Transfer Report (From MPhil to PhD)

The Development of a Resilient Control System for Flexible Manufacturing Cells

Aram Taholakian

November 1993

Synopsis

The aim of the research is to develop a Resilient Control System for Flexible Manufacture. The research previously carried out in this area by P. Gray claims to have produced a methodology to develop dependable software for the control of a Flexible Manufacturing Cell (FMC).

The MPhil work has consisted of an extensive literature search in the fields of Flexible Manufacturing Cells, Parallel Processing, Transputers and Petri Nets. The work to date has also given Gray's methodology a thorough investigation. These are discussed further in the body of this report.

The FMC at the School of Engineering (S.O.E.) forms the basis of the research. The control of the FMC is divided into three levels as follows:
Level 1; the control of sensors, actuators etc. using PLCs (Programmable Logic Controllers)
Level 2; the control of m/c tools, robots and PLCs controlling Level 1
Level 3; decision making level

However Gray's methodology gives no consideration to Level 1 or the possible errors occurring within the FMC. An FMC is an environment in which the reliability of its constituent elements cannot be guaranteed. The work to date has focused on the complete and smooth operation of the devices in Level 1, constantly testing them and monitoring any errors occurring during the process. All PLCs in this level were reprogrammed to accommodate these or any other possible errors occurring during the control of the FMC.

The MPhil work to date has also consisted of the completion of the School's FMC which included the fixture and PLC programming of the Gantry Robot. In addition the appropriate Transputer and interfacing hardware to carry out the control of Levels 2 and 3 was investigated and chosen. The hardware will be used for the development and control of the FMC at all three levels to achieve a resilient control system during the PhD phase of the project.

The report provides evidence that the development of dependable software concentrating on Levels 2 and 3 is not adequate for the control of a system such as a Flexible Manufacturing Cell. However Gray's methodology has potential for modular growth.

The PhD work will produce a methodology for the development of a resilient Transputer based system accounting for all levels of the control. The system will include an Error Handling process concurrently with the Decision Making and Status Handling processes and will handle errors effectively and safely.

It is my belief that the future work will produce a methodology for the development of highly efficient Flexible Manufacturing Cells. The work is novel and will be a contribution to knowledge.

Key Words

FMC - Flexible Manufacturing Cell

S.O.E. - School of Engineering

S.H.U. - Sheffield Hallam University

PLC - Programmable Logic Controller

Table Of Contents

1. Introduction

Flexible Manufacturing Cells (FMCs) have been well established for a number of years and are used in batch manufacturing industries to improve the efficiency of production. They consist of a number of computer numerical controlled (CNC) machine tools, automated work/tool handling equipment and a control system to synchronise the operation of the machine tools and handling equipment.

The safe and reliable operation of FMCs is clearly desirable, if not essential for their effective use. However FMCs are complex systems the elements of which operate concurrently, and interact at irregular times depending upon the components to be produced. Therefore the development of a control system is not a trivial affair[1]. The control system not only consists of a computer algorithm, but also the computer hardware, the communications protocols and the cell monitoring equipment (sensors, transducers etc.).

Flexible manufacturing operational strategies such as distributed control are now established practices in industry. Real time control is achieved by distributing the various functions of a control system which operate concurrently. It is essential however that these functions or elements communicate with each other to achieve complete synchronisation of the whole system. Such communication could be achieved by networking individual PCs, which each run sequential programs. The control of the FMC in the School of Engineering (S.O.E.) at Sheffield Hallam University (S.H.U.) is currently carried out using this method. However as the number of processors increases, the management overhead increases[2].

The programming strategies and languages available for conventional microprocessors do not model a parallel architecture[3]. Apart from developing the program for each processor separately from others, the programmer has to introduce techniques for synchronisation and communication of one processor with others. However the software is directly dependant on the hardware. By increasing the number of processors, the complexity of the software design, development and testing will multiply. Expansion and modification of the operation of the hardware requires major modification of the software.

A Transputer based parallel architecture is capable of overcoming the above mentioned problems. Occam, the software developed by Inmos, is a combination of sequential and parallel programs executed by the Transputer. It is a powerful and expressive calculus for describing concurrent algorithms[4].

2. The FMC at the S.O.E.

The FMC consists of a CNC lathe, CNC milling machine, and three work handling stations;

- A loading and unloading station for raw material and finished parts.
- A station at the lathe for loading and unloading the lathe.
- A station at the miller for loading and unloading the miller.

The loading and unloading of raw materials and finished parts is carried out by a Puma robot. The loading and unloading of the lathe is carried out by a gantry robot, while a cylinder loads and unloads the miller by ramming a workpiece to and from the milling table. A conveyor track transports material to and from the stations. This equipment is referred to as the work handling equipment (Fig. 1).

2.1. Levels of Control

The equipment used to perform the logical control of all the elements within the FMC is distributed between PCs and PLCs (Programmable Logic Controllers), and divided into three levels of control as shown in (fig 2).

Level 1

The simple sequential control of primitive devices, such as actuators and sensors, is carried out in level 1 using PLCs.

One PLC is devoted to Station 1 (fig 1), also referred to as the Puma Station, and to the conveyor. The opening of the dogs on the conveyor, closing them when the pallets have moved on, and the monitoring of the appropriate sensors at the Puma Station, to determine the arrival/departure of pallets at/from the station, is carried out through this PLC.

A second PLC is devoted to Station 2 or the Miller Station. It monitors the appropriate sensors at the Miller Station to determine the arrival of the pallets, activates the loading/unloading cylinder, the milling table clamp and instructs the miller to start or stop machining. The PLC also checks the Machining in progress status.

A third PLC is used to control and monitor Station 3, or the Lathe Station, and the gantry robot, synchronising the opening and shutting of the lathe door with the entry and exit of the robot arm whilst loading and unloading workpieces. The execution of the Start Machining command, and the monitoring of the machining is also carried out through this PLC.

Level 2

The control of the machine tools, robots and the PLCs conducting Level 1 control is achieved in this level.

The Puma and conveyor PC instructs the Puma robot to execute the program "Loadcell", which is used to place a blank from the raw material stock in the vice, or "Unloadcell" (finished parts). It also instructs the conveyor PLC to index pallets.

The Miller PC instructs the Miller PLC to transfer the vice at the Miller Station into the fixture on the milling table and clamp it in place, or vice versa.

The Lathe and Gantry PC instructs both the gantry robot, to load or unload the lathe, and the lathe station PLC, to monitor the arrival of pallets and the opening and closing of their vice accordingly.

Level 3

Level 3 is conducted by the Cell Controller. The Cell Controller's task is to ensure that a given schedule of jobs is processed by the cell.

For example; a) instructing the Lathe PC to start the CNC program for turning. b) instructing the Miller PC to start the CNC program for milling and c) Instructing the Puma and conveyor PC to unload a finished component from the cell.

2.2. Distributed Control

Having established that the control of the S.O.E.s FMC is achieved by distributed control, it is important to explain the reasons behind such a design. Why, for instance, is the whole cell not controlled by only one computer?

Two essential reasons;

a) To reduce the complexity of the control algorithms/computer programs.

Even with the small FMC at the S.O.E., the number of sensors that need monitoring, and the number of actuators which need operating is very large. Since the various parts of the cell operate concurrently, the control of such an environment by one sequential algorithm becomes virtually impossible, and certainly inefficient. Changes within the cell would mean redesigning the complete control algorithm, hence making future growth a very difficult task. There follows a simple example of concurrency within the cell.

Example; a part is to be unloaded from the conveyor by the Puma robot, also a part is to be loaded onto the miller and whilst this is being carried out, the lathe finishes machining and requires its workpiece to be removed.

b) To achieve the required communications necessary to operate the machine tools and work handling equipment.

There are several types of data that have to be communicated within the cell. For example, whether a sensor is on or off, which program the Puma is to run, what component is to arrive at the Miller. This data is communicated in different ways depending on the complexity of the data. For instance, a sensor's data may be on or off (24 or 0 volts) and requires only two wires to transmit its data. However, instructing the Puma to run a certain program will require the name of the program to be communicated and the command to execute it (e.g. "EXEC LOADPART"), hence it uses a much more sophisticated communication method (in this case RS 232C). Other data may be conveyed as messages specific to the products being produced, e.g. "job 12 requires turning and milling, CNC program numbers 7 & 4".

A Local Area Network (LAN) is required to transmit data between the PCs because there is the chance of more than one PC wishing to communicate at once. For instance the Lathe Controller and the Miller Controller might both communicate with the Cell Controller at the same time that the Cell Controller wishes to instruct the Puma Controller to load the conveyor. The LAN's communication protocol ensures that none of the messages are lost.

3. Petri Nets, Transputers, Occam and FMCs

Over the years, Petri Nets have been successful in modelling safety critical systems[5]. FMCs are safety critical systems in that, although automated they work in conjunction with humans. Also, modelling an FMC greatly reduces faults in the system by removing them before use. Section 5.2 discusses the possible faults or errors within the FMC.

Occam and Transputers considerably simplify the design of a concurrent system such as an FMC. However recent research carried out within the S.O.E. has shown that in modelling the School's FMC the Petri Net graph of the cell becomes very complex (fig. 3). The graph is not readable and difficult to follow. In addition, the graph cannot be modelled simply in Occam due to the parallelism being hidden, which provides no easy access for modular growth. Therefore dependable software cannot be produced from the Petri Net.

For these reasons and inspired by the capabilities of Occam and Petri Nets, a methodology was developed by Mr. P. Gray (PhD student) and supervisors Dr. W. M. M. Hales and Prof. F. Poole[6][7], to produce dependable distributed control systems for flexible manufacture (section 4).

It is important however to discuss Transputers, Occam and Petri Nets, in more detail, for a better understanding of the methodology.

3.1. Transputers and Occam

The combination of the Inmos developed hardware and software called Transputer and Occam respectively is rapidly being recognised as a solution to the problem of programming concurrent systems of all kinds[3] .

A Transputer is a microcomputer with its own local memory, and with links for connecting one Transputer to another Transputer[8]. It can be used in a single processor system, or in networks to build high performance parallel architectures. By linking processors together, a linear increase in data processing capacity can be achieved, as opposed to the limited processing capacity of typical multi processor control systems (MPCS). MPCS can only be increased to a certain limit before experiencing a drop off in effective computing power[9].

3.1.1. Occam: Language Definition

Occam simplifies the writing of concurrent programs by taking most of the burden of synchronisation away from the programmer[10]. Occam uses channels for communicating values and does not mind whether the two processes (programs) which are to communicate are running on different computers, or concurrently on the same computer. However channels are one way only, and therefore two would be needed for a two-way communication.

Although Occam provides synchronised communication, the programmer is still left with the responsibility of avoiding "deadlock", i.e. a process waiting for something that will never arrive, for it is prepared to do so forever . For the non professional programmer, or rather the professional engineer and system designer, avoiding deadlock, in even a relatively simple FMC such as the one in the S.O.E., is difficult.

3.2. Petri Nets

Petri nets are a tool for the study of systems, and a mathematical representation of systems[11]. The application of Petri Nets is mainly through modelling.

The structure of a Petri Net is as follows; a set of places (represented by circles on a Petri Net graph, fig. 4), a set of transitions (bars on a graph), input and output arcs (arrows), and a marking which is an assignment of tokens to the places of a Petri Net (represented by dots on a graph). Directed arcs connect the places and the transitions. Arcs directed from a place to a transition define the place to be an input of the transition. Arcs directed from a transition to a place define the place to be an output of the transition. The execution of a Petri Net is controlled by the number and distribution of tokens in the Petri Net[8]. Execution is carried out by firing transitions. However a transition may only fire if it is enabled, i.e. each of its input places has at least as many tokens as there are arcs from the place to the transition, (fig. 5). Deadlock is ruled out of a Petri Net by making sure all transitions are capable of firing.

4. Methodology

The methodology highlights the simplicity of both Occam and Petri Nets. It demonstrates the effective use of both for the design and development of dependable software for a distributed control system, such as the FMC at S.H.U..

Rather than using Petri Nets to merely model the system, the methodology proposes the use of Petri Nets to design the system , from which a dependable Occam code could be produced. Although in general Occam code is generated from data flow diagrams (DFDs)[12], the use of Petri Nets in the same way is more beneficial. Petri Nets are capable of analysing a system as well as modelling it. Concurrency is explicitly represented in the Petri Net (sub-section 4.1), making it capable of modular growth (section 6), thus simplifying the design process.

The five stages of the development of a computer system are the requirements and specifications, design\modelling, program design, implementation and finally, testing and maintenance[13] . The conventional method of developing such systems is treating these stages separately, where the appropriate formal and non formal techniques are used to help develop systems at each stage. However techniques such as SSADM[14] (Structured System Analysis and Design Methodology) and SIFT (Software Implemented Fault Tolerant) operating systems[15] which cover more than one stage, are currently being developed.

The methodology aims to produce a dependable system by concentrating on the top stages, defined as the requirements, design and program code levels. It uses Petri Nets at the design level and Occam at the program code level. However, while the SIFT operating system insists on verifying that both the design and the code meet the requirements , the methodology uses Petri Nets to design the model directly from the requirements, in such a way that it can be accurately converted into Occam. Therefore simplifying the verification process and hence contributing to the simplification of the design process.

4.1. The Methodology and the S.O.E.s FMC

The methodology simplifies the design process of the FMC in that it clearly divides the process into three steps.

Step 1

Since the aim is to design the system from the requirements, using Petri Nets, the first step of the methodology is the identification of the operations which could take place concurrently within the cell. For instance, although the milling machine can not be loaded or unloaded while it is in the process of machining a component, other operations such as loading or unloading the lathe, loading or unloading the conveyor, or rotating the conveyor could be desirable concurrent with the milling process.

By identifying the above mentioned operations, the distribution of the control system through the relevant elements is clarified. The following table represents the possible concurrent operations and the distributed elements (controllers) of the control system.

Parallel Operations	Elements
Lathe (loading, machining, unloading)	Lathe controller
Miller (loading, machining, unloading)	Miller controller
Puma robot (loading, unloading)	Puma robot controller
Conveyor (transporting workpieces)	Conveyor controller
Decision making (load lathe, etc.)	Cell controller
Monitoring the status of the cell	Status handler

Table of parallel operations and distributed elements within the cell

Step 2

The second step of the methodology is to separately produce a Petri Net graph for each of the controllers, clearly listing its inputs (where they come from), its logical operations and outputs (where they go to). Fig. 6 gives an example of two of the logical operations carried out by the cell controller. The places on the left represent the inputs to the cell controller. The places on the right represent outputs from the cell controller which are sent to become input places of other controllers. The transitions for each output place are also clearly marked.

Fig. 6 shows that to load the lathe (for instance), all the conditions (input places) have to be determined first. These conditions are as follows: There is a pallet at the lathe, the pallet is full (has a part in it), the part is to be machined on the lathe, the lathe is idle, the lathe work handling equipment is also idle (gantry robot and transfer device).

Once this procedure is carried out fully, it becomes apparent that these input places are the output places from another controller's Petri Net, in this case the Status handler (fig. 7).

The complete logical operation of the Cell controller is shown in fig. 8.

Step 3

The third step is graphically linking the individual Petri Nets of all the controllers to produce a Petri Net for the entire control system. Fig. 9 represents the general layout of the system, and unlike fig. 3 it clearly shows the parallelism and the communications between the separate controllers. The controllers are laid out in such a way that the communications between them is one way, left to right (fig. 9). The purpose of this is to eliminate 'deadlock' from the system.

For instance, the Cell controller sends the command "load lathe" to the Lathe controller. The Lathe controller does not report back to the Cell controller, but the Status handler, hence keeping the direction of communication to one way. However there still remains the possibility of 'deadlock' between the Status handler and the Cell controller. The Cell controller receives data from the Status handler (the status of the cell) as well as sending it data, hence the potential for 'deadlock'. This is ruled out of the control system when converting the Petri Net into Occam code, by using the ALT (Alternation) command. The Status handler receives messages from the various elements of the system, including the Cell controller, by Alternation. However, the Status handler reports back to the Cell controller only if requested to do so by the Cell controller, hence avoiding the risk of 'deadlock'. *For more detail refer to [10].*

5. Progress of Research

To establish a good foundation for the research, the initial period was devoted to the full understanding of the working of the FMC and the previous research carried out.

In conjunction with the extensive literature survey carried out in the fields of FMCs, parallel processing, Transputers, Occam, petri nets and P. Gray's research, regular tests were carried out on the smooth and safe operation of the devices in level 1 (section 2.1.). As a result of this additional sensors were included and certain actuators upgraded within the cell. Hence the control of level 1 was revised. The program for each PLC was therefore modified to accommodate these changes.

A full understanding of PLCs was attained during this period. The Lathe PLC was programmed to control and monitor Station 3, the status of the lathe and the gantry robot. Unlike the PLCs at Stations 1 and 2 which were programmed using Ladder Diagrams, the Lathe PLC was programmed using Step Ladder Diagrams. Due to the complexity of the operations occurring at Station 3 a Mitsubishi F2 series PLC was used because of its ability to execute Step Ladder Programs[16] [17].

5.1. PLC Programming

A PLC program consists of a ladder diagram (fig. 10) or a sequence of ladder diagrams[18]. The required actions of the program are represented sequentially by lines on the ladder diagram[19]. The input signals to the PLC are marked on the left hand side of the diagram. These signals may be on or off and form the conditions for the output signals (instructions) from the PLC which are marked on the right hand side of the ladder. The input and output contacts (I/Os) of the PLC are clearly marked on the ladder (I = X0, X1, X2 etc., O = Y30, Y31, Y32 etc.). The PLC's internal relays are also marked as M numbers (M100, M101, etc.).

A software package called MEDOC (Mitsubishi Electric Documentation) was used to produce the PLC programs. Once the I/Os and internal relays are labelled, MEDOC offers the option of showing the description of the I/Os and internal relays on the ladder diagram itself. The software also checks the format of the ladder and points out illegal operations. In addition, it provides downloading facilities from a PC to a PLC and verifies that the downloaded program matches with the one on the PC.

However, although sequential, a PLC constantly scans through a ladder program in a cyclic fashion and only terminates if all the instructions are **successfully** carried out, or if an **error** occurs and generates an Error Status. *Sub-section 5.3.1 discusses both terms further stating their importance in the creation of a resilient control system.* Therefore care must be taken when including opposing and repeated instructions within a ladder. The input conditions must be carefully determined for each output instruction to avoid future complications (fig. 11).

Although the PLC at the Puma Station controls and monitors Station 1 and the conveyor, the ladder program for this particular PLC is relatively simple. The reason for this is that the programming of the Puma robot is carried out using a PC. *Appendix 1 represents the ladder program for the PLC at Station 1, clearly showing the five sequential operations involved during the loading or unloading of the cell. It also lists the sensors monitored and the timers controlled by the PLC.*

The task of the PLC at Station 3 is more complex than those of stations 1 and 2, in that it controls and monitors the gantry robot as well as the lathe and Station 3. There are thirteen sensors on the gantry alone that need monitoring by the PLC. Also the high number of operations (movements) involved in loading or unloading the lathe and the determination of the correct condition for each instruction (made by the PLC) make it very difficult, if not impossible, to control the gantry, the lathe and Station 3 using a ladder program. Therefore the programming of the PLC in this case was carried out using Step Ladders.

5.1.1. Step Ladder Programs

As the name suggests, the use of Step Ladders allows a complex PLC program to be divided into a number of steps[20] , depending on the complexity of the operations carried out by the PLC.

Each operation is given a step number at the beginning of the Step ladder and an input address which forms the condition for that particular Step Ladder to be Set (to start). The Step Ladder is Reset only if the required operation is carried out successfully. The layout of the required actions in between the Set and the Reset lines is similar to that of a standard Ladder diagram. However repeated instructions carried out by the PLC do not cause programming complications as they can be grouped in separate Step Ladders. *Appendix 2 represents the Step Ladder Program for the PLC at the Station 3.* For example the instruction "Vert Down", instructing the gantry robot to go vertically down, occurs four times during the loading and the unloading of the Lathe. However, by dividing the "Load Lathe" operation into two Steps "Grip Workpiece" and "Workpiece to Chuck" for instance, the PLC will instruct the gantry to go vertically down again only after Step S601 is Set and Reset, and Step S602 is Set (appendix 2, pages 4 & 7). Therefore the programming of the PLC is simplified and the risk of errors reduced.

5.2. Choosing the Appropriate Hardware

In order to carry out the control involved in Levels 2 and 3 as suggested by Gray's methodology, the research to date has also concentrated on the task of choosing the appropriate Transputer hardware for the FMC to replace the existing network of PCs.

There are many types of Transputer hardware available in the market today. Transtech, Sension, Parsytec and Inmos are some of the suppliers of such devices. The included features of the Transputers are reflected in their prices and vary from the basic T400 model to the latest, top of the range T9000.

However the T400 Transputer has only two links which means that the only possible network topology is a linear chain[21] as shown in fig. 12. With the upper models, such as the T425's and the T805's for instance, one can go further and configure a tree or a toroidal mesh[21] (fig. 12).

The configuration of the controllers and the communication protocol within the Cell is shown in fig. 13. The need for Transputers with four links each is clearly seen. The availability of the free links on the Lathe, Miller and Puma Controllers allow future modular growth, as intended by the research in the attempt to build resilience into the system.

The more recent Transputers such as the T805's and the T9000's are 'super-fast' micro computers which are ideal for Multitransputer Workstations[22] and Database Systems for banks[23]. However with the FMC at the S.O.E. the aim was not to exploit the speed of Transputers, but to acquire Transputer hardware capable of handling the communication with the various parts of the system.

The Status Handler is currently a process within the Cell Controller, as suggested by Gray's methodology. As part of the PhD program further research will be carried out regarding the Status Handler and an additional element for the handling of errors (section 6). This is why no decision has yet been made regarding the type and configuration of the final hardware for the complete system. However enough hardware has been purchased from Transtech to carry out the control of one of the stations within the FMC adequately and cost effectively.

This hardware comprises of a TMB04 PC Transputer board featuring an IMST425 Transputer (Cell Controller) and four slots for TRAM daughterboards. A size one T425 Transputer TRAM (Lathe Controller or Miller Controller or Puma Controller) occupies one slot, a size one TTM21 RS232-C TRAM (to communicate with Lathe, Miller or Puma) occupies another and a size two IOT332 digital I/O TRAM (to communicate with PLCs) the final two slots, (appendix 3). As the name suggests, the TMB04 PC board is PC compatible, which allows the Cell Controller to communicate with the PC, i.e. report to the monitor.

