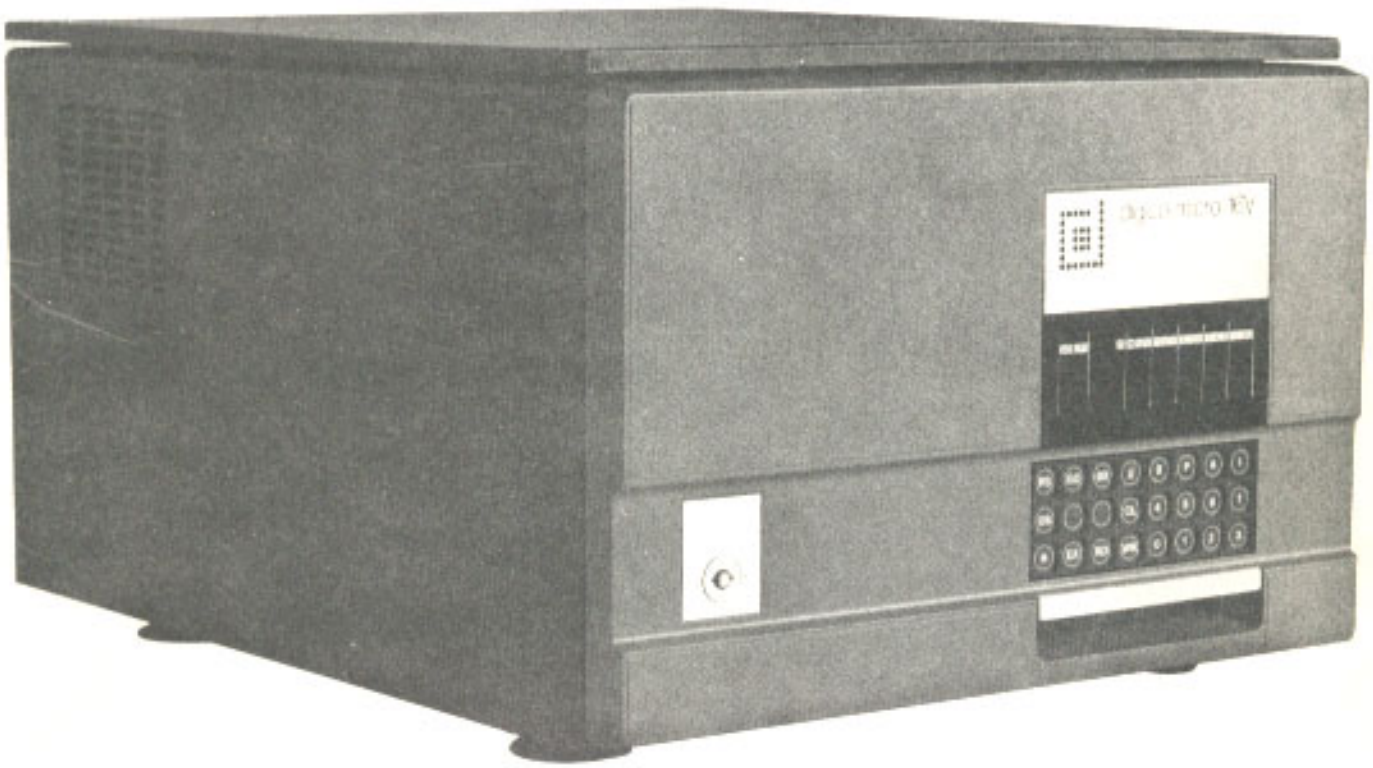
16V

micro16V computer manual



digico

LIST OF CONTENTS

		Page
1.	INTRODUCTION	1
	1.1 Hardware	
	1.2 Software	
	1.3 Applications	
2.	HARDWARE PHILOSOPHY	3
	2.1 Simple Hardware Organisation	
3.	OPERATION	4
	3.1 Controls	
	3.2 Reading from Core and Writing to Core	
	3.3 Examples of the Use of the Control Panel	
4.	INSTRUCTION FORMAT	7
	4.1 Addressing Instructions	
	4.2 Non-Addressing Instructions	
	4.3 Input/Output Instructions	
5.	INSTRUCTION REPERTOIRE OF MICRO 16V	9
	5.1 Direct Store Addressing Instructions	
	5.2 Indirect Store Addressing Instructions	
	5.3 Indirect for Transfer of Control (Jumps)	
	5.4 Non-Addressing Instructions	
6.	BOOTSTRAP	17
7.	INTERRUPTS	18
	7.1 Standard Interrupt Facility	
	7.2 Multilevel Priority Interrupt Option	
8.	MEMORY PARTITION OPTION	24
	8.1 Restricted Instructions	
	8.2 Instructions to Control the Memory Protect Option	
	8.3 The Use of the Memory Partition Option for Executive	
9.	RESERVED STORE LOCATIONS	27
10.	MAINS FAIL — RESTART	29
11.	REAL TIME CLOCK OPTION	30
12.	FLOATING POINT UNIT	31

13. MEDIUM SPEED INTERFACE 32

14. DIRECT MEMORY ACCESS (HESITATE) 37

15. INPUT/OUTPUT TELETYPE 41
- 15.1 Operation
- 15.2 Controls
- 15.3 Teletype Instruction Set
- 15.4 Notes on Programming

16. SOFTWARE INTRODUCTION 48
- 16.1 Executive
- 16.2 Symbolic Assembler
- 16.3 Mathchat
- 16.4 Floating Point Package
- 16.5 Utility Routines
- 16.6 User Programs

17. SERVICES 50
- 17.1 Application Development
- 17.2 Turnkey Systems
- 17.3 Customer Assistance
- 17.4 Training Courses
- 17.5 Maintenance

1. INTRODUCTION

The Digico Micro 16V is the latest addition to the highly successful Micro 16 series of computers. By keeping pace with economic advantages of developments in electronics Digico offer even better value to the purchaser of a Micro 16V computer.

1.1 Hardware

The Micro 16V has 16 bit binary words with a 950 nS core cycle time. The core memory is available in 4K modules from 4K words to 64K words. A good instruction set allows hardware direct and indirect addressing.

The wide variety of available peripherals include Teletypes, paper tape readers and punches, card readers and punches, industry compatible and cassette magnetic tapes, discs and drums, terminals for both alphanumeric and graphic displays, communication data links, line printers, graph plotters, analogue/digital and digital/analogue converters and a coupler for the ICL 1900 series of computers.

1.2 Software

The basic programming language is Symbolic Assembler. As well as the normal assembler facilities a wide variety of macros are available including multiply and divide, floating point instructions, square root, natural logarithm etc. Compilers are available for certain mathematical languages including Mathchat which is an on-line conversational interpretive language designed to be used as a tool for solving problems expressed in standard mathematical form.

Utility routines provided include buffered input and output routines, disc utilities, octal debug and mathematical functions.

Through Digico's policy of providing complete turnkey services a number of software packages have already been written for certain specific applications. Where required, Digico's staff will be pleased to apply their expertise to the solving of customers' problems.

1.3 Applications

The applications for which the members of the Digico Micro 16 series of computers have been used are numerous and widely varying. The uses encompass all fields of modern industry and scientific research. The remaining portion of this section lists some of the many utilisations of the Micro 16, and if any particular application corresponds to problems you have Digico will be delighted to send further details and advice.

1. Animal feed mix control
2. Bookshop stock control
3. Car park control
4. Census analysis
5. Colour matching
6. Key to Disc Systems
7. Electro-encephalograph (EEG)
8. Electron Spin resonance/Nuclear-magnetic resonance (ESR/NMR)
9. ICL 1900 front ending
10. Gas chromatography
11. Computer Monitoring of Explosions
12. Invoicing
13. Machine Tool control
14. Rock Magnetometer
15. Mass Spectrometry
16. Meteorology
17. Pathlab
18. Quantity Surveying
19. Radioimmunoassay
20. Computers in Railway Environment
21. Stock Control
22. X-Ray Spectroscopy
24. Analysis of Aircraft Radio Navigational Aids
25. Typesetting

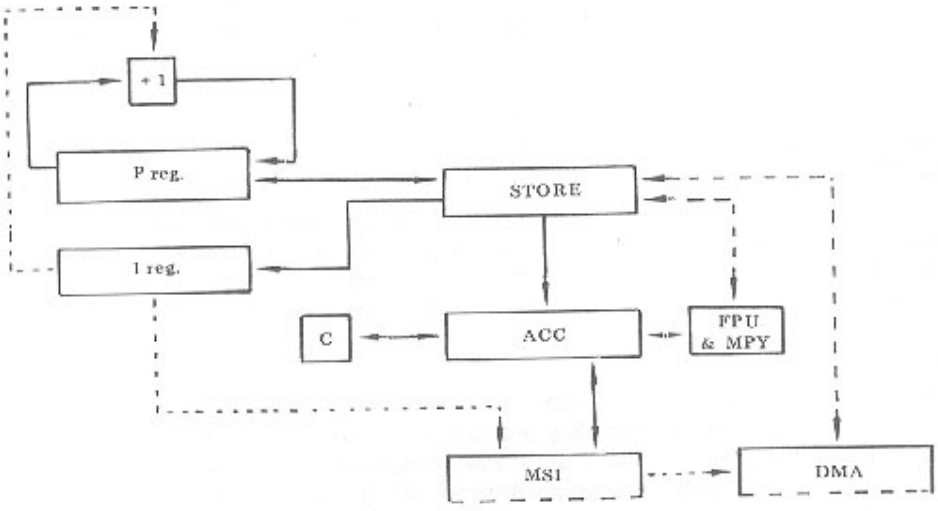


DIAGRAM 1 — HARDWARE CONFIGURATION

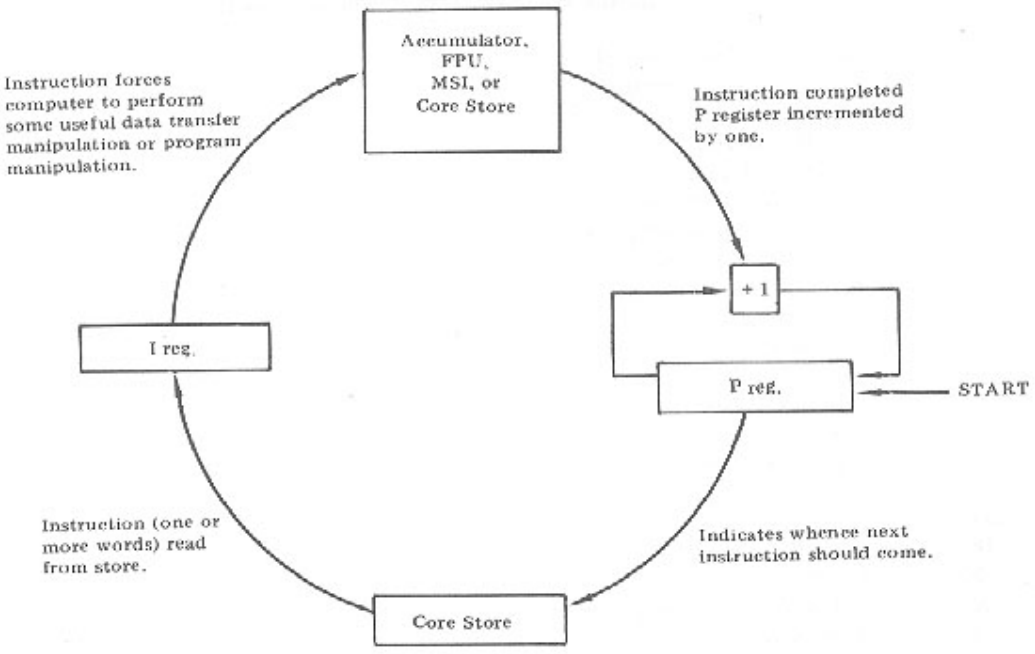


DIAGRAM 2 — PROGRAM CYCLE

2. HARDWARE PHILOSOPHY

The Micro 16V is a high speed general purpose digital computer, designed to work at the heart of a complete system. The computer is built for simplicity of programming and operation, hence the requirement of only one accumulator. Digico have already been highly successful in this field as can be seen from the impressive list of applications, which have been achieved with the earlier Micro 16S and 16P. The Micro 16V, however, has several advantages over these two machines, although the main visible advantages are the featherlight keyboard which allows entry of data in octal, and the display of any register. These are all explained in detail in Section 3, on the Operation of the Computer.

Some peripherals require very fast data transfer, and for these the Direct Memory Access (DMA) option is available. Such peripherals are usually storage devices such as discs, magnetic tapes. Employing DMA, a maximum rate of transfer of 857 thousand words per second is possible. Control of DMA is effected by the Medium Speed Interface, which also is used to transfer data to and from the accumulator, but at much lower speeds.

2.1 Simple Hardware Organisation

Registers

The general hardware configuration is illustrated in Diagram 1, showing the interconnection of the various registers with regard to the core store. Various options such as DMA, and FPU are also included. In the execution of a program a certain cycle of events is carried out. This is indicated in Diagram 2, and the way each register is brought into play.

The *Accumulator* is a 16 bit register for holding data temporarily, for arithmetic operations and input and output of data via the Medium Speed Interface. For arithmetic purposes the Most Significant Bit is conveniently considered as the sign bit. (See Appendix 1).

The *Carry Register* records the carry which occurs as a result of operations on the accumulator carried out by ADD, ADI, SUB, CRS, CRU. The register is set by ADD or ADI if the addition (A + S) is greater than *177777. The register is set by SUB, if the addition (A + ($\overline{S}$ + 1)) is greater than *177777, where A represents the contents of the accumulator and S represents the contents of the store.

The *Instruction Register* is a 16 bit register into which instructions are loaded for decoding and execution.

The *Program Address Register* is also a 16 bit register and contains the address of the next program instruction to be executed. In general, one is added to this register as each instruction is obeyed.

Data Transfer

The *Medium Speed Interface* is capable of transferring data between peripherals and the accumulator at speeds of up to 80 thousand characters per second

Direct Memory Access is an option which provides eight channels through which peripheral devices can transfer data directly to and from store. This facility is needed by peripherals which have very high data transfer rates, and for which it is not possible to transfer through the Medium Speed Interface. Such a peripheral is program controlled via the MSI, but transfers data via the DMA.

Other options include the *Floating Point Unit* which provides fast floating point mathematics, *Multilevel Interrupt*, *Memory Partitioning*, and *Real Time Clock*. All these are dealt with in more detail in their relevant sections.

Other facilities available include the *Bootstrap*, and *Mains Fail-Restart*, which again are dealt with in their relevant sections. The *Switchbank Register* is included in Section 3 on Operation.

Core Store

The core is divided into 'stacks' of 4,096 16 bit words, the minimum capacity of the computer being 4,096 words and the maximum capacity being 65,536 words. A program may directly address any word in the same stack. Words in other stacks may be indirectly addressed.

3. OPERATION

The control panel of the Digico Micro 16V offers an ease of operation by the use of featherlight push buttons, and a unique octal display. In fact it can be operated comfortably one-handed, leaving the other free for operation of other peripherals, etc. Numbers are entered in octal using a shift register system, providing great advantages over systems of entering the numbers on a string of binary switches. For the user's convenience the display is also shown in binary. Any register can be displayed by simply pressing the relevant button.

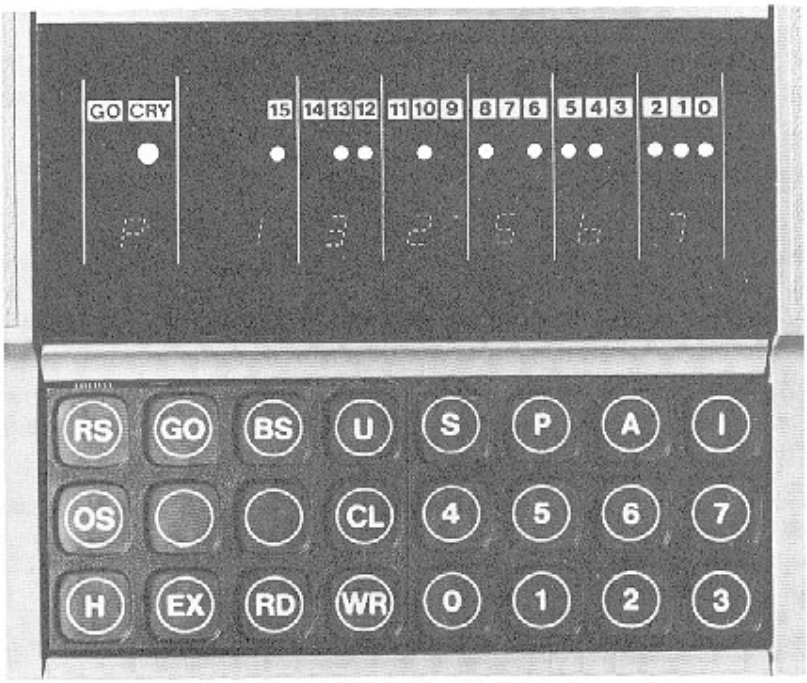


DIAGRAM 3 — CONTROL PANEL

3.1 Controls

P.A.I Depression of one of these keys displays the contents of either the Program Register, Accumulator or the Instruction Register in octal, and the relevant letter.

CL This key clears the contents of the displayed register.

0 to 7 Numeric keys enter numbers into the right hand end of the displayed register and shift the original contents 1 octal digit (3 bit positions) to the left.

S This key causes the display of the Switchbank (S) Register and the letter S. Altering the S register is different from altering the P, A or I registers. As soon as CL or a numeric Key is pressed, the register displayed changes to U (for Update Switchbank Register) and the display will show the number which has been keyed in. The Switchbank register is cleared when the computer is switched on. The contents of the switchbank register are not altered by reading in a program.

U When the U register is displayed, this key causes the updating of the contents of the switchbank register. This particular operation may take place whether the computer is going or stopped.

RS When pressed while the computer is halted, RS causes a peripheral reset. All interrupt statuses are unset further interrupts are inhibited and, if the multilevel priority interrupt option is fitted, the contents of the interrupt detected and levels in progress registers are cleared. The contents of the interrupt enable and switchbank registers are unchanged. When the computer is going RS has no effect.

GO This key starts the program at the current value of the P register.

OS When OS is pressed while the computer is halted, the instruction whose address is given in the P register is loaded into the I register, the contents of bits 11 to 0 of the P register are incremented by 1 the instruction is obeyed and the computer stops.

Note: A Jump instruction may result in a further alteration of the P register.

H If the computer is going, this key halts the program at the end of the current instruction.

EX If the computer is halted this key causes the execution of the instruction in the I register. Unless the particular instruction involves a jump or skip, it does not alter the contents of the P register.

