible with the device's performance and reliability parameters, and also the package must be strong enough to permit the entire structure to withstand stresses occuring during manufacturing, assembly, test and actual use.

The task of the device designer is to achieve the highest possible reliability at the lowest possible cost. This has led to the development of two major package families. Hermetically sealed packages provide the higher level of reliability, but at a higher cost. Plastic encapsulated packages provide a lesser level of protection, but at a much lower cost.

To ensure general acceptance, microcircuit packages must provide the following attributes:

Versatility
- Variable lead count
- Compatible with all die attach/interconnect methods
- Capable of high speed signal transmission
- Rugged, small and lightweight

Readily available
- Simple design
- Easy customization
- Fast production turn-around

Cost effective
- Low cost
- Mass handling/automatic handling

High reliability
- Electrical and environmental characteristics consistent with design requirements
- Compatible package-to-board interconnect thermal coefficients of expansion (TCE) characteristics to eliminate interface effects
- Electrostatic Discharge (ESD) protected

Unlike other facets of surface mounting, converting a chip in a DIP, flatpack, or conventional leaded chip carrier to surface mounted form has proven a pleasant task. Often, no major alterations in the reworked chip or carrier are neccesary, except perhaps for chips that dissipate power. Minor rerouting of signal leads and various power-tab techniques are then employed, in which case the frame pins of one or two sides of the plastic leaded chip carrier (PLCC) package are tied to the die pad, and an external heat sink is added. Work is under way to bring the power limit to 5W but at present 2,5W is the best available.

As mentioned before, the Joint Electronic Device Engineering Council (JEDEC) is a well recognized purveyor of standards for the electronic industry. Standardization is reserved for those package types that meet more stringent requirements, enjoy widespread support, and are produced by a number of manufacturers. Most of the devices discussed in the following sections have been registered by JEDEC.

A1.4.2 Discrete semiconductor package styles

Most discrete semiconductors (transistors and diodes) are housed in plastic-encapsulated packages as per Figure A1.8. The most common packages for discrete semiconductors are the SOT (small outline transistor) configurations: SOT-23, SOT-89, and SOT-143. These packages are post-molded plastic outlines that cover small and medium power applications to about 500 mW dissipation. Higher power packages are less well defined.

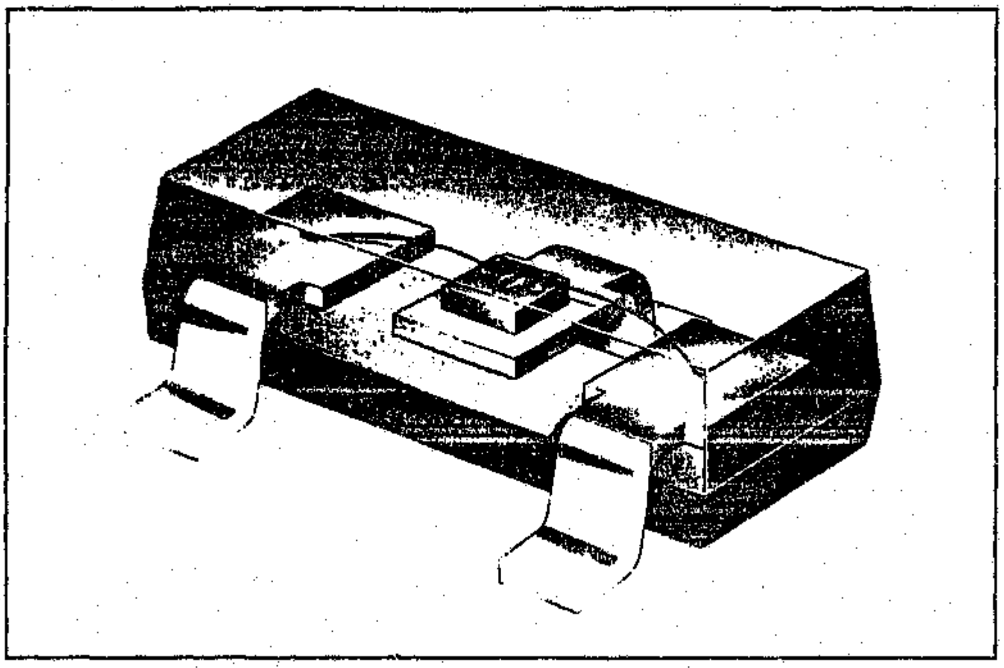


Figure A1.8 Internal construction of SOT-23 package.

Two types of hermetically sealed packages are available: the cylindrical MELF package is often used for diodes, while transistors are usually packaged in ceramic chip carriers.

The high power transistor is one area where SMDs are not being introduced. The specialised requirements of heatsinking these components usually transcend other aspects of design.

A1.4.3 Selection criteria for discrete semiconductors

Plastic encapsulated devices are recommended for most commercial and industrial applications in which ambient operating temperatures do not exceed 70°C. For more extreme enviroments, devices should be protected within a hermetically sealed package.

When plastic packages are acceptable, selection criteria is relatively straightforward. Only two significant issues must be resolved: selection of a specific SOT-23 outline and decision between SOT and MELF packages for diodes.

The JEDEC outline presently allows two variations. The **TO-236AA high profile** outline is used for reflow soldering applications. The large clearance between package and board permits cleaning solvents to penetrate and remove flux residues from under the device. The **TO-236AB low profile** outline is used when the component must be adhesively mounted to the board. The minimal standoff height ensures that the adhesive will fill the gap between component and board.

When hermeticity is neccesary, the alternatives are limited. MELF packages are recommended for diodes, but there are few choices for transistors. Although a hermetic version of the SOT-23 has been developed, it is not widely available.

A1.4.4 General integrated circuit (IC) design considerations

The first generation of integrated circuits were constructed as DIPs with 100-mil centres. The second generation ICs were the Small-Out-Line ICs (SOICs) and chip carriers (CC) with 50-mil centres. Today the industry is considering the third generation of ICs to accomodate pin counts in the 124 to 300 range on diminishing center spacings.

Surface mount ICs have either very short leads or no leads at all. They are generally produced on 0,050" lead centers. However, designs do exist on 0,040", 0,025", and 0,020" lead centers. This spacing refers to the center-to-center spacing width between adjacent leads and is reffered to as "pitch". This is a major design change from the traditional 0,100" center spacing of DIPs, a change which allows many more leads to be utilized on a package of similar dimensions.

Issues of general importance in the design of surface mount IC packages include the following:

- pinout configuration
- leadless vs. leaded design
- lead configuration
- lead coplanarity error

1 Pinout configurations

Surface mount packages are produced in two general pinout configurations. For small devices from about 8 to 28 pins, a dual-in-line approach is used. Pins are arranged in two parallel rows similar to the through-hole DIP. For higher pin counts, a quad configuration is preferred. This style employs pins on all four sides of a square or rectangular package.

2 Leadless vs. leaded design

The very first surface mount flatpacks were designed with long leads that extended straight out from the package body. Because the leads were fragile and easily deformed, flatpacks could not be automatically placed onto the PCB. Design changes led to the development of leadless ceramic chip carriers. These packages do not employ external leads. Instead, they contact the board through metallized electrodes deposited directly on the ceramic package, thus allowing for automatic placement. The absence of leads reduces the cost of the package and improves the electrical performance due to lower parasitic lead reactances.

However, leadless packages have their own set of problems. The coefficients of thermal expansion between a ceramic chip carrier and the subsrate to which it is being mounted are not well matched. For packages of about 28 pins and above, joint failure can occur after relatively few thermal cycles. Various methods can be employed to overcome this difficulty, but at a considerable increase in cost compared to standard techniques.

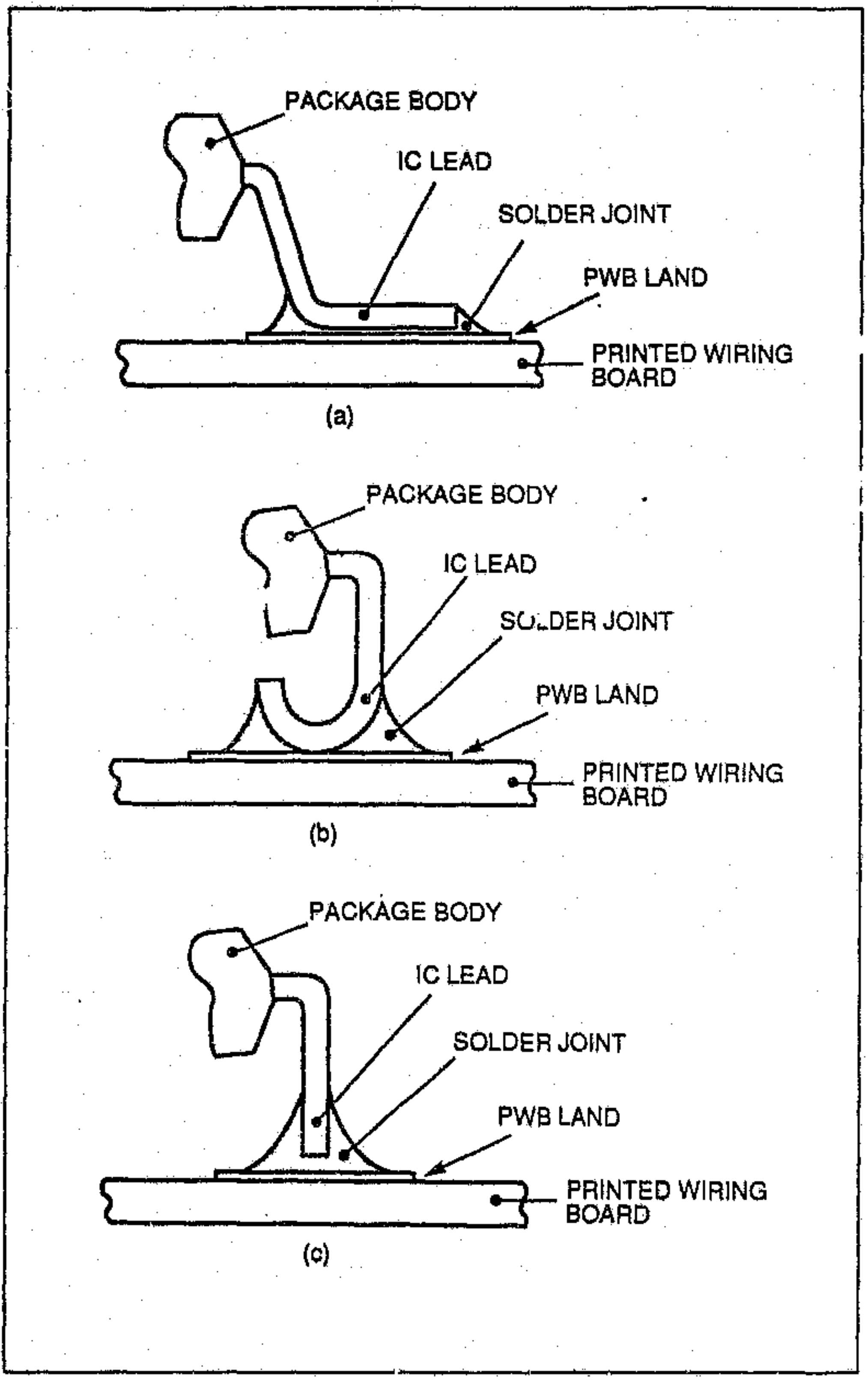
Most plastic packages now use a leaded configuration to provide a measure of compliance between package and board. Unlike the old flatpacks, leads are designed for compatibility with automatic assembly equipment.

3 Lead configuration

Three lead configurations are commonly used for surface mount IC packages: gull-wing, J-lead, and I-lead. All are compatible with automated assembly equipment, but each has various advantages and disadvantages. (Refer to Figure A1.9 below)

Gull-wing

The gull-wing lead stretches down and outward away from the device body, forming a lap joint interconnection **Figure A1.9a**. A primary feature of this lead is the ease with which the finished joint can be visually inspected. Another advantage is that the device can be electrically tested through the use of a relatively simple probe fixture.



a. Gull-wing
b. J-lead
c. I-lead

Figure A1.9 SMT lead configurations

The exposed lead design also presents problems. It is susceptible to damage during handling, and even slight stresses can deform thin leads beyond specification limits. Leads bent out of the seating plane by as little as 0,12mm (0,005in) may not solder properly at assembly. Another disadvantage is the large package footprint in relation to body size. Although the body is small, the extended leads increase the overall outline by a considerable amount.

J-lead

The J-lead was designed to overcome the disadvantages of gull-wing leads in that the lead is rolled under the package for protection against ordinary mechanical damage. Two basic lead styles exist. The first configuration, shown in **Figure A1.9b**, is considered a compliant-type lead. The lead is bent at a 90-degree angle under the package at the board interface where it becomes soldered to a metallized footprint. The other type, is considered non-compliant, and describes a lead that bends at the board surface and bends back to be attached to the package's underside. The difference between the two is considered critical due to the inability of the non-compliant lead style to withstand numerous thermal/power cycles, contributing failure characteristics similar to those of leadless chip carriers.

I-lead

The I-lead (or butt joint), is illustarted in **Figure A1.9c**. It is a relatively new concept that offers the advantages of the J-lead without most of the drawbacks. Leads are formed in a simple shearing operation that assures precise palanatiry. Once formed, they are very resistant to damage. Even if a lead is slightly bent, there is little risk of soldering problems because it remains in the same plane as all the other leads. The deformation would have to be so severe as to position the lead entirely off the corresponding land before a serious problem would occur. The mechanical strength of a properly formed I-lead has been shown to equal or exceed that of the gull-wing and J-lead configurations.

The primary disadvantage of the I-lead is that leads must be tinned with solder after shearing. Otherwise, the exposed base metal at the bottom of the lead will not exhibit acceptable solderability, and solder voids can occur under the lead. Large stress concentration can build up in the solder surrounding the void, increasing the risk of fractures and subsequent mechanical failure of the joint. Solder tinning must be performed by dipping the leads in a solder bath or by use of a wave soldering machine.

A1.5 Integrated Circuit package configurations

A multitude of package styles is available for surface mount ICs. These range from relatively inexpensive post-molded plastic packages to hermetic ceramic packages. Characteristics of several of the more prevalent package styles are described below.

A1.5.1 Small Outline Integrated Circuit (SOIC)

As illustrated in **Figure A1.10**, the SOIC employs the dual-in-line lead configuration (DIP) with gull-wing leads spaced on 0,050-in (1.27-mm) centres. Two package outlines have been standardized by JEDEC under the MS-012 and MS-013 drawings. The narrow-body outline has a body width of 0,150-in (3,81-mm) and covers lead counts of 8, 14, and 16 leads. The wide body-outline, sometimes called the SOL (small-outline-large), has a body width of 0,30 in (7,62 mm) and covers lead counts of 16, 18, 20, 24, and 28 leads. The package loses its attractiveness above the 28-pin count, at which point (1) the package becomes fragile and hard to handle, (2) lead-inductance problems begin to match those of the DIP and (3) board real estate advantages become insignificant.

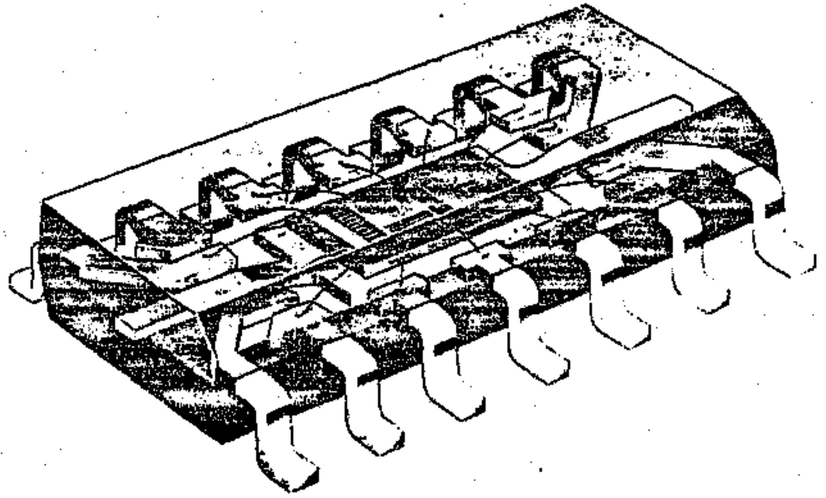
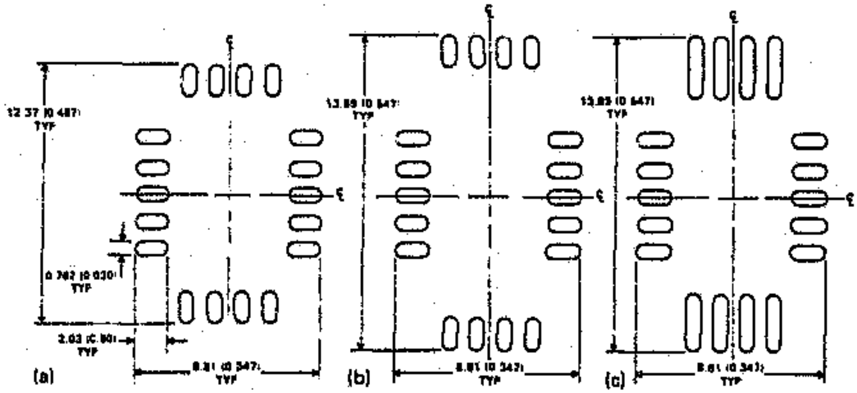


Figure A1.10 Internal construction of SOIC device

A1.5.2 Plastic Leaded Chip Carrier (PLCC)

This product was developed to meet the need for a low-cost plastic package for higher lead-count devices. It employs J-leads on 0,050-in (1,27-mm) centers in a quad configuration. In the square outline used for most digital and linear ICs, the lead count is divided equally among all four sides. A rectangular outline is preferred for memory chips because it more closely matches the geometry of the silicon chip.

The JEDEC MO-047 drawing defines outlines for square packages with lead counts of 20, 28, 44, 52, 68, 84, 100, and 124 leads. The MO-052 drawing covers rectangular packages with 18, 22, 28, and 32 leads. One complication is the exsistance of two sizes of 18-lead rectangular packages. The smaller outline Figure A1.11a, was developed for 64k dynamic random access memory (DRAM) chips. When 256k DRAMs were introduced, they also needed an 18-pin package but were too large to fit into the outline for the 64k chip. A package with an extended outline Figure A1.11b was developed to meet this need. The design of the extended version is such that both packages can be accomodated by a single land pattern design with extended land patterns in the long direction.



a: 64K DRAM
b: 256K DRAM
c: Composite land pattern

Figure A1.11 Land patterns for rectangular PLCC packages.

A1.5.3 Advanced package configurations

The size of an IC package is rarely dictated by the size of the internal die. The limiting parameter is usually the space necessary to fan out the leads to the required pitch. As lead counts continue to grow, the 0,050-in lead pitch becomes extremely inefficient. Although the PLCC package outlines have been defined for lead counts well above 100 pins, such packages are difficult to manufacture and consume an excessive amount of board space. A practical limit to maximum package size seems to occur when the length of the package reaches about 25-32mm (1,0-1,3in) along a side. For a package with 0,050-in lead pitch, this limit is reached at about 84 leads. For higher pin counts, package size must be reduced by adopting a finer lead pitch.

The quadpack was the first to exploit the higher density possible with tighter spacing, but its unprotected gull-wing lead format has discouraged widespread use. Several improved outlines are currently in development, but none are widely available and it is uncertain which [illegible] evolve into usable standards. It is likely [illegible] of these packages will undergo considerable [illegible] before being made commercially availa[illegible] descriptions are intended to ill[illegible] industry trends rather than endorse specific o[illegible]

Plastic quad flatpack (PQF)

Using the gull-wing lead format, the existing quadpack offers advantages that many users find attractive. Lead pitches of at least 0,60mm (0,024 in) are possible, greatly improving packing density compared to the PLCC. Two serious disadvantages, however, have limited its usage: leads are easily damaged, and the thin body outline is prone to cracking.

To avoid the problems of the quadpack while using a similar configuration, a group of US manufacturers has proposed the package shown in Figure A1.12. It has gull-wing leads on a 0,025-in pitch and accomodates lead counts from 44 to 244 leads. A prominent feature is the inclusion of "bumpers" at the corners to protect the leads. This permits packages to be transported in tape-and-reel or tube formats without damage to the leads. Except for the ears, body dimensions are identical to those of the PLCC.

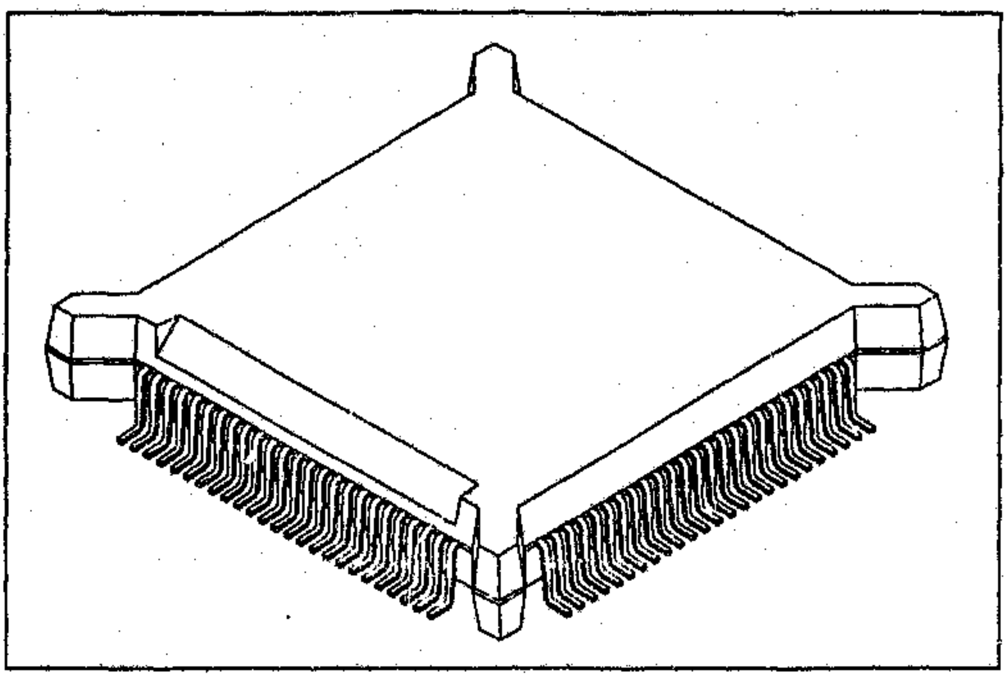


Figure A1.12 JEDEC-registered plastic quad flatpack

Chip-on-Board (COB)

COB is the placement of the unpackaged semiconductor device, discrete or integrated circuit, directly onto a substrate. This substrate is usually a conventional PC board and passive components are discrete chip, surface mount types instead of thick film.

Various techniques have been devised for mounting bare semiconductor die directly onto PCBs. These include chip-and-wire bonding, TAB, and flip-chip attachment.

Few standards currently exist for devices or mounting processes. COB technology is mainly used in very high volume applications which can justify the effort to develop an entire manufacturing process.

A1.6 Connectors and Electromechanical Devices

Advances in electromechanical devices such as connectors, relays, sockets, and switches have traditionally followed rather than lead other component technologies. This has again been the case in the conversion to SMT. Compared to the state of standardization for passive and active components, electromechanical devices lag far behind. As end-users wholeheartedly convert from through-hole to surface mount, they increasingly discover that the only remaining through-hole components are the electromechanical devices. Faced with the prospects of supporting an entire through-hole assembly line for these few devices, they are strongly lobbying for new surface mountable product families.

A1.6.1 Connectors

Printed circuit edge connectors epitomize the technical problems of converting electromechanical devices from through-hole to SMT. They are often large, bulky devices that are awkward to handle with automatic equipment, and must sometimes withstand repeated insertions and withdrawals without physical damage. In many cases they also serve as the sole form of mechanical support for the PCB. Four factors have been identified as being critical to connector design. These are:

- lead configuration
- plastic molding compound
- mechanical support
- lead finishes

Details of these problems are beyond the scope of this document, but the reader should be aware of their exsistance.

A1.6.2 IC Sockets

Sockets for integrated circuits have many uses. During engineering development, they permit ICs to be rapidly changed so that circuit performance over a large number components can be evaluated. In production, they are often used for custom ROM (read-only memory) chips or ASICs (application-specific ICs) that must be individually configured based on customer specifications. Wherever ICs must be rapidly and routinely changed, IC sockets are desirable.

Surface mount sockets come in two styles. The first, designed for through-hole insertion is used to adapt a surface mount IC for through-hole mounting (See Figure A1.13).

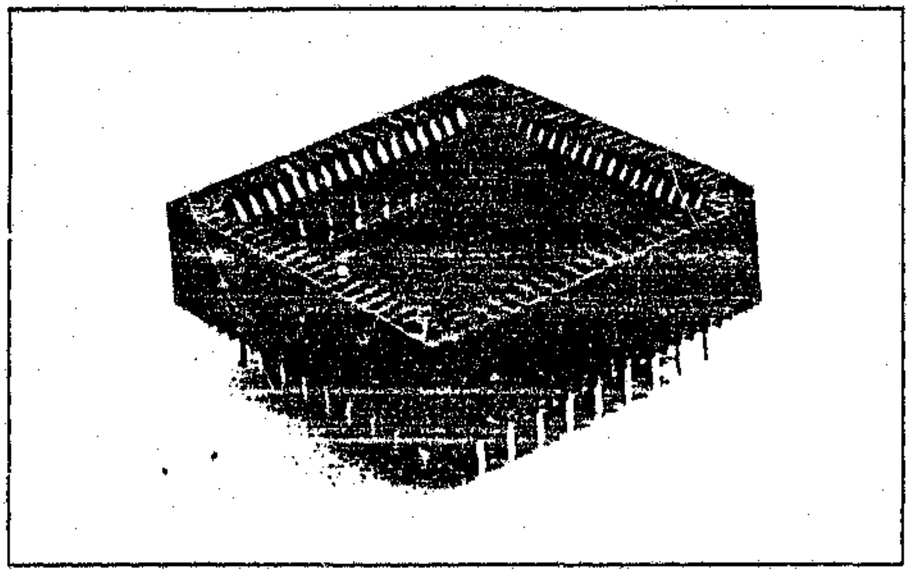


Figure A1.13 Socket to adapt surface mount ICs for through-hole mounting

This is attractive when the benefits of the surface mount package (such as small size and reduced parasitic reactances) are desired on a board that would otherwise be totally through-hole.

The second type of socket, is itself designed for surface mounting. It fits roughly the same footprint as the original package, so if designed properly, the board can accept either the IC or the socket interchangeably.

Since sockets do not form metallurgical bonds with the component leads but rather depend strictly on mechanical contact, they are not as reliable as a soldered connection. Contacts can corrode in high humidity environments, and mechanical contact can be interrupted during shock or vibration. They should be used only when their advantages outweigh their disadvantages.

Appendix 2. IN PROCESS INSPECTION

A2.1 Final Inspection

This is the last opportunity to catch defects before shipping the product and is the most common inspection point. It is performed after all processing has been completed, usually in conjunction with a final electrical test.

Final inspection is not an ideal point to monitor process performance. The observed defects are a composite of the entire process, making it difficult to ascribe a specific defect to a particular process step. Often, final inspection is used primarily as a "gate" to prevent defective products from escaping. In light of the previous discussion, it would seem that this type of inspection is at odds with world-class manufacturing philosophy. Technically this may be true, but the practicalities of real-world manufacturing often make such an inspection essential.

Consider a surface mount manufacturing facility that achieves defect levels of 100 parts per million on a per-joint basis. This means that out of every 1 million joints produced, it can be expected that about 100 of them will be defective. (This defect level, while not extraordinarily low, would be an aggressive goal for many factories). This probability that a given joint will be acceptable is thus:

$$P_J = 0.999900$$

and the probability that any given board will be acceptable is determined by the binomial probability distribution:

$$P(n) = P_J^n$$

where n = number of joints per board.

The probability of a given board being defective is then just:

$$P_d = 1 - P(n) \ldots\ldots\ldots\ldots\ldots \quad (1)$$

Figure A2.1 plots eqn [1] for various process defect levels. For example, if the above process were used to assemble boards with 5000 joints per board (typical of a computer memory board), about 40% of the boards produced would contain at least one defect. The need for a comprehensive final inspection is then readily understood - it is of little comfort to know the process is in control if nearly half the boards shipped are defective! Until process performance can be improved by several orders of magnitude, a final outgoing inspection will continue to be mandatory.

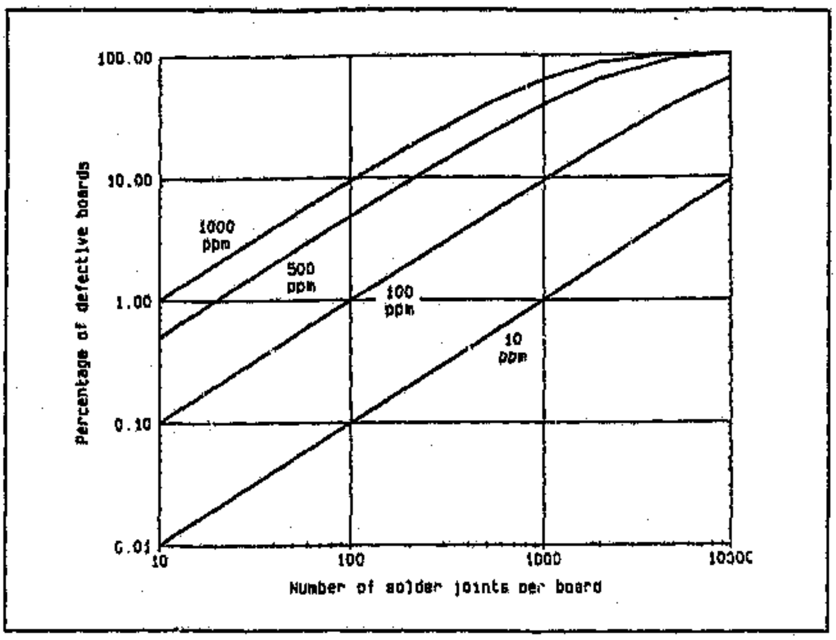


Figure A2.1 Impact of solder process defect levels on final assembly yield. The ppm defect levels refer to the process yield on a per-joint basis.

A2.2 Post-Solder Inspection

This station is designed to monitor the quality of the SMT soldering process. It tracks solder joint defects, such as inadequate solder volume, poor wetting, solder bridges, solder balls, etc. These types of defects are exceedingly difficult to inspect automatically, so human vision is most often used. Some success has been reported with X-ray and infrared techniques.

Post-solder inspection suffers from limitations similar to those at final inspection: since it is so far downstream in the process, it is difficult to tell whether a particular defect is due to the soldering operation or a previous process. For example, if a component is missing, did it fall off during the soldering process or was it never mounted by the component-placement machine? To minimize uncertainty, the soldering system should be preceded by a post-placement inspection step. In this way, the quality of the product prior to soldering can readily be determined.

A question that often arises is how to account for defects that are most probably due to problems at previous processes. If a reflow-soldered joint is defective because it apparently never received solder paste, should it be counted as a soldering defect or a screen printing defect? According to one school of thought, the inspector should make an intelligent determination of which process step was the probable cause and ascribe the defect to that process.

Another school of thought argues that regardless of where the defect actually occurred, since it was first observed at the post-solder inspection step, it should be counted as a post-solder defect.

Process monitoring however, is intended to identify potential process problems, not to place blame on a particular machine or operator. By counting the defect at the point where it was first observed, someone is assured of taking ownership of the problem. Otherwise it is easy for an inspector to dismiss it as being caused by another process step upstream without following up to confirm his suspicion.

A2.3 Post-Placement Inspection

This step provides feedback on the quality of the component-placement process. It looks for placement-related errors, such as missing, skewed, and incorrect components, and polarized parts installed backwards. Although often performed manually, it is easier to automate that post-solder inspection.

A2.4 Post-Screen Printing Inspection

The one point in the process where rework is actually easy is immediately after screen printing. If a defect is found, the paste is simply washed away and the board rescreened. The inspection process at this point need not be exotic. Frequently it consists of a straightforward visual examination by the screen printer operator. If this person doesn't like the results of the print, he or she simply sets the board aside for later washing and reprinting.

It is easy to let this inspection point remain informal and not record the data. However, it is good to consider the possibility of maintaining a simple control chart to track the process. This might be nothing more than a pn chart that plots the number of consecutive good boards produced before discovering a defect. The advantage of the control chart is that it helps identify evolving problems that may escape the notice of an operator whose primary concern is to meet the production schedule.

Appendix 3. FOOTPRINT SPECIFICATIONS

A3.1 Footprints for wave soldering

To determine the footprint of an SMD for a wave soldered substrate, there are four main interactive factors to consider:

- the component dimensions plus tolerances - determined by the component manufacturer;
- the substrate metallization - positional tolerance of the solder land with respect to a reference point on the substrate;
- the solder resist - positional tolerance of the solder resist pattern with respect to the same reference point; and
- the placement tolerance - the ability of an automated placement machine to accurately position the SMD on the substrate.

The co-ordinates of patterns and SMDs have to meet a number of requirements. Some of these have a general validity, for example, the minimum overlap of the SMD metallization and solder land, and available space for solder maniscus. Others are specifically required to allow successful wave soldering. One has, for example, to take account of factors like the "shadow effect" , the risk of solder bridging (see below), and the available space for a dot of adhesive.

The "shadow effect"

In wave soldering, the way in which the substrate addresses the wave is important. Unlike wave soldering of conventional printed boards, where there are no component bodies to restrict the waves freedom to transverse the whole surface since there are no components on the solder side, wave soldering of SMD substrates is inhibited by the presence of SMDs on the solder side of the board. The solder is forced around and over the SMDs as shown in Figure A3.1a, and the surface tension of the molten solder prevents it reaching the far end of the component, resulting in a dry-joint downstream of the solder flow. This is known as the "shadow effect".

The shadow effect becomes critical with high component bodies. However, wetting of the solder lands during wave soldering can be improved by enlarging each land as shown in Figure A3.1b. The extended substrate metallization makes contact with the solder, allowing it to flow back and around the component metallization to form the joint.

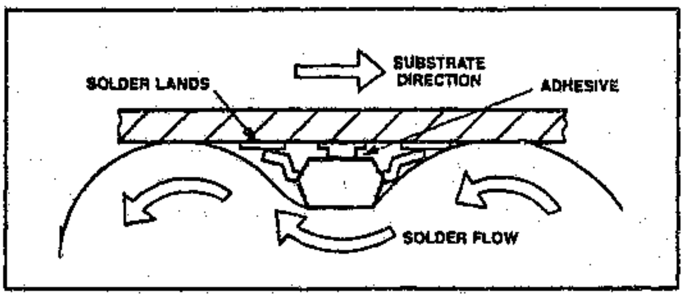


Figure A3.1a Surface tension preventing the solder reaching the downstream end of the SMD.

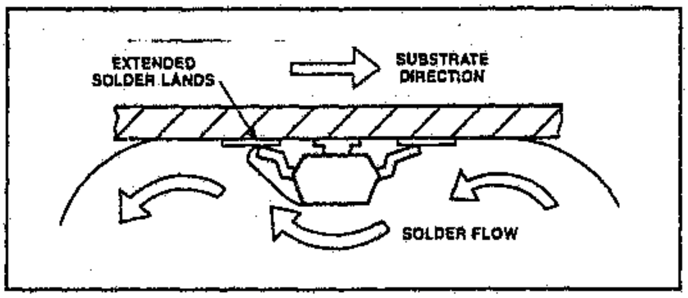


Figure A3.1b Extending the solder land to overcome the shadow effect.

The use of the dual-wave soldering technique also partially alleviates this problem because the first turbulent wave has sufficient upward pressure to force solder onto the component metallization, and the second, smooth wave 'washes' the substrate to form good fillets of solder. Similarly, introducing oil on the surface of the solder wave lowers the surface tension which lessens the shadow effect, but this technique introduces problems of contaminants in the solder when the oil decomposes. (See details on soldering later).

Solder bridging

On wave-soldered substrates the orientation of SO (small outline) and VSO (very small outline) ICs is critical for the prevention of solder bridge formation (See Section 8). Optimum solder penetration is achieved when the central axis of the IC is parallel to the flow of the solder as shown in Figure A3.2a. The SO package may also be transversely orientated, as shown in Figure A3.2b, but this is totally unacceptable for the VSO package.

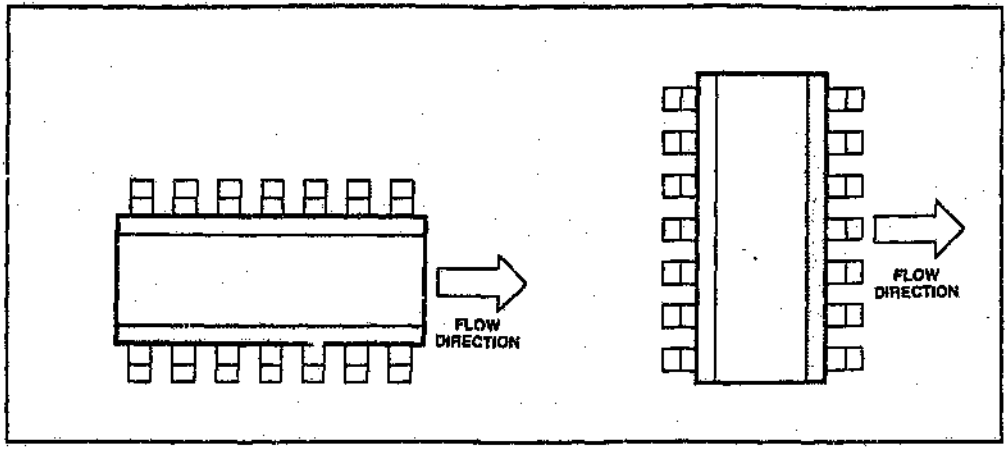


Figure A3.2a Parallel orientation for SO and VSO packages. A3.2b Transverse orientation for SO packages only.

Another major cause of solder bridges on SO ICs and PLCCs is a slight misalignment as shown in Figure A3.3. The close spacing of the leads on these devices means that any inaccuracy in placement drastically reduces the space between adjacent IC pins and solder lands, thus increasing the chance of solder bridge formation.