5.3. Possible Errors Within the Schools FMC

The research carried out by Mr. P. Gray has produced a methodology which aims to produce reliable software for a distributed control system, and is based on the schools FMC. The aim of the current work is to research the resilience of the whole system. The identification of possible errors within the system and the handling of such errors forms a major part in the production of a resilient system.

An important part of the work to date has been to identify the possible errors occurring within the cell and categorising them as below:

Work Handling Errors

- Failure of sensors, solenoids, valves, etc.
- Manipulating errors (positioning, speed, etc.)
- Material errors (size, shape, alignment, position, etc.)
- Robot failure
- Power failure (air, electricity)

Machine Tool Errors

- Catastrophic failure
- Gradual failure (wear)
- Work handling (fixturing) failure
- Tool failure
- CNC program errors

Control Errors (Hardware and Software)

- Correctness, reliability
- Communication errors (timing of messages, message corruption, deadlock)
- Initial data errors
- Logic of control
- Speed of software
- Failure of hardware (PLCs, Transputers, trams, digital I/O)
- Power failure

Human Errors

- Misuse, inappropriate use
- Mistakes (incorrect choice of material, tools etc.)

The methodology in section 4 claims to be a tool for developing a dependable distributed control system for flexible manufacture. Although the FMC at the S.O.E. forms the basis of Gray's research, no consideration has been given to any of the above mentioned errors which are likely to occur within the cell. However Gray claims that the methodology simplifies the modular growth of the dependable software, in an effort to build resilience into the complete system (section 6).

5.3.1. Handling of Errors

Having identified the possible errors which are likely to occur during the control of the Cell, it is important to discuss the various methods by which these errors are handled.

Ideally the elimination of the chance of errors occurring is most favourable. In reality this is not always possible due to the unpredictability of some errors. For instance, in the case of gradual failure of machine tools, efforts such as constant preventative maintenance and statistical process control will help eliminate errors. However a work handling device, such as an Infra-Red sensor, could fail suddenly and therefore needs to be accounted for by the PLC program to avoid catastrophic consequences.

It is also essential to establish a good understanding of the terms **successful** and **error** during the control of a resilient FMC. This is best explained with an example. The PLC at Station 3 instructs the Gantry robot to load the lathe. A fault with the material results in the diameter of the bar (for instance) in the vice being smaller than specified. Although failing to grip the workpiece in reality, the robot will carry out the operations "GRIP WORKPIECE" and "WORKPIECE TO CHUCK" successfully, hence transporting "fresh air" which in turn is machined by the Lathe. Such control may be dependable, and even safe for that matter, it is not however in any way resilient.

Other considerations such as deciding the 'Robot to Safe Position' for the Gantry robot is also important. It was decided not to leave the arm of the robot directly above the Lathe after transporting a workpiece to the chuck. Tests showed that in the event of air supply failure, the robot arm will only remain in the 'Up' position for a limited amount of time before rapidly falling. This would result in the arm breaking the glass door at the top of the Lathe and colliding with the rotating chuck. Therefore an extra Step Ladder was included in the programming of the Gantry robot, taking the arm of the robot to a much safer position, above the conveyor.

However not all errors can be dealt with in Level 1. Communication errors such as message corruption and deadlock , and even failure of hardware such as PLCs and Transputers are classed as control errors and should be accounted for by the software in the top levels of the control. Human rectifications, or rectifications by the operator are also an important part of the error handling process and need to be carefully dealt with by the Cell Controller at the decision making stage (section 6).

6. Future Work for the PhD

The most effective way of handling errors in any system is by eliminating them prior to their occurrence. However in an environment such as a Flexible Manufacturing Cell the reliability of the constituent elements cannot be guaranteed. This does not imply that the achievement of resilience is an impossible task. A reliable system failing catastrophically is less resilient than a relatively unreliable one failing safely. Therefore the handling of such possible errors efficiently is very important.

The research carried out to date has focused on the reliable and safe control of the devices in Level 1. It is my belief that by continuing the current research further and focusing on the control carried out in Levels 2 and 3 in conjunction with Level 1, a resilient control system can be achieved as a step forward from the dependable control claimed by Gray.

The PhD work will explore Mr. Gray's methodology further, investigating its capabilities of modular growth without the complete alteration of the original design and code. The aim is to include an additional element in parallel with the existing controllers (listed on page 8). This element will be named the Error Handler the task of which will be the constant monitoring of the errors occurring within the Cell.

However, similar to the Status Handler, although the Lathe Controller, the Miller Controller and the Puma/Conveyor Controller would all report to the Error Handler and not the Cell Controller, there would still be the potential for 'deadlock' between the Cell Controller and the Error Handler. Careful consideration is needed to address this problem fully. The frequency at which the Cell Controller demands updates from the Status and Error Handlers, the configuration of the Error and Status processes and the communication between them are all areas in need of further research.

The possible errors occurring during the control of Level 1 have been identified and accounted for by the PLCs, and will be constantly studied further during the PhD work period. By further notifying the related Controllers, who in turn send the data to the Error Handler, these errors will be differentiated by the Cell Controller. The Cell Controller's task then is to make sure these errors are clearly labelled (source and type of error) and acknowledged by the operator, who can in turn track and rectify them.

However, in order to achieve resilience, it is important that the Cell Controller handles errors efficiently and makes the correct decisions. For example, in the event of an error on behalf of the Gantry robot during the loading of the Lathe, and if the Miller is being loaded at the same time, is it safe for the Cell Controller to instruct the operator to attend to the Gantry or does it need to instruct the Miller Controller to stop loading the Miller first, or even temporarily ignore the error until the Miller is Loaded.

The various possible configurations of processors and processes will also be considered to determine the best layout for the resilient control of the FMC. For example, similar to the Status Handler, the Error Handler could be included as a process within the Cell Controller, or as a process within the Status Handler whereby the Status Handler requests error updates from it and in turn report back to the Cell Controller, or even as a separate processor reporting back to its own station (PC monitor) and with its own decision making process.

The procedures of the methodology will be followed and a Petri Net for the whole FMC produced. Once designed and successfully modelled, the Petri Net will be converted into Occam code. The Transputer hardware to carry out the complete control of the whole system will be chosen and installed during this period.

In summary, a dependable control system concentrating on the top levels is not adequate for the development of a resilient FMC, nor is the safe and reliable control of the lower level. However it is my belief that by further researching into both areas and the existing methodology, a novel methodology can be produced for the development of a resilient control system for Flexible Manufacturing Cells.

7. List of Figures

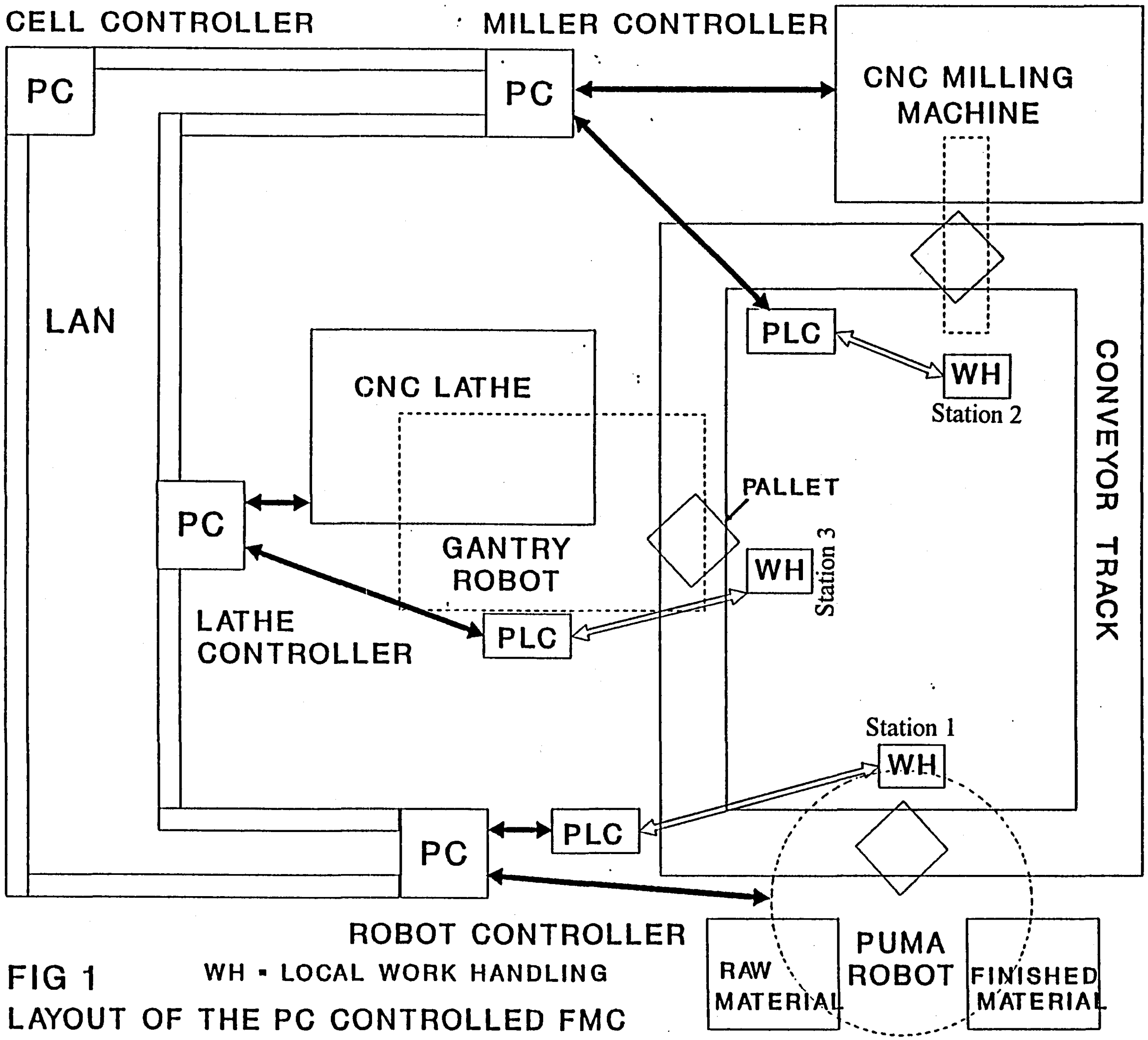


FIG 1
LAYOUT OF THE PC CONTROLLED FMC

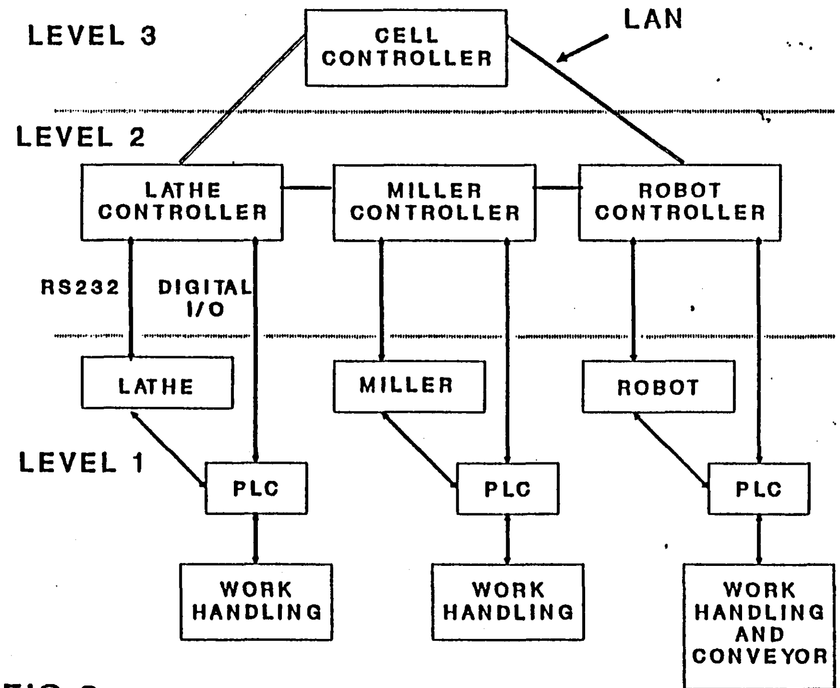


FIG 2
THE LEVELS OF CONTROL IN THE PC CONTROLLED FMC

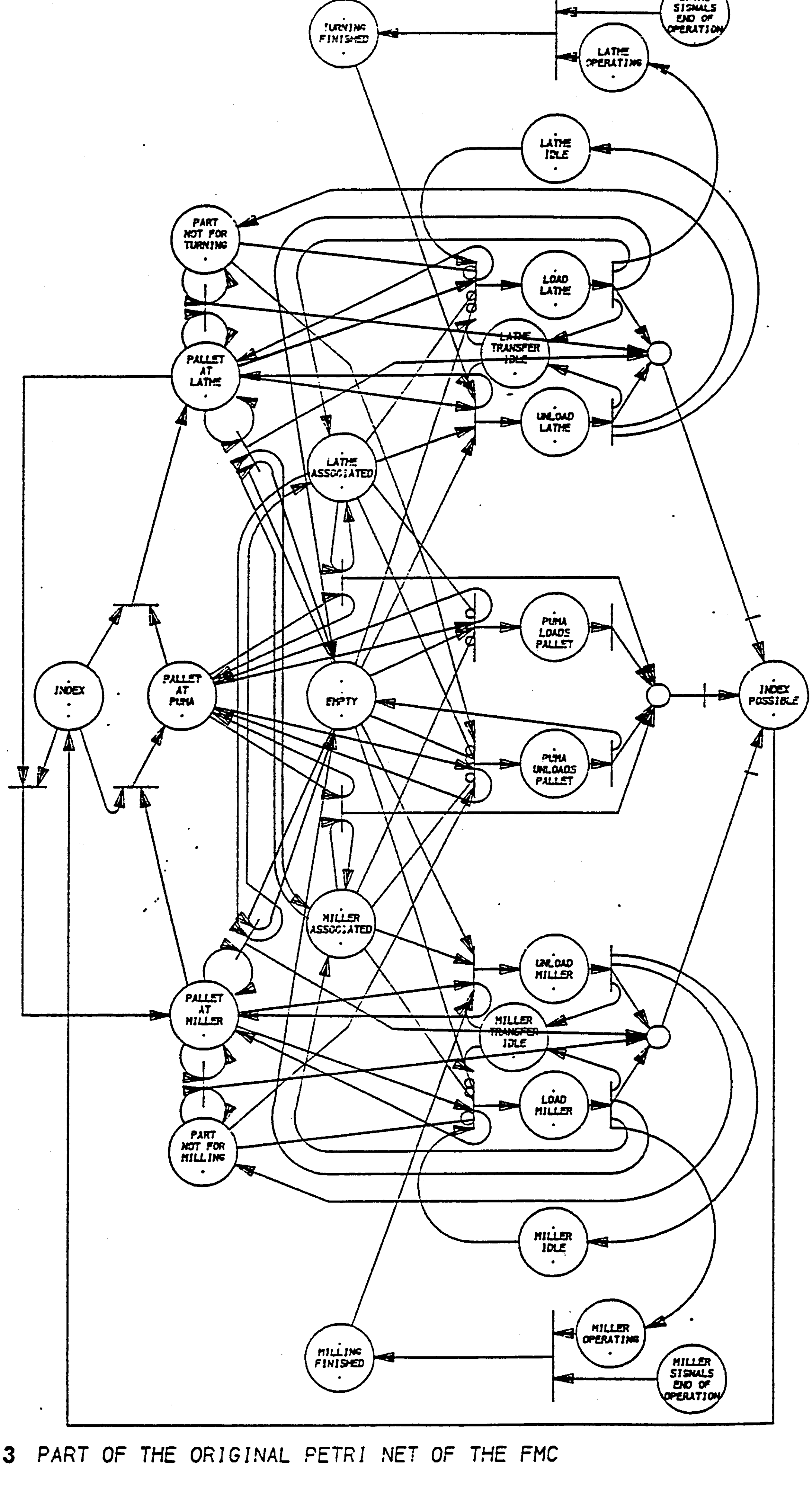


FIG. 3 PART OF THE ORIGINAL PETRI NET OF THE FMC

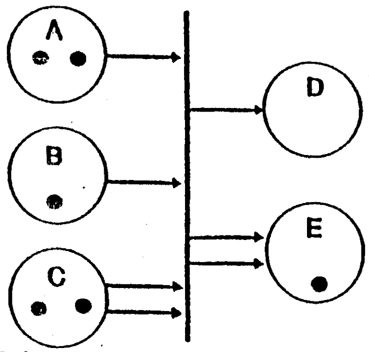


FIG. 4
AN EXAMPLE PETRI NET

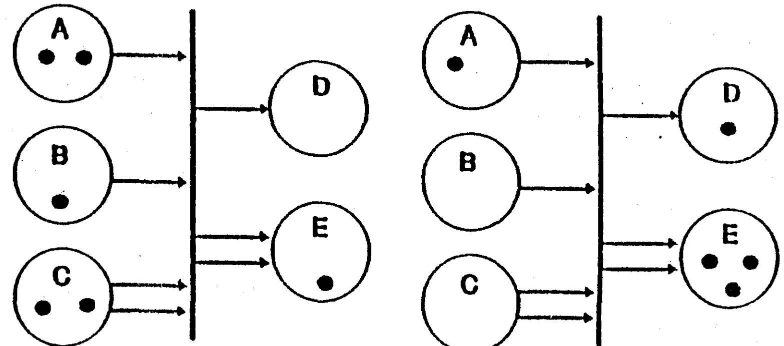
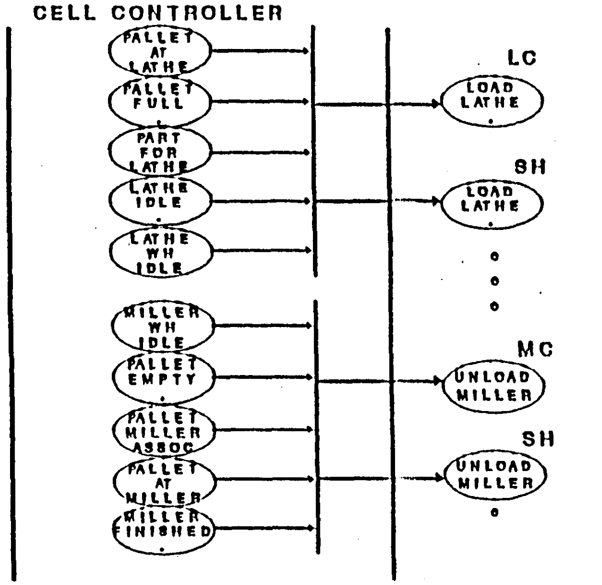
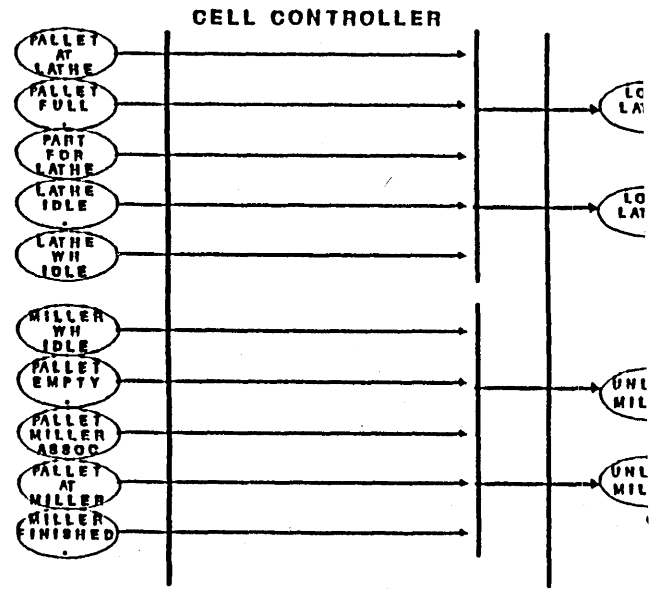


FIG. 5
AN EXAMPLE PETRI NET BEFORE AND AFTER FIRING



WH = WORK HANDLING, SH = TO STATUS HANDLER
LC & MC = TO LATHE & MILLER CONTROLLERS

FIG. 6 THE TRANSITIONS AND INPUT PLACES FOR TWO OF THE CELL CONTROLLER OUTPUT PLACES (STEP 2 OF THE METHODOLOGY)



WH = WORK HANDLING, SH = TO STATUS HAND
LC & MC = TO LATHE & MILLER CONTROLLERS

FIG. 7 THE INPUT PLACES TO THE TRANSITIONS IN FIGURE BECOME THE INPUT PLACES TO THE CELL CONTROLLER PETRI NET (STEP 2 OF THE METHODOLOGY)