RD If the computer is halted this key reads the contents of the location addressed by the P register into the accumulator and then bits 11 to 0 of the P register are incremented by 1. After obeying RD the I register contains *04 in bits 15 to 12 and the original contents of bits 11 to 0 of the P register in bits 11 to 0. When the accumulator is displayed RD provides a quick way of inspecting successive locations of store.

WR If the computer is halted this key writes the contents of the accumulator to the location addressed by the P register, clears the accumulator and then bits 11 to 0 of the P register are incremented by 1. After obeying WR the I register contains *05 in bits 15 to 12 and the original contents of bits 11 to 0 of the P register in bits 11 to 0. When the accumulator is displayed WR provides a quick way of altering successive locations of store.

BS If the computer is halted and after RS is operated, the hardware bootstrap program may be loaded into store and entered by depressing BS, and holding while GO is pressed. This facility loads a simple 9 word program into locations 0 to 8. The program is automatically entered at location 0. In the bootstrap (see Section 6), XX refers to the channel number of the peripheral used to read an initial program into store. Under normal conditions this would be the Binary Read Program. The channel number is hard wired in the computer and usually refers to a paper tape reader, however it may refer to any device for which the read and busy instructions are 51XX and 74XX respectively e.g. disc.

Note. If a paper tape reader is being used the appropriate tape should be loaded before BS is operated.

GO Lamp

The GO lamp is on while a program is running. It is switched off when the H key is pressed.

POWER OFF/ON LOCK

The Key may be extracted in the OFF or LOCK positions, but not in the ON position. In the LOCK position the operator can only select the register to be displayed. No alteration can be made to the registers as all the other keys are rendered inoperative.

3.2 Reading from Core and Writing to Core

The contents of a core location may be inspected using the RD button. This causes the contents of the core location specified by the P register to be transferred to the accumulator. The P register should be set up prior to pressing RD then the accumulator displayed by pressing A. The twelve least significant bits of P are also incremented by 1 when RD is pressed; thus repeated depression of RD causes the display of subsequent core locations without resetting P.

Core locations may be altered by the WR key, this is similar to the RD key except that the contents of the A register are transferred to the core location specified by the P register, and A is zeroised.

3.3 Examples of the Use of the Control Panel

1. To put the number *430 into the P register (note: the GO lamp must be off, by pressing H if necessary). The following sequence of buttons are pushed.

   P ; CL ; 4 ; 3 ; 0

2. To read the contents of locations *4146 to *4151 (note: again the GO lamp must be off). The following sequence of buttons are pressed.

   P ; CL ; 4 ; 1 ; 4 ; 6
   A ;
   RD ; (The contents of *4146 are displayed
   RD ; (The contents of *4147 are displayed
   RD ; (The contents of *4150 are displayed
   RD ; (The contents of *4151 are displayed.

3. To enter *1350 and *7 into locations *2133 and *2134 (note: again the GO lamp must be off). The following sequence of buttons are pressed.

```
P ; CL ; 2 ; 1 ; 3 ; 3 ;
A ; CL ; 1 ; 3 ; 5 ; 0 ;
WR ;                        (The contents of *2133 is set to *1350 the
                            (display is cleared and P is incremented.

7 ;
WR ;                        (The contents of *2134 is set to *7, the display
                            (is cleared and P is incremented.
```

4. INSTRUCTION FORMAT

Instructions used in the Micro 16V can be of two types, firstly instructions which contain store addresses and secondly ones that do not. There are three types of addressing instructions, direct, indirect and jump instructions, which cause a transfer of control. Non-addressing instructions can likewise be divided into shift instructions, input and output instructions, and a number of miscellaneous instructions. All instructions employed in a Micro 16V have a three letter Assembler mnemonic, such as GET, STO, etc.

4.1 Addressing Instructions

These instructions contain an operation code in bits 15 to 12 (the 4 most significant bits) and an address in bits 11 to 0 (the 12 least significant bits).

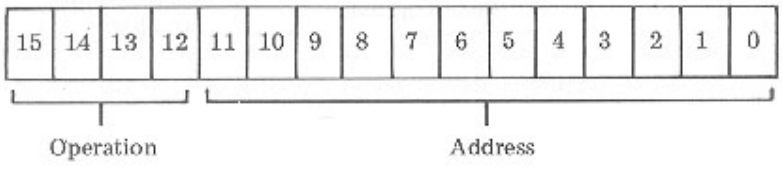


DIAGRAM 4 — FORMAT OF ADDRESSING INSTRUCTIONS

This restricts the address to the range *0 to *7777 or in decimal 0 to 4095, hence only store locations within the current stack can be directly addressed. To address a location in another stack an indirect addressing instruction is used, such as GTI, STI, etc. (see Section 5). This is a double-address operation, one address (12 bits) being in the instruction, and the other address (16 bits) being in the same stack as the instruction. The operation code is then associated with the 16 bit address which can point to any location within 64K.

4.2 Non-Addressing Instructions

Instructions for register manipulation do not require an address. Bits 15 to 12 of each instruction are all zeros and bits 11 to 0 contain the function code.

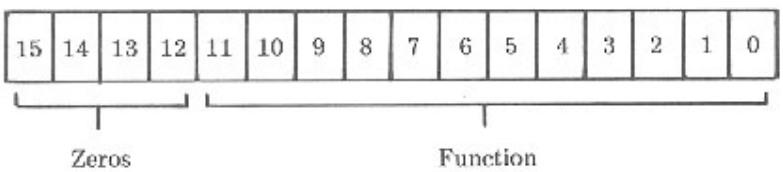


DIAGRAM 5 — NON-ADDRESSING INSTRUCTIONS

Shift Instructions

The Micro 16V has 6 shift instructions. These shift the accumulator, either left or right, by between 0 and 16 bit positions. The arrangement of the bits of the instruction are as shown:-

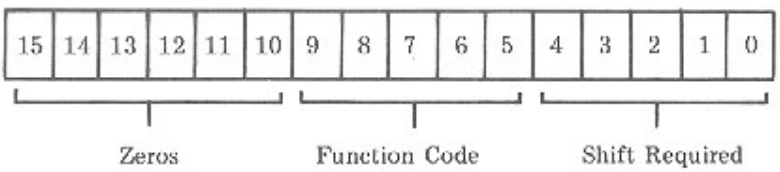


DIAGRAM 6 — SHIFT INSTRUCTIONS

4.3 Input/Output Instructions

These instructions are employed for controlling any peripheral connected to the computer. Each instruction contains the channel number of the peripheral and code commanding the function of the peripheral. Input and output instructions have the function code in bits 11 to 6 and the peripheral channel in bits 5 to 0. The function codes are in the range *40 to *77, i.e. bit 11 is a 1. Bits 15 to 12 are always zero.

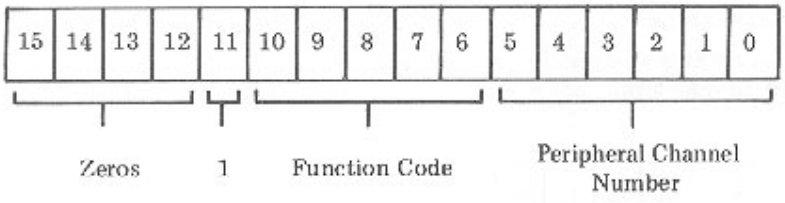


DIAGRAM 7 — INPUT/OUTPUT INSTRUCTIONS

5. INSTRUCTION REPERTOIRE OF MICRO 16V

The instructions available for use with the Micro 16V can be conveniently divided into four sections. These are direct and indirect store addressing instructions; transfer of control instructions (jump instructions); non-addressing instructions. For practicability the final section is sub-divided into shift instructions; miscellaneous instructions; input and output instructions.

5.1 Direct Store Addressing Instructions

Copy Store into Accumulator — GET

Code *04 NNNN Timing 2.15 μS

Function: Clear the accumulator and load the word whose address is given in the instruction. This instruction does not affect the carry register.

DIAGRAM 8 — GET

Copy Accumulator into Store — STO

Code *05 NNNN Timing 2.15 μS

Function: Store the accumulator in the word whose address is given in the instruction. The carry register is neither used nor affected.

DIAGRAM 9 — STO

Addition — ADD

Code *14 NNNN Timing 2.15 μS

Function: To the accumulator add the word whose address is given in the instruction. If the length of sum exceeds 16 bits set the carry register.

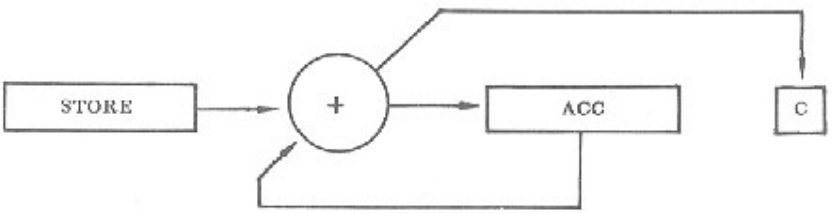


DIAGRAM 10 — ADD

Subtraction — SUB

Code *15 NNNN Timing 2.15 μS

Function: Subtract the word whose address is given in the instruction from the accumulator. This is achieved by the addition of the twos complement of the contents in store to the accumulator.

The carry register is set if the result of the addition $(A + (\overline{S} + 1))$ is greater than *177777.

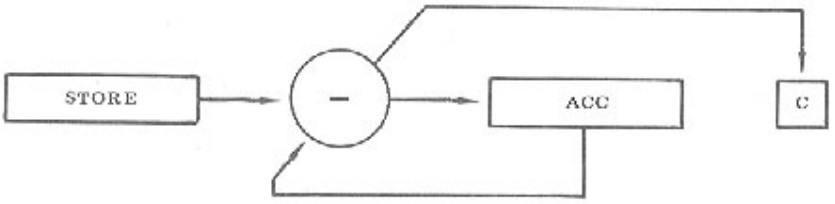


DIAGRAM 11 — SUB

Logical AND AND

Code *10 NNNN Timing 2. 3 μS

Function: A logical AND is performed on the accumulator and the word whose address is given in the instruction. The result is placed in the accumulator. The contents of core store are unchanged.

The result of a one bit AND is as follows:-

0 & 0 = 0 1 & 1 = 1

0 & 1 = 0 1 & 0 = 0

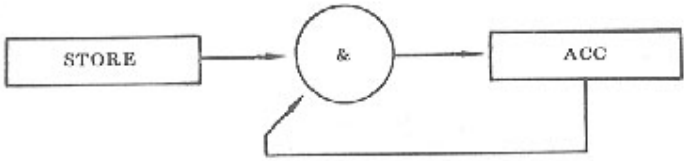


DIAGRAM 12 — AND

Increment Store and Test INC

Code *01 NNNN Timing 2.3 μS

Function: One is added to the word whose address is given in the instruction. If the result is zero then the next instruction in sequence is skipped.

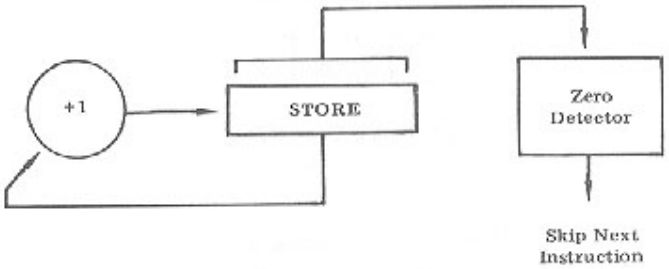


DIAGRAM 13 — INC

Decrement Store and Test DEC

Code *11 NNNN Timing 2.3 μS

Function: One is subtracted from the word whose address is given in the instruction. If the result is zero, then the next instruction in sequence is skipped.

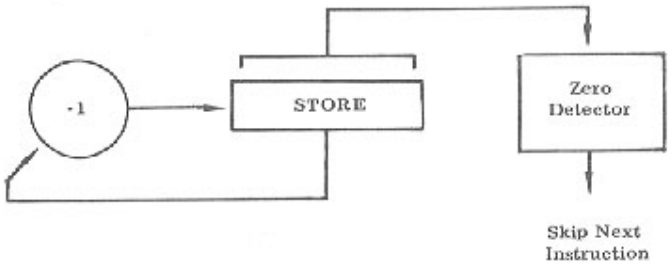


DIAGRAM 14 — DEC

5.2 Indirect Store Addressing Instructions

The effect of an indirect addressing instruction is very similar to its direct counterpart. The only difference is that the address in the instruction contains the address of the required store location, whereas in the direct case the address given in the instruction is the required store location. Hence in the indirect case any stack can be addressed, whereas in the direct case only the current stack can be addressed.

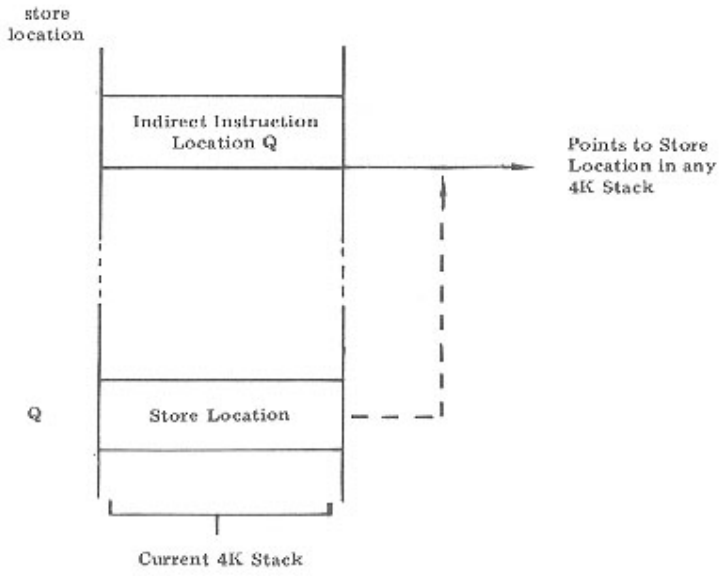


DIAGRAM 15 — INDIRECT ADDRESSING INSTRUCTIONS

Indirect Get GTI

Code *06 NNNN Timing 3.15 μS

Function: Clear the accumulator and load the word whose address is given in the word whose address is given in the instruction.

Indirect Store STI

Code *07 NNNN Timing 3.15 μS

Function: Store the accumulator in the word whose address is given in the word whose address is in the instruction.

Indirect Add ADI

Code *16 NNNN Timing 3.15 μS

Function: To the accumulator add the word whose address is given in the word whose address is given in the instruction.

5.3 Instructions for Transfer of Control (Jumps)

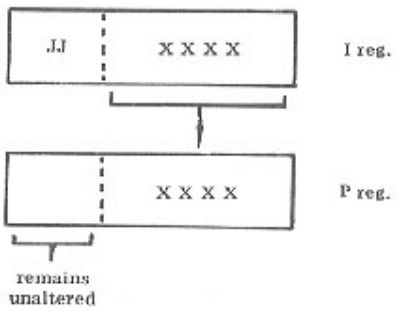


DIAGRAM 16 — JUMP INSTRUCTIONS (JPU, JPZ, JPN, JPS)

Unconditional Jump JPU

Code *02 NNNN Timing 1.15 μS

Function: Program control is transferred unconditionally within the same stack to the address indicated by bits 11 to 0 of the instruction.

Jump if Zero JPZ

Code *03 NNNN Timing 1.15 μS

Function: If the accumulator is zero, program control is transferred within the same stack to the address indicated by bits 11 to 0 of the instruction. Otherwise program control is transferred to the next instruction in sequence.

Jump if Negative JPN

Code *13 NNNN Timing 1.15 μS

Function: If the accumulator is negative (i.e. if bit 15 is a 1), program control is transferred within the same stack to the address indicated by bits 11 to 0 of the instruction. Otherwise (i.e. if bit 15 is a 0), program control is transferred to the next instruction in sequence.

Jump to a Subroutine JPS

Code *12 NNNN Timing 2.3 μS

Function: The current content of the P Register is incremented by 1 and stored in the word in the current stack which is indicated by bits 11 to 0 of the instruction. This word is later used as a link to return program control to the main sequence.

Program control is then transferred to NNNN + 1. An LKJ instruction, immediately preceding the link, is normally used to return program control to the instruction following the JPS.

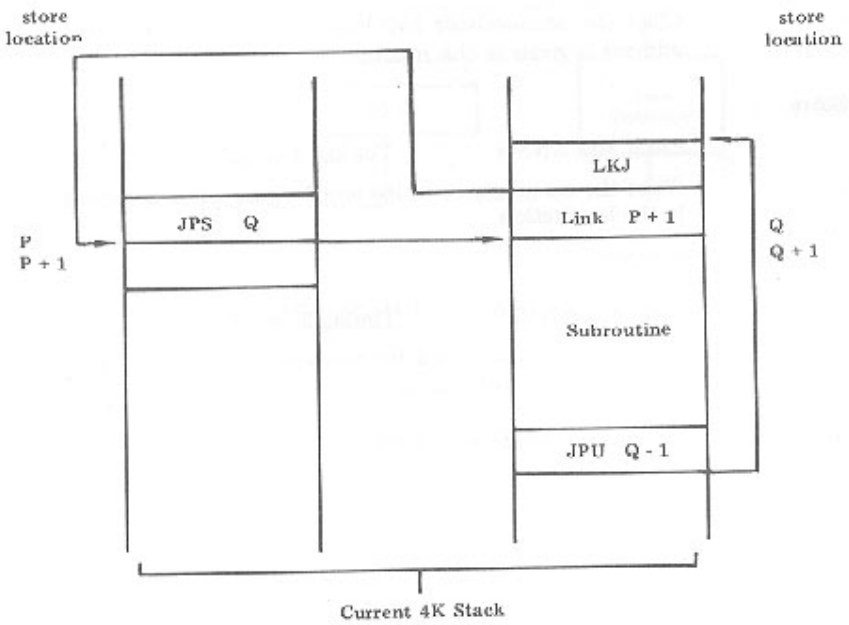


DIAGRAM 17 — JPS

Indirect Jump to a Subroutine JSI

Code *17 NNNN Timing 3.3 μS

Function: The current content of the P Register is incremented by 1 and stored in the word (link) whose 16 bit address is contained by the word which is indicated by bits 11 to 0 of the instruction. Program control is then transferred to the instruction immediately following the link.