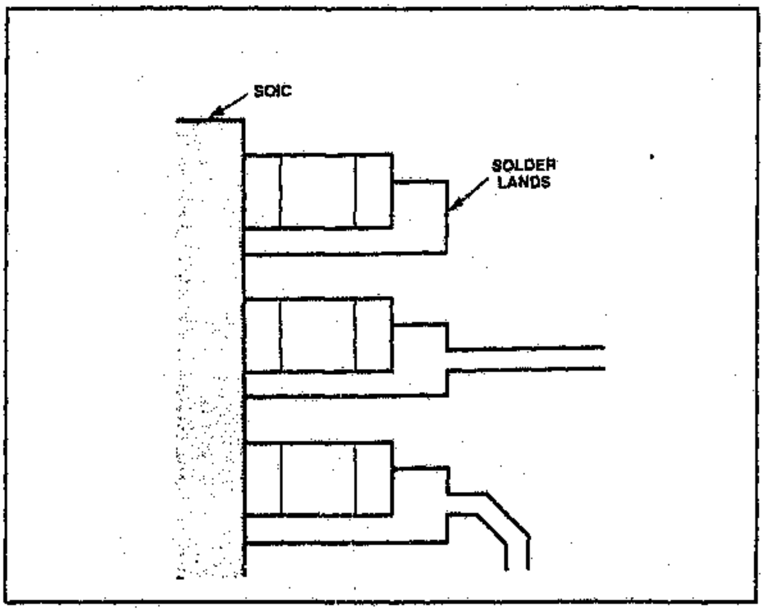


Figure A3.3 Misaligned placement of SO package increases the possibility of solder bridging.

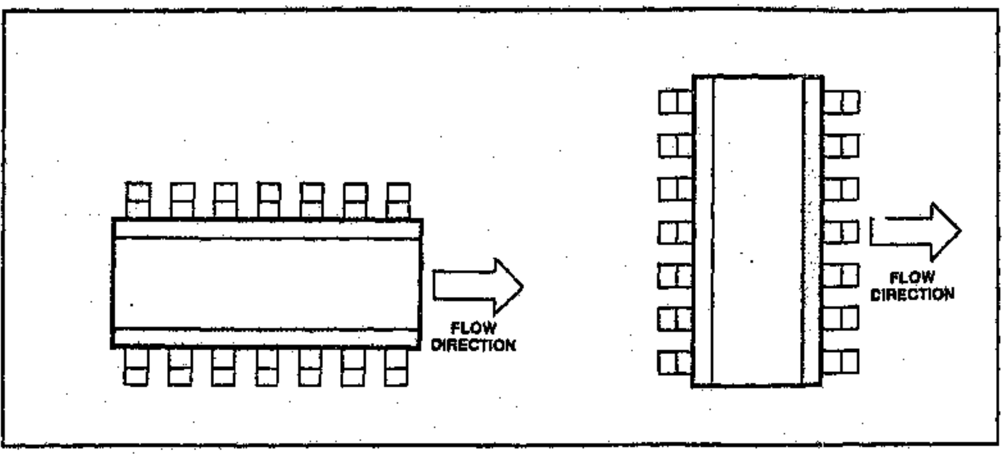


Figure A3.2a Parallel orientation for SO and VSO packages.

A3.2b Transverse orientation for SO packages only.

Another major cause of solder bridges on SO ICs and PLCCs is a slight misalignment as shown in Figure A3.3. The close spacing of the leads on these devices means that any inaccuracy in placement drastically reduces the space between adjacent IC pins and solder lands, thus increasing the chance of solder bridge formation.

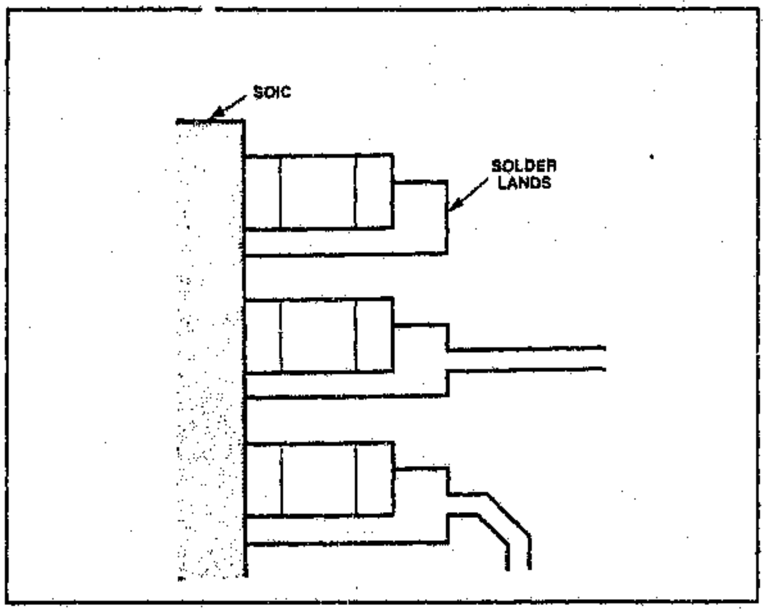


Figure A3.3 Misaligned placement of SO package increases the possibility of solder bridging.

Dummy tracks for adhesive application

For wave soldering, an adhesive is required to affix components to the substrate. This is necessary to hold the components in place between the placement operation and the soldering process (this is discussed in detail in Section X). The amount of adhesive applied is critical for two reasons: first, the adhesive dot must be high enough to reach the SMD, and second, there must not be too much adhesive which could foul the solder land and prevent the formation of a solder joint.

The solution to this problem is to place a dummy land under the device as shown in **Figure A3.4**. This will reduce the effective component standoff height and controls the spread of adhesive. As an alternative, a functional circuit trace may be routed under the component if the high density of SMD substrates necessitates the routing of normal tracks between solder lands, but where it does not, a short dummy track should be introduced.

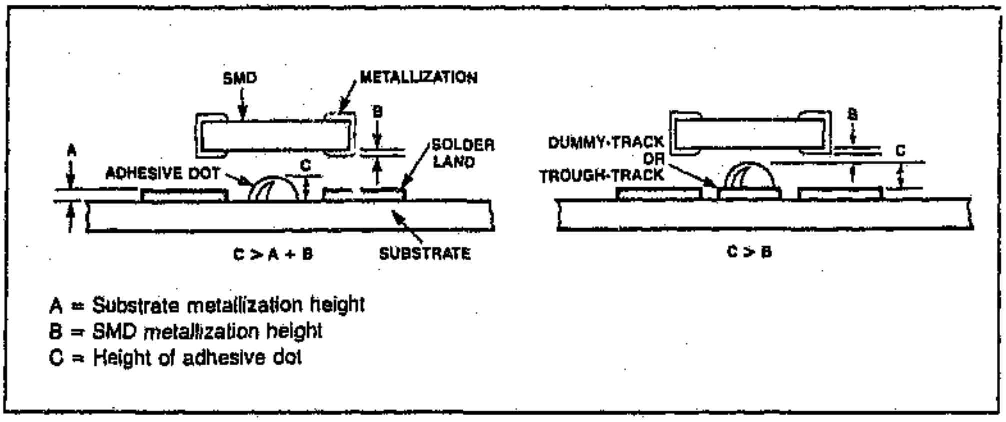


Figure A3.4 Adhesive dot height criteria. Through-track or dummy track to modify dot height criteria.

A3.2 Footprints For Reflow Soldering

To determine the footprint of an SMD for a reflow soldered substrate, there are five interactive factors to consider. The four that effect wave solder footprints (although solder resist may be omitted), plus an additional factor relating to the solder cream application which is applied using screen printing techniques. That is, the positional tolerance of the screen printed solder cream with respect to the solder lands. The solder cream density combined with the required amount of solder, determines the minimum area of solder land. The footprint dimensions for the solder cream pattern are typically identical to those for the solder lands used in wave soldering.

A3.3 Standard Land Pattern Formulae

A convenient technique for specifying component clearances is to define, for each device land pattern, a "clear area" that must be free of components. The clearance between any two types of SMCs is determined by the larger of the two individual clearances. The recommended land patterns in this section include suggested clear area requirements to accomodate the needs of test and visual inspection, and are denoted by the "A" and "B" dimensions.

A3.3.1 Passive component land patterns

Land patterns for passive components can be found from the following formulae:

$X = W_{max} - K$ (wave soldered)

$X = W_{max} + K$ (reflow soldered)

$Y = H_{max}/2 + T_{max} + K$

$G = L_{min} - 2T_{max} - K$

where
X = land width
Y = land length
G = gap between lands
W = component width
H = component height
L = component length
T = termination length
K = 0,25mm for reflow soldered components
0,50mm for wave soldered components

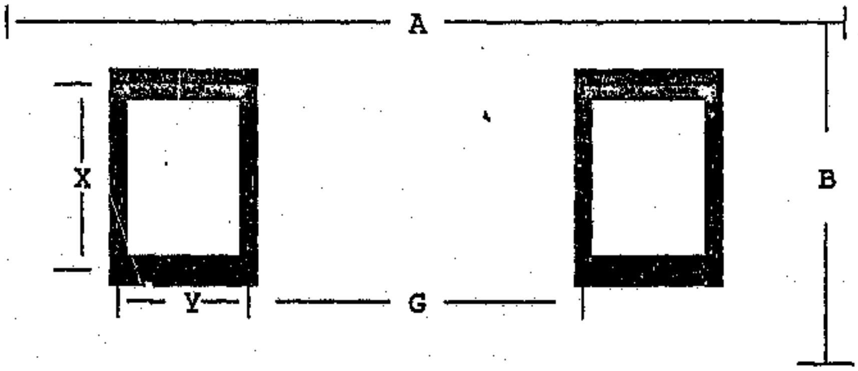


Figure. A3.1: Land pattern relationships for two-terminal passive devices

The approximate formulae described above break down for extremely small components, such as 0805 capacitors and resistors.

The recommended land pattern relationships for chip resistors and ceramic capacitors are shown below.

Package Style	Dimensions (mm) A	B	G	X	Y
0805	4.0	2.0	0.8	1.4	1.3
1206	7.0	2.5	1.8	2.0	1.3
1210	7.0	3.5	1.8	3.0	1.3

Table 1 Land pattern relationships for reflow soldered chip resistors.

Package Style	Dimensions (mm) A	B	G	X	Y
0805	4.5	2.0	0.8	1.0	1.5
1206	7.0	2.5	1.8	2.0	1.6
1210	7.0	3.[illegible]	1.8	3.0	1.6

Table 2 Land pattern relationships for wave soldered chip resistors.

Package Style	Dimensions (mm) A	B	G	X	Y
0805	4.0	2.0	0.8	1.4	1.3
1206	7.0	2.5	1.8	2.0	1.7
1210	7.0	3.5	1.8	3.0	1.8
1812	8.5	5.0	3.2	3.7	1.8

Table 3 Land pattern relationships for reflow soldered chip capacitors.

Package Style	Dimensions (mm) A	B	G	X	Y
0805	4.5	2.0	0.8	1.0	1.5
1206	7.5	2.5	1.8	1.3	2.0
1210	7.5	3.5	1.8	2.2	2.0
1812	Not recommended for wave soldering				

Table 4 Land pattern relationships for wave soldered chip capacitors.

It is recommended that tantalum capacitors and wirewound inductors be attached exclusively by reflow soldering. While it is possible to safely wave solder these devices, their tall outlines and protected leads are likely to promote solder shadowing.

Cylindrical MELF components can be wave soldered or reflow soldered. If reflow soldered, they should be adhesively attached to the board to prevent unwanted rolling of the parts. Recommended MELF land patterns are illustrated in Figure A3.2.

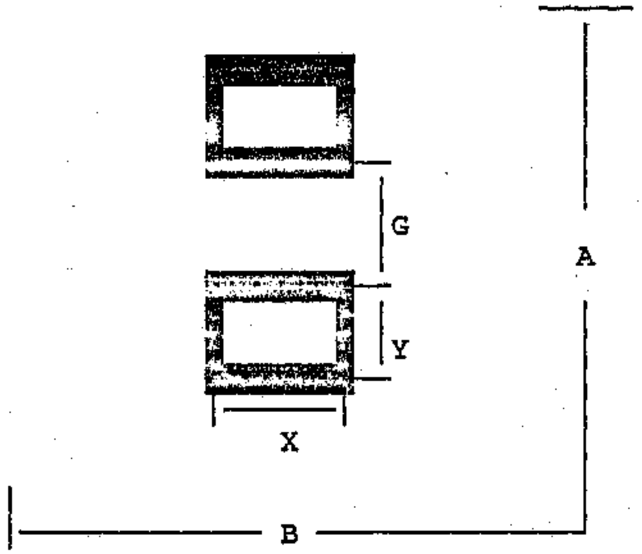


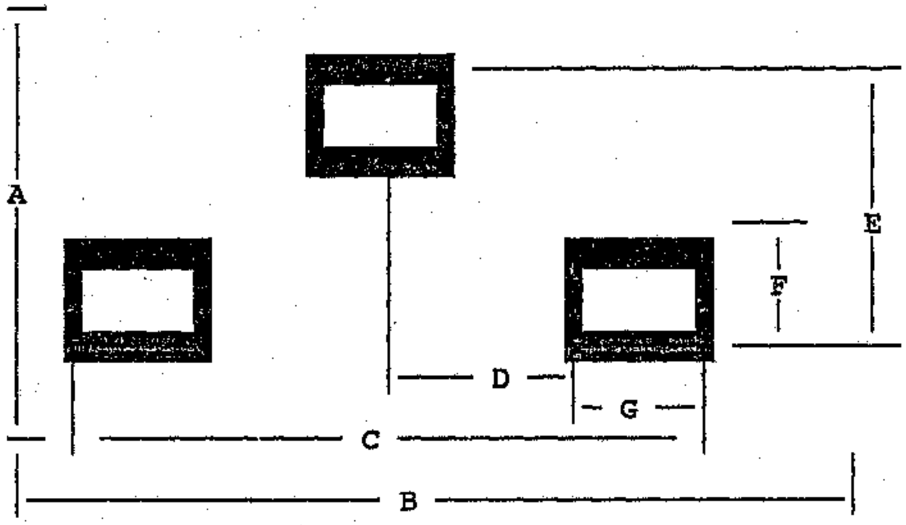
Figure. A3.2: Land pattern dimensions for cylindrical components.

Package Style	A	B	G	X	Y
	Dimensions (mm)				
Resistor 0.125W	6.5	3.0	2.0	0.9	1.6
Resistor 0.250W	8.5	4.0	4.1	1.5	1.7
Diode DO-213AA	6.5	3.5	2.0	1.1	1.6
Diode DO-213AB	1.8	2.1	3.5	8.5	4.5

Table 5 Land pattern dimensions for cylindrical components.

A3.3.2 Discrete semiconductor land patterns

Although no formulae are available to calculate generalized land patterns for these devices, the recommended dimensions are given in Figures A3.4, A3.5, A3.6.



	Dimensions (mm)						
Package Style	A	B	C	D	E	F	G
TO-236	1.2	0.9	2.8	0.5	3.6	5.0	5.0
SC-59	1.3	0.9	2.8	0.5	3.6		

Fig. A3.4: Land patterns for SOT-23 packages (dimension in mm).

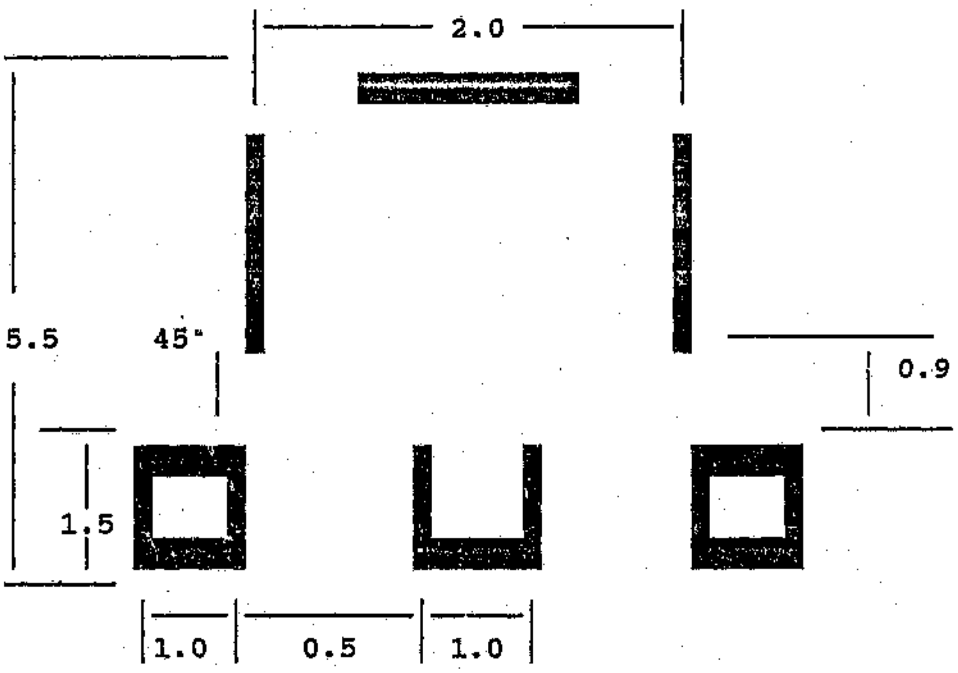


Fig A3.5: Land pattern for SOT-89 package (dimensions in mm).

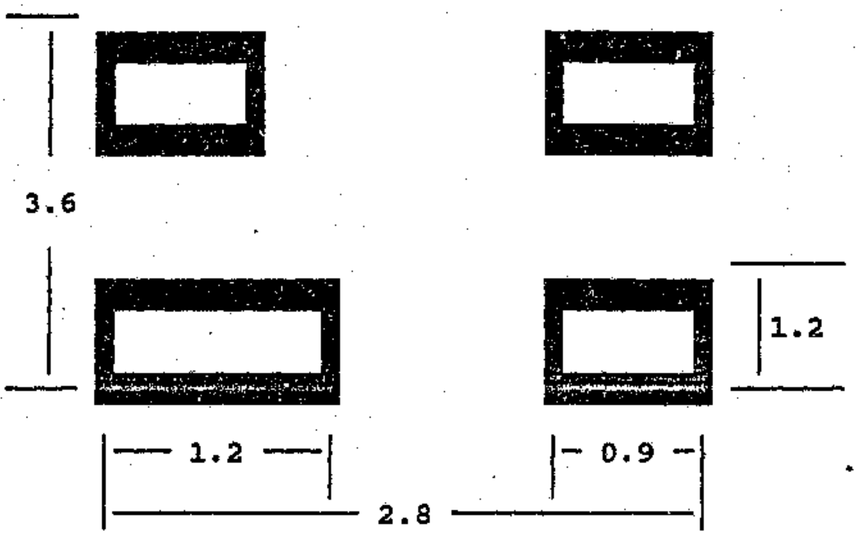


Fig. A3.6: Land pattern for SOT-143 package (dimensions in mm).

A3.3.3 SOIC package land patterns

The land pattern geometry for SOIC components is shown in Figure A3.7 and can be calculated from the following formulae:

$X = 0{,}60$ mm (0,024 in)

$Y = 1{,}80$ mm (0,070 in)

$G = A_{min}$

where X = land width
Y = land length
G = gap between the two rows of lands
A = component body width

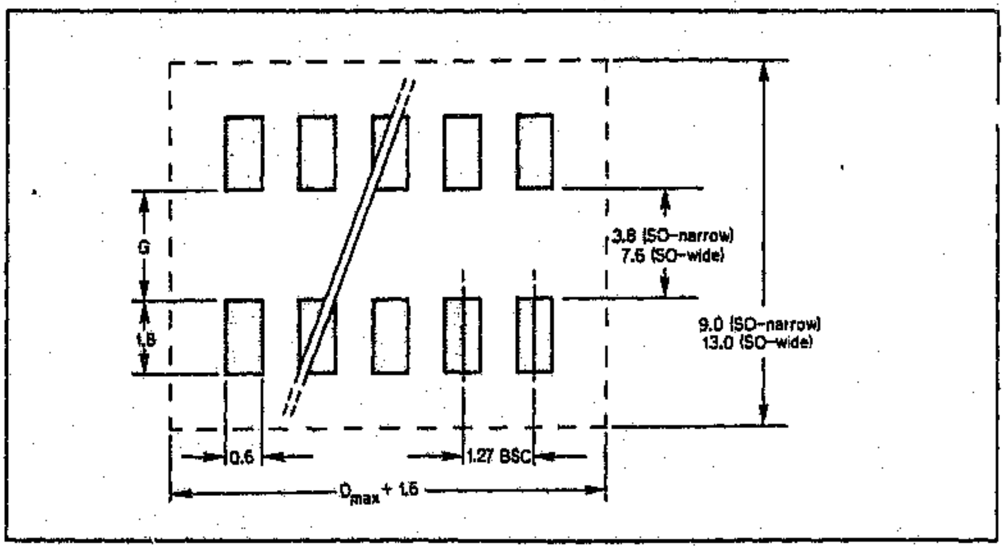


Figure A3.7 Footprints for SOICs.

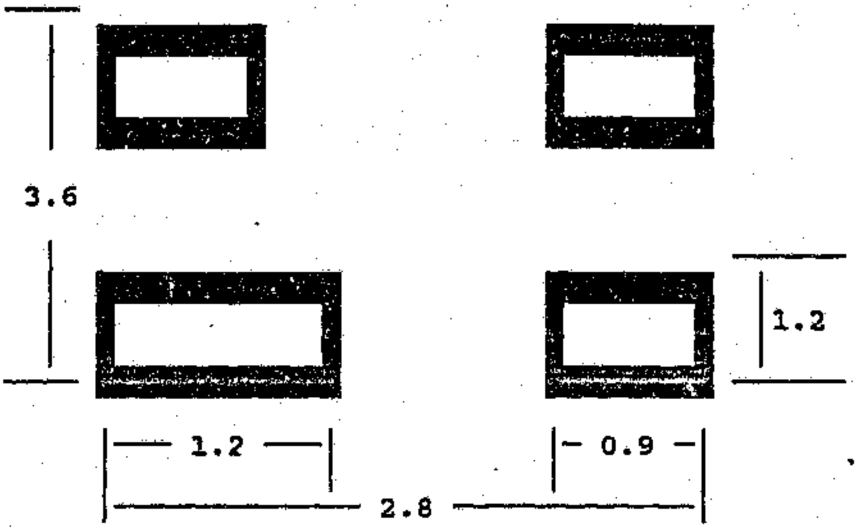


Fig. A3.6: Land pattern for SOT-143 package (dimensions in mm).

A3.3.3 SOIC package land patterns

The land pattern geometry for SOIC components is shown in Figure A3.7 and can be calculated from the following formulae:

$X = 0{,}60$ mm (0,024 in)

$Y = 1{,}80$ mm (0,070 in)

$G = A_{min}$

where
X = land width
Y = land length
G = gap between the two rows of lands
A = component body width

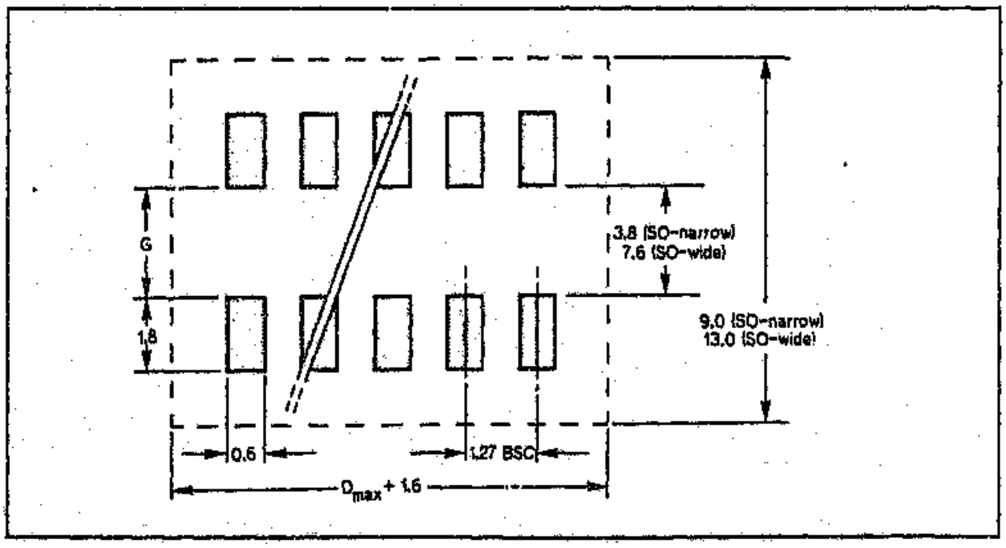


Figure A3.7 Footprints for SOICs.

A3.3.4 PLCC package land patterns

Land patterns for PLCC components can be calculated from the following formulae and are shown in Figure A3.8:

X = 0,60 mm (0,024 in)

Y = 1,80 mm (0,070 in)

$W = E_{max} + K$

where X = land width
Y = land length
W = distance between outer edges of opposing lands
E = distance between outer edges of opposing component leads
K = 0,75 mm (0,030 in)

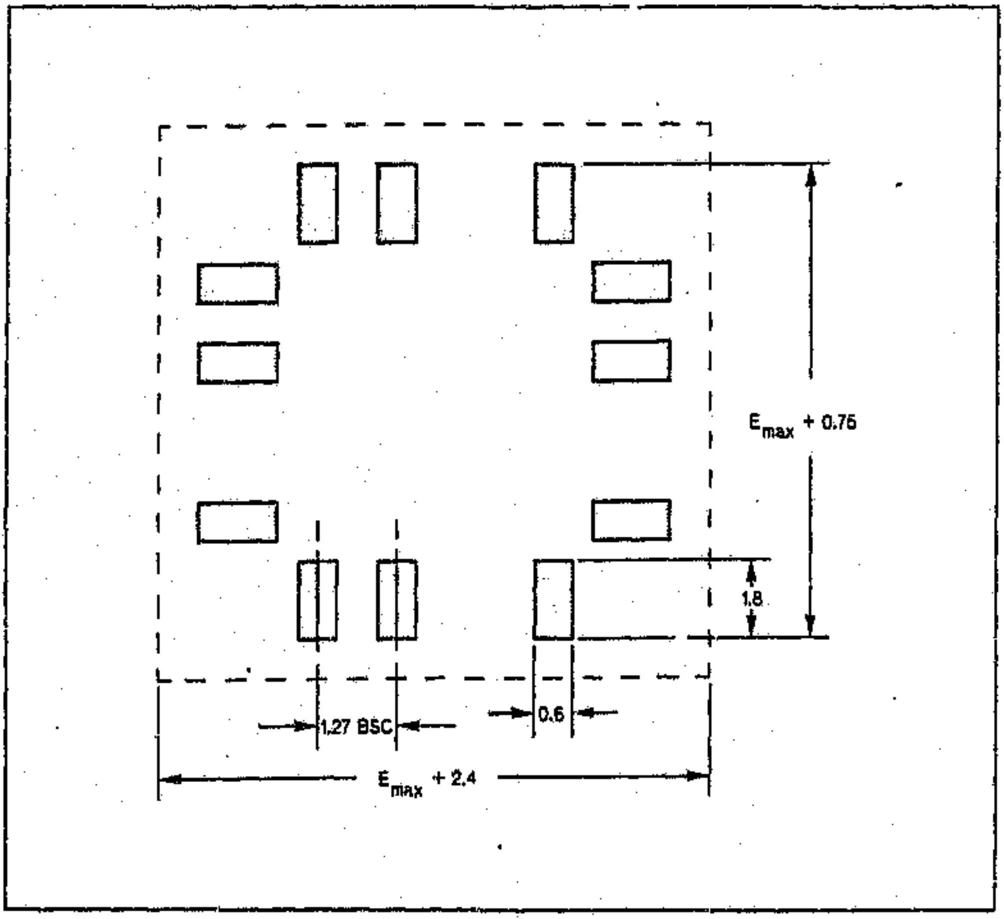


Figure A3.8 Footprints for PLCCs.

A3.3.5 LCCC package land patterns

Land patterns for 0,050-in lead pitch LCCCs are the same as for PLCC packages. Since an LCCC is about 1,0-mm (0,040-in) smaller than a corresponding PLCC, the value for K is larger. When using LCCC outer dimensions, we use the value K = 1,75 mm (0,070 in).

A3.3.6 Gull-wing quadpacks

Land patterns for gull-wing quadpacks can generally be found from the following formulae and are shown in Figure A3.9:

$$X = B_{max} + S$$

$$Y = F_{max} + K$$

$$W = E_{max} + 2K$$

where
X = land width
Y = land length
W = distance between outer edges of opposing lands
B = component lead width
F = length of component foot
E = distance between outer edges of opposing component leads
K = 0,4 mm (0,016 in)
S = 0,1 mm (0,004 in)

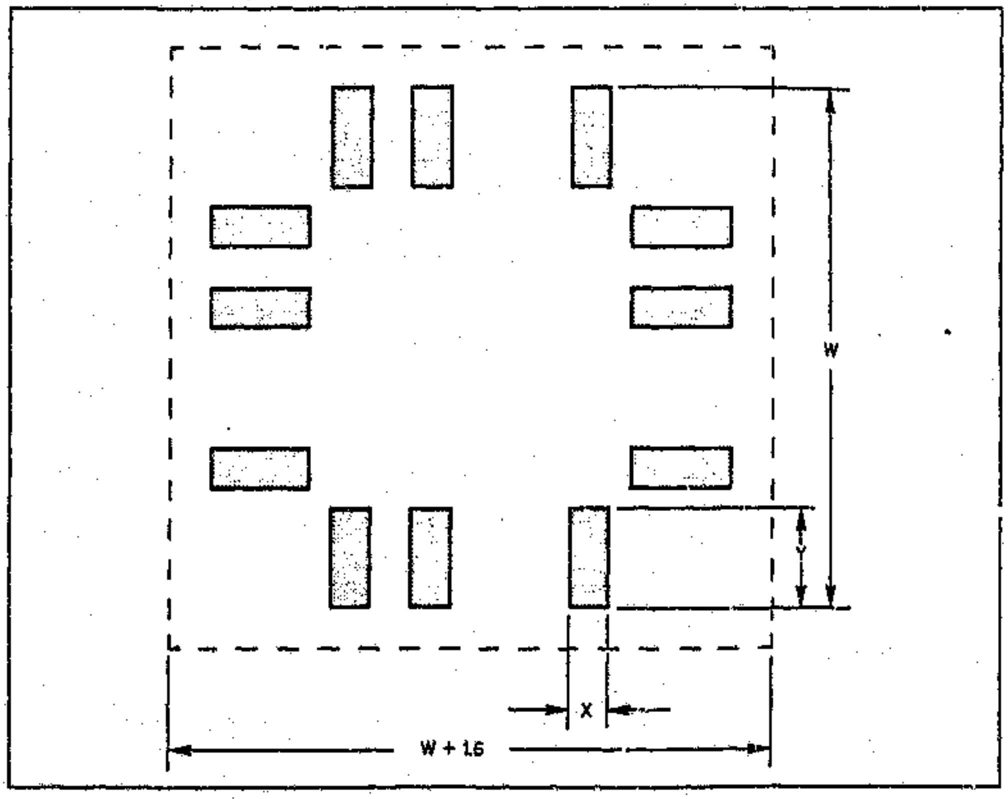


Figure A3.9 Generalized land pattern relationships for gull-wing quadpacks.

A4 PLACEMENT EQUIPMENT CHARACTERISTICS

The following is a detailed discussion of the three characteristics of primary importance in component placement equipment, namely:

- accuracy
- speed
- flexibility

A4.1 Accuracy

Many definitions of machine accuracy have been used by equipment manufacturers. When comparing equipment it is essential that the definition used in the product data sheet be fully understood. Some of the terms in common use are:

- Placement accuracy.
- Resolution.
- Repeatability.

A4.1.1 Placement accuracy

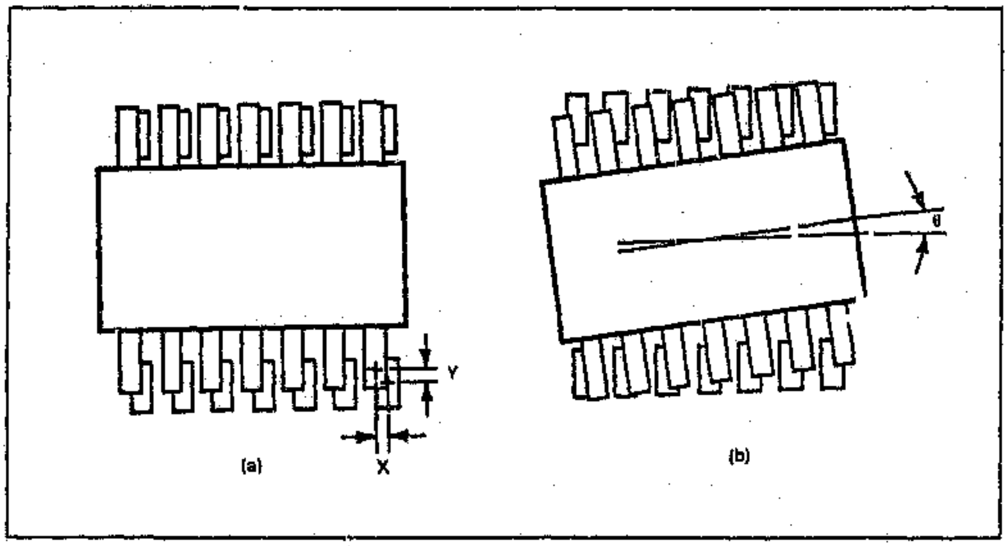


Figure A4.1 Placement accuracy error components:
a. Translational error, b: Rotational error.

The term "placement accuracy" describes how accurately a component can be positioned with respect to its target location on the board. It is defined as the position error of the component termination whose deviation from target position is largest. It consists of the two error components in Figure A4.1:

1] Translation error (misalignment of component centroid).

2] Rotational error (angular displacement of component axes).

Translation error

Translation error primarily results from inaccuracies in the X-Y positioning system and includes offset, scaling, and axis orthogonality errors. The component centering mechanism is also a factor if it does not accurately align the component centroid to the placement of tool axis.

Ideally, translational error should be specified as a "True Position Radius" [TPR] about the design centre. Many equipment manufacturers however, specify simple X-Y tolerances. Although this gives the appearance of an improved specification, the greatest deviation from design location can be greater than the implied specification by as much as a factor of 1.4 .

If only X and Y tolerances are specified, TPR error can be found from the equation:

$$T = \sqrt{X_t^2 + Y_t^2}$$

where T = true position radius error due to translation errors.

X_t = error component along X-axis.

Y_t = error component along Y-axis.

Rotational error

Rotational error results from inaccuracy in the component centering mechanism and from the rotational precision of the placement tool. It is specified as an angular tolerance about the target placement orientation.

Rotational error is greatest on those terminations farthest from the component centroid. To simplify the analysis, this error is commonly approximated by calculating the displacement of the package corner. Referring to Figure A4.2, displacement can be found from the equation:

$$R = 2L \sin (\pm/2)$$

where R = true position displacement due to rotational error.

L = distance from centre of component to package corner.

θ = maximum angular deviation from target orientation.

Calculating the components of rotational error along the X and Y axis:-

$$X_r = 2L\sin(\theta/2)\sin\phi$$
$$Y_r = 2L\sin(\theta/2)\cos\phi$$

where X_r = X-axis error due to rotational error.

Y_r = Y-axis error due to rotational error.

ϕ = angle from the centre of the component to the lead, referenced to the component X-axis per Figure A4.2.

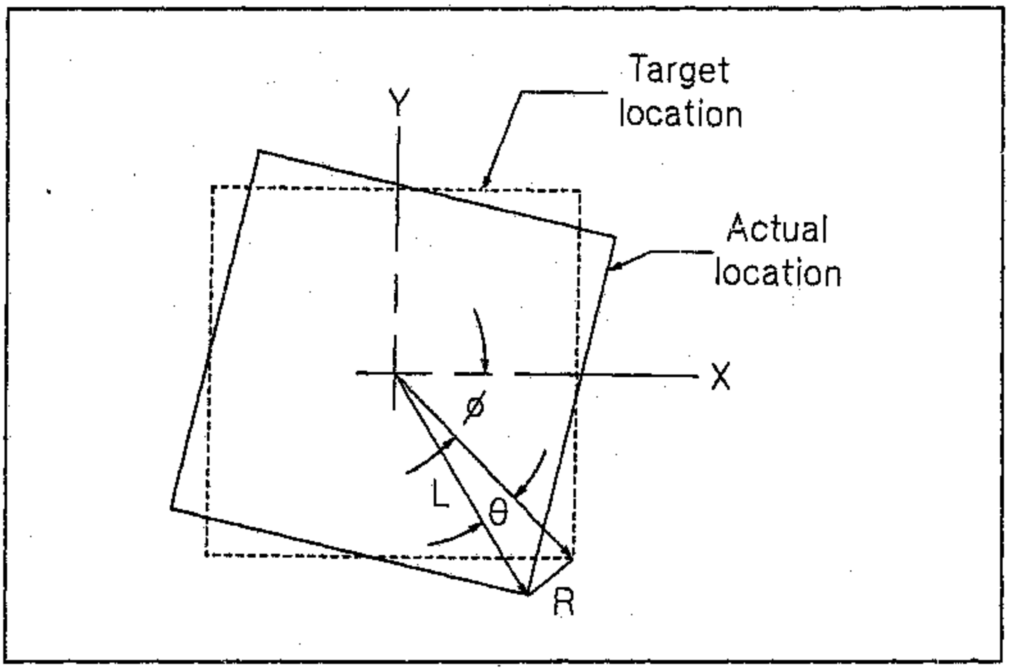


Figure A4.2 Package corner displacement due to rotational error.

Total error

Rotational error combines with translation error to produce a cumulative effect. Total TPR error is found by the vector addition of the two components. The X-axis and Y-axis error are found from:

$$T_x = X_t + X_r$$

$$T_y = Y_t + Y_r$$

where T_x = X-axis component of total error.

T_y = Y-axis component of total error.

Total error is then found from:

$$TPR = \sqrt{T_x^2 + T_y^2}$$

Since the effect of rotational error depends on component size, it is not possible to define a single number to represent overall machine performance. Instead, translational error and rotational error must be specified separately. Knowing the types of components to be placed, overall placement accuracy can be calculated from these two numbers.