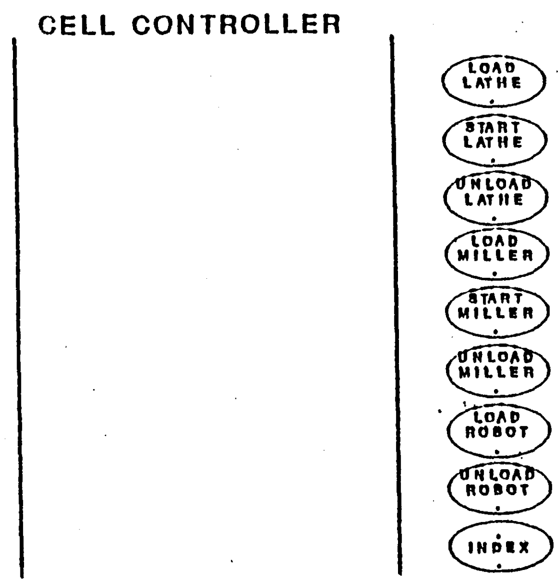


FIG. 8 THE OUTPUT PLACES AND BOUNDARIES OF THE CELL CONTROLLER PETRI NET

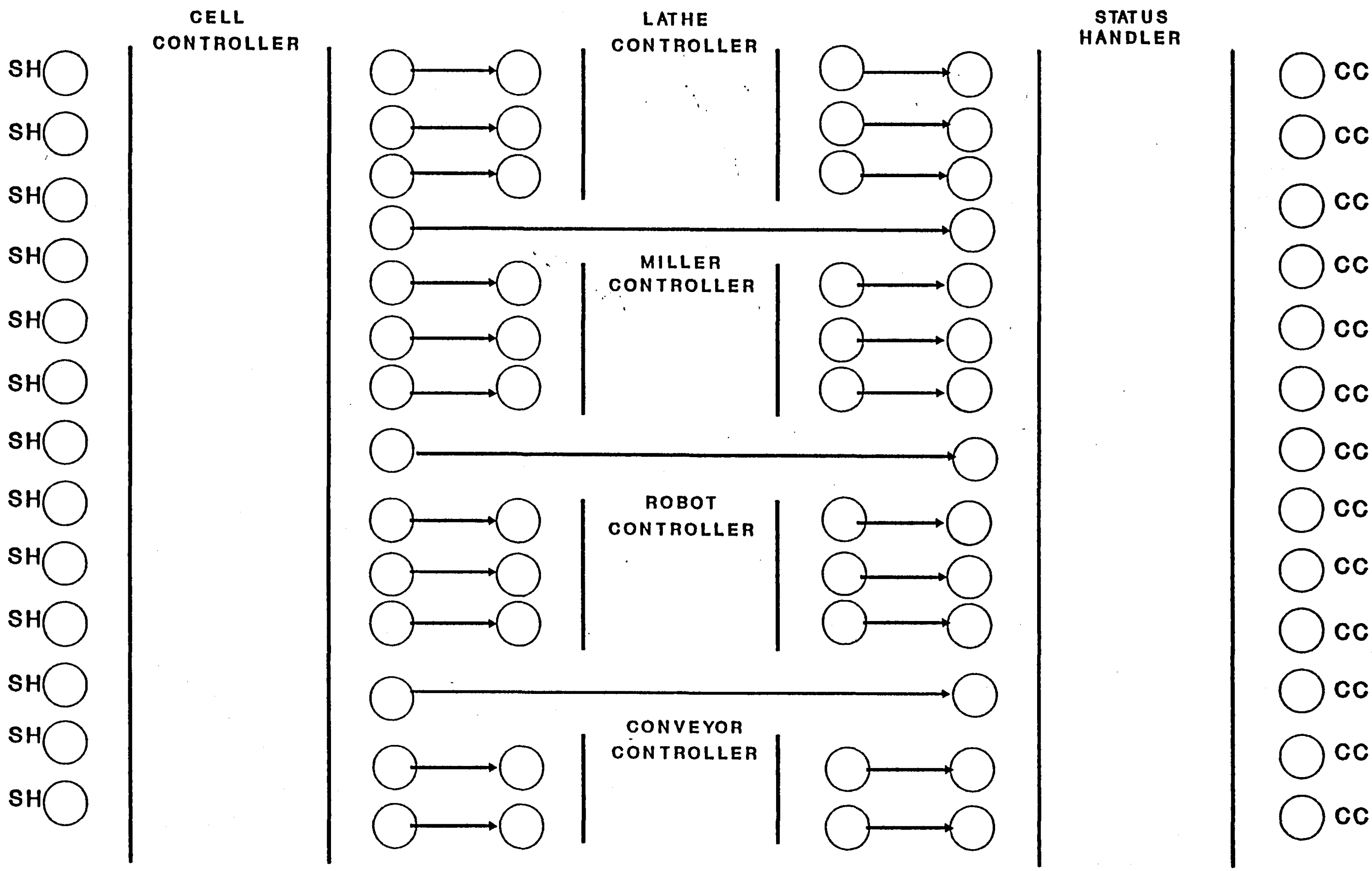


FIG 9 THE PETRI NET OF THE WHOLE CONTROL SYSTEM SHOWING ITS INHERENT PARALLELISM AND THE UNI-DIRECTIONAL STRUCTURE OF MESSAGES SENT BETWEEN INDIVIDUAL CONTROLLERS

SH = FROM STATUS HANDLER
CC = TO CELL CONTROLLER
DETAIL HAS BEEN OMITTED

CONTROL OF WORK HANDLING AND START OF MACHINING FOR MILLER STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD CITY POLYTECHNIC	Date: 25/2/93	Proj:MILLER
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 1

VICE TO TABLE (LOAD)

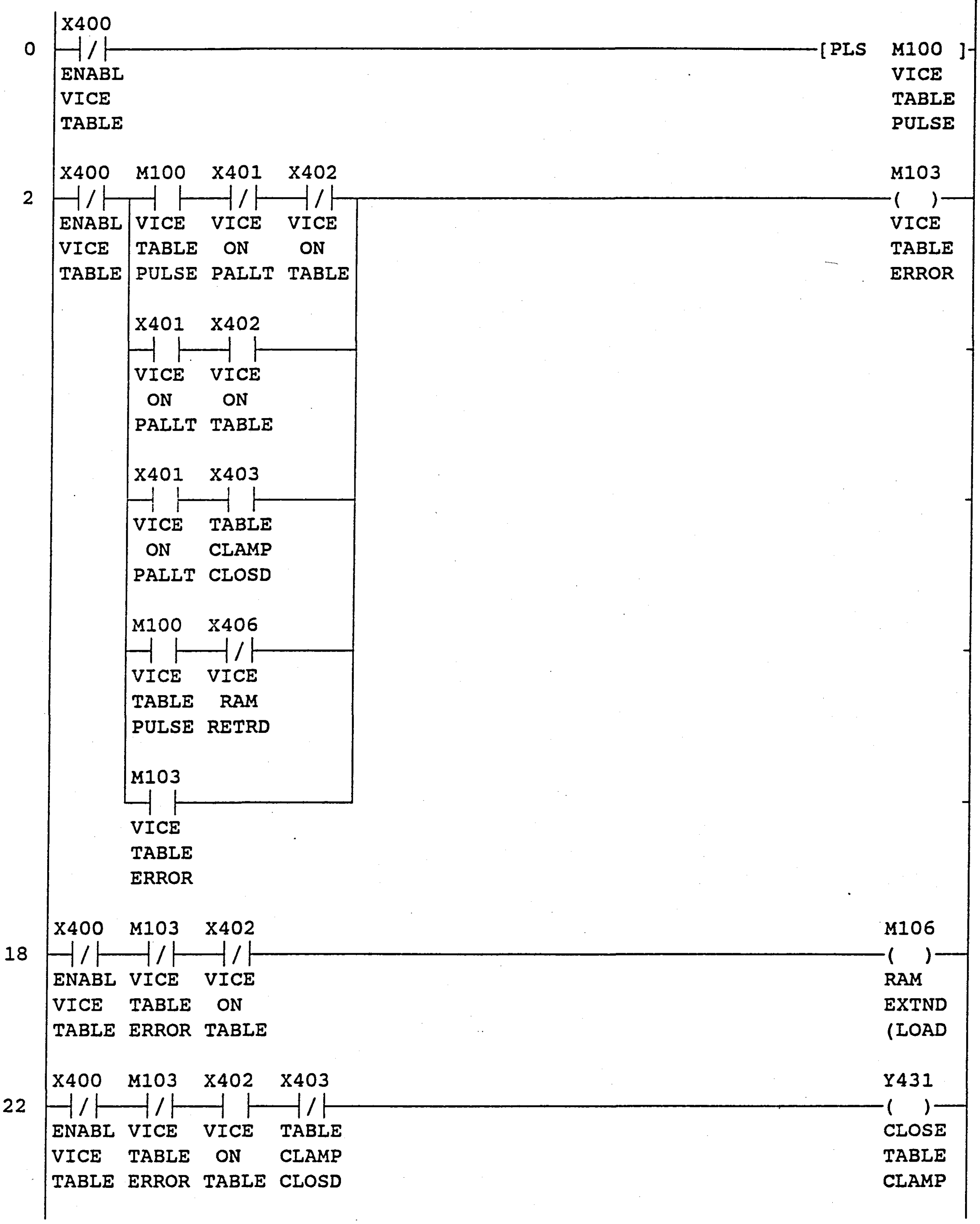


FIG. 10 AN EXAMPLE LADDER DIAGRAM

```
   |X403                                                           K2.0
27 |-| |---------------------------------------------------------(T50 )-|
   |TABLE                                                          TABLE
   |CLAMP                                                          CLAMP
   |CLOSD                                                          TIMER
   |
   |X400  M103  X402  T50                                          M110
30 |-|/|---|/|---| |---| |-------------------------------------------(  )-|
   |ENABL VICE  VICE  TABLE                                        RAM
   |VICE  TABLE  ON   CLAMP                                        RETR
   |TABLE ERROR TABLE TIMER                                        (LOAD
   |
   |X400  M103  X402  T50   X406                                   Y432
35 |-|/|---|/|---| |---| |---| |-------------------------------------(  )-|
   |ENABL VICE  VICE  TABLE VICE                                   VICE
   |VICE  TABLE  ON   CLAMP  RAM                                   TABLE
   |TABLE ERROR TABLE TIMER RETRD                                  COMPL
   |
```

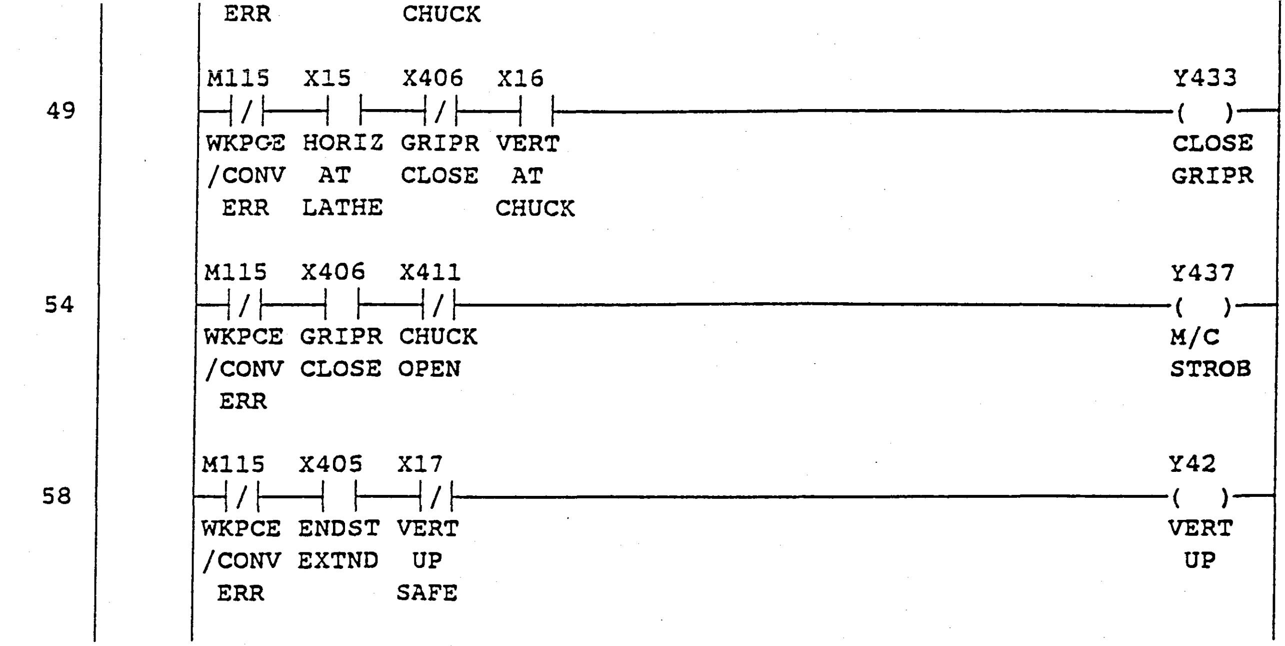


FIG. 11 AN EXAMPLE LADDER DIAGRAM CONTAINING OPPOSING INSTRUCTIONS WHERE THE INPUT CONDITIONS ARE INCOMPLETE. THE RESULT IS THE SOLENOID OSCILLATING DUE TO THE EXECUTION OF VERT UP DIRECTLY AFTER VERT DOWN AND VICE VERSA.

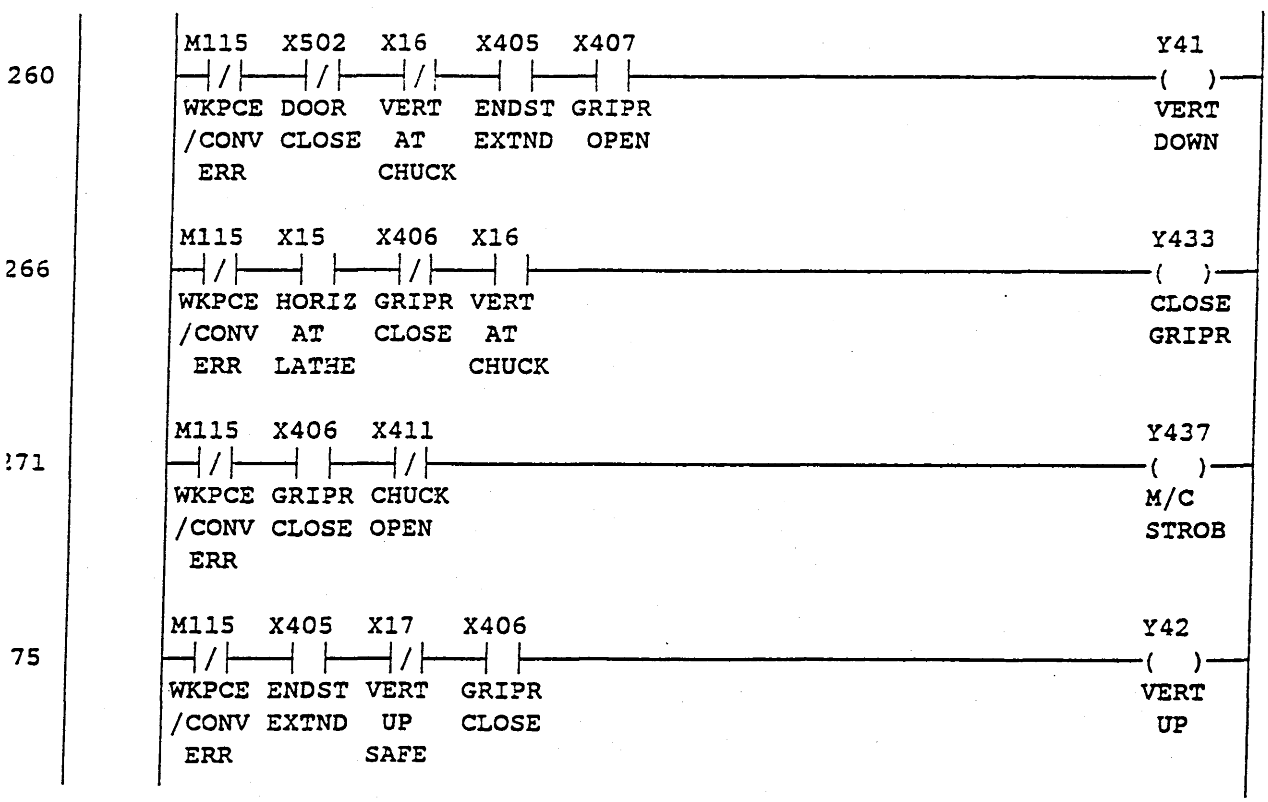


FIG. 11 THE ABOVE LADDER DIAGRAM WITH THE COMPLETE AND CORRECT INPUT CONDITIONS. VERT UP IS ONLY EXECUTED IF VERT DOWN, CLOSE GRIPR AND M/C STROB HAVE BEEN EXECUTED.

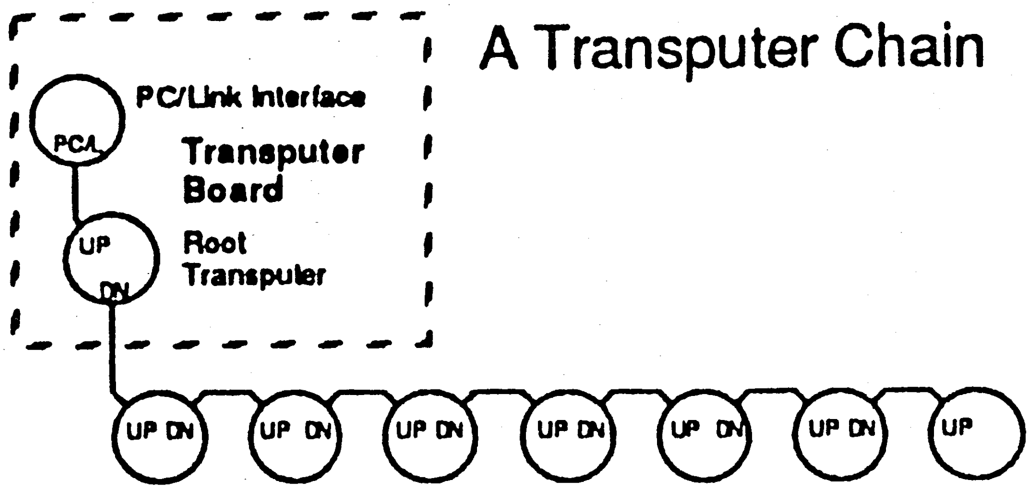


FIG.12 AN EXAMPLE LINEAR CHAIN TRANSPUTER NETWORK

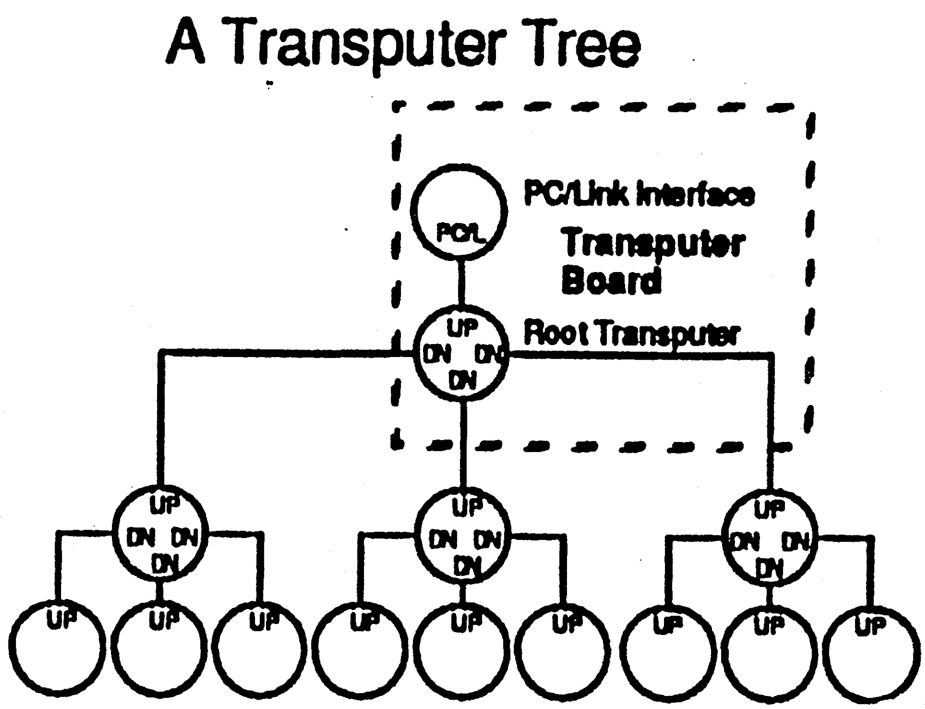


FIG.12 AN EXAMPLE TOROIDAL MESH TRANSPUTER NETWORK

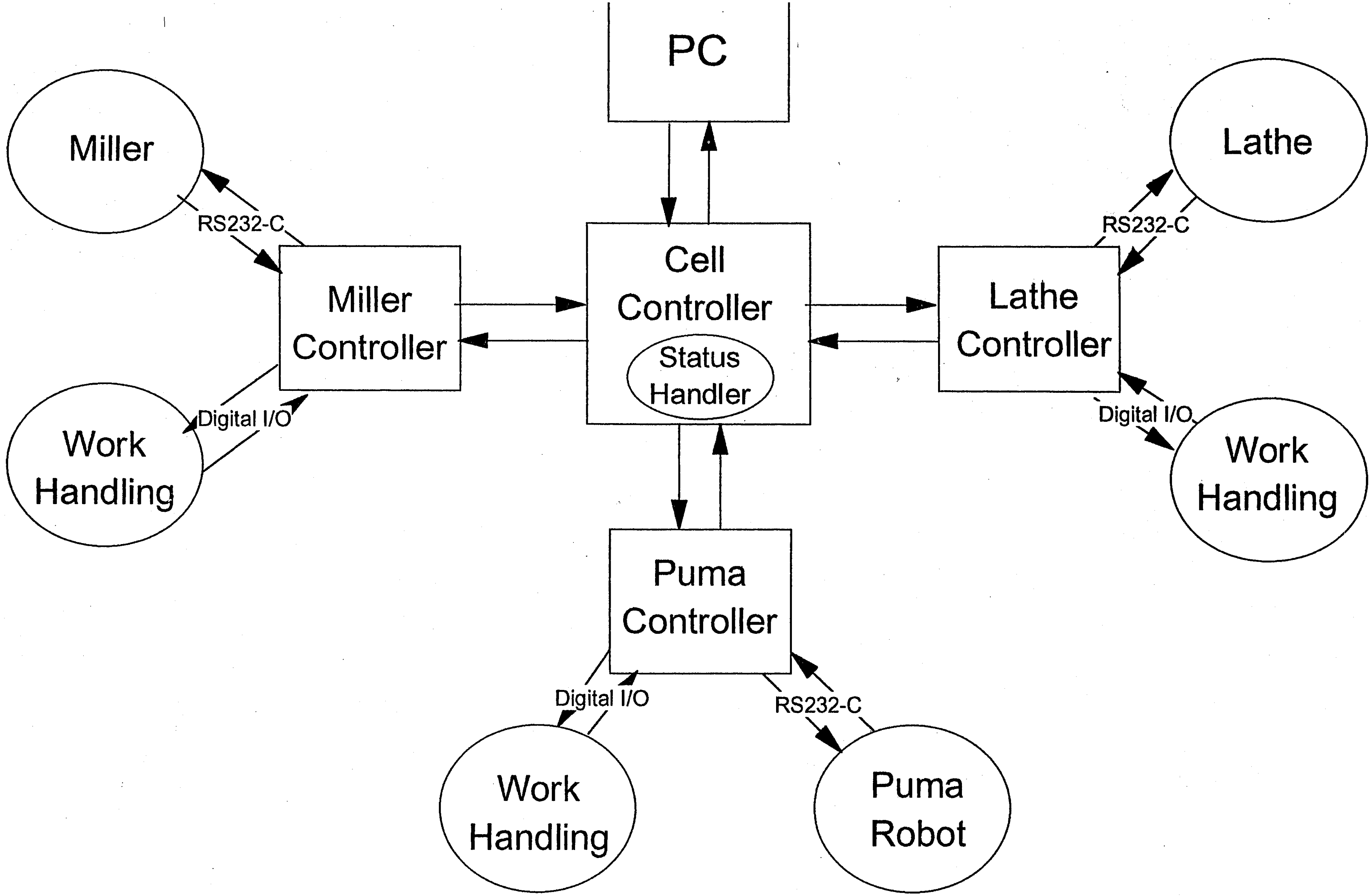


FIG.13 THE CONFIGURATION OF THE CONTROLLERS (TRANSPUTERS) AND THE COMMUNICATION PROTOCOL WITHIN THE SCHOOL'S FMC

8. List of Appendices

CONTROL OF WORK HANDLING AND CONVEYOR INDEXING FOR PUMA STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD HALLAM UNIVERSITY	Date: 22\3\93	Proj:PUMA
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Name
	Draw.no:	Sign: A.T.	Page: 1

