TALKING ELECTRONICS®

$1.80★
$2.40NZ

Issue No 11

2 'Add-ons' for the TEC-1

★8x8 DISPLAY
★RELAY DRIVER BOARD

DUAL POWER SUPPLY
BLACK JACK
FM BUG

BOZENNA CAROLINE ALAN

TALKING ELECTRONICS

Vol. 1 No: 11.

Editorial...

Understandably, the introduction of a computer kit for under $100 created quite a reaction. In the weeks following the release of issue 10, orders came thick and fast.

But we soon found some supply problems creeping in. Many of the chips used in the computer became difficult to obtain. And obviously the omission of any one component makes a kit useless. One of the factors causing this is the Worldwide rise in demand for components. It began in the U.S. and has increased dramatically over the past few months. It has come to the stage of almost completely exhausting supplies of some components. And we suffered.
As you know, the business world has wound itself down to the lowest level possible during the recent years of minimal trading and no-one is holding surplus stocks.

But now it looks like the tide has turned. Everything is getting into gear. So don't be left behind.

If you have been waiting for a project which will teach you about the actual workings of a computer, the TEC-1 is the answer. If you further delayed your purchase pending the scope of the project, I hope this issue will allay your fears. The TEC-1 is is not a one-off, half-finished, one-issue-and-it's-forgotten project. It's a continuing fully-fledged applications machine.

Behind the scenes we have been working frantically on "add-ons", programming details and presentation which will make the TEC-1 a perfect demonstrator for both school and personal use. We have now completed (nearly) NINE add-on boards, making the TEC-1 one of the most universal micro projects on the market.
The fact that we chose the Z80 chip and Machine Code programming takes us far ahead of anything else on the market. The Z80 is the lowest priced, most advanced, processor available and after only a few hours experimenting with loading the programs from last issue, a number of readers have phoned to say how much they have learnt.

It's an ABSOLUTE learning tool.

And THE most interesting fact about Machine Code programming has never ever been mentioned before.
Once you remember some of the Z80 codes, you can read the program just like a sentence in English.
Some of the add-ons already in the final stages of development include a speech board, additional memory, video display board and EPROM burner, remote control keyboard, in/out port expansion including CTC to give real-time capabilities, latch board with 24 outputs per board and a latch board with 16 buffered outputs and one 8-bit input (1 byte).
The latest request, from Allan Bolton, was to be able to examine the registers. In an hour or so, Ken had re-written the monitor program to be able to do this. Such is the universality of a computer. Its possibilities are endless.
To date we have delivered nearly 1,000 computers to our readers. If you haven't got into computers yet, now's your chance.

Otherwise YOU'LL be needing the special disk put out by APPLE computers!

Colin Mitchell.

INDEX

TECHNICAL

Ken Stone

ARTWORK

Ken Stone

ENQUIRIES

10 Minute queries will be answered on 584 2386 9am - 6pm.

ADVERTISING

Talking Electronics: (03) 584 2386

COMING!

Look out for our next model, the TEC-1A. It has the regulator under the PC and two new display drivers. Either model will fit onto the computer case and these are available from TE for $19.50 incl. post and packing.

PUBLISHER

Registered by Australia Post Publication Number VBP 4256

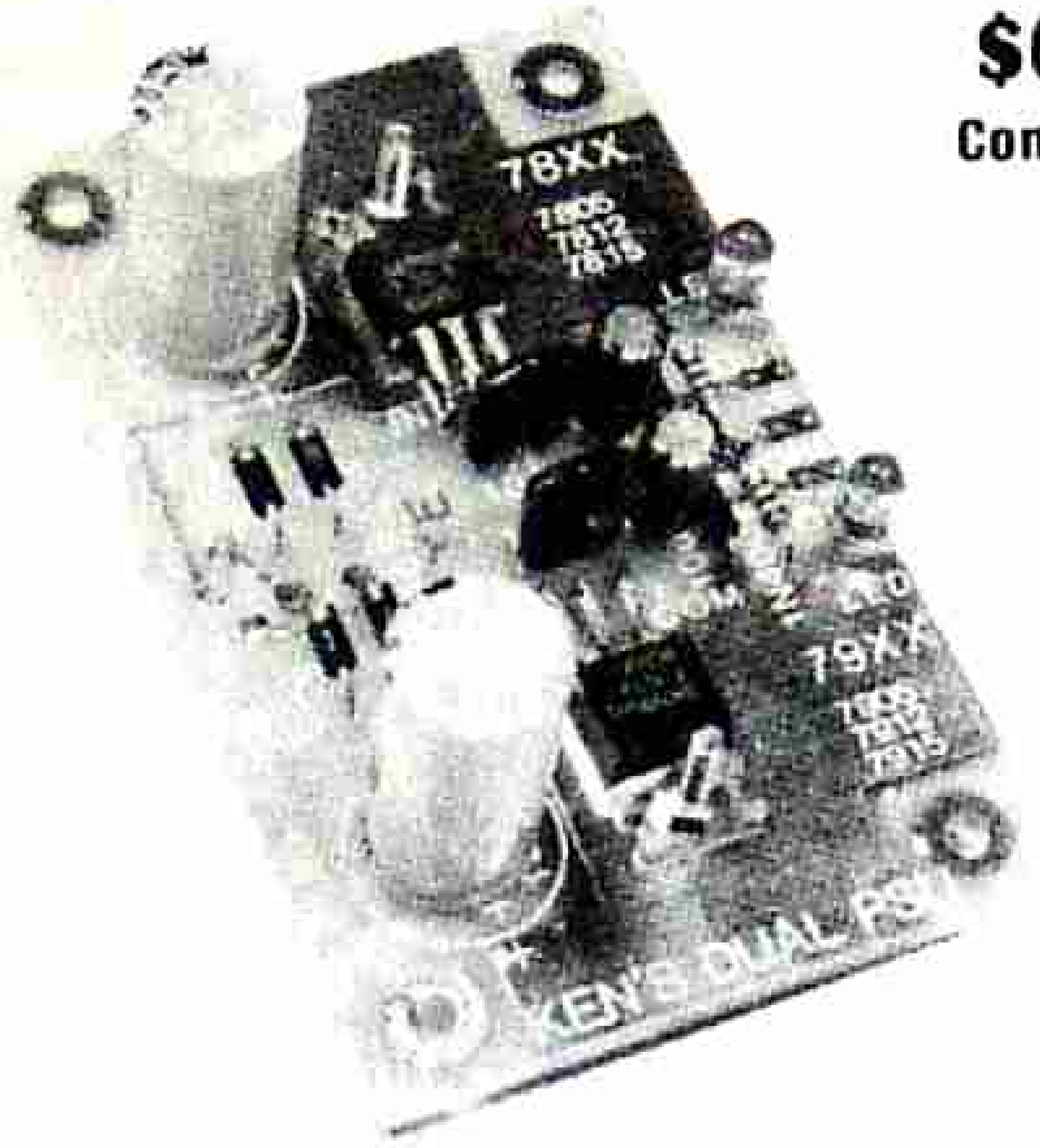
TALKING ELECTRONICS is designed by Colin Mitchell of CPW INDUSTRIES, at 35 Rosewarne Ave., Cheltenham, Victoria, 3192. Australia. Articles suitable for publication should be sent to this address. You will receive full assistance with final presentation.

★Maximum recommended retail price only.

Printed Web Offset (big rolls of paper) by Standard Newspapers Ltd, 10 Park Rd., Cheltenham, 3192.

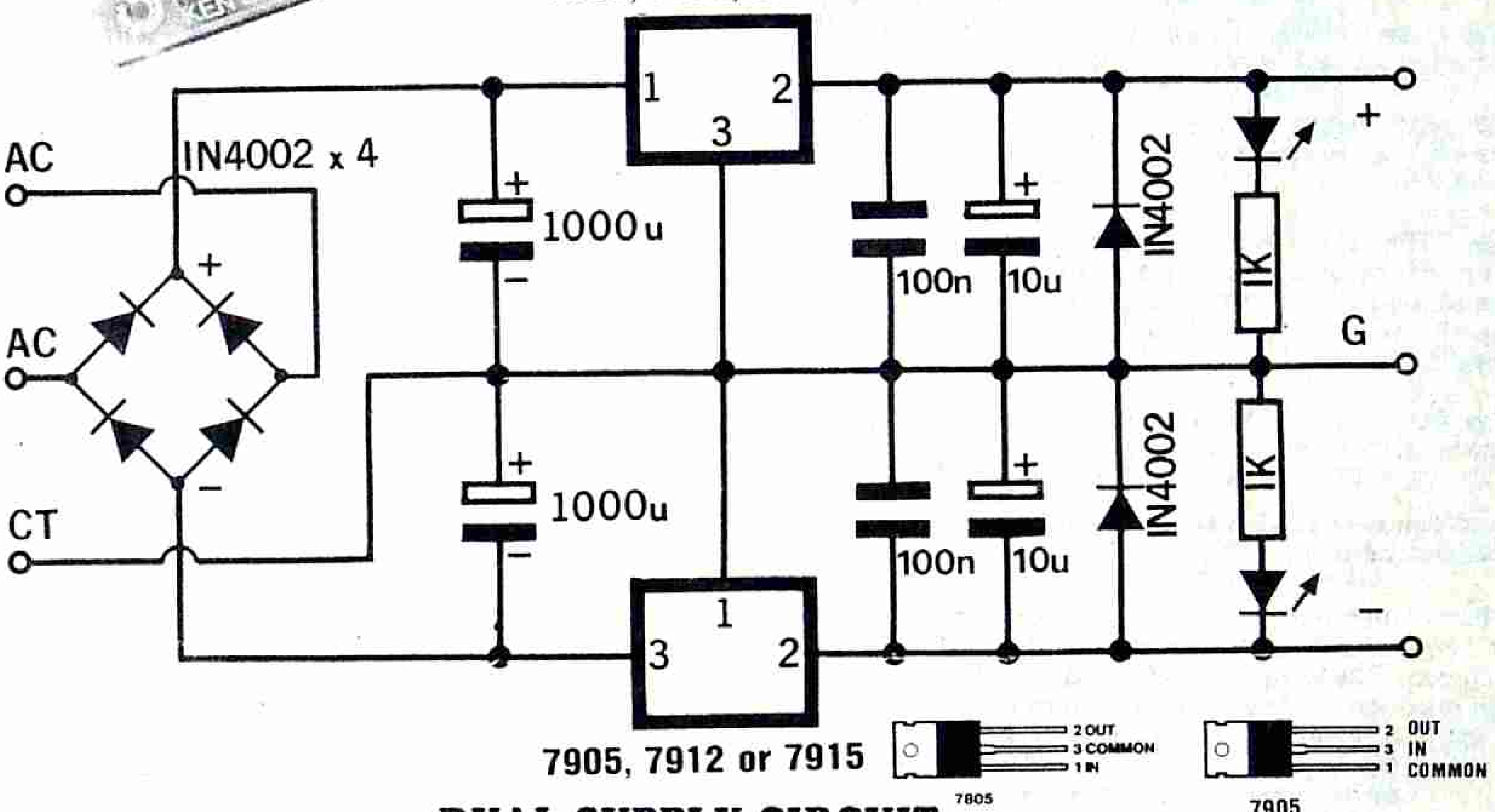
Distributed in Australia by Gordon & Gotch.