Placement accuracy requirements

The accuracy requirement for a component is a function of such factors as component type, design criteria, and reliability requirements. The factors of greatest importance are:

* COMPONENT TYPES

The effects of rotational errors are seldom significant for small chip components but can become the dominant error source for the corner leads of large PLGCCs or ceramic chip carriers. As a result, equipment designed to place integrated circuits must be more accurate than equipment that places only chip components.

* ACCEPTABLE LEAD-TO-LAND MISALIGNMENT

The extent to which the component termination is allowed to overhang the edge of the land has a major impact on required equipment accuracy. Some specifications define that a component is acceptably located if no more than one-half of the termination width extends beyond the circuit land. Others restrict this to one quarter of the width, and still others do not allow any of the termination to extend off the pad. Acceptance criteria are driven exclusively by product requirements - a high-reliability life-support system would permit less deviation than an inexpensive consumer product.

* LAND PATTERN DESIGN

One way to reduce overhang is to widen the lands so that greater misalignment can occur before exceeding the lead-to-land registration specification.

However, this approach can only be a partial solution. Lands that are too wide can cause increased soldering defects due to component skewing and tombstoning.

Larger solder fillets also have lower ductility and degrade the reliability of noncompliant joints. Finally, large circuit lands reduce the amount of space available for conductor routing on the board. Generally, lands should be kept small enough to permit at least one trace to be routed between IC leads and under chip components.

Large lead-count integrated circuits demand the highest placement accuracy. Since translational and rotational errors combine to produce the total error, some tradeoff is possible. For a given component type, a machine with small rotational inaccuracy can tolerate greater translational error than one with larger rotational inaccuracy.

A4.1.2 Resolution

Any real machine has a finite ability to resolve successive points in space. Machine resolution is fixed by the resolution of stepping motors and the rotary or linear encoders on the axis drive mechanisms. When the axes are programmed to travel to a particular point, they will actually go to the nearest point capable of being resolved. This can result in a position error (called "quantization error") or up to one-half of the machine resolution.

While resolution defines the ultimate precision of the machine, it does not necessarily have a direct relationship of total placement accuracy. Resolution is simply a measure of the finest increments the machine can move. Accuracy includes all other error contributions, some of which can be far larger than quantization error. In fact, it is possible for one machine with high resolution to have worse total accuracy than a different machine with lower resolution.

Although resolution can be important in certain special cases, it is much less useful as an overall measure of machine performance. It should never be the sole specification of machine accuracy.

A4.1.3 Repeatability

This term describes the ability of the placement tool to repeatedly return to a target point. Bidirectional repeatability is defined by the National Machine Tool Builders' Association as the "expected dispersion from the mean resulting when the approach to any given point is programmed from both directions in a series of trials".

It is normally specified as a 3-signal limit, as shown in Figure A4.3.

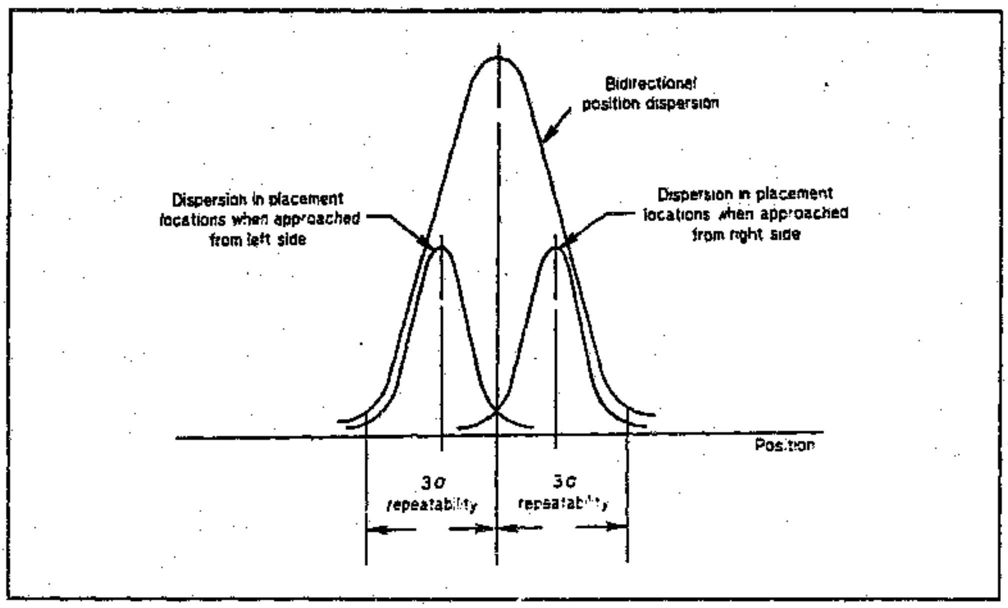


Figure A4.3 Definition of bidirectional repeatability.

The relationship between repeatability, resolution, and placement accuracy should be clearly understood. Standard practice is to include repeatability within the placement accuracy specification, as shown in Figure A4.4.

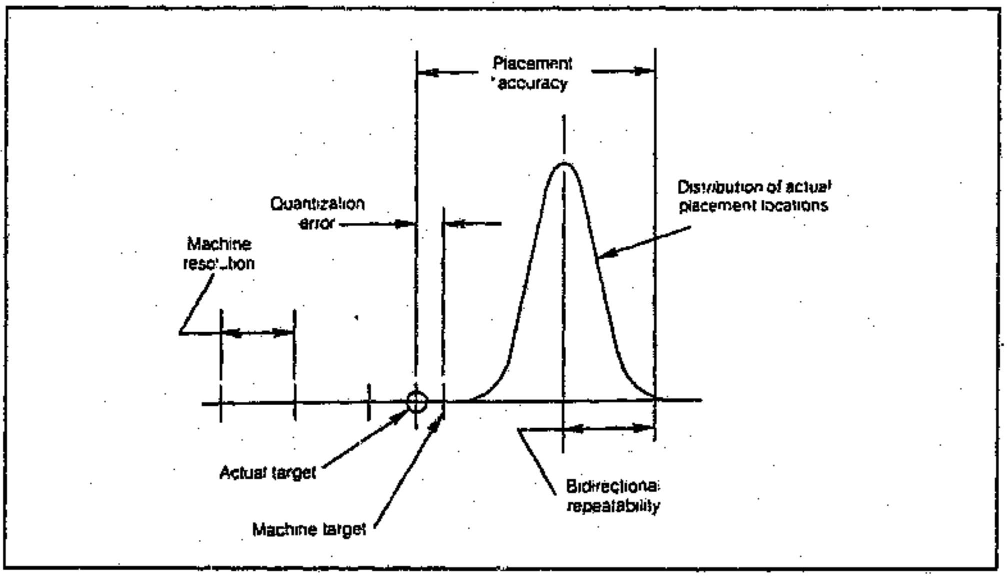


Figure A4.4 Relationship between various accuracy-related terms.

For example, a machine with an accuracy specification of 0,1mm true position radius and repeatability of 0,05mm true position radius can be expected to place the component termination within 0,1mm of the design location 99,7% of the time (if specified to a 3-sigma limit). Over a large number of placements to the same location, the deviation of any single placement will not exceed 0,5mm of the centre of the distribution. Not every machine manufacturer uses this convention, so it is wise to review published specifications carefully.

A4.2 Speed

In most surface mount factories, the component-placement operation is the slowest element in the process. The speed of the placement machine is thus the limiting factor in overall line capacity. Since faster machines do not incur proportionally higher overhead expenses, they are capable of achieving lower factory operating costs.

Unfortunately, comparing the speed of one machine to that of another similar machine is not an easy task. No standard methods have yet been developed. Data published by equipment manufacturers tends to present their own machines in the most favourable light. Some users have been unpleasantly surprised when they discovered, after purchasing a specific machine, that actual production throughputs were considerably less than projections based on data-sheet information.

The problem arises because actual speeds are heavily dependent on a number of factors outside the control of the equipment manufacturer. Board design, number and location of component feeders, production lot sizes, setup complexity, and the efficiency of the board loading program all heavily influence the volume of product that can be produced in a given day. Equipment specifications can only represent the performance achieved on some "average" board that may have little relation to actual customer designs. Few manufacturers adequately define how published data was taken or how users can extrapolate that information to their own boards.

Several of the more commonly used definitions of machine speed are presented below:

A4.2.1 Equipment placement rate

This is the most common data-sheet specification, defined as the speed at which an average placement cycle is completed excluding any external factors. A placement cycle consists of a complete round trip from pickup site to placement site and return, during which a component is actually placed.

On machines with multiple heads it is defined as the effective rate of placement when all heads are included in the calculation. The amount of head travel permitted during the cycle is restricted, either by specifying a maximum distance of travel, a maximum distance between successive feeders, or both. A typical specification reads as follows:

> Placement rate is measured as cycles from pickup point to placement point and return. Feeder travel is not to exceed 12 locations between consecutive 8mm tape feeders. Under these conditions, placement rate is at least 4500 components per hour.

A4.2.2 Cycle rate

This is the most basic measure of machine speed. It is similar to placement rate except that the machine is operated in a dry cycle mode in which no parts are actually placed. This number thus excludes the effects of component pickup, centering, and placement, and so is somewhat higher than the equivalent placement rate. Although cycle rate can be useful in comparing machines, it should be used carefully. Unless information is provided describing how to derate this number for actual component placement, it cannot be used to predict production throughput. Some manufacturers use the term test rate instead of cycle rate.

A4.2.3 Production throughput

This is the most important parameter to the equipment user. It is defined as the number of components placed per hour over an entire production shift. For example, if a certain machine places 50,000 components during an 8-hour day, the hourly production throughput is 50,000/8 = 6,250 components per hour.

Production throughput is derived by applying a number of derating factors to the placement cycle rate. The most significant are:

Board load/unload time

Before the placement sequence can begin, the board must be loaded onto the machine; upon completion, it must be removed. This loading and unloading period is "dead" time during which the machine cannot place components. Board load/unload time can vary from as little as 5 or 10 seconds for an automated system to a minute or more if boards must be loaded manually.

Production mix

In factories that must handle a large number of board types, the number of different components might exceed the machine's feeder capacity. It is then necessary to periodically stop production to change the feeder setup. For boards of different physical sizes, the board support system must also be readjusted. These factors contribute dead time that is beyond the control of the equipment manufacturer.

Machine configuration

The feeders on sequential placement machines are generally arranged in rows at the sides of the machine. The time required to pick a component depends on the distance the head must travel to access the feeder. It may take twice as long to access feeders near the end of the row compared to those near the center.

It is usually possible to optimize feeder placement for any particular board, but if several board types are built with a fixed setup, speed will inevitably be compromised.

Component mix

Machine cycle time is a function of component type. Integrated circuits must be positioned more precisely than small chip components, so a board with large numbers of ICs will generally take longer to populate than a board with a similar number of resistors and capacitors.

Available hours

In most cases, an eight-hour shift does not represent eight full hours of production. Lunch periods, coffee breaks, and miscellaneous other time off can reduce the actual time available for production to as little as six or seven hours. Factories with highly automated equipment that can be left running during these periods are less sensitive than factories that depend on the presence of operators.

Unscheduled downtime

This is potentially the most significant factor but the one least frequently discussed.
Unscheduled downtime can result from such varied factors as poorly designed boards, components that do not meet specification, or poor placement-machine design. It is virtually certain to occur at least occasionally, so the machine should be designed to facilitate problem diagnosis and repair.

Because production throughput depends on many factors beyond the control of the equipment manufacturer, it cannot be specified on equipment data sheets. The ideal data sheet would list equipment placement rate and provide derating factors that can be used to estimate production throughput. Although rare, some manufacturers do provide this derating data. Table 4.1 lists the factors that must be considered for one such machine.

A4.3 Flexibility

Flexibility measures the ability of a machine to accommodate varying placement requirements. The factors that contribute to machine flexibility include:

- Component variety
- Number of feeders
- Ease of setup

I. MINIMUM CYCLE TIME

0,600sec (this yields a maximum throughput of 6000 components per hour).

II MACHINE CYCLE ADDED FACTORS

a. Component Spacing

Up to 1 in: no increase in run time
Over 1 in: add 0,056sec/in

b. Feeder Spacing

Up to 2.5 in: no increase in run time
Over 2.5 in: add 0,170sec/in

c. Rotation

Up to 90° : no increase in run time
Over 90° : add 0,002sec/degree

d. Tweezering

Normal : no increase in run time
Alternate : add 0,460sec

III COMPLETE RUN ADDED FACTORS

a. Tool Change

Add 2,800sec PLUS:
Add 0,170sec/in in excess of 2,5 in of feeder movement

b. Manual Load/Unload

Operator dependent : estimate 40 seconds

c. Automatic Feed

Add 15 seconds

IV OVERALL RUN TIME

Calculated by taking a minimum cycle time times number of machine cycles PLUS the largest time from Section II for each machine cycle PLUS each addition from Section III.

TABLE 4.1 Placement cycle derating factors for the Dynapert/Precima MPS-500 sequential placement machine.

A4.3.1 Component variety

A machine that can place a wide variety of components is more versatile than one that accepts only small chip devices. Although the fundamental limit on component variety is set by machine placement accuracy, this is not the only influence. The placement tool and centering mechanism must be compatible with the components, and appropriate component feeders must be available.

Most placement tools can accommodate only a limited range of component sizes. Increased flexibility is obtained through the use of two or more interchangeable tools. Ordinarily, both the top tip and centering jaws (if present) are changed as a unit. To be most effective, the machine should be able to change tools automatically under program control.

The most common component feeders include tape-and-reel, magazine, bulk, and waffle tray. Some equipment, especially high-speed machines, can accept only limited feeder styles. Most versatile equipment is able to handle all or most types of feeders, generally at some sacrifice in speed.

A4.3.2 Number of feeders

Changing feeders on a placement machine is a slow process that increases cost and reduces production throughput. On small machines with limited feeder capacity, it might be necessary to reconfigure the machine with every change in board type. Larger machines require less setup. When feeder capacity increases to about 120-150 feeders, it might be possible to permanently store all components on line. No time is then consumed in feeder setup when changing from one production run to another.

The feeder capacity of a machine is usually expressed in terms of the maximum number of 8-mm tape feeders that can be mounted onto the machine. Some manufacturers instead specify the number of inches of feeder space. This can be converted to the equivalent number of 8-mm tape feeders by knowing the width of the feeder. This is often 1 in (25mm) wide, although some machines house two tapes on one feeder.

Not all parts can be packaged in 8-mm tape, so the actual capacity of the machine will always vary from the specification. Feeders for wider tapes consume more space and diminish the total capacity. Magazine feeders consume less space and diminish the total capacity. However, magazine feeders hold far fewer parts than tapes and must be replenished more often. Feeder designs vary by manufacturer, so it is not possible to define space requirements in a general way.

With some machines, feeders can be positioned only at fixed locations, typically in 1-in increments. In this case, a feeder that is 1.5in wide will actually consume two full inches of feeder space. Other machines permit feeder compacting. This makes it possible to compress all feeders together and increase the effective capacity of the machine. For this feature to be realizable, the equipment firmware must be capable of accessing any point along the feeder axis.

It is important to compare similar numbers when studying the capabilities of several machines. Some manufacturers have been known to emphasize the absolute maximum feeder capacity in their data sheets. This number, obtainable only when using magazine feeders exclusively, can be triple the 8-mm tape feeder capacity.

A4.3.3 Ease of setup

The steps necessary to change from building one type of board to another combine to determine machine setup time. Elements that contribute to setup time can include any or all of the following:

- Machine reprogramming
- Feeder changeover
- Board support system adjustment
- Placement head adjustment/changeover

Machine reprogramming

Programming the placement machine to load a new board can be as simple as downloading a file from a computer or as complex as manually teaching the machine by stepping it through the desired placement sequence. Simple machines generally use teach-mode programming, while more sophisticated equipment offers several options.

Most production-grade machines now use magnetic floppy disks for program storage and have the ability to receive programs from an external computer system. A few machines still use magnetic tape as the program storage medium, but this is rapidly becoming obsolete. In either case, these machines are generally programmed in an off-line mode. Design coordinates for each component are entered into a separate computer, from which the actual placement program is generated. This approach is fast and does not take the machine out of production during the programming process. With boards designed on a CAD system, automatic program generation is possible by downloading the CAD data directly to the program-generating computer.

Lower-volume machines for prototyping and small-volume production are usually less flexible. Placement programs often must be generated by manually stepping the machine through the placement sequence. Once programmed, the data can usually be stored electronically for future recall. This teach-mode programming is more time-consuming and less precise than off-line programming. However, it has the advantage of inherently compensating for any systematic inaccuracy in the placement machine. (The machine accuracy on the least expensive equipment is frequently very poor. Accurate placement cannot be guaranteed if design coordinates are used to determine placement location).

Feeder changeover

Unless all components can be stored on line, it will be necessary to change at least some of the component feeders when setting up for a new production run. Most equipment manufacturers have given special consideration to this problem and have attempted to reduce the time required for setup.

The most common solution is the "quick-release" feeder, which allows single feeders to be inserted and removed by simply operating a lever. An even faster method involves changing entire banks of feeders in one operation. Using this approach, separate feeder banks can be maintained for each product or group of products to be built on the line. Changeover then becomes a simple task of removing one feeder bank and installing another.

Board support system adjustment

If the new board is smaller or larger than the machine can currently accommodate, the board support system must be adjusted. On automated machines, boards are transported on conveyor rails that contact only the edges of the board. Rail width is adjusted either manually or automatically under program control. Less expensive machines frequently make use of a dedicated tooling plate. In this case, setup consists of removing the old plate and installing a new one.

Setup can be eliminated by using workpiece holders to transport boards of all sizes. However, the benefit obtained by this technique must be balanced against the added tooling cost incurred in the holders and the extra labour necessary to install and remove boards from holders.

Placement head adjustment/changeover

The placement head must be changed or adjusted if it is not currently able to handle all components in the new production run. In many machines, this happens automatically under program control, but less expensive machines require manual adjustment.

Appendix 5. PLACEMENT EQUIPMENT CLASSIFICATION

A5.1 Sequential Placement

Most surface mount machines employ a sequential placement technique. Individual components are picked from feeders and placed successively by the placement head. Unlike through-hole technology, sequencing and placement are performed on the same equipment, so no separate component sequencer (or sequenced tape) is required.

Equipment is usually classified into one of four categories, depening on its intended position in the marketplace:

* Entry-level

* General-purpose

* High-speed

* Precision

A5.1.1 Entry-level

This category consists of small, relatively inexpensive equipment with limited capability. Machines in this category are primarily intended for prototype or very low volume production use. They have limited feeder capacities, small maximum board sizes, and relatively low placement rates. They are not usually designed for the heavy, continuous usage as would occur on a volume production floor.

A5.1.2 General-purpose

Placement systems suitable for continuous production usage can cost hundreds of thousands of rands. With such a significant investment, there is strong incentive to purchase a single machine capable of meeting all placement needs. General-purpose machines are targeted for this market. They are highly flexible with moderate production throughput. The majority of all installed systems fall into this classification.

Machines designed for general-purpose applications often employ a single placement tool that accesses a linear row of component feeders. To increase feeder capacity, feeders can be mounted in two rows - one at the front and the other at the rear of the machine. Speed can be increased by mounting two tools on the head, one to pick components from the front row and the other to pick from the back row. This allows one tool to be picking a component while the other is placing, cutting cycle time approximately in half.

Typical general-purpose equipment places 3000 to 6000 components per hour. Parts are placed sequentially using a programmable placement head. Feeder capacity is at least one hundred 8-mm tape feeders and exceeds two hundred in some designs. The range of compatible components extends from small chip capacitors to large plastic leaded chip carriers in tape, magazine, or bulk packages. Some machines also accept matrix trays and bare semiconductor die. Accuracy is sufficient to permit placement of 0,050-in lead pitch devices to at least 84 leads, and finer pitch devices can frequently be accommodated. The typical maximum board size exceeds 300 x 450mm (12 x 18 inches), and some equipment is able to handle 450 x 600mm (18 x 24 inches) boards.

This high flexibility invariably leads to compromises. General-purpose machines are slower than equipment optimized for a more limited range of components. They are also less accurate than machines designed for precision placement. These compromises, however, are what make this class of equipment so useful. Although not providing exceptional performance in any single area, general-purpose machines offer a range of performance that is often sufficient to satisfy the entire demand of a given factory.

A5.1.3 High-speed

This category consists of machines similar to general-purpose equipment but optimized for rapid placement of fewer component types. Because the earliest machines were only able to place small passive devices fed from 8-mm tape, they have commonly been called "chip shooters". More recent products can also accommodate larger components in wider tape sizes, but still cannot place the full range of components handled by general-purpose equipment.

Typical high-speed machines are capable of sequentially placing components at rates between 9000 and 24,000 per hour. Feeder capacities extend from under sixty to over one hundred twenty 8-mm tapes. Nearly all machines in this category accept only tape-fed parts. One exception is equipment for placing cylindrical MELF resistors, these parts are loaded onto the machine from bulk.

A5.1.4 Precision

High lead-count integrated circuits, especially those with lead pitches below 0,050in, present special challenges during placement. Because of their large bodies and closely spaced leads, they must be placed extremely accurately. Moreover, their fragile leads are easily damaged and are not compatible with mechanical centering tools. general-purpose machines are not usually adequate for these complex parts.

A relatively new class of equipment has emerged to address this more specialized need. Precision placement equipment is designed to place complex parts to accuracies of ±0,05 to ±0,1mm at rates of 500 to 1000 components per hour. Parts are fed from magazines, waffle packs, or in some instances, tape-and-reel packaging. Several design approaches exist, but certain similarities are usually present. Vision systems are essential, both for board alignment and optical component centering. Robotic arms are frequently used to achieve the required level of accuracy.

A6 COMPONENT FEEDING SYSTEMS

The following is a detailed discussion of the following component feeding systems:

- tape and reel
- magazine (stick)
- bulk
- matrix tray.

A6.1 Tape-And-Reel Feeding

For most component types, tape-and-reel containers are preferred. Tape feeders are extremely reliable, and large quantities of components can be held in a single reel. Tape packaging also provides individual protection for each component. The main drawback is the somewhat higher packaging cost over other formats. This is usually more than offset by the improved feeder reliability during placement.

A6.1.1 Tape-and-reel specifications

A typical component tape is shown in **Figure A6.1**. It consists of two parts: a carrier tape that holds the components and a cover tape that prevents them from falling out of the carrier. The carrier tape can be made from any of several materials. Embossed tapes made from plastic or aluminium are most common. Punched paper or plastic is also used, principally in the smaller tape widths. The cover tape is usually a transparent polyester that has been glued or seam welded to the carrier tape. Opaque covers are sometimes used, but this makes visual inspection of the parts more difficult.

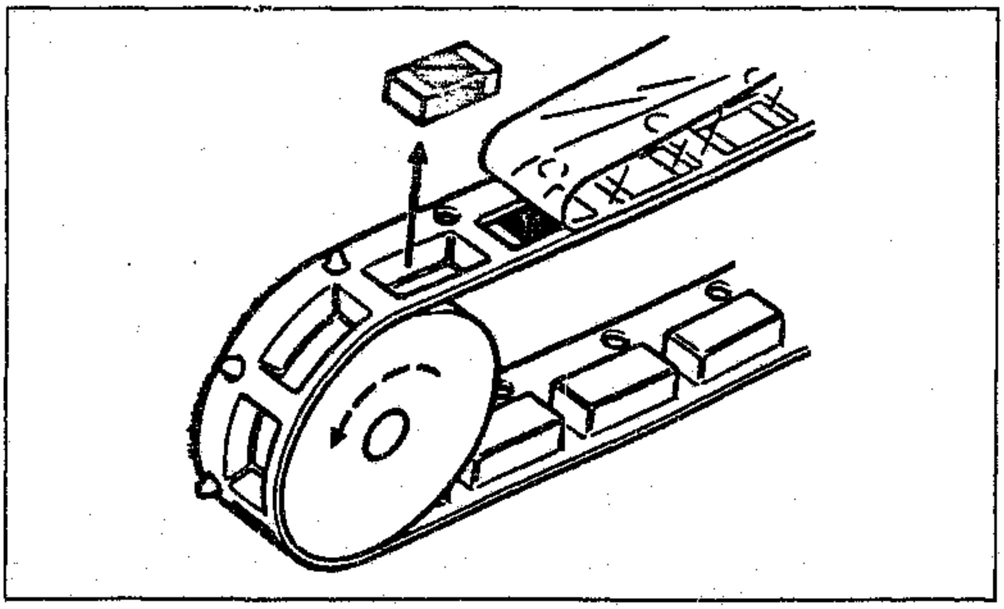


Figure A6.1 Typical component tape.

Antistatic or electrically conductive materials to prevent electrostatic damage are gaining popularity for both carrier and cover tapes.

Both EIA and IEC have prepared standards for component tapes and reels. In most areas, these document are in agreement, but a few specifications are different. Most notable is the range of allowable variations in cover tape peel strength The EIA specification permits peel strength to vary from 0,1 to 0,7 Newtons (10 to 70 grams). The IEC document permits an even greater variation, from 0,2 to 1,3 Newtons (20 to 130 grams). The wide latitude allowed by either standard has made feeder design extremely difficult for equipment manufacturers.

A6.1.2 Tape feeders

A typical tape feeder is illustrated in Figure A6.2. The feeder must accurately index the part for pickup while simultaneously removing the cover tape. It must operate smoothly at all times so that parts are not jarred out of their sockets prior to being picked by the placement tool.

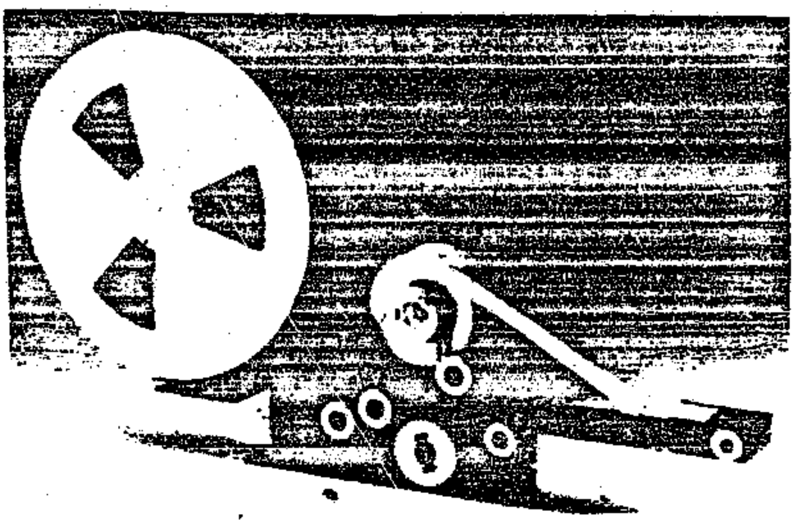


Figure A6.2 Typical tape feeder.

No feeder is totally immune to vibration, and the longer the part is exposed, the more chance that it will vibrate out of position. Therefore, feeders that do not remove the cover tape until the final indexing step are preferred over those that expose several components at a time. Some feeders even employ a shutter that further covers the exposed part, retracting only after the indexing step is complete and the tool is ready to pick the part.

A6.2 Magazine Feeding

Magazines, also called sticks or tubes, are frequently used for integrated circuits and other large parts Figure A6.3. Typical magazines consist of a semitransparent polyvinyl chloride (PVC) extrusion that has been treated with an antistatic coating.

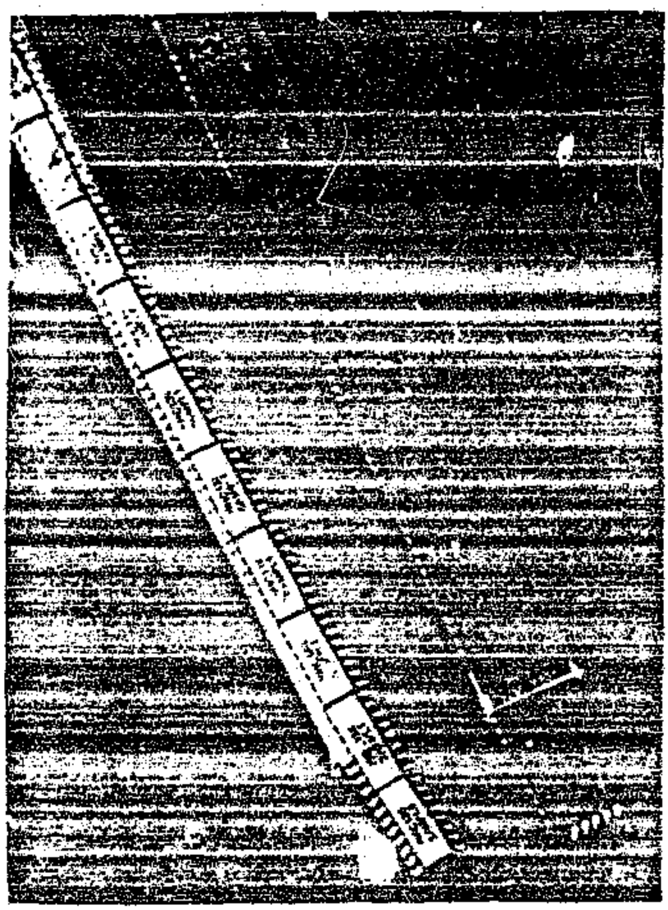


Figure A6.3 Components in magazines.

Magazines are less expensive than tape-and-reel packaging but hold far fewer parts. For larger components where tape-and-reel packaging is not readily available, they are a popular substitute. They are also preferred for lesser-used parts where the tape-and-reel format would represent a sizable inventory. Perhaps their most important advantage is the reduction in feeder space they offer when compared to tape feeders. This allows a significant increase in the maximum feeder capacity of the placement machine. This benefit, however, must be weighed against the disadvantage of having to replace empty feeders much more often.

A6.2.1 Magazine specifications

Wide variations exist in magazines from different manufacturers. This has made feeder design extremely difficult. A specific feeder may work well with magazines from one manufacturer and not at all with those from a second manufacturer. This situation is unlikely to change until widely disseminated international standards can be developed and implemented.

A6.2.2 Magazine feeders

Typical magazine feeders are shown in Figures 10.31 and 10.32. They can hold several independent magazines in separate parallel tracks. Although the magazines are usually held at an angle to take advantage of gravity, feeders do not generally rely strictly on gravity to feed the parts. Most designs incorporate a mechanical vibratory action to insure reliable component movement. The amplitude of this vibration must be adjusted to match the weight of the parts being delivered. If insufficient, the part may not move into position by the time the head is ready to pick. If excessive, the part may vibrate all the way out of the track, causing a feeder jam.

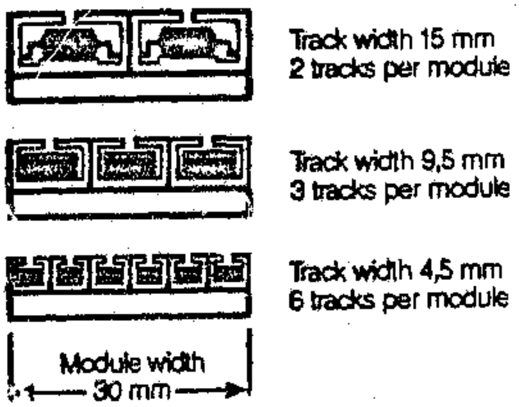


Figure A6.4 Vibratory module for stick magazines.

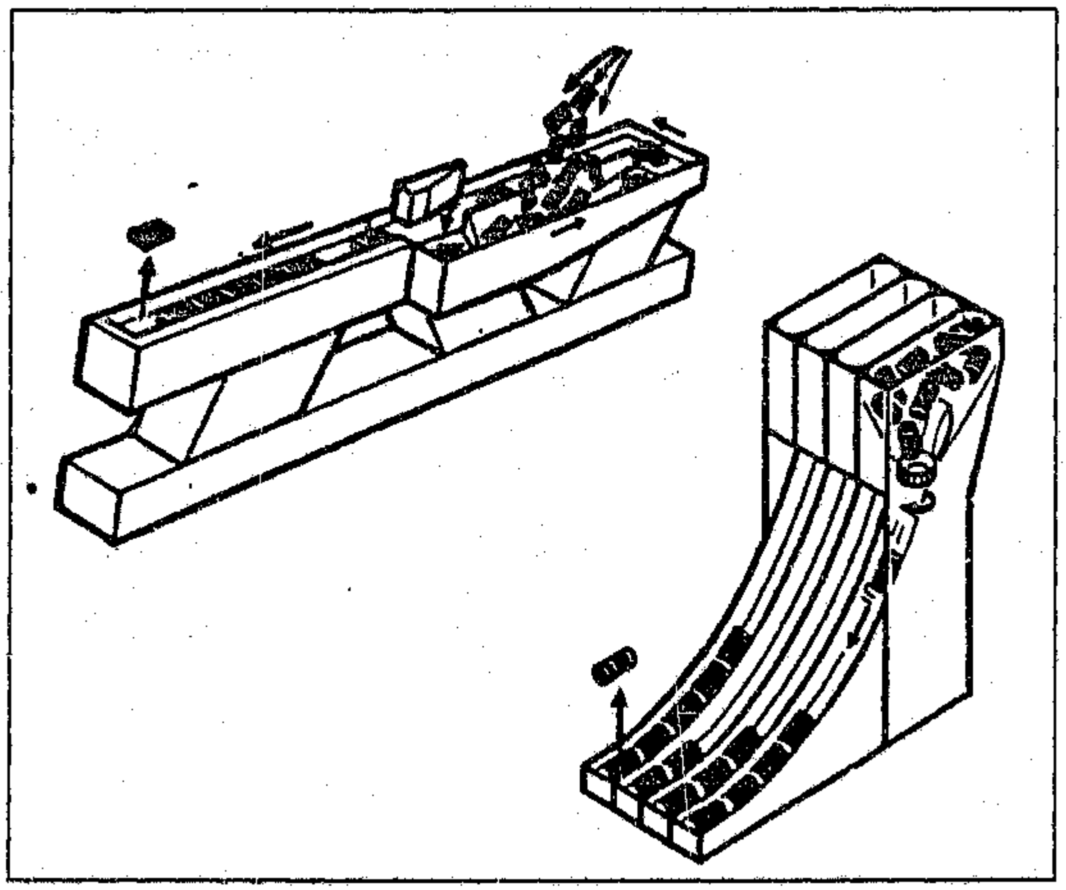
In some designs, the vibratory action operates continuously. In others, it is turned on only long enough to position a new part. The best approach incorporates several features to insure reliable delivery of parts without allowing them to vibrate out of the track. The feeder track should be designed to totally capture all components except the one being picked. A shutter should be employed over this part so that it cannot be accidentally dislodged. Finally, the vibratory action should operate intermittently and be turned off when the shutter is opened for part pickup.

Certain newly introduced magazine feeders employ a conveyor system to move parts into location for pickup. Feeder reliability should be improved because of the more positive action this approach provides.

A6.3 Bulk Feeding

The packaging cost of bulk parts is lower than that of any other format, so there has been considerable interest in this type of feeder. Unfortunately, the low reliability of bulk feeders usually increases assembly costs far beyond the savings realized in packaging.

The typical bulk feeder (See **Figure A6.5a & b**) consists of a linear vibratory track that includes a series of baffles to prevent any part not correctly oriented from reaching the front of the feeder. Rejected parts are automatically dropped back into the parts reservoir and sent through the sequence repeatedly until they finally achieve the correct orientation. Bulk feeders can accommodate rectangular and cylindrical chip devices, as well as various small outline semiconductors. They cannot be used with polarized devices unless the part includes a distinct mechanical feature indicating polarity.



Figures A6.5a and A6.5b Vibratories for cylindrical and cubic bulk.

Bulk feeders suffer problems similar to those of magazine feeders. Vibration amplitude is extremely critical and must be matched to the mass of the part. The optimum setting becomes a compromise between insuring reliable feeding and preventing parts from jumping out of the track. In addition, bulk feeders are sensitive to exact part mechanical dimensions. The baffles must be adjusted to permit correctly oriented parts to pass without accepting those that are incorrectly oriented. Unfortunately, the dimensional tolerances of many chip components are so wide that a single baffle setting will not work across the entire range of the specification. It may therefore be necessary to readjust the baffles for each new lot of parts. Even then the feeder reliability depends on a certain amount of similarity within the lot.

A6.4 Matrix Tray Feeding

Matrix trays have traditionally been employed to hold bare semiconductor die for hybrid assembly. They have also been adapted for use with large quadpacks that are not compatible with other feeding methods. They have not been widely utilized in surface mount assembly but are useful in certain situations.

Unlike all the previous feeders, the matrix tray feeder does not deliver all parts to the same location for pickup. Instead, a grid of indented pockets in the tray holds the parts to be placed, and the placement head must be able to access each pocket. Parts are accessed in a regular pattern, left to right, front to rear. The placement machine firmware keeps track of which part to pick next and when the tray must be replenished with parts.

A7 Fundamentals Of Soldering

A7.1 Particular Demands On The SMD Solder Joint

A7.1.1 Mechanical strength

The solder joint of an SMD establishes both the electrical and the mechanical connection to the board. In contrast to through-hole technology, where the component leads are locked to the board and thus relieve the solder joint from mechanical stress (Figure A7.1), the SMD solder joint also has to hold the component on the board. Both compressive and tensile forces act on the SMD solder joint (Figure A7.2). Apart from cases of extreme acceleration, the tensile strength is usually sufficient. Shear strength may be more critical. Due to different expansion coefficients of board and SMD, temperature changes lead to changes in length and thus create shear forces. These forces have to be kept low in order to avoid destruction of the solder joint. With smaller SMDs the strength of the solder joint becomes less important.

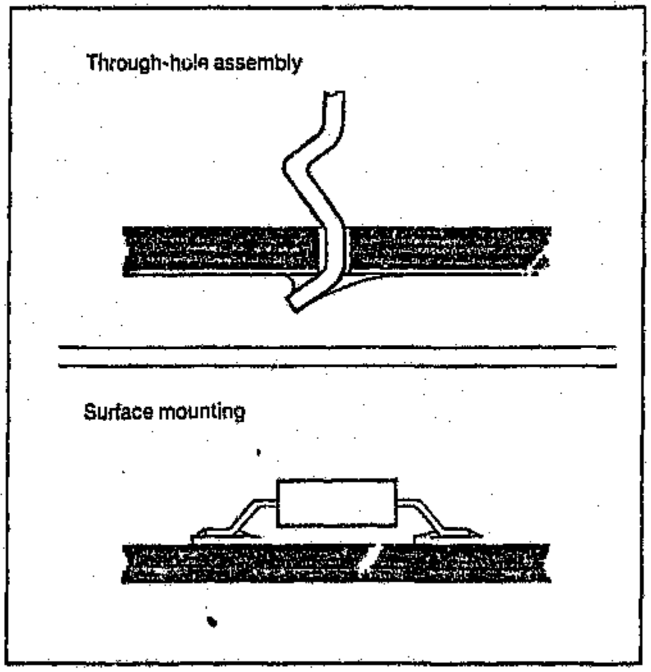


Figure A7.1 Typical solder joint in through-hole assembly and surface mounting.