I/O	Name	Comment	Remark
X400	ENABLVICE TABLE	Input from IBM-PC	Active Low
X401	ENABLOPEN VICE	Input from IBM-PC	Active Low
X402	ENABLCLOSEVICE	Input from IBM-PC	Active Low
X403	ENABLVICE PALLT	Input from IBM-PC	Active Low
X404	ENABLCONVRINDEX	Input from IBM-PC	Active Low
X405	ENABLPUMA CHECK	DEFERRED	
X406	ENABLCONVRCHECK	DEFERRED	
X407	VICE ON TABLE	Input from Sensor 1	Ind. Prox.
X410	VICE RAM RETRD	Input from Sensor 2	Magn. Reed
X411	VICE CLOSD	Input from Sensor 3	Press. Sw.
X412	VICE ON PALLT	Input from Sensor 4	IR Refl.
X413	PALLT AT PUMA	Input from Sensor 5	Cap. Prox.
X500	PALLT AT LATHE	Input from Sensor 6	Cap. Prox.
X501	PALLT AT MILLR	Input from Sensor 7	Cap. Prox.
Y430	PUMA STATNCOMPL	Output to IBM-PC	
Y431	PUMA STATNERROR	Output to IBM-PC	
Y432	CONVRINDEX COMP	Output to IBM-PC	
Y433	CONVRINDEXERROR	Output to IBM-PC	
Y530	OPEN VICE	Output to Vice Cyl. (Retract)	Double Sol
Y531	CLOSE VICE	Output to Vice Cyl. (Extend)	Double Sol
Y532	OPEN PUMA DOG	Output to Puma Stop Cyl.	Single Sol
Y533	OPEN LATHE DOG	Output to Lathe Stop Cyl.	Single Sol
Y534	OPEN MILLR DOG	Output to Miller Stop Cyl.	Single Sol
Y535	EXTNDVICE RAM	Output to Ram Cyl. (Extend)	Double Sol
Y536	RETRTVICE RAM	Output to Ram Cyl. (Retract)	Double Sol
M100	VICE TABLEPULSE		
M103	VICE PALLTPULSE		
M104	CONVRINDEXPULSE		
M107	VICE TABLEERROR		
M110	VICE TABLECOMPL		
M111	VICE PALLTERROR		
M112	VICE PALLTCOMPL		
M113	OPEN VICE ERROR		
M114	OPEN VICE COMPL		
M115	CLOSEVICE ERROR		
M116	CLOSEVICE COMPL		
T50	DOG OPEN TIMER		
T51	VICE OPEN TIMER		

CONTROL OF WORK HANDLING AND CONVEYOR INDEXING FOR PUMA STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD HALLAM UNIVERSITY	Date: 22\3\93	Proj:PUMA
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 1

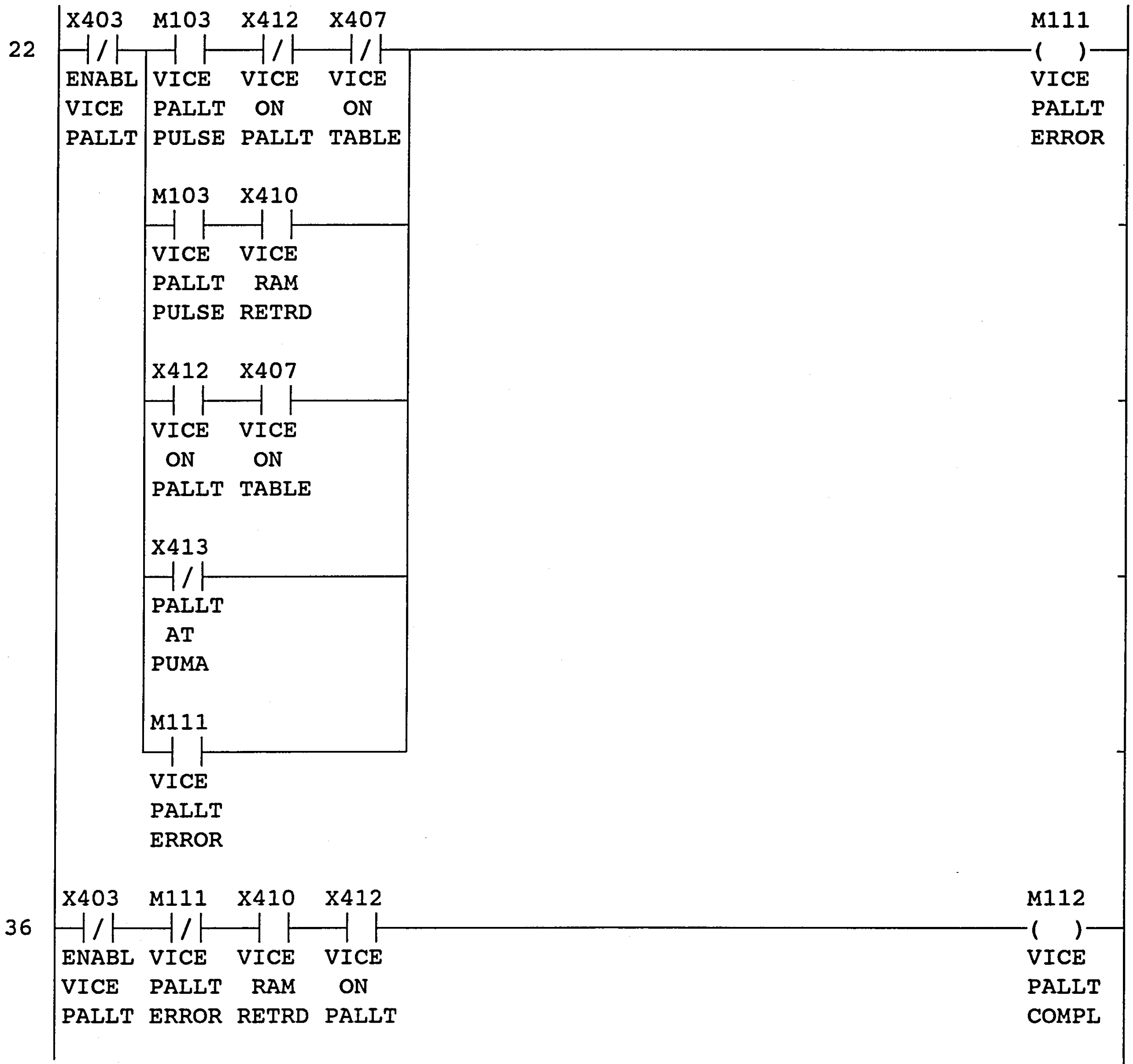
VICE TO TABLE

```
   X400
0 -|/|---------------------------------------------------[PLS  M100 ]-
   ENABL                                                       VICE
   VICE                                                        TABLE
   TABLE                                                       PULSE

   X400  M100  X412  X407                                      M107
2 -|/|---| |---|/|---|/|-+-------------------------------------(   )-
   ENABL|VICE  VICE  VICE |                                    VICE
   VICE |TABLE  ON    ON  |                                    TABLE
   TABLE|PULSE PALLT TABLE|                                    ERROR
        |                 |
        |X412  X407       |
        +-| |---| |-------+
        |VICE  VICE       |
        | ON    ON        |
        |PALLT TABLE      |
        |                 |
        |M107             |
        +-| |-------------+
         VICE
         TABLE
         ERROR

    X400  M107  X407                                           Y535
12 -|/|---|/|---|/|----------------------------------------------(   )-
    ENABL VICE  VICE                                           EXTND
    VICE  TABLE  ON                                            VICE
    TABLE ERROR TABLE                                           RAM

    X400  M107  X407                                           M110
16 -|/|---|/|---| |----------------------------------------------(   )-
    ENABL VICE  VICE                                           VICE
    VICE  TABLE  ON                                            TABLE
    TABLE ERROR TABLE                                          COMPL
```

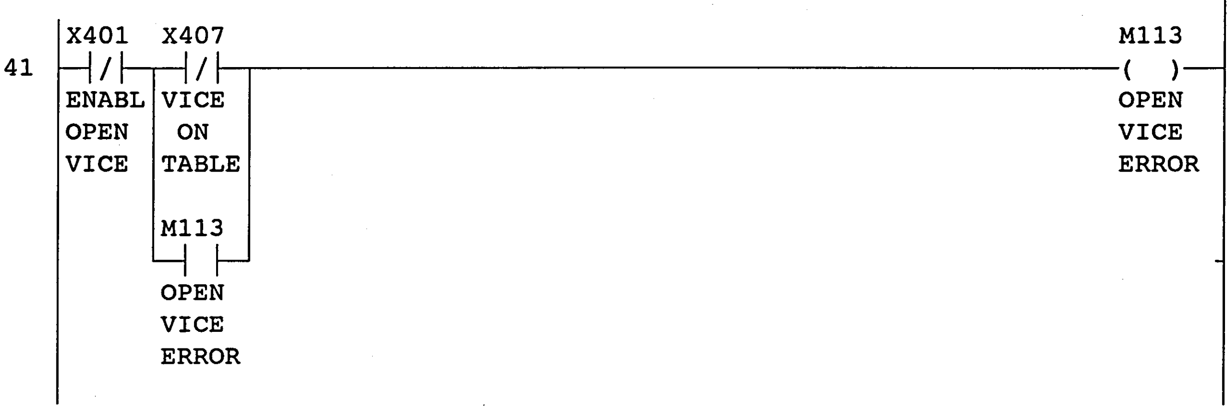
VICE TO PALLET

```
    X403
20 -|/|--------------------------------------------------[PLS  M103 ]-
    ENABL                                                      VICE
    VICE                                                       PALLT
    PALLT                                                      PULSE
```

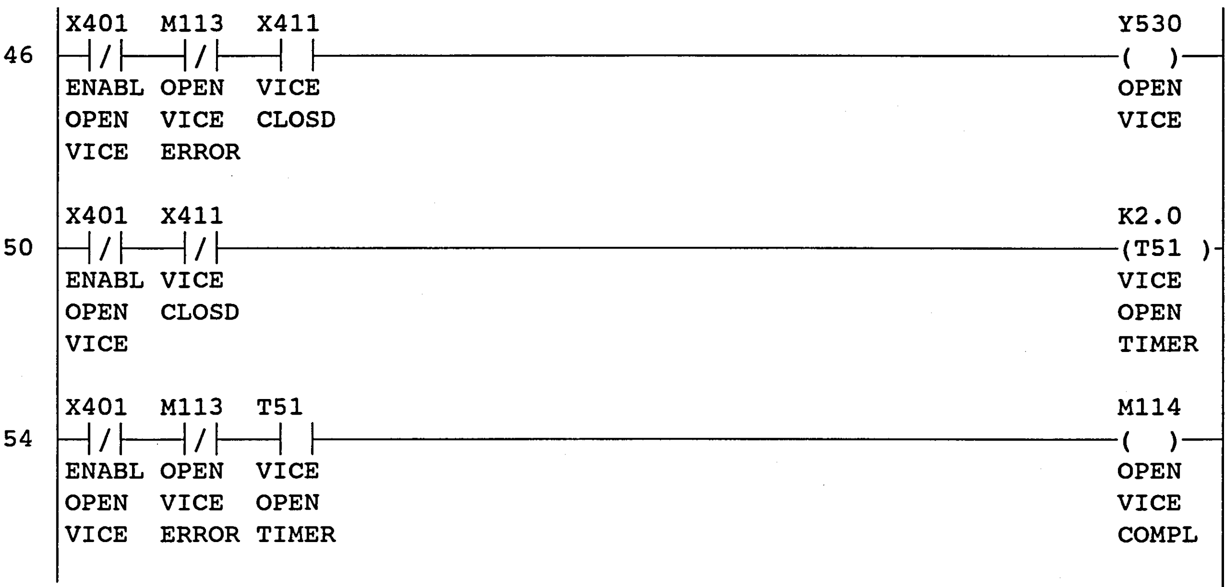
CONTROL OF WORK HANDLING AND CONVEYOR INDEXING FOR PUMA STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD HALLAM UNIVERSITY	Date: 22\3\93	Proj:PUMA
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 2



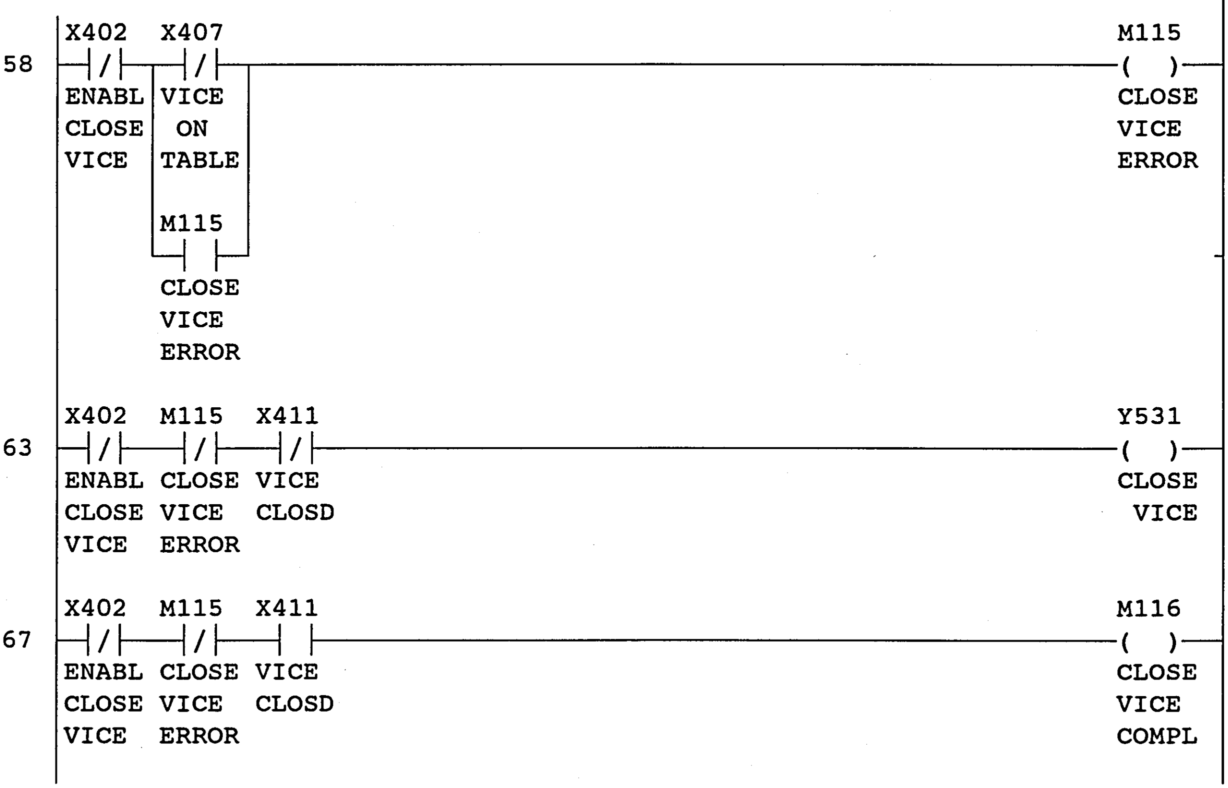
OPEN VICE



CONTROL OF WORK HANDLING AND CONVEYOR INDEXING FOR PUMA STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD HALLAM UNIVERSITY	Date: 22\3\93	Proj:PUMA
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 3



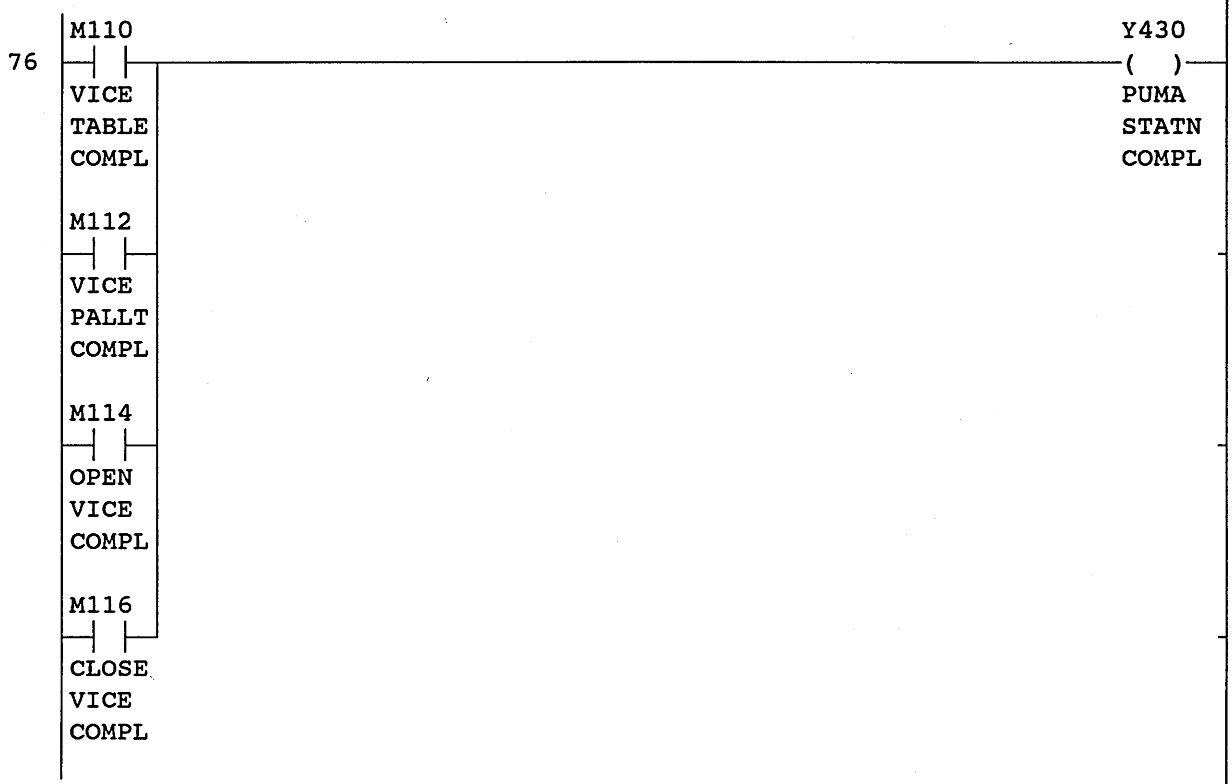
CLOSE VICE



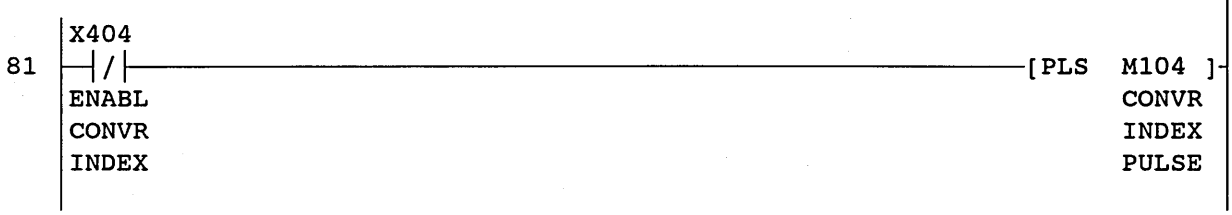
GENERATE ERROR STATUS

CONTROL OF WORK HANDLING AND CONVEYOR INDEXING FOR PUMA STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD HALLAM UNIVERSITY	Date: 22\3\93	Proj:PUMA
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 5

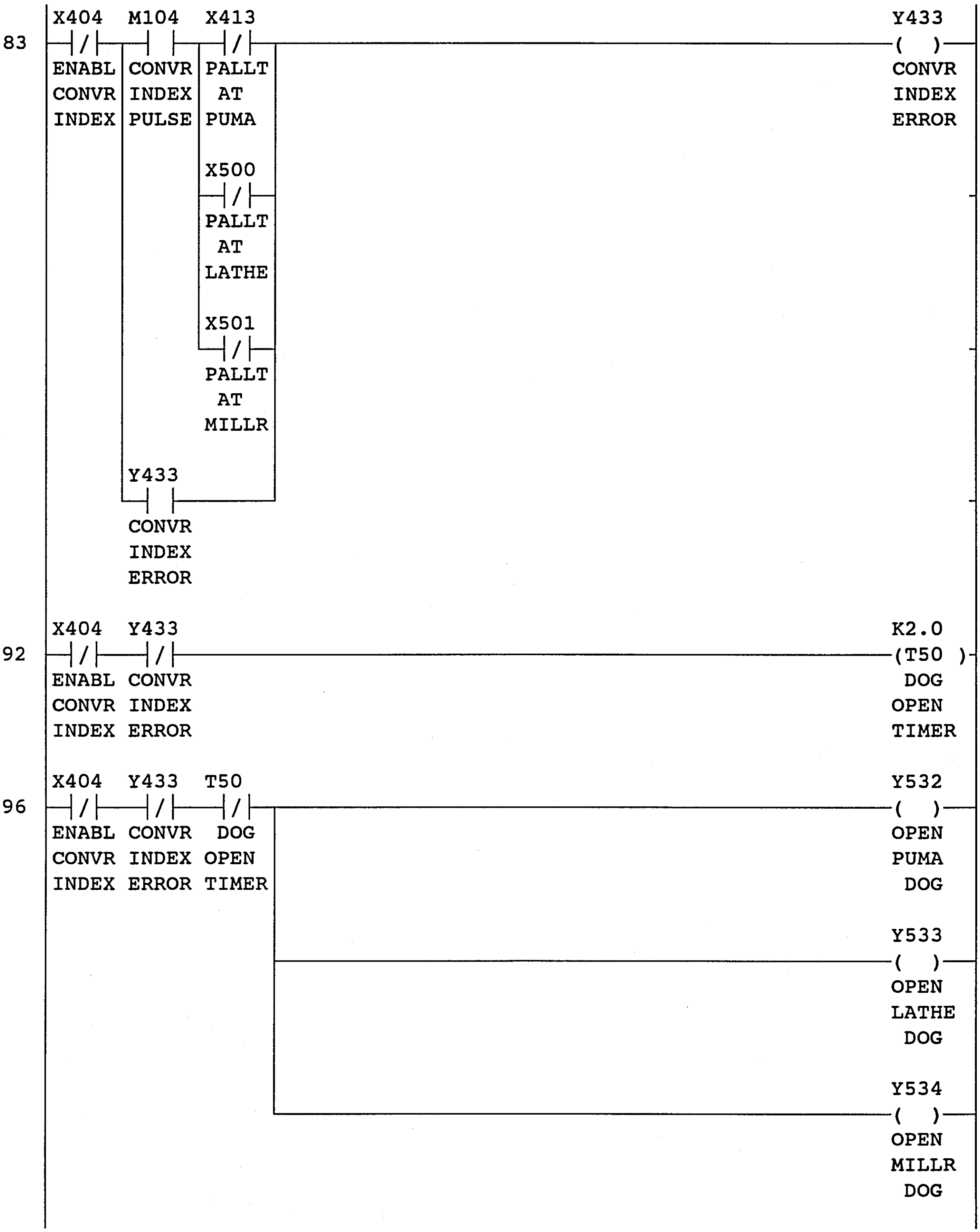
GENERATE COMPLETION STATUS



CONVEYOR INDEX



CONTROL OF WORK HANDLING AND CONVEYOR INDEXING FOR PUMA STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD HALLAM UNIVERSITY	Date: 22\3\93	Proj:PUMA
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 6