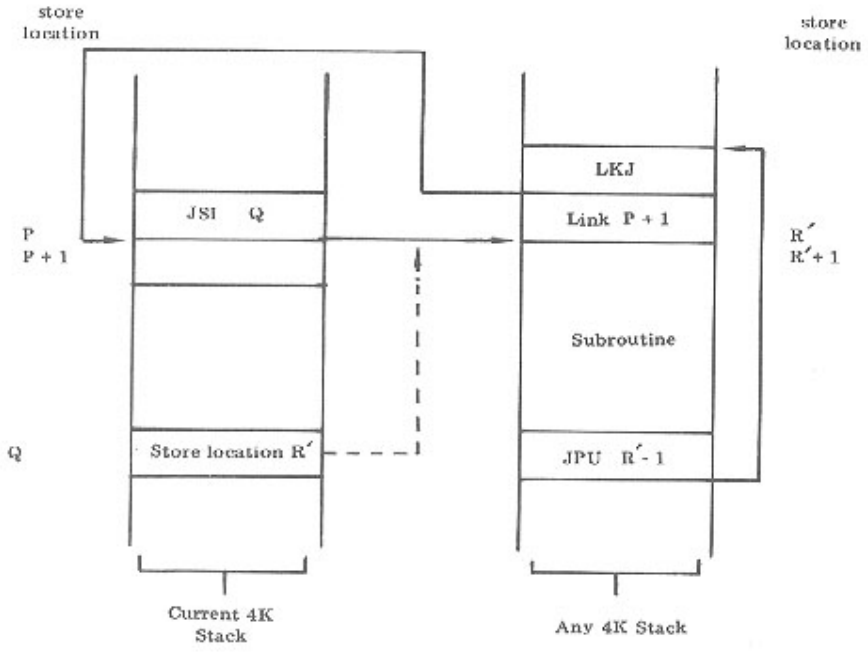


DIAGRAM 18 — JSI

Link Jump LKJ

Code *00 0400 Timing 2.3 μS

Function: Program control is transferred to the address stored in the word immediately following the LKJ instruction. This instruction is used to exit from subroutines, or as a jump between stacks.

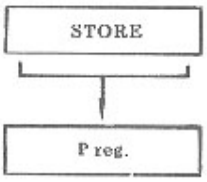


DIAGRAM 19 — LKJ

5.4 Non-Addressing Instructions

Shift Instructions

Shift Right SHR

Code *00 1000 + XX Timing (1.5 + xx/6) μS max. = 4 μS

Function: Shift the contents of the accumulator XX positions to the right. Zeros are propagated at the most significant end while bits from the least significant end are lost.

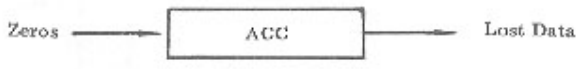


DIAGRAM 20 — SHR

Shift Left SHL

Code *00 1300 + XX Timing (1.5 + xx/6) μS max. = 4 μS

Function: Shift the contents of the accumulator XX positions to the left. Zeros are propagated at the least significant end while bits from the most significant end are lost.

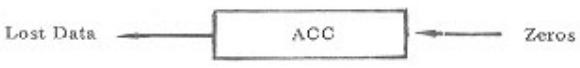


DIAGRAM 21 — SHL

Arithmetic Shift Right — RSS

Code *00 1200 + XX Timing (1.5 + xx/6) μS max. = 4 μS

Function: Shift the contents of the accumulator XX positions to the right. The sign bit (most significant bit) is propagated at the most significant end while bits from the least significant end are lost.

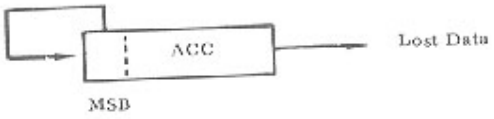


DIAGRAM 22 — RSS

Right Shift Carry — RSC

Code *00 1400 + XX Timing (1.5 + xx/6) μS max. = 4 μS

Function: Shift the contents of the accumulator XX positions to the right. The carry bit is propagated at the most significant end while bits from the least significant end are lost. The carry register is unaltered.

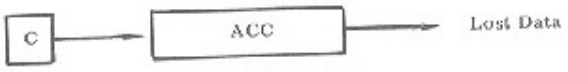


DIAGRAM 23 — RSC

Circulate Right — CIR

Code *001100 + XX Timing (1.5 + xx/6) μS max. = 4 μS

Function: Circulate the contents of the accumulator so that bits overflowing at the least significant end re-enter at the most significant end. No bits are lost.

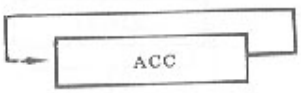


DIAGRAM 24 — CIR

Right Circulate Invert — RCI

Code *00 1040 + XX Timing (1.5 + xx/6) μS max. = 4μS

Function: Circulate data through the accumulator as in CIR but invert the data before it enters at the most significant end, i.e. zeros and ones leaving the accumulator at the least significant end enter the most significant end as ones and zeros respectively.

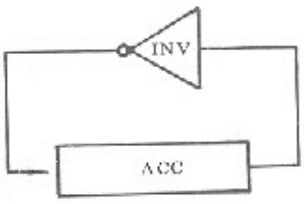


DIAGRAM 25 — RCI

Miscellaneous Instructions

Do Nothing Instruction — NOT

Code *00 0000 Timing 1.15 μS

Function: This instruction does nothing apart from the normal updating of the P Register. It may be useful in program modification or patching of programs.

Halt Instruction HLT

Code *00 0040 Timing 1.15 μS

Function: This instruction stops the program. Operator intervention is required to restart the program.

Note: This is a restricted instruction, see Section 8.

Read Carry Instruction CRY

Code *00 0001 Timing 1.15 μS

Function: The accumulator is cleared and the content is set at 0 or 1 depending on the state of the carry register. The carry register remains unchanged.

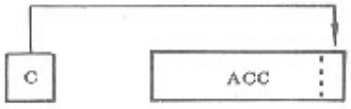


DIAGRAM 26 — CRY

Negate Carry Instruction NCA

Code *00 1420 Timing 4 μS

Function: The accumulator is cleared and set to all zeros or all ones depending on whether the carry register is 0 or 1 respectively. The carry register remains unchanged.

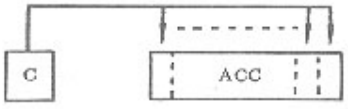


DIAGRAM 27 — NCA

Set Carry Register CRS

Code *00 0200 Timing 1.15 μS

Function: Set the carry register to one.

Unset Carry Register CRU

Code *00 0100 Timing 1.15 μS

Function: Set the carry register to zero.

Zeroise Accumulator CLA

Code *00 1020 Timing 4 μS

Function: Set the accumulator to zero.

Ones Complement Accumulator OCA

Code *00 1060 Timing 4 μS

Function: An exclusive OR instruction is executed between 1 and the accumulator. i.e. all bits in the accumulator are inverted. Zeros become ones, and ones become zeros.

Twos Complement Accumulator TCA

Code *00 1061 Timing 1.3 μS

Function: The content of the accumulator is subtracted from 0 and the answer is placed in the accumulator. The twos complement is, in fact, the ones complement plus one.

Get Switchbank Register SWB

Code *00 0002 Timing 1.3 μS

Function: Clear the accumulator and load the contents of the switchbank register into it.

Input and Output Instructions

Input and output instructions on the Micro 16V take either 9.3 or 4.3 microseconds depending on the channel number allocated to the peripheral. On channel numbers *00 to *37 the time is 9.3 microseconds while on channel numbers *40 to *77 the time is 4.3 microseconds. These times are exclusive of the time required to transfer data between the computer and its peripherals.

Note: Some of these instructions are restricted, see Section 8.

For examples showing the use of the instructions in this section see Appendix VI.

6. BOOTSTRAP

The bootstrap is a very simple program used to read in a more comprehensive read program capable of checking for errors. This error checking program is referred to as the Binary Read Program. In loading the Binary Read program the bootstrap uses an intermediate program located in words *35 through *47. Once used this intermediate program becomes redundant and if the MLI is used will, in general, be overwritten.

Actual Bootstrap Instructions

Location	0	00	74XX	SRB	XX
	1	02	0000	JPU	0
	2	00	51XX	RST	XX
	3	03	0000	JPZ	0
	4	00	1104	CIR	4
	5	05	0140	STO	140
	6	11	0005	DEC	5
	7	00	0400	LKJ	
	10	00	0000	0	

The sequence of instructions is automatically loaded into store, when bootstrap is activated, and entered at word 0.

The channel number XX is fixed by hardware. If it is required to use a different channel the bootstrap may be loaded from the keys. The P Register should be set to zero and the instructions listed above entered, in turn, to the accumulator and stored using the WR key. These are entered at zero with the accumulator at zero.

Note: Bootstrap uses locations *0 - *140. On completion Binary Read occupies *50 - *127.

7. INTERRUPTS

When a peripheral requires attention from the Micro 16V computer, it interrupts the central processor.

With the standard interrupt facility a common 'interrupt request' line is pulled down to 0 volts and, provided that interrupts are permitted, a program interrupt occurs as soon as the current program instruction has been completed.

The multilevel priority interrupt option may be fitted in order to achieve maximum use of the hardware and introduce increased flexibility into a given system of software and hardware. This provides 16 priority interrupt levels to which peripherals can be connected in the order of priority required.

7.1 Standard Interrupt Facility

An interrupt takes two memory cycles to execute and inhibits all other interrupts except power fail. Program control is transferred to word 1 of stack 0 and the link address is deposited in word 0 of stack 0. After an interrupt the program should service the interrupting peripheral, permit interrupts again and return to the main program via the link address deposited in word 0.

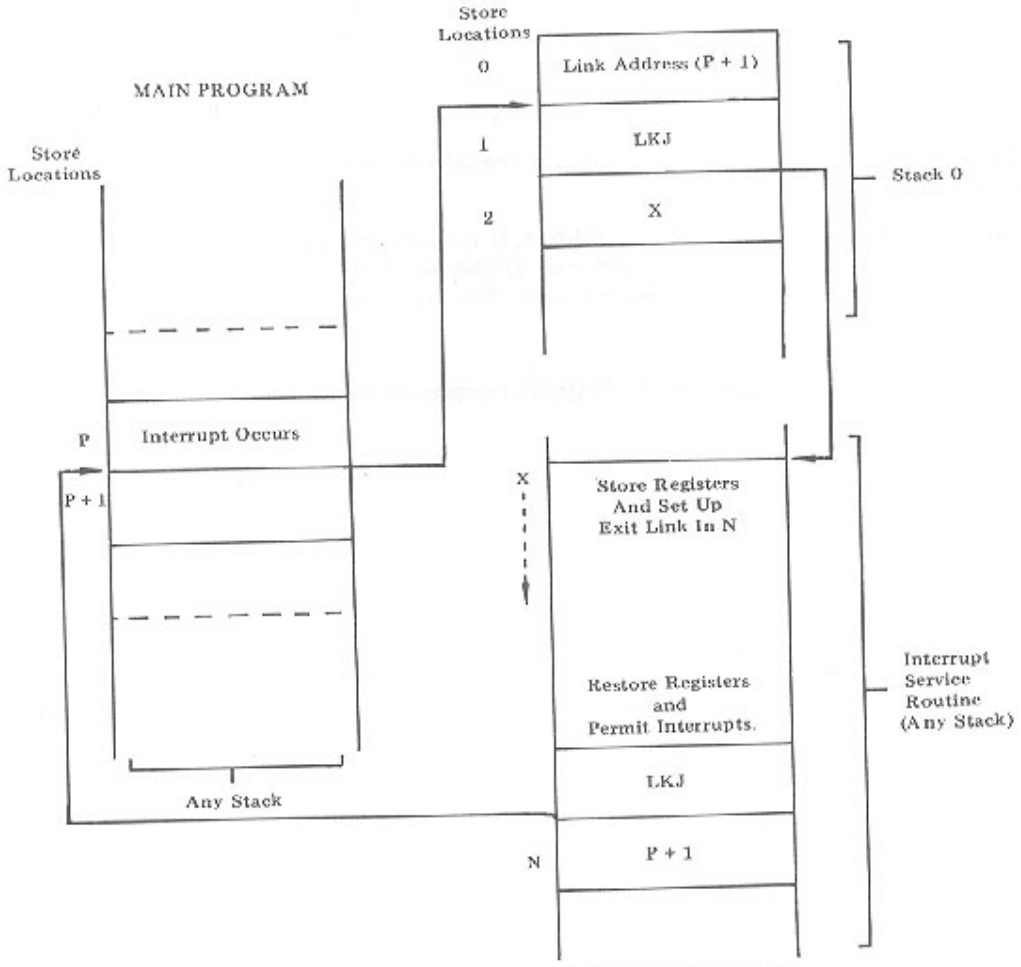


DIAGRAM 28 — STANDARD INTERRUPT

An interrupt servicing routine must find out which peripheral has interrupted and take appropriate action. This action must include the unsetting of the peripherals interrupt status. Any registers, such as the accumulator or carry, which are used by the interrupt routine, must be stored at the start of the routine and be restored at the end of the routine.

Inhibition and Permission of Interrupts

Two program instructions are available which respectively inhibit and permit interrupts.

Forbid Interrupts FIN

Code *00 0020 Timing 1.15 μS

Function: Program interrupts due to all interrupt requests except power fail are inhibited. The effect of this instruction is automatic when a normal program interrupt occurs.

Permit Interrupts PIN

Code *00 0030 Timing 1.15 μS

Function: This instruction permits the next LKJ instruction in the program sequence to enable the program interrupt circuits.

A PIN instruction does not necessarily need to be followed immediately by an LKJ instruction. The logic of the computer has been designed in this way to prevent the possibility of a PIN instruction within an interrupt routine allowing an immediate interrupt which would overwrite the link word and thus prevent the return of control to the main program sequence. When the multilevel priority interrupt option is fitted a PIN eventually permits those interrupts which are defined by the interrupt Enable Register.

Interrupts and Their Inhibition

Interrupts are inhibited by :-

a. An interrupt occuring. Once an Interrupt has occurred further interrupts are inhibited until they are again permitted by an LKJ following a PIN. Usually this is at the point when program control is returned to the main program sequence.

b. A FIN instruction;

c. General Reset given by the mains being switched 'on' and 'off';

d. General Reset given via the control panel. When the computer is stopped (either by a HLT instruction in program, or by the operator pressing H on the control panel) and RS is pressed. This also switches off peripheral interrupts and switches off peripheral interrupt statuses.

Interrupt Servicing Routine

The interrupt servicing routine must preserve the contents of any register, such as the accumulator and carry which it uses, and restore the contents before exit. It must find out which peripheral interrupted and take appropriate action including the unsetting of the peripherals interrupt status.

7.2 Multilevel Priority Interrupt Option

Where a Micro 16-V computer has several peripherals attached or where an executive program is required, the multilevel priority option enables considerably more efficient servicing of peripherals. Peripherals whose servicing is most urgent give high level priority interrupts while peripherals whose servicing is less urgent give lower level priority interrupts. During the interrupt service routine for level N interrupts for lower levels are inhibited but interrupts for higher levels are still permitted. The program will not be affected by interrupts on levels lower than the current level of priority. The highest priority is that of the highest level number. To allow an interrupt to occur on a particular level an interrupt enable command must be executed for that level and the permit interrupts command must be executed.

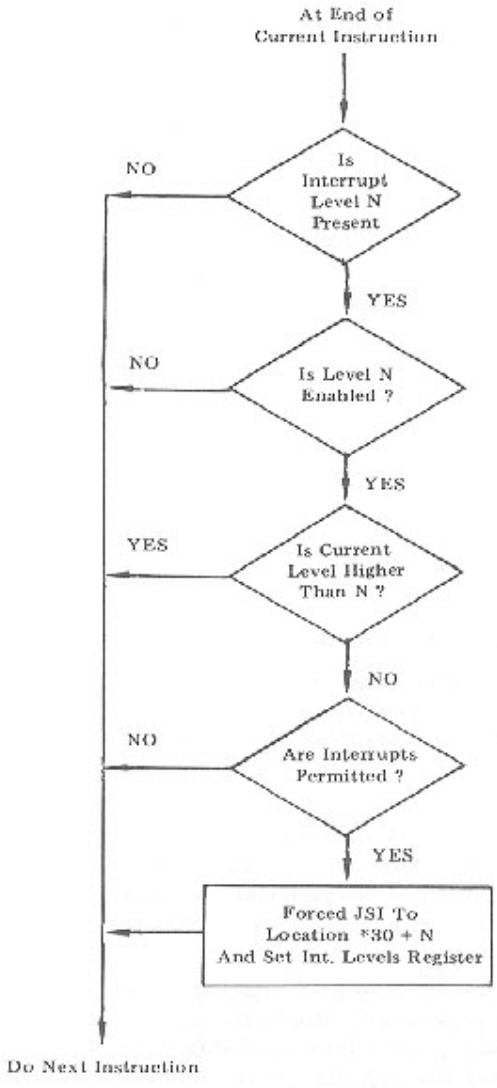


DIAGRAM 29 — FLOW DIAGRAM FOR MULTILEVEL PRIORITY INTERRUPT

The locations of store from *30 to *47 inclusive are reserved as the indirect pointers for the forced JSI instruction for the interrupt. The allocation of addresses is shown below:-

Priority Level	Location	Priority Level	Location
0	*30	8	*40
1	*31	9	*41
2	*32	10	*42
3	*33	11	*43
4	*34	12	*44
5	*35	13	*45
6	*36	14	*46
7	*37	15	*47

The program sets the content of the dedicated locations. Each is used as the indirect store address for the forced JSI instruction which executes JSI *30 + N where N is the Priority Level. From a hardware point of view, the multilevel priority interrupt option provides 16 separate interrupt request lines to which peripherals could be attached to the same priority level, there are usually enough lines to put each peripheral on a different priority level.

It is normal to allocate the highest priority levels to any peripherals where fast interrupt servicing is critical.

Digico will be pleased to advise customers on the best priorities to be allocated to individual peripherals in a particular configuration. Standard priority levels will be allocated to most peripherals.