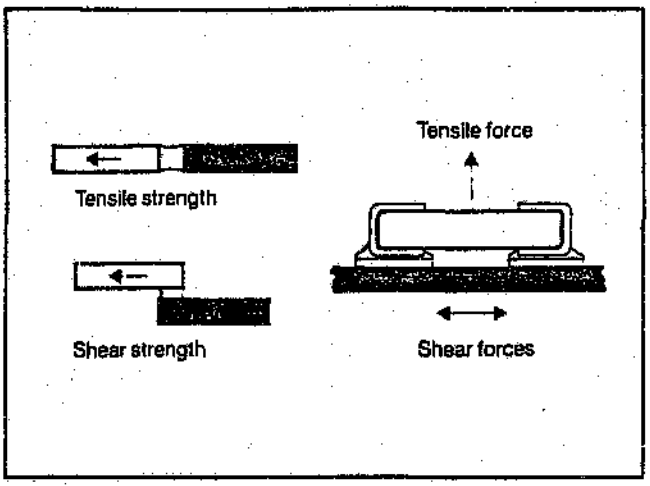


Figure A7.2 Mechanical forces acting on surface mounted components.

A7.1.2 Solders In Use

The most common solders in the production of electronic assemblies are the eutectic solders (L-Sn63Pb or L-Sn60Pb), which have a melting point of approximately 183°C. Solder pastes usually contain a certain percentage of silver, e.g. L-Sn63PbAg. Other solders, e.g. solders with lower or higher melting points, are insignificant. The purity of the solder is far more important than the exact tin/lead ratio (see below).

A7.1.3 Wetting And Surface Tension Of The Solder

When a drop of liquid solder rests on a solderable surface, several forces act on the drop. Cohesive forces tend to minimize the surface of the solder. The surface energies resulting from the cohesive forces would draw the liquid solder into a sphere (surface tension). Adhesive forces, on the other hand, tend to spread the liquid on the solid. In other words, wetting of the solid surface by solder will occur, with the degree depending on the material used. Besides the material, the cleanliness of the surface is a decisive factor for wettability. Contamination, e.g. by oxides, has a detrimental effect on wetting. Generally, fluxes are used to remove contamination.
(See Appendix 9 for Flux Description).

The typical degrees of wetting are illustrated in Figure A7.3.

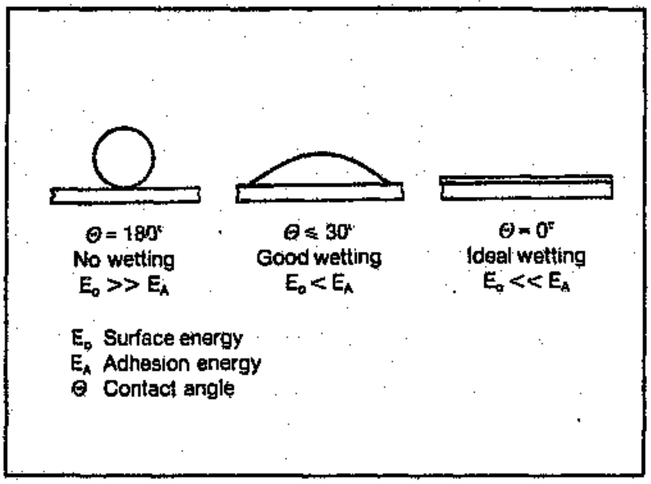


Figure A7.3 Degrees of wetting.

Various contaminants such as oxides may reduce the surface tension of the solder. For this reason, in wave soldering, excessive solder cannot sufficiently recede and forms bridges and icicles.

The surface tension of the solder is not only important for wetting, but also for reflow soldering. Here the surface tension acts on both the solder and the component itself; the positive effect will be self-alignment of the component, the negative effect misalignment.

The surface tension may be influenced by the flux, the soldering atmosphere, and alloy additives. These conditions, in turn, lead to different soldering results.

The surface tension is highly dependent on temperature. The solder always tends to flow in the direction of the higher temperature (Marangoni effect). For conventional soldering techniques the temperature gradients are usually unfavourable, e.g. in case of multi-layer boards, where the solder must rise from the hot solder side to the cooler component side. In SMD technology, reflow soldering enables the board to be heated from both sides, which has a favourable influence on the flow properties of the solder.

A7.1.4 Solderability

It is not enough to carefully specify and maintain the soldering process; strict control must also be exercised over the quality of the incoming material.

The term solderability is used to describe the ability of a component termination to be wet by solder during a specific soldering operation. It is a function of the exact process and materials employed; a termination may exhibit acceptable solderability for some processes but not for others.

The solderability of a surface is characterized by the degree to which it wets and forms a mettalurgical bond with the solder. The degrees of wetting can be classified as follows:

- **Non-Wetting.** In this case, no metallurgical bond is formed and the interface between the solder and the surface remains distinct.

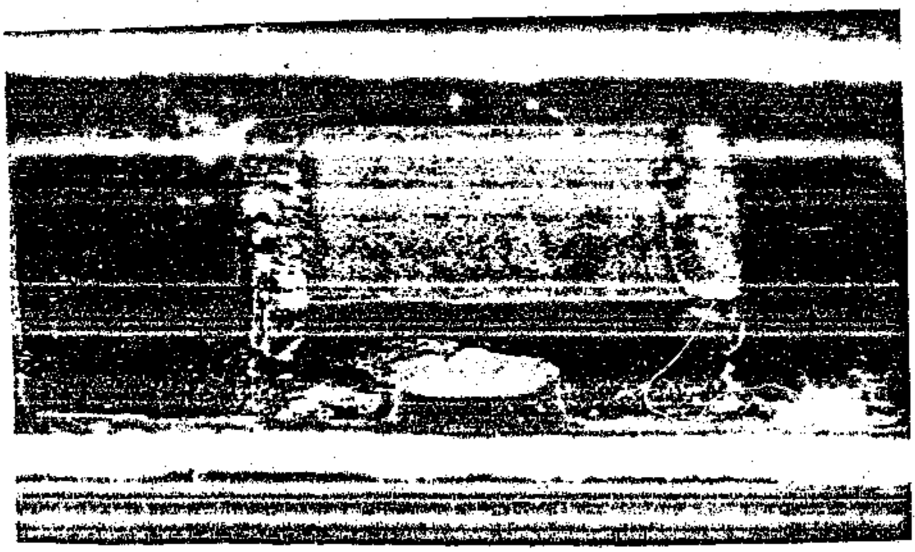
- **Wetting.** The surface energy of a clean mettalic surface is higher than that of molten solder. Under these circumstances, the solder will wet the surface and form a metallurgical bond at the interface. As wetting proceeds, a thin intermettalic layer grows at the interface, forming the basis of a reliable joint.

  Dewetting. The intermetallics that grow at the interface are tin-rich compounds that draw their tin from the tin-lead solder. As tin is consumed from the solder, it leaves behind lead-rich regions with relatively poor solderability. If left at this temperature long enough, the extent of these regions will be sufficient to cause the solder to recede from previously wetted regions, a phenomenon called dewetting.

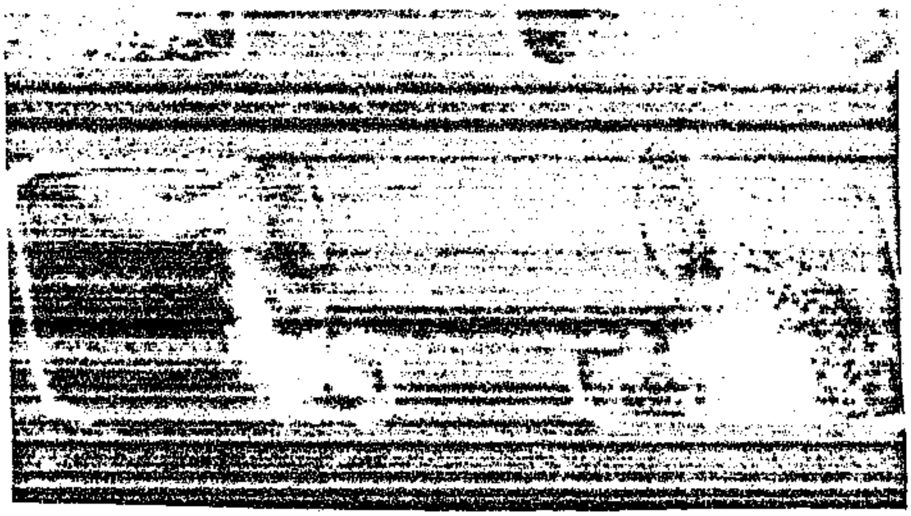
Poor component solderability is a major cause of defects in reflow soldered surface mount assemblies. In the wave soldering process, the joint is washed by an essentially infinite supply of solder. Small amounts of contamination are quickly diluted and carried away. The reflow process does not provide this cleansing operation; contaminants that are present on the component terminations will remain in the completed joint.

APPENDIX 8 SMD Solder Profiles

a) Unacceptable

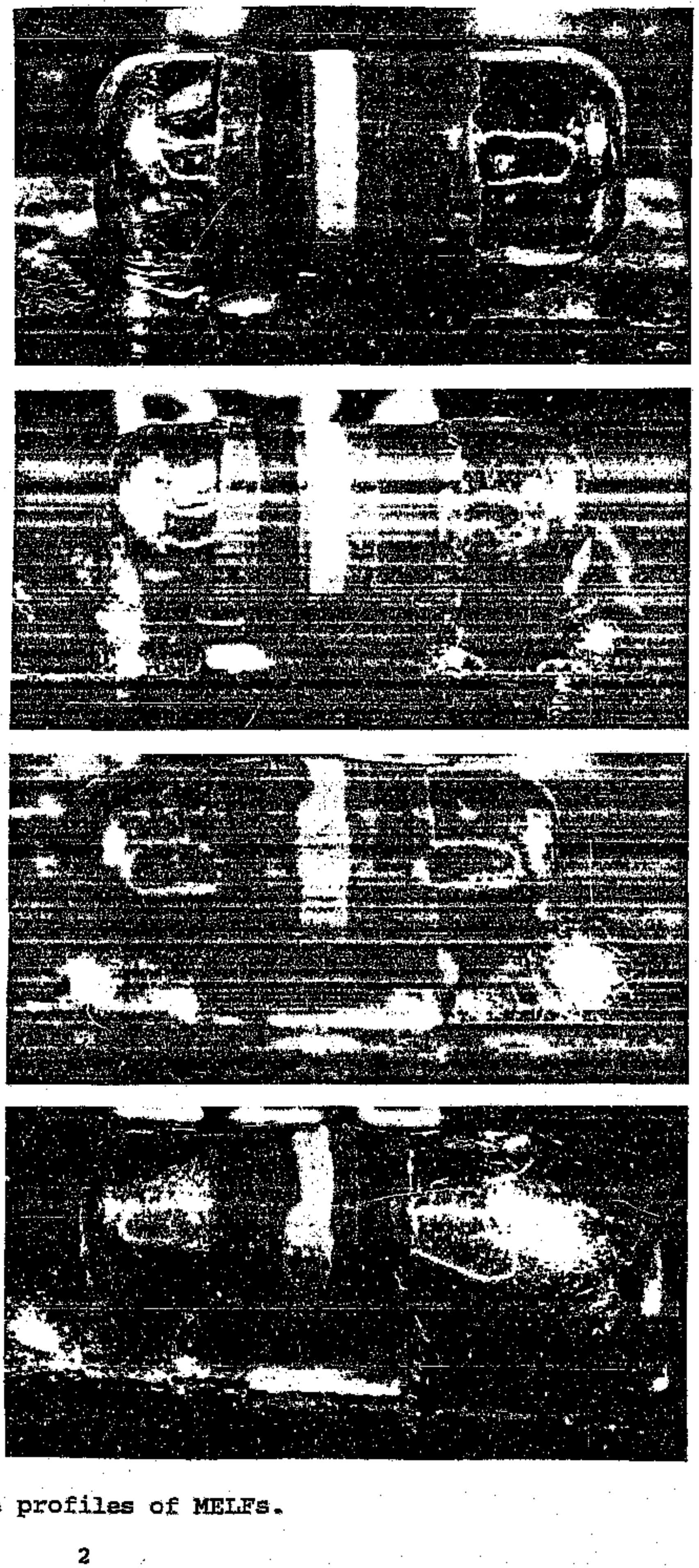


b) Acceptable



c) Excessive

Figure A8.1 Solder volume profiles of chip capacitors.



a) **Unacceptable**

b) **Minimum acceptable**

c) **Maximum acceptable**

d) **Excessive**

Figure A8.1 Solder volume profiles of MELFs.

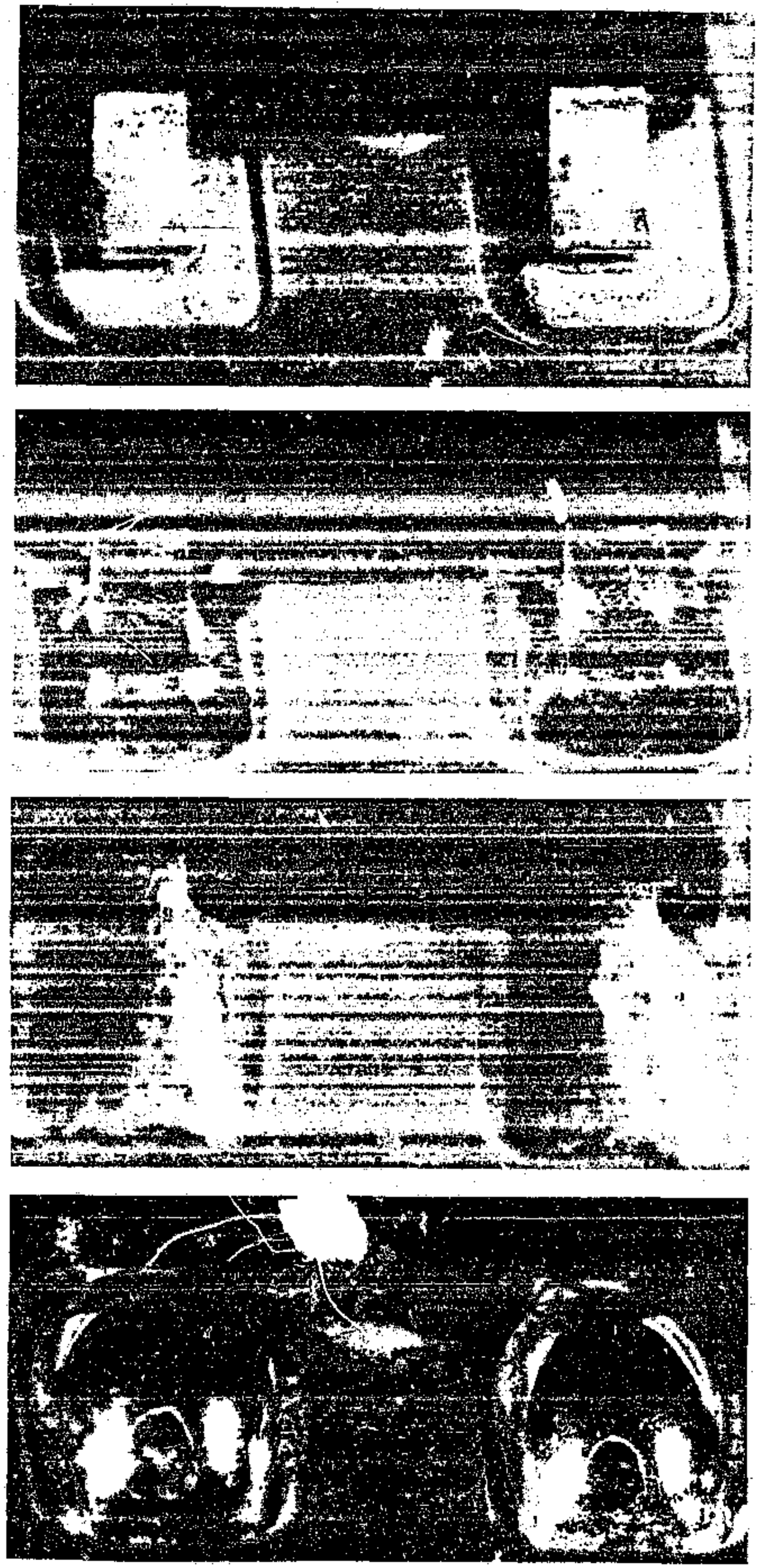
a) **Unacceptable**

b) **Minimum acceptable**

c) **Maximum acceptable**

d) **Excessive**

Figure A8.2 End on solder volume profile view of MELFs.



a) **Unacceptable**

b) **Minimum acceptable**

c) **Maximum acceptable**

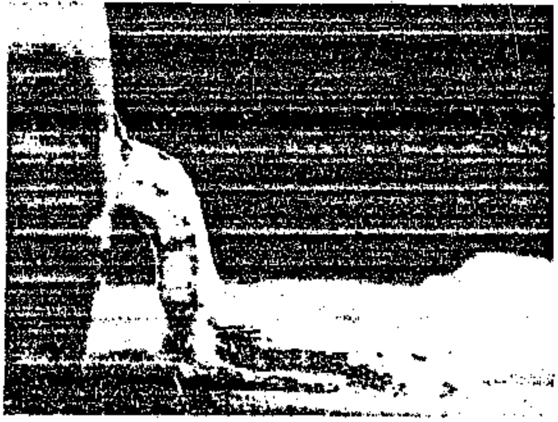
d) **Excessive**

Figure A8.3 Solder volume profiles of (TO-236), SOT 23.

a) **Unacceptable**

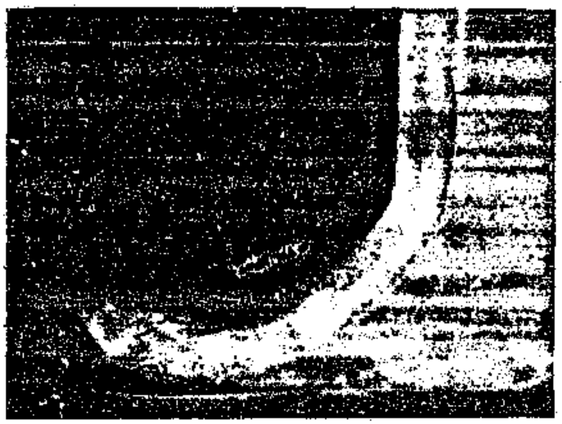
b) **Minimum acceptable**

c) **Maximum acceptable**



d) **Excessive**

Figure A8.4 Solder volume profiles of SOICs, (Gull wings).



a) Unacceptable

b) Minimum acceptable

c) Maximum acceptable

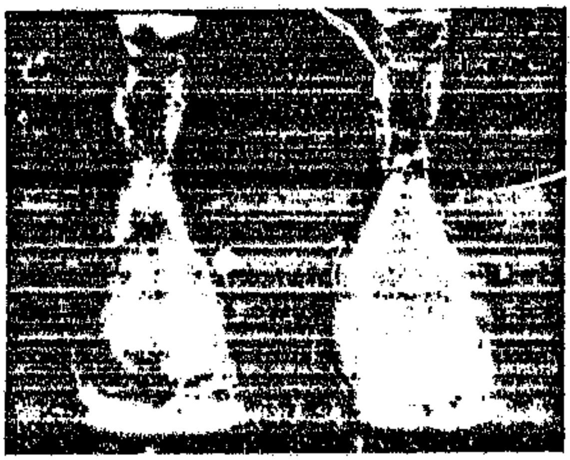
d) Excessive

Figure A8.4 Solder volume profiles of PLCCs, ("J leads").

a) Unacceptable

b) Minimum acceptable

c) Maximum acceptable



d) Excessive

Figure A8.4 Solder volume profiles of LCCCs.

A9 WAVE SOLDERING - FLUX DESCRIPTION

A9.1 Flux - Description And Application

Populating a substrate involves the soldering of a variety of terminations simultaneously. In one operation, a mixture of tinned copper, tin/lead or gold plated nickel-iron, palladium-silver, tin/lead plated nickel-barrier, and even materials like Kovar, each possessing varying degrees of solderability, must be attached to a common substrate using a single solder alloy.

It is for this reason that the choice of flux is so important. The correct flux will remove surface oxides, prevent reoxidization, help transfer heat from the source to the joint area, and leave non-corrosive, or easily removable corrosive residues on the substrate. It will also improve the wettability of the solder joint surfaces.

The wettability of a metal surface is its ability to promote the formation of an alloy at its interface with the solder to ensure a strong, low-resistance joint.

However, the use of flux does not eliminate the need for adequate surface preparation. This is very important in the soldering of SMD substrates, where any temptation to use a highly-active flux in order to promote rapid wetting of ill-prepared surfaces should be avoided because it can cause serious problems later when the corrosive flux residues have to be removed. Consequently, optimum solderability is an essential factor for SMD substrate assembly.

Flux is applied before the wave soldering process, and during the reflow soldering process (where flux and solder are combined in a solder cream). By coating both bare metal and solder, flux retards atmospheric oxidization which would otherwise be intensified at soldering temperature. In the areas where the oxide film has been removed, a direct metal-to-metal contact is established with one, low energy interface. It is from this point of contact that the solder will flow.

A9.2 Types Of Flux

There are two main characteristics of flux. The first is efficacy - its ability to promote wetting of surfaces by solder within a specified time. Closely related to this is the activity of the flux, that is its ability to chemically clean the surfaces. (See Appendix 7)

The second is the corrosivity of the flux, or rather the corrosivity of its residues remaining on the substrate after soldering. This is again linked to the activity; the more active the flux, the more corrosive are its residues.

Although there are many different fluxes available, and many more are being developed, they fall into two basic categories. Those with residues soluble in organic liquids, and those with residues soluble in water.

A9.2.1 Organic soluble fluxes

Most of the fluxes soluble in organic liquids are based on colophony or rosin (a natural product obtained from pine sap that has been distilled to remove the turpentine content). Solid colophony is difficult to apply to a substrate during machine soldering, so it is dissolved in a thinning agent, usually an alcohol. It has a very low efficacy and hence limited cleaning power, so activators are added in varying quantities to increase it. These take the form of organic salts that are chemically active at soldering temperatures. It is therefore, convenient to classify the colophony-based fluxes by their activator content.

A9.2.2 Water soluble fluxes

The water soluble fluxes are generally used to provide high fluxing activity. Their residues are more corrosive and more conductive than the rosin-based fluxes, and consequently must always be removed from the finished substrate. Although termed water soluble, this does not necessarily imply that they contain water, they may also contain alcohol or glycols. It is the flux residues that are water soluble.

The usual composition of a water soluble flux is as follows:

(1) A chemically active component for cleaning the surfaces.

(2) A wetting agent to promote the spreading of flux constituents.

(3) A solvent to provide even distribution.

(4) Substances such as glycols or water soluble polymers to keep the activator in close contact with the metal surfaces.

Although these substances can be dissolved in water, other solvents are generally used, as water has a tendency to spatter during soldering.

Solvents with higher boiling points, such as ethylene glycol or polyethylene glycol are preferred.

Regardless of the type of flux used, it must be applied in a uniform layer prior to soldering. For wave soldering, liquid flux is normally used and the most common application methods are foam fluxing, wave fluxing and spray fluxing.

A10 WAVE SOLDERING - BOARD PREHEATING

A10.1 Board Preheating

The three primary reasons for preheating are:

o solvent evaporation

o reduction of thermal shock

o activation of rosin fluxes

A10.1.1 Solvent evaporation (flux drying)

Solvent that remains in the flux at the soldering step will boil violently, spattering solder balls across the board. It also consumes heat from the wave, altering the heating dynamics and possibly delaying the onset of wetting.

Rosin fluxes use solvents with low vapour pressures and are easily dried. Fluxes containing water require longer drying times. It is virtually impossible to completely evaporate all moisture, and the small amount that remains causes a certain amount of spattering during soldering. For this reason, most water-soluble fluxes do not actually contain water but instead use alcohols or glycols, which are more easily dried.

A10.1.2 Reduction of thermal shock

Molten solder has an extremely high heat capacity, and thermal shock of the assembly is always a concern. With through-hole technology, the primary concern is the PC board. If a board at room temperature is exposed directly to the solder wave, it can suffer excessive warpage and possible delamination.

Surface mount components introduce an additional concern because they are immersed directly in the wave. Excessive temperature gradients can damage chip components or plastic-encapsulated semiconductors. Preheating reduces this risk by lowering the temperature difference between the solder and the PCB.

The preheat temperature is defined as being measured on the top side of the board. It can be measured with either a thermocouple or temperature-sensitive paint. Preheat temperatures for through-hole boards range between 80^{0}C for simple single-sided boards to as high as 125^{0}C for thick multilayer boards. The preheat profile for surface mount boards must be more tightly controlled than has been neccesary for through-hole boards: besides controlling the maximum temperature, the rate of temperature increase must also be controlled.

A10.1.3 Activation of rosin fluxes

Rosin fluxes (Refer to Appendix 9) do not become fully active until they reach temperatures of about 80°C. Preheating assures that they remain at this temperature long enough to remove surface oxides and other contaminants.

A11 WAVE SOLDERING - DUAL WAVE SOLDERING

A11.1 Wave Soldering

The problems of shadowing and capillary depression gave rise to the concept of dual wave soldering, i.e. the division of the soldering process into two steps:

- Primary wetting
- Final soldering

The first, turbulent wave has a higher energy of flow than the second wave. It ensures good wetting of all areas that would not be reached by a normal wave. Due to the turbulence of the first wave the point of solder separation from the PCB bottom, i.e. from the component terminations, cannot be defined. The result would be solder accumulation and bridging; the second wave, however, corrects these irregularities. Yet, improvement of the soldering results implies longer dwell times.

Figure A11.1 shows the procedures used at present for the generation of turbulent pre-waves. With the first and most popular method the solder stream contacts the PCB bottom tangentially (hollow waves). With the second method the solder impinges on the PCB vertically. For generation of a hollow wave the liquid solder is pumped into a nozzle with a slot-shaped aperture. When ejected from the nozzle, the solder stream assumes the shape of a trajectory parabola. The solder moves in the same direction as the PCB, but with higher speed. The board cuts the wave crest with its bottom dipping into the solder.

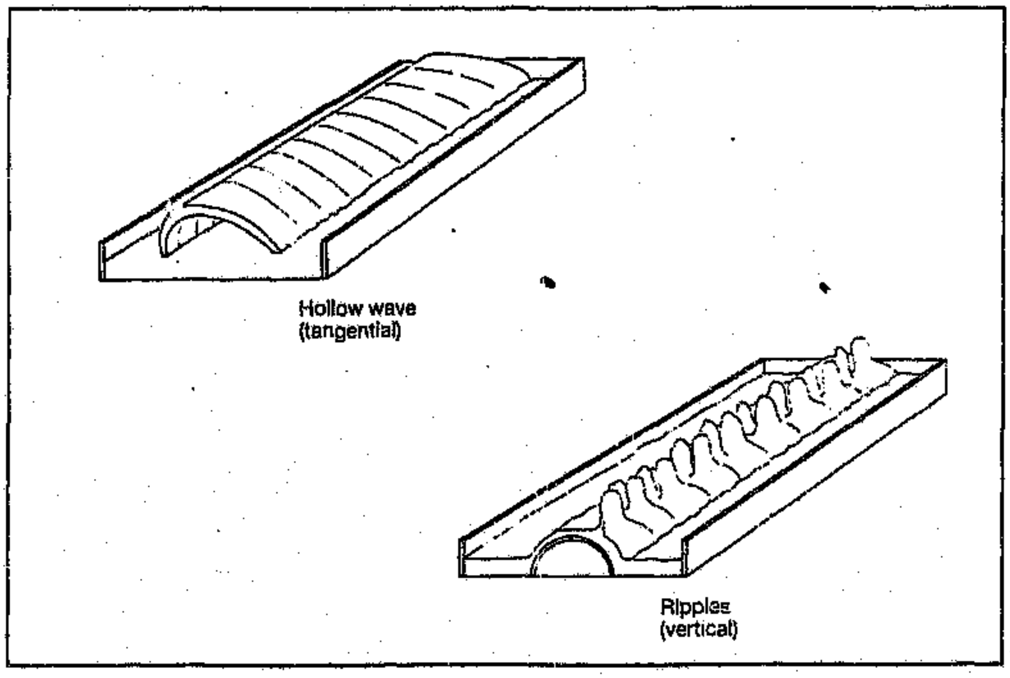


Figure A11.1 Two methods of generating turbulent waves.

This kind of solder movement ensures reliable wetting of the metallization areas at the SMD rear, thereby preventing the shadow effect. The hollow wave's high velocity of flow creates a dynamic impact pressure also acting in vertical direction and thus compensating capillary depression. Even PCBs with openings and milled apertures, e.g. PCB clusters, can be reliably soldered in the hollow wave, since the tangential contact virtually eliminates the risk of solder rising to the PCB top.

The second principle is implemented by a staggered arrangement of the nozzles. The solder is pumped up at high pressure and forms ripples with turbulent motion, mainly in the vertical direction. This method also enables compensation of the shadow effect and capillary depression. Due to the high pressure of flow it is, however, necessary to provide appropriate covers for PCBs with large apertures to prevent the solder from rising to the top.

The two waves can be produced from one common tank or from two separate tanks. In any case, the two pumps should be separately adjustable.

Soldering in the second wave follows the same pattern as single wave soldering. The separation of the solder from the PCB is determined by the interaction of cohesive and adhesive forces between the solder and the wave and between the solder and the board, respectively Figure A11.2. The line of solder peel-back is not constant. It depends on the soldering parameters, the PCB layout and the component configuration on the PCB bottom.

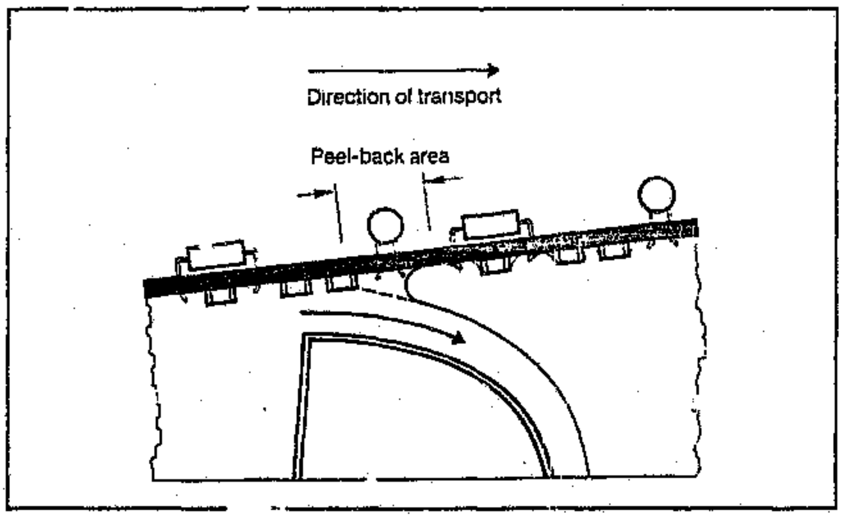


Figure A11.2 Solder peel-back of laminar main wave.

The inclination of the conveyor system towards the wave usually is adjustable. Small angles produce voluminous fillets, whereas steeper angles produce lean fillets. Extremely small angles increase the risk of icicling.

Appropriate adjustment of the transport velocity ensure that the solder peel-back is uniform and reproducible. The maximum possible transport velocity is determined by the PCB layout, the component configuration, and the lead spacing of multi-terminal components. If these factors are not taken into consideration, higher transport velocity leads to a disproportionate increase of solder bridging.

After the second and final soldering step the PCB assemblies are allowed to cool down. During the phase transition from liquid to solid, the solder alloy is in a "pasty" phase, the length of which depends on the composition of the solder. In this pasty phase the assembly must not be subjected to mechanical shock or vibration before the solidus temperature is reached, since any motion is liable to cause micro-cracks. In some cases the solidification process is accelerated by cooling with forced air.

A11.2 Soldering Temperature Profile

The temperature of the solder wave is a trade-off between two competing factors. On the one hand, it should be much higher than the solder melting temperature to reduce the potential for bridging. On the other hand, it should be as low as possible to minimize the potential of thermal damage to the components and board.

Solder wave temperatures for through-hole technology commonly range between 245-280°C. At lower temperatures, an increase in bridging is often noted. As the differential between the actual solder temperature and the melting temperature decreases, the solder is more likely to solidify before it can peel cleanly away from the board. In fact, a common solution to the problem of excess bridging is to increase the temperature of the solder wave.

The temperature sensitivity of surface mount components sets a practical limit for the overall wave soldering thermal profile. For most applications, wave temperatures should be in the range 235-250°C. At temperatures near the low end of this range, it may be necessary to use a supplementary technique, such as hot air knife, to remove solder bridges.

Preheating is essential for minimizing thermal shock. In many applications, ceramic capacitors are the most thermally sensitive parts. Because the barium-titanate dielectric used in these capacitors undergoes a change in crystal structure at the Curie point temperature of about 120°C, the temperature gradient through this region must be carefully controlled. The board temperature should be gradually elevated to a point well above the Curie temperature, generally to within 100-125°C of the soldering temperature.

As with reflow soldering, the speed of this ramp should be 2-4°C/sec. A suggested temperature profile for the total wave soldering process is shown in **Figure A11.3.**

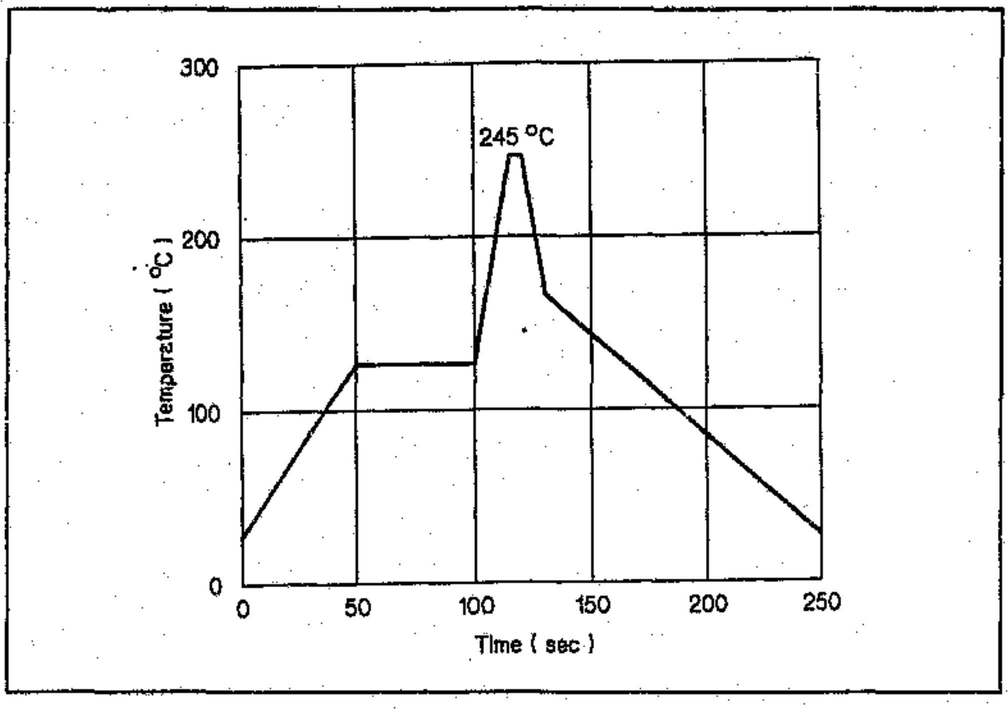


Figure A11.3 Idealized temperature profile for single-wave soldering system.

The actual temperature profile for a dual-wave system is slightly more complicated than the ideal profile. As the board makes the transition between the first and second waves, it cools slightly, causing a characteristic "double peak" profile (See Figure A11.4).

It has been suggested that this rapid temperature fluctuation increases the risk of cracking ceramic capacitors. In this regard, a system such as the Omega wave that combines both turbulent and laminar waves into a single wave may have an advantage over a system in which the two waves are physically separated. Much more experimental work needs to be performed to confirm this hypothesis.

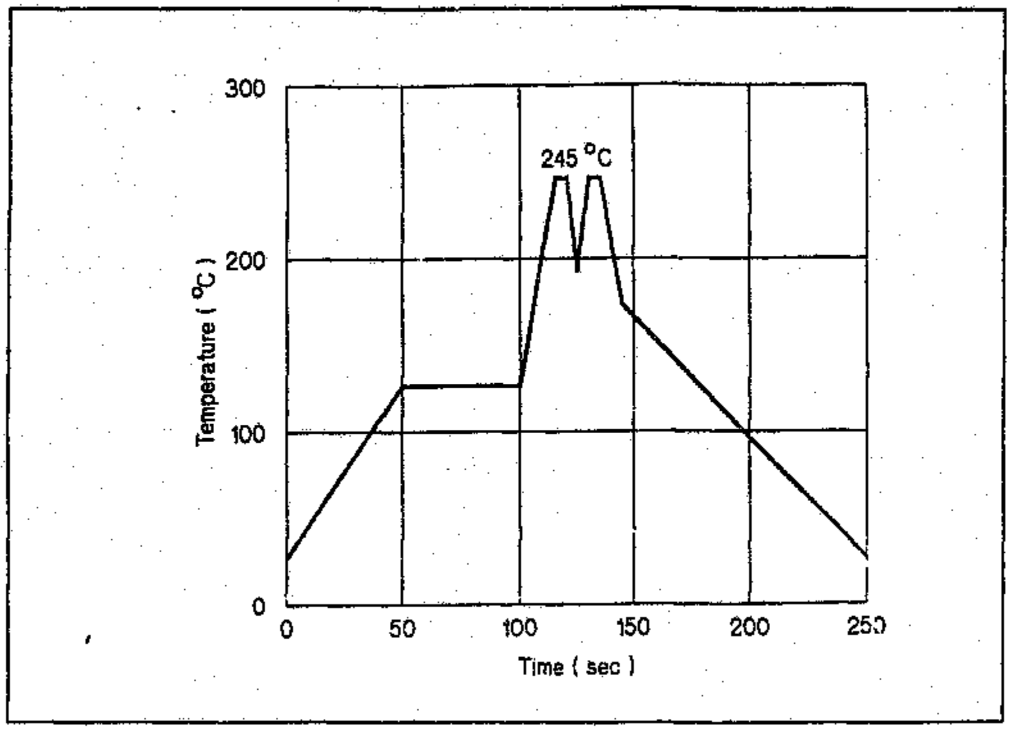


Figure A11.4 Idealized temperature profile for dual-wave soldering system.