KEN'S DUAL POWER SUPPLY



$6.90
Components

$3.30
PC Board



DUAL SUPPLY CIRCUIT

'Necessity is the mother of invention.

That's how this project came to be.

It's a simple 500mA power supply providing a positive and negative rail with an output voltage which is governed by the voltage of the regulator you use.

The PC layout is somewhat simplified by the fact that the board acts as the heatsink. It will dissipate a total of 5 watts for a rise of 75°C. This equates to 2.5 watts per regulator and will give a 600mA output when the voltage-difference between input and output is 4 volts.

If this voltage difference increases, the output current must be reduced to prevent overheating. If you don't understand this, it is discussed later in the article.

The main need for a simple dual power supply was born when Ken began designing synthesiser projects for future issues of TE. These were basically audio projects and they required both positive and negative rails.

As no dual supply is readily available on the market at a low price, he had to assemble one on Matrix board each time he needed a supply. After producing more than 7 of these, he decided it was time to present a PC design for TE. He could then take a PC board from stock and run up a supply in double-quick time.

This project is the result of specific needs and also fills a gap for a simple dual supply using the minimum of components.

PARTS LIST

2 - 1k ¼watt
2 - 100n ceramic or greencap

6 - 1N 4002 diodes

2 - 10mfd 16v electrolytics
2 - 1000mfd 25v electrolytics

1 - 7815 regulator (positive)
1 - 7915 regulator (negative)

2 - 3mm red LEDs

2 sets nuts and bolts
small amount of thermal grease
15cm hook-up wire, 6 colours

1 - DUAL PSU PC BOARD

1 or 2 - 2155 transformers
or 6672 transformer (see text).

It is mainly designed as a +15v and −15v circuit but can be used as a single 5v or 15v supply. It can even be used with different value regulators however this is not recommended as the power lost in the lower-voltage regulator will determine the maximum current.

Because there are a number of options for output voltages, it is necessary to know what you are doing and what parts will be required, before starting.

We will cover 4 possibilities and show which transformer is required. If you use the wrong transformer or the wrong tappings, you will fail to get the full voltage and current.

The two transformers we have selected are the 2155 and 6672.

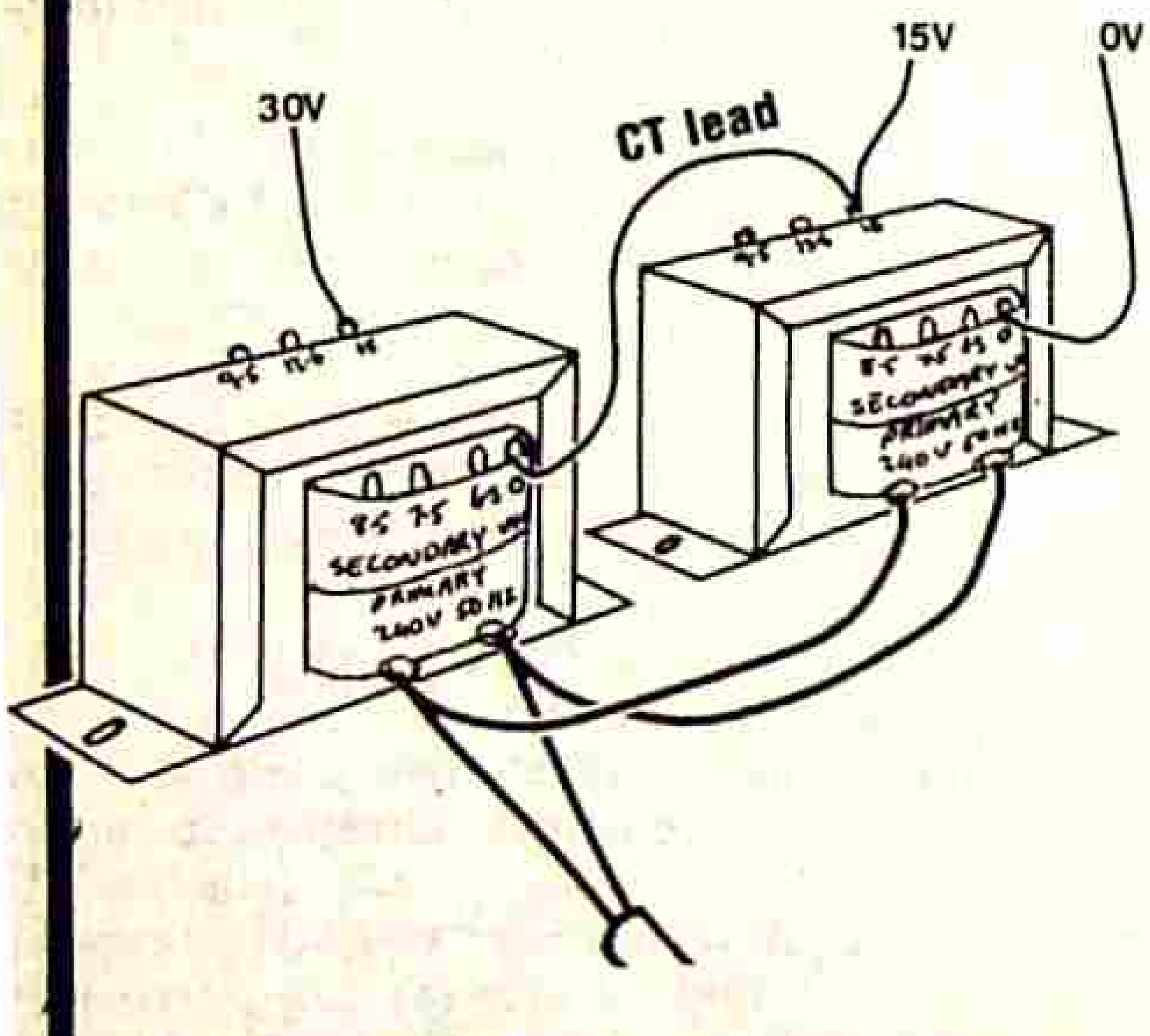
Although they are not ideal designs, the are the cheapest and most readily available.

The 2155 is a 15v types, rated at 1 amp. These are AC values and later in the text you will see how this current rating must be reduced for DC conditions.

The 6672 is a 30v type @1 amp. Again, these are AC values.

In effect, two 2155's are equal to one 6672 transformer.

When using two 2155's, they must be connected as shown in the diagram. This is called PHASING and will produce a transformer equal to a single 30v winding. When phasing is correct, the voltage between the 0v tap on one transformer and the 15v tap on the other will be 30v. If the phase is incorrect, the output voltage will be zero across these terminals.



PHASING TWO 2155 TRANSFORMERS

To produce the correct phasing, the 240v inputs on each transformer are connected in parallel and the secondary terminals are connected in series. We are assuming that the primary and secondary of both transformers are wound in the same direction.

HOW THE CIRCUIT WORKS

The 4 diodes form a full-wave bridge rectifier but the operation of this bridge and the transformer is a little different to normal.

If we take the case of plus and minus 5v using a single 2155 transformer, we can explain the operation as follows:

Take the positive half of the supply. The 7805 sees the transformer as two separate windings and receives a pulse of energy from one of the windings and one of the diodes near the 7805 during the first half of the AC cycle. During the second half of the cycle, the other winding supplies energy to the 7805 via one of the lower diodes.

At the same time, the alternate winding of the transformer is supplying power to the negative regulator via the other two diodes in the bridge.

This means both windings of the transformer are delivering throughout the cycle and two of the diodes are in use at any one time.

+15v −15v MODE

This is the most useful mode for this project. It provides and ideal positive and negative rail voltage for op-amps and other dual-rail chips.

To supply this voltage you will need two 2155's or a 6672 transformer. See the notes on phasing two 2155's to obtain the 0v - 15v - 30v outputs.

SINGLE +15v SUPPLY

In this mode a single 2155 transformer will be needed and one link must be made on the board as follows:
The two lower diodes are connected at their anodes and this point is connected to the CT hole on the board with a jumper wire.

The 0v and 15v tappings of the 2155 are connected to the other 2 input lines. The output is taken from the +ve and ground terminals.

SINGLE 5v MODE

This mode requires a single 2155 transformer plus the link as described in the previous mode. The 0v and 7.5v tapping on the 2155 are used for this mode to prevent the positive regulator overheating.

+5v −12v MODE

Whenever two different voltage regulators are used in this circuit, the lower voltage regulator will have a considerably reduced output current. This is due to the input voltage to the board being high to suit the other regulator. It is not possible to adjust the taps so say 7.5v for the lower voltage regulator and 12.5v for the high regulator. This is because the regulators take it in turns to receive a pulse of energy from each winding.

In this case the 12.5v winding on two 2155's must be used. This gives a very high voltage differential across the 5v regulator and will limit the output current to:

$$= \frac{2.5}{12.5\sqrt{2} - 5}$$

$$= \frac{2.5}{12} \text{Amp}$$

$$= 210\,\text{mA}$$

AN OVERALL LOOK AT THE PC BOARD

This dual power supply project uses a double-sided PC board. This enables the regulators to be bolted directly onto the PC board so that no additional heat-fins are required for currents up to 500mA.