CONTROL OF WORK HANDLING AND CONVEYOR INDEXING FOR PUMA STATION IN SCHOOL OF ENGINEERG FMC	SHEFFIELD HALLAM UNIVERSITY	Date: 22\3\93	Proj:PUMA
		Rev.dat:	Syst:F1/F2
		Rev.no: 3	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 7

```
    X404  Y433  Y532  X413  X500  X501                            Y432
.02 -|/|---|/|---|/|---| |---| |---| |----------------------------( )-
    ENABL CONVR OPEN  PALLT PALLT PALLT                           CONVR
    CONVR INDEX PUMA   AT    AT    AT                             INDEX
    INDEX ERROR  DOG  PUMA  LATHE MILLR                            COMP

.09 --------------------------------------------------------------[END ]-
```

CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Name
	Draw.no:	Sign: A.T.	Page: 1

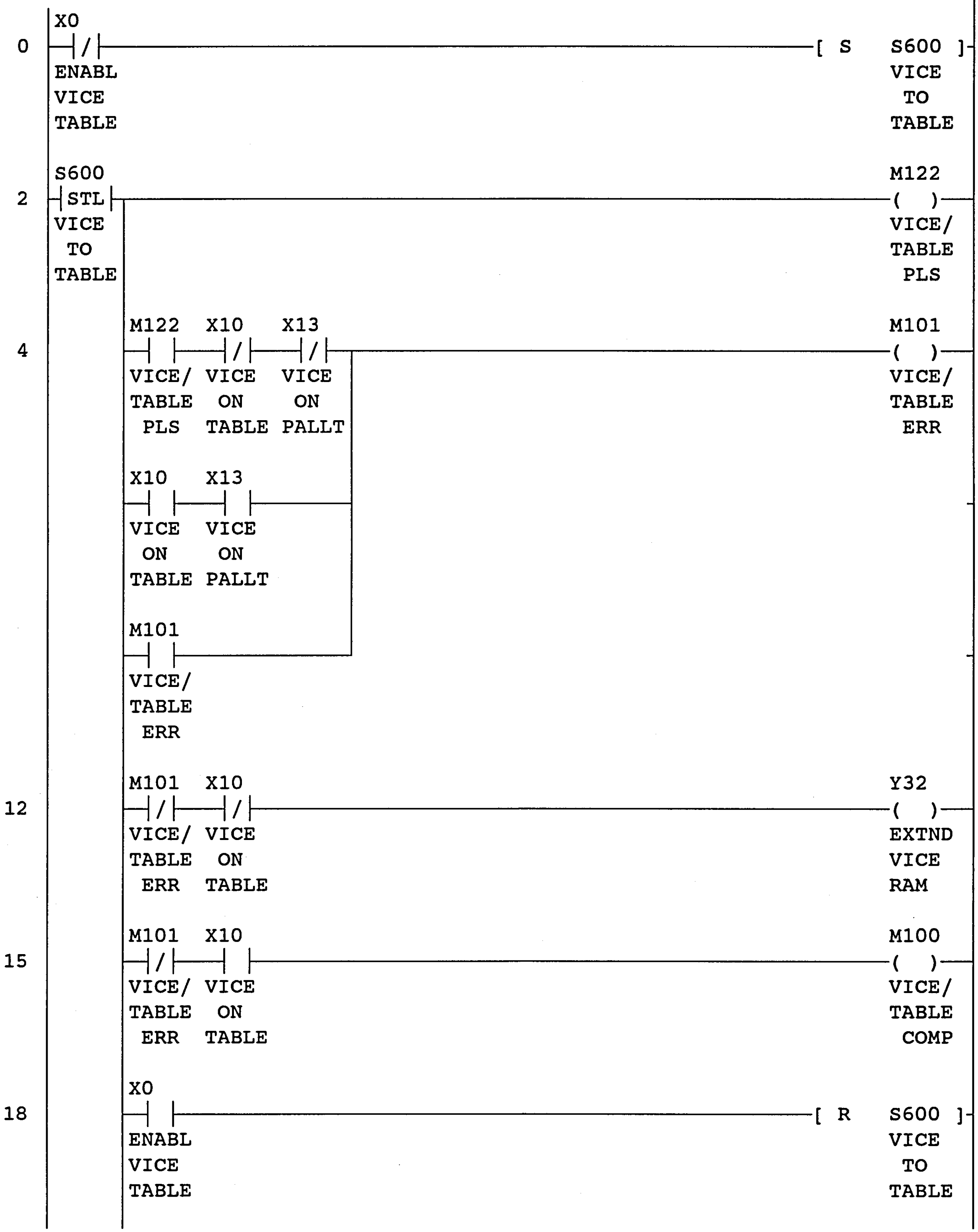
I/O	Name	Comment	Remark
X0	ENABLVICE TABLE	Input from IBM-PC	Active Low
X1	ENABLGRIP WKPCE	Input from IBM-PC	Active Low
X2	ENABLWKPCE CHUK	Input from IBM-PC	Active Low
X3	ENABLROBOT SAFE	Input from IBM-PC	Active Low
X4	ENABLVICE PALLT	Input from IBM-PC	Active Low
X5	ENABLSTARTM/C'G	Input from IBM-PC	Active Low
X6	ENABLWKPCE CONV	Input from IBM-PC	Active Low
X7	ENABLWKPCE VICE	Input from IBM-PC	Active Low
X10	VICE ON TABLE	Input from sensor 12	Ind. Prox.
X11	VICE RAM RETRD	Input from sensor 13	Magn. Reed
X12	VICE CLOSE	Input from sensor 14	Press. Sw.
X13	VICE ON PALLT	Input from sensor 15	I.R. Refl.
X14	HORIZ AT CONVR	Input from sensor B01	Ind. Prox.
X15	HORIZ AT LATHE	Input from sensor B02	Ind. Prox.
X16	VERT AT CHUCK	Input from sensor B03	Ind. Prox.
X17	VERT UP SAFE	Input from sensor B04	Ind. Prox.
X400	MIDSTVERT EXT	Input from sensor B05	Press. Sw.
X401	MIDSTVERT RET	Input from sensor B06	Magn. Reed
X402	ROTAT 0 DEG	Input from sensor B07	Ind. Prox.
X403	ROTAT 90 DEG	Input from sensor B09	Ind. Prox.
X404	ENDSTRETRT	Input from sensor B11	Magn. Reed
X405	ENDSTEXTND	Input from sensor B12	Magn. Reed
X406	GRIPRCLOSE	Input from sensor B13	Magn. Reed
X407	GRIPR OPEN	Input from sensor B14	Press. Sw.
X410	GRIPR AT VICE	Input from sensor B17	Ind. Prox.
X411	CHUCKOPEN	CNC M/C CODE 61	Int. Relay
X412	CHUCKCLOSE	CNC M/C CODE 62	Int. Relay
X413	M/C'GFINSH	CNC M/C CODE 63	Int. Relay
X500	IN CYCLE	CNC Cycle Interrupt	Int. Relay
X501	ERROR	CNC Cycle Interrupt	Int. Relay
X502	DOOR CLOSE	Input from sensor 16	
Y30	LATHESTATNCOMP	Output to IBM-PC	
Y31	LATHESTATNEROR	Output to IBM-PC	
Y32	EXTNDVICE RAM	Output to Ram cyl.(Extend)	Double Sol
Y33	RETRTVICE RAM	Output to Ram cyl.(Retract)	Double Sol
Y34	OPEN VICE	Output to Vice cyl.(Retract)	Double Sol
Y35	CLOSE VICE	Output to Vice cyl.(Extend)	Double Sol
Y36	AIR ON	Output to Tog clamp	1Sol Sprng
Y37	HORIZ TO CONVR	Output to Sol. Y1	3 Pos 2Sol
Y40	HORIZ TO LATHE	Output to Sol. Y2	3 Pos 2Sol
Y41	VERT DOWN	Output to Sol. Y3	3 Pos 2Sol
Y42	VERT UP	Output to Sol. Y4	3 Pos 2Sol
Y43	EXT. VERT MIDST	Output to Sol. Y5	Double Sol
Y44	RET. VERT MIDST	Output to Sol. Y6	Double Sol
Y45	ROTATTO 0 DEG	Output to Sol. Y7	1Sol Sprng
Y430	ROTATTO 90DEG	Output to Sol. Y9	1Sol Sprng
Y431	EXTND END STOP	Output to Sol. Y10	Double Sol

CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Name
	Draw.no:	Sign: A.T.	Page: 2

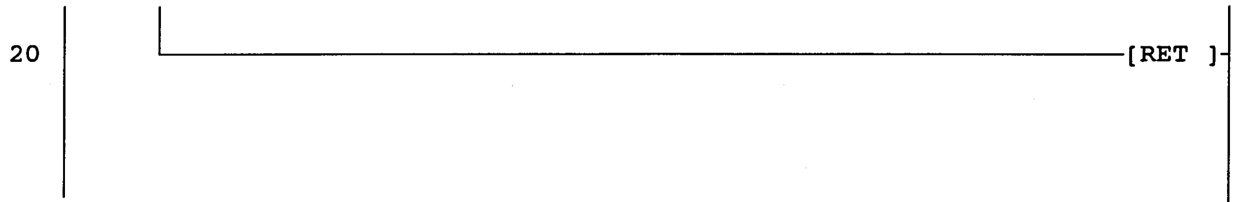
I/O	Name	Comment	Remark
Y433	CLOSEGRIPR	Output to Sol. Y12	Double Sol
Y434	OPEN GRIPR	Output to Sol. Y13	Double Sol
Y435	CYCLESTART	Output to Lathe	Int. Relay
Y436	CYCLESTOP	Output to Lathe	Int. Relay
Y437	M/C STROB	Output to Lathe	Int. Relay
Y530	OPEN DOOR	Output to Lathe	Int. Relay
Y531	CLOSEDOOR	Output to Lathe	Int. Relay
M100	VICE/TABLE COMP		
M101	VICE/TABLE ERR		
M102	GRIP WKPCE COMP		
M103	GRIP WKPCE ERR		
M104	WKPCE/CHUK COMP		
M105	WKPCE/CHUK ERR		
M106	ROBOT/SAFE COMP		
M107	ROBOT/SAFE ERR		
M110	VICE/PALLTCOMP		
M111	VICE/PALLT ERR		
M112	STARTM/C'GCOMP		
M113	STARTM/C'G ERR		
M114	WKPCE/CONV COMP		
M115	WKPCE/CONV ERR		
M116	WKPCE/VICE COMP		
M117	WKPCE/VICE ERR		
M121	M/C'GCOMPL		
M122	VICE/TABLE PLS		
M123	GRIP WKPCE PLS		
M124	WKPCECHUCK PLS		
M125	ROBOT SAFE PLS		
M126	VICE PALLT PLS		
M127	STARTM/C'G PLS		
M130	WKPCECONVR PLS		
M131	WKPCEVICE PLS		
S600	VICE TO TABLE		
S601	GRIP WORK PIECE		
S602	WKPCE TO CHUCK		
S603	ROBOT SAFE POSN		
S604	VICE TO PALLT		
S605	STARTM/C'G		
S606	WKPCE TO CONVR		
S607	WKPCE TO VICE		

CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 1

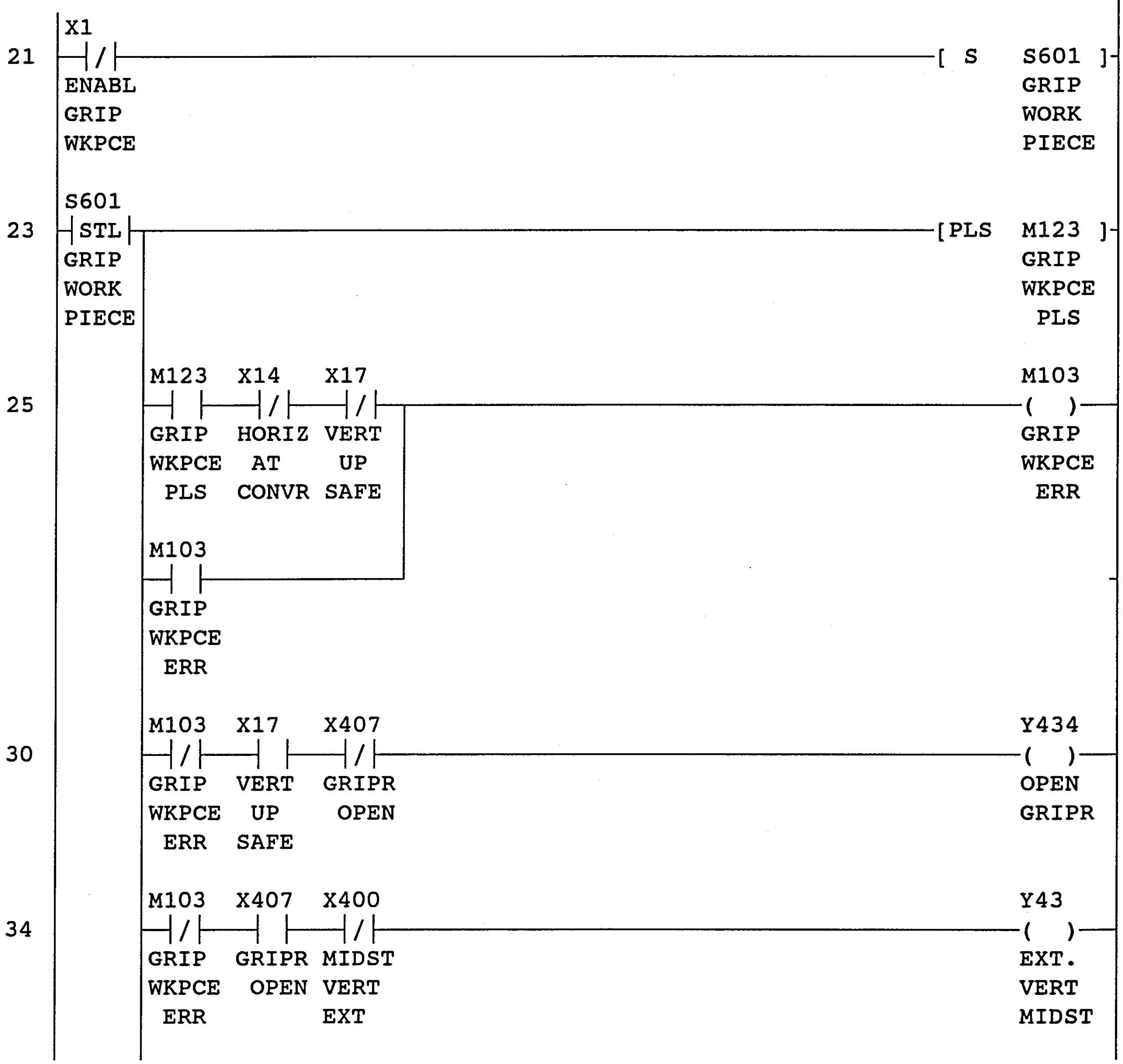
VICE TO TABLE



CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 2



GRIP WORKPIECE



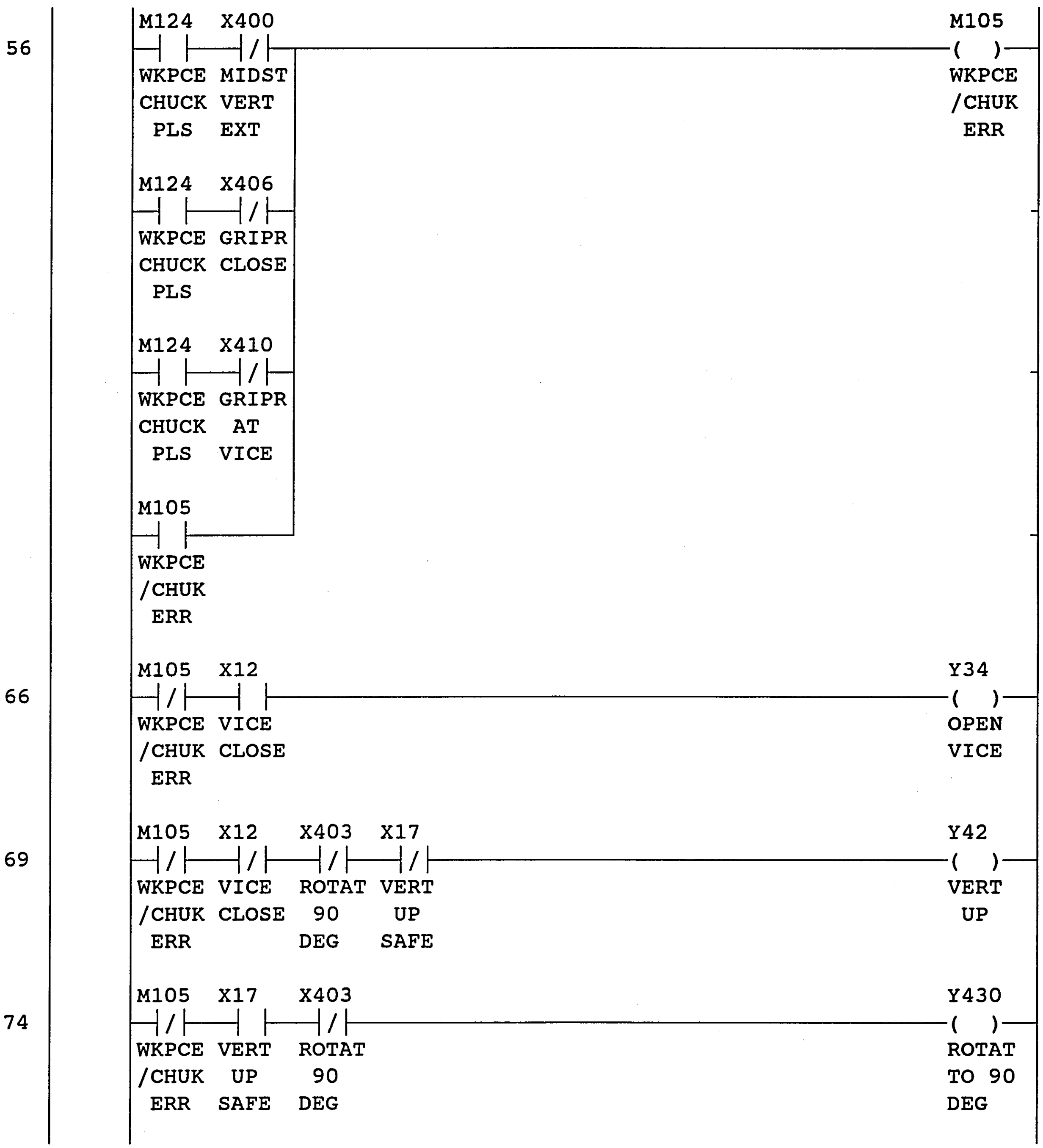
CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 3

```
          M103  X400  X410                                            Y41
38      --|/|---| |---|/|---------------------------------------------( )-
          GRIP  MIDST GRIPR                                           VERT
          WKPCE VERT   AT                                             DOWN
           ERR  EXT   VICE

          M103  X410  X406                                            Y433
42      --|/|---| |---|/|---------------------------------------------( )-
          GRIP  GRIPR GRIPR                                           CLOSE
          WKPCE  AT   CLOSE                                           GRIPR
           ERR  VICE

          M103  X406                                                  M102
46      --|/|---| |---------------------------------------------------( )-
          GRIP  GRIPR                                                 GRIP
          WKPCE CLOSE                                                 WKPCE
           ERR                                                         COMP

          X1
49      --| |-------------------------------------------------[ R     S601 ]-
          ENABL                                                       GRIP
          GRIP                                                        WORK
          WKPCE                                                       PIECE

51      --------------------------------------------------------------[RET ]-
```