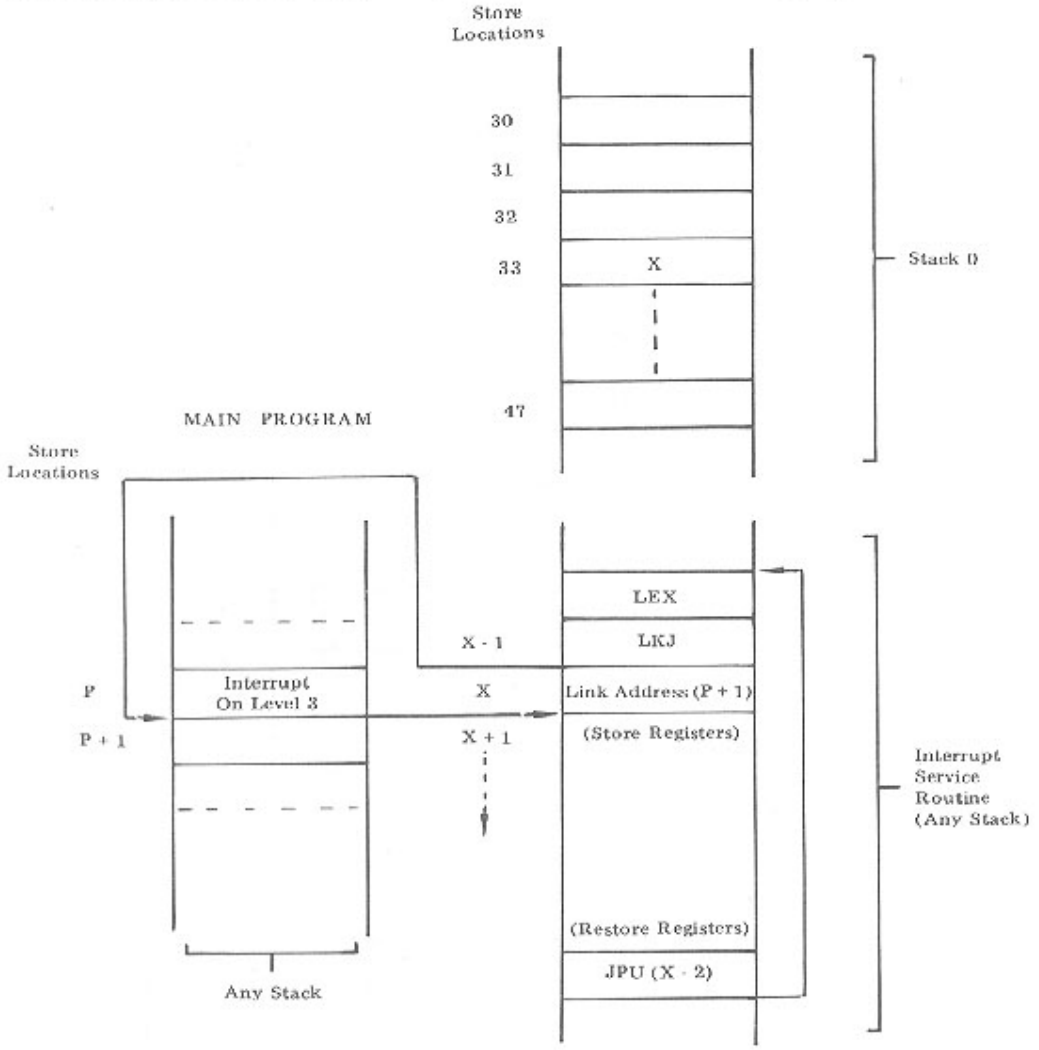


DIAGRAM 30 — EXAMPLE OF MLI FOR A SINGLE INTERRUPT

Additional Hardware Registers for the Multilevel Priority Interrupt Option

Three additional 16 bit hardware registers are provided for control of the interrupt levels. The registers are:-

1. The Interrupt Detected Register.
2. The Interrupt Enable Register.
3. The Interrupt Level Register.

Each register contains one bit for each level of interrupt and 1 and 3 can be accessed by program.

The *Interrupt Detected Register* (IDR) is actually two registers, one for interrupts caused by peripherals and one for interrupts caused by program. These registers indicate which level of interrupt would, if permitted, cause an interrupt.

When a bit is set through a peripheral requiring attention, the appropriate bit is unset when the peripheral no longer requires attention, usually, when it has been serviced within the interrupt service routine for that level. When a bit is set by an SID instruction within program, the appropriate bit is unset by the entry to the interrupt service routine for that level. Bit N is assigned to interrupt level N. General reset will unset all bits in the IDR

The SID and RID instructions are used in connection with IDR.

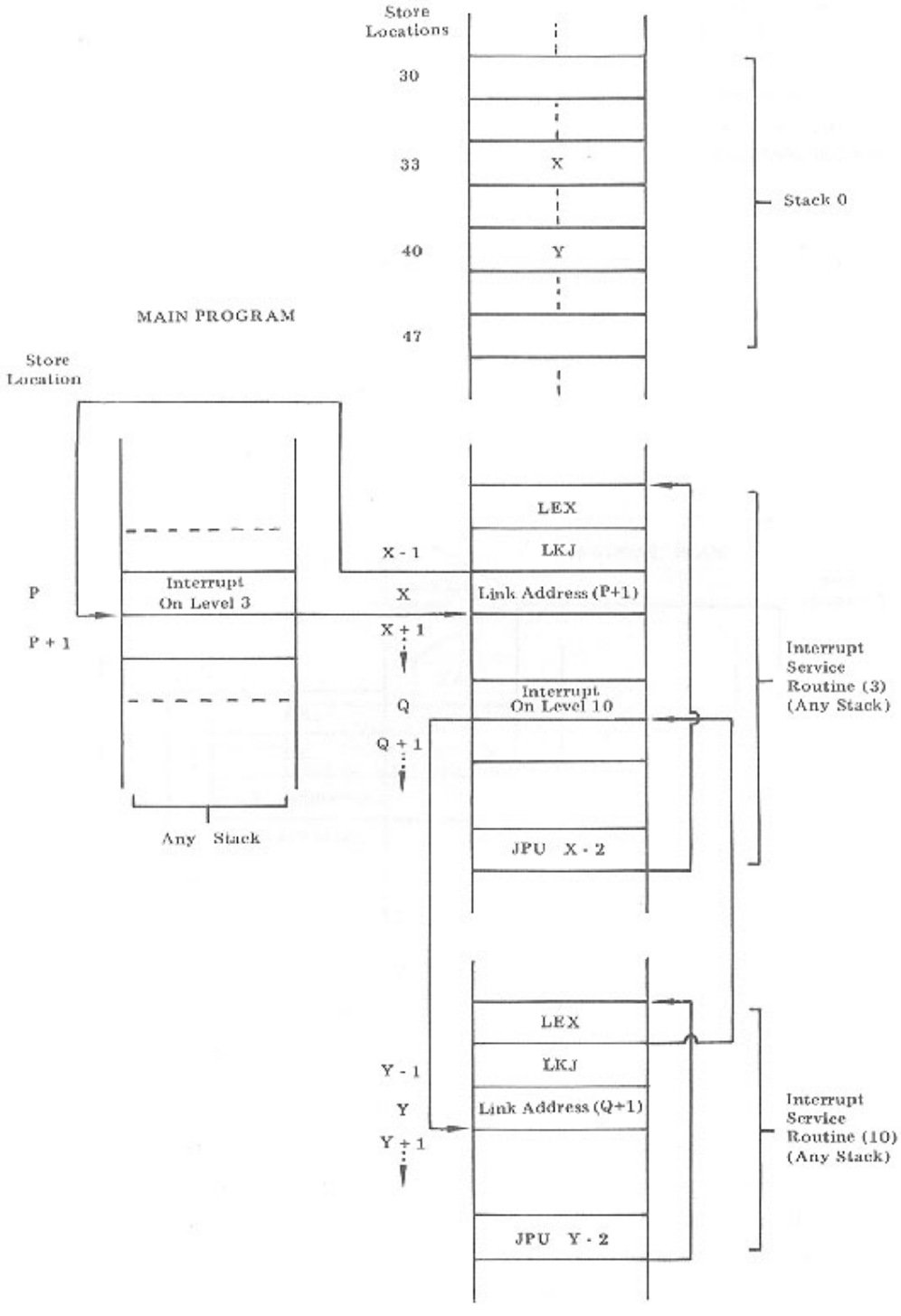


DIAGRAM 31 — EXAMPLE OF MLI FOR A DOUBLE INTERRUPT

The *Interrupt Enable Register*. (IER) only permits interrupts for those levels for which the appropriate bit is set to 1. Bit N is assigned to interrupt level N. General Reset and the FIN instruction inhibit all interrupts but do not alter the IER.

The contents of the IER are set by and SIE instruction.

The *Interrupt Levels Register* is used to indicate the interrupts which have forced entry to the appropriate level of interrupt service routine until exit is made from that level.

The appropriate bit is set automatically when an interrupt service routine is entered and is unset when the level is cleared by an LEX immediately followed by an LKJ to the link for that level.

Bit N is assigned to interrupt level N. General reset will unset all the bits in the levels in progress register.

The RLP instruction gets the contents of the interrupt levels in progress register into the accumulator.

Additional instructions with the multilevel priority interrupt option 5 additional instructions are provided to enable the best use to be made of the multilevel priority interrupt option.

Note:- These instructions are restricted, see Section 8.

Exit Level — LEX

Code *00 0031 Timing 3.8 μS

Function: After the LKJ, which should immediately follow the LEX instruction, the bit for the appropriate level in the Interrupt Levels Register is cleared. All interrupts are inhibited until the completion of the LKJ instruction. Thereafter further interrupts are permitted on the level exited and higher levels. The LEX instruction is normally used at the end of an interrupt service routine.

Set Interrupt Detected Register — SID

Code *00 0032 Timing 3.8 μS

Function: A logical OR is performed between the content of the accumulator and part of the IDR which can be altered by program. The result is placed in the IDR for program interrupts. The content of the accumulator is unchanged. Bit N is assigned to interrupt level N.

Set Interrupt Enable Register — SIE

Code *00 0033 Timing 3.8 μS

Function: The content of the accumulator is read into the Interrupt Enable Register. The content of the accumulator is unchanged.

Read Interrupt Detected Register — RID

Code *00 0036 Timing 3.8 μS

Function: The accumulator is cleared and the contents of the two parts of the IDR are inclusive OR-ed into the accumulator. The contents of the IDR are unchanged.

Read Levels in Progress Register — RLP

Code *00 0037 Timing 3.8 μS

Function: The accumulator is cleared and the content of the interrupt level in progress register is read into the accumulator.

Notes on Programming with MLI. Normally a sub-routine may not be used in two or more interrupt levels or program. The sub-routine should be repeated for each entry level which uses it since, otherwise the sub-routine link is liable to be overwritten. This difficulty may be avoided by certain programming techniques but care must be taken to avoid undue delay in servicing interrupts and, in particular, to avoid the possible overwriting of a vital link.

If the mains restart option is in use it is necessary, after power fail is detected, to store the contents of the interrupt levels, interrupt detected and interrupt enable registers if the program is to proceed since the content of these registers is cleared on restart.

8. MEMORY PARTITION OPTION

This option, in conjunction with the multilevel priority interrupt option, allows the definition of a store area in which specific instructions have restricted execution modes. This enables executive programs to be written to enable the execution of an application program. It is also possible to write an executive program capable of controlling the independent execution of several application programs. The definition of the restricted area is achieved by the setting of two 8 bit limit registers. One register gives the lower limit of the restricted area while the other gives the upper limit of the restricted area.

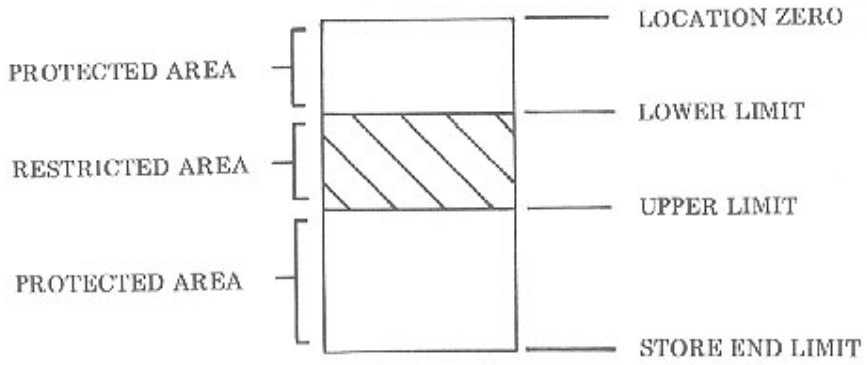


DIAGRAM 32 — DEFINITION OF STORE AREA

The option functions as if it were a peripheral on channel 63 (*77) and is mainly of interest to executive programmers. Executives may be written by user programmers but will usually be written by Digico. With the memory partition option associated interrupts always occur on interrupt priority level 8 (*10). Thus to enable the memory partition option it is necessary to permit interrupts, initially by a PIN, and set bit 8 of the Interrupt Enable Register.

The Option may be used in two modes,

i) two limits mode

ii) single limit mode.

Under a reset condition the upper and lower address limits are set to zero and this represents a disabled state for the option.

In the *two limits mode*, the option is enabled by setting a finite restricted area, with the lower limit less than the upper limit. The option will then detect restricted instructions when:

$$\text{Lower Limit} \leqslant \text{Program Counter} < \text{Upper Limit.}$$

The *single limit mode* is a special case of the two limits mode, where the upper limit is set to zero. The restricted area consists of all stores address $\geqslant$ lower limit, and restricted instructions are detected when:

$$\text{Program Counter} \geqslant \text{Lower Limit}$$

The limits are set from the accumulator employing the following format:

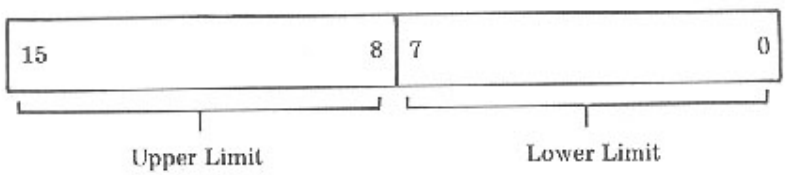


DIAGRAM 33 — FORMAT FOR LIMITS AS SET FROM ACCUMULATOR

The Lower Limit is inclusive, and the upper limit exclusive of the restricted area, giving a minimum finite area of 256 locations.

8.1 Restricted Instructions

For intentional exitting from the restricted area a PEX instruction is executed, this is a restricted NOT which will in general be an interrupt on level 8 (*10). A restricted instruction, when detected in the restricted area will also cause an exit from the area, and will force a NOT instruction and an interrupt or level 8 (*10). The stored link for this interrupt is equal to the address of the restricted instruction responsible. If the service routine lies in the protected area the option is disenabled and is reenabled when the program counter enters the restricted area. The service routine must not be in the restricted area.

Note: Before entering the restricted area the executive or control program must enable interrupts on level 8, otherwise no escape can be made from restricted mode.

The restrictions apply to the following instructions:-

1. Input and output instructions to all peripherals and other instructions using an I/O format (*4000 to *7777).
2. Instructions which directly or indirectly modify interrupt control states (*20 to *37).
3. Instructions which would modify the contents of store locations in protected areas.
4. When their execution would set the program counter across a store limit, the following jump instructions:-

   JPU
   LKJ
   JPZ ┐ when jump condition
   JPN ┘ is true.

5. When in the last word of restricted area is about to increment the P register to transfer program control outside the upper limit or in the last word of stack and about to set the P Register to word 0 of the stack which happens to be outside the limit, all other instructions.
6. Invalid instruction codes which would modify MLI or Halt, these are:-

   * 20 — * 77
   * 120 — * 177
   * 220 — * 277
   * 320 — * 377

7. HLT.

8.2 Instructions to Control the Memory Protect Option

The three instructions which are available with this option all cause an interrupt when executed within the restricted area of store.

Set Program Limits **SPL**

Code *00 4177 Timing 4.3 μS

Function: Set the upper limit of the restricted area to (256 times the number contained in bits 15 to 8 of the accumulator) minus one; set the lower limit of the restricted area to (256 times the number contained in bits 7 to 0 of the accumulator). When bits 15 to 8 of the accumulator contain zero the upper limit will be set to the store end limit.

Example to set the memory limits to *400 and *1000.

```
* 150   04   0200   GET
* 151   00   4177   SPL (Set Memory Limits to *400/*1000.
  :
* 200   00   1001
```

Program Exit **PEX**

Code *00 4277 Timing 4.3 μS

Function: Transfer program control from the program in the restricted area to executive by forcing an interrupt. Any other invalid instruction may also be used.

Skip if Program Interrupt **SPI**

Code *00 4377 Timing 4.3 μS

Function: Skip the next instruction if the memory partition option has not caused a program interrupt. Otherwise unset the program interrupt. This instruction must be obeyed to unset the interrupt.

8.3 The Use of the Memory Partition Option for Executive

The memory partition option, in conjunction with the other options, makes it possible to write executive programs for a wide variety of purposes. Executive programs make it much easier to write user programs but this costs an amount of core store which is occupied by executive.

The following notes deal with functions which are sometimes performed by executive programs.

A Single Programming Executive

In a computer working under the control of a single programming executive there are normally two programs in the computer. One is the executive which is held in the protected area of store and the other is the user program which is held in the restricted area of store. Entry to the executive program, whose basic priority level is 8, is achieved either by a peripheral interrupt or the use of a restricted instruction within the user program, while entry to the user program is usually achieved via executive.

Servicing of Peripherals

Nearly all executives handle the interrupt servicing of peripherals and certain other functions of input and output. Since the priority interrupt addresses are in words *30 to *47 (24 to 39), these are normally within the executive area.

File Handling

File handling, for bulk storage devices such as magnetic tapes and exchangeable discs, may be handled by executive, by user program or by a combination of the two. Buffer areas would normally be within user programs.

Loading and Deletion of User Programs

Executives usually provide standard facilities for the loading and deletion of programs.

Other Extracode Facilities

By using restricted function codes the executive writer can provide executive facilities to perform frequently required functions which are not directly provided by hardware. It is desirable, however, to maintain as much compatibility as possible with other computers of the same type.

Multiprogramming Executives, etc.

In a multiprogramming executive only the executive program or any one user program is functioning at any instant of time. Before entering a particular user program executive must set the restricted area limits to the operational program limits for that program. Before transferring control to a second user program the restricted area limits must be reset to the operational limits for the second program.

The multilevel priority interrupt option makes it possible to write real time executives for appropriate hardware configurations.

9. RESERVED STORE LOCATIONS

Certain store locations at the start of the first stack are reserved. Computers, which are supplied with executive, will have a larger area of store reserved.

Normal Interrupt Entry (Locations *0 to *2)

If the multilevel priority interrupt option is not fitted, these locations are used by programs to transfer program control between the main program and the interrupt service routine. A link address, and LKJ instruction and the address of the interrupt service routine are usually stored in these locations.

Entry Points (Locations *3 to *5)

Standard entry is made to programs via location *3. When required, locations *4 and *5 may be used for additional entry points.

Power Fail and Restart (Locations *6 and *7)

Locations *6 and *7 are reserved for the mains fail and automatic restart routine. Location *6 contains the address of the mains fail routine and location *7 contains the address of the restart routine.

Direct Memory Access (Locations *10 to *27)

If the DMA option is fitted, for each DMA channel one word is allocated for a data address and a second is allocated for a count of data word transfers.

Multilevel Priority Interrupt Addresses (Locations *30 to *47)

When the multilevel priority interrupt option is fitted, interrupt service routine entries are provided for each interrupt priority level. The address for the interrupt service routine for priority level N is contained in location *30 + N.