This provides a slight saving in cost and leaves your supply of heatsinks for high current appplications.

The input to the board is designed for a centre-tapped transformer. You may not think a 2155 has a centre tap but this can be created by connecting the 7.5v tap to the CT connection on the PC board. When using a 6672, the centre voltage terminal is the 15v tap. The zero-volts and highest voltage tap are each taken to holes near the 'AC' identification. This gives us 3 input lines as found on a centre-tapped transformer.

The positive regulator goes in the 78xx position and the negative regulator in the 79xx position. They cannot be reversed or swapped over!

Nor will the board work with 2 positive or 2 negative regulators. You will notice the 'IN, COMMON and OUT' terminals are in different positions for the positive and negative regulators. This makes them non-operational in the wrong location.

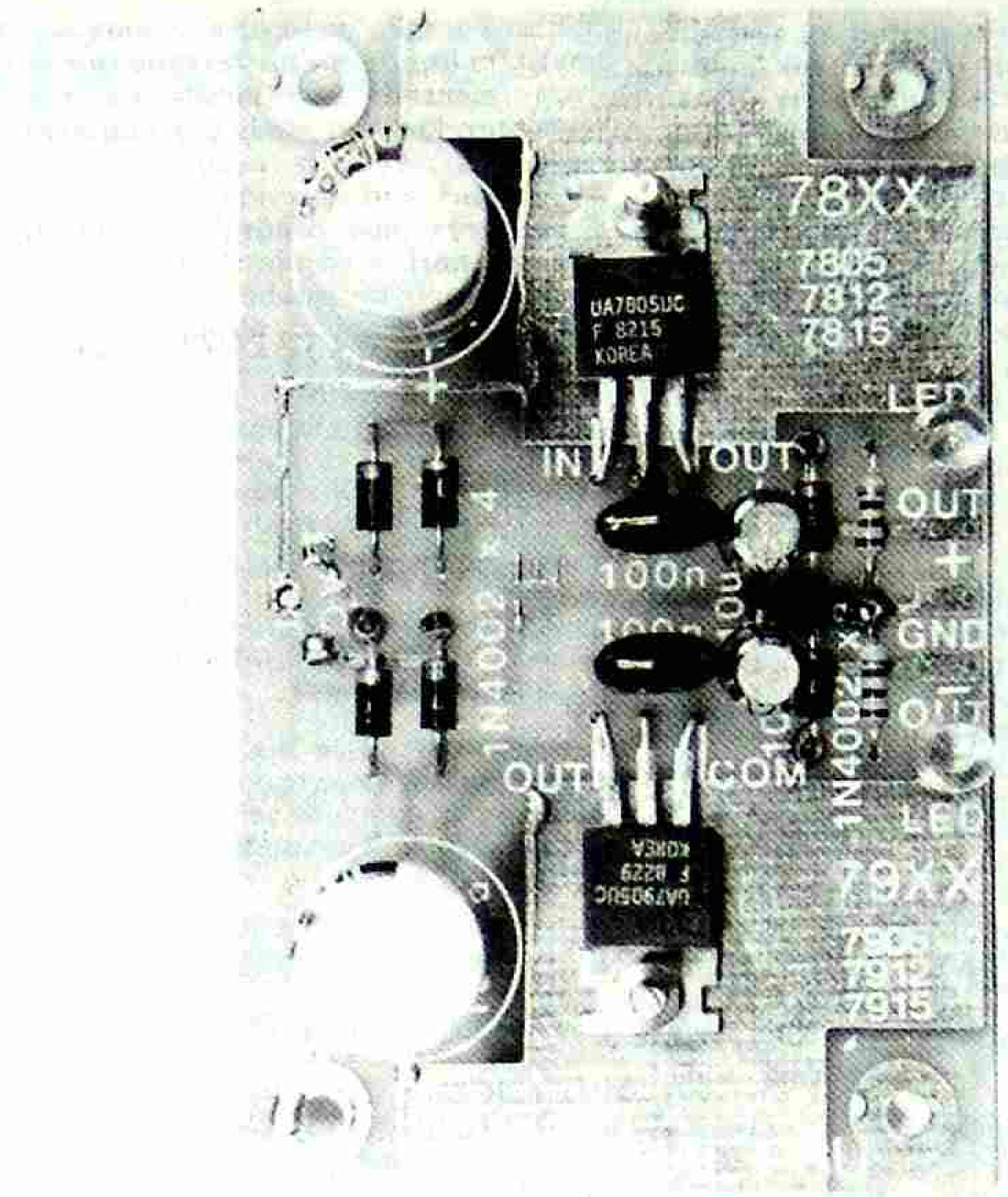
The 100n capacitor prevents high frequency oscillations from occurring. The 10mfd electrolytic on the output provides a small amount of filtering.

The resistor and LED provide an indication of the size of the output voltage and when the power is applied.

The two diodes across the output are extremely important. They provide short-circuit protection in case the positive lead is connected to the negative terminal. If this were to occur, the current limiting inside the regulators would not detect the short and both would be damaged if the diodes were not present.

The way they operate is very clever.

The lower diode becomes forward biased when the positive is connected to the negative and thus it forms a short-circuit to turn on the 7815. The same occurs with the upper diode and the 7915. Thus we save two regulators from being destroyed for the cost of a couple of diodes.



The complete layout with 7808 and 7905 fitted.

CURRENT RATING

We consider the maximum current is 600mA for each of the regulators. This maximum coincides with two other design factors. They are:
1. The maximum current rating of the transformer and the maximum wattage which can be safely dissipated by the regulator and its heat fin.

We obtained the max. dissipation by loading our prototype until the temp of the regulator reached max. We then reduced the current and re-tested the temp of the regulator. You can generally judge the correct temp of the regulator by feeling it with your fingers. You should be able to touch it for at least 5 seconds. At this temp, the output current was measured as well as the voltage across the input and output terminals. These are the results:
Input voltage: 9v
Output voltage: 5v
Current: 625mA.

This gives (9 - 5) x .625 = 2.5 watts. This is the power lost in the regulator and it must be transferred to the heat fin to prevent the regulator from over heating.

The other factor which determines the maximum current available at the output of a power supply is the rating of the transformer. We hear so much about the rating of a 2155 as being 15v at 1 amp. But little do we realize this rating is an AC value.

Working out the DC current rating is quite complex and we will approach it in a simple way.

The power rating of a 2155 is obtained by multiplying the secondary volts by the current. This gives 15v x 1 amp = 15VA. This is 15 volt-amps but for a simple discussion we can call it 15 watts. This is the power rating of the transformer.

Now let's see what happens when we connect the transformer to a full-wave bridge rectifier circuit. The DC voltage which appears across the bridge circuit is $15\sqrt{2}$ volts DC. We readily accept and appreciate the higher voltage which appears across these circuits but fail to take into account the fact that the current rating must be reduced by the same ratio to maintain the value of 15 watts.

This means the output current must be reduced by the ratio:

$$\frac{15}{15\sqrt{2}}$$

This gives us 700mA max for the DC condition. This is a far cry from the 1 amp we so readily accept and expect.

The transformer will deliver higher currents if the load is increased however the output voltage will fall with the result that the regulator may drop out of regulation.

This will means you will have to use a higher voltage tapping and the energy lost in the heatsink will increase dramatically.

When using the 7.5v tapping and drawing 600mA, this project will operate very reliably for long periods of time.

HEATSINKING

We have determined that the maximum power dissipation for each rail of the power supply will be 2.5 watts. When this amount of energy is being dissipated by the regulator and PC heatsink, the regulator is still cool enough to be touched on its plastic case with your finger.

Because the copper fin on the PC board is very thin, it is essential that the regulator be adequately heat-sinked. This involves filing the surface of the board to remove any of the bumps and the use of heat-sinking compound. This compound is absolutely necessary if you wish to run the regulator for longer than 10 minutes at greater than 100mA.

The nut and bolt used for the regulator must be brass or steel to conduct the heat to the other side of the PC board.

If you wish to increase the dissipation, an additional heatfin will be required. The critical part of the fin is the thickness of the material between the regulator and the PC board. This part has the greatest effect on conducting the heat from the regulator to the free air. (Free air is normal air. You can also have fan forced air). A piece of aluminium bent into an L shape will be suitable provided it offers a flat surface to the regulator and PC board. The maximum dissipation with this type of arrangement is 5 - 6 watts.

CONSTRUCTION

The first components to be mounted are the two regulators. File the surface of the PC board to remove any high spots. Place a very small amount of heat conducting grease on the back of the positive regulator and fit its three leads into the holes next to the 78xx identification on the PC board. Use a 1/8" or 6BA nut and bolt to tighten the regulator onto the board. Watch the grease ooze out from around the sides of the regulator. Repeat with the negative regulator.

The board is now ready for mounting the rest of the components. All the parts should be pushed firmly onto the board so that they almost touch it. We still see projects sent in by readers with parts high above the board. And it always looks very messy. In this project there is an exception. The two 1000mfd electrolytics can be kept slightly above the board so that the heat from the fin is not conducted into the electrolytic itself. If they were to get too hot, they would dry out and lose their capacitance.

Solder the lands on the top of the PC board. The last items to connect are three leads to the input (say yellow, white and blue) and three leads to the output (red, black and green).

Mount the board on 4 standoffs and it is now ready to be tested.

TESTING THE UNIT

We will assume you have used the most common voltage regulators giving a +15v and +15v supply, however the same procedure will apply to any arrangement.

Firstly the power supply is tested under NO LOAD conditions. Place a 47ohm or 100ohm ¼ watt resistor in each of the two outside input lines. Make sure the 3 output leads are not shorting together.

Switch on the AC and note the two output indicator LEDs will illuminate. The two protection resistors should also remain cold throughout this test. If they get hot or burn out, a fault will exist. Fix the fault immediately.

Remove the two protection resistors and wire the input directly to the transformer.