A12 REFLOW SOLDERING ALTERNATIVES

A12.1 Reflow By Thermal Conduction

A12.1.1 Conveyorized production systems

Conveyorized systems for production use are are similar to thatin Figure A12.1. Commercial machines include at least two heating zones, one for preheating and the second for reflow. Additional preheat zones are sometimes included to permit custom tailoring of the reflow profile. The cooling zone normally consists of a metal heat sink over which cool air is blown.

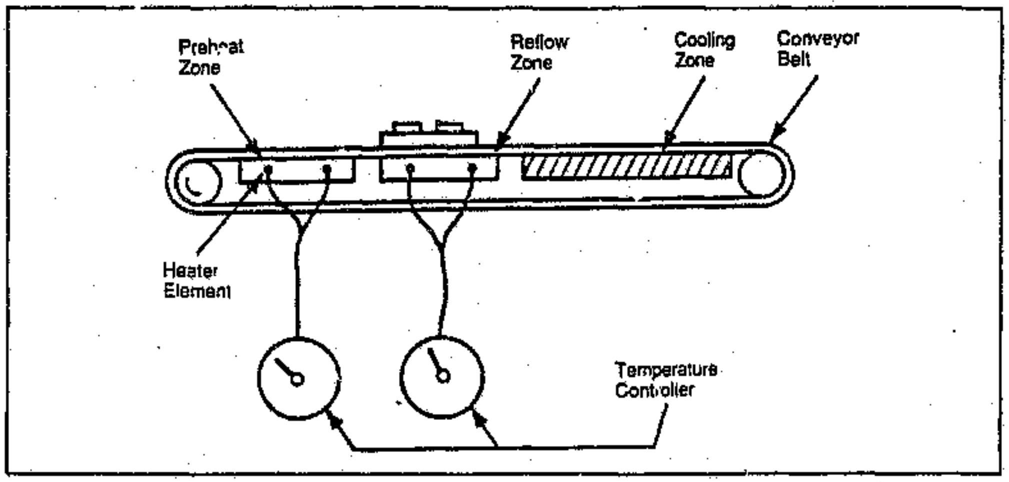


Figure 12.1 Conveyorized thermal conduction reflow system.

Two types of conveyors can be used. One type is solid belt made of Teflon-coated fibreglass. The assembly rides on top of the belt while progressing through the heating zones. The other type of conveyor does not use a belt but instead employs a series of sweeper bars attached to a drive chain. The assembly is positioned so that a sweeper bar pushes it from behind. Improved heat transfer can be achieved with this technique because the substrate is in direct contact with the hot plates. However, interfacing the system to a component placement machine is more difficult; assemblies must be sequenced to enter the system between successive bars. Sequencing is not necessary when using a solid belt. The solid belt is also better suited for transporting odd-shaped assemblies.

Because heat must be transferred through the substrate, conveyorized systems are best suited for flat substrates with high thermal conductivity. Ceramic or porcelain-enamel steel substrates are ideal materials. Glass-epoxy printed wiring boards are less compatible because of their relatively low thermal conductivity.

To elevate the top of the board to reflow temperature, the hot-plate temperature must be increased above 250°C. At such high temperatures, a certain amount of board charring and delamination is inevitable, and the flame retardancy properties of the board can be degraded. A typical temperature profile for a two-stage system is illustarated in Figure A12.2.

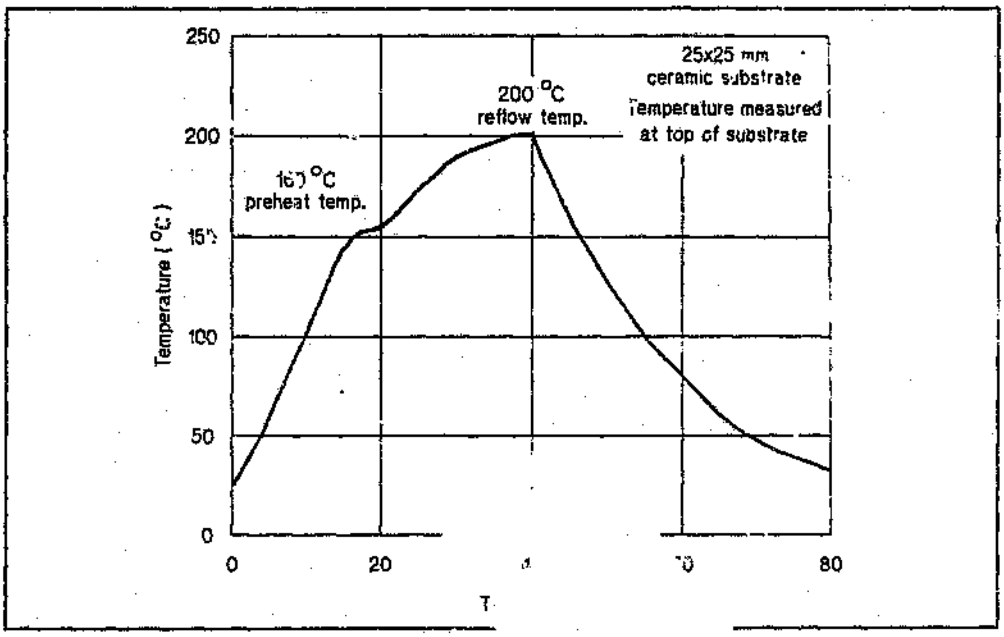


Figure A12.2 Typical tempera [illegible] le for ceramic PC board on conve[illegible] two-stage hot plate system.

A12.2 Vapor Phase Reflow

A12.2.1 Vapor phase theory

A basic vapor phase system consists of a container that holds a quantity of fluid. The fluid temperature is raised to its boiling point by a suitable heater. Above the boiling fluid is a saturated vapor zone that provides the heat for soldering. At the top of the container is a set of condensing coils. The coils reduce vapor loss due to evaporation into the environment.

In operation, the assembly to be soldered is lowered into the saturated vapor zone. The vapors condense on the relatively cool assembly, transferring the latent heat of vaporization to the part. Heat continues to be transferred until the assembly reaches thermal equilibrium with the vapor. By employing a fluid that boils at a temperature above the melting point of the solder, reflow can be achieved.

The vapor phase process has several advantages over other reflow methods. These include:

- Well-controlled maximum temperature.
- Excellent temperature uniformity across the assembly.
- Soldering occurs in a virtually oxygen-free environment.
- Heating is relatively independent of the geometry of the assembly.

A12.2.2 Large temperature gradient

The inherent design of the vapor phase system causes the assembly to make the transition from room temperature to reflow temperature at a rate limited only by its thermal mass. Typical temperature gradients of 15-20°C/sec can damage certain types of components.

Thermal shock can be reduced by including a separate preheat stage in the process. Some in-line equipment can be purchased with an integral preheater at the inlet to the system. When this is not available, a separate IR preheater can be installed in line with the vapor phase system. When using a separate preheater, care must be taken to minimize the temperature drop between the outlet of the preheater and the inlet of the vapor phase system.

A12.3 Infrared Reflow

Solder reflow by application of direct infrared energy ("IR reflow") has recently gained widespread popularity. A system is illustrated schematically in Figure A12.3. It is comprised of a conveyor belt that carries the assembly through a series of heating zones. Each zone consists of a set of infrared emitters positioned above and below the belt. The temperature profile seen by the board is controlled by adjusting the emitter temperatures, the distances between the emitters and conveyor, and the speed of the belt.

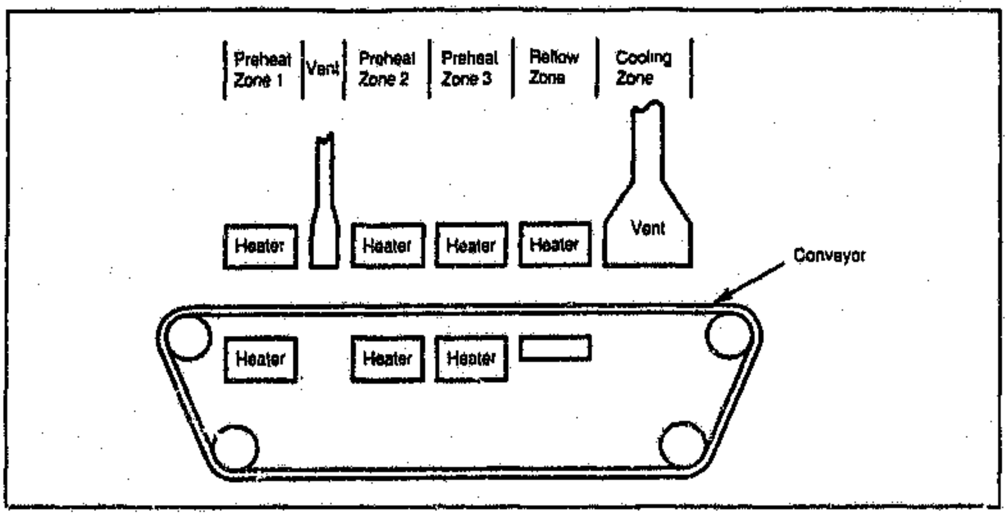


Figure A12.3 Construction of IR reflow system.

The advantages of IR reflow are:

- The temperature profile seen by the board can be precisely controlled.
- Energy transfer by direct radiation is faster than by thermal conduction or convention.
- Radiated energy penetrates inside the joint, whereas conducted or convected energy heats only the surface of the joint.

A12.3.1 Temperature control

Precise temperature control is achieved by employing several heating zones. Each zone can be individually adjusted to tailor the temperature gradient seen by the board. Commercial systems employ as few as 3 or as many as 20 zones to control the profile. (The interpretation of what constitutes a "zone" is decided by each manufacturer. Some manufacturers count both top and bottom elements as a single zone, while others consider them as separate zones).

A12.3.2 Heat transfer rate

Heat transfer by direct radiation follows the Stefan-Boltzman law:

$$E = K(T^4_1 - T^4_2)$$

where E = amount of energy transferred

T_1 = temperature of emitter

T_2 = temperature of assembly

K = a constant

As can be seen, energy is transferred at a rate proportional to the fourth power of the temperature difference between emitter and assembly. This is a much higher rate than transfer by conduction or convection, which is proportional to the simple difference in temperature.

A12.3.3 Heat penetration

While convective and conductive reflow approaches are surface-heating phenomena, direct radiation penetrates more deeply into the joint. This has several potential benefits. Solvents in the solder paste are more easily driven off without spattering, so a separate preheat step is not necessary. In addition, the entire joint heats up as a unit, reducing the possibility of preferential solder wicking up the leads of PLCC devices. The combination of gradual temperature rise and penetrating heat virtually eliminates the possibility of chip component tombstoning.

Infrared reflow has several potential disadvantages. These include:

- For any given equipment setting, the actual temperature profile seen by board is highly dependent on its thermal mass.
- Energy absorption is sensitive to the absorptivity (color) of the components and printed wiring board.
- Direct infrared energy is blocked by tail components.

A12.3.4 Thermal mass sensitivity

Infrared reflow is a non-equilibrium process. Because the emitters do not operate efficiently at the relatively low reflow temperature, they must be operated at much higher temperatures. The actual temperature of the board as it passes through any zone is a function of the board thermal mass and the speed of the belt. **Figure A12.4** shows the relationship between the emitted temperatures and the board surface temperature for a board of given thermal mass.

When the machine settings are not optimized for the particular board being reflowed, several things can happen. If the board has a low thermal mass, it will become overheated, causing discoloration or charring. Overheated components can be irreparably damaged. In extreme cases the board material can even catch fire. On the other hand, if the board has a high thermal mass, it may not reach reflow temperature. Even if the temperature does reach the reflow point, it may not remain their long enough for all joints to fully form.

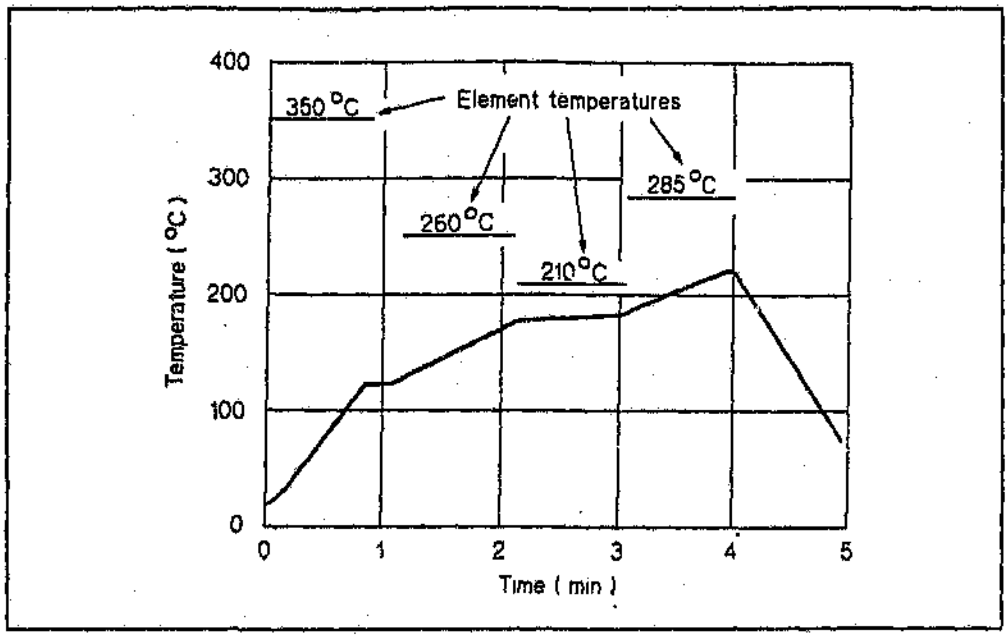


Figure A12.4 Typical temperature profile for area source infrared reflow system.

A12.3.5 Temperature profiling

Because IR reflow is a non-equilibrium process, a specific temperature profile must be developed for each board type. This step is necessary regardless of whether a lamp or a panel emitter system is employed. The objective in profiling is to determine the specific emitter temperatures, emitter-conveyor spacings, and belt speed that produce the temperature profile nearest the ideal.

Profiling is accomplished by attaching a set of thermocouples to a representative sample board and monitoring board temperature as it progresses through the oven. The thermocouples are attached to long lengths of heatresistant wire. Since a number of runs may be needed to determine the optimum profile, the test is considered destructive to the sample board.

For best results, the board should be populated with components, although they need not be electrically functional. The number of thermocouples should be sufficient to measure the temperature uniformity across the board surface and to monitor any local areas of unusually small or large thermal mass.

From a practical standpoint, it is desirable to adjust only those parameters that can easily be adjusted in production. Emitter-conveyor distance, for example, is usually difficult to change. Every attempt should be made to leave this parameter fixed for all board types. Belt speed is usually easiest to adjust, followed by emitter temperature.

A recommended profiling sequence is as follows:

1. Attempt to identify the optimum profile by adjusting only the speed of the conveyor belt. Maintain the emitter heights and temperatures at some nominal value.

2. If an acceptable profile cannot be achieved by adjusting belt speed alone, adjust emitter temperatures as necessary. Keep the temperature changes as small as possible and use belt speed to make gross adjustments.

3. If the profile cannot be optimized through use of emitter temperature and belt speed adjustments, adjust emitter heights as necessary.

In the interest of maintaining a just-in-time manufacturing capability, consider using a single setting for a number of different boards. Although for some boards this compromise profile may deviate from ideal, the difference may not have a practical impact on product quality. All board types should still be profiled per the above procedure to assure compatibility with the standard settings.

A12.4 Laser Reflow

All the reflow methods described thus far subject the entire assembly to the reflow temperature for as long as 30-60 seconds. Some types of components are damaged by exposure to these temperature extremes. Many hybrid circuits, for example, employ components that have themselves been soldered with eutectic tin-lead solder. The lid seal on some hermetic devices is also made with eutectic tin-lead. Obviously, subjecting these devices to a second reflow operation is detrimental.

Laser reflow soldering was developed to address this concern. Unlike the previous methods, heat energy is directed only onto the joints being soldered and each joint must be formed individually. Temperature sensitive components can more readily be soldered without fear of damage.

A typical system utilizes a CO_2 or Nd:YAG laser, a mechanical X-Y positioning stage, and a computer controller. The controller moves the board under the laser as necessary to reflow all the joints sequentially.

The benefits of laser soldering include:

- Heating is highly localized, reducing the potential for damage to thermally sensitive devices.
- Joints are formed very rapidly, reducing the potential for intermetallic growth.
- Stresses in the joint are reduced compared to mass reflow methods.

A12.4.1 Slow soldering speed

The effective rate at which joints can be formed includes both the actual soldering time and the time required to position the laser beam. Measured throughputs range from 4-10 joints per second or about 15,000-35,000 joints per hour. Depending on the nature of the board, this can be an order of magnitude less than the capacity of a vapor phase system.

A12.5 Reflow By Thermal Convection

Convention ovens are rarely used to reflow solder paste. Compared to vapor phase or IR methods, the rate of heat transfer is much less. They are normally used only in low-volume applications where equipment cost is a primary concern and throughput is not. Two exceptions are the IR-heated convection oven and the hot gas repair stations

Appendix 13

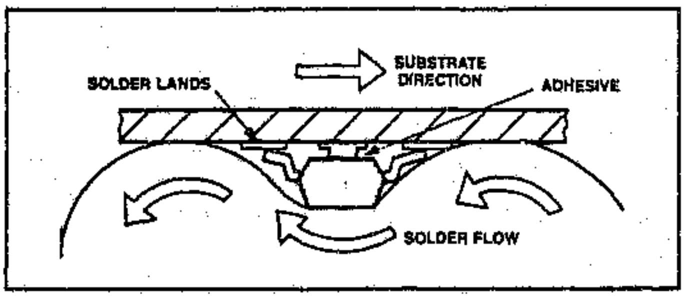


Figure A3.1a Surface tension preventing the solder reaching the downstream end of the SMD.

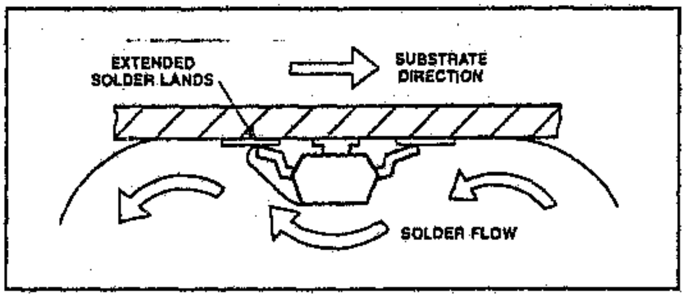


Figure A3.1b Extending the solder land to overcome the shadow effect.

The use of the dual-wave soldering technique also partially alleviates this problem because the first turbulent wave has sufficient upward pressure to force solder onto the component metallization, and the second, smooth wave 'washes' the substrate to form good fillets of solder. Similarly, introducing oil on the surface of the solder wave lowers the surface tension which lessens the shadow effect, but this technique introduces problems of contaminants in the solder when the oil decomposes. (See details on soldering later).