Octal Location (Stack 0)		
0	Link Address	Normal Interrupt Entry
1	LKJ	
2	Interrupt Address	
3	Jump to 1st Entry Point	Program Entry Points
4	Jump to 2nd Entry Point	
5	Jump to 3rd Entry Point	
6	Power Fail Interrupt Address	Power Fail and Restart Addresses
7	Mains Fail Restart Address	
10	DMA	Data Addresses and Counts of Data Word Transfers
11	Channel 0	
12	DMA	
13	Channel 1	
14	DMA	
15	Channel 2	
16	DMA	
17	Channel 3	
20	DMA	
21	Channel 4	
22	DMA	
23	Channel 5	
24	DMA	
25	Channel 6	
26	DMA	
27	Channel 7	
30	Address for Priority Level 0	Priority Interrupt Service Routines
31	Address for Priority Level 1	
32	Address for Priority Level 2	
33	Address for Priority Level 3	
34	Address for Priority Level 4	
35	Address for Priority Level 5	
36	Address for Priority Level 6	
37	Address for Priority Level 7	
40	Address for Priority Level 8	
41	Address for Priority Level 9	
42	Address for Priority Level 10	
43	Address for Priority Level 11	
44	Address for Priority Level 12	
45	Address for Priority Level 13	
46	Address for Priority Level 14	
47	Address for Priority Level 15	

DIAGRAM 34 — RESERVED STORE LOCATIONS

10. MAINS FAIL — RESTART

To accommodate the eventuality of a power failure the basic CPU card includes a power fail feature. This is always enabled, even when the processor has given a FIN command.

When power fail is detected, an interrupt is forced by executing a JSI *6, which entails a subroutine entry to the location addressed in location 6. This allows at least 500 microseconds before the program must be halted. After this interrupt, any further peripheral interrupts are inhibited.

The basic CPU card also includes auto restart when power is applied, however this is only enabled if:

i) no control panel is fitted

ii) a control panel is fitted and the user has requested it to be enabled.

Auto restart is achieved by forcing the processor to obey a JSI *7, in this way the program starts at a location defined by the contents of location 7.

Note: If restart is enabled then all programs used should have a restart link specified, even if this is just HLT. Otherwise on switch on, entry to an unknown point may be made and the program corrupted.

11. REAL TIME CLOCK OPTION

The real time clock provides crystal controlled interrupts every 10 milliseconds, with a tolerance of $\pm$ 0.1%. It is operated as a peripheral on channel 63 (*77) and it is normally assigned to level 14, but, at customer request it may like any peripheral be assigned any level of priority. Instructions for the operation of the Real Time Clock Option are as follows.

Enable Clock Interrupts ECI

Code *00 4477 Timing 4.3 μS

Function: The timer is started and clock interrupts are enabled. The first interrupt will occur 10 milliseconds after the ECI is executed.

Inhibit Clock Interrupts ICI

Code *00 4577 Timing 4.3 μS

Function: Clock interrupts are inhibited and unset. The timer is reset to 0.

Skip if No Clock Interrupt SCI

Code *00 4677 Timing 4.3 μS

Function: Skip the next instruction if there is no clock interrupt. Otherwise unset the clock interrupt.

Skip if No 'Not Serviced' Interrupt SNS

Code *00 4777 Timing 4.3 μS

Function: Skip the next instruction if no unserviced clock interrupt. Otherwise unset the 'not serviced' clock interrupt.

Notes on Programming for the Real Time Clock Option

General reset inhibits all clock interrupts. 'Not serviced' is set if the previous clock interrupt has not been unset or inhibited when the next interrupt occurs. The normal clock interrupt is unset when 'not serviced' is set. Further clock interrupts are inhibited until 'not serviced' is serviced. For these reasons it is important that the interrupt service routine for the appropriate level contains both an SCI and an SNS instruction.

12. FLOATING POINT UNIT

Since the inception of the Micro 16 series of minicomputers, floating point operations such as floating point add, subtract, divide, etc., have all been executed under software control. The techniques involved can achieve quite speedy timings due to the fast cycle time of the core store. However, to produce more rapid floating point manipulation, Digico are in the process of perfecting a floating point unit for use in the Micro 16V crate.

The Floating Point Unit (FPU) itself occupies two board positions and interfaces with the machine via the Medium Speed Interface and direct store access. The DMA board is unnecessary, but fitting the FPU does not exclude the use of DMA.

Provision is also provided for the recognition of *2000 or *3000 series codes, in order that the MULTIPLY/DIVIDE unit is compatible with the Micro 16 and 16P.

The FPU consists of two parts, firstly a microprogram controller, and secondly a 72 bit arithmetic unit complete with two 72 bit data registers, two 16 bit exponent registers and a 16 bit address register. This will enable the computer to effect an accuracy of 20 + digits. It is expected to work in 2, 3 and 5 word formats, the first two of which can be handled by standard software at present. The five word format would be as follows:

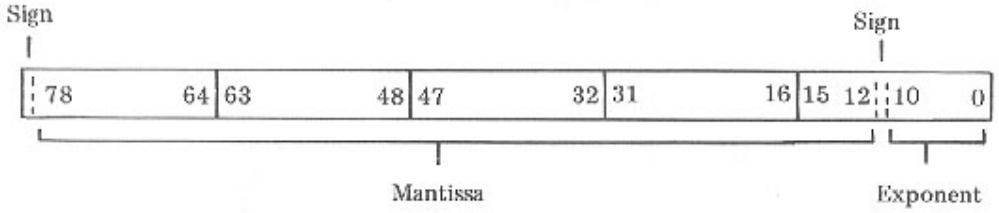


DIAGRAM 35 — 5 WORD FLOATING POINT FORMAT

At the time of going to press, exact timings of the operations cannot be included, however, approximate timings are given in the example below for 3 word floating point functions:

	Software Time	F.P.U. Time
ADD	~ 310 μS	Average 15 μS
MPY	~ 1,000 μS	Average 30 μS

The advantage of the unit can immediately be seen, the timings of the operations can be cut by a factor of at least 20.

Note: This is only preliminary information and may be altered without prior notice.

13. MEDIUM SPEED INTERFACE

To effect control and data transfer between the computer and peripherals the two have to be interfaced. For peripherals such as paper tape readers and punches, which are slow, relative to such units as discs or magnetic tapes, the Medium Speed Interface is used. For the faster units Direct Memory Access is employed, see Section 14.

Data transfer is achieved under program control, and takes place on two 16-bit parallel data highways; 16 data input bits are used to set the accumulator, and 16 data output bits, operating through driver gates, carry the contents of the accumulator to the peripheral. This transfer can take place at speeds of up to 80 thousand characters per second.

The instruction format is discussed in Section 4 on Input and Output Instructions, however, it should be noted that only the 11 least significant bits are actually transferred across the Data Highway.

The interface can drive up to 64 peripherals, on parallel data and control highways. Internal peripherals, which include options such as Real Time Clock and Memory Partition, employ the 156 Way connector inside the Micro 16V crate, external peripherals, the two 44 way connectors. The Channel Address and I/0 Instruction Lines are connected to external peripherals via an output bus driver, and the Data Output Lines may be connected to up to 30 external peripherals each sinking 1.6mA.

Since all data and control highways operate in parallel and the fact that each peripheral has its own six bit channel number, addition of further peripherals is facilitated.

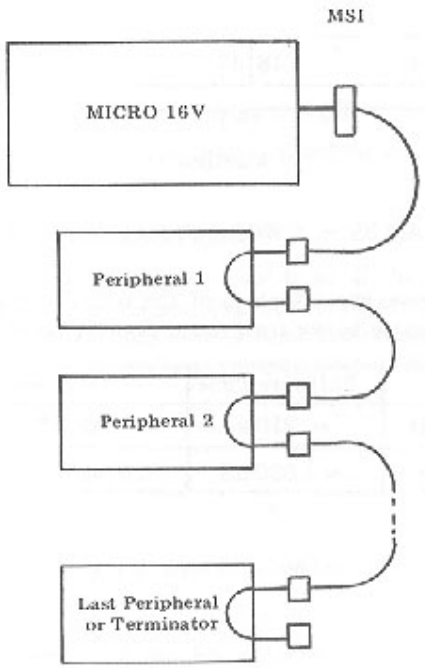


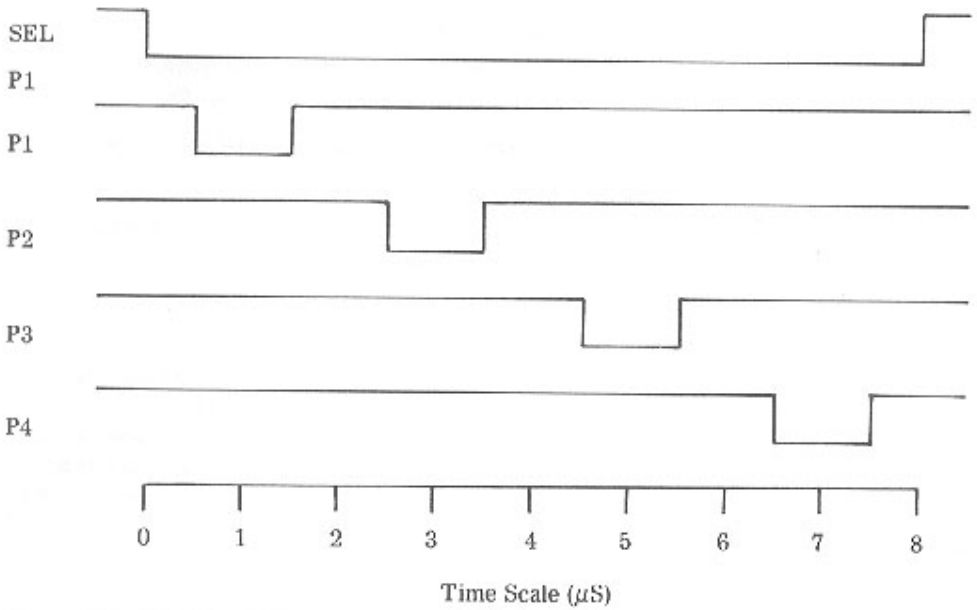
DIAGRAM 36 — MEDIUM SPEED PERIPHERAL CONNECTION

Timing

During an input/output instruction, pulses are sent across the interface from the computer, as shown in the diagram. Data lines, instruction bits and timing pulses are ignored except during the presence of a SELECT pulse, and it is essential that no irreversible action is taken until the front endge of P1.

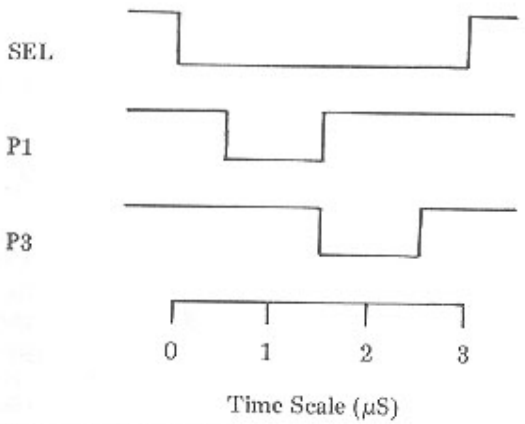
A peripheral is only allowed to use input lines to the computer during the SELECT pulse, and inputs to the computer should be at least 1 microsecond.

The standard interface enables data to be transferred at two possible rates depending on the address allocated to the various peripherals. Peripheral transfers on channels *00 to *37 take approximately 8 microseconds and peripheral transfers on channels *40 to *47 take approximately 3 microseconds. In both cases the minimum gap between transfers in half a microsecond.



None of P1, P2, P3 and P4 overlap.

DIAGRAM 37 — TIMING PULSES FOR CHANNELS *00 to *37



P1 and P3 may overlap

DIAGRAM 38 — TIMING PULSES FOR CHANNELS *40 to *77

MEDIUM SPEED INTERFACE CONNECTIONS

Note: P - represents a signal between +3.0V and +5.25V when present
and a signal between -0.25V and +0.5V when absent
N - represents a signal between -0.25V and +0.5V when present
and a signal between +3.0V and +5.25V when absent

Micro 16V Crate 156-Way	External 2 x 44 Way Connector	Signal Level	Signal Name	Function
A1				— Logic 0V
A20				— Switch on Reset
A39				— Logic 0V
A40	[1B1]			+5V Power Connection Max. load per board = 6A. Max. load per crate = 40A. } M16V only
A47	2B17	N	IL15	Interrupt Priority Levels (Optional) 16 lines are allocated to the 16 interrupt levels used in conjunction with the multilevel priority interrupt option The signals can be generated by the peripheral interfaces at any time.
A48	2B16	N	IL14	
A49	2B15	N	IL13	
A50	2B14	N	IL12	
A51	2B13	N	IL11	
A52	2B12	N	IL10	
A53	2B11	N	IL9	
	2B10	N	IL8	
	2B9	N	IL7	
	2B8	N	IL6	
	2B7	N	IL5	
	2B6	N	IL4	
	2B5	N	IL3	
	2B4	N	IL2	
	2B3	N	IL1	
	2B2	N	IL0	
A54		N	INT	Interrupt Source This line enables a peripheral to inform the central processor that it requires service. The signal can be generated by the peripheral interface at any time.
A56	1B19	N	SKI	Skip Next Instruction This control line enables a peripheral to initiate a skip function in the program. The pulse has a minimum duration of 1 microsecond and is gated by P1 at the peripheral interface.
A57	1B18	N	CLA	Clear Accumulator Control Line This control line enables peripherals to initiate clearance of the accumulator prior to the transmission of data.
A58	2A17	N	P4	Timing Select Pulse Lines SEL and P1 to P4 5 pulse lines are used to propagate timing pulses to peripheral interface circuits in order to synchronise transfer of data with the central processor operation. Output bus drivers are employed at the computer end.
A59	2A16	N	P3	
A60	2A15	N	P2	
A61	2A14	N	P1	
A62	2A13	N	SEL	

A63	2A2	P	I0	Channel Address Lines: 6 channel address lines run from bits 0 to 5 of the instruction register.
A64	2A3	P	I1	
A65	2A4	P	I2	
A66	2A5	P	I3	
A67	2A6	P	I4	
A68	2A7	P	I5	
A69	2A8	P	I6	I/0 Instruction Lines: 5 input/output instruction lines run from bits 6 to 10 of the instruction register.
A70	2A9	P	I7	
A71	2A10	P	I8	
A72	2A11	P	I9	
A73	2A12	P	I10	
A74	[2A19]			-6V Power Connection Max. load per board 2.5A– M16V only
A75	[2A20]			+12V Power Connection Max. load per board 2.5A– M16V only
A77	[2A21]			+24V Power Connection Max. load per board 1A. – M16V only
A78				— Logic 0V
B1				— Logic 0V
B39				— Logic 0V
B40	[1B1]			+5V Power Connection Max. load per board 6A. Total load per crate 40A. M16V only
B44	1A2	P	AO1	Data Output Lines: 16 data output lines are used for transmitting information from the Micro 16-V Computer accumulator to the I/0 peripheral buffer registers.
B45	1A3	P	AO2	
B46	1A4	P	AO3	
B47	1A5	P	AO4	
B48	1A6	P	AO5	
B49	1A7	P	AO6	
B50	1A8	P	AO7	
B51	1A9	P	AO8	
B52	1A10	P	AO9	
B53	1A11	P	AO10	
B54	1A12	P	AO11	
B55	1A13	P	AO12	
B56	1A14	P	AO13	
B57	1A15	P	AO14	
B58	1A16	P	AO15	
B59	1A17	P	AO16	
B60	1B2	N	AI0	Data Input Lines: 16 data input lines are used for transmitting information from the I/0 peripheral buffer registers to the Micro 16-V computer accumulator. Each bus line is terminated at the central processor end in a 910 ohm pull up resistor +5V. A typical drive is a DTL 944/958 type gate.
B61	1B3	N	AI1	
B62	1B4	N	AI2	
B63	1B5	N	AI3	
B64	1B6	N	AI4	
B65	1B7	N	AI5	
B66	1B8	N	AI6	
B67	1B9	N	AI7	
B68	1B10	N	AI8	
B69	1B11	N	AI9	
B70	1B12	N	AI10	
B71	1B13	N	AI11	
B72	1B14	N	AI12	
B73	1B15	N	AI13	
B74	1B16	N	AI14	
B75	1B17	N	AI15	
B77				-24V Power Connection Max. load per board 1A — M16V only
B78				— Logic 0V
	1B21			— FRAME
	2A22			— Logic 0V
	2B21			— FRAME
	2B22			— Logic 0V

Notes on Interface Connections for Internal Peripheral Slots

1. There is a polarising/locating key in position A76, B76 of the 156 way connector. In the M16V crate the peripheral positions are 0 to 7. These are on 12.7mm (0.5 inch) spacing. Any components should be on the A side and must be less than 8.9mm (0.35 inch) for single spacing.

2. Interrupt level 8 is allocated to the memory partition option. The lower levels 7 to 0 will probably not be used for peripherals. The peripheral cards can be directly linked for interrupt levels 15 to 9. For other levels a link is required on the backplane from INT SOURCE N to the appropriate interrupt level on the options card i.e.

Level	0	1	2	3	4	5	6	7	8
Pin (Position 11)	A62	A61	A60	A59	A58	A57	A56	A55	A54

3. The SWITCH ON RESET condition in external peripherals must come direct from its own power supply. The RESET condition across the interface is given by P1N and P2N both low simultaneously.

4. External peripherals (using the 2 x 44 connectors) should be connected by separate power wiring. These pins have been allocated for use in special cases, and are therefore in parenthesis. In these cases:

   a. The peripheral power must be joined to these pins. In standard peripherals there is no connection.

   b. The cable from M16V options board will need extra wires and the options board will need power links fitted.

5. Logic levels are standard TTL levels, 0 to 0.7 volts being logic '0' and 3.2 to 5 volts being logic '1'. The loading on any input signal must be less than 5 lower power loads (.8mA) and the maximum wire/track length is 75mm (3 inches).

14. DIRECT MEMORY ACCESS (HESITATE)

When faster peripheral units are to be used in conjunction with the Micro 16V, such as the Disc, Magnetic tape and Drums, it is desirable to ensure that the associated data is transferred at high speed. To accommodate this the Direct Memory Access (DMA) option is available. DMA occupies one board position in the Micro 16V crate, and will allow up to eight fast peripherals to be multiplexed.

DMA is a 16 bit parallel interface which can transfer data at a total word rate of up to 857 thousand words per second. This option requires an additional logic board and enables the seven internal peripheral slots to use DMA. For external peripherals an extender cable can be employed, but this must not exceed 10 metres (30 feet).