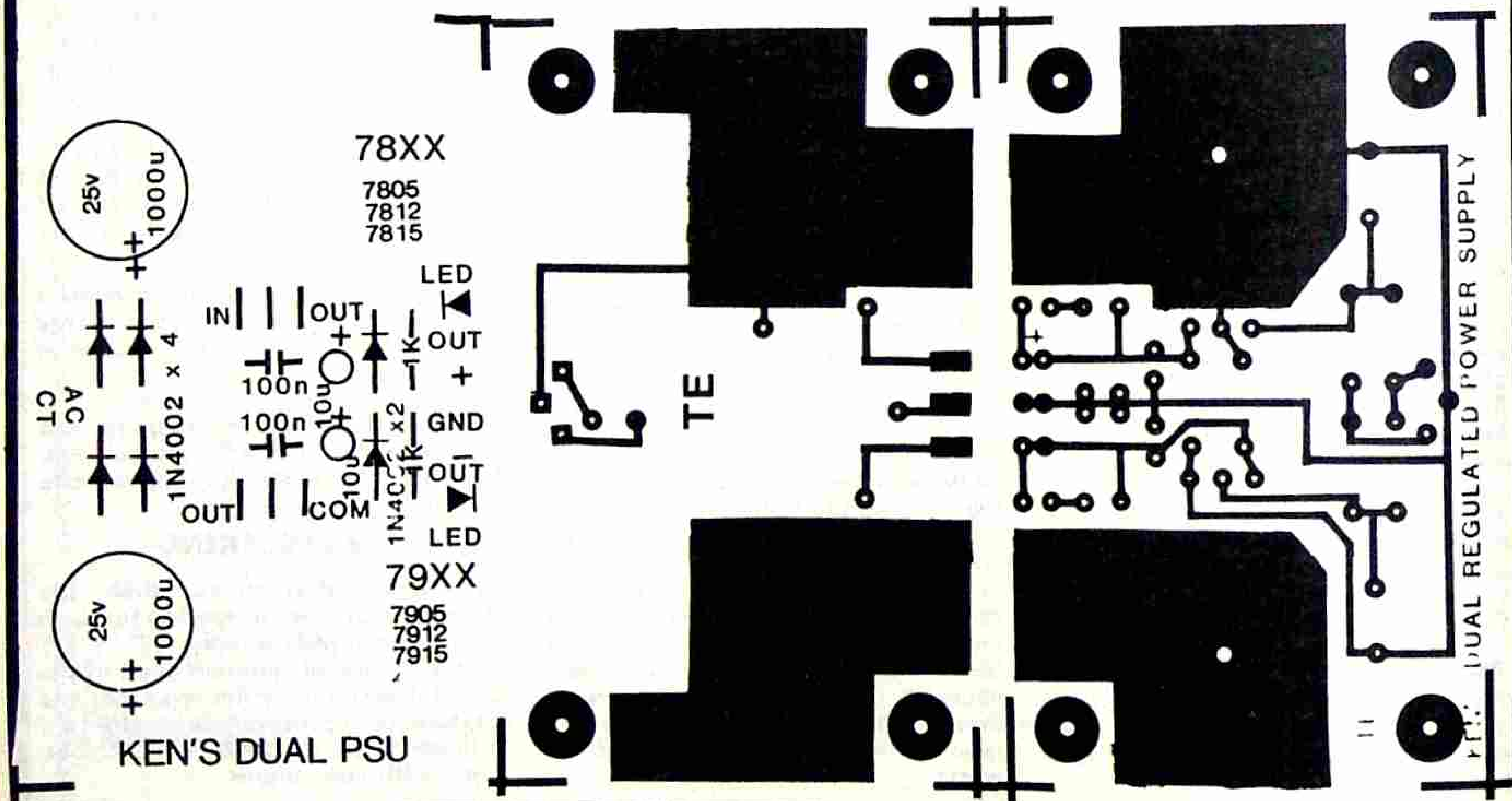
The best way of determining the current capability of the power supply is to connect a specific load to the output. The most accurate arrangement is to use one or more wire wound resistors to dissipate the heat. If you have 8R2 10R or 15R in either 5watt or 10watt sizes, they will be ideal. Globes or motors do not give a very reliable indication of the current flow as they draw a varying current according to the load or brightness. And the current will vary enormously with only a slight change in voltage.

For a 15v regulator, you can use three 8R2 resistors in series for an accurate determination of the 7815 current and a globe on the other regulator for an approximate load.

At 600mA, the regulators can still be touched with a finger while the nut and bolt is a better conductor and will be too hot to touch.

If everything is working correctly, the test load can be removed and the PSU connected to a project.

I hope you find it as handy as we have.



700

Z80 Machine Codes

SHEET 11

This table contains over 700 Machine Code instructions for the Z80.
It has been compiled from Zilog Data sheets, SGS Data books, Z80 Programming by P. Levison (now out of print) and Micro-Professor Programming Handbooks.
Two books to help with the interpretation of this table are: **Z80 ASSEMBLY LANGUAGE** by Lance A. Leventhal. (Mc Graw Hill). and **PROGRAMMING the Z80** by Rodnay Zaks. (Sybex).

ADC A,(HL)	8E
ADC A,(IX+dis)	DD 8E XX
ADC A,(IY+dis)	FD 8E XX
ADC A,A	8F
ADC A,B	88
ADC A,C	89
ADC A,D	8A
ADC A,dd	CE dd
ADC A,E	8B
ADC A,H	8C
ADC A,L	8D
ADC HL,BC	ED 4A
ADC HL,DE	ED 5A
ADC HL,HL	ED 6A
ADC HL,SP	ED 7A
ADD A,(HL)	86
ADD A,(IX+dis)	DD 86 XX
ADD A,(IY+dis)	FD 86 XX
ADD A,A	87
ADD A,B	80
ADD A,C	81
ADD A,D	82
ADD A,dd	C6 dd
ADD A,E	83
ADD A,H	84
ADD A,L	85
ADD HL,BC	09
ADD HL,DE	19
ADD HL,HL	29
ADD HL,SP	39
ADD IX,BC	DD 09
ADD IX,DE	DD 19
ADD IX,IX	DD 29
ADD IX,SP	DD 39
ADD IY,BC	FD 09
ADD IY,DE	FD 19
ADD IY,IY	FD 29
ADD IY,SP	FD 39
AND (HL)	A6
AND (IX+dis)	DD A6 XX
AND (IY+dis)	FD A6 XX
AND A	A7
AND B	A0
AND C	A1
AND D	A2
AND dd	E6 dd
AND E	A3
AND H	A4
AND L	A5
BIT 0,(HL)	CB 46
BIT 0,(IX+dis)	DD CB XX 46
BIT 0,(IY+dis)	FD CB XX 46
BIT 0,A	CB 47
BIT 0,B	CB 40
BIT 0,C	CB 41
BIT 0,D	CB 42
BIT 0,E	CB 43
BIT 0,H	CB 44
BIT 0,L	CB 45
BIT 1,(HL)	CB 4E
BIT 1,(IX+dis)	DD CB XX 4E
BIT 1,(IY+dis)	FD CB XX 4E
BIT 1,A	CB 4F
BIT 1,B	CB 48
BIT 1,C	CB 49
BIT 1,D	CB 4A
BIT 1,E	CB 4B
BIT 1,H	CB 4C
BIT 1,L	CB 4D
BIT 2,(HL)	CB 56
BIT 2,(IX+dis)	DD CB XX 56
BIT 2,(IY+dis)	FD CB XX 56
BIT 2,A	CB 57
BIT 2,B	CB 50
BIT 2,C	CB 51
BIT 2,D	CB 52
BIT 2,E	CB 53
BIT 2,H	CB 54
BIT 2,L	CB 55
BIT 3,(HL)	CB 5E
BIT 3,(IX+dis)	DD CB XX 5E
BIT 3,(IY+dis)	FD CB XX 5E
BIT 3,A	CB 5F
BIT 3,B	CB 58
BIT 3,C	CB 59
BIT 3,D	CB 5A
BIT 3,E	CB 5B
BIT 3,H	CB 5C
BIT 3,L	CB 5D
BIT 4,(HL)	CB 66
BIT 4,(IX+dis)	DD CB XX 66
BIT 4,(IY+dis)	FD CB XX 66
BIT 4,A	CB 67
BIT 4,B	CB 60
BIT 4,C	CB 61
BIT 4,D	CB 62
BIT 4,E	CB 63
BIT 4,H	CB 64
BIT 4,L	CB 65
BIT 5,(HL)	CB 6E
BIT 5,(IX+dis)	DD CB XX 6E
BIT 5,(IY+dis)	FD CB XX 6E

BIT 5,A	CB 6F
BIT 5,B	CB 68
BIT 5,C	CB 69
BIT 5,D	CB 6A
BIT 5,E	CB 6B
BIT 5,H	CB 6C
BIT 5,L	CB 6D
BIT 6,(HL)	CB 76
BIT 6,(IX+dis)	DD CB XX 76
BIT 6,(IX+dis)	FD CB XX 76
BIT 6,A	CB 77
BIT 6,B	CB 70
BIT 6,C	CB 71
BIT 6,D	CB 72
BIT 6,E	CB 73
BIT 6,H	CB 74
BIT 6,L	CB 75
BIT 7,(HL)	CB 7E
BIT 7,(IX+dis)	DD CB XX 7E
BIT 7,(IY+dis)	FD CB XX 7E
BIT 7,A	CB 7F
BIT 7,B	CB 78
BIT 7,C	CB 79
BIT 7,D	CB 7A
BIT 7,E	CB 7B
BIT 7,H	CB 7C
BIT 7,L	CB 7D
CALL ADDR	CD XX XX
CALL C,ADDR	DC XX XX
CALL M,ADDR	FC XX XX
CALL NC,ADDR	D4 XX XX
CALL NZ,ADDR	C4 XX XX
CALL P,ADDR	F4 XX XX
CALL PE,ADDR	EC XX XX
CALL PO,ADDR	E4 XX XX
CALL Z,ADDR	CC XX XX
CCF	3F
CP (HL)	BE
CP (IX+dis)	DD BE XX
CP (IY+dis)	FD BE XX
CP A	BF
CP B	B8
CP C	B9
CP D	BA
CP dd	FE dd
CP E	BB
CP H	BC
CP L	BD
CPD	ED A9
CPDR	ED B9
CPI	ED A1
CPIR	ED B1
CPL	2F
DAA	27
DEC (HL)	35
DEC (IX+dis)	DD 35 XX
DEC (IY+dis)	FD 35 XX
DEC A	3D
DEC B	05
DEC BC	0B
DEC C	0D
DEC D	15
DEC DE	1B
DEC E	1D
DEC H	25
DEC HL	2B
DEC IX	DD 2B
DEC IY	FD 2B
DEC L	2D
DEC SP	3B
DI	F3
DJNZ dis	10 XX
EI	FB
EX (SP),HL	E3
EX (SP),IX	DD E3
EX (SP),IY	FD E3
EX AF,AF	08
EX DE,HL	EB
EXX	D9
HALT	76
IM 0	ED 46
IM 1	ED 56
IM 2	ED 5E
IN A,(C)	ED 78
IN A,port	DB XX
IN B,(C)	ED 40
IN C,(C)	ED 48
IN D,(C)	ED 50
IN E,(C)	ED 58
IN H,(C)	ED 60
IN L,(C)	ED 68
INC (HL)	34
INC (IX+dis)	DD 34 XX
IBC (IY+dis)	FD 34 XX
INC A	3C
INC B	04
INC BC	03
INC C	0C
INC D	14
INC DE	13
INC E	1C
INC H	24
INC HL	23
INC IX	DD 23
INC IY	FD 23
INC L	2C
INC SP	33
IND	ED AA
INDR	ED BA
INI	ED A2
INIR	ED B2
JP (HL)	E9
JP (IX)	DD E9
JP (IY)	FD E9
JP ADDR	C3 XX XX
JP C,ADDR	DA XX XX
JP M,ADDR	FA XX XX
JP NC,ADDR	D2 XX XX
JP NZ,ADDR	C2 XX XX