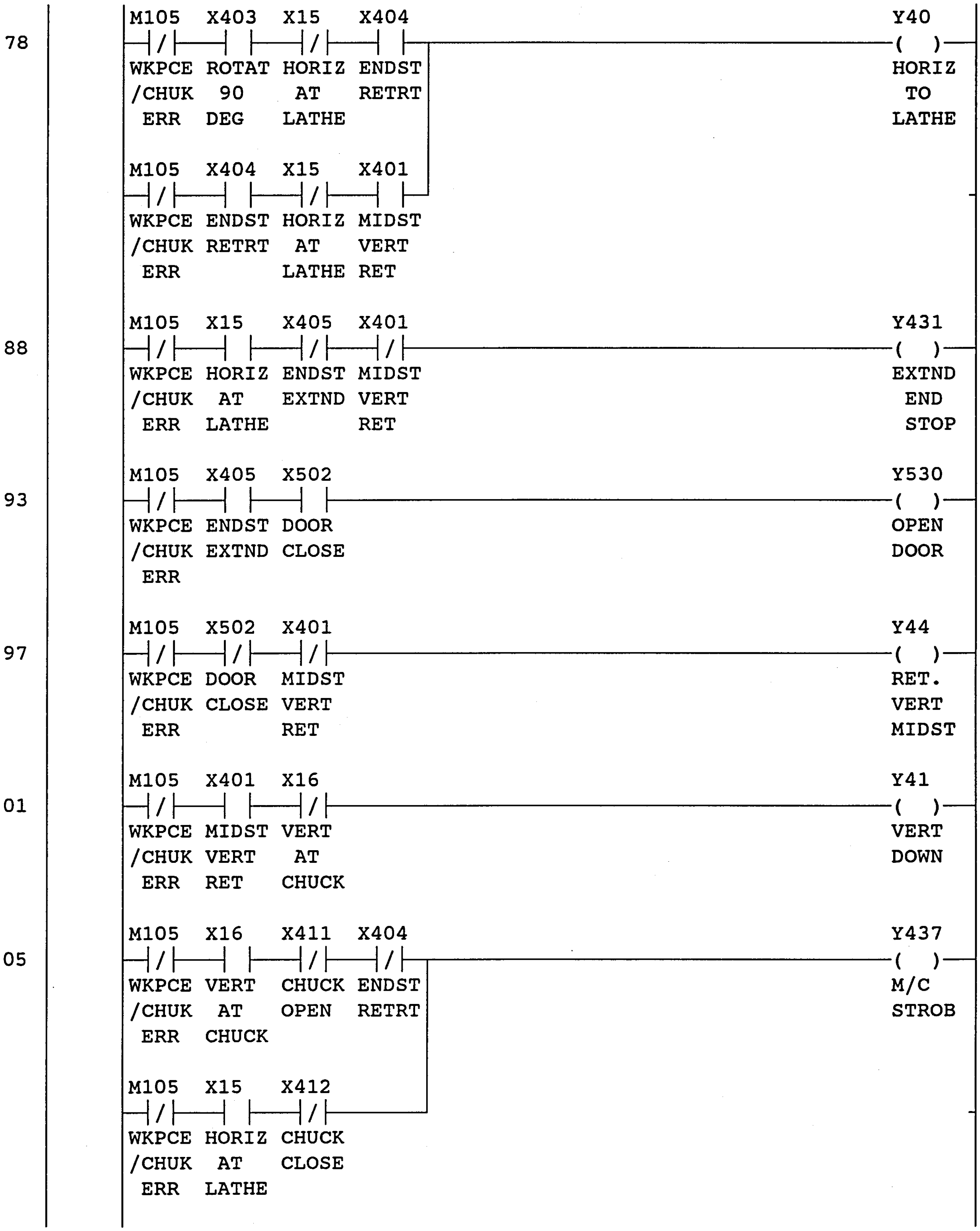
WORKPIECE TO CHUCK

```
    X2
52  --|/|---------------------------------------------------------[ S     S602 ]-
    ENABL                                                               WKPCE
    WKPCE                                                                 TO
     CHUK                                                               CHUCK

    S602
54  --|STL|-------------------------------------------------------[PLS   M124 ]-
    WKPCE                                                               WKPCE
      TO                                                                CHUCK
    CHUCK                                                                PLS
```

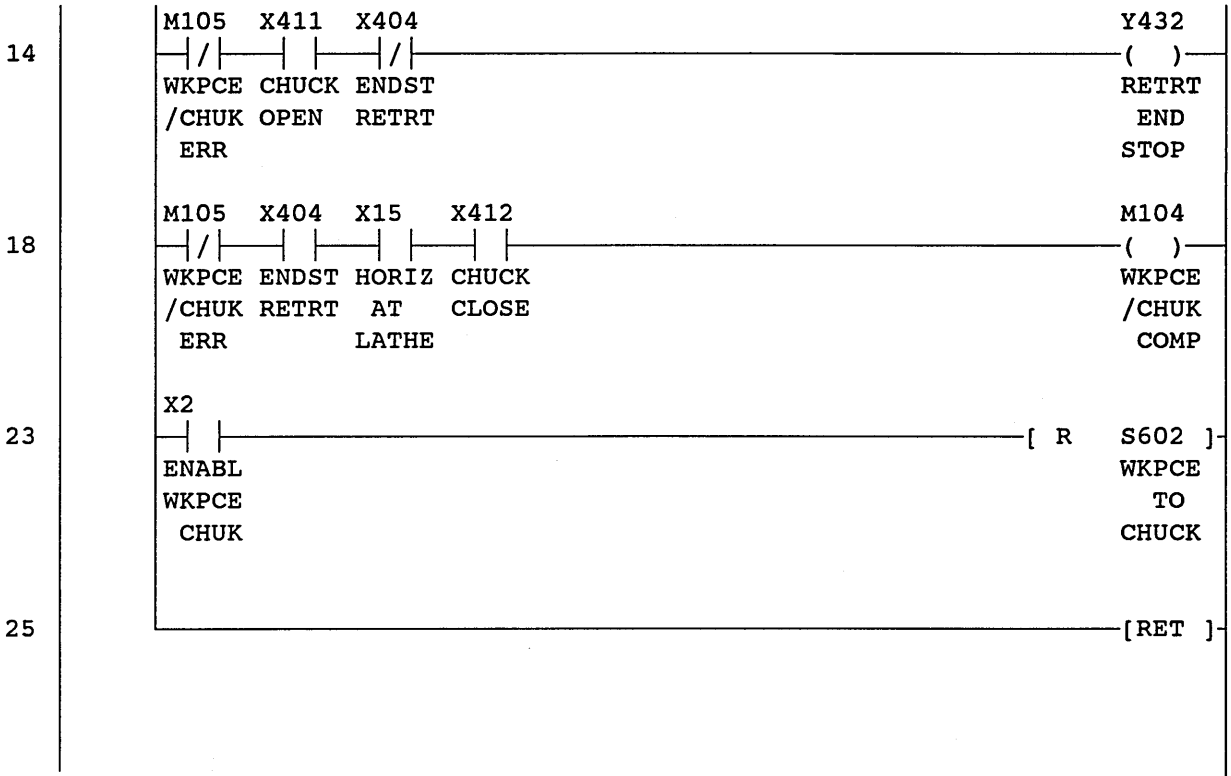
CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 4



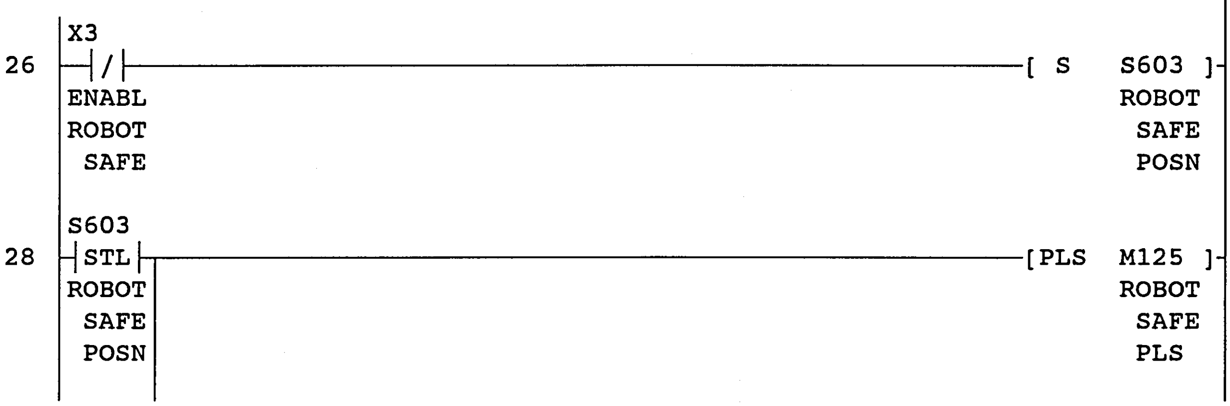
CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 5



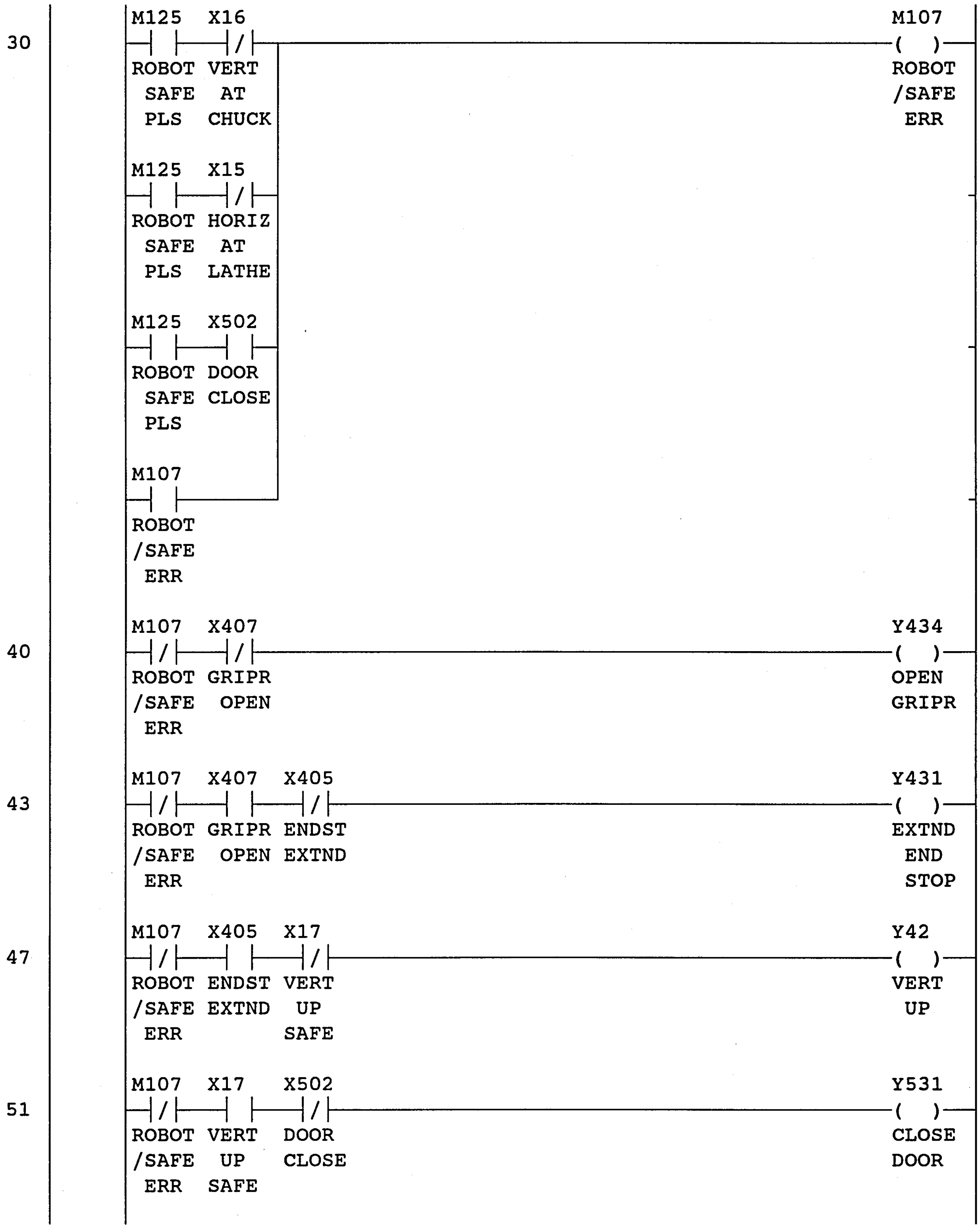
CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 6



ROBOT TO SAFE POSITION



CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 7



CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 8

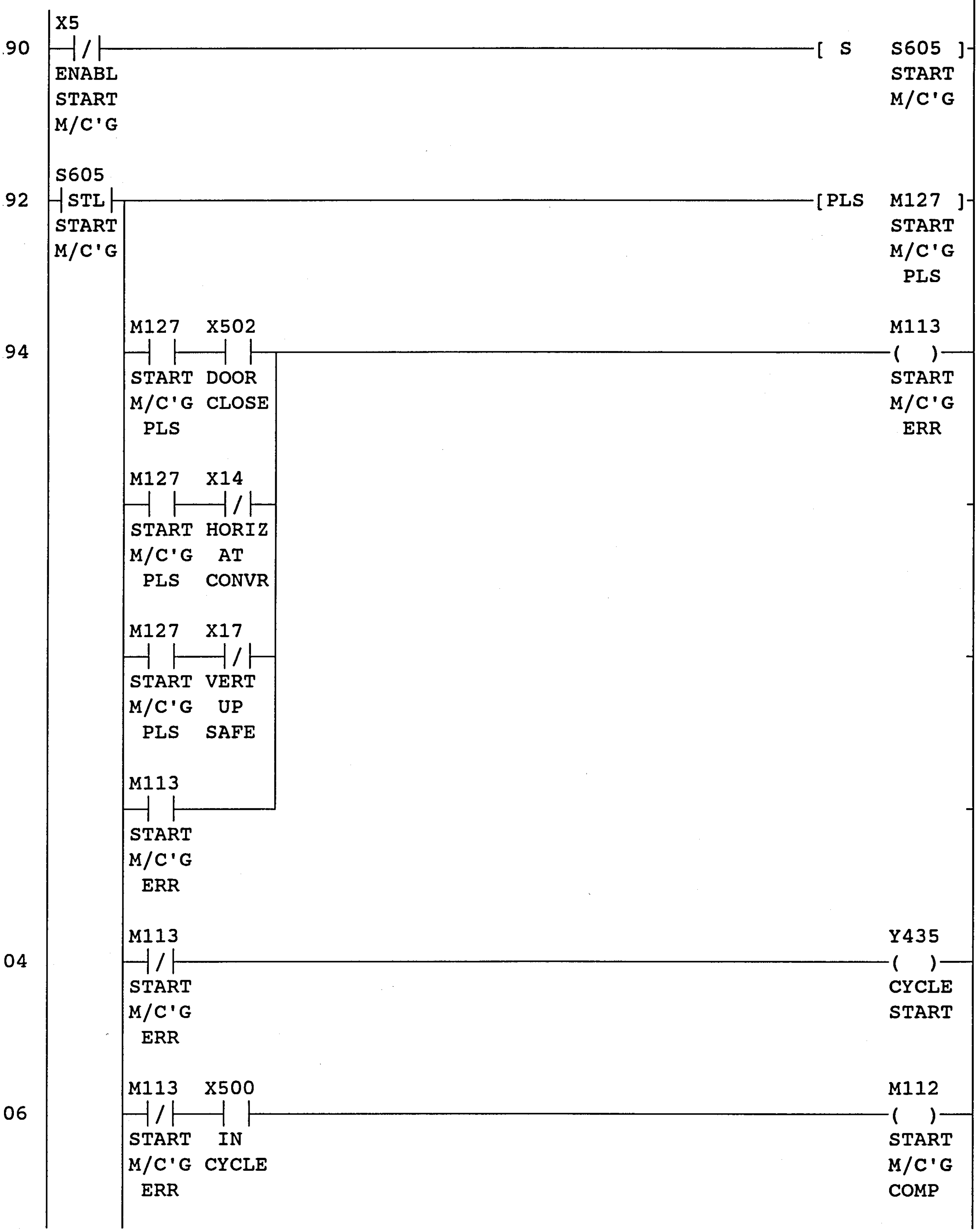
```
          M107  X502  X14                                                   Y37
55        -|/|---| |---|/|--------------------------------------------------(   )-
          ROBOT DOOR  HORIZ                                                 HORIZ
          /SAFE CLOSE  AT                                                    TO
           ERR        CONVR                                                 CONVR

          M107  X14                                                         M106
59        -|/|---| |--------------------------------------------------------(   )-
          ROBOT HORIZ                                                       ROBOT
          /SAFE  AT                                                         /SAFE
           ERR  CONVR                                                        COMP

          X3
62        -| |---------------------------------------------------[ R       S603 ]-
          ENABL                                                             ROBOT
          ROBOT                                                              SAFE
           SAFE                                                              POSN

64        ----------------------------------------------------------------[RET  ]-

 VICE TO PALLET

     X4
65   -|/|--------------------------------------------------------[ S       S604 ]-
     ENABL                                                                  VICE
     VICE                                                                    TO
     PALLT                                                                  PALLT

     S604
67   -|STL|------------------------------------------------------[PLS     M126 ]-
     VICE                                                                   VICE
      TO                                                                    PALLT
     PALLT                                                                   PLS
```

CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 9

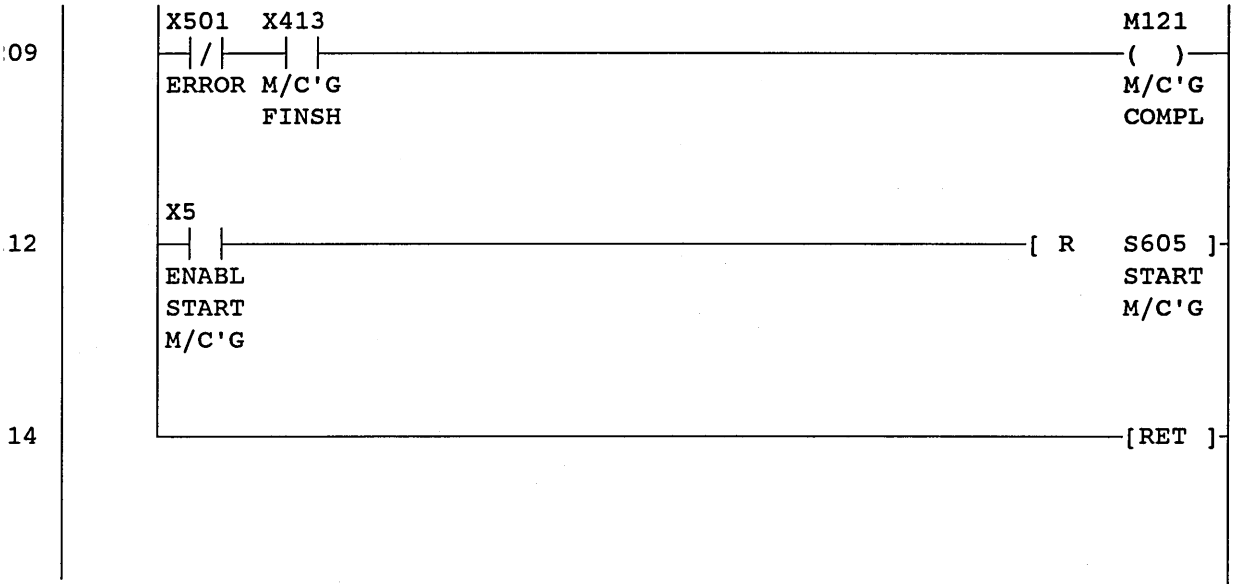
```
          M126   X13    X10                                              M111
 69     |--| |---|/|----|/|--+-------------------------------------------( )--|
        | VICE   VICE   VICE |                                           VICE/
        | PALLT   ON     ON  |                                           PALLT
        |  PLS   PALLT TABLE |                                            ERR
        |                    |
        | M126   X11         |
        |--| |---| |---------+
        | VICE   VICE        |
        | PALLT  RAM         |
        |  PLS   RETRD       |
        |                    |
        | X13    X10         |
        |--| |---| |---------+
        | VICE   VICE        |
        |  ON     ON         |
        | PALLT  TABLE       |
        |                    |
        | M111               |
        |--| |---------------+
        | VICE/
        | PALLT
        |  ERR
        |
        | M111   X13                                                     Y33
 80     |--|/|---|/|-----------------------------------------------------( )--|
        | VICE/  VICE                                                    RETRT
        | PALLT   ON                                                     VICE
        |  ERR   PALLT                                                    RAM
        |
        | M111   X11    X13                                              M110
 83     |--|/|---| |----| |----------------------------------------------( )--|
        | VICE/  VICE   VICE                                             VICE/
        | PALLT  RAM     ON                                              PALLT
        |  ERR   RETRD  PALLT                                            COMP
        |
        | X4
 87     |--| |------------------------------------------------[ R      S604 ]-|
        | ENABL                                                          VICE
        | VICE                                                            TO
        | PALLT                                                          PALLT
        |
 89     |----------------------------------------------------------------[RET ]-|
```

CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 10

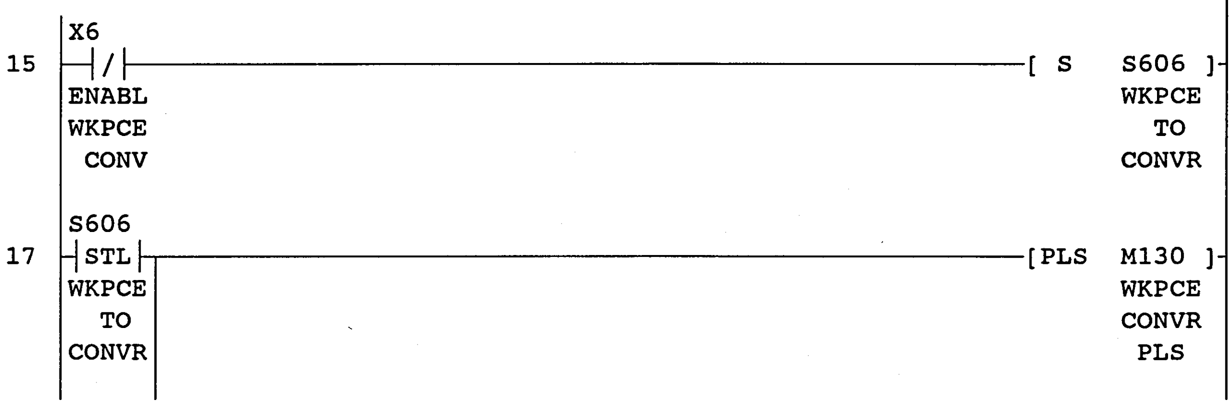
START MACHINING



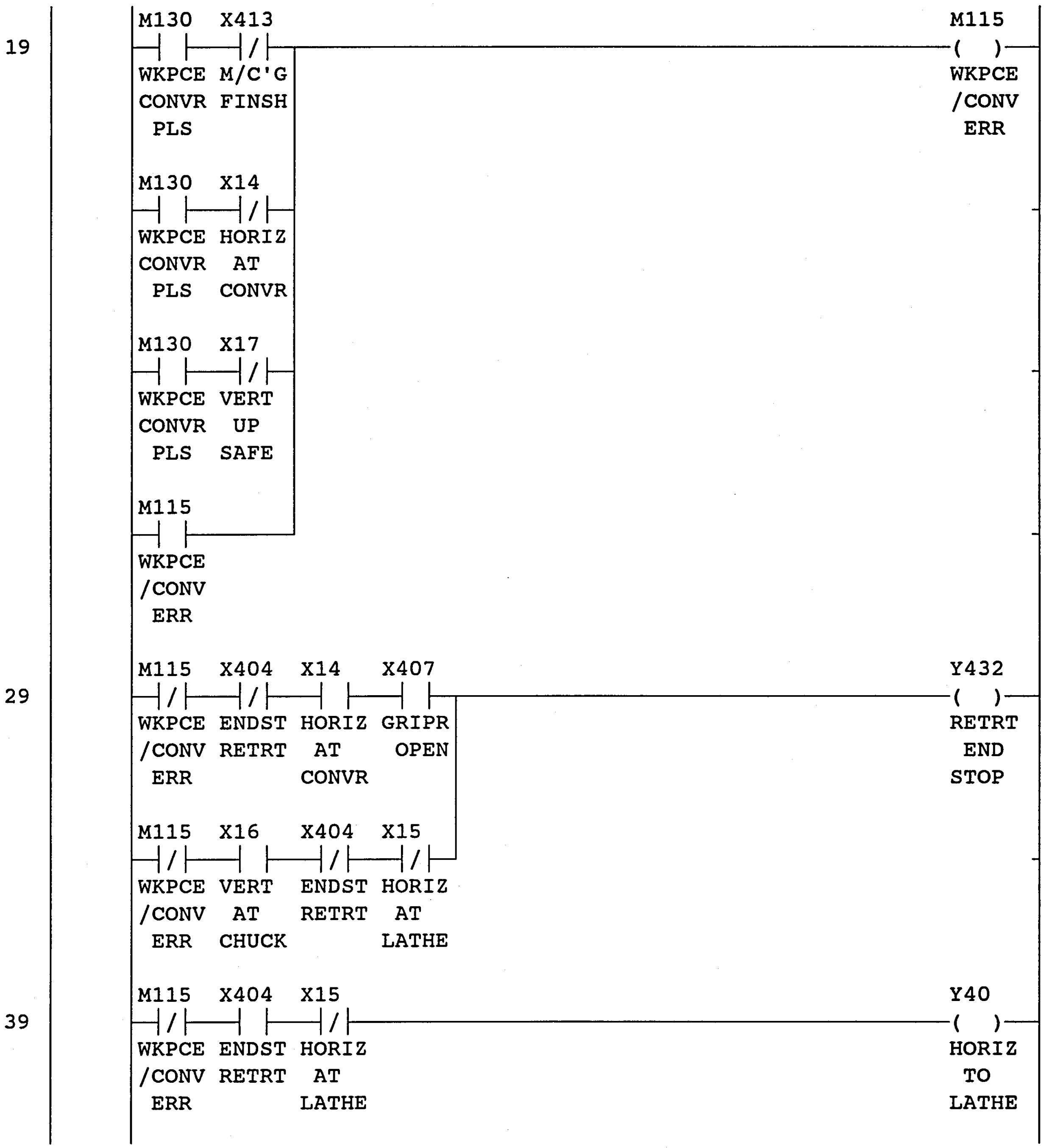
CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 11