Solder bridging

On wave-soldered substrates the orientation of SO (small outline) and VSO (very small outline) ICs is critical for the prevention of solder bridge formation (See Section 8). Optimum solder penetration is achieved when the central axis of the IC is parallel to the flow of the solder as shown in Figure A3.2a. The SO package may also be transversely orientated, as shown in Figure A3.2b, but this is totally unacceptable for the VSO package.

```
DEFINE Distribution as AN INTEGER, stream 9 VARIABLE

DEFINE Mean.interarrival.time and Sim.length as REAL VARIABLES

DEFINE Delay.in.queue and Days.run as REAL VARIABLES

DEFINE Total.run.time and Workday.length as REAL VARIABLES

DEFINE First.report, Second.report, Third.report,
        and Fourth.report as REAL VARIABLES

DEFINE First.run, Second.run, Third.run, and Fourth.run
                          as REAL VARIABLES

DEFINE Total.no.of.sim and Rep as INTEGER VARIABLES

DEFINE Total.Head1.worktime and Total.Head2.worktime as REAL VARIABLES

DEFINE Layout and Man as pointer variables

DEFINE Number, Number1, Number2, Number3, Number4, Number5,
                Number6, and Number7  as INTEGER VARIABLES

DEFINE Clocktime as a Double Variable

DEFINE Description and Position as TEXT VARIABLES

DEFINE .Right to mean 0

DEFINE .Left to mean -PI.C

DEFINE .Up to mean PI.C/2

DEFINE .Down to mean -PI.C/2

DEFINE Field.id as a text variable
DEFINE Form as a pointer variable
DEFINE Field as a pointer variable
DEFINE Device.id as a pointer variable
Define names as a 1-dim text array

TALLY Mean.ws.delay.in.q AS THE MEAN OF Ws.delay.in.queue

TALLY Mean.jb.delay.in.q AS THE MEAN OF Jb.delay.in.queue

ACCUMULATE Avg.num.mach.working AS THE AVERAGE OF Ws.num.machine.working

ACCUMULATE Avg.num.in.station.q AS THE AVERAGE OF N.q.work.station

ACCUMULATE Avg.num.mach.occupied AS THE AVERAGE OF N.x.work.station

DISPLAY Variables include Clocktime and N.Q.Work.station

GRAPHIC Entities include  Status1, Status2, Status3, Status4,
                Status5, Status6, Status7, Status8, Status9, Status10,
                Status11, Queue1, Queue2, Queue3, Queue4, Queue5,
                Queue6, Queue7, Queue8, Queue9, Queue10, Queue11,and Load

DYNAMIC Graphic Entities include Job

END ''PREAMBLE
```

```
DEFINE Distribution as AN INTEGER, stream 9 VARIABLE

DEFINE Mean.interarrival.time and Sim.length as REAL VARIABLES

DEFINE Delay.in.queue and Days.run as REAL VARIABLES

DEFINE Total.run.time and Workday.length as REAL VARIABLES

DEFINE First.report, Second.report, Third.report,
        and Fourth.report as REAL VARIABLES

DEFINE First.run, Second.run, Third.run, and Fourth.run
                            as REAL VARIABLES

DEFINE Total.no.of.sim and Rep as INTEGER VARIABLES

DEFINE Total.Head1.worktime and Total.Head2.worktime as REAL VARIABLES

DEFINE Layout and Man as pointer variables

DEFINE Number, Number1, Number2, Number3, Number4, Number5,
                Number6, and Number7  as INTEGER VARIABLES

DEFINE Clocktime as a Double Variable

DEFINE Description and Position as TEXT VARIABLES

DEFINE .Rignt to mean 0

DEFINE .Left to mean -PI.C

DEFINE .Up to mean PI.C/2

DEFINE .Down to mean -PI.C/2

DEFINE Field.id as a text variable
DEFINE Form as a pointer variable
DEFINE Field as a pointer variable
DEFINE Device.id as a pointer variable
Define names as a 1-dim text array

TALLY Mean.ws.delay.in.q AS THE MEAN OF Ws.delay.in.queue

TALLY Mean.jb.delay.in.q AS THE MEAN OF Jb.delay.in.queue

ACCUMULATE Avg.num.mach.working AS THE AVERAGE OF Ws.num.machine.working

ACCUMULATE Avg.num.in.station.q AS THF AVERAGE OF N.q.work.station

ACCUMULATE Avg.num.mach.occupied AS THE AVERAGE OF N.x.work.station

DISPLAY Variables include Clocktime and N.Q.Work.station

GRAPHIC Entities include  Status1, Status2, Status3, Status4,
                Status5, Status6, Status7, Status8, Status9, Status10,
                Status11, Queue1, Queue2, Queue3, Queue4, Queue5,
                Queue6, Queue7, Queue8, Queue9, Queue10, Queue11,and Load

DYNAMIC Graphic Entities include Job

END ''PREAMBLE
```

```
MAIN

CALL Initialize
CALL Menu

CALL setvt.r(7,0)
CALL vclears.r
CALL vgotoxy.r(20,2)
PRINT 1 line  thus
Simulation Running,           Hours Completed

OPEN unit 2 for output, file name is "SMTOUT1"

Activate an Arrival.generator in First.run Minutes

Activate an Arrival.generator in Second.run Minutes

Activate an Arrival.generator in Third.run Minutes

Activate an Arrival.generator in Fourth.run Minutes

Activate a fieldtest now

START SIMULATION

Close Unit 2

END '' MAIN
```

```
PROCESS ARRIVAL.GENERATOR

CALL VGOTOXY.R(0,0)
LET Number3 = 0
LET Location.a(Queue10) = Location.f(50,9)
Display Queue10

LET Total.run.time = 0
FOR I = 1 to N.work.station
DO
    IF N.x.work.station(I) = 0
    LET X.coord = St.Xlocation(I)
    LET Y.coord = St.Ylocation(I)
    Print 2 LINES WITH I,X.coord, and Y.coord THUS
    STATION **  X = *** AND Y = ***

    LET Description = "Idle"
    Select case I

        case 5
        LET Location.a(Status5) = Location.f(X.coord,Y.coord)
        Display Status5

        case 6
        LET Location.a(Status6) = Location.f(X.coord,Y.coord)
        Display Status6

        case 10
        LET Location.a(Status10) = Location.f(X.coord,Y.coord)
        Display Status10

        DEFAULT

    Endselect
    ENDIF
LOOP

LET Description = "Busy"

FOR I = 1 to Rep
DO
    Call vgotoxy.r(0,3)
    Let Job.type = 1
    FOR J = 1 to number.of.jobs(job.type)
    DO
        WAIT Uniform.f(0,Mean.Interarrival.Time,3) Minutes
        Activate a Job giving Job.type NOW
        LET VXFORM.V = 1
        Display JOB with "Job.icn" at (20,200)
    LOOP
LOOP

END  '' Arrival.Generator
```

```
PROCESS FIELDTEST

define field.id as text variable
define menu as pointer variable
```

```
SHOW menu WITH "PNPMENU.FRM"
LET Field.ID = ACCEPT.F(menu,'MENUBAR.CTL')
LET Field.ID = Field.ID

END '' Fieldtest




PROCESS FINAL.REPORT

Call vfcolor.r(15)
Call vgotoxy.r(2,1)

IF Time.v < Second.Run
    Print 2 lines with Time.v/60 thus
First Run Completed and Report Logged at Time ***.** Hours

    LET Total.run.Time = Time.v - First.Run

ELSE
    IF Time.v < Third.run
        Call vgotoxy.r(5,1)
        Print 2 lines with Time.v/60 thus
Second Run Completed and Report Logged at Time ***.** Hours

        LET Total.run.Time = Time.v - Second.Run

    ELSE
        IF Time.v < Fourth.run
            Call vgotoxy.r(8,1)
            Print 2 lines with Time.v/60 thus
Third Run Completed and Report Logged at Time ***.** Hours

            LET Total.run.Time = Time.v - Third.run

        ELSE
            Call vgotoxy.r(11,1)
            Print 2 lines with Time.v/60 thus
 Last Run Completed and Report Logged at Time ***.** Hours

            LET Total.run.Time = Time.v - Fourth.run

        Endif
    Endif
Endif


IF Time.v < Second.run
USE unit 2 for output
PRINT 2 lines thus
               SIMULATION RESULTS - FIRST RUN
               ------------------------------
SKIP 2 lines

ELSE
    IF Time.v < Third.run
        USE unit 2 for output
         PRINT 2 lines thus
               SIMULATION RESULTS - SECOND RUN
               -------------------------------
        SKIP 2 lines

    ELSE
```

```
        IF Time.v < Fourth.run
            USE unit 2 for output
            PRINT 2 lines thus
               SIMULATION RESULTS - THIRD RUN
               ------------------------------
            SKIP 2 lines

        ELSE
            USE unit 2 for output
            PRINT 2 lines thus
               SIMULATION RESULTS - FOURTH RUN
               -------------------------------
            SKIP 2 lines

        ENDIF
    ENDIF
ENDIF

Print 3 lines with Time.v/60 and total.run.Time/60 thus
Total running Time = ***.** Hours

Simulation Run Time = ***.** Hours
Skip 1 line

LET Total.run.Time = 0

Print 6 lines thus
Job Report:
----------

           Average Total
Job Type   Delay in Queue   Number Completed
--------   --------------   ----------------

skip 2 lines

    FOR each job.type
    do
        print 2 lines with Job.type, Mean.jb.delay.in.q(job.type)/60,
                        and Num.completed(job.type) thus
   **         ***.**                   ****

    LET  Num.completed(job.type) = 0

    LOOP
    SKIP 2 lines
    PRINT 5 lines thus
Station Report:
--------------
            Average Num     Average Delay    Prop Time   Prop Time   Prop Time
Station     in queue        in Queue         Working     Idle        Blocked
-------     -----------     -------------    ---------   ---------   ---------

        FOR each Work.station
        DO
          LET Prop.working = Avg.num.mach.working(Work.station)/
                             Ws.num.machines(Work.station)
          LET Prop.idle = 1.0 - (Avg.num.mach.occupied(Work.station)/
                                Ws.num.machines(Work.station))
          LET Prop.blocked = 1.0 - Prop.working - Prop.idle

          print 3 lines with Work.station, Avg.num.in.station.q(work.station)
          Mean.ws.delay.in.q/60, Prop.working, Prop.idle, and Prop.blocked th

  **          ***.***          ***.***          *.***       *.***       *.***
```

```
       loop
Skip 2 lines

USE Unit 6 FOR OUTPUT

FOR Each Work.station
DO
    RESET the Totals of Ws.delay.in.queue(work.station),
                        Ws.num.machine.working(work.station),
                        N.q.work.station(work.station),
                        and N.x.work.station(work.station)
LOOP
RESET Totals of Jb.delay.in.queue(1)

END   ''Report




PROCESS JOB given job.number

    DEFINE job.number as an INTEGER VARIABLE
    DEFINE time.of.arrival and total.delay.in.queue as REAL VARIABLES
    DEFINE current.station, current.machine, and task as INTEGER VARIABLES

    LET delay.in.queue = 0
    LET Total.delay.in.queue = 0
    LET Total.run.time = 0
    LET Velocity.a(Job) = 0

    FOR each task in routing(job.number)
    DO

        CALL vfcolor.r(14)
        CALL vgotoxy.r(20,22)
        PRINT 1 line with time.v/60 thus
***.**

        LET current.station = task.work.station(task)
        LET time.of.arrival = time.v

        SELECT Case Current.station

            Case 1
                IF U.Work.station(current.station) = 0
                LET X.coord = Display.X(current.station)
                LET Y.coord = Display.Y(current.station)
                LET Number = N.Q.work.station(current.station) + 1
                LET location.a(Queue1) = Location.f(X.coord,Y.coord)
                Display Queue1
                ENDIF

            Case 3
                IF U.Work.station(current.station) = 0
                LET X.coord = Display.X(current.station)
                LET Y.coord = Display.Y(current.station)
                LET Number = N.Q.work.station(current.station) + 1
                LET location.a(Queue3) = Location.f(X.coord,Y.coord)
                Display Queue3
                ENDIF

            Case 5
                IF U.Work.station(current.station) = 0
```

```
       loop
Skip 2 lines

USE Unit 6 FOR OUTPUT

FOR Each Work.station
DO
    RESET the Totals of Ws.delay.in.queue(work.station),
                        Ws.num.machine.working(work.station),
                        N.q.work.station(work.station),
                        and N.x.work.station(work.station)
LOOP
RESET Totals of Jb.delay.in.queue(1)

END   ''Report





PROCESS JOB given job.number

    DEFINE job.number as an INTEGER VARIABLE
    DEFINE time.of.arrival and total.delay.in.queue as REAL VARIABLES
    DEFINE current.station, current.machine, and task as INTEGER VARIABLES

    LET delay.in.queue = 0
    LET Total.delay.in.queue = 0
    LET Total.run.time = 0
    LET Velocity.a(Job) = 0

    FOR each task in routing(job.number)
    DO

        CALL vfcolor.r(14)
        CALL vgotoxy.r(20,22)
        PRINT 1 line with time.v/60 thus
***.**

        LET current.station = task.work.station(task)
        LET time.of.arrival = time.v

        SELECT Case Current.station

            Case 1
                IF U.Work.station(current.station) = 0
                LET X.coord = Display.X(current.station)
                LET Y.coord = Display.Y(current.station)
                LET Number = N.Q.work.station(current.station) + 1
                LET location.a(Queue1) = Location.f(X.coord,Y.coord)
                Display Queue1
                ENDIF

            Case 3
                IF U.Work.station(current.station) = 0
                LET X.coord = Display.X(current.station)
                LET Y.coord = Display.Y(current.station)
                LET Number = N.Q.work.station(current.station) + 1
                LET location.a(Queue3) = Location.f(X.coord,Y.coord)
                Display Queue3
                ENDIF

            Case 5
                IF U.Work.station(current.station) = 0
```

```
        LET X.coord = Display.X(current.station)
        LET Y.coord = Display.Y(current.station)
        LET Number1 = N.Q.work.station(current.station) + 1
        LET location.a(Queue5) = Location.f(X.coord,Y.coord)
        Display Queue5
        ENDIF

    Case 6
        IF U.Work.station(current.station) = 0
        LET X.coord = Display.X(current.station)
        LET Y.coord = Display.Y(current.station)
        LET Number2 = N.Q.work.station(current.station) + 1
        LET location.a(Queue6) = Location.f(X.coord,Y.coord)
        Display Queue6
        ENDIF

    Case 10
        IF U.Work.station(current.station) = 0
        LET X.coord = Display.X(current.station)
        LET Y.coord = Display.Y(current.station)
        ENDIF

    DEFAULT
ENDSELECT

REQUEST 1 unit of work.station(current.station)

LET current.machine = current.station
LET X.coord = St.Xlocation(current.station)
LET Y.coord = St.Ylocation(current.station)

    Select case Current.station

    case 1
    LET location.a(Status1) = Location.f(X.coord,Y.coord)
    Display Status1

    case 3
    LET location.a(Status3) = Location.f(X.coord,Y.coord)
    Display Status3

    case 5
    LET location.a(Status5) = Location.f(X.coord,Y.coord)
    Display Status5
    LET Current.head = Total.Head1.worktime

    case 6
    LET location.a(Status6) = Location.f(X.coord,Y.coord)
    Display Status6
    LET Current.head = Total.Head2.worktime

    case 10
    LET location.a(Status10) = Location.f(X.coord,Y.coord)
    Display Status10

    DEFAULT

    Endselect

LET delay.in.queue = Time.v - Time.of.Arrival
LET Ws.delay.in.Queue(current.station) = delay.in.queue
ADD delay.in.queue to total.delay.in.queue
ADD 1 to ws.num.machine.working(current.station)
LET Stream = current.station
    IF Stream > 9
        LET Stream = 9 - Stream
```

```
    ENDIF
IF Current.station = 5 OR Current.station = 6
    WORK Current.Head Minutes
ELSE
    WORK Gamma.f(task.work.time(task),2,Stream) Minutes
ENDIF
SUBTRACT 1 from ws.num.machine.working(current.station)

RELINQUISH 1 unit of work.station(current.station)

    Select case Current.station

    case 1
    IF N.x.work.station(current.machine) = 0
        Description = "Idle"
        LET X.coord = St.Xlocation(current.station)
        LET Y.coord = St.Ylocation(current.station)
        LET location.a(Status1) = Location.f(X.coord,Y.coord)
        Display Status1
    ENDIF
    LET X.coord = Display.X(current.station)
    LET Y.coord = Display.Y(current.station)
    LET location.a(Queue1) = Location.f(X.coord,Y.coord)
    LET Number = N.Q.work.station(current.station)
    Display Queue1

    case 3
    IF N.x.work.station(current.machine) = 0
    Description = "Idle"
    LET X.coord = St.Xlocation(current.station)
    LET Y.coord = St.Ylocation(current.station)
    LET location.a(Status3) = Location.f(X.coord,Y.coord)
    Display Status3
    ENDIF
    LET X.coord = Display.X(current.station)
    LET Y.coord = Display.Y(current.station)
    LET location.a(Queue3) = Location.f(X.coord,Y.coord)
    LET Number = N.Q.work.station(current.station)
    Display Queue3

    case 5
    IF N.x.work.station(current.machine) = 0
    Description = "Idle"
    LET X.coord = St.Xlocation(current.station)
    LET Y.coord = St.Ylocation(current.station)
    LET location.a(Status5) = Location.f(X.coord,Y.coord)
    Display Status5
    ENDIF
    LET X.coord = Display.X(current.station)
    LET Y.coord = Display.Y(current.station)
    LET location.a(Queue5) = Location.f(X.coord,Y.coord)
    LET Number1 = N.Q.work.station(current.station)
    Display Queue5

    case 6
    IF N.x.work.station(current.machine) = 0
    Description = "Idle"
    LET X.coord = St.Xlocation(current.station)
    LET Y.coord = St.Ylocation(current.station)
    LET location.a(Status6) = Location.f(X.coord,Y.coord)
    Display Status6
    ENDIF
    LET X.coord = Display.X(current.station)
    LET Y.coord = Display.Y(current.station)
    LET location.a(Queue6) = Location.f(X.coord,Y.coord)
    LET Number2 = N.Q.work.station(current.station)
```

```
            Display Queue6

            case 10
            IF N.x.work.station(current.machine) = 0
            Description = "Idle"
            LET X.coord = St.Xlocation(current.station)
            LET Y.coord = St.Ylocation(current.station)
            LET location.a(Status10) = Location.f(X.coord,Y.coord)
            Display Status10
            ENDIF
            LET X.coord = Display.X(current.station)
            LET Y.coord = Display.Y(current.station)

            DEFAULT

            Endselect

        LET Description = "Busy"
        LET X.coord = Conveyor.Xlocation(current.station)
        LET Y.coord = Conveyor.Ylocation(current.station)
        LET Direction = Conveyor.direction(current.station)
        LET Length = Conveyor.length(current.station)
        LET Speed = 12

        LET Vxform.v = 1
        Display JOB with "Job.icn" at (X.coord, Y.coord)
        LET Velocity.a(Job) = Velocity.f(Speed,Direction)
        WORK Length/speed Minutes
        LET Velocity.a(Job) = 0

        CALL vfcolor.r(14)
        CALL vgotoxy.r(20,22)
        PRINT 1 line with time.v/60 thus
***.**

     LOOP

    ADD 1 to num.compLETed(job.number)
    LET Number3 = Num.completed(job.number)
    LET location.a(Queue10) = Location.f(50,9)
    Display Queue10
    LET jb.delay.in.queue(job.number) = total.delay.in.queue
    call vfcolor.r(7)
    LET Total.run.time = Time.v - Total.run.time

IF Num.completed(job.number) = Number.of.Jobs(job.number)
   Activate a Final.Report Now
ALWAYS

END  ''JOB
```

```
Routine GRAPHICS.INIT

Define Devptr as a POINTER VARIABLE

Create Status1
Create Status3
Create Status5
Create Status6
Create Status10
Create Queue1
Create Queue3
Create Queue5
Create Queue6
Create Queue10

LET Timesync.v = 'Clock.Update'

   ''Create a Graphic display using 2 new I/O units

  Call devinit.r giving "vt,graphics" yielding devptr
  Open 7 for input, device is devptr
  Open 8 for output, device is devptr, recordsize is 1024
  Use 8 for graphic output

   ''Select a viewing transform and establish its mapping

LET VXFORM.V = 2  '' For Text Display
Call Setworld.r(1.0, 80.0, 1.0, 25.0)

Let VXFORM.V = 1  '' For Graphics Output
Call SETWORLD.R giving 0.0,250.0,0.0,250.0

LET VXFORM.V = 3  '' For Text Display
Call Setworld.r(1.0, 79.0, 1.0, 24.0)

LET Drtn.a(Status5) = 'V.Status5'
LET Drtn.a(Status6) = 'V.Status6'
LET Drtn.a(Queue5) = 'V.Queue5'
LET Drtn.a(Queue6) = 'V.Queue6'
LET Drtn.a(Queue10) = 'V.Queue10'

 Let VXFORM.V = 1
 Show layout with "layout.frm"
 Display layout

Display Clocktime with "Clock.grf"


End ''GRAPHICS.INIT




ROUTINE INITIALIZE

CALL Vbcolor.r(1)
CALL Vfcolor.r(15)
call vCLEARS.R
CALL Setvt.r(7,0)
CALL Vgotoxy.r (5,10)

  OPEN UNIT 2 FOR INPUT, FILE NAME IS "SMTTEST.TXT"    ''"SMTIN.DAT"
  USE 2 FOR INPUT
```

```
 READ N.work.station
 CREATE every Work.station
 FOR each Work.station
 DO
     READ Ws.num.machines(work.station)
     LET U.work.station(work.station) = Ws.num.machines(work.station)
     READ Conveyor.Length(work.station)
     READ Conveyor.Direction(work.station)
     READ Conveyor.Xlocation(work.station)
     READ Conveyor.Ylocation(work.station)
     READ Ws.Xlocation(work.station)
     READ Ws.Ylocation(work.station)
     READ St.Xlocation(work.station)
     READ St.Ylocation(work.station)
     READ Q.Xlocation(work.station)
     READ Q.Ylocation(work.station)
     READ Display.X(work.station)
     READ Display.Y(work.station)

LOOP

 READ Mean.interarrival.time
 READ N.job.type
 CREATE every Job.type

 FOR each Job.type
 DO
   READ Number.of.jobs(job.type)
   . READ Num.tasks
     FOR I = 1 to Num.tasks
         DO
         CREATE a Task
         READ Task.work.station(task)
         READ Task.work.time(task)
         FILE this task in Routing(job.type)
     LOOP
 LOOP

 READ Rep
 READ Total.no.of.sim
 FOR I = 1 to Total.no.of.sim
 DO
   IF I = 1
   READ First.run
   READ First.report
   ELSE
      IF I = 2
      READ Second.run
      READ Second.report
      ELSE
         IF I = 3
         READ Third.run
         READ Third.report
         ELSE
            IF I = 4
            READ Fourth.run
            READ Fourth.report
            ENDIF
        ENDIF
     ENDIF
 . ENDIF
 LOOP

     READ Total.head1.worktime
     READ Total.head2.worktime
```

```
    CLOSE unit 2
LET Timescale.v = 100

END '' INITIALIZE






ROUTINE JOB.SCHEDULE

    Define DONE as a text variable
    Define choice as an alpha variable

    Call vclears.r
    Call vbcolor.r(1)
    Call vfcolor.r(15)

    UNTIL Done = "y"
    Do
        Call vgotoxy.r(0,0)
        Print 3 lines thus
DISTRIBUTION FUNCTION OF JOB TYPES
----------------------------------

        skip 1 line

        IF n.job.type = 0
            Print 1 line thus
No data in the system yet.
        Else
            FOR Each random.e in distribution
            Print 2 lines with ivalue.a(random.e) and prob.a(random.e) thus

Type **         Cumulative probability = *.**

    Endif

        Call vgotoxy.r(15,0)
        Write as "Alter values (Y/N) => ",+
        Call rcr.r
        Read Choice as A 1
        Call vgotoxy.r(15,0)
        Call vclearl.r

        Select case Choice

        case "y", "Y"

            print 1 line thus
Enter the number of jobs to be sent through the line
            Read n.job.type
            Call vgotoxy.r(16,0)
            Call vclearl.r
            Call vgotoxy.r(15,0)
            Call vclearl.r
```

```
            Print 1 line thus
Enter the cumulative probability of each job [end with an asterix]
            Read distribution
            Call vgetxy.r yielding row,column
            LET Column = 0
            FOR increment = 15 to row
                Do
                    Call vgotoxy.r(increment,0)
                    Call vclearl.r
                Loop
            Call vclears.r

        case "N", "n"
            Done = "y"

        Endselect
    Loop

END  '' Job.schedule




ROUTINE MENU

DEFINE Choice as an ALPHA VARIABLE
DEFINE Done and Graphics as TEXT VARIABLES

LET DONE = "n"
LET Graphics = "E"
Call vbcolor.r(1)      ''  set text background to blue
Call vfcolor.r(15)     ''  set text foreground to white
LET lines.v = 0        ''  turn off 55 lines/page

   Until Done = "y"
     Do
       Call vclears.r
       PRINT 7 LINES THUS

              èëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëë£
              ¤  FLEXIBLE MANUFACTURING SMT LINE MODEL   ¤
              àëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëëë¥

                            Default values

    PRINT 1 LINE WITH Number.of.jobs(1)  THUS
           Number Per Batch = ***

    PRINT 4 LINES WITH Total.no.of.sim, MEAN.INTERARRIVAL.TIME,
                Graphics, and Timescale.v THUS
           Number of Simulations  = ***
           Job Interarrival Times are Exponential With a Mean of  ***.** Hour
           Enable/Disable Graphics Display = *
           Graphics Timescale Unit = ***
       print 10 lines thus

           1) Review the Job-Workstation Distribution
           2) Review Job Scheduling
           3) Change the Number of Simulation Runs and Report Times
           4) Enable/Disable Graphics Output
           5) Charge the Graphics Timescale Unit
```

```
           R) Run the simulation
           E) Exit the simulation

 Call vgotoxy.r(22,0)
 Write as "Enter your choice => ", +
 Call rcr.r
 Read CHOICE as A 1
 Call vgotoxy.r(22,0)
 Call vclearl.r

       Select case CHOICE


       case "1"
             Call ws.task.setup

       case "2"
            Call job.schedule

       case "3"
            Write as "Enter new Number of Workstations => ",+
            Read N.WORK.STATION
            IF N.WORK.STATION < 0
                LET N.WORK.STATION = 1
            Always
            Create every work.station
            FOR each Work.station
             DO
                Call vgotoxy.r(22,0)
                Call vclearl.r
           Write as "Enter the Number of Machines at each Workstation => ",+
           Read Ws.num.machines(work.station)
           LET U.work.station(work.station) = ws.num.machines(work.station)
            LOOP


       case "4"
           Call vgotoxy.r(22,0)
           Call vclearl.r
           Write as "Graphics Diplay Enabled/Disabled (E/D)?   => ",+
           Read Graphics

       case "5"
            Write as "Enter new Graphics Timescale => ",+
            Read Timescale.v

       case "R", "r", "G", "g"
         DONE = "y"   ''  choice made... now leave the menu

       case "E", "e", "X", "x", "Q", "q"
         Stop

       Default
       Endselect

     LOOP

        IF GraphicS <> "D"
        Call Graphics.init
        Endif

END '' MENU
```

```
ROUTINE MENUBAR.CTL Given FIELD.ID AND FORM YIELDING STATUS

DEFINE Dialog as a POINTER VARIABLE
DEFINE field.id as a TEXT VARIABLE
DEFINE field.ptr and form as POINTER VARIABLES
DEFINE icons as a 1-DIM TEXT ARRAY
DEFINE status as an INTEGER VARIABLE
DEFINE Plot1 and Plot2 as TEXT VARIABLES

    IF Field.id eq "MENU"
        LET field.id = dtval.a(dfield.f("menu",form))
    ENDIF

SELECT CASE Field.id

    Case "INITIALIZE"

    Case "BACKGROUND"
    LET field.ptr = dfield.f("menu",form)
    LET dival.a(field.ptr) = 0
    Display Field.ptr

    Case "QUIT"
    LET Status = 1

    Case "ICONS"

    Case "PNP HEAD1"

    Show dialog with "ICON.FRM"
    Reserve Icons as 2
    LET Icons(1) = "Level Meter"
    LET Icons(2) = "Line Graph"
    LET field.ptr = dfield.f("PICK",dialog)
    LET Dary.a(Field.ptr) = Icons(*)

    IF Accept.f(dialog,0) ne "cancel" AND Ddval.a(field.ptr) ne 0
        LET Icons(*) = Dary.a(Field.ptr)
        LET Plot1 = Icons(Ddval.a(Field.ptr))
    ENDIF

    IF Plot1 = "Level Meter"
         Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),1
         Display n.Q.work.station(5) with "level1.grf"
         Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),0

    ELSE Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),1
         Display N.q.work.station(5) with "trace1.grf"
         Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),0
         LET Plot1 = "T"
    ENDIF

    Case "PNP HEAD2"

    Show dialog with "ICON.FRM"
    Reserve Icons as 2
    LET Icons(1) = "Level Meter"
    LET Icons(2) = "Line Graph"
```

```
    LET field.ptr = dfield.f("PICK",dialog)
    LET Dary.a(Field.ptr) = Icons(*)

    IF Accept.f(dialog,0) ne "cancel" AND Ddval.a(field.ptr) ne 0
        LET Icons(*) = Dary.a(Field.ptr)
        LET Plot2 = Icons(Ddval.a(Field.ptr))
    ENDIF

    IF Plot2 = "Level Meter"
         Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),1
         Display n.Q.work.station(6) with "level2.grf"
         Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),0

    ELSE IF Plot1 = "T"
            Erase N.q.work.station(5)
         ENDIF
         Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),1
         Display N.q.work.station(6) with "trace2.grf"
         Call Gpriority.r giving segid.a(Dfield.f("Block2",layout)),0
    ENDIF


    Case "ERASE"

    Show Dialog with "ICON.FRM"
    LET Icons(1) = "Head1 Display"
    LET Icons(2) = "Head2 Display"
    LET field.ptr = dfield.f("PICK",dialog)
    LET Dary.a(Field.ptr) = Icons(*)

    IF Accept.f(dialog,0) ne "cancel" AND Ddval.a(field.ptr) ne 0
        LET Icons(*) = Dary.a(Field.ptr)
        LET Plot2 = Icons(Ddval.a(Field.ptr))
    ENDIF

Default
Endselect

END  '' Menuproc









ROUTINE REPORT.SETUP

''DEFINE Ans as an ALPHA VARIABLE

call vclears.r

Print 10 lines thus
                Run Time and Report Setup
                -------------------------




    Run No.     Time                Report No.          Time
    -------     ----                ----------          -----
```

```
call vgotoxy.r(21,1)
Write as "Enter the Number of Simulations (Max 4) => ", +
Read Total.no.of.sim
call vgotoxy.r(21,1)
Call vclearl.r
CALL VGOTOXY.R(3,0)
print 1 line with Total.no.of.sim thus
Number of Simulations = *

call vgotoxy.r(21,1)
Write as "Enter the Number of Repitions  => ", +
Read Rep
call vgotoxy.r(21,1)
Call vclearl.r
call vgotoxy.r(5,0)
Print 1 line with rep thus
The Number of Batch Reptitions = **

Let line = 9
For I = 1 to Total.no.of.sim
DO
    call vgotoxy.r(21,1)
    Print 1 line with I thus
Enter the Starting Time of Run No. **
    Call vgotoxy.r(21,40)
    Read runtime

    call vgotoxy.r(21,1)
    Print 1 line with I thus
Enter the Logging Time of Report No. **
    Call vgotoxy.r(21,40)
    Read logtime
    call vgotoxy.r(21,1)
    Call vclearl.r

    call vgotoxy.r(line + i,0)
    print 1 line with RUNTIME AND logtime thus
    ****          ***

    If I = 1
    Let First.run = runtime
    Let First.report = logtime
    ELSE If i = 2
    Let Second.run = runtime
    Let Second.report = logtime
    ELSE If i = 3
    Let Third.report = logtime
    Let Third.run = runtime
    Else
    Let fourth.run = runtime
    Let Fourth.report = logtime
    endif
    endif
    endif

LOOP

''        call vgotoxy.r(15,0)
  ''      write as "Alter values (Y/N) => ",+
    ''    read ANS

End
```

```
ROUTINE RESET

USE 2 FOR INPUT
    READ N.work.station
    FOR each Work.station
    DO
        READ Ws.num.machines(work.station)
        LET U.work.station(work.station) = Ws.num.machines(work.station)
        READ Conveyor.Length(work.station)
        READ Conveyor.Direction(work.station)
        READ Conveyor.Xlocation(work.station)
        READ Conveyor.Ylocation(work.station)
        READ Ws.Xlocation(work.station)
        READ Ws.Ylocation(work.station)
        READ St.Xlocation(work.station)
        READ St.Ylocation(work.station)
        READ Q.Xlocation(work.station)
        READ Q.Ylocation(work.station)
        READ Display.X(work.station)
        READ Display.Y(work.station)

   LOOP

    READ Mean.interarrival.time
    READ N.job.type
    FOR each Job.type
    DO
      READ Number.of.jobs(job.type)
        READ Num.tasks
        FOR I = 1 to Num.tasks
            DO
            READ Task.work.station(task)
            READ Task.work.time(task)
            FILE this task in Routing(job.type)
        LOOP
    LOOP

    READ Rep
    READ Total.no.of.sim
    FOR I = 1 to Total.no.of.sim
    DO
      IF I = 1
      READ First.run
      READ First.report
      ELSE
         IF I = 2
         READ Second.run
         READ Second.report
         ELSE
            IF I = 3
            READ Third.run
            READ Third.report
            ELSE
               IF I = 4
               READ Fourth.run
               READ Fourth.report
               ENDIF
           ENDIF
```

```
          ENDIF
       ENDIF
      LOOP

      READ N.Component

      FOR EACH Component
      DO
          READ Head(Component)
          READ Mounting.pos(Component)
          READ Feeder.pos(Component)
          READ Num.per.board(Component)
          READ Nozzle.size(Component)
      LOOP

      CLOSE unit 2

  END  '' Reset
```

```
Display ROUTINE queue1 Given queue1

Define queue1 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 3
Let String = CONCAT.F("Q = ",ITOT.F(Number))   ''"qrstuvxyz"
Call Textcolor.r(15)
Call Textsize.r giving 7
Call WGtext.r(string,0,0)

END  '' V.STATUS
```

```
Display ROUTINE queue10 Given queue10

Define queue10 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 3
Let String = CONCAT.F("           Part Count = ",ITOT.F(Number3))
Call Textcolor.r(15)
Call Textsize.r giving 75
Call WGtext.r(string,0,0)
```

```
END  '' V.STATUS





Display ROUTINE Queue3 Given Queue3

Define Queue3 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 3
    Let String = CONCAT.F("Q = ",ITOT.F(Number))
Call Textcolor.r(15)
Call Textsize.r giving 586
Call WGtext.r(string,0,0)

END  '' V.STATUS




Display ROUTINE Queue5 Given Queue5

Define Queue5 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 3
Let String = CONCAT.F("Q = ",ITOT.F(Number1))
Call Textcolor.r(15)
Call Textsize.r giving 575    ''86
Call WGtext.r(string,0,0)

END  '' V.STATUS





Display ROUTINE queue6 Given queue6

Define queue6 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 3
Let String = CONCAT.F("Q = ",ITOT.F(Number2))
Call Textcolor.r(15)
```

```
Call Textsize.r giving 575    ''86
Call WGtext.r(string,0,0)

END  '' V.STATUS
```

```
Display ROUTINE Status1 Given STATUS1

Define Status1 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 2
IF Description = "Idle"
    Let String = "I"
ELSE
    Let String = "B"
ENDIF
Call Textcolor.r(15)
Call Textsize.r giving 586
Call WGtext.r(string,0,0)

END  '' V.STATUS
```

```
Display ROUTINE Status10 Given STATUS10

Define Status10 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 2

IF Description = "Idle"
LET String = "I"
ELSE
Let String = "B"
ENDIF

Call Textcolor.r(15)
Call Textsize.r giving 586
Call WGtext.r(string,0,0)

END  '' V.STATUS
```

```
Display ROUTINE Status3 Given STATUS3

Define Status3 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 2

IF Description = "Idle"
LET String = "I"
ELSE
Let String = "B"
ENDIF

Call Textcolor.r(15)
Call Textsize.r giving 586
Call WGtext.r(string,0,0)

END  '' V.STATUS
```

```
Display ROUTINE Status5 Given STATUS5

Define Status5 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 2

IF Description = "Idle"
LET String = "Idle"
ELSE
Let String = "Busy"
ENDIF

Call Textcolor.r(15)
Call Textsize.r giving 586
Call WGtext.r(string,0,0)

END  '' V.STATUS
```

```
Display ROUTINE Status6 Given STATUS6

Define Status6 as a POINTER Variable

Define String as a TEXT VARIABLE

Let VXFORM.V = 2

IF Description = "Idle"
LET String = "Idle"
ELSE
Let String = "Busy"
ENDIF

Call Textcolor.r(15)
Call Textsize.r giving 586
Call WGtext.r(string,0,0)


END  '' V.STATUS
```

```
ROUTINE WS.TASK.SETUP

define Done and Display as text variables
define Choice as an alpha variable

let done = "n"
  until Done = "y"
    do
    call vclears.r

    If task > 0
    Let display = "T"

        FOR EACH JOB.TYPE
        DO
            PRINT 5 LINES WITH JOB.TYPE THUS
JOB TYPE ** ROUTING

Station     Mean Service Time (in Hours)
-------     ----------------------------

            FOR EACH TASK IN ROUTING(JOB.TYPE)
            PRINT 2 LINES WITH TASK.WORK.STATION(TASK) AND
                           TASK.WORK.TIME(TASK) THUS
  **                  ****.**


        LOOP
    endif

        If Display = "T"
        call vgetxy.r yielding row,column
        call vgotoxy.r(row +2,0)
        Endif
```

```
        write as "Alter data values (Y/N) => ",+
        call rcr.r
        read choice as a 1

        select case Choice

        case "Y", "y"

        call vclears.r

        ''CREATE EVERY JOB.TYPE
        FOR EACH JOB.TYPE
        DO
            PRINT 1 LINE WITH JOB.TYPE THUS
Enter the number of tasks for job type No ***
            READ NUM.TASKS      ''Number of tasks for a particular job type
            FOR I = 1 TO NUM.TASKS      '' Declared locally
            DO
                CREATE A TASK

                call vgetxy.r  yielding row,column
                if row > 23
                    call vclears.r
                endif
                PRINT 1 LINE WITH I THUS
For task No ***, enter the work station number and the mean service time
                READ TASK.WORK.STATION(TASK) and TASK.WORK.TIME(TASK)
                    FILE THIS TASK IN ROUTING(JOB.TYPE)
           LOOP
        LOOP

        case "N", "n"
        Done = "y"
      endselect
    loop

END
```