JP P,ADDR	F2 XX XX
JP PE,ADDR	EA XX XX
JP PO,ADDR	E2 XX XX
JP Z,ADDR	CA XX XX
JR C,dis	38 XX
JR dis	18 XX
JR NC,dis	30 XX
JR NZ,dis	20 XX
JR Z,dis	28 XX
LD (ADDR),A	32 XX XX
LD (ADDR),BC	ED 43 XX XX
LD (ADDR),DE	ED 53 XX XX
LD (ADDR),HL	ED 63 XX XX
LD (ADDR),HL	22 XX XX
LD (ADDR),IX	DD 22 XX XX
LD (ADDR),IY	FD 22 XX XX
LD (ADDR),SP	ED 73 XX XX
LD (BC),A	02
LD (DE),A	12
LD (HL),A	77
LD (HL),B	70
LD (HL),C	71
LD (HL),D	72
LD (HL),dd	36 dd
LD (HL),E	73
LD (HL),H	74
LD (HL),L	75
LD (IX+dis),A	DD 77 XX
LD (IX+dis),B	DD 70 XX
LD (IX+dis),C	DD 71 XX
LD (IX+dis),D	DD 72 XX
LD (IX+dis),dd	DD 36 XX dd
LD (IX+dis),E	DD 73 XX
LD (IX+dis),H	DD 74 XX
LD (IX+dis),L	DD 75 XX
LD (IY+dis),A	FD 77 XX
LD (IY+dis),B	FD 70 XX
LD (IY+dis),C	FD 71 XX
LD (IY+dis),D	FD 72 XX
LD (IY+dis),dd	FD 36 XX dd
LD (IY+dis),E	FD 73 XX
LD (IY+dis),H	FD 74 XX
LD (IY+dis),L	FD 75 XX
LD A,(ADDR)	3A XX XX
LD A,(BC)	0A
LD A,(DE)	1A
LD A,(HL)	7E
LD A,(IX+dis)	DD 7E XX
LD A,(IY+dis)	FD 7E XX
LD A,A	7F
LD A,B	78
LD A,C	79
LD A,D	7A
LD A,dd	3E dd
LD A,E	7B
LD A,H	7C
LD A,I	ED 57
LD A,L	7D
LD A,R	ED 5F
LD B,(HL)	46
LD B,(IX+dis)	DD 46 XX
LD B,(IY+dis)	FD 46 XX
LD B,A	47
LD B,B	40
LD B,C	41
LD B,D	42
LD B,dd	06 dd
LD B,E	43
LD B,H	44
LD B,L	45
LD BC,(ADDR)	ED 4B XX XX
LD BC,dd	01 dd dd
LD C,(HL)	4E
LD C,(IX+dis)	DD 4E XX
LD C,(IY+dis)	FD 4E XX
LD C,A	4F
LD C,B	48
LD C,C	49
LD C,D	4A
LD C,dd	0E dd
LD C,E	4B
LD C,H	4C
LD C,L	4D
LD D,(HL)	56
LD D,(IX+dis)	DD 56 XX
LD D,(IY+dis)	FD 56 XX
LD D,A	57
LD D,B	50
LD D,C	51
LD D,D	52
LD D,dd	16 dd
LD D,E	53
LD D,H	54
LD D,L	55
LD DE,(ADDR)	ED 5B XX XX
LD DE,dddd	11 dd dd
LD E,(HL)	5E
LD E,(IX+dis)	DD 5E XX
LD E,(IY+dis)	FD 5E XX
LD E,A	5F
LD E,B	58
LD E,C	59
LD E,D	5A
LD E,dd	1E dd
LD E,E	5B
LD E,H	5C
LD E,L	5D
LD H,(HL)	66
LD H,(IX+dis)	DD 66 XX
LD H,(IY+dis)	FD 66 XX
LD H,A	67
LD H,B	60
LD H,C	61
LD H,D	62
LD H,dd	26 dd
LD H,E	63
LD H,H	64
LD H,L	65
LD HL,(ADDR)	ED 6B XX XX

LD HL,(ADDR)	2A XX XX
LD HL, dddd	21 dd dd
LD I,A	ED 47
LD IX,(ADDR)	DD 2A XX XX
LD IX,dddd	DD 21 dd dd
LD IY,(ADDR)	FD 2A XX XX
LD IY,dddd	FD 21 dd dd
LD L,(HL)	6E
LD L,(IX+dis)	DD 6E XX
LD L,(IY+dis)	FD 6E XX
LD L,A	6F
LD L,B	68
LD L,C	69
LD L,D	6A
LD L,dd	2A dd
LD L,E	6B
LD L,H	6C
LD L,L	6D
LD R,A	ED 4F
LD SP,(ADDR)	ED 7B XX XX
LD SP,dddd	31 dd dd
LD SP,HL	F9
LD SP,IX	DD F9
LD SP,IY	FD F9
LDD	ED A8
LDDR	ED B8
LDI	ED A0
LDIR	ED B0
NEG	ED 44
NOP	00
OR (HL)	B6
OR (IX+dis)	DD B6 XX
OR (IY+dis)	FD B6 XX
OR A	B7
OR B	B0
OR C	B1
OR D	B2
OR dd	F6 dd
OR E	B3
OR H	B4
OR L	B5
OTDR	ED BB
OTIR	ED B3
OUT (C),A	ED 79
OUT (C),B	ED 41
OUT (C),C	ED 49
OUT (C),D	ED 51
OUT (C),E	ED 59
OUT (C),H	ED 61
OUT (C),L	ED 69
OUT port,A	D3 port
OUTD	ED AB
OUTI	ED A3
POP AF	F1
POP BC	C1
POPDE	D1
POPHL	E1
POPIX	DD E1
POPIY	FD E1
PUSH AF	F5
PUSH BC	C5
PUSH DE	D5
PUSH HL	E5
PUSH IX	DD E5
PUSH IY	FD E5
RES 0,(HL)	CB 86
RES 0,(IX+dis)	DD CB XX 86
RES 0,(IY+dis)	FD CB XX 86
RES 0,A	CB 87
RES 0,B	CB 80
RES 0,C	CB 81
RES 0,D	CB 82
RES 0,E	CB 83
RES 0,H	CB 84
RES 0,L	CB 85
RES 1,(HL)	CB 8E
RES 1,(IX+dis)	DD CB XX 8E
RES 1,(IY+dis)	FD CB XX 8E
RES 1,A	CB 8F
RES 1,B	CB 88
RES 1,C	CB 89
RES 1,D	CB 8A
RES 1,E	CB 8B
RES 1,H	CB 8C
RES 1,L	CB 8D
RES 2,(HL)	CB 96
RES 2,(IX+dis)	DD CB XX 96
RES 2,(IY+dis)	FD CB XX 96
RES 2,A	CB 97
RES 2,B	CB 90
RES 2,C	CB 91
RES 2,D	CB 92
RES 2,E	CB 93
RES 2,H	CB 94
RES 2,L	CB 95
RES 3,(HL)	CB 9E
RES 3,(IX+dis)	DD CB XX 9E
RES 3,(IY+dis)	FD CB XX 9E
RES 3,A	CB 9F
RES 3,B	CB 98
RES 3,C	CB 99
RES 3,D	CB 9A
RES 3,E	CB 9B
RES 3,H	CB 9C
RES 3,L	CB 9D
RES 4,(HL)	CB A6
RES 4,(IX+dis)	DD CB XX A6
RES 4,(IY+dis)	FD CB XX A6
RES 4,A	CB A7
RES 4,B	CB A0
RES 4,B	CB A1
RES 4,C	CB A2
RES 4,D	CB A3
RES 4,E	CB A4
RES 4,H	CB A5
RES 5,(HL)	CB AE
RES 5,(IX+dis)	DD CB XX AE
RES 5,(IY+dis)	FD CB XX AE
RES 5,A	CB AF