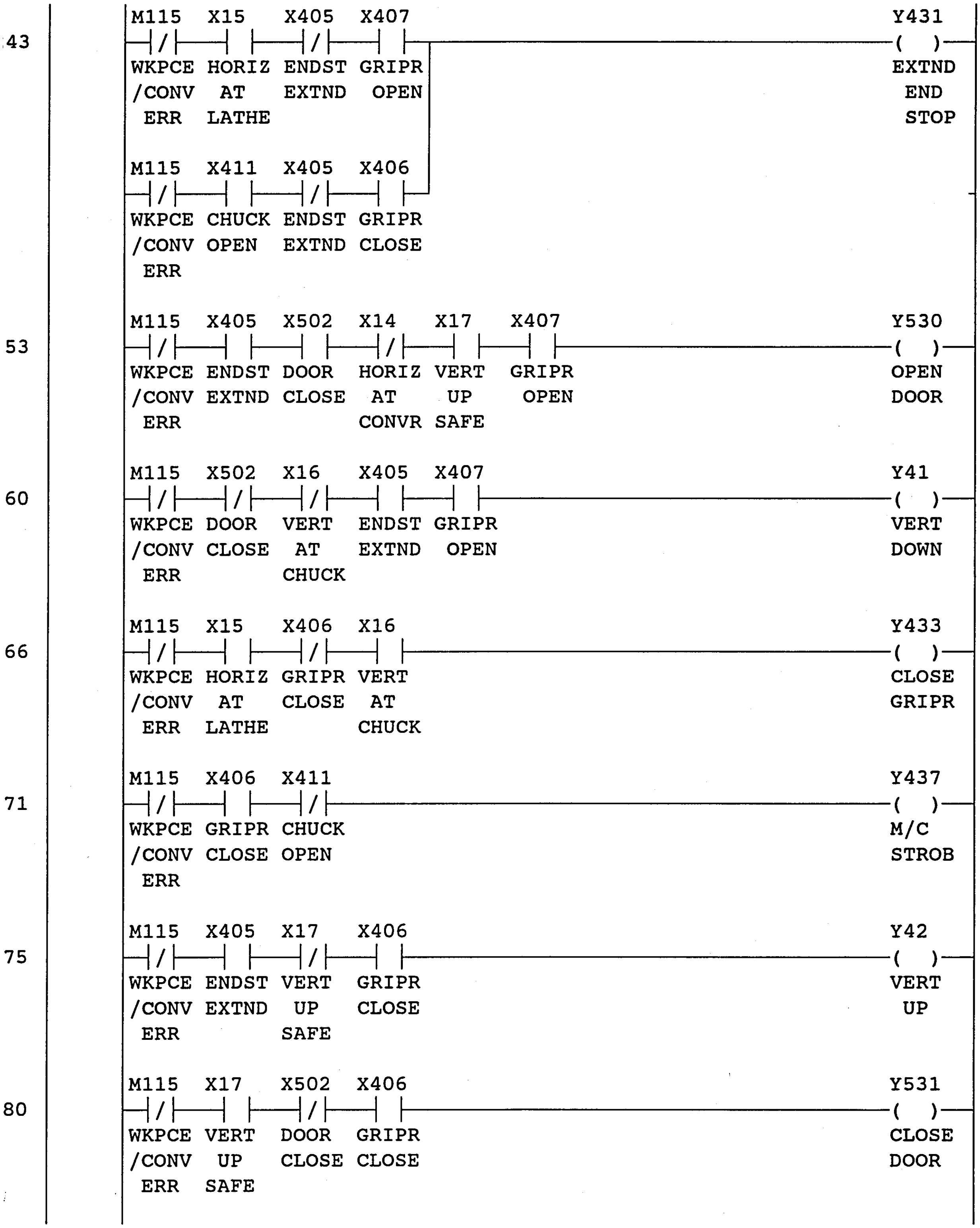
WORKPIECE TO CONVEYOR



CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 12



CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 13



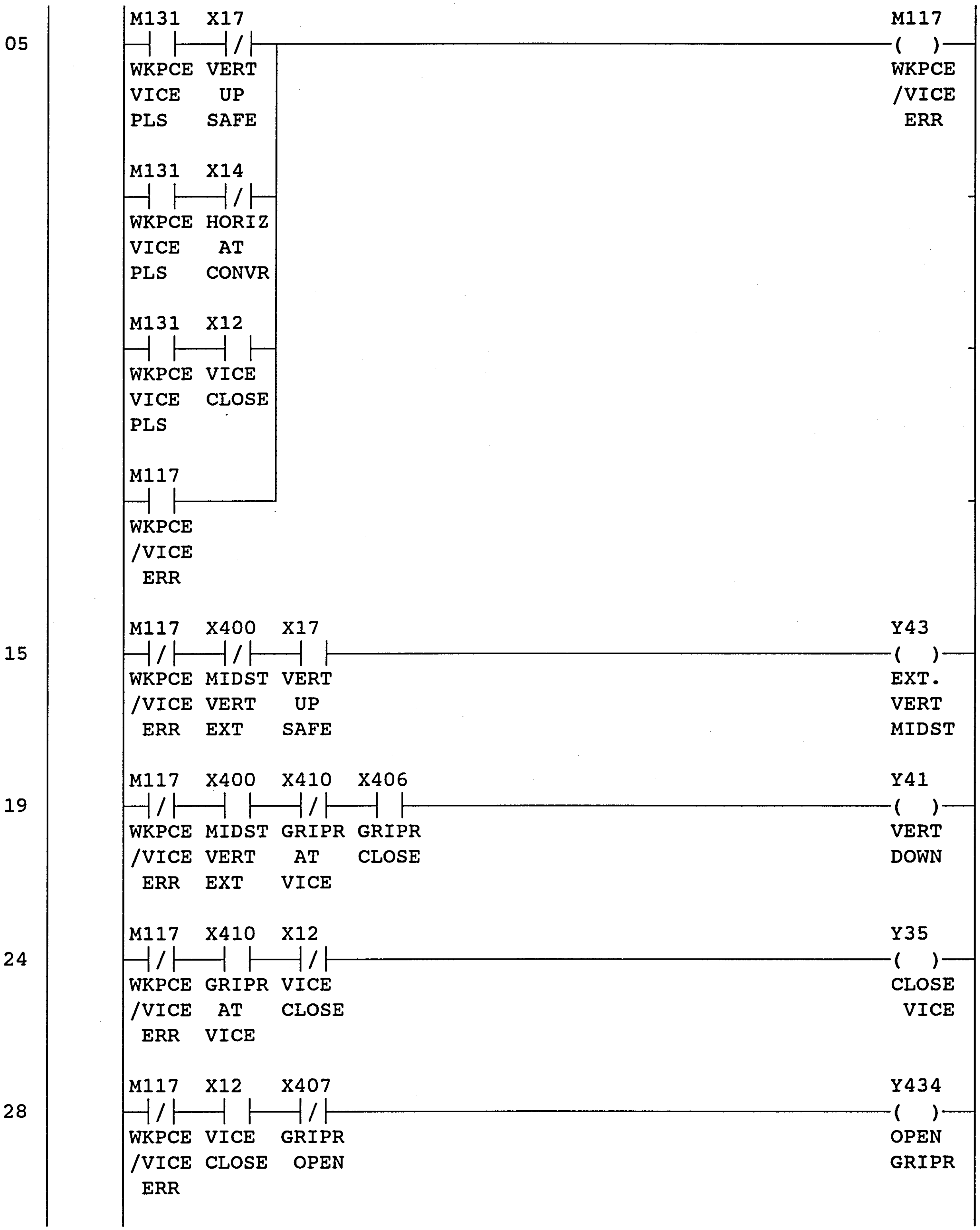
CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 14

```
          M115  X502  X14   X406                                    Y37
85       -|/|---| |---|/|---| |-------------------------------------( )-
          WKPCE DOOR  HORIZ GRIPR                                   HORIZ
          /CONV CLOSE  AT   CLOSE                                    TO
           ERR        CONVR                                         CONVR

          M115  X14   X402  X406                                    Y45
90       -|/|---| |---|/|---| |-------------------------------------( )-
          WKPCE HORIZ ROTAT GRIPR                                   ROTAT
          /CONV  AT     0   CLOSE                                   TO 0
           ERR  CONVR  DEG                                          DEG

          M115  X402                                                M114
95       -|/|---| |-------------------------------------------------( )-
          WKPCE ROTAT                                               WKPCE
          /CONV   0                                                 /CONV
           ERR   DEG                                                 COMP

          X6
98       -| |-----------------------------------------------[ R     S606 ]-
          ENABL                                                     WKPCE
          WKPCE                                                       TO
           CONV                                                     CONVR

00       ---------------------------------------------------------[RET  ]-
```

WORKPIECE TO VICE

```
    X7
01 -|/|-------------------------------------------------[ S     S607 ]-
    ENABL                                                     WKPCE
    WKPCE                                                       TO
     VICE                                                     VICE

    S607
03 -|STL|-----------------------------------------------[PLS   M131 ]-
    WKPCE                                                     WKPCE
     TO                                                       VICE
    VICE                                                      PLS
```

CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 15

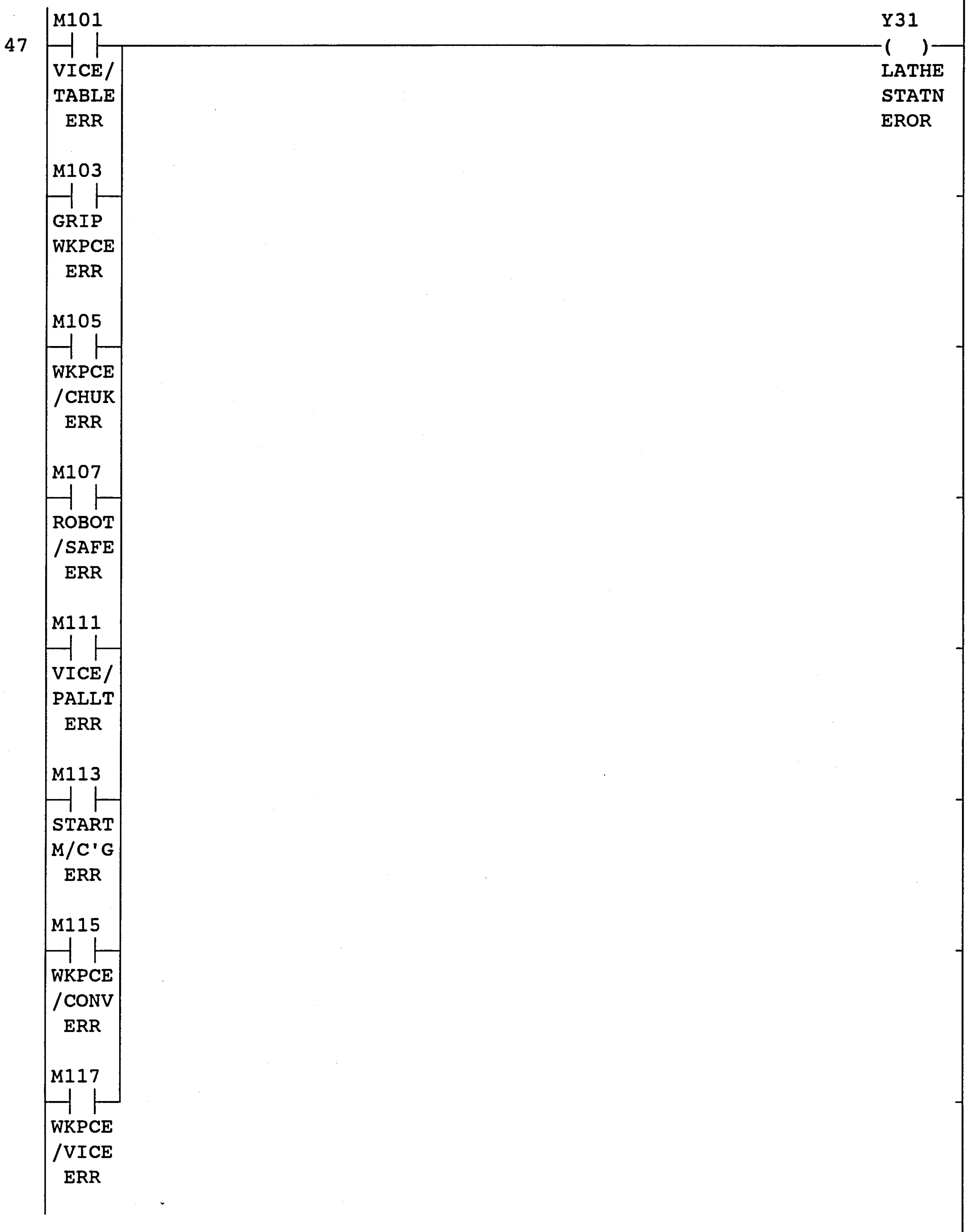


CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 16

```
     M117  X407  X17                                         Y42
32   --|/|---| |---|/|-------------------------------------( )--
     WKPCE GRIPR VERT                                        VERT
     /VICE  OPEN  UP                                          UP
      ERR        SAFE

     M117  X17   X404  X407                                  Y432
36   --|/|---| |---|/|---| |-------------------------------( )--
     WKPCE VERT  ENDST GRIPR                                 RETRT
     /VICE  UP   RETRT  OPEN                                  END
      ERR  SAFE                                              STOP

     M117  X404                                              M116
41   --|/|---| |-------------------------------------------( )--
     WKPCE ENDST                                             WKPCE
     /VICE RETRT                                             /VICE
      ERR                                                     COMP

     X7
44   --| |-------------------------------------------[ R   S607 ]-
     ENABL                                                   WKPCE
     WKPCE                                                    TO
      VICE                                                   VICE

46   --------------------------------------------------------[RET ]-
```

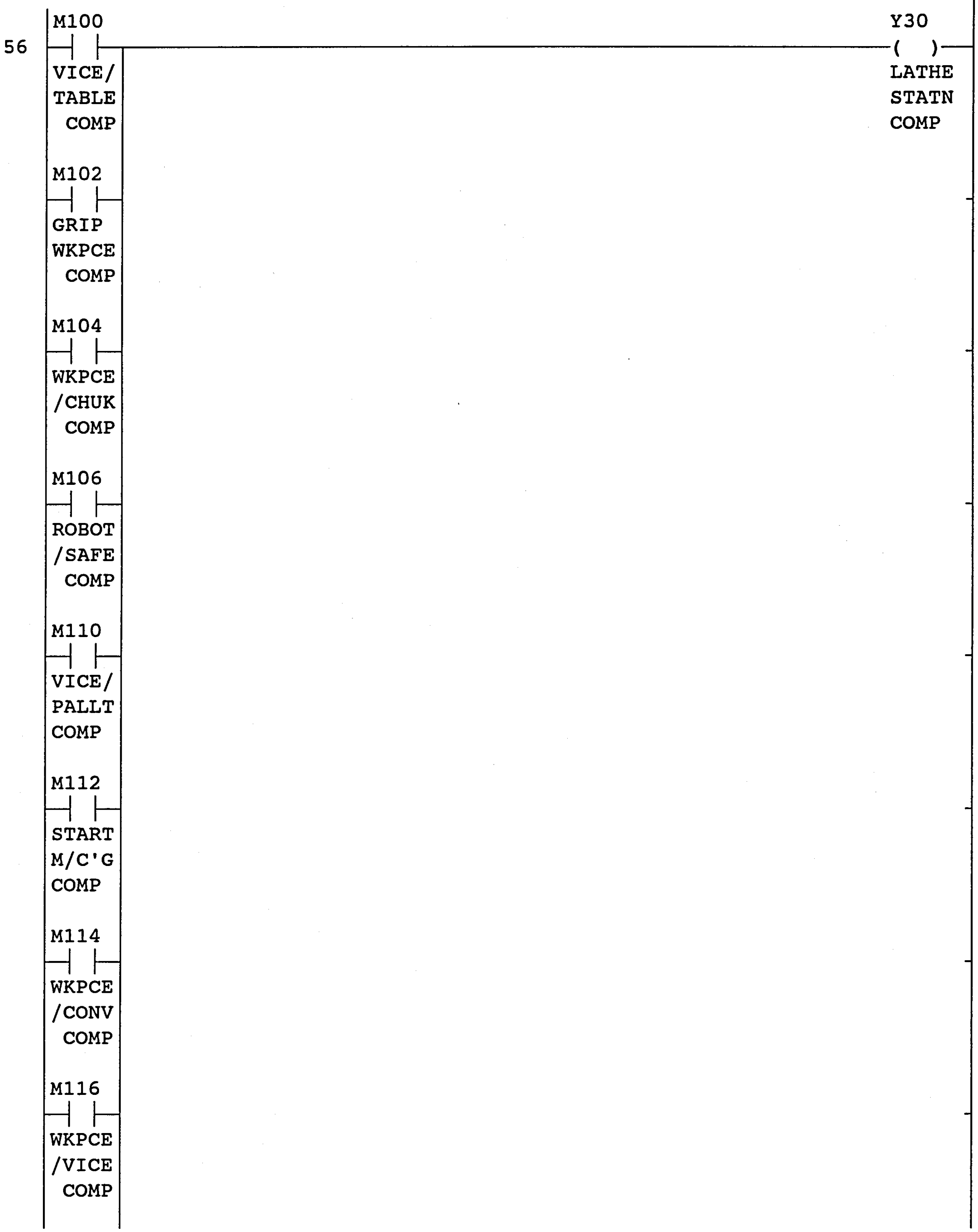
CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 17

GENERATE ERROR STATUS



CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 18

GENERATE COMPLETION STATUS



CONTROL OF WORK HANDLING, GANTRY ROBOT AND MACHINING FOR LATHE STATION IN THE SCHOOL OF ENGINEERING'S FMC	SHEFFIELD HALLAM UNIVERSITY	Date:21/04/93	Proj:LATHE
		Rev.dat:	Syst:F1/F2
		Rev.no: 6	Type:Ladder
	Draw.no:	Sign: A.T.	Page: 19

TMB

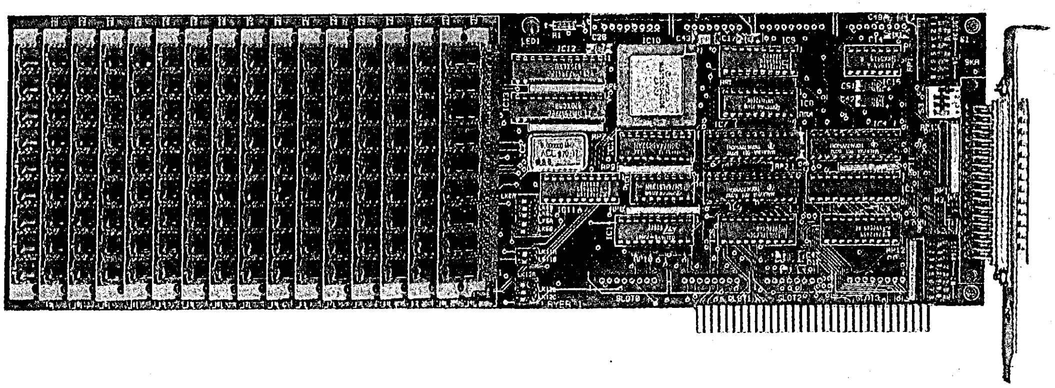
Transtech TMB04

TMB

AN EXPANDABLE TRANSPUTER BOARD FOR THE IBM PC

Features

- ♦IMST800, IMST425 or IMST414 transputer options
- ♦1 to 16 MBytes of dynamic RAM with zero wait state option
- ♦20, 25 or 30 MHz transputer speed options
- ♦Four standard TRAM slots
- ♦DMA/IRQ capability for fast MS.DOS I/O transfer
- ♦Plugs directly into IBM PC AT or XT and compatibles
- ♦Supports Occam TDS development and 3L's scientific languages
- ♦Compatible with the Transtech range of TRAMs



Introduction The Transtech TMB04 is part of a compatible family of transputer development boards. The TMB04 will fit any IBM PC AT or XT (and compatibles) expansion slot, providing the interface between the PC running a file server under MS.DOS and the processing power of transputer systems.