In a similar manner peripherals connected for data transfer to the MSI, fast peripherals can be connected to DMA interface via two 44 way connectors. Control is still dictated by the MSI and hence the connection will be something like this:

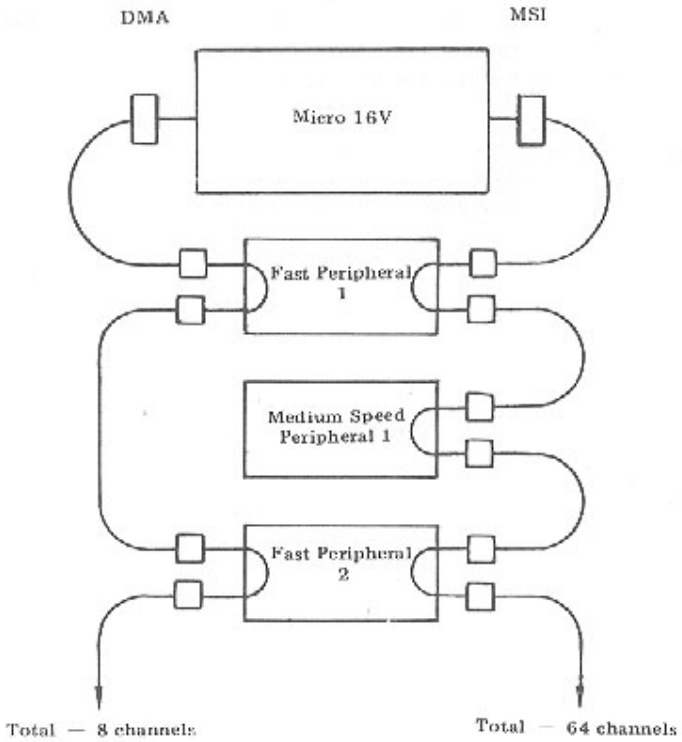


DIAGRAM 39 — CONNECTION FOR FAST AND MEDIUM SPEED PERIPHERALS

Modes of Operation

There are 4 standard modes of operation which are specified by the peripheral controller. The modes are as follows:-

1H mode.	The data address and word count are both supplied by the peripheral controller. The maximum data transfer speed is 857,000 words per second.
H1 mode.	Histogram mode in which the addressed store locations is incremented by 1. As in 1H mode, the store address and word count are both supplied by the peripheral controller.
2H mode.	The data address is held in core store: the word count is supplied by the peripheral controller. The maximum data transfer rate is 429,000 words per second.
3H mode.	The data address and word count are both held in core store, the word count being held in negative form. The data address, as in 2H mode, and the word count are both incremented by 1 after each data transfer. The maximum data transfer rate is 286,000 words per second.

Fixed Addresses

When using DMA in the 2H or 3H modes certain fixed addresses in core store are used. When required, these are:

Channel No.	0	1	2	3	4	5	6	7
Data Address (2H and 3H modes only)	*10	*12	*14	*16	*20	*22	*24	*26
Count (3H mode)	*11	*13	*15	*17	*21	*23	*25	*27

Servicing Times

The servicing time is measured from the DMA request from the peripheral to the processor to the DATA FIN pulse from the processor to the peripheral. In general this is the priority determining time in the DMA control board (approximately .5μS) plus the time waiting for core store access (a maximum of 2 core cycles - between 1/6 and 2 1/6μS) and the time for the DMA cycles (1 1/6μS for 1H and H1 modes, 2 1/3μS for 2H mode 3 1/2μS for 3H mode).

When more than one channel is in use, channels wait for the current cycle (3½μS in 3H mode) to finish and for DMA cycles of higher priority to be finished. Channel 0 is the highest priority channel; channel 7 is the lowest priority channel. The following table gives some typical servicing times for single word transfers using DMA:-

Servicing Times in Microseconds

		Minimum	Typical	Maximum
A single 1H mode channel		1 2/3	3	4
A single 2H mode channel		2 5/6	4	5 1/6
A single 3H mode channel		4	5	6 1/3
Two 1H mode channels	Higher priority	1 2/3	3	4
	Lower priority	1 2/3	4	5 1/6
A 1H mode channel and a 3H mode channel	1H higher priority	1 2/3	3	5
	3H lower priority	4	6	7 1/2
Two 1H mode channels & two 3H mode channels	1H channel priority 1	1 2/3	3	5
	1H channel priority 2	1 2/3	4	6 1/6
	3H channel priority 3	4	6	9
	3H channel priority 4	4	6	12 1/2

Peripherals requiring 2 or more words at a time take longer to service. The maximum service time for other peripherals attached to the same computer DMA is also increased.

When used in conjunction with individual peripheral specifications, the above table enables loading to be estimated. Most Digico designed peripherals do not require a servicing time of less than 10 microseconds.

Effect on Instructions Times

As the store utilisation by the Micro 16-V is about 90%, DMA cycles will slow programs down. Each DMA cycle normally adds 1 1/6 microseconds to program times but when a DMA cycle starts during the SELECT pulse of a peripheral command no delay occurs in the program.

DIRECT MEMORY ACCESS CONNECTIONS

Note: P - represents a signal between +3.0V and +5.25V when present and a signal between -0.25V and +0.5V when absent
N - represents a signal between -0.25V and +0.5V when present and a signal between +3.0V and +5.25V when absent

Micro 16V Crate 156-Way	External 2 x 44 Way Connector	Signal Level	Signal Name	Function
A1			—	Logic 0V
A2	1A20	N		Data Finish Line This line issued by the processor to inform peripherals that the end of the current data transfer block has been reached
A3	1A18	N		Count Zero Line This line is used by the processor in the 3H mode of operation to inform an active peripheral that the current data area is full.
A4	1A2	P	DO0	
A5	1A3	P	DO1	
A6	1A4	P	DO2	
A7	1A5	P	DO3	
A8	1A6	P	DO4	
A9	1A7	P	DO5	Data Output Lines 16 data output lines are used to transmit information from the Micro 16-V core store to the I/O peripheral buffer registers. Up to 8 peripherals may be connected to these lines.
A10	1A8	P	DO6	
A11	1A9	P	DO7	
A12	1A10	P	DO8	
A13	1A11	P	DO9	
A14	1A12	P	DO10	
A15	1A13	P	DO11	
A16	1A14	P	DO12	
A17	1A15	P	DO13	
A18	1A16	P	DO14	
A19	1A17	P	DO15	
A21	2A18	P	HCHA0	
A22	2A1	P	HCHA1	Gating Signals One line for each channel is used to gate the Address, Mode Write and Add signals to the direct memory access bus and to gate Address Accepted.
A23	2B2	P	HCHA2	
A24	2B3	P	HCHA3	
A25	2B4	P	HCHA4	
A26	2B5	P	HCHA5	
A27	2B6	P	HCHA6	
A28	2B7	P	HCHA7	
A29	2A2	P	ADDR0	
A30	2A3	P	ADDR1	
A31	2A4	P	ADDR2	
A32	2A5	P	ADDR3	Core Address Lines 16 address lines enable a peripheral to address any location within the Micro 16-V core store.
A33	2A6	P	ADDR4	
A34	2A7	P	ADDR5	
A35	2A8	P	ADDR6	
A36	2A9	P	ADDR7	
A37	2A10	P	ADDR8	
A38	2A11	P	ADDR9	
A39	2B22		—	Logic 0V

A40	2A12	P	ADDR10	Core Address Lines Continued
A41	2A13	P	ADDR11	
A42	2A14	P	ADDR12	
A43	2A15	P	ADDR13	
A44	2A16	P	ADDR14	
A45	2A17	P	ADDR15	
A78			—	Logic 0V
B1			—	Logic 0V
B2	1A19	N		Address Accepted Line This line is used by the processor to inform peripherals that a requested core address is valid and confirmed.
B3	1A22		—	Logic 0V
B4	1B2	P	DI0	
B5	1B3	P	DI1	
B6	1B4	P	DI2	
B7	1B5	P	DI3	
B8	1B6	P	DI4	
B9	1B7	P	DI5	Data Input Lines 16 data input lines are used to transmit information from the I/0 peripheral buffer registers to the Micro 16-V core store. Up to 8 peripherals may be connected to these lines.
B10	1B8	P	DI6	
B11	1B9	P	DI7	
B12	1B10	P	DI8	
B13	1B11	P	DI9	
B14	1B12	P	DI10	
B15	1B13	P	DI11	
B16	1B14	P	DI12	
B17	1B15	P	DI13	
B18	1B16	P	DI14	
B19	1B17	P	DI15	
B20	IB22		—	Logic 0V
B21	2A19	P	HCHB0	
B22	2B8	P	HCHB1	Gating Signals One line for each channel is used to gate the Data to the direct memory access bus and to strobe Data In with Data Finish.
B23	2B9	P	HCHB2	
B24	2B10	P	HCHB3	
B25	2B11	P	HCHB4	
B26	2B12	P	HCHB5	
B27	2B13	P	HCHB6	
B28	2B14	P	HCHB7	
B29	2A22		—	Logic 0V
B30	1B20	N	HREQ0	
B31	2B15	N	HREQ1	Hesitate Request Lines One line for each channel is used to enable the peripherals to inform the processor that they are ready to transfer data.
B32	2B16	N	HREQ2	
B33	2B17	N	HREQ3	
B34	2B18	N	HREQ4	
B35	2B19	N	HREQ5	
B36	2B20	N	HREQ6	
B37	2B21	N	HREQ7	
B38	2A20	N	ADD	
B39	2B22		—	Logic 0V
B41	1B21	N	WRITE	Write Line This line is used by peripherals to indicate the direction of data transfer required.
B42	1B18	N	Mode Line 1	Two mode lines are used by peripherals to indicate which mode of operation is required.
B43	1B19	N	Mode Line 2	
B78			—	Logic 0V
	2A21		—	FRAME

15. INPUT/OUTPUT TELETYPE

The ASR-33 series of teletype includes a paper tape punch and reader to enable paper tape to be processed at a rate of ten characters per second to match the printer speed or a twenty-five characters per second reader is also available, if required. It can be used for the preparation of paper tapes, or it can print messages or punch tapes from information emanating from the computer or its own keyboard. Furthermore the teletypewriter can operate on-line to a similar device at a remote station, thus providing a direct communication link.

15.1 Operation

The teletype consists of four sub units, the typewriter, keyboard, tape reader and punch. The typing unit translates various electrical code combinations into mechanical printing actions. The keyboard is similar to that of a standard typewriter and generates an 8-channel electrical code. The tape reader and punch both employ a standard 8 channel, 1 inch tape.

The teletype communicates with the computer via the MSI, and the coding employed for all typewriter characters and operations is a sub-set of American Standard Code for Information Interchange (ASC11) of which there is a list in Appendix V.

15.2 Controls

Right Hand

M16-V/LOCAL/REMOTE Switch.

The teletype operates on-line to the computer when the 'M16-V' mode is selected. In the 'LOCAL' mode the teletype can be used for program and data tape preparation. Direct single current communication between one teletype and a remote station is achieved in the 'REMOTE' mode, information being transmitted from the keyboard and received via the printer/punch.

RDR STEP push button.

When depressed in 'LOCAL' mode the reader feeds one character to the printer/punch.

START push button.

Momentory depression of this key causes the teletype motor to be switched on for about 60 seconds. Operation of the printer or the punch causes the motor to stay on. This period is effective from the last teletype operation. The punch is a mechanical slave to the printer and the motor must be on in order to use either device. The motor may be switched off by computer program

ATT push button.

Depression of this button operates a program interrupt in the 'M16-V' mode. The button is illuminated to indicate that power is connected to the teletype.

Control above Punch Mechanism

REL push button.

When depressed the tape punch releases the tape.

BSP push button.

Depression this button manually backspaces the punched paper tape by one character, thus facilitating error deletion.

ON push button.

Enables paper tape to be punched.

OFF push button.

Inhibits punching of paper tape.

15.3 Teletype Instruction Set

Octal Code	Mnemonic	Description
00 50XX	RED	Read Character: Computer accumulator cleared, character in reader buffer is copied into bits 0-7 and RI status unset.
00 40XX	REO	Read Character Or: Character in reader buffer is inclusive OR'ed into bits 0-7 of computer accumulator and RI status is unset.
00 51XX	RST	Read and Reader Start: Computer accumulator cleared, character in reader buffer is copied into bits 0-7 and RI status is unset. Reader operation is started and RI is set when next character is waiting in reader buffer and the reader is ready for next operation.
00 41XX	RSO	Reader OR and Reader Start: Character in reader buffer is inclusive OR'ed into bits 0-7 of the computer accumulator and RI status is unset. Reader operation is started and RI is set when the next character is waiting in the reader buffer and the reader is ready for the next operation.
00 64XX	SRI	Skip next Instruction if RI not set.
00 74XX	SRB	Skip next instruction if Read is not Busy: the reader is mechanically busy as from the execution of an RST or RSO instruction up to the setting of RI status.
00 52XX	KBD	Get Keyboard Character: Computer accumulator is cleared, character in keyboard buffer is copied into bits 0-7 and KI status is unset. KI will set if another character is typed and put into the buffer.
00 43XX	PRT	Print/Punch: bits 0-7 of the computer accumulator are copied into the printer buffer. KI status is unset. The print/punch is initiated and KI will set when the printer becomes ready for the next cycle.
00 72XX	SKB	Skip next Instruction if Keyboard not Busy: the keyboard is mechanically busy from receipt of a PRT instruction to the setting of KI. It is also busy while the keyboard is being used for typing. A PRT instruction should not be given if the keyboard is busy.
00 45XX	MON	Motor On: Keyboard/printer/punch motor is switched on. 'KBD busy' is set for 200ms following this instruction after which time KI is set.
00 44XX	MOF	Motor Off: Keyboard/printer/punch motor is switched off.
00 62XX	SKI	Skip next instruction if KI is not set.
00 61XX	SAI	Skip next instruction if AI is not set: if set, AI status is unset by the test.
00 71XX	TTI	Teletype Interrupt Skip: skip next instruction if neither KI, RI nor AI status is set.

15.4 Notes On Programming

Teletype Interrupts

Programs written for the teletype may run with interrupts permitted or interrupts forbidden. Programs written for interrupts forbidden may use the interrupt status or the busy status.

Three interrupts are associated with the teletype.

1. Attention Interrupt.

   This is set when the attention button on the teletype is pressed, and is unset by the test for the attention interrupt i.e. the SAI instruction will unset the interrupt status if set, and no skip takes place.

2. Reader Interrupt.

   This is set at the end of the reader operate cycle, when the character just read is available to the computer, and is unset by transferring the character in the reader buffer to the computer.

3. Keyboard Interrupt.

   This is set either when a character from the keyboard is available for the computer, or when the print/punch cycle has finished with a character supplied by the computer and the printer/punch is ready for the next one. It is unset by the next print instruction or by transferring the character in the keyboard buffer to the computer

Busy Status

The simplest way of programming for the teletype is to loop on the busy status. However, this has the disadvantage of monopolising the use of the central processor.

The teletype reader is said to be 'busy' from the issue of an RST or RSO instruction to the end of the reader operate cycle at which point the character just read is available in the reader buffer.

The following subroutine may be used to read one character from the teletype reader using the busy status:

```
LKJ
READ=
+ 0
SRB             (SKIP IF READER NOT BUSY
JPU T-1         (WAIT FOR NOT BUSY
RST             (READER BUFFER TO ACCUMULATOR - START READ CYCLE
JPU READ-1      (EXIT WITH CHARACTER IN ACC.
```

The following example uses the above subroutine to read a block of paper tape into an area of store starting at location *2000. Reading from the paper tape ceases when an end-of-tape character (*204) is read. One character is stored in each word.

```
GET /*2000      (SET DATA POINTER TO *2000
STO /POINT
JPS READ        (DUMMY READ - TRANSFER CHARACTER IN
                (BUFFER & START READING 1ST CHARACTER
ALLOP=
JPS READ        (READ A CHARACTER
SUB /*204
JPZ T+5         (TEST FOR E.O.T.
ADD /*204
STI POINT       (STORE CHARACTER IN DATA BUFFER
INC POINT       (UPDATE POINTER
JPU ALOOP       (JUMP TO READ NEXT CHARACTER
HLT             (E.O.T. DETECTED
```

The teletype keyboard is 'busy' from the issue of a PRT instruction up the end of the print/punch cycle. A subsequent PRT should not be given if the keyboard is 'busy'.

The following subroutine may be used to print a character contained in the least significant eight bits of the accumulator on entry.

```
LKJ
PRINT=          (ENTRY WITH CHARACTER IN ACC.
+ 0
SKB             (SKIP IF KEYBOARD NOT BUSY
JPU T-1         (WAIT FOR NOT BUSY
PRT             (PRINT CHARACTER FROM ACCUMULATOR
JPU PRINT-1
```

The paper tape punch is a slave of the printer. Punching only takes place if the punch has been manually switched on.

The teletype keyboard is 'busy' during the entry of a character from the keyboard i.e. during the key depression. The character becomes available to the computer when the mechanism once again becomes not busy. Thus a routine to accept a keyboard character will wait for not busy, wait for busy, wait for not busy and then transfer the character to the computer.

The following subroutine may be used to accept one character from the keyboard using the busy status.

```
LKJ
KEYIN=
+ 0
SKB             (SKIP IF KEYBOARD NOT BUSY
JPU T-1         (WAIT FOR NOT BUSY
SKB             (SKIP IF KEYBOARD NOT BUSY
JPU T+2
JPU T-2         (WAIT FOR BUSY
SKB             (SKIP IF KEYBOARD NOT BUSY
JPU T-2         (WAIT FOR NOT BUSY
KBD             (TRANSFER CHARACTER TO ACCUMULATOR
JPU KEYIN-1
```

Interrupt Status

If interrupts are forbidden the hardware will not force the JPS or JSI instruction when an interrupt status becomes set. However the program may make use of the fact that the status is set e.g. to wait for the pressing of the attention but not before continuing program execution.

```
SAI             (SKIP IF ATTENTION INTERRUPT NOT SET
JPU T+2
JPU T-2
```

The teletype reader interrupt status is set at the end of the reader operate cycle. It is unset by transferring the contents of the reader buffer to the accumulator i.e. by RST,/RSO, RED, or REO instruction.

The following subroutine may be used to read one character from the teletype reader:

```
LKJ
READ1=
+ 0
RST             (UNSET INTERRUPT STATUS & START READ CYCLE
SRI             (SKIP IF NOT READER INTERRUPT STATUS
JPU T+2
JPU T-2         (LOOP ON INTERRUPT STATUS
RED             (TRANSFER CHARACTER TO ACCUMULATOR
JPU READ1-1
```

Note: On first entry to the routine the reader is assumed not Busy.

The keyboard interrupt status is set at the end of a print/punch cycle. It is unset by a PRT or KBD instruction.