RES 5,B	CB A8
RES 5,C	CB A9
RES 5,D	CB AA
RES 5,E	CB AB
RES 5,H	CB AC
RES 5,L	CB AD
RES 6,(HL)	CB B6
RES 6,(IX+dis)	DD CB XX B6
RES 6,(IY+dis)	FD CB XX B6
RES 6,A	CB B7
RES 6,B	CB B0
RES 6,C	CB B1
RES 6,D	CB B2
RES 6,E	CB B3
RES 6,H	CB B4
RES 6,L	CB B5
RES 7,(HL)	CB BE
RES 7,(IX+dis)	DD CB XX BE
RES 7,(IY+dis)	FD CB XX BE
RES 7,A	CB BF
RES 7,B	CB B8
RES 7,C	CB B9
RES 7,D	CB BA
RES 7,E	CB BB
RES 7,H	CB BC
RES 7,L	CB BD
RET	C9
RET C	D8
RET M	F8
RET NC	D0
RET NZ	C0
RET P	F0
RET PE	E8
RET PO	E0
RET Z	C8
RETI	ED 4D
RETN	ED 45
RL (HL)	CB 16
RL (IX+dis)	DD CB XX 16
RL (IY+dis)	FD CB XX 16
RL A	CB 17
RL B	CB 10
RL C	CB 11
RL D	CB 12
RL E	CB 13
RL H	CB 14
RL L	CB 15
RLA	17
RLC (HL)	CB 06
RLC (IX+dis)	DD CB XX 06
RLC (IY+dis)	FD CB XX 06
RLC A	CB 07
RLC B	CB 00
RLC C	CB 01
RLC D	CB 02
RLC E	CB 03
RLC H	CB 04
RLC L	CB 05
RLCA	07
RLD	ED 67
RR (HL)	CB 1E
RR (IX+dis)	DD CB XX 1E
RR (IY+dis)	FD CB XX 1E
RR A	CB 1F
RR B	CB 18
RR C	CB 19
RR D	CB 1A
RR E	CB 1B
RR H	CB 1C
RR L	CB 1D
RRA	1F
RRC (HL)	CB 0E
RRC (IX+dis)	DD CB XX 0E
RRC (IY+dis)	FD CB XX 0E
RRC A	CB 0F
RRC B	CB 08
RRC C	CB 09
RRC D	CB 0A
RRC E	CB 0B
RRC H	CB 0C
RRC L	CB 0D
RRCA	0F
RRD	ED 6F
RST 00	C7
RST 08	CF
RST 10	D7
RST 18	DF
RST 20	E7
RST 28	EF
RST 30	F7
RST 38	FF
SBC A,(HL)	9E
SBC A,(IX+dis)	DD 9E XX
SBC A,(IY+dis)	FD 9E XX
SBC A,A	9F
SBC A,B	98
SBC A,C	99
SBC A,D	9A
SBC A,dd	DE dd
SBC A,E	9B
SBC A,H	9C
SBC A,L	9D
SBC HL,BC	ED 42
SBC HL,DE	ED52
SBC HL,HL	ED 62
SBC HL,SP	ED 72
SCF	37
SET 0,(HL)	CB C6
SET 0,(IX+dis)	DD CB XX C6
SET 0,(IY+dis)	FD CB XX C6
SET 0,A	CB C7
SET 0,B	CB C0
SET 0,C	CB C1
SET 0,D	CB C2
SET 0,E	CB C3
SET 0,H	CB C4
SET 0,L	CB C5
SET 1,(HL)	CB CE
SET 1,(IX+dis)	DD CB XX CE

SET 1,(IY+dis)	FD CB XX CE
SET 1,A	CB CF
SET 1,B	CB C8
SET 1,C	CB C9
SET 1,D	CB CA
SET 1,E	CB CB
SET 1,H	CB CC
SET 1,L	CB CD
SET 2,(HL)	CB D6
SET 2,(IX+dis)	DD CB XX D6
SET 2,(IY+dis)	FD CB XX D6
SET 2,A	CB D7
SET 2,B	CB D0
SET 2,C	CB D1
SET 2,D	CB D2
SET 2,E	CB D3
SET 2,H	CB D4
SET 2,L	CB D5
SET 3,(HL)	CB DE
SET 3,(IX+dis)	DD CB XX DE
SET 3,(IY+dis)	FD CB XX DE
SET 3,A	CB DF
SET 3,B	CB D8
SET 3,C	CB D9
SET 3,D	CB DA
SET 3,E	CB DB
SET 3,H	CB DC
SET 3,L	CB DD
SET 4,(HL)	CB E6
SET 4,(IX+dis)	DD CB XX E6
SET 4,(IY+dis)	FD CB XX E6
SET 4,A	CB E7
SET 4,B	CB E0
SET 4,C	CB E1
SET 4,D	CB E2
SET 4,E	CB E3
SET 4,H	CB E4
SET 5,(HL)	CB EE
SET 5,(IX+dis)	DD CB XX EE
SET 5,(IY+dis)	FD CB XX EE
SET 5,A	CB EF
SET 5,B	CB E8
SET 5,C	CB E9
SET 5,D	CB EA
SET 5,E	CB EB
SET 5,H	CB EC
SET 5,L	CB ED
SET 6,(HL)	CB F6
SET 6,(IX+dis)	DD CB XX F6
SET 6,(IY+dis)	FD CB XX F6
SET 6,A	CB F7
SET 6,B	CB F0
SET 6,C	CB F1
SET 6,D	CB F2
SET 6,E	CBF3
SET 6,H	CB F4
SET 6,L	CB F5
SET 7,(HL)	CB FE
SET 7,(IX+dis)	DD CB XX FE
SET 7,(IY+dis)	FD CB XX FE
SET 7,A	CB FF
SET 7,B	CB F8
SET 7,C	CB F9
SET 7,D	CB FA
SET 7,E	CB FB
SET 7,H	CB FC
SET 7,L	CB FD
SLA (HL)	CB 26
SLA (IX+dis)	DD CB XX 26
SLA (IY+dis)	FD CB XX 26
SLA A	CB 27
SLA B	CB 20
SLA C	CB 21
SLA D	CB 22
SLA E	CB 23
SLA H	CB 24
SLA L	CB 25
SRA (HL)	CB 2E
SRA (IX+dis)	DD CB XX 2E
SRA (IY+dis)	FD CB XX 2E
SRA A	CB 2F
SRA B	CB 28
SRA C	CB 29
SRA D	CB 2A
SRA E	CB 2B
SRA H	CB 2C
SRA L	CB 2D
SRL (HL)	CB 3E
SRL (IX+dis)	DD CB XX 3E
SRL (IY+dis)	FD CB XX 3E
SRL A	CB 3F
SRL B	CB 38
SRL C	CB 39
SRL D	CB 3A
SRL E	CB 3B
SRL H	CB 3C
SRL L	CB 3D
SUB (HL)	96
SUB (IX+dis)	DD 96 XX
SUB (IY+dis)	FD 96 XX
SUB A	97
SUB B	90
SUB C	91
SUB D	92
SUB dd	D6 dd
SUB E	93
SUB H	94
SUB L	95
XOR (HL)	AE
XOR (IX+dis)	DD AE XX
XOR (IY+dis)	FD AE XX
XOR A	AF
XOR B	A8
XOR C	A9
XOR D	AA
XOR dd	EE dd
XOR E	AB
XOR H	AC
XOR L	AD

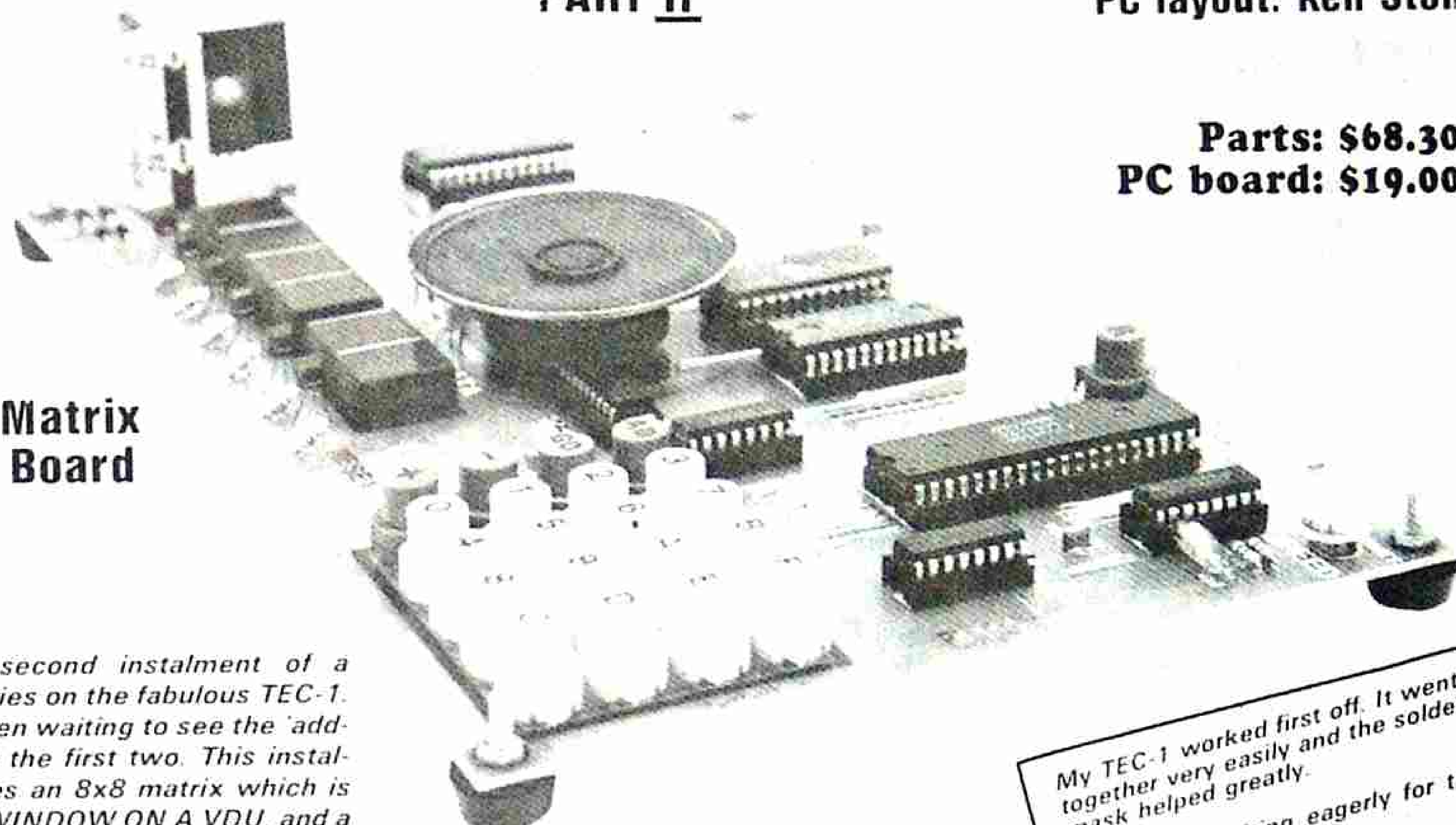
TEC-1

TALKING ELECTRONICS COMPUTER

PART II

—by John Hardy
PC layout: Ken Stone.