Appendix 14

```
SET HEADING OFF
SET SAFETY OFF
SET TALK OFF
CLEAR ALL
SET CONSOLE ON
SET DEVICE TO SCREEN
SET PRINT OFF
SET DELIMITERS ON
SET DELIMITERS TO "[]"
SET STATUS OFF
SET BELL OFF
close all
CLEAR
DEFINE WINDOW PARAMETER FROM 6,5 TO 14,73 DOUBLE
ACTIVATE WINDOW PARAMETER
@ 2,8 SAY 'FLEXIBLE MANUFACTURING SIMULATION AND OPTIMIZATION'
@ 4,8 SAY '      Copyright 1989, Standard Telephones & Cables'
@ 6,8 say '                              Written By M. Chodos'
WAIT ''
RELEASE WINDOW PARAMETER

DO MAIN
```

```
* COMPSELECT PROCEDURE

PARAMETERS NAME,NOZZLE

SELECT 6

DEFINE POPUP MAINMENU FROM 7,45 MESSAGE "Place Highlighted Bar Over Choice an
DEFINE BAR 1 OF MAINMENU PROMPT "STANDARD SMALL TOOLING SET"
DEFINE BAR 2 OF MAINMENU PROMPT "SPECIAL TOOLING SET I"
DEFINE BAR 3 OF MAINMENU PROMPT "SPECIAL TOOLING SET II"
DEFINE BAR 4 OF MAINMENU PROMPT "         EXIT"

ON SELECTION POPUP MAINMENU DO MAINMEN

DEFINE POPUP COMPONENT FROM 7,45 MESSAGE "Place Highlighted Bar Over Choice a
DEFINE BAR 1 OF COMPONENT PROMPT "SOT 23"
DEFINE BAR 2 OF COMPONENT PROMPT "SOT 89, SOT143"
DEFINE BAR 3 OF COMPONENT PROMPT "CHIP CAPS - 0805,0504,1005,1505"
DEFINE BAR 4 OF COMPONENT PROMPT "CHIP CAPS - 1206,1210,1505"
DEFINE BAR 5 OF COMPONENT PROMPT "CHIP CAPS - 1825, 2225"
DEFINE BAR 6 OF COMPONENT PROMPT "SOIC's - 8, 14, 14L, 16"
DEFINE BAR 7 OF COMPONENT PROMPT "SOIC's - 16L, 18, 18L, 20, 20L, 24"
DEFINE BAR 8 OF COMPONENT PROMPT "DIP (4 PIN)"
DEFINE BAR 9 OF COMPONENT PROMPT "PLCC - 18, 20, 28, 32"
DEFINE BAR 10 OF COMPONENT PROMPT "LCC - 16, 20, 24, 28, 32, 36, 40"
DEFINE BAR 11 OF COMPONENT PROMPT "CYLINDRICALS"
DEFINE BAR 12 OF COMPONENT PROMPT "MELF - 14, 18, TYPE"
DEFINE BAR 13 OF COMPONENT PROMPT "SOD 80"
DEFINE BAR 14 OF COMPONENT PROMPT "MLL 34, 41"
DEFINE BAR 15 OF COMPONENT PROMPT "TO 92"
DEFINE BAR 16 OF COMPONENT PROMPT "LED HLMP 6000/6001"
DEFINE BAR 17 OF COMPONENT PROMPT "POT 10K433B"
DEFINE BAR 18 OF COMPONENT PROMPT "EXIT TO MAIN MENU"
ON SELECTION POPUP COMPONENT DO COMP

DEFINE POPUP COMPI FROM 7,45 MESSAGE "Place Highlighted Bar Over Choice and H
DEFINE BAR 1 OF COMPI PROMPT "SOIC's - 14, 16, 18"
DEFINE BAR 2 OF COMPI PROMPT "SOIC's - 18L,20,20L,24,28,28L"
DEFINE BAR 3 OF COMPI PROMPT "PLCC - 18"
DEFINE BAR 4 OF COMPI PROMPT "SOT - 136A, 190"
DEFINE BAR 5 OF COMPI PROMPT "SOL - 24, 28, 40"
DEFINE BAR 6 OF COMPI PROMPT "SOW - 24, 56"
DEFINE BAR 7 OF COMPI PROMPT "PLCC - 28, 32"
DEFINE BAR 8 OF COMPI PROMPT "LCC - 32"
DEFINE BAR 9 OF COMPI PROMPT "EXIT TO MAIN MENU"
ON SELECTION POPUP COMPI DO COMPONENTI

DEFINE POPUP COMPII FROM 7,45 MESSAGE "Place Highlighted Bar Over Choice and
DEFINE BAR 1 OF COMPII PROMPT "PLCC - 28, 32, 44"
DEFINE BAR 2 OF COMPII PROMPT "PLCC - 68, 84"
DEFINE BAR 3 OF COMPII PROMPT "SOL - 24, 40"
DEFINE BAR 4 OF COMPII PROMPT "SOW - 24"
DEFINE BAR 5 OF COMPII PROMPT "LCC - 36, 40, 44, 48"
DEFINE BAR 6 OF COMPII PROMPT "TINY BELL IC"
DEFINE BAR 7 OF COMPII PROMPT "EXIT TO MAIN MENU"
ON SELECTION POPUP COMPII DO COMPII

ACTIVATE POPUP MAINMENU

RELEASE POPUPS MAINMENU
RELEASE POPUPS COMPONENT
RELEASE POPUPS COMPI
RELEASE POPUPS COMPII
RETURN

*------------------------------------------------------------------------
```

```
        CASE BAR() = 1
        ACTIVATE POPUP COMPONENT

        CASE BAR() = 2
        ACTIVATE POPUP COMPI

        CASE BAR() = 3
        ACTIVATE POPUP COMPII

        CASE BAR() = 4
        DEACTIVATE POPUP
ENDCASE

RETURN

*----------------------------------------

PROCEDURE COMP

DO CASE
        CASE BAR() = 1
        STORE 'SOT 23' TO NAME
        STORE 0.042 TO NOZZLE

        CASE BAR() = 2
        STORE 'SOT 89/SOT 143' TO NAME
        STORE 0.042 TO NOZZLE

        CASE BAR() = 3
        STORE 'CHIP CAPS 0805..1505' TO NAME
        STORE 0.042 TO NOZZLE

        CASE BAR() = 4
        STORE 'CHIP CAPS 1206..1505' TO NAME
        STORE 0.058 TO NOZZLE

        CASE BAR() = 5
        STORE 'CHIP CAPS 1825/2225' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 6
        STORE 'SOICs 8/14/14L/16' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 7
        STORE 'SOICs 16L..24' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 8
        STORE 'DIP (4 PIN)' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 9
        STORE 'PLCC 18/20/28/32' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 10
        STORE 'LCC 16..40' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 11
        STORE 'CYLINDRICALS' TO NAME
        STORE 0.042 TO NOZZLE

        CASE BAR() = 12
        STORE 'MELF 14/18' TO NAME
        STORE 0.042 TO NOZZLE
```

```
        STORE 0.042 TO NOZZLE

        CASE BAR() = 14
        STORE 'MLL 34/41' TO NAME
        STORE 0.042 TO NOZZLE

        CASE BAR() = 15
        STORE 'TO 92' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 16
        STORE 'LED HLMP 6000/6001' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 17
        STORE 'POT 10K4338' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 18
        DEACTIVATE POPUP

ENDCASE

@ 4,35 SAY '['+(NAME)+'' COLOR R+

        IF BAR() <> 18
                DEACTIVATE POPUP
        ENDIF

RETURN
*-----------------------------------------
PROCEDURE COMPONENTI

DO CASE
        CASE BAR() = 1
        STORE 'SOICs 14/16/18' TO NAME
        STORE 0.085 TO NOZZLE

        CASE BAR() = 2
        STORE 'SOICs 18L..28L' TO NAME
        STORE 0.225 TO NOZZLE

        CASE BAR() = 3
        STORE 'PLCC - 18' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 4
        STORE 'SOT 136A,190' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 5
        STORE 'SOL 24/28/40' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 6
        STORE 'SOW 24/56' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 7
        STORE 'PLCC 28/32' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 8
        STORE 'LCC - 32' TO NAME
        STORE 0.109 TO NOZZLE

        CASE BAR() = 9
```

```
        IF BAR() <> 9
                DEACTIVATE POPUP
        ENDIF

RETURN

*-------------------------------------------

PROCEDURE COMPII

DO CASE
        CASE BAR() = 1
        STORE 'PLCC 28/32/44' TO NAME
        STORE 0.208 TO NOZZLE

        CASE BAR() = 2
        STORE 'PLCC 68/84' TO NAME
        STORE 0.208 TO NOZZLE

        CASE BAR() = 3
        STORE 'SOL 24,40' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 4
        STORE 'SOW 24' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 5
        STORE 'LCC 36/40/44/48' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 6
        STORE 'TINY BELL IC' TO NAME
        STORE 0.148 TO NOZZLE

        CASE BAR() = 7
        DEACTIVATE POPUP
ENDCASE
@ 4,35 SAY '['+(NAME)+'' COLOR R+


        IF BAR() <> 7
                DEACTIVATE POPUP
        ENDIF

RETURN

*---------------------------------------------
```

```
*------------- PROCEDURE DELETE -------------------------
close all
SET DELETE ON

SELECT 1
USE PARAMETE
SELECT 2
USE REPORT
SELECT 3
USE TASK
SELECT 4
USE HEAD
SELECT 5
USE COMP

select a

DEFINE WINDOW PARAMETER FROM 4,3 TO 15,73 DOUBLE
ACTIVATE WINDOW PARAMETER

DEFINE POPUP FILE FROM 1,50 MESSAGE "Place Highlighted Bar Over Choice and Hi
DEFINE BAR 1 OF FILE PROMPT "COMPONENTS"
DEFINE BAR 2 OF FILE PROMPT "PLACEMENT HEADS"
DEFINE BAR 3 OF FILE PROMPT "GEN PARAMETERS"
DEFINE BAR 4 OF FILE PROMPT "REPORTS"
DEFINE BAR 5 OF FILE PROMPT "JOB TASKS"
DEFINE BAR 6 OF FILE PROMPT "  EXIT"

ON SELECTION POPUP FILE DO FILESEL

@ 4,2 SAY 'SELECT DATABASE TO DELETE:'

ACTIVATE POPUP FILE

RELEASE POPUPS NOZZLE
RELEASE WINDOW PARAMETER
SET DELETE OFF

RETURN

*---------------------------------------

PROCEDURE FILESEL
STORE ' ' TO BASE
DO CASE
        CASE BAR() = 1
        SELECT COMP

        CASE BAR() = 2
        SELECT HEAD

        CASE BAR() = 3
        SELECT PARAMETE

        CASE BAR() = 4
        SELECT REPORT

        CASE BAR() = 5
        SELECT TASK

        CASE BAR() = 6
        DEACTIVATE POPUP
ENDCASE

GO TOP
```

```
@ 6,2 CLEAR
STORE 'N' TO MSELECT
@ 4,2 SAY 'ARE YOU SURE YOU WANT TO DELETE DATABASE (Y/N):' GET MSELECT PICTU
READ
IF MSELECT = 'Y'
PACK
@ 6,1 SAY "DATABASE DELETED"
ENDIF

RETURN
```

```
* NOZSELECT PROCEDURE

PARAMETER FEEDER

SELECT 6

DEFINE POPUP NOZZLE FROM 5,50 MESSAGE "Place Highlighted Bar Over Choice and
DEFINE BAR 1 OF NOZZLE PROMPT "MATRIX TRAY"
DEFINE BAR 2 OF NOZZLE PROMPT "TAPE FEEDER"
DEFINE BAR 3 OF NOZZLE PROMPT "MAGAZINE FEEDER"
DEFINE BAR 4 OF NOZZLE PROMPT "  EXIT"

ON SELECTION POPUP NOZZLE DO TEST

ACTIVATE POPUP NOZZLE

RELEASE POPUPS NOZZLE
RETURN

*-----------------------------------------

PROCEDURE TEST

DO CASE
        CASE BAR() = 1
        STORE 'MATRIX TRAY' TO FEEDER

        CASE BAR() = 2
        STORE 'TAPE FEEDER' TO FEEDER

        CASE BAR() = 3
        STORE 'MAGAZINE FEEDER' TO FEEDER

        CASE BAR() = 4
        DEACTIVATE POPUP
ENDCASE
@ 4,28 SAY '[' +(FEEDER)+ ']' COLOR R+

IF BAR() <> 4
        DEACTIVATE POPUP
ENDIF

RETURN
*------------------------------------------
```

```
* 05/01/90
*
STORE 'F' TO LEFT,DONE
STORE 'F' TO COMPCHECK, PNPFULL
STORE 1 TO B1COUNT, B2COUNT, E1COUNT, E2COUNT
STORE 'F' TO BASE1, BASE2, EIM1, EIM2
STORE 'F' TO FIRST_FIND, SECOND_FIND, THIRD_FIND, FOURTH_FIND
STORE 'F' TO EIM1FULL, EIM2FULL, BASE1FULL
STORE 0 TO HEAD1WK, HEAD2WK, HD, PERCENT,COMPNUM1, COMPNUM2, GLUETIME
STORE 0.03455 TO EIMTIME
STORE 0.03083 TO BASETIME

CLOSE ALL
SELECT 7
USE COMP
SELECT 8
USE HEAD

*----------------- CALCULATE GLUEING TIMES ----------------------

CLEAR
DEFINE WINDOW PARAMETER FROM 7,1 TO 14,75 DOUBLE COLOR R+*
ACTIVATE WINDOW PARAMETER

@ 3,8 SAY 'CALCULATING GLUEING TIMES.. PLEASE WAIT' COLOR W+

SELECT COMP
STORE 'F' TO DONE
INDEX ON SIDE TO NEW
GO TOP
SEEK 'BOTTOM'

DO WHILE DONE = 'F'

        DO CASE
                CASE COMP_NAME =  'SOT 23'
                STORE 0.042 TO GLUETIME

                CASE COMP_NAME =  'SOT 89/SOT 143'
                STORE 0.042 TO GLUETIME

                CASE COMP_NAME =  'CHIP CAPS 0805..1505'
                STORE 0.042 TO GLUETIME

                CASE COMP_NAME =  'CHIP CAPS 1206..1505'
                STORE 0.058 TO GLUETIME

                CASE COMP_NAME = 'CHIP CAPS 1825/2225'
                STORE 0.143 TO GLUETIME

                CASE COMP_NAME =  'SOICs 8/14/14L/16'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME =  'SOICs 16L..24'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME =  'DIP (4 PIN)'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME =  'PLCC 18/20/28/32'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME =  'LCC 16..40'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME =  'CYLINDRICALS'
```

```
                STORE 0.042 TO GLUETIME

                CASE COMP_NAME = 'SOD 80'
                STORE 0.042 TO GLUETIME

                CASE COMP_NAME = 'MLL 34/41'
                STORE 0.042 TO GLUETIME

                CASE COMP_NAME =  'TO 92'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME = 'LED HLMP 6000/6001'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME = 'POT 10K4338'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME = 'SOICs 14/16/18'
                STORE 0.085 TO GLUETIME

                CASE COMP_NAME = 'SOICs 18L..28L'
                STORE 0.225 TO GLUETIME

                CASE COMP_NAME = 'PLCC - 18'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME = 'SOT 136A,190'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME = 'SOL 24/28/40'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME = 'SOW 24/56'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME =  'PLCC 28/32'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME =  'LCC - 32'
                STORE 0.109 TO GLUETIME

                CASE COMP_NAME = 'PLCC 28/32/44'
                STORE 0.208 TO GLUETIME

                CASE COMP_NAME = 'PLCC 68/84'
                STORE 0.208 TO GLUETIME

                CASE COMP_NAME = 'SOL 24,40'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME = 'SOW 24'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME = 'LCC 36/40/44/48'
                STORE 0.148 TO GLUETIME

                CASE COMP_NAME = 'TINY BELL IC'
                STORE 0.148 TO GLUETIME

        ENDCASE

IF .NOT. EOF()
        SKIP
ELSE
        STORE 'T' TO DONE
ENDIF
```

```
ENDDO

STORE GLUETIME/BASETIME TO COMPNUM1
* STORE GLUETIME TO HEAD1WK

@ 3,0 CLEAR
RELEASE WINDOW PARAMETER

DEFINE WINDOW PARAMETER FROM 7,1 TO 17,75
ACTIVATE WINDOW PARAMETER
@ 3,2 SAY 'GLUE-TIME CALCULATION COMPLETE'
@ 5,2 SAY 'TOTAL GLUEING TIME REQUIRED FOR BOARD = '+STR(GLUETIME,4,2)+' SECO
WAIT ""
*--------------- END OF GLUEING PROCEDURE ----------------------


CLEAR
STORE 1 TO MSELECT
@ 2,2 SAY 'OPTIMIZATION METHOD 1 - BASED ON NUMBER OF COMPONENTS PER BOARD'
@ 4,2 SAY 'OPTIMIZATION METHOD 2 - BASED ON PRIORITIZED NOZZLE PLACEMENT'
@ 6,2 SAY 'SELECT OPTION => 'GET MSELECT PICTURE "9"
READ
IF MSELECT = 1
        RELEASE WINDOW PARAMETER
        DO METHOD1
ELSE
        RELEASE WINDOW PARAMETER
        DO METHOD2
ENDIF
CLOSE ALL
RETURN

*-------------------END OF MAIN PROCEDURE --------------------

*------------------- PROCEDURE METHOD1 -----------------------

PROCEDURE METHOD1

* SELECT 6
CLEAR
 DEFINE WINDOW PARAMETER FROM 7,3 TO 14,73 DOUBLE COLOR R+*
 ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '  - DO NOT INTERRUPT -' COLOR W+

*----------------- SET FLAG MARKERS -------

SELECT COMP
GO TOP
DO WHILE .NOT. EOF()
REPLACE MARKER WITH 'F'
SKIP
ENDDO

*---------------- STARTING OPTIMIZATION ROUTINE ---------------

INDEX ON NOZZ_SIZE TO NEW
STORE 0 TO COUNT
GO TOP

DO WHILE .NOT. EOF()

STORE NOZZ_SIZE TO X

        IF MARKER = 'F'
```

```
                                STORE 'F' TO EIM1
                                STORE 'T' TO EIM2
                                DO CHECKEIM1
                                STORE COMPNUM1 + NUM_PER_BD TO COMPNUM1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO FIRST_FIND
                        ELSE
                        IF SECOND_FIND = 'F'
                                DO WHILE NOZZ_SIZE = X
                                STORE 'T' TO EIM1
                                STORE 'F' TO EIM2
                                DO CHECKEIM2
                                STORE COMPNUM2 + NUM_PER_BD TO COMPNUM2
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO SECOND_FIND
                        ELSE
                                IF THIRD_FIND = 'F' .AND. COMPNUM1 =< COMPNUM
* NB !!!!  HEAD1WK <= HEAD2WK
                                DO WHILE NOZZ_SIZE = X
                                STORE 'F' TO EIM1
                                STORE 'T' TO EIM2
                                DO CHECKEIM1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO EIM1FULL
                                STORE 'T' TO THIRD_FIND
                        ELSE
                        IF FOURTH_FIND = 'F'
                                DO WHILE NOZZ_SIZE = X
                                STORE 'T' TO EIM1
                                STORE 'F' TO EIM2
                                DO CHECKEIM2
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO EIM2FULL
                                STORE 'T' TO FOURTH_FIND
                        ELSE
                                DO CHECKBASE1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                        ENDIF
                        ENDIF
                        ENDIF
                        ENDIF
                ELSE
                        DO CHECKBASE1
                        IF .NOT. EOF()
                                SKIP
                        ENDIF

                ENDIF
        ELSE
                SKIP
        ENDIF
ENDDO
```

```
REPLACE HEAD2_TIME WITH HEAD2WK

CLOSE ALL
STORE 4 TO X
DO DELAY WITH X

DO OPTREPORT
IF PERCENT < 95
STORE PERCENT TO NEWPERCENT

DEFINE WINDOW PARAMETER FROM 8,1 TO 14,75 DOUBLE
ACTIVATE WINDOW PARAMETER
CLEAR
STORE 'N' TO MSELECT
@ 1,2 SAY 'OPTIMIZATION PERCENTAGE LESS THAN 95%'
@ 3,2 SAY 'RE-OPTIMIZE TO IMPROVE OPTIMIZATION PERCENTAGE (Y/N) => ' GET MSEL
READ
CLEAR
STORE 'N' TO SELECT
@ 2,2 SAY 'PRINT OUT RESULTS FROM INITIAL OPTIMIZATION (Y/N) => ' GET SELECT
READ
IF SELECT = 'Y'
        DO PRINTOPT
ENDIF
RELEASE WINDOW PARAMETER
ENDIF

IF MSELECT = 'Y'
        DO REOPT
ENDIF

RETURN

*----------------------- END PROCEDURE METHOD1 ----------------------

*---------------------- PROCEDURE CHECKBASE1 ------------------------

PROCEDURE CHECKBASE1

IF BASE1 = 'F'
        IF B1COUNT <= 20
                        DO SETBASE1
        ELSE
                        DO CHKBAS2
        ENDIF

ELSE
        DO CHKBAS2

ENDIF

RETURN
*-------------------------------------------------------------------------

*-------------------- PROCEDURE CHKBAS2 --------------------------

PROCEDURE CHKBAS2

IF BASE2 = 'F'
        IF B2COUNT <= 20
                DO SETBASE2
        ELSE
                STORE 'T' TO COMPCHECK
                DO CHECKEIM1
        ENDIF
ELSE
        STORE 'T' TO COMPCHECK
```

```
RETURN

*-----------------------------------------------------------------------
*----------------- PROCEDURE CHECKEIM1 ---------------------------------

PROCEDURE CHECKEIM1

IF EIM1 = 'F'
        IF E1COUNT <= 43
                DO SETEIM1
        ELSE
                DO CHECKEIM2
        ENDIF

ELSE
        DO CHECKEIM2
ENDIF

RETURN

*-----------------------------------------------------------------------
*----------------- PROCEDURE CHECKEIM2 ---------------------------------

PROCEDURE CHECKEIM2

IF EIM2  = 'F'
        IF E2COUNT <= 43
                DO SETEIM2
        ELSE
                IF COMPCHECK = 'T'
                        STORE 'T' TO PNPFULL
                ELSE
                        DO CHECKBASE1
                ENDIF
        ENDIF
ELSE
        DO CHECKBASE1
ENDIF

RETURN

*-----------------------------------------------------------------------
*----------------- PROCEDURE SETEIM1 -----------------------------------

PROCEDURE SETEIM1

        REPLACE FEEDER_POS WITH E1COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'E'
        REPLACE PNP_HEAD WITH 1
        STORE 'T' TO EIM1
        STORE E1COUNT + 1 TO E1COUNT
        STORE (NUM_PER_BD * EIMTIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD1WK TO HEAD1WK
RETURN

*-----------------------------------------------------------------------

*-----------------  PROCEDURE SETEIM2 -----------------------------------

PROCEDURE SETEIM2

        REPLACE FEEDER_POS WITH E2COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'E'
        REPLACE PNP_HEAD WITH 2
```

```
        STORE WORKTIME + HEAD2WK TO HEAD2WK
RETURN

*--------------------------------------------------------------------------

*--------------------- PROCEDURE SETBASE1 ----------------------------

PROCEDURE SETBASE1

        REPLACE FEEDER_POS WITH B1COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'B'
        REPLACE PNP_HEAD WITH 1
        STORE 'T' TO BASE1
        STORE B1COUNT + 1 TO B1COUNT
        STORE (NUM_PER_BD * BASETIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD1WK TO HEAD1WK
RETURN

*--------------------------------------------------------------------------

*--------------------- PROCEDURE SETBASE2      -------------------------

PROCEDURE SETBASE2

        REPLACE FEEDER_POS WITH B2COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'B'
        REPLACE PNP_HEAD WITH 2
        STORE 'F' TO BASE1
        STORE B2COUNT + 1 TO B2COUNT
        STORE (NUM_PER_BD * BASETIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD2WK TO HEAD2WK
RETURN

*--------------------------------------------------------------------------

PROCEDURE DELAY
PARAMETER TIME
START_TIME = TIME()
TIME1 = VAL(SUBSTR(START_TIME,1,2))*3600+;
        VAL(SUBSTR(START_TIME,4,2))*60+;
        VAL(SUBSTR(START_TIME,7,2))
TIME2 = 0

DO WHILE TIME2 - TIME1 < X
STORE TIME() TO END_TIME
TIME2 = VAL(SUBSTR(END_TIME,1,2))*3600+;
        VAL(SUBSTR(END_TIME,4,2))*60+;
VAL(SUBSTR(END_TIME,7,2))
ENDDO
RETURN

*--------------------- PROCEDURE OPTREPORT ---------------------------

PROCEDURE OPTREPORT

 DEFINE WINDOW PARAMETER FROM 8,1 TO 14,75 DOUBLE
 ACTIVATE WINDOW PARAMETER

STORE 0 TO TOTCOMP,BASE1COUNT, BASE2COUNT, EIM1COUNT, EIM2COUNT
STORE 0 TO HD

RELEASE WINDOW PARAMETER
```

```
USE COMP
INDEX ON NOZZ_SIZE TO NUM
CLEAR

@ 1,0

TEXT
DESCRIPTION          QUNTY  HEAD  FEEDER POS. BASE/EIM  NOZZLE  W.TIME
----------------------------------------------------------------------
ENDTEXT

GO TOP
DO WHILE .NOT. EOF()
@ X,0 SAY COMP_NAME
@ X,18 SAY NUM_PER_BD
@ X,28 SAY PNP_HEAD
@ X,36 SAY FEEDER_POS
@ X,49 SAY MOUNT_POST
@ X,55 SAY NOZZ_SIZE
@ X,63 SAY WK_TIME

STORE TOTCOMP + NUM_PER_BD TO TOTCOMP
IF MOUNT_POST = 'B' .AND. PNP_HEAD = 1
        STORE BASE1COUNT + 1 TO BASE1COUNT
ELSE
        IF (MOUNT_POST = 'B') .AND. (PNP_HEAD = 2)
                STORE BASE2COUNT + 1 TO BASE2COUNT
        ENDIF
ENDIF

IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 1)
        STORE EIM1COUNT + 1 TO EIM1COUNT
ELSE
        IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 2)
                STORE EIM2COUNT + 1 TO EIM2COUNT
        ENDIF
ENDIF

SKIP
STORE X + 1 TO X

IF X = 18
        STORE 5 TO X
ENDIF
ENDDO

@ 18,1
WAIT "Hit any key to continue"

RELEASE WINDOW PARAMETER
DEFINE WINDOW PARAMETER FROM 2,1 TO 22,75 DOUBLE
ACTIVATE WINDOW PARAMETER

CLEAR
IF HEAD1WK > HEAD2WK
        STORE (HEAD2WK/HEAD1WK)*100 TO PERCENT
        STORE 1 TO HD
ELSE
        STORE (HEAD1WK/HEAD2WK)*100 TO PERCENT
        STORE 2 TO HD
ENDIF
TEXT
                                   HEAD1                HEAD2
                                   -----                -----
ENDTEXT
@ 3,1 SAY 'TOTAL NO. OF COMPONENTS ON BASE = '
```

```
@ 4,56 GET EIM2COUNT PICTURE "999"
@ 6,1 SAY 'TOTAL NUMBER OF COMPONENTS = ' GET TOTCOMP PICTURE "9999"
@ 8,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD1 = ' GET HEAD1WK PICTURE "999.99"
@ 8,44 SAY 'Minutes'
@ 10,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD2 = ' GET HEAD2WK PICTURE "999.99"
@ 10,44 SAY 'Minutes'
@ 12,1 SAY 'OPTIMIZATION PERCENTAGE = ' GET PERCENT PICTURE "999.9"
@ 12,36 SAY '%'
@ 16,1
WAIT "Hit any key to continue"

RELEASE WINDOW PARAMETER

RETURN

*---------------------- PROCEDURE REOPT ------------------------------

PROCEDURE REOPT

IF HD = 1

        STORE 0 TO HD
        DO REOPT1
        DO OPTREPORT
        DEFINE WINDOW PARAMETER FROM 6,1 TO 14,75 DOUBLE
        ACTIVATE WINDOW PARAMETER
        CLEAR
        IF NEWPERCENT > PERCENT
        @ 2,2 SAY 'NO IMPROVEMENT IN OPTIMIZATION PERCENTAGE '
        ENDIF
        STORE 'N' TO MSELECT
        @ 4,2 SAY 'PRINT OUT RE-OPTIMIZATION RESULTS (Y/N) =>' GET MSELECT PI
        READ
        IF MSELECT = 'Y'
                DO PRINTOPT
        ENDIF
        RELEASE WINDOW PARAMETER

ELSE
IF HD = 2
        STORE 0 TO HD
        DO REOPT2
        DO OPTREPORT
        DEFINE WINDOW PARAMETER FROM 6,1 TO 14,75 DOUBLE
        ACTIVATE WINDOW PARAMETER
        CLEAR
        IF NEWPERCENT > PERCENT
        @ 2,2 SAY 'NO IMPROVEMENT IN OPTIMIZATION PERCENTAGE '
        ENDIF
        STORE 'N' TO MSELECT
        @ 4,2 SAY 'PRINT OUT RE-OPTIMIZATION RESULTS (Y/N) =>' GET MSELECT PI
        READ
        IF MSELECT = 'Y'
                DO PRINTOPT
        ENDIF
        RELEASE WINDOW PARAMETER

ENDIF
ENDIF

RETURN

*---------------------PROCEDURE REOPT1-------------------------------------

PROCEDURE REOPT1
* TRANSFERRING FEEDER FROM HEAD 1 TO HEAD 2
```

```
ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'RE-OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '    - DO NOT INTERRUPT -' COLOR W+

SELECT COMP

STORE 'F' TO DONE,FOUND ,FLAG,EF

SORT TO ASCEND ON NUM_PER_BD
* INDEX ON MOUNT_POST TO NUM
USE ASCEND
GO TOP

DO WHILE FOUND = 'F'
IF MOUNT_POST = 'B' .AND. PNP_HEAD = 1
        REPLACE FEEDER_POS WITH B2COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'B'
        REPLACE PNP_HEAD WITH 2
        STORE B1COUNT - 1 TO B1COUNT
        STORE B2COUNT + 1 TO B2COUNT
        STORE (NUM_PER_BD * BASETIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD2WK TO HEAD2WK
        STORE HEAD1WK - WORKTIME TO HEAD1WK
* FORCE RE-OPTIMIZING UNLESS THE FOLLOWING CONDITION IS SATISFIED
        IF HEAD1WK =< HEAD2WK
                STORE 'T' TO FOUND
        ENDIF
ELSE
        SKIP
ENDIF
        IF EOF()
                STORE 'T' TO FOUND
        ENDIF
ENDDO

* RESTORING DATABASE TO ORIGINAL POSITION
COPY TO COMP

STORE 6 TO X
DO DELAY WITH X

RELEASE WINDOW PARAMETER

RETURN

*-------------------- PROCEDURE REOPT2 ------------------------------

PROCEDURE REOPT2
                                   * TRANSFERRING FEEDER FROM HEAD 2 TO HEAD 1
SELECT 6
CLEAR
DEFINE WINDOW PARAMETER FROM 7,3 TO 14,73 DOUBLE COLOR R+*
ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'RE-OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '    - DO NOT INTERRUPT -' COLOR W+

SELECT COMP

STORE 'F' TO DONE,FOUND ,FLAG,EF
STORE ' ' TO TEMP, NAM
STORE 0 TO COUNT
SORT TO ASCEND ON NUM_PER_BD
* INDEX ON MOUNT_POST TO NUM
```

```
DO WHILE FOUND = 'F'
IF MOUNT_POST = 'B' .AND. PNP_HEAD = 2

        REPLACE FEEDER_POS WITH B1COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'B'
        REPLACE PNP_HEAD WITH 1
        STORE B2COUNT - 1 TO B2COUNT
        STORE B1COUNT + 1 TO B1COUNT
        STORE (NUM_PER_BD * BASETIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD1WK TO HEAD1WK
        STORE HEAD2WK - WORKTIME TO HEAD2WK
* FORCE REOPTIMIZATION UNTIL OPT SOLUTION OBTAINED
        IF HEAD2WK =< HEAD1WK
                STORE 'T' TO FOUND
        ENDIF
ELSE
        SKIP
ENDIF
IF EOF()
        STORE 'T' TO FOUND
ENDIF
ENDDO

* RESTORING DATABASE TO ORIGINAL POSITION
COPY TO COMP

STORE 6 TO X
DO DELAY WITH X

RELEASE WINDOW PARAMETER

RETURN



*------------------------- PROCEDURE METHOD2 ----------------------------

PROCEDURE METHOD2


SELECT 6
CLEAR
DEFINE WINDOW PARAMETER FROM 7,3 TO 14,73 DOUBLE COLOR R+*
ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '  - DO NOT INTERRUPT -' COLOR W+

SELECT COMP
GO TOP
DO WHILE .NOT. EOF()
REPLACE MARKER WITH 'F'
SKIP
ENDDO

*------ STARTING OPTIMIZATION ROUTINE ---------------

SORT TO DESC ON NUM_PER_BD/D
USE DESC
STORE 0 TO COUNT
GO TOP

DO WHILE .NOT. EOF()

STORE NOZZ_SIZE TO X
```

```
REPLACE HEAD2_TIME WITH HEAD2WK


CLOSE ALL
STORE 4 TO X
DO DELAY WITH X

RELEASE WINDOW PARAMETER

DO OPTREPORT

IF PERCENT < 95
        STORE PERCENT TO NEWPERCENT

        DEFINE WINDOW PARAMETER FROM 8,1 TO 14,75 DOUBLE
        ACTIVATE WINDOW PARAMETER
        CLEAR
        STORE 'N' TO MSELECT
@ 1,2 SAY 'OPTIMIZATION PERCENTAGE LESS THAN 95%'
@ 3,2 SAY 'RE-OPTIMIZE TO IMPROVE OPTIMIZATION PERCENTAGE (Y/N) => ' GET MSEL
READ
        CLEAR
        STORE 'N' TO SELECT
@ 2,2 SAY 'PRINT OUT RESULTS FROM INITIAL OPTIMIZATION (Y/N) => ' GET SELECT
READ

IF SELECT = 'Y'
        DO PRINTOPT
ENDIF

IF MSELECT = 'Y'
        DO REOPT
ENDIF

*RELEASE WINDOW PARAMETER

ENDIF

RETURN


*-----------------------------------------------------------------

PROCEDURE PRINTOPT

STORE 0 TO TOTCOMP,BASE1COUNT, BASE2COUNT, EIM1COUNT, EIM2COUNT
*CLOSE ALL
SELECT COMP
CLEAR
@ 2,2 SAY 'HIT ANY KEY TO START PRINTING =>'
WAIT ""
SET DEVICE TO PRINT
STORE 'T' TO PRINT
@ 1,0 SAY 'DESCRIPTION              QUNTY  HEAD  FEEDER POS. BASE/EIM  NOZZLE
@ 2,0 SAY '--------------------------------------------------------------------
GO TOP
DO WHILE .NOT. EOF()
@ X,0 SAY COMP_NAME
@ X,18 SAY NUM_PER_BD
@ X,28 SAY PNP_HEAD.
@ X,36 SAY FEEDER_POS
@ X,49 SAY MOUNT_POST
@ X,55 SAY NOZZ_SIZE
@ X,63 SAY WK_TIME

STORE TOTCOMP + NUM_PER_BD TO TOTCOMP
IF MOUNT_POST = 'B' .AND. PNP_HEAD = 1
```

```
        ENDIF
ENDIF

IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 1)
        STORE EIM1COUNT + 1 TO EIM1COUNT
ELSE
        IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 2)
                STORE EIM2COUNT + 1 TO EIM2COUNT
        ENDIF
ENDIF

SKIP
STORE X + 1 TO X

ENDDO

IF HEAD1WK > HEAD2WK
        STORE (HEAD2WK/HEAD1WK)*100 TO PERCENT
        STORE 1 TO HD
ELSE
        STORE (HEAD1WK/HEAD2WK)*100 TO PERCENT
        STORE 2 TO HD
ENDIF
SKIP
@ X-1,0 SAY '                                     HEAD1                 HEAD2
@ X,0 SAY '                                     -----                   -----
@ X+1,1 SAY 'TOTAL NO. OF COMPONENTS ON BASE = '
@ X+1,35 SAY BASE1COUNT PICTURE "999"
@ X+1,56 SAY BASE2COUNT PICTURE "999"
@ X+2,1 SAY 'TOTAL NO. OF COMPONENTS ON EIM = '
@ X+2,35 SAY EIM1COUNT PICTURE "999"
@ X+2,56 SAY EIM2COUNT PICTURE "999"
@ X+4,1 SAY 'TOTAL NUMBER OF COMPONENTS = '
@ X+4,29 SAY TOTCOMP PICTURE "9999"
@ X+6,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD1 = '
@ X+6,34 SAY HEAD1WK PICTURE "999.99"
@ X+6,44 SAY 'Minutes'
@ X+8,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD2 = '
@ X+8,34 SAY HEAD2WK PICTURE "999.99"
@ X+8,44 SAY 'Minutes'
@ X+10,1 SAY 'OPTIMIZATION PERCENTAGE = '
@ X+10,27 SAY PERCENT PICTURE "999.9"
@ X+10,36 SAY '%'
*@ 16,1

SET DEVICE TO SCREEN
RETURN
```

```
*--------------------- MAIN PROCEDURE ------------------------------

DO WHILE .T.

CLEAR ALL
DEFINE WINDOW PARAMETER FROM 3,5 TO 19,73 DOUBLE
ACTIVATE WINDOW PARAMETER
STORE 1 TO MSELECT
CLEAR
TEXT
                    SIMULATION AND OPTIMIZATION
                       DATA BASE MAIN MENU

        [1] - Edit Database

        [2] - Delete Existing Databases

        [3] - Print Optimization Report


        [88] - Exit to DBaseIV Command Level
        [99] - Exit to Dos
ENDTEXT
@ 14,3 SAY 'Enter Selection =>' GET MSELECT PICTURE "99"
READ

DO CASE
        CASE MSELECT = 1
        RELEASE WINDOW PARAMETER
        DO MAINPROC

        CASE MSELECT = 2
        RELEASE WINDOW PARAMETER
        DO DELETE
        DO MAIN

        CASE MSELECT = 3
        RELEASE WINDOW PARAMETER

        CASE MSELECT = 88
        RELEASE WINDOW PARAMETER
        SET STATUS ON
        CANCEL

        CASE MSELECT = 99
        RELEASE WINDOW PARAMETER
        * CANCEL
        QUIT

        OTHERWISE
        @ 14,3
        @ 14,3 SAY 'ERROR G010 - ILLEGAL OPTION'
        *DO DELAY WITH 2
ENDCASE

ENDDO

*--------------------- M A I N  P R O C E D U R E -------------------
PROCEDURE MAINPROC

DO MENUDEF
SET CLOCK TO 1,69
ACTIVATE MENU SMTMENU
RELEASE MENU SMTMENU
RELEASE POPUPS IMENU
```

```
RETURN

*-------------------------------------------------------------------

PROCEDURE MENUDEF

DEFINE MENU SMTMENU MESSAGE "MAIN SIMULATION PARAMETER MENU"
DEFINE PAD INIT OF SMTMENU PROMPT "Initialization"
DEFINE PAD WSTATION OF SMTMENU PROMPT "Workstations"
DEFINE PAD TASK OF SMTMENU PROMPT "Job Setup"
DEFINE PAD REPORT OF SMTMENU PROMPT "Run Times & Reports"
DEFINE PAD QUIT OF SMTMENU PROMPT "Quit"  MESSAGE "====> EXIT <===="

ON SELECTION PAD INIT OF SMTMENU ACTIVATE POPUP IMENU
ON SELECTION PAD WSTATION OF SMTMENU ACTIVATE POPUP WMENU
ON SELECTION PAD TASK OF SMTMENU ACTIVATE POPUP TMENU
ON SELECTION PAD REPORT OF SMTMENU ACTIVATE POPUP RMENU
ON SELECTION PAD QUIT OF SMTMENU DEACTIVATE MENU

DEFINE POPU IMENU FROM 10,30 MESSAGE "Setup INITIALIZATION Parameters"
DEFINE BAR 1 OF IMENU PROMPT "Setup Parameters" skip
DEFINE BAR 3 OF IMENU PROMPT "New Production Plan"
DEFINE BAR 4 OF IMENU PROMPT "Initialization"
DEFINE BAR 5 OF IMENU PROMPT "Exit to Main Menu"
ON SELECTION POPUP IMENU DO SELini WITH BAR()

DEFINE POPU WMENU FROM 6,30 MESSAGE "Setup WORKSTATION Parameters"
DEFINE BAR 1 OF WMENU PROMPT "Setup WS Parameters" skip
DEFINE BAR 3 OF WMENU PROMPT "Conveyor Length"
DEFINE BAR 4 OF WMENU PROMPT "Conveyor Direction"
DEFINE BAR 5 OF WMENU PROMPT "Conveyor Xcoord"
DEFINE BAR 6 OF WMENU PROMPT "Conveyor Ycoord"
DEFINE BAR 7 OF WMENU PROMPT "Ws Xcoord"
DEFINE BAR 8  OF WMENU PROMPT "Ws Ycoord"
DEFINE BAR 9 OF WMENU PROMPT "Status Xcoord"
DEFINE BAR 10 OF WMENU PROMPT "Status Ycoord"
DEFINE BAR 11 OF WMENU PROMPT "Queue Xcoord"
DEFINE BAR 12 OF WMENU PROMPT "Queue Ycoord"
DEFINE BAR 13 OF WMENU PROMPT "Display Xcoord"
DEFINE BAR 14 OF WMENU PROMPT "Display Ycoord"
DEFINE BAR 15 OF WMENU PROMPT "Number of Machines"
DEFINE BAR 16 OF WMENU PROMPT "Exit to Main Menu"

ON SELECTION POPUP WMENU DO SELws WITH BAR()

DEFINE POPU TMENU FROM 10,30 MESSAGE "Setup TASK Parameters"
DEFINE BAR 1 OF TMENU PROMPT "Setup TASK Parameters" skip
DEFINE BAR 3 OF TMENU PROMPT "Task Workstations"
DEFINE BAR 4 OF TMENU PROMPT "Task Work-Time"
DEFINE BAR 5 OF TMENU PROMPT "PnP Component Setup"
DEFINE BAR 6 OF TMENU PROMPT "PnP Machine Optimization"
DEFINE BAR 7 OF TMENU PROMPT "Exit to Main Menu"

ON SELECTION POPUP TMENU DO SELMENUtask

DEFINE POPU RMENU FROM 10,30 MESSAGE "Setup REPORT Parameters"
DEFINE BAR 1 OF RMENU PROMPT "Setup REPORT Parameters" skip
DEFINE BAR 3 OF RMENU PROMPT "Run-Time Setup"
DEFINE BAR 4 OF RMENU PROMPT "Report-Time Setup"
DEFINE BAR 5 OF RMENU PROMPT "Print/Save Data"
DEFINE BAR 6 OF RMENU PROMPT "Exit to Main Menu"

ON SELECTION POPUP RMENU DO SELMENUrep

*---------------------DOWN: MENU FOR INITIALIZATION :DOWN---------------
```

```
DO CASE
        CASE TEMP = 3
        CLEAR
        DEFINE WINDOW PARAMETER FROM 8,5 TO 20,73 DOUBLE
        ACTIVATE WINDOW PARAMETER
        USE PRODUCT
        IF EOF()
                APPEND BLANK
        ENDIF

        GO TOP
        TEXT
                        NEW PRODUCTION INFORMATION

        JOB NAME/CODE =

        WAVE SOLDERING REQUIRED (Y/N) =

        SOLDER REFLOW REQUIRED (Y/N) =

        ENDTEXT

        @ 3,23 SAY '' GET JOB_NAME
        @ 5,38 SAY '' GET SOLDER
        @ 7,38 SAY '' GET REFLOW
        READ

        RELEASE WINDOW PARAMETER

        CASE TEMP = 4

        DEFINE WINDOW PARAMETER FROM 6,5 TO 23,73 DOUBLE
        ACTIVATE WINDOW PARAMETER
        USE PARAMETE
        GO TOP
        TEXT
                        PARAMETER INFORMATION

  NUMBER OF WORKSTATIONS =           INTERARRIVAL TIME =

  NUMBER OF JOBS =              NUMBER PER BATCH =

  NUMBER OF TASKS =         TOTAL NUMBER OF SIMULATIONS =

  NUMBER OF REPITITIONS PER SIMULATION =

  NUMBER OF DIFFERENT COMPONENT TYPES FOR THE PnP MACHINE =
        ENDTEXT
        @ 3,26 SAY '' GET NUM_MACHINE
        @ 3,57 SAY '' GET INTER_TIME
        @ 5,18 SAY '' GET NUM_JOBS
        @ 5,52 SAY '' GET NUM_PER_BA
        @ 7,19 SAY '' GET NUM_TASKS
        @ 7,57 SAY '' GET NO_OF_SIMS
        @ 9,42 SAY '' GET REPS
        @ 11,60 SAY '' GET C[illegible]_TYPES
        READ

        RELEASE WINDOW PARAMETER

        CASE TEMP = 5
        DEACTIVATE POPUP

        ENDCASE
close all
RETURN
```

```
*------------------DOWN: MENU PROCEDURES FOR WS SET UP: DOWN-----------------

PROCEDURE SELws
PARAMETERS TEMP
        select 4
        clear

        IF TEMP <> 16
        DEFINE WINDOW PARA FROM 1,5 TO 12,73 DOUBLE
        ACTIVATE WINDOW PARA
        @ 1,1 SAY 'MACHINE 1 = SOLDER PASTE MACHINE'
        @ 2,1 SAY 'MACHINE 2 = GRAVITY PLATFORM 1'
        @ 3,1 SAY 'MACHINE 3 = INPUT ELEVATOR'
        @ 4,1 SAY 'MACHINE 4 = CONVEYOR BELT'
        @ 5,1 SAY 'MACHINE 5 = HEAD1 OF PnP'
        @ 6,1 SAY 'MACHINE 6 = HEAD2 OF PnP'
        @ 1,35 SAY 'MACHINE 7 = CONVEYOR WITH INSPCT'
        @ 2,35 SAY 'MACHINE 8 = CUREFLOW OVEN'
        @ 3,35 SAY 'MACHINE 9 = CONVEYOR WITH INSPCT'
        @ 4,35 SAY 'MACHINE 10 = OUTPUT ELEVATOR'
        @ 5,35 SAY 'MACHINE 11 = GRAVITY PLATFORM'
        ENDIF

DO CASE
        CASE TEMP = 3

        DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
        ACTIVATE WINDOW PARAMETER
        USE PARAMETE
        GO TOP
        STORE NUM_MACHINE TO TEMP
        STORE 1 TO COUNT
        USE WORKSTAT
        GO TOP
        DO WHILE COUNT <> (TEMP + 1)
        IF EOF()
        APPEND BLANK
        ENDIF
        @ 2,0
        @ 2,6 SAY 'CONVEYOR LENGTH FOLLOWING MACHINE ['+STR(COUNT,2,0)+' ] =
        @ 2,48 SAY CONVEY_LNG
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,52 GET CONVEY_LNG
                READ
        ENDIF
        STORE "N" TO ANS
        @ 4,4
        STORE COUNT + 1 TO COUNT
        IF EOF()
        APPEND BLANK
        ELSE
        SKIP
        ENDIF
        ENDDO
        RELEASE WINDOW PARAMETER

        CASE TEMP = 4
        DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
        ACTIVATE WINDOW PARAMETER
        USE PARAMETE
        GO TOP