Flexibility The TMB04 has been developed with Transtech's unrivalled experience of transputer boards to provide the most flexible single transputer board available.

Capable of supporting any speed variant of the IMST800, IMST425 and IMST414 transputers, the TMB04 also has options for 1,2,4,8 or 16 MBytes of DRAM, which can be accessed in 3,4,5 or 6 cycles depending on the speed of processor and memory. The four links of the transputer can be accessed via a standard 37-way D-type edge connector compatible with a wide range of other transputer boards. These links can be set to run at 10 or 20 Mbits/sec.

Transtech TMB04

TMB

IBM PC Bus Interface An interface to the PC bus is provided via an IMSC012 link adaptor for communication between the host machine and a transputer network. The interface supports software polling of the link adaptor, used in many earlier transputer boards and also a DMA mechanism allowing transfer rates of between 200 and 300 KBytes/sec to be achieved. The TMB04 also has the ability to interrupt the host PC on a number of user defined events.

Expandable More flexiblity is provided by the addition of four TRAM (TRAnsputer Module) slots, which allow up to four standard TRAMs to be added to the board. The TMB04 accepts the whole range of TRAMs from Transtech and is compatible with those of other manufacturers, giving customers the freedom to choose many different processor and memory combinations or application specific TRAMs. Eight links from the four TRAM slots are also taken to the 37-way D-type edge connector, allowing them to be connected to external devices as well as to the master transputer on the TMB04. Further information on the Transtech range of TRAMs is available from Transtech or your local distributor, while details on the TRAM standard and TRAM motherboard architecture are published by Prentice Hall in 'Transputer Technical Notes' ISBN 0-130929126-1.

System Control The reset, error and analyse system control of the transputer is user definable, by selecting one of a variety of reset configurations. The master transputer can be reset either from the external world via the edge connector, or from the PC, while the remaining TRAM slots can be reset from either the same source as the master transputer or from a sub-system generated by the TMB04. The generation of the sub-system enables the TMB04 to control a large system of transputers while still running the TDS.

Software The TMB04 can be programmed with software development packages to run Occam, C, FORTRAN, Pascal and other transputer language compilers as well as the Helios, TransIDRIS, Express and other transputer operating systems. It can also be used as the processing hardware for a number of application specific software packages that are available for the transputer. The board is also supplied with a diagnostic test program.

Ordering Information

TMB04 -	PROCESSOR OPTION	MEMORY OPTION
	A=IMST414-20 B=IMST425-20 C=IMST425-25 D=IMST425-30 E=IMST800-20 F=IMST800-25 G=IMST800-30	1=1 MBYTE 2=2 MBYTES 4=4 MBYTES 8=8 MBYTES 16=16 MBYTES

FAST 3 CYCLE RAM IS AVAILABLE FOR ALL 20 MHz PROCESSOR OPTIONS AND SHOULD BE SPECIFIED BY ADDING "F" TO THE PART NUMBER e.g.TMB04-E-2F HAS AN IMST800-20 WITH 2 MBYTES OF FAST 3 CYCLE DRAM

TRANSTECH DEVICES LIMITED
Unit 17, Wye Industrial Estate
London Road
High Wycombe
Buckinghamshire
HP11 1LH
England
Telephone: [+44] 0494 464303
Facsimile: [+44] 0494 463686

Document Reference:TMB04FLY0789

4. INSTALLATION

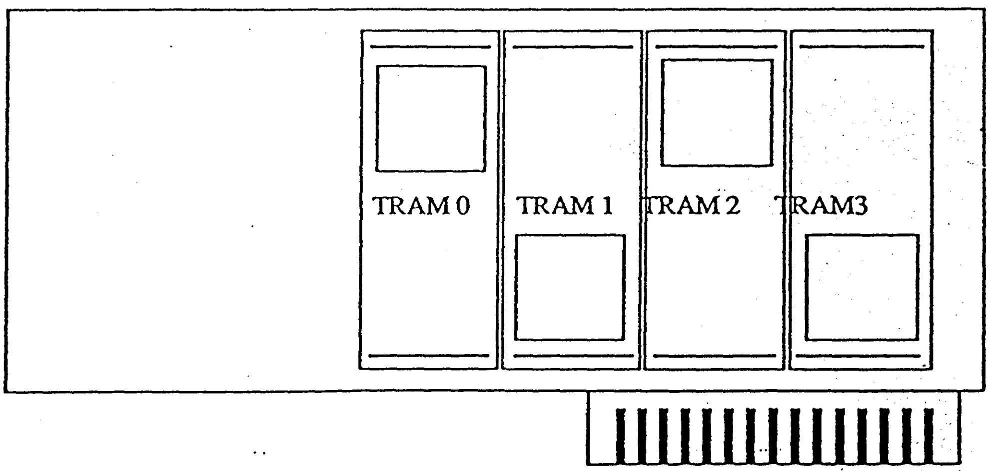
4.1 TRAM Fitting and Handling

Care must be taken when fitting or removing TRAMs from the TMB04, to ensure no damage occurs to the TRAM pins. A white circle or triangle in the corner of each TRAM slot indicates Pin 1 of that slot on the TMB04. TRAMs also have an indicator for pin 1, which should be matched with the marking on the motherboard to ensure correct orientation. A TRAM plugged in the wrong way round may result in damage to the TRAM and the motherboard. The TMB04 has some components mounted on the board between the TRAM sockets, which foul some of the TRAMs available. If a TRAM cannot be fitted without fouling these components, stand-off strips or larger pins supplied with the TRAM should be fitted to allow the TRAM to be raised above these components.

4.2 Installing a TMB04

Remove the TRAM that is to be the root processor from its protective packing, observing the appropriate anti-static handling precautions.

Plug the TRAM into slot 0 ensuring pin 1 matches pin 1 on the TMB04. The 16 pins that carry the signals should fit into the required slot, so a TRAM larger than size 1 that fits over more that one slot will cover adjacent slots to TRAM slot 0, i.e. a size 2 TRAM will cover slots 0 and 1 and a size 4 TRAM will cover slot 0, 1, 2 and 3. If the TRAM is larger than size 1 insert link jumpers in the slots that do not carry signals to ensure the continuity of the hardwired pipeline. A jumper is an 8 way connector that connects link 1 to link 2 on a slot that does not carry any signals, and should be inserted at one end of the slot with its indicator lining up with the Pin 1 index of the slot. If a TRAM is stacked on top of it the jumpers need to be removed from the slots used for stacking. Install other TRAMs as required in the other slots of the TMB04, not necessarily in order, provided that the pipeline is maintained as discussed in section 2.2. Pipe jumpers should also be fitted in the slots that are not being used as shown in the following diagram where slots 2 and 3 have been jumped out to complete the pipeline. Once all the TRAMs have been fitted follow the instructions in your PC manual for installing an option board.



1 Introduction

This module is intended to allow the user a vehicle whereby the simple basic RS232 interface can be accessed from directly within a transputer network. This interface is not intended to perform at very high data rates and there is only a limited amount of ram on the module to allow buffering of data, but this interface is intended to function more than adequately for such applications as: mouse interfacing, keyboard interfacing, printer interfacing, instrument interfacing.

The module contains its own EPROM which allows the module to become a standalone sub-system. The simple example is to treat the module as an Inmos link to RS232 interface converter which requires no initialisation. Any bytes sent down the link will appear at the interface with no setup required.

By moving away from the obvious functions of serial interfacing it can be seen that this module can also be used in areas where direct signal sampling at moderate data rates is required.

The port lines are controlled from a 22v10 pal which is memory mapped into the transputer and is available as 4 bits of read data and 4 bits of write data, the read and write data registers are not read/write registers but are write only/read only registers mapped at the same address.

Access to the data is by simple PLACED access of a memory location for the setting (write) of data or the sample (read) of data.

PRELIMINARY

2 Hardware Description

2.1 T222 system functions

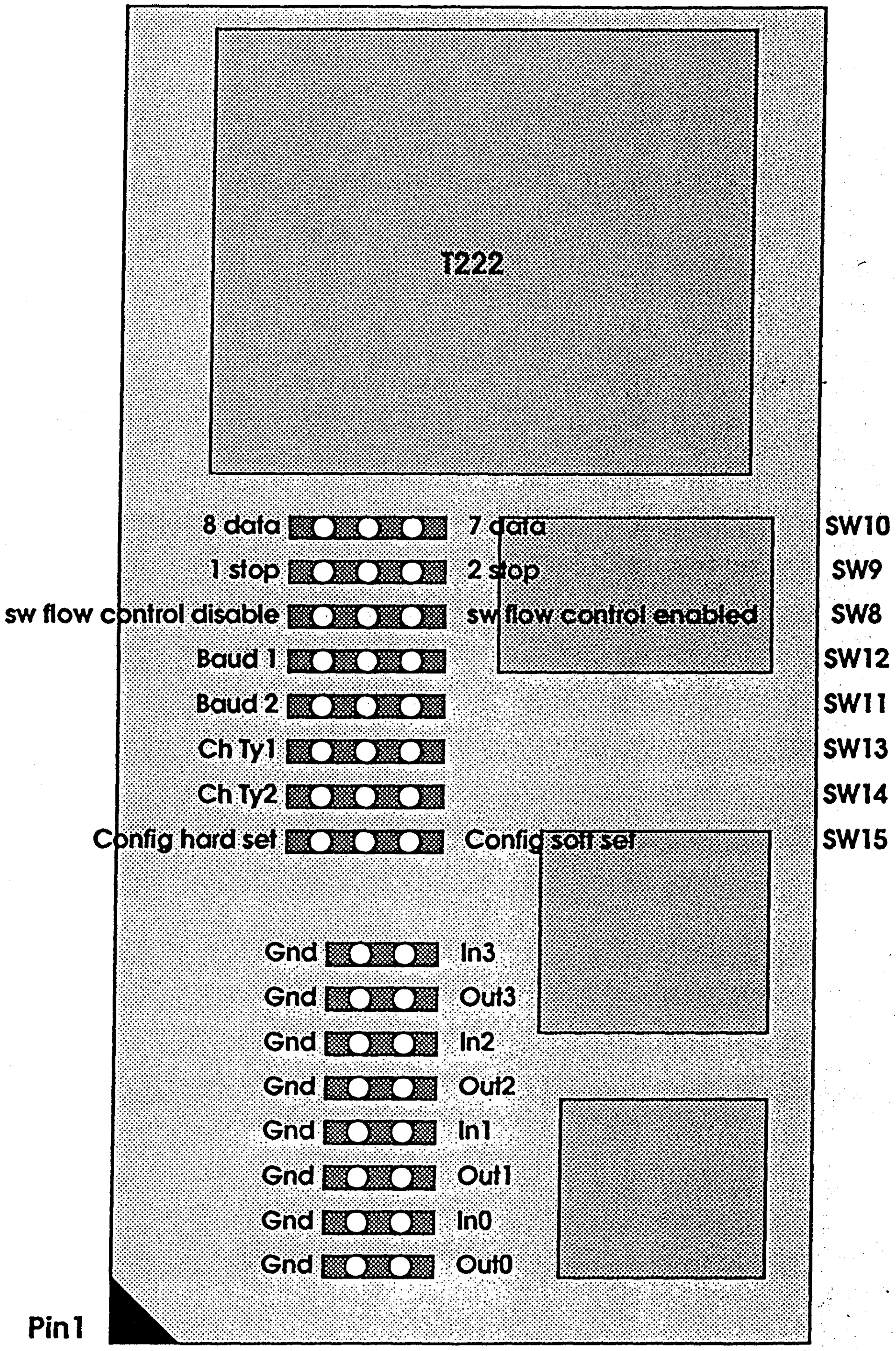
The T222 transputer uses its four links to act as a peripheral controller capable of receiving data from four sources. This means that adjacent transputers can access the peripheral system directly without having to go through the IBM system. The T222 is normally used in boot from rom mode NOT boot from link mode, this is selected by the position of a zero ohm resistor identified on the underside of the board.

The T222 has a 64K memory map, and on this module it is split into two 32K sections: the lower 32k ($8000 to $FFFF) being for up to 32K * 8 ram (normally 8K * 8), which will allow word access in single cycles where the on chip ram exists, otherwise in 4 processor cycles, the top 8 bytes of this ram site are ($FFF8 to $FFFF) also mapped into the on board function latch. Which are read as the 8 switch inputs on data lines D15 to D8 and the input 4 serial lines on D3 to D0. The output 4 serial data lines are also on D0 to D3.

The top 32K ($0000 to $7FFF) is primarily intended as a rom site to allow the module to boot up using pre-configured code. The wait state gen-

PRELIMINARY

erator can be set to use 200ns (9 cycle access) or 250ns (11 cycle access) rom devices.



Sunnyside
systems limited

iot332

32 Channel Digital Input/Outpı Transputer Module

NEW PRODUCT INFORMATION

Features

- 32 TTL-compatible Input/Output lines
- I/O lines selectable in groups of 8
- 64mA output drive capability
- Readback capability on all lines
- All lines set to inputs on power up
- Inmos size 2 TRAM format

The iot332 is a size 2 programmable digital input/output module, whose 32 TTL-compatible lines are organised as 4 groups of 8 lines. Each group can be selected, under software control, as either input or output and all lines give full, bipolar TTL drive compatibility, with up to 64mA drive capability.
Each line, in an 8 line group, can be cleared or set under software control. The status of lines set to output can be read back for verification purposes.
Interface to the Transputer bus is via an Inmos CO11 Link Adapter. External connections are made by a 50-way IDC connector, compatible with the ARCOM™ range of signal conditioning cards. The module comes with C callable drivers for the 3L Compiler/Configurer and the Inmos D7214B Toolset.

specifications

Number of channels	32
Configuration	4 groups of 8, selectable as either Input or Output
I/O levels	TTL compatible
Output drive capability	64mA
Read/Write Update Rate	500,000 channels/sec
Power Requirement	+5 Volts (+/-5%) @ 100mA quiescent @ 1.6 A maximum load
Temperature Range	5C to 50C Operating -25C to 85C Storage
Module Size	Size 2 TRAM, 2.15" x 3.66" (5.5x9 cm)

Sunnyside Systems Ltd, Rosebank Nursery, Kirkton Campus, Livingston EH54 7AN, Scotland Tel : 0506 460345 Fax : 0506 460314

 iot332/11/91

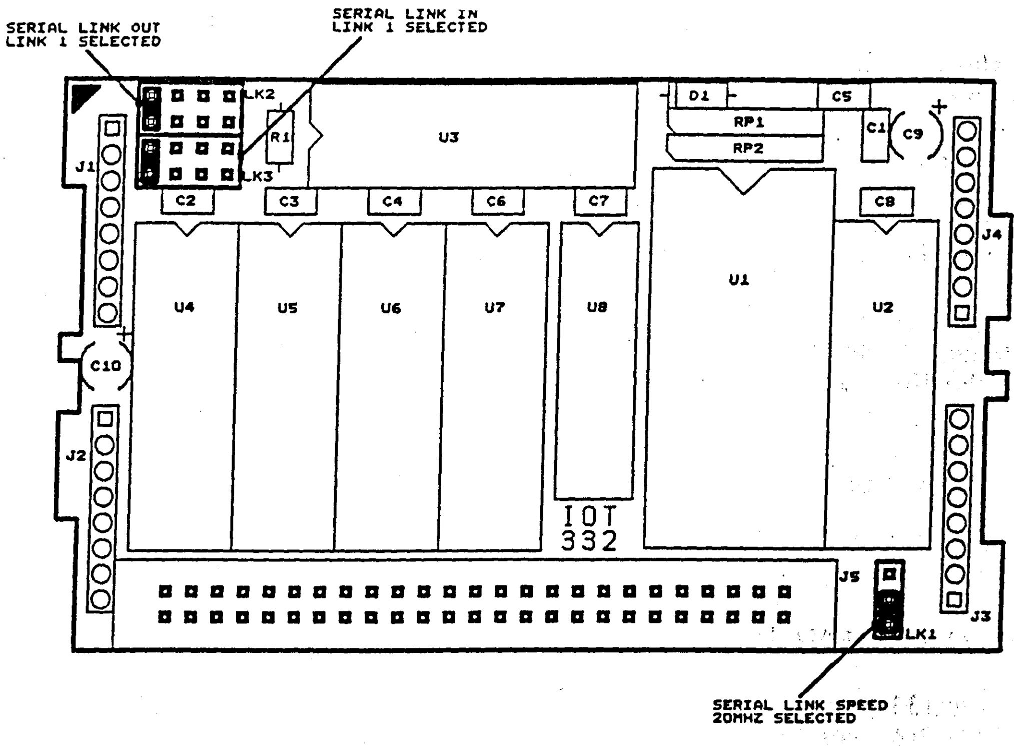


Figure 2.1.1 Jumper Positions and Factory Set Conditions

2.2 Confidence Test

Before running the confidence test, change the jumper settings for the serial link configuration as shown in Figure 2.2.1 below. The jumper links should be set such that the iot332 communicates on link0.

The test simulates the action of an eight bit counter on each of the four groups (ports) of inputs/outputs, (the 32 I/O lines are organised as four ports, each with eight lines). The maximum frequency is present on the LSB I/O pin, the minimum on the MSB I/O pin.

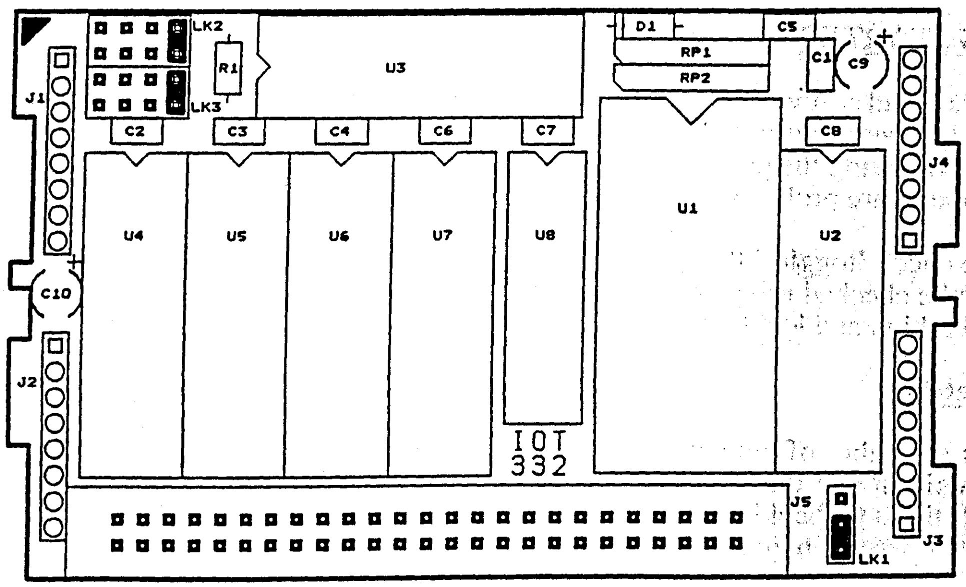


Figure 2.2.1 Confidence Test Serial Link Positions

9. ~~List of~~ References

1 - Bill's Notes ?

? ?

2 - Das P. K., Fay Freund E., 1983, "Fast Non-linear Control With Arbitrary Pole Placement for Individual Robots and Manipulators". The International Journal of Robotics Research, 1(1) : 65-78.

3 - F. Naghdy, P. Strickland, 1989, "Parallel Control of Advanced Production Machines Based On Transputer/Occam". Procs Centre for Advanced Technology In Teleco Australian Transputer and Occam User Group Conference Proceedings, pp 113-21.

4 - Occam 2 Reference Manual, 1988 Inmos Ltd., C.A.R. Hoare Series Editor.

5 - N. Leveson, J. Stolzy. "Safety Analysis Using Petri Nets". IEEE SE13 n3 March 1987, pp 386-397.

6 - "The Development of a Dependable Distributed Control System for Flexible Manufacture". W. M. M. Hales, P. Gray, F. Poole, Sheffield Hallam University.

7 - "The Application of Transputers to the Control of a FMC". P. Gray, PhD Thesis to be published.

8 - The Transputer Instruction Set; A Compiler Writer's Guide. "Transputer Architecture and Overview; Transputer Technical Specifications". 1990 Inmos Ltd.

9 - Plastics Technology, March 1992, pp 13-15. "High Speed Multi Tasking Ability Claimed For New Transputer Controls".

10 - Occam 2 Toolset User Manual, 1989 Inmos Ltd. "A Tutorial Introduction To Occam Programming"

11 - "Petri Net Theory and The Modelling of Systems". James L. Peterson, 1981.

12 - "The Use of Software Engineering Techniques for the Development of Control Software in a Network Transputer Systems FMS Operation". S. Low. In B. Worthington (Ed.), Advances in Manufacturing Technology III, Production Research, Sheffield. ISBN 1-85091-704-3. 1988, p425.

13 - "Software Design for Real Time Systems", J. Cooling. ISBN 0 412 34180 8. Chapman and Hall, London 1991.

14 - "Structured System Analysis and Design Methodology", G. Cutts. ISBN 0 632 02818 1. Blackwell, 1987.

15 - "Formal Verification of Safety Critical Systems", L. Moser and Melliar-Smith. Software-Practice and Experience, v20, Aug. 8 1990, pp 799-821.

16 - Mitsubishi Electric Co., Advancing Technology, Industrial Automation and Control Equipment, "Programmable Logic Controllers for Every Level of Control". Cat. AFPC/5000/11/89.

17 - Mitsubishi Electric Co., Programming Manual, Enhanced F2 Series. JY992D15701A.

18 - "Programmable Logic Controllers; Architecture and Applications". Gilles Michel, copyright 1990 John Wiley & Sons Ltd.

19 - "Programmable Logic Controllers; Principles and Applications (Second Edition)". John Webb, copyright 1992 Macmillan Publishing Company, ISBN 0-02-424970-X.

20 - "Systematic Logic Controller Programming; A Self Study Course". D. Wardman, School of Engineering @ Sheffield Hallam University.

21 - Transputer Education Kit; User Guide. "Theory of Operation, Installation, Schematics", 1990 Inmos Ltd.

22 - "Configuring Occam Programs to Meet Real-Time Constraints in a Multitransputer Workstation". P. R. Croll, PhD Thesis 1989, Sheffield University.

23 - "Transaction Recovery Architectures for Parallel Database Machines". R. J. Oates, PhD Thesis 1991, Sheffield University.