Parts: $68.30
PC board: $19.00



★ 8 x 8 Matrix
★ Relay Board

This is the second instalment of a continuing series on the fabulous TEC-1. If you have been waiting to see the 'add-ons', here are the first two. This instalment describes an 8x8 matrix which is effectively a WINDOW ON A VDU, and a RELAY BOARD which contains a set of 8 relays so that the TEC-1 will access the outside world.

You can operate globes or motors via the relays or drive them directly via the set of transistors included on the board.

The 8x8 matrix is multiplexed and driven by its own set of latches. In the ultimate you will be able to get incredible movement, but in the elementary stage its simple illumination and shift patterns.

Now that you have got this far, read on.. .. .. ..

The introduction of the TEC-1 in the previous issue caused quite a lot of interest from a new group of hobbyists. Was this due to the colour cover or the presentation of a cheap computer? Who knows?

In any case, we are pleased it took their attention. Everyone will benefit with the increased sales it produced.

We noted the number of orders increased dramatically with many coming from names and places not on our mailing list or files.

The requests for TEC-1 outstripped the availability of kits and we soon realized the small markets in Australia had to be by-passed in preference to direct importing.

Sales are still peaking but I think many readers are still waiting for the full range of "add-ons" before launching into purchasing a kit. Let's hope some of your answers will be answered when you see the extent of the projects in this issue. And this is only just the beginning.

We have already designed more than 9 different expansions for the TEC-1 and this will take it into the field of a fully-fledged demonstrator.

Within the first week we received 5 letters from constructors who had the TEC-1 operating from the instant of switch-on.

Although extremely simple, the TEC-1 works very well. Some of its features are novel while others are a little outdated. The speed control is a novelty while the 8212's have been around for years and are now getting towards the end of production. We found this out as they are now quite

. . .cont. P. 14.

My TEC-1 worked first off. It went together very easily and the solder mask helped greatly.

I am now waiting eagerly for the next installment.

Steven Truscott, 2287

When the TEC-1 was introduced in issue 10, my son and I agreed it should be a good place to learn about computers. We built the kit and it worked straight away. We were quite impressed by the quality of the PC board and the technical details in issue 10.

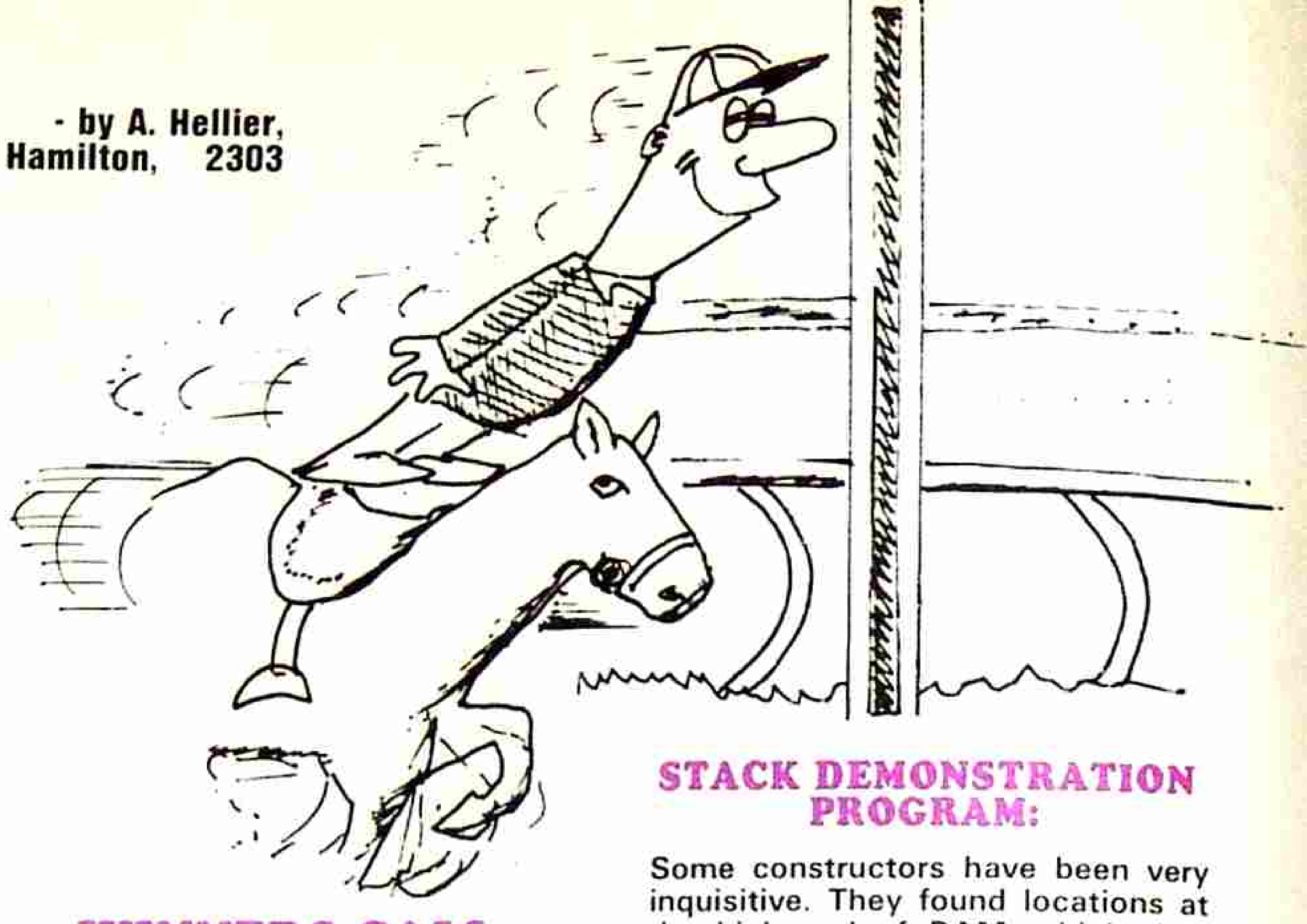
We are now in the process of making a case for the TEC-1 and are in complete agreement that the computer should be exposed so that we keep in touch with the "operations". We will be fitting a hinged perspex cover to keep out dust etc. The only problem is the heatsink on the 7805. We have decided to mount the regulator under the board, near one corner and run three lines to the appropriate lands. Everything will then be neat, firm and tamper-proof.

Martin Hulsman, 7310.

. . . In the expansion port on the TEC-1 I would mount one of those IC sockets with a little lever on it. They are expensive but make it easier to remove the expansion plugs.

Raymond Green.

- by A. Hellier, Hamilton, 2303



BIG BEN CHIMES

RESET 00 + 09 + 3E + 00 + 32 + 00 + 08 + 3E + 09 + 32 + 01 + 08 + CD + BO + 01 + 3E + 0A + 32 + 01 + 08 + CD + 70 + 02 + C3 + 02 + 08.

ADDRESS: 0900 + 11 + OD + OF + 08 + 00 + 00 + 00 + 00 + 08 + OF + 11 + OD + 00 + 00 + 00 + 00 + 11 + OF + OD + 08 + 00 + 00 + 00 + 00 + 08 + OF + 11 + OD + 00 + 00 + 00 + 00 + OD + 00 + 00 + 00 + 00 + OD + 00 + 00 + 00 + 00 + OD + 00 + 00 + 00 + 00 OD + 1F.

ADDRESS OAOO + 02 + 09 + 07 + 00 + 02 + 05 + OD + 00 + 00 + 00 + 00 + 00 + 00 + 00 + 1F.

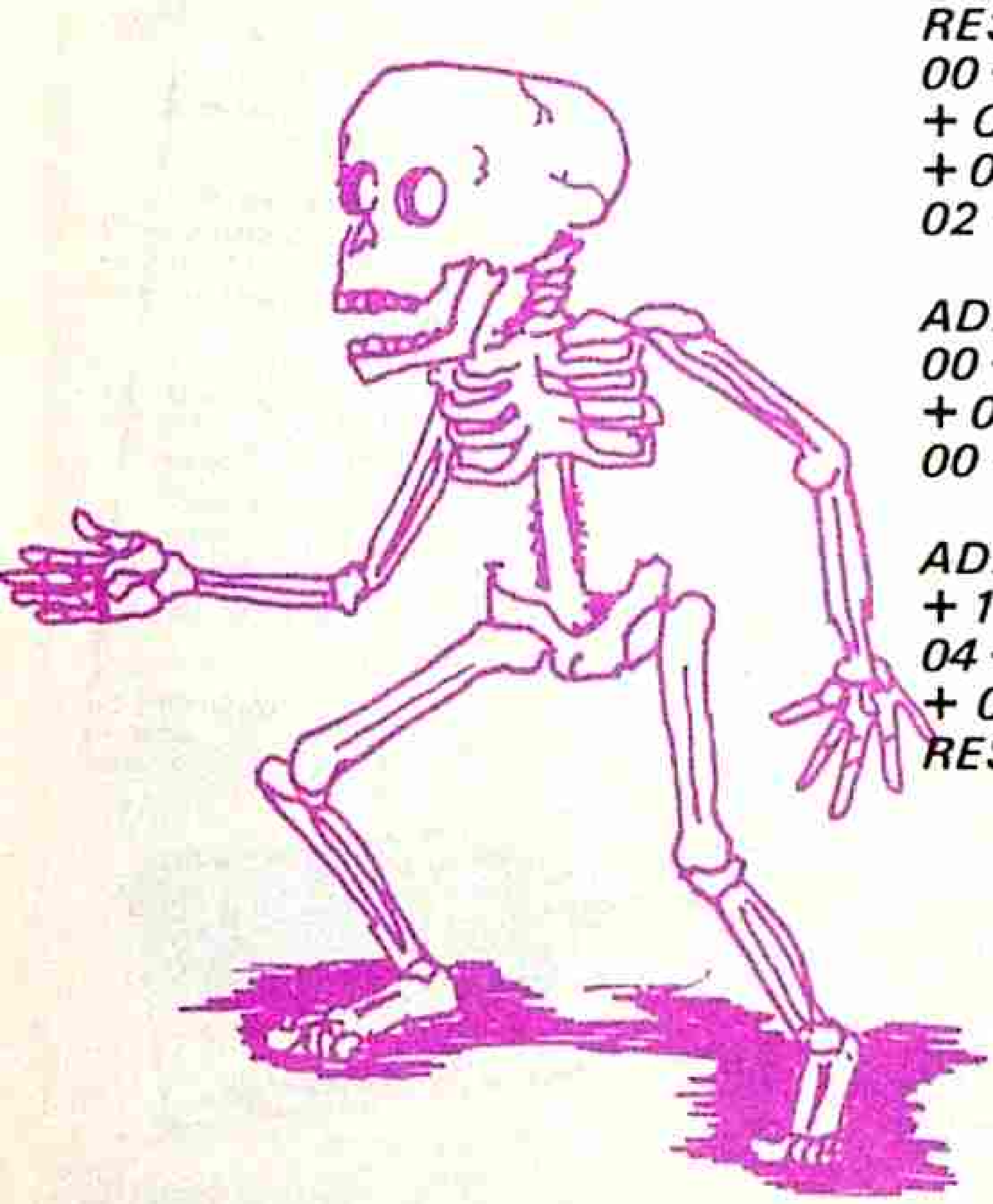
RESET + + GO GO.