Thus a routine to print one character using the interrupt status is as follows:

```
LKJ
PRINT1=
+ 0
PRT             (UNSET KEYBOARD INTERRUPT STATUS & START PUNCH CYCLE
SKI             (SKIP IF NO KEYBOARD INTERRUPT STATUS
JPU PRINT1-1    (EXIT
JPU T-2         (WAIT FOR INTERRUPT STATUS
```

Note: On first entry to the routine the reader is assumed not Busy.

The keyboard interrupt status is also set when a character has been typed on the keyboard and is available in the keyboard buffer.

The following subroutine may be used to accept a single character from the keyboard using the interrupt status:

```
LKJ
KEYIN1=
+ 0
KBD             (CLEAR KEYBOARD INTERRUPT STATUS
SKI             (SKIP IF NO KEYBOARD INTERRUPT STATUS
JPU T+2
JPU T-2         (WAIT FOR KEYBOARD INTERRUPT STATUS
KBD             (CLEAR INTERRUPT & TRANSFER CHARACTER TO ACCUMULATOR
JPU KEYIN1-1
```

Interrupts Permitted

The normal state of the Micro 16V is with peripheral interrupts forbidden. Thus when the machine is switched on or when a general reset is sent to the processor all interrupts except mains fail - restart are automatically forbidden.

To attain the most efficient use of the computer the teletype may be used under the Micro 16V interrupt system.

In general the main program sequence will

a. permit interrupts,

b. initiate the transfer (usually of a string of characters).

The user will also provide an interrupt servicing routine which will in general

a. save the various registers,

b. find the cause of the interrupt,

c. service that interrupt - which must include unsetting the interrupt status,

d. restore the various registers

e. permit further interrupts (for Multilevel interrupt option on that and lower levels),

f. return to interrupted process.

A 'skip chain' is used to find the cause of the interrupt. A general skip chain for a system with teletype only might be:

```
SKI             (SKIP IF NO KEYBOARD INTERRUPT
JPU KISERV      (GO TO SERVICE KEYBOARD INTERRUPT
SRI             (SKIP IF NO READER INTERRUPT
JPU RISERV      (GO TO SERVICE READER INTERRUPT
SAI             (SKIP IF NO ATTENTION INTERRUPT
JPU AISERV      (GO TO SERVICE ATTENTION INTERRUPT
```

The following example shows how a sequence of characters terminated by an end-of-tape character may be read into store from location *2000, simultaneously with performing some process defined by the routine &PROCESS. If the process is completed before the transfer of characters is completed, the program will wait for the end-of-tape character before proceeding to the next part of the program.

The standard single level interrupt option is assumed.

MAIN PROGRAM.

```
GET /*2000
STO /POINT      (SET DATA POINTER TO *2000
CLA
STO /FINIS      (SET END OF TAPE FLAG TO ZERO
SRB             (CHECK FOR READER NOT BUSY
JPU T-1
RST             (INITIATE READ
PIN             (PERMIT INTERRUPTS
LKJ             (PIN INSTRUCTION NOT EFFECTIVE UNTIL LKJ
+T+1

JPS &PROCESS    (PERFORM PROCESS
GET FINIS
JPZ T-1         (TRANSFER NOT COMPLETE
JPU NEXT        (GO TO NEXT PART OF PROGRAM
```

```
SAVE=           (SET UP ENTRY TO INTERRUPT SERVICING
T= /1           (ENTER ABSOLUTE LOADING MODE AT ADDRESS 1
JPU &IRSERV     (GO TO INTERRUPT SERVICING
T= SAVE         (GO BACK TO RELOCATABLE MODE

&IRSERV=
STO /ACC        (SAVE ACCUMULATOR
NCA
STO /CAR        (SAVE CARRY STATE
SKI
JPU KISERV      (KEYBOARD INTERRUPT STATUS SET
SRI
JPU RISERV      (READER INTERRUPT STATUS SET
SAI
NOT             (ATTENTION INTERRUPT STATUS SET - UNSET BY SAI
XIT=
GET 0           (GET INTERRUPT LINK ADDRESS I.E.
STO LINK        (RETURN ADDRESS FOR RE-STARTING INTERRUPTED PROGRAM
GET CAR
ADD /1          (RESET CARRY STATE
GET ACC         (RESET ACCUMULATOR
PIN
LKJ             (RESTART INTERRUPTED PROGRAM - ALLOW FURTHER INTERRUPTS
LINK=
+ 0

KISERV=         (SERVICE KEYBOARD INTERRUPT
KBD             (UNSET KEYBOARD INTERRUPT STATUS
JPU XIT

RISERV          (SERVICE READER INTERRUPTS
RED             (UNSET INTERRUPT STATUS & TRANSFER CHARACTER
SUB /*204
JPZ RI2         (END OF TAPE CHARACTER READ?
ADD /*204
STI POINT       (STORE CHARACTER IN BUFFER
INC POINT       (UPDATE POINTER
RST             (START NEXT READ CYCLE
JPU XIT

RI2=
OCA             (ACCUMULATOR = -1, SET FLAG TO INDICATE
STO FINIS       (FINISHED READING
JPU XIT
```

The keyboard interrupt status is set as a result of:

a. a key depression

b. the end of a print/punch cycle.

The software must determine whether the keyboard interrupt status is the result of an input or an output transfer. This is generally done by means of a flag. In the following example KIMODE is negative zero or positive depending on whether the keyboard is being used for input, not being used or being used for output, respectively.

Interrupt servicing

```
SKI             )
JPU KISERV      )   interrupt servicing skip chain
```

```
KISERV=
GET KIMODE
JPN INPUT
JPZ IGNORE
JPS &OUTPUT     (CATER FOR OUTPUT
JPU XIT

IGNORE=
KBD             (UNSET KEYBOARD INTERRUPT STATUS
JPU XIT
```

```
INPUT=
JPS &INPUT          (CATER FOR INPUT

XIT=
GET 0               (EXIT FROM INTERRUPT SERVICING
STO LINK
etc.
```

Teletype Motor

The teletype motor may be switched on and off under program control. The motor should be switched on before printing/punching; otherwise the first few characters may be incorrectly printed/punched. To minimise wear in the mechanism the motor should be switched off once printing/punching is complete.

The motor may be switched off by preceding the message with two delete characters (all 1's non-printing), as in serial telegraph transmission. The motor will then switch off one minute after the last character is printed. Note: the reader also requires the motor to be on, therefore RST must be given to commence reading, however subsequent instructions may be either RED or RST.

Following a motor on instruction, an interrupt occurs approximately 200 milliseconds later when the motor is up to normal operating speed. The 200 milliseconds delay is related to the MON instruction and occurs even if the motor is already on the mechanism should be given at least 200 milliseconds to allow the motor to reach normal operating speed

Before switching the motor off, the program should check that the mechanism is not busy. This ensures that the printing of the last character is completed before the motor is switched off.

Teletype Channel Numbers

The above examples assume a teletype on channel zero

For a teletype on a channel other than zero, say octal 32 then each instruction mnemonic is qualified by the channel number.

```
e.g.    SKB   *32
        RST   *32
```

16. SOFTWARE INTRODUCTION

The Micro 16V being completely compatible with the other members of the Micro 16 series of computers already has a comprehensive software library covering a wide variety of techniques and applications. This library is constantly being expanded to include many new routines and programs for use with the many available options of the Micro 16V.

Digico maintain a large library of software routines. A software catalogue is supplied to each micro 16-V installation. Paper tapes and documentation for this software is available at a nominal cost.

The newest technique being employed is the Executive.

16.1 Executive — File Reference 5301

Executive programs are available from Digico for Micro 16V computers fitted with the multilevel priority interrupt/memory partition option. Each Executive is tailored for a particular machine configuration.

Loading and Deleting User Programs

User programs are read using a binary relocatable load program. When programs are deleted, the space which they used is made available for other programs.

Interrupt Servicing, Input and Output

Executive looks after the servicing of all interrupts. This eases both documentation and operating since a certain degree of standardisation is automatically achieved amongst all user programs.

All input and output function calls in user programs automatically cause a jump to Executive. A wide variety of input and output functions are provided.

Operator Communication

Facilities are provided to allow operator communication with user programs and Executive via the console teletype.

File Handling

Available with Executive is a software package enabling user programs to easily handle indexed files on disc storage.

Multiprogramming

Also available for the Micro 16V computer is a multiprogramming facility. Executive makes use of the memory partition feature to ensure that no program, whether tested or otherwise, can interfere with another program area.

16.2 Symbolic Assembler — File Reference 5008

This is the basic programming language for the Micro 16V computer. The assembler allows program instructions to be written as mnemonics and refer to store locations by name. Once assembled the program may be loaded with Binary Relocatable loader to anywhere in the computer store. An example of an assembler routine appears in Appendix 6.

16.3 Mathchat — File Reference 5030

Mathchat is an on-line interpretive language designed to solve problems that can be expressed in standard mathematical form. All internal computation is done using the three word floating point package. This gives an accuracy of nine significant decimal digits.

Direct and Indirect Commands

A direct command is interpreted and executed immediately on receipt of the line terminator, i.e. carriage return. Direct commands are adequate for relatively simple problems which can be stated as a series of expressions but are not suitable for problems which require iterative procedures. Indirect commands, which are prefixed by numbers, are not executed immediately but are stored for later execution.

An example of a Mathchat program appears in Appendix 6.

16.4 Floating Point Package (2 word) — File Reference 5003

The usual convention on the Micro 16V computer is for words to represent binary integers, i.e. a fixed point representation. Arithmetic on such numbers is easy using the basic hardware instructions but, when dealing with numbers of greatly differing orders of magnitude, complicated scaling problems arise, By using floating point numbers this problem is greatly eased. The 2 word format of floating point numbers is as shown:-

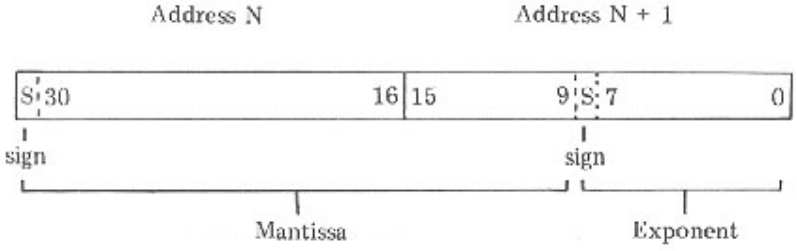


DIAGRAM 40 — 2 WORD FLOATING POINT FORMAT

22 bits plus sign are used for the mantissa and 8 bits plus sign are used for the exponent. This enables all numbers to be used in the range from 10^{-76} to 10^{+76} with an accuracy of six significant decimal digits.

16.5 Utility Routines

These are available for various purposes such as decimal to binary and binary to decimal arithmetic, double length integer arithmetic, floating point arithmetic and mathematical functions, routines to control various peripherals and octal debug.

16.6 User Programs

Despite the amount of software which has already been written for the Micro 16 series of computers it is probable that a program has not been written for the precise requirement of a user. The various programming languages and the incorporation of suitable subroutines make the task of the user programmer much easier. Alternatively Digico provide a consultancy service to solve problems.

17. SERVICES

In realising that the sale of mini-computers in the current market is by no means simply a question of price and performance, Digico Limited offer a wide range of additional services in support of all Micro 16V customers and prospective customers.

The following is a summary of some of these services.

17.1 Application Development

Digico Limited take a personal interest in total systems and application development. The members of our sales staff are selected with regard to their ability to discuss in depth a very wide range of applications for which minicomputers are specifically designed. There are, however, some development areas where the user may not realise the full advantage and hence not realise the full benefit which may be obtained by the introduction of a computer. In these situations our staff will take special care of clients for the development of such applications. The variety of these applications is limitless and embraces both scientific and commercial sectors.

17.2 Turnkey Systems

In many cases the client may wish to be supplied with a total system which includes both hardware and software.

For customer services, Digico Limited have within their organisation two specific departments, the software and hardware departments, which are subdivided into two categories.

a. Market based.

b. Customer based.

Hence we are able to draw upon considerable resources from both departments in the customer based section. Our staff is therefore well qualified to specify and configure systems of any size to suit a variety of jobs. Having specified the job we are then able to follow it through to its successful conclusion and installation.

17.3 Customer Assistance

In areas where the user of Digico equipment wishes to develop his own system we are able to offer assistance. For example the Micro 16V has seven vacant slots for special interface cards. General purpose cards housing 100 integrated circuit chips can be made available. Point to point wiring is achieved by wire wrapping to pins.

Free advice can also be made available for limited consultations on both hardware and software matters.

17.4 Training Courses

A number of software training courses are made available to all Digico customers. The purpose of these courses is to allow members of the customers' own staff to acquaint themselves with all aspects of Digico software. It presupposes that people attending these courses already have a basic understanding of software techniques. A highly qualified member of our staff supervises each course; and individual attention is given to members of each course.

In addition to the software course, we also run maintenance training courses to enable users to undertake their first line maintenance. To ensure the best personal attention to attenders, each course will be limited to approximately ten persons.

Each course lasts for two weeks and includes lunch and light mid-session refreshments. The cost for attending these courses does not include any hotel accommodation which is generally the responsibility of the client.

17.5 Maintenance

Following the warranty period afforded to all Digico customers, we are able to offer a continuing maintenance service, which is carried out by a team of specially trained engineers. The extent of maintenance cover is varied according to negotiation, but in general falls within two categories:-

On Call Service

Some very small systems, which may include a teletype as the only electro-mechanical peripheral in the system, may not justify a fully comprehensive cover. In this case we offer call out service which is charged at an hourly rate, plus the cost of components used.

The alternative is to provide 500 hours routine contract maintenance on the teletype at a fixed charge per year with emergency call extra, being charged at an hourly rate.

Comprehensive

On larger systems it is usual to provide comprehensive maintenance, whereby Digico contracts with the customer to provide all emergency and routine maintenance. This takes the form of an insurance policy with an agreed percentage of the capital cost of the system being paid annually in advance.

APPENDIX I

Binary

The majority of computers, including the Micro 16V, operate on numbers expressed in binary. These are numbers working on a base of 2, which is ideally suited to the computer as an ON signal can be regarded as 1 and an OFF signal as 0. Like decimal numbers, binary numbers are written with the least significant bit on the right.

e.g. 10110

$= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

= 22 in decimal

The Micro 16V uses a 16 bit binary number, typically of the form

0011010110101101

Octal

Binary numbers are very awkward to use, hence in the Micro 16V all instructions can be entered in octal form. These have a similar appearance to decimal numbers but have a base of 8, and are directly related to binary numbers.

To convert the binary number to octal, the number is divided into groups of 3 (giving each group a value of up to $2^3 = 8$), starting from the right hand side. Hence with the previous number

0 011 010 110 101 101

Each group is now treated individually, and given its octal value

(i.e. $101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$)

Thus the above number becomes

*032655.

Note * denotes an octal number.

Bit Significance

Occasionally it is convenient to refer to the structure of binary numbers by talking about 'bit positions'. In a standard Micro 16V, the bits are labelled from 0 through to 15, from right to left. The number of the bit gives the significance of the number in binary, hence bit N has the value of 2^N. This means that bit 0 is the least significant bit (LSB) with a value of 1 and bits 15 is the most significant bit (MSB) with a value of 32,768.

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary number	0	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1
Bit significance	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

APPENDIX II

GLOSSARY OF TERMS

		Reference Section
Processor (CPU)	The control part of the computer including registers.	
Core	The storage part of the computer, size measured in computer words.	2
Stack	4K words of storage, all program in one stack is directly addressable.	2
K	1024 when referring to words of core store	2
MSI	Medium Speed Interface	13
DMA	Direct Memory Access	14
bit	The smallest unit of information, may only have the status '0' or '1'.	
byte	8 bits.	
Word	16 bits for the M16V, may contain binary data, an instruction or 2 bytes. The smallest addressable unit of storage.	
character	1 byte of printable information, alphanumeric or symbolic.	
MSB	Most Significant Bit	A1
LSB	Least Significant Bit	A1
A Register	Accumulator - working register of processor.	2
P Register	Program register - holds address of next instruction.	2
I Register	Instruction register - holds present instruction.	2
S Register	Switchbank register - used for external communication during the running of a program.	2
C Register	Carry register - used as bit 16 in ADD, SUB, etc.	2
IDR	Interrupt Detected Register used in MLI.	7
IER	Interrupt Enable Register used in MLI.	7
ILR	Interrupt Levels Register used in MLI.	7
buffer	An intermediate storage area temporarily holding input/output data.	
bootstrap	The basic initial program used to load in binary read.	6
binary read	Reads in program, normally from paper tape, in a binary checkable form.	6
MLI	Multilevel Interrupt	7
FPU	Floating Point Unit	12
Interrupt	Special signal raised by any device which causes normal flow of processor to be temporarily suspended.	7
Channel	6 bit address defining a unique line between a peripheral and CPU, used for data transfer.	
Peripheral	A device used for transferring data to or from the computer, or for storing data or performing data conversion.	
*	Octal.	

APPENDIX III

SPECIFICATION

MICRO 16V		16 bit parallel machine with 6MHz clock rate.
CORE STORE		Available in units of 4K, from 4K to 64K.
CORE CYCLE TIME		950 μS.
ADDRESSING	— DIRECT	Any word in the current 4K stack.
	— INDIRECT	Any word up to 64K.
CONTROL PANEL	1)	6 digit octal display.
	2)	16 bit binary display
	3)	Register displayed indicator
	4)	GO and carry indicators.
	5)	24 key keyboard for control and data..
INPUT AND OUTPUT	1)	The Micro 16V will support up to 64 peripherals on Medium speed transfer at speeds of up to 3M bits/second.
	2)	Fast transfer by DMA at speeds up to 13.7M bits/second.
COMPATIBILITY		All programs and peripherals used with any of the previous members of the Micro 16 series can be used on the Micro 16V.
RELIABILITY		MTBF is 15,000 hours, based upon the summation of individual component figures.
PHYSICAL CHARACTERISTICS		*Environment* Ambient Temperature - 0^{o}C to 50^{o}C. Relative Humidity - 0% to 90% non-condensing.

Dimensions of Basic Computer

Height	307 mm	(12.125 inches)
Width	483 mm	(19 inches)
Depth	587 mm	(23.125 inches)
Weight	28 Kg	(60 lbs)

Construction

The Micro 16V is available as a table top model or 19 in. rack mounted.

ELECTRICAL REQUIREMENTS	*Mains Input:*	110 - 115 or 200 - 250V 50/60 Hz
		Other voltages and frequencies to order.
	Power consumption:	basic computer 750 watts. ASR33 teletype 200 watts.
STANDARD FEATURES		Automatic Bootstrap, Mains Fail and Restart
ADDITIONAL OPTIONS		Multilevel interrupt, memory partition, real time clock, Floating Point Unit, Direct Memory Access.
PERIPHERALS		The range of peripherals available is very wide. They include ASR33 Teletype single and twin Exchangeable Discs, Paper Tape Readers and Punches, Line Printers, Magnetic Drums and Magnetic Tape.