        STORE NUM_MACHINE TO TEMP
        STORE 1 TO COUNT
        USE WORKSTAT
        GO TOP
```

```
ENDIF
@ 2,0
@ 2,6 SAY 'CONVEYOR DIRECTION FOLLOWING MACHINE ['+STR(COUNT,2,0)+' ]
@ 2,51 SAY CONVEY_DIR
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,55 GET CONVEY_DIR
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 5

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'CONVEYOR X-LOCATION FOR MACHINE ['+STR(COUNT,2,0)+' ] = '
@ 2,46 SAY CONVEYXLOC
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,50 GET CONVEYXLOC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 6

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'CONVEYOR Y-LOCATION FOR MACHINE ['+STR(COUNT,2,0)+' ] = '
```

```
IF ANS = "Y"
        @ 2,50 GET CONVEYYLOC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 7

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'MACHINE ['+STR(COUNT,2,0)+' ] X-COORD = '
@ 2,35 SAY WSXLOCATN
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,39 GET WSXLOCATN
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 8

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'MACHINE ['+STR(COUNT,2,0)+' ] Y-COORD ='
@ 2,35 SAY WSYLOCATN
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,39 GET WSYLOCATN
        READ
```

```
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETFR

CASE TEMP = 9

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'STATUS DISPLAY X-COORD FOR MACHINE ['+STR(COUNT,2,0)+' ] =
@ 2,51 SAY STATUSXLOC
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,55 GET STATUSXLOC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 10

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'STATUS DISPLAY Y-COORD FOR MACHINE ['+STR(COUNT,2,0)+' ] =
@ 2,51 SAY STATUSYLOC
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,55 GET STATUSYLOC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
```

```
CASE TEMP = 11

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'QUEUE X-COORD FOR MACHINE ['+STR(COUNT,2,0)+' ] = '
@ 2,40 SAY QUEUEXLOC
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,44 GET QUEUEXLOC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 12

DEFINE WINDOW PARAMETER FROM 15,5 TO [illegible]BLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'QUEUE Y-COORD FOR MACHINE ['+STR(COUNT,2,0)+' ] = '
@ 2,40 SAY QUEUEYLOC
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,44 GET QUEUEYLOC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 13
```

```
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'QUEUE VALUE DISPLAY X-COORD FOR MACHINE ['+STR(COUNT,2,0)+
@ 2,54 SAY DISPLAYXLC
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,56 GET DISPLAYXLC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 14

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
GO TOP
DO WHILE COUNT <> (TEMP + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'QUEUE VALUE DISPLAY Y-COORD FOR MACHINE ['+STR(COUNT,2,0)+
@ 2,52 SAY DISPLAYYLC
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,56 GET DISPLAYYLC
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARAMETER

CASE TEMP = 15

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE PARAMETE
GO TOP
STORE NUM_MACHINE TO TEMP
STORE 1 TO COUNT
USE WORKSTAT
```

```
        APPEND BLANK
        ENDIF
        @ 2,0
        @ 2,6 SAY 'NUMBER OF MACHINES AT MACHINE ['+STR(COUNT,2,0)+'] ='
        @ 2,41 SAY NUM_MACHIN
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,45 GET NUM_MACHIN
                READ
        ENDIF
        STORE COUNT + 1 TO COUNT
        IF EOF()
        APPEND BLANK
        ELSE
        SKIP
        ENDIF
        ENDDO
        RELEASE WINDOW PARAMETER

        CASE TEMP = 16
        DEACTIVATE POPUP

        OTHERWISE
        DEACTIVATE POPUP

ENDCASE
RELEASE WINDOW PARA

CLOSE ALL
RETURN

*--------------------UP: MENU PROCEDURES FOR WS SET UP :UP-----------------


*-----------------DOWN: MENU PROCEDURES FOR TASK SET UP: DOWN----------------


PROCEDURE SELMENUtask

DO CASE
        CASE BAR() = 3

        select 4
        clear
        DEFINE WINDOW PARA FROM 1,5 TO 12,73 DOUBLE
        ACTIVATE WINDOW PARA
        @ 1,1 SAY 'MACHINE 1 = SOLDER PASTE MACHINE'
        @ 2,1 SAY 'MACHINE 2 = GRAVITY PLATFORM 1'
        @ 3,1 SAY 'MACHINE 3 = INPUT ELEVATOR'
        @ 4,1 SAY 'MACHINE 4 = CONVEYOR BELT'
        @ 5,1 SAY 'MACHINE 5 = HEAD1 OF PnP'
        @ 6,1 SAY 'MACHINE 6 = HEAD2 OF PnP'
        @ 1,35 SAY 'MACHINE 7 = CONVEYOR WITH INSPCT'
        @ 2,35 SAY 'MACHINE 8 = CUREFLOW OVEN'
        @ 3,35 SAY 'MACHINE 9 = CONVEYOR WITH INSPCT'
        @ 4,35 SAY 'MACHINE 10 = OUTPUT ELEVATOR'
        @ 5,35 SAY 'MACHINE 11 = GRAVITY PLATFORM'

        DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
        ACTIVATE WINDOW PARAMETER

        USE PARAMETE
        GO TOP
        STORE NUM_TASKS TO TASKS
        STORE 1 TO COUNT
```

```
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'ENTER THE MACHINE NO. FOR TASK NUMBER ['+STR(COUNT,2,0)+'
@ 2,52 SAY '['+STR(TASK_WS,2,0)+']'
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,56 GET TASK_WS
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
ENDDO
RELEASE WINDOW PARA
RELEASE WINDOW PARAMETER

CASE BAR() = 4

select 4
clear
DEFINE WINDOW PARA FROM 1,5 TO 12,73 DOUBLE
ACTIVATE WINDOW PARA
@ 1,1 SAY 'MACHINE 1 = SOLDER PASTE MACHINE'
@ 2,1 SAY 'MACHINE 2 = GRAVITY PLATFORM 1'
@ 3,1 SAY 'MACHINE 3 = INPUT ELEVATOR'
@ 4,1 SAY 'MACHINE 4 = CONVEYOR BELT'
@ 5,1 SAY 'MACHINE 5 = HEAD1 OF PnP'
@ 6,1 SAY 'MACHINE 6 = HEAD2 OF PnP'
@ 1,35 SAY 'MACHINE 7 = CONVEYOR WITH INSPCT'
@ 2,35 SAY 'MACHINE 8 = CUREFLOW OVEN'
@ 3,35 SAY 'MACHINE 9 = CONVEYOR WITH INSPCT'
@ 4,35 SAY 'MACHINE 10 = OUTPUT ELEVATOR'
@ 5,35 SAY 'MACHINE 11 = GRAVITY PLATFORM'

DEFINE WINDOW PARAMETER FROM 15,5 TO 23,73 DOUBLE
ACTIVATE WINDOW PARAMETER
USE paramete
GO TOP
STORE NUM_TASKS TO TASKS
STORE 1 TO COUNT
USE TASK
GO TOP
DO WHILE COUNT <> (TASKS + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'ENTER THE TASK TIME FOR MACHINE NUMBER ['+STR(COUNT,2,0)+'
@ 2,52 SAY '['+STR(TASK_TIME,2,0)+']'
@ 4,4
ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
IF ANS = "Y"
        @ 2,56 GET TASK_TIME
        READ
ENDIF
STORE COUNT + 1 TO COUNT
IF EOF()
APPEND BLANK
ELSE
SKIP
ENDIF
```

```
RELEASE WINDOW PARAMETER

CASE BAR() = 5

SELECT 4
CLEAR
DEFINE WINDOW PARAMETER FROM 3,3 TO 17,77 DOUBLE
ACTIVATE WINDOW PARAMETER
STORE ' ' TO ANS

USE PARAMETE
GO TOP
STORE COMP_TYPES TO COMPONENTS
STORE 1 TO COUNT
USE COMP
GO TOP

DO WHILE COUNT <> (COMPONENTS + 1)
IF EOF()
APPEND BLANK
ENDIF
@ 2,3 SAY 'FOR COMPONENT TYPE [    ] ENTER THE FOLLOWING INFO:'
@ 2, 23 SAY ''+STR(COUNT,2,0)+'' COLOR R+*
@ 4,3 SAY 'COMPONENT NAME OR CODE NUMBER: '
@ 4,35 SAY '['+(COMP_NAME)+ ']' COLOR R+
@ 6,0 CLEAR
ACCEPT "<ENTER> TO ACCEPT, `A' TO ALTER VALUES :" TO ANS
IF ANS = "A"
        STORE COMP_NAME TO NAME
        STORE NOZZ_SIZE TO NOZZLE
        DO COMPSELE WITH NAME, NOZZLE
        SELECT 4
        REPLACE NOZZ_SIZE WITH NOZZLE
        REPLACE COMP_NAME WITH NAME
ENDIF
@ 4,0
@6,0 CLEAR

IF ANS = "A"
@ 2, 23 SAY ''+STR(COUNT,2,0)+'' COLOR R+*
@ 4,3 SAY 'COMPONENT NAME OR CODE NUMBER: '
@ 4,35 SAY '['+(COMP_NAME)+ ']' COLOR R+*
@ 6,0 CLEAR
ACCEPT "<ENTER> TO ACCEPT, `A' TO ALTER VALUES :" TO ANS
IF ANS = "A"
        @ 4,35 GET COMP_NAME
        READ
ENDIF
@ 4,0
@ 6,0
ENDIF

@ 4,3 SAY 'IS THE COMPONENT ON THE SOLDER OR COMPONENT SIDE: '
@ 4,53 SAY '[ ' +(SIDE)+ ' ]'COLOR R+
@ 6,0 CLEAR
ACCEPT "<ENTER> TO ACCEPT, `A' TO ALTER VALUES :" TO ANS
IF ANS = "A"
        STORE SIDE TO PLACE
        DO PCBSELEC WITH PLACE
        SELECT 4
        REPLACE SIDE WITH PLACE
ENDIF
@ 4,4


@ 4,3 SAY 'TOTAL NUMBER OF COMPONENTS PER BOARD: '
@ 4,42 SAY '[ ' +STR(NUM_PER_BD,3,2)+ ' ]'COLOR R+
```

```
        @ 4,42 GET NUM_PER_BD
        READ
ENDIF
@ 4,4


@ 4,3 SAY 'RECOMMENDED NOZZLE REQUIRED : '
@ 4,31 SAY '[ '+STR(NOZZ_SIZE,6,4)+ ']' COLOR R+
@ 6,0 CLEAR
ACCEPT "<ENTER> TO ACCEPT, `A' TO ALTER VALUES :" TO ANS
IF ANS = "A"
        STORE NOZZ_SIZE TO NOZZLE
        DO NOZSELEC WITH NOZZLE
        SELECT 4
        REPLACE NOZZ_SIZE WITH NOZZLE
ENDIF

@ 4,0 CLEAR
@ 6,0 CLEAR

IF ANS = "A"
@ 2, 23 SAY ''+STR(COUNT,2,0)+'' COLOR R+*
@ 4,3 SAY 'SELECTED NOZZLE SIZE: '
@ 4,31 SAY '['+STR(NOZZ_SIZE,6,4)+ ']' COLOR R+*
@ 6,0 CLEAR
ACCEPT "<ENTER> TO ACCEPT, `A' TO ALTER VALUES :" TO ANS
IF ANS = "A"
        @ 4,35 GET NOZZ_SIZE
        READ
ENDIF

ENDIF


@ 4,0 CLEAR
@6,0 CLEAR

@ 4,3 SAY 'TYPE OF FEEDER REQUIRED:'
@ 4,28 SAY '[ ' +(FEED_TYPE)+ ']'COLOR R+
@ 6,0 CLEAR
ACCEPT "<ENTER> TO ACCEPT, `A' TO ALTER VALUES :" TO ANS
IF ANS = "A"
        STORE FEED_TYPE TO FEEDER
        DO FEEDSELE WITH FEEDER
        SELECT 4
        REPLACE FEED_TYPE WITH FEEDER
ENDIF
@ 4,4

IF ANS = "A"
@ 2, 23 SAY ''+STR(COUNT,2,0)+'' COLOR R+*
@ 4,3 SAY 'SELECTED FEEDER TYPE: '
@ 4,28 SAY '['+(FEED_TYPE)+ ']' COLOR R+*
@ 6,0 CLEAR
ACCEPT "<ENTER> TO ACCEPT, `A' TO ALTER VALUES :" TO ANS
IF ANS = "A"
        @ 4,35 GET FEED_TYPE
        READ
ENDIF

ENDIF

@ 4,0 CLEAR
@6,0 CLEAR

@ 4,3 SAY 'IS THE COMPONENT VERIFIABLE?'
@ 4,33 SAY '[' +(VERIFICAT)+ ']'COLOR R+
```

```
                READ
        ENDIF
        @ 4,4

        STORE COUNT + 1 TO COUNT
        IF EOF()
        APPEND BLANK
        ELSE
        SKIP
        ENDIF
        ENDDO

        RELEASE WINDOW PARAMETER

        CASE BAR() = 6
        SELECT 4
        USE COMP
        INDEX ON SIDE TO NUM
        SEEK 'BOTTOM'

        IF FOUND()
                DO GLUEOPT
        ELSE
                DO OPTIMIZE
        ENDIF

        CASE BAR() = 7
        DEACTIVATE POPUP


ENDCASE
CLOSE ALL
RETURN

*-----------------UP: MENU PROCEDURES FOR TASK SET UP: UP----------------


*-----------------------------------------------------------------------

*-----------------DOWN: MENU PROCEDURES FOR REPORT SET UP: DOWN-------------


PROCEDURE SELMENUrep
DO CASE

        CASE BAR() = 3
        DEFINE WINDOW PARAMETER FROM 10,5 TO 23,73 DOUBLE
        ACTIVATE WINDOW PARAMETER

        USE REPORT
        GO TOP
        STORE 1 TO COUNT

        DO WHILE COUNT <> 5
        IF EOF()
        APPEND BLANK
        ENDIF
        @ 2,0
        @ 2,6 SAY 'ENTER RUN TIME NUMBER ['+STR(COUNT,2,0)+' ]: '
        DO CASE
                CASE COUNT = 1
                @ 2,36 SAY '['+STR(RUN1,7,2)+']'
                @ 4,4
                ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
                IF ANS = "Y"
```

```
        CASE COUNT = 2

        @ 2,36 SAY '['+STR(RUN2,7,2)+']'
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,36 GET RUN2
                READ
        ENDIF
        STORE COUNT + 1 TO COUNT

        CASE COUNT = 3

        @ 2,36 SAY '['+STR(RUN3,7,2)+']'
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,36 GET RUN3
                READ
        ENDIF
        STORE COUNT + 1 TO COUNT

        CASE COUNT = 4

        @ 2,36 SAY '['+STR(RUN4,7,2)+']'
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,36 GET RUN4
                READ
        ENDIF
        STORE COUNT + 1 TO COUNT
ENDCASE
ENDDO
RELEASE WINDOW PARAMETER

CASE BAR() = 4
DEFINE WINDOW PARAMETER FROM 10,5 TO 21,73 DOUBLE
ACTIVATE WINDOW PARAMETER

USE REPORT
GO TOP
STORE 1 TO COUNT

DO WHILE COUNT <> 5
IF EOF()
APPEND BLANK
ENDIF
@ 2,0
@ 2,6 SAY 'ENTER REPORT TIME NUMBER ['+STR(COUNT,2,0)+' ]: '
DO CASE
        CASE COUNT = 1
        @ 2,39 SAY '['+STR(REPORT1,7,2)+']'
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,39 GET REPORT1
                READ
        ENDIF
        STORE COUNT + 1 TO COUNT

        CASE COUNT = 2

        @ 2,39 SAY '['+STR(REPORT2,7,2)+']'
        @ 4,4
```

```
        ENDIF
        STORE COUNT + 1 TO COUNT


        CASE COUNT = 3

        @ 2,39 SAY '['+STR(REPORT3,7,2)+']'
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,39 GET REPORT3
                READ
        ENDIF
        STORE COUNT + 1 TO COUNT


        CASE COUNT = 4

        @ 2,39 SAY '['+STR(REPORT4,7,2)+']'
        @ 4,4
        ACCEPT "ALTER VALUES (Y/N)? :" TO ANS
        IF ANS = "Y"
                @ 2,39 GET REPORT4
                READ
        ENDIF
        STORE COUNT + 1 TO COUNT


ENDCASE
ENDDO
RELEASE WINDOW PARAMETER


CASE BAR() = 5
SELECT 1
USE PARAMETE
SELECT 2
USE WORKSTAT

DEFINE WINDOW PARAMETER FROM 10,5 TO 21,73 DOUBLE
ACTIVATE WINDOW PARAMETER
STORE "P" TO SEL
@ 4,6 SAY 'SEND OUTPUT TO PRINTER, SCREEN OR FILE (P/S/F)?:' GET SEL
READ
IF SEL = "P"
        @ 4,6
        @ 6,6 SAY 'Printing, please wait..... '
        SET DEVICE TO PRINT
        SELECT PARAMETE
        GO TOP
        STORE NUM_MACHINE TO MACHINES
        SELECT WORKSTAT
        GO TOP
        STORE 0 TO COUNT
        DO WHILE COUNT <> (12*MACHINES)
                @ (COUNT),0 SAY CONVEY_LNG
                @ (COUNT+1),0 SAY CONVEY_DIR
                @ (COUNT+2),0 SAY CONVEYXLOC
                @ (COUNT+3),0 SAY CONVEYYLOC
                @ (COUNT+4),0 SAY WSXLOCATN
                @ (COUNT+5),0 SAY WSYLOCATN
                @ (COUNT+6),0 SAY STATUSXLOC
                @ (COUNT+7),0 SAY STATUSYLOC
                @ (COUNT+8),0 SAY QUEUEXLOC
                @ (COUNT+9),0 SAY QUEUEYLOC
                @ (COUNT+10),0 SAY DISPLAYXLC
                @ (COUNT+11),0 SAY DISPLAYYLC
```

```
        SET DEVICE TO SCREEN

ELSE
IF SEL = "F"
        @ 4,6
        @ 4,6 SAY 'Saving to file: <SMTIN>.<DAT> ' COLOR R+*
        SET CONSOLE OFF
SELECT 1
USE PARAMETE
SELECT 2
USE REPORT
SELECT 3
USE WORKSTAT
SELECT 4
USE TASK
SELECT 5
USE COMP
SELECT 6
USE HEAD
        SELECT PARAMETE
        GO TOP
        SET ALTERNATE TO C:\SIM\PNP\SMTTEST.TXT
        SET ALTERNATE ON
        ? NUM_MACHINE
        STORE NUM_MACHINE TO MACHINES
        SET ALTERNATE OFF
        SELECT WORKSTAT
        GO TOP
        STORE 0 TO COUNT
        DO WHILE COUNT <> MACHINES
                SET ALTERNATE ON
                ? NUM_MACHINE
                ? CONVEY_LNG
                ? CONVEY_DIR
                ? CONVEYXLOC
                ? CONVEYYLOC
                ? WSXLOCATN
                ? WSYLOCATN
                ? STATUSXLOC
                ? STATUSYLOC
                ? QUEUEXLOC
                ? QUEUEYLOC
                ? DISPLAYXLC
                ? DISPLAYYLC

                SET ALTERNATE OFF
                SKIP
                STORE COUNT + 1 TO COUNT
        ENDDO
        SELECT PARAMETE
                SET ALTERNATE ON
                ? INTER_TIME
                ? NUM_JOBS
                ? NUM_PER_BA
                ? NUM_TASKS
                SET ALTERNATE OFF
        STORE 0 TO COUNT
        STORE NUM_TASKS TO TASKS
        SELECT TASK
        INDEX ON TASK_WS TO NUM1
        go top
        seek 1
        DO WHILE COUNT <> TASKS
                SET ALTERNATE ON
                ? TASK_WS
                ? TASK_TIME
                SET ALTERNATE OFF
```

```
                SELECT PARAMETE
                        SET ALTERNATE ON
                        ? REPS
                        ? NO_OF_SIMS
                        SET ALTERNATE OFF
                SELECT REPORT
                GO TOP
                        SET ALTERNATE ON
                        ? RUN1
                        ? REPORT1
                        ? RUN2
                        ? REPORT2
                        ? RUN3
                        ? REPORT3
                        ? RUN4
                        ? REPORT4

                        SET ALTERNATE OFF

                SELECT HEAD
                        SET ALTERNATE ON
                ? HEAD1_TIME
                ? HEAD2_TIME
                ?
                        SET ALTERNATE OFF

                CLOSE DATABASES
                SET CONSOLE ON
        ELSE
        IF SEL = "S"
                @ 4,6
                SELECT WORKSTAT
                GO TOP
                BROWSE
        ENDIF
        ENDIF
        ENDIF
        RELEASE WINDOW PARAMETER

        CASE BAR() = 6
        DEACTIVATE POPUP

NDCASE
LOSE ALL
ETURN

*---------------------------------------------------------------------
```

```
* NOZSELECT PROCEDURE

PARAMETER NOZLE

SELECT 6

DEFINE POPUP NOZZLE FROM 1,61 MESSAGE "Place Highlighted Bar Over Choice and
DEFINE BAR 1 OF NOZZLE PROMPT "0.042"
DEFINE BAR 2 OF NOZZLE PROMPT "0.058"
DEFINE BAR 3 OF NOZZLE PROMPT "0.148"
DEFINE BAR 4 OF NOZZLE PROMPT "0.109"
DEFINE BAR 5 OF NOZZLE PROMPT "0.085"
DEFINE BAR 6 OF NOZZLE PROMPT "0.225"
DEFINE BAR 7 OF NOZZLE PROMPT "0.208"
DEFINE BAR 8 OF NOZZLE PROMPT "EXIT"

ON SELECTION POPUP NOZZLE DO TEST

ACTIVATE POPUP NOZZLE

RELEASE POPUPS NOZZLE
RETURN

*----------------------------------------

PROCEDURE TEST

DO CASE
        CASE BAR() = 1
        STORE 0.042 TO NOZLE

        CASE BAR() = 2
        STORE 0.058 TO NOZLE

        CASE BAR() = 3
        STORE 0.148 TO NOZLE

        CASE BAR() = 4
        STORE 0.109 TO NOZLE

        CASE BAR() = 5
        STORE 0.085 TO NOZLE

        CASE BAR() = 6
        STORE 0.225 TO NOZLE

        CASE BAR() = 7
        STORE 0.208 TO NOZLE

        CASE BAR() = 8
        DEACTIVATE POPUP
ENDCASE

@ 4,31 SAY '['+STR(NOZLE,6,4)+']' COLOR R+

IF BAR() <> 8
        DEACTIVATE POPUP
ENDIF

RETURN
*-----------------------------------------
```

```
* 21/12/89
*
STORE 'F' TO LEFT
STORE 'F' TO COMPCHECK, PNPFULL
STORE 1 TO B1COUNT, B2COUNT, E1COUNT, E2COUNT
STORE 'F' TO BASE1, BASE2, EIM1, EIM2
STORE 'F' TO FIRST_FIND, SECOND_FIND, THIRD_FIND, FOURTH_FIND
STORE 'F' TO EIM1FULL, EIM2FULL, BASE1FULL
STORE 0 TO HEAD1WK, HEAD2WK, HD, PERCENT,COMPNUM1, COMPNUM2
STORE 0.03455 TO EIMTIME
STORE 0.03083 TO BASETIME

*SELECT 7
*USE COMP
SELECT 8
USE HEAD

CLEAR
DEFINE WINDOW PARAMETER FROM 6,1 TO 16,75 DOUBLE
ACTIVATE WINDOW PARAMETER

CLEAR
STORE 1 TO MSELECT
@ 2,2 SAY 'OPTIMIZATION METHOD 1 - BASED ON NUMBER OF COMPONENTS PER BOARD'
@ 4,2 SAY 'OPTIMIZATION METHOD 2 - BASED ON PRIORITIZED NOZZLE PLACEMENT'
@ 6,2 SAY 'SELECT OPTION => 'GET MSELECT PICTURE "9"
READ
IF MSELECT = 1
        RELEASE WINDOW PARAMETER
        DO METHOD1
ELSE
        RELEASE WINDOW PARAMETER
        DO METHOD2
ENDIF
CLOSE ALL
RETURN

*-------------------- PROCEDURE METHOD1 -------------------------

PROCEDURE METHOD1

SELECT 6
CLEAR
 DEFINE WINDOW PARAMETER FROM 7,3 TO 14,73 DOUBLE COLOR R+*
 ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '  - DO NOT INTERRUPT -' COLOR W+
SELECT COMP
GO TOP
DO WHILE .NOT. EOF()
REPLACE MARKER WITH 'F'
SKIP
ENDDO

*----- STARTING OPTIMIZATION ROUTINE ---------------

INDEX ON NOZZ_SIZE TO NEW
STORE 0 TO COUNT
GO TOP

DO WHILE .NOT. EOF()

STORE NOZZ_SIZE TO X

        IF MARKER = 'F'
```

```
                                STORE 'F' TO EIM1
                                STORE 'T' TO EIM2
                                DO CHECKEIM1
        STORE COMPNUM1 + NUM_PER_BD TO COMPNUM1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO FIRST_FIND
                        ELSE
                        IF SECOND_FIND = 'F'
                                DO WHILE NOZZ_SIZE = X
                                STORE 'T' TO EIM1
                                STORE 'F' TO EIM2
                                DO CHECKEIM2
        STORE COMPNUM2 + NUM_PER_BD TO COMPNUM2
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO SECOND_FIND
                        ELSE
IF THIRD_FIND = 'F' .AND. COMPNUM1 < COMPNUM2
                                DO WHILE NOZZ_SIZE = X
                                STORE 'F' TO EIM1
                                STORE 'T' TO EIM2
                                DO CHECKEIM1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO EIM1FULL
                                STORE 'T' TO THIRD_FIND
                        ELSE
                        IF FOURTH_FIND = 'F'
                                DO WHILE NOZZ_SIZE = X
                                STORE 'T' TO EIM1
                                STORE 'F' TO EIM2
                                DO CHECKEIM2
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO EIM2FULL
                                STORE 'T' TO FOURTH_FIND
                        ELSE
                                DO CHECKBASE1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                        ENDIF
                        ENDIF
                        ENDIF
                        ENDIF
                ELSE
                        DO CHECKBASE1
                        IF .NOT. EOF()
                                SKIP
                        ENDIF

                ENDIF
        ELSE
                SKIP
        ENDIF
ENDDO

SELECT HEAD
```

```
CLOSE ALL
STORE 4 TO X
DO DELAY WITH X

DO OPT1REPORT
IF PERCENT < 95
STORE PERCENT TO NEWPERCENT

DEFINE WINDOW PARAMETER FROM 8,1 TO 14,75 DOUBLE
ACTIVATE WINDOW PARAMETER
CLEAR
STORE 'N' TO MSELECT
@ 1,2 SAY 'OPTIMIZATION PERCENTAGE LESS THAN 95%'
@ 3,2 SAY 'RE-OPTIMIZE TO IMPROVE OPTIMIZATION PERCENTAGE (Y/N) => ' GET MSEL
READ
CLEAR
STORE 'N' TO SELECT
@ 2,2 SAY 'PRINT OUT RESULTS FROM INITIAL OPTIMIZATION (Y/N) => ' GET SELECT
READ
IF SELECT = 'Y'
        DO PRINTOPT
ENDIF
RELEASE WINDOW PARAMETER
ENDIF

IF MSELECT = 'Y'
        DO REOPT
ENDIF

RETURN

*---------------------- END PROCEDURE METHOD1 ----------------------

*--------------------  PROCEDURE CHECKBASE1  -----------------------

PROCEDURE CHECKBASE1

IF BASE1 = 'F'
        IF B1COUNT <= 20
                        DO SETBASE1
        ELSE
                        DO CHKBAS2
        ENDIF

ELSE
        DO CHKBAS2

ENDIF

RETURN
*--------------------------- ---------------------------------------

*-------------------  PROCEDURE CHKBAS2  ---------------------------

PROCEDURE CHKBAS2

IF BASE2 = 'F'
        IF B2COUNT <= 20
                DO SETBASE2
        ELSE
                STORE 'T' TO COMPCHECK
                DO CHECKEIM1
        ENDIF
ELSE
        STORE 'T' TO COMPCHECK
        DO CHECKEIM1
```

```
*---------------------------------------------------------------------
*----------------- PROCEDURE CHECKEIM1  ------------------------------

PROCEDURE CHECKEIM1

IF EIM1 = 'F'
        IF E1COUNT <= 43
                DO SETEIM1
        ELSE
                DO CHECKEIM2
        ENDIF

ELSE
        DO CHECKEIM2
ENDIF

RETURN

*---------------------------------------------------------------------
*---------------- PROCEDURE CHECKEIM2 --------------------------------

PROCEDURE CHECKEIM2

IF EIM2  = 'F'
        IF E2COUNT <= 43
                DO SETEIM2
        ELSE
                IF COMPCHECK = 'T'
                        STORE 'T' TO PNPFULL
                ELSE
                        DO CHECKBASE1
                ENDIF
        ENDIF
ELSE
        DO CHECKBASE1
ENDIF

RETURN

*----------------------------------------------------------------------
*----------------- PROCEDURE SETEIM1  --------------------------------

PROCEDURE SETEIM1

        REPLACE FEEDER_POS WITH E1COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'E'
        REPLACE PNP_HEAD WITH 1
        STORE 'T' TO EIM1
        STORE E1COUNT + 1 TO E1COUNT
        STORE (NUM_PER_BD * EIMTIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD1WK TO HEAD1WK
RETURN

*----------------------------------------------------------------------

*----------------  PROCEDURE SETEIM2  ---------------------------------

PROCEDURE SETEIM2

        REPLACE FEEDER_POS WITH E2COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'E'
        REPLACE PNP_HEAD WITH 2
        STORE 'F' TO EIM1
```

```
        STORE WORKTIME + HEAD2WK TO HEAD2WK
RETURN

*--------------------------------------------------------------------

*-------------------- PROCEDURE SETBASE1 ----------------------------

PROCEDURE SETBASE1

        REPLACE FEEDER_POS WITH B1COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'B'
        REPLACE PNP_HEAD WITH 1
        STORE 'T' TO BASE1
        STORE B1COUNT + 1 TO B1COUNT
        STORE (NUM_PER_BD * BASETIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD1WK TO HEAD1WK
RETURN

*---------------------------------------------------------------------

*--------------------- PROCEDURE SETBASE2     -------------------------

PROCEDURE SETBASE2

        REPLACE FEEDER_POS WITH B2COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'B'
        REPLACE PNP_HEAD WITH 2
        STORE 'F' TO BASE1     ,
        STORE B2COUNT + 1 TO B2COUNT
        STORE (NUM_PER_BD * BASETIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD2WK TO HEAD2WK
RETURN

*----------------------------------------------------------------------

PROCEDURE DELAY
PARAMETER TIME
START_TIME = TIME()
TIME1 = VAL(SUBSTR(START_TIME,1,2))*3600+;
        VAL(SUBSTR(START_TIME,4,2))*60+;
        VAL(SUBSTR(START_TIME,7,2))
TIME2 = 0

DO WHILE TIME2 - TIME1 < X
STORE TIME() TO END_TIME
TIME2 = VAL(SUBSTR(END_TIME,1,2))*3600+;
        VAL(SUBSTR(END_TIME,4,2))*60+;
VAL(SUBSTR(END_TIME,7,2))
ENDDO
RETURN

*--------------------- PROCEDURE OPTREPORT ---------------------------

PROCEDURE OPTREPORT

 DEFINE WINDOW PARAMETER FROM 8,1 TO 14,75 DOUBLE
 ACTIVATE WINDOW PARAMETER

STORE 0 TO TOTCOMP,BASE1COUNT, BASE2COUNT, EIM1COUNT, EIM2COUNT
STORE 0 TO HD

RELEASE WINDOW PARAMETER
 DEFINE WINDOW PARAMETER FROM 1,1 TO 22,75 DOUBLE
```

```
USE COMP
INDEX ON NOZZ_SIZE TO NUM
CLEAR

@ 1,0

TEXT
DESCRIPTION          QUNTY  HEAD  FEEDER POS. BASE/EIM  NOZZLE  W.TIME
----------------------------------------------------------------------
ENDTEXT

GO TOP
DO WHILE .NOT. EOF()
@ X,0 SAY COMP_NAME
@ X,18 SAY NUM_PER_BD
@ X,28 SAY PNP_HEAD
@ X,36 SAY FEEDER_POS
@ X,49 SAY MOUNT_POST
@ X,55 SAY NOZZ_SIZE
@ X,63 SAY WK_TIME

STORE TOTCOMP + NUM_PER_BD TO TOTCOMP
IF MOUNT_POST = 'B' .AND. PNP_HEAD = 1
        STORE BASE1COUNT + 1 TO BASE1COUNT
ELSE
        IF (MOUNT_POST = 'B') .AND. (PNP_HEAD = 2)
                STORE BASE2COUNT + 1 TO BASE2COUNT
        ENDIF
ENDIF

IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 1)
        STORE EIM1COUNT + 1 TO EIM1COUNT
ELSE
        IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 2)
                STORE EIM2COUNT + 1 TO EIM2COUNT
        ENDIF
ENDIF

SKIP
STORE X + 1 TO X

IF X = 18
        STORE 5 TO X
ENDIF
ENDDO

@ 18,1
WAIT "Hit any key to continue"

RELEASE WINDOW PARAMETER
DEFINE WINDOW PARAMETER FROM 2,1 TO 22,75 DOUBLE
ACTIVATE WINDOW PARAMETER

CLEAR
IF HEAD1WK > HEAD2WK
        STORE (HEAD2WK/HEAD1WK)*100 TO PERCENT
        STORE 1 TO HD
ELSE
        STORE (HEAD1WK/HEAD2WK)*100 TO PERCENT
        STORE 2 TO HD
ENDIF

USE PRODUCT
@ 1,1 SAY ' OPTIMIZATION RESULTS FOR '+(JOB_NAME)+' PCB BOARD'
TEXT

                                   HEAD1                HEAD2
```

```
@ 3+2,1 SAY 'TOTAL NO. OF COMPONENTS ON BASE = '
@ 3+2,35 GET BASE1COUNT PICTURE "999"
@ 3+2,56 GET BASE2COUNT PICTURE "999"
@ 4+2,1 SAY 'TOTAL NO. OF COMPONENTS ON EIM = '
@ 4+2,35 GET EIM1COUNT PICTURE "999"
@ 4+2,56 GET EIM2COUNT PICTURE "999"
@ 6+2,1 SAY 'TOTAL NUMBER OF COMPONENTS = ' GET TOTCOMP PICTURE "9999"
@ 8+2,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD1 = ' GET HEAD1WK PICTURE "999.99"
@ 8+2,44 SAY 'Minutes'
@ 10+2,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD2 = ' GET HEAD2WK PICTURE "999.99"
@ 10+2,44 SAY 'Minutes'
@ 12+2,1 SAY 'OPTIMIZATION PERCENTAGE = ' GET PERCENT PICTURE "999.9"
@ 12+2,36 SAY '%'
@ 16,1
WAIT "Hit any key to continue"

RELEASE WINDOW PARAMETER

RETURN

*---------------------- PROCEDURE REOPT ------------------------------

PROCEDURE REOPT

IF HD = 1

        STORE 0 TO HD
        DO REOPT1
        DO OPTREPORT
        DEFINE WINDOW PARAMETER FROM 6,1 TO 14,75 DOUBLE
        ACTIVATE WINDOW PARAMETER
        CLEAR
        IF NEWPERCENT > PERCENT
        @ 2,2 SAY 'NO IMPROVEMENT IN OPTIMIZATION PERCENTAGE '
        ENDIF
        STORE 'N' TO MSELECT
        @ 4,2 SAY 'PRINT OUT RE-OPTIMIZATION RESULTS (Y/N) =>' GET MSELECT PI
        READ
        IF MSELECT = 'Y'
                DO PRINTOPT
        ENDIF
        RELEASE WINDOW PARAMETER

ELSE
IF HD = 2
        STORE 0 TO HD
        DO REOPT2
        DO OPTREPORT
        DEFINE WINDOW PARAMETER FROM 6,1 TO 14,75 DOUBLE
        ACTIVATE WINDOW PARAMETER
        CLEAR
        IF NEWPERCENT > PERCENT
        @ 2,2 SAY 'NO IMPROVEMENT IN OPTIMIZATION PERCENTAGE '
        ENDIF
        STORE 'N' TO MSELECT
        @ 4,2 SAY 'PRINT OUT RE-OPTIMIZATION RESULTS (Y/N) =>' GET MSELECT PI
        READ
        IF MSELECT = 'Y'
                DO PRINTOPT
        ENDIF
        RELEASE WINDOW PARAMETER

ENDIF
ENDIF

RETURN
```

```
                                * TRANSFERRING FEEDER FROM HEAD 1 TO HEAD 2

SELECT 6
CLEAR
DEFINE WINDOW PARAMETER FROM 7,3 TO 14,73 DOUBLE COLOR R+*
ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'RE-OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '    - DO NOT INTERRUPT -' COLOR W+

SELECT COMP

STORE 'F' TO DONE,FOUND ,FLAG,EF

SORT TO ASCEND ON NUM_PER_BD
USE ASCEND
*INDEX ON MOUNT_POST TO NUM
GO TOP

DO WHILE FOUND = 'F'
IF MOUNT_POST = 'B' .AND. PNP_HEAD = 1
        REPLACE FEEDER_POS WITH B2COUNT
        REPLACE MARKER WITH 'T'
        REPLACE MOUNT_POST WITH 'B'
        REPLACE PNP_HEAD WITH 2
        STORE B1COUNT - 1 TO B1COUNT
        STORE B2COUNT + 1 TO B2COUNT
        STORE (NUM_PER_BD * BASETIME) TO WORKTIME
        REPLACE WK_TIME WITH WORKTIME
        STORE WORKTIME + HEAD2WK TO HEAD2WK
        STORE HEAD1WK - WORKTIME TO HEAD1WK
        IF HEAD1WK =< HEAD2WK
        STORE 'T' TO FOUND
        ENDIF
ELSE
        SKIP
ENDIF
IF EOF()
        STORE 'T' TO FOUND
ENDIF
ENDDO

COPY TO COMP

STORE 6 TO X
DO DELAY WITH X

RELEASE WINDOW PARAMETER

RETURN

*-------------------- PROCEDURE REOPT2 --------------------------------

PROCEDURE REOPT2
                                * TRANSFERRING FEEDER FROM HEAD 2 TO HEAD 1
SELECT 6
CLEAR
DEFINE WINDOW PARAMETER FROM 7,3 TO 14,73 DOUBLE COLOR R+*
ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'RE-OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '    - DO NOT INTERRUPT -' COLOR W+

SELECT COMP

STORE 'F' TO DONE,FOUND ,FLAG,EF
```

```
SORT TO ASCEND ON NUM_PER_BD
USE ASCEND
GO TOP

DO WHILE FOUND = 'F'
IF MOUNT_POST = 'B' .AND. PNP_HEAD = 2

          REPLACE FEEDER_POS WITH B1COUNT
          REPLACE MARKER WITH 'T'
          REPLACE MOUNT_POST WITH 'B'
          REPLACE PNP_HEAD WITH 1
          STORE B2COUNT - 1 TO B2COUNT
          STORE B1COUNT + 1 TO B1COUNT
          STORE (NUM_PER_BD * BASETIME) TO WORKTIME
          REPLACE WK_TIME WITH WORKTIME
          STORE WORKTIME + HEAD1WK TO HEAD1WK
          STORE HEAD2WK - WORKTIME TO HEAD2WK
          IF HEAD2WK =< HEAD1WK
                    STORE 'T' TO FOUND
          ENDIF
ELSE
          SKIP
ENDIF
IF EOF()
          STORE 'T' TO FOUND
ENDIF
ENDDO

COPY TO COMP

STORE 6 TO X
DO DELAY WITH X

RELEASE WINDOW PARAMETER

RETURN


*---------------------- PROCEDURE METHOD2 --------------------------

PROCEDURE METHOD2


SELECT 6
CLEAR
DEFINE WINDOW PARAMETER FROM 7,3 TO 14,73 DOUBLE COLOR R+*
ACTIVATE WINDOW PARAMETER

@ 3,18 SAY 'OPTIMIZATION IN PROGRESS' COLOR W+
@ 4,18 SAY '  - DO NOT INTERRUPT -' COLOR W+
SELECT COMP
GO TOP
DO WHILE .NOT. EOF()
REPLACE MARKER WITH 'F'
SKIP
ENDDO

*----- STARTING OPTIMIZATION ROUTINE -------------

SORT TO DESC ON NUM_PER_BD/D
USE DESC
STORE 0 TO COUNT
GO TOP

DO WHILE .NOT. EOF()
```

```
                IF VERIFICAT = 'Y'
                        IF FIRST_FIND = 'F'.AND. HEAD1WK < HEAD2WK
                                DO WHILE NOZZ_SIZE = X
                                STORE 'F' TO EIM1
                                STORE 'T' TO EIM2
                                DO CHECKEIM1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO FIRST_FIND
                        ELSE
                        IF SECOND_FIND = 'F'
                                DO WHILE NOZZ_SIZE = X
                                STORE 'T' TO EIM1
                                STORE 'F' TO EIM2
                                DO CHECKEIM2
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO SECOND_FIND
                        ELSE
                        IF THIRD_FIND = 'F' .AND. HEAD1WK < HEAD2WK
                                DO WHILE NOZZ_SIZE = X
                                STORE 'F' TO EIM1
                                STORE 'T' TO EIM2
                                DO CHECKEIM1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO EIM1FULL
                                STORE 'T' TO THIRD_FIND
                        ELSE
                        IF FOURTH_FIND = 'F'
                                DO WHILE NOZZ_SIZE = X
                                STORE 'T' TO EIM1
                                STORE 'F' TO EIM2
                                DO CHECKEIM2
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                                ENDDO
                                STORE 'T' TO EIM2FULL
                                STORE 'T' TO FOURTH_FIND
                        ELSE
                                DO CHECKBASE1
                                IF .NOT. EOF()
                                        SKIP
                                ENDIF
                        ENDIF
                        ENDIF
                        ENDIF
                        ENDIF
                ELSE
                        DO CHECKBASE1
                        IF .NOT. EOF()
                                SKIP
                        ENDIF

                ENDIF
        ELSE
                SKIP
        ENDIF
ENDDO
COPY TO COMP
```

```
REPLACE HEAD1_TIME WITH HEAD1WK
REPLACE HEAD2_TIME WITH HEAD2WK


CLOSE ALL
STORE 4 TO X
DO DELAY WITH X

RELEASE WINDOW PARAMETER

DO OPTREPORT

IF PERCENT < 95
        STORE PERCENT TO NEWPERCENT

        DEFINE WINDOW PARAMETER FROM 8,1 TO 14,75 DOUBLE
        ACTIVATE WINDOW PARAMETER
        CLEAR
        STORE 'N' TO MSELECT
@ 1,2 SAY 'OPTIMIZATION PERCENTAGE LESS THAN 95%'
@ 3,2 SAY 'RE-OPTIMIZE TO IMPROVE OPTIMIZATION PERCENTAGE (Y/N) => ' GET MSEL
READ
        CLEAR
        STORE 'N' TO SELECT
@ 2,2 SAY 'PRINT OUT RESULTS FROM INITIAL OPTIMIZATION (Y/N) => ' GET SELECT
READ

IF SELECT = 'Y'
        DO PRINTOPT
ENDIF

IF MSELECT = 'Y'
        DO REOPT
ENDIF

*RELEASE WINDOW PARAMETER

ENDIF

RETURN


*---------------------------------------------------------------

PROCEDURE PRINTOPT

STORE 0 TO TOTCOMP,BASE1COUNT, BASE2COUNT, EIM1COUNT, EIM2COUNT
*CLOSE ALL
SELECT COMP
CLEAR
@ 2,2 SAY 'HIT ANY KEY TO START PRINTING =>'
WAIT ""
SET DEVICE TO PRINT
STORE 'T' TO PRINT
@ 1,0 SAY 'DESCRIPTION                 QUNTY  HEAD  FEEDER POS. BASE/EIM  NOZZLE
@ 2,0 SAY '-----------------------------------------------------------------------
GO TOP
DO WHILE .NOT. EOF()
@ X,0 SAY COMP_NAME
@ X,18 SAY NUM_PER_BD
@ X,28 SAY PNP_HEAD
@ X,36 SAY FEEDER_POS
@ X,49 SAY MOUNT_POST
@ X,55 SAY NOZZ_SIZE
@ X,63 SAY WK_TIME

STORE TOTCOMP + NUM_PER_BD TO TOTCOMP
```

```
        IF (MOUNT_POST = 'B') .AND. (PNP_HEAD = 2)
                STORE BASE2COUNT + 1 TO BASE2COUNT
        ENDIF
ENDIF

IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 1)
        STORE EIM1COUNT + 1 TO EIM1COUNT
ELSE
        IF (MOUNT_POST = 'E') .AND. (PNP_HEAD = 2)
                STORE EIM2COUNT + 1 TO EIM2COUNT
        ENDIF
ENDIF

SKIP
STORE X + 1 TO X

ENDDO

IF HEAD1WK > HEAD2WK
        STORE (HEAD2WK/HEAD1WK)*100 TO PERCENT
        STORE 1 TO HD
ELSE
        STORE (HEAD1WK/HEAD2WK)*100 TO PERCENT
        STORE 2 TO HD
ENDIF
SKIP
@ X-1,0 SAY '                                    HEAD1                 HEAD2
@ X,0 SAY '                                     -----                  -----
@ X+1,1 SAY 'TOTAL NO. OF COMPONENTS ON BASE = '
@ X+1,35 SAY BASE1COUNT PICTURE "999"
@ X+1,56 SAY BASE2COUNT PICTURE "999"
@ X+2,1 SAY 'TOTAL NO. OF COMPONENTS ON EIM = '
@ X+2,35 SAY EIM1COUNT PICTURE "999"
@ X+2,56 SAY EIM2COUNT PICTURE "999"
@ X+4,1 SAY 'TOTAL NUMBER OF COMPONENTS = '
@ X+4,29 SAY TOTCOMP PICTURE "9999"
@ X+6,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD1 = '
@ X+6,34 SAY HEAD1WK PICTURE "999.99"
@ X+6,44 SAY 'Minutes'
@ X+8,1 SAY 'TOTAL PLACEMENT TIME FOR HEAD2 = '
@ X+8,34 SAY HEAD2WK PICTURE "999.99"
@ X+8,44 SAY 'Minutes'
@ X+10,1 SAY 'OPTIMIZATION PERCENTAGE = '
@ X+10,27 SAY PERCENT PICTURE "999.9"
@ X+10,36 SAY '%'
*@ 16,1

SET DEVICE TO SCREEN
RETURN
```

Author: Chodos Mark Steven.
Name of thesis: The Design And Implementation Of A Flexible Manufacturing System For A Surface Mounting Production Line.