ASR-33 TELETYPE
SPECIFICATION

PRINTER	216 mm (8.5 inch) friction feed, allowing original plus one copy. [Optional - 216 mm (8.5 inch) sprocket feed, allowing original plus two copies].
FORMAT	10 characters per inch. 72 characters per line.
SPEED	10 characters per second. (optional reader only - 25 characters per second).
CODE	Keyboard and Printer — ASCII Reader and Punch — all codes on 5, 6, 7 and 8 channel paper tape.
PHYSICAL CHARACTERISTICS	*Environment* Ambient Temperature - 5°C to 35°C. Relative Humidity - 20% to 90% non-condensing.

Dimensions

Height	840 mm	(33 inches)
Width	560 mm	(22 inches)
Depth	483 mm	(19 inches)
Weight	25.4 Kg	(56 lbs)

APPENDIX IV

MICRO 16V INSTRUCTION SUMMARY

Under 'Effect' each abbreviation refers to a content

- S = Store, the word addressed by NNNN
- A = Accumulator
- C = Carry register
- P = Program address register
- ' = New contents of

Brackets are used in indirect instructions to indicate that the contents referred to are those of the pointer whose address is given in the word addressed by the operand itself.

Mnemonic	Octal Code	Time in Microseconds	Effect
GET	04 NNNN	2.15	A' = S
GTI	06 NNNN	3.15	A' = (S)
STO	05 NNNN	2.15	S' = A
STI	07 NNNN	3.15	(S)' = A
ADD	14 NNNN	2.15	A' = A + S
ADI	16 NNNN	3.15	A' = A + (S)
SUB	15 NNNN	2.15	A' = A - S
AND	10 NNNN	2.3	A' = A & S
INC	01 NNNN	2.3	S' = S + 1, skip next instruction, if S' = 0
DEC	11 NNNN	2.3	S' –S - 1, skip next instruction, if S' = 0
JPU	02 NNNN	1.15	P' = NNNN
JPZ	03 NNNN	1.15	P' = NNNN if A = 0
JPN	13 NNNN	1.15	P' = NNNN if A 0, i.e. Bit 15 = 1
JPS	12 NNNN	2.3	P' = NNNN + 1, S' = P + 1
JSI	17 NNNN	3.3	P' = (NNNN) + 1, (S') = P + 1
LKJ	00 0400	2.3	P' = (P + 1)
NOT	00 0000	1.15	Do nothing
HLT	00 0040	1.15	Halt
CRY	00 0001	1.15	A' = C
NCA	00 1420	4.15	A' = - C
CRS	00 0200	1.15	C' = 1
CRU	00 0100	1.15	C' = 0
CLA	00 1020	4	A' = 0
OCA	00 1060	4	A' = $\overline{A}$
TCA	00 1061	1.3	A' = $\overline{A}$ + 1
SWB	00 0002	1.3	A' = Switchbank

SHL	00 1300	1.5 + n/6	Shift left propagating zeros
SHR	00 1000	1.5 + n/6	Shift right propagating zeros
CIR	00 1100	1.5 + n/6	Circulate right propagating least significant bit.
RSS	00 1200	1.5 + n/6	Shift right propagating sign bit
RCI	00 1040	1.5 + n/6	Circulate right propagating inverse of least significant bit.
RSC	00 1400	1.5 + n/6	Shift right propagating carry
FIN	00 0020	1.15	Forbid interrupts
PIN	00 0030	1.15	Permit interrupts after next LKJ instruction.

Teletype Instructions

RED	00 50 PP	Clear accumulator and read a character.
REO	00 40 PP	Read a character using inclusive OR logic.
RST	00 51 PP	Start reader, clear accumulator and read a character.
RSO	00 41 PP	Start reader, read a character using inclusive OR logic.
KBD	00 52 PP	Clear accumulator and get keyboard character.
KBO	00 42 PP	Get keyboard character using inclusive OR logic.
PRT	00 43 PP	Print punch or write a character from the accumulator.
MON	00 45 PP	Switch motor on.
MOF	00 44 PP	Switch motor off.
SAI	00 61 PP	Skip next instruction if AI is not set.
SKI	00 62 PP	Skip next instruction if KI is not set.
SRI	00 64 PP	Skip next instruction if RI is not set.
SKB	00 72 PP	Skip next instruction if device is not busy.
SRB	00 74 PP	Skip next instruction if reader is not busy.
TTI	00 71 PP	Skip next instruction if none of AI, RI, KI is set.

Note: These instructions may also apply to other peripherals such as Fast Readers and Fast Punches, where relevant. PP represents the channel number.

Multi-Level Interrupt Option

LEX	00 0031	3.8	Permit interrupts for current level after LKJ
SID	00 0032	3.8	Set interrupt detect register from accumulator
SIE	00 0033	3.8	Set interrupt enable register from accumulator
RID	00 0036	3.8	Read interrupt detect register into accumulator
RLP	00 0037	3.8	Read interrupt level register into accumulator

Memory Partition Option

SPL	00 4177	4.15	Set partition limits from the accumulator
PEX	00 4277	4.15	Generate exit interrupt from restricted area of store
SPI	00 4377	4.15	Skip if no interrupt from program restrict option

Real Time Clock Option

ECI	00 4477	4.15	Enable clock interrupts and start clock
ICI	00 4577	4.15	Unset clock interrupts and reset clock
SCI	00 4677	4.15	Skip next instruction if no clock interrupt and unset interrupt
SNS	00 4777	4.15	Skip if 'not serviced' interrupt and unset interrupt.

Floating Point Option

Mnemonic	Octal Code	Effect
MPY	00 2000	Fixed point multiply
DIV	00 2001	Fixed point divide
FAD	00 2XXX	Floating point add
FSB	00 2XXX	Floating point subtract
FMY	00 2XXX	Floating point multiply
FDV	00 2XXX	Floating point divide
FGT	00 2XXX	Get number into floating point accumulator
FST	00 2XXX	Store number from floating point accumulator

Note: XXX — to be defined at a later date.

APPENDIX V

OCTAL/DECIMAL TABLES
I

Octal 0-377 → Decimal 0-255

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
10	8	9	10	11	12	13	14	15
20	16	17	18	19	20	21	22	23
30	24	25	26	27	28	29	30	31
40	32	33	34	35	36	37	38	39
50	40	41	42	43	44	45	46	47
60	48	49	50	51	52	53	54	55
70	56	57	58	59	60	61	62	63
100	64	65	66	67	68	69	70	71
110	72	73	74	75	76	77	78	79
120	80	81	82	83	84	85	86	87
130	88	89	90	91	92	93	94	95
140	96	97	98	99	100	101	102	103
150	104	105	106	107	108	109	110	111
160	112	113	114	115	116	117	118	119
170	120	121	122	123	124	125	126	127
200	128	129	130	131	132	133	134	135
210	136	137	138	139	140	141	142	143
220	144	145	146	147	148	149	150	151
230	152	153	154	155	156	157	158	159
240	160	161	162	163	164	165	166	167
250	168	169	170	171	172	173	174	175
260	176	177	178	179	180	181	182	183
270	184	185	186	187	188	189	190	191
300	192	193	194	195	196	197	198	199
310	200	201	202	203	204	205	206	207
320	208	209	210	211	212	213	214	215
330	216	217	218	219	220	221	222	223
340	224	225	226	227	228	229	230	231
350	232	233	234	235	236	237	238	239
360	240	241	242	243	244	245	246	247
370	248	249	250	251	252	253	254	255

Octal 400-777 → Decimal 256-511

	0	1	2	3	4	5	6	7
400	256	257	258	259	260	261	262	263
410	264	265	266	267	268	269	270	271
420	272	273	274	275	276	277	278	279
430	280	281	282	283	284	285	286	287
440	288	289	290	291	292	293	294	295
450	296	297	298	299	300	301	302	303
460	304	305	306	307	308	309	310	311
470	312	313	314	315	316	317	318	319
500	320	321	322	323	324	325	326	327
510	328	329	330	331	332	333	334	335
520	336	337	338	339	340	341	342	343
530	344	345	346	347	348	349	350	351
540	352	353	354	355	356	357	358	359
550	360	361	362	363	364	365	366	367
560	368	369	370	371	372	373	374	375
570	376	377	378	379	380	381	382	383
600	384	385	386	387	388	389	390	391
610	392	393	394	395	396	397	398	399
620	400	401	402	403	404	405	406	407
630	408	409	410	411	412	413	414	415
640	416	417	418	419	420	421	422	423
650	424	425	426	427	428	429	430	431
660	432	433	434	435	436	437	438	439
670	440	441	442	443	444	445	446	447
700	448	449	450	451	452	453	454	455
710	456	457	458	459	460	461	462	463
720	464	465	466	467	468	469	470	471
730	472	473	474	475	476	477	478	479
740	480	481	482	483	484	485	486	487
750	488	489	490	491	492	493	494	495
760	496	497	498	499	500	501	502	503
770	504	505	506	507	508	509	510	511

Octal 1000-1377 → Decimal 512-767

	0	1	2	3	4	5	6	7
1000	512	513	514	515	516	517	518	519
1010	520	521	522	523	524	525	526	527
1020	528	529	530	531	532	533	534	535
1030	536	537	538	539	540	541	542	543
1040	544	545	546	547	548	549	550	551
1050	552	553	554	555	556	557	558	559
1060	560	561	562	563	564	565	566	567
1070	568	569	570	571	572	573	574	575
1100	576	577	578	579	580	581	582	583
1110	584	585	586	587	588	589	590	591
1120	592	593	594	595	596	597	598	599
1130	600	601	602	603	604	605	606	607
1140	608	609	610	611	612	613	614	615
1150	616	617	618	619	620	621	622	623
1160	624	625	626	627	628	629	630	631
1170	632	633	634	635	636	637	638	639
1200	640	641	642	643	644	645	646	647
1210	648	649	650	651	652	653	654	655
1220	656	657	658	659	660	661	662	663
1230	664	665	666	667	668	669	670	671
1240	672	673	674	675	676	677	678	679
1250	680	681	682	683	684	685	686	687
1260	688	689	690	691	692	693	694	695
1270	696	697	698	699	700	701	702	703
1300	704	705	706	707	708	709	710	711
1310	712	713	714	715	716	717	718	719
1320	720	721	722	723	724	725	726	727
1330	728	729	730	731	732	733	734	735
1340	736	737	738	739	740	741	742	743
1350	744	745	746	747	748	749	750	751
1360	752	753	754	755	756	757	758	759
1370	760	761	762	763	764	765	766	767

Octal 1400-1777 → Decimal 768-1023

	0	1	2	3	4	5	6	7
1400	768	769	770	771	772	773	774	775
1410	776	777	778	779	780	781	782	783
1420	784	785	786	787	788	789	790	791
1430	792	793	794	795	796	797	798	799
1440	800	801	802	803	804	805	806	807
1450	808	809	810	811	812	813	814	815
1460	816	817	818	819	820	821	822	823
1470	824	825	826	827	828	829	830	831
1500	832	833	834	835	836	837	838	839
1510	840	841	842	843	844	845	846	847
1520	848	849	850	851	852	853	854	855
1530	856	857	858	859	860	861	862	863
1540	864	865	866	867	868	869	870	871
1550	872	873	874	875	876	877	878	879
1560	880	881	882	883	884	885	886	887
1570	888	889	890	891	892	893	894	895
1600	896	897	898	899	900	901	902	903
1610	904	905	906	907	908	909	910	911
1620	912	913	914	915	916	917	918	919
1630	920	921	922	923	924	925	926	927
1640	928	929	930	931	932	933	934	935
1650	936	937	938	939	940	941	942	943
1660	944	945	946	947	948	949	950	951
1670	952	953	954	955	956	957	958	959
1700	960	961	962	963	964	965	966	967
1710	968	969	970	971	972	973	974	975
1720	976	977	978	979	980	981	982	983
1730	984	985	986	987	988	989	990	991
1740	992	993	994	995	996	997	998	999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

Octal 2000-2377 → Decimal 1024-1279

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

Octal 2400-2777 → Decimal 1280-1535

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

Octal 3000-3377 → Decimal 1536-1791

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

Octal 3400-3777 → Decimal 1792-2047

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

Octal → Decimal
4000-4377 → 2048-2303

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

Octal → Decimal
4400-4777 → 2304-2559

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

Octal → Decimal
5000-5377 → 2560-2815

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

Octal → Decimal
5400-5777 → 2816-3071

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

Octal 6000-6377 → Decimal 3072-3327

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

Octal 6400-6777 → Decimal 3328-3583

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

Octal 7000-7377 → Decimal 3584-3839

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

Octal 7400-7777 → Decimal 3840-4095

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

OCTAL REPRESENTATION OF ASCII

CHARACTERS

CHAR.	SHIFTED. BITS 8-15	OCTAL. BITS 0-7	CHAR.	SHIFTED. BITS 8-15	OCTAL. BITS 0-7
0	030	060	SP	120	240
1	1304	261	!	0204	041
2	131	262	"	021	042
3	0314	063	#	1214	243
4	132	264	£	022	044
5	0324	065	%	1224	245
6	033	066	&	1223	246
7	1334	267	'	0234	047
8	134	270	(	024	050
9	0344	071	)	1244	251
A	0404	101	*	125	252
B	041	102	+	0254	053
C	1414	303	,	126	254
D	042	104	-	0264	055
E	1424	305	.	027	056
F	143	306	/	1274	257
G	0434	107	:	035	072
H	044	110	;	1354	273
I	1444	311	<	036	074
J	145	312	=	1364	275
K	0454	113	>	137	276
L	146	314	?	0374	077
M	0464	115	@	140	300
N	047	116	[	1554	333
O	1474	317	\	056	134
P	050	120	]	1564	335
Q	1504	321	↑	157	336
R	151	322	←	0574	137
S	0514	123	L/F	005	012
T	152	324	C/R	1064	215
U	0524	125	EOT	102	204
V	053	126	BELL	1034	207
W	1534	327	ESCAPE	0154	033
X	154	330	DELETE	1774	377
Y	0544	131			
Z	055	132			

APPENDIX VI

ASSEMBLER EXAMPLE

Subroutine to sum a table of single length positive numbers.

Name	SUM
Function	To sum a table from 1 to 32,767 single length positive numbers in adjacent locations to obtain a double length result.
Mode of Entry	The mode of entry to the subroutine is as follows:- JPS SUM + T A B L E where + TABLE is the address of the first word of the table of numbers to be summed. Also at entry to the subroutine the accumulator must contain a count of the number to be summed.
Result	The sum is placed in MAJOR (more significant half) and MINOR (less significant half). If the number count is out with the range 1 to 32,768, MAJOR and MINOR are zeroised and exit is made from the subroutine to the JPS (or JSI) instruction + 2. If during summation a negative number is encountered, the number is left in the accumulator and the subroutine exits to the JPS (or JSI) instruction + 3. Otherwise the subroutine exits to the JPS (or JSI) instruction + 4.

SUBROUTINE:

```
SOUT=       (TAG FOR EXIT ROUTINES
INC SUM     (STANDARD EXIT ROUTINE
INC SUM     (EXIT ROUTINE FOR NEGATIVE NUMBER
INC SUM     (EXIT ROUTINE IF COUNT OUT OF RANGE
LKJ         (EXIT INSTRUCTION
SUM=        (TAG FOR LINK
0           (LINK TO RETURN TO MAIN PROGRAM
STO /COUNT  (STORE COUNT OF NUMBERS
CLA         (ZEROISE ACCUMULATOR
STO /MAJOR  (CLEAR GREATER HALF OF ANSWER
STO /MINOR  (CLEAR LESSER HALF OF ANSWER
GET COUNT   (GET COUNT OF NUMBERS
JPZ SOUT+2  (JUMP IF COUNT IS ZERO
JPN SOUT+2  (JUMP IF COUNT IS NEGATIVE
GTI SUM     (GET LINK
STO /TEM    (STORE LINK
SLOOP=      (TAG FOR MAIN LOOP OF SUBROUTINE
GTI TEMP    (GET A NUMBER
JPN SOUT+1  (JUMP IF NUMBER IS NEGATIVE
ADD MINOR   (ADD LESS SIGNIFICANT HALF OF SUM
STO MINOR   (STORE LESS SIGNIFICANT HALF OF SUM
CRY         (LOAD CARRY
JPZ T+2     (JUMP IF CARRY IS ZERO
INC MAJOR   (OTHERWISE ADD ONE TO GREATER HALF
INC TEMP    (SET POINTER TO NEXT NUMBER
DEC COUNT   (REDUCE COUNT BY ONE
JPU SLOOP   (JUMP BACK TO START OF MAIN LOOP
JPU SOUT    (JUMP TO STANDARD EXIT ROUTINE
```

MATHCHAT EXAMPLE

This is an iterative example which sets a gunner a problem.

A missile is approaching and the problem is to hit it before it hits the gunner. After the enemy is sighted, an attempt should be made to set the velocity and elevation of the gun for a direct hit. The distance by which the target is missed is recorded, for a direct hit this value should be less than 50 feet.

The gun is reset, but meanwhile the missile has come 5,000 feet nearer, hence account for this must be taken in the next iteration.

The program and first iteration are shown below.

```
10S DI=100000*FRAN(0);IF ( DI-10000),10,20,20
20T !,"ENEMY SIGHTED ! SET VELOCITY(FT/SEC) AND ",!
30T "ELEVATION(DEG) OF YOUR GUN FOR A DIRECT HIT",!
40T !,"ENEMY SIGHTED @ ",27.00,DI,"FEET",!
50T !,"VELOCITY ";A VE;T !,"ELEVATION" A EL
60S EL=EL*3.14159/180
70S RA=2*VE'2*FSIN(EL)*FCOS(EL)/32.2
80IF (FABS(DI - RA) - 50), 120,90,90
90T !,"MISSED TARGET BY",DI-RA," FEET",!
100S DI=DI-5000;IF (DI-50),110,110,40
110T !!,"BANG, BANG !! YOU'RE DEAD",!!!;QUIT
120T !!,"CONGRATULATIONS, DIRECT HIT",!!;GO 10
```

```
*GO
ENEMY SIGHTED  !  SET VELOCITY (FT/SEC) AND
ELEVATION(DEG) OF YOUR GUN FOR A DIRECT HIT
ENEMY SIGHTED  @       50000 FEET
VELOCITY  :  1290
ELEVATION :  45
MISSED TARGET BY  -    1680 FEET
```

NOTES

digico

Digico Limited
Wedgwood Way
Stevenage
Hertfordshire
England SG1 4PY

Telephone
Stevenage (0438) 4381