WINNERS CALL

RESET + 02 + 08 + 06 + 0A + OD + 12 + 12 + 12 + OD + OD + OD + OA + OD + OA + 06 + 00 + 06 + OA + OD + 12 + 12 + 12 + 06 + 06 + 06 + OD + OD + OD + 00 + 00 + 1E.

ADDRESS 01B0 GO GO.

"YOUR'RE DEAD" FUNERAL DIRGE



RESET 00 + 09 + 3E + 00 + 32 + 00 + 08 + 3E + 09 + 32 + 01 + 08 + CD + BO + 01 + 3E + OA + 32 + 01 + 08 + CD + 70 + 02 + C3 + 02 + 08.

ADDRESS 0900 + 03 + 00 + 03 + 00 + 03 + 03 + 00 + 06 + 00 + 05 + 05 + 00 + 03 + 00 + 02 + 03 + 00 + 1F.

ADDRESS OAOO + 16 + OE + 14 + 11 + 05 + 00 + 04 + 05 + 01 + 04 + 1A + 1A + 00 + 00 + 00 + 00 + 00 + 1F.
RESET + + GO GO.

STACK DEMONSTRATION PROGRAM:

Some constructors have been very inquisitive. They found locations at the high end of RAM which they could not remove! (This is because the TEC-1 was replacing them again). This was quite puzzling as we know anything in RAM can be removed and written over.

But this area is special and is called the STACK area.

When a PUSH instruction is executed by the Z80, the contents of the location are loaded into this area. The stack starts at OFFO for MON-1A EPROMS and OFDO for MON-1B EPROMS. It advances downwards, towards the LOW addresses in the 6116.

Each time a POP (or PULL) instruction is executed, the last item to put onto the stack is removed and the stack gets smaller. Otherwise it gets bigger and bigger.

If a program contains too many PUSH instructions, the stack will grow and eventually hit the program. This will make the computer CRASH!

The simple program below pushes register pair (they must be in pairs) **AF** into the stack and completely fills the RAM.

Try the program and watch the computer CRASH.

11 AA BB
D5
C3 03 08

RESET, GO.

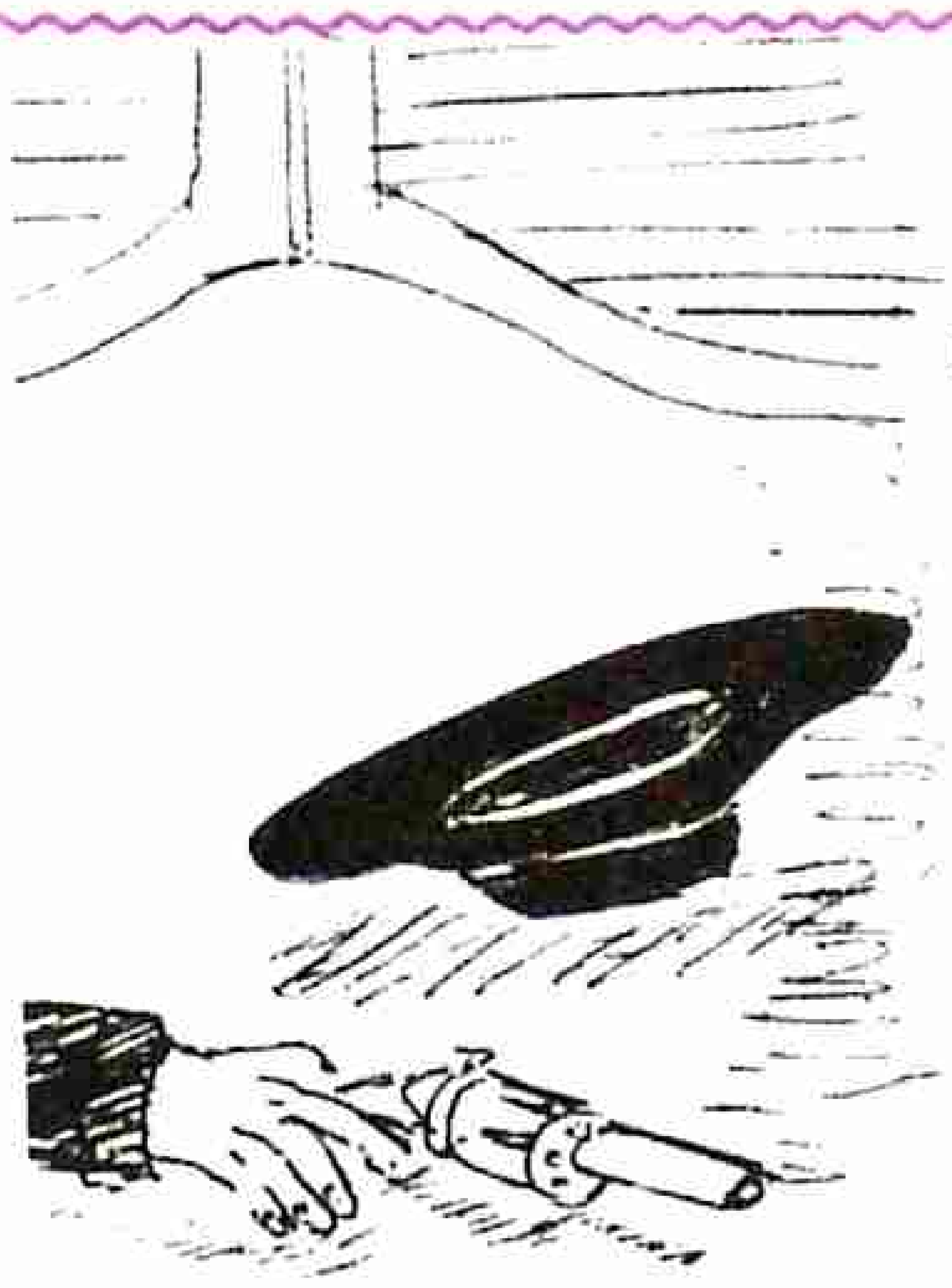
Increment the address and prove the 6116 is completely filled with AA, BB, AA, BB etc.

Change AA, BB to CC, DD and repeat. Check the RAM and read its contents.

QUICK DRAW

THE PROGRAM

```
        LD A,00        800  3E 00
        OUT(1),A       802  D3 01
START   LD DE,00       804  11 00 00
DELAY   DEC DE         807  1B
        LD A,D         808  7A
        OR E           809  B3
        JP NZ Delay    80A  C2 07 08
        LD A,E3        80D  3E E3
        OUT (2),A      80F  D3 02
        LD A,08        811  3E 08
        OUT (1),A      813  D3 01
LOOP 1  HALT           815  76
        AND OF         816  E6 0F
        CP OC          818  FE 0C
        JP Z,Right     81A  CA 24 08
        OR A           81D  B7
        JP Z,Left      81E  CA 29 08
        JP Loop 1      821  C3 15 08
RIGHT   LD A,01        824  3E 01
        JP Finish      826  C3 2B 08
        LD A,20        829  3E 20
        OUT (1),A      82B  D3 01
LEFT    LD A,20        82D  3E 28
FINISH  OUT (2),A      82F  D3 02
        HALT           831  76
        JP Start       832  C3 00 08
```



QUICK DRAW is a reaction game for two players.

To start the game, press RESET, GO.

After a DELAY, as determined by the delay routine at 804, the letter G will appear on the screen. The first player to press his button will be detected by the computer and result in the figure 1 appearing on the appropriate end of the display.

Player 1 uses the + button and player 2 uses the 'C' button.

Any button can be pressed to reset the game.

The first instruction is to load the accumulator with zero and output this to port 1 to prevent odd segments lighting up when the game is reset.

At address 804, the register pair DE is loaded with the value 00, 00. Surprisingly, this creates the longest delay as the first operation in the delay routine is to decrease the lower byte (register E) by one. This immediately removes the value of zero from the pair and when D is loaded into the accumulator, and the logical OR operation performed, the answer will only be zero when both the accumulator and register E are completely zero.

If one or both are not zero, the program will jump to instruction 807 whereupon register pair DE will be decremented by ONE. This loop will be cycled 256 x 256 times and each time will occupy quite a number of machine cycles.

This results in the letter G (for GO) taking a few seconds to appear on the screen. This creates the same effect as the delay circuit in the Quick Draw project described in issue 5.

When the register pair becomes zero, the program is advanced one address and the accumulator is loaded with the value E3. The value E3 will produce the letter G on the screen and the location of this letter is determined by loading the accumulator with 8 and outputting it to port 1.

The computer is now HALTED and waits for an input instruction. If any of the keys are pressed, the 74c923 will activate the NON-MARSKABLE INTERRUPT line and present data to the Z80 according to the value of the key.

If key C is pressed, the value 1010 is placed in the accumulator. This is then logically ANDed with the value F (1111) and the result appears in the accumulator.

When a number O-F is ANDed with 1111, the answer will be exactly the same as the number itself. In actual fact, this AND 0F operation is not required for the jump right command and you can ignore it.

After the AND OF operation, the number we are looking for is 1010 (for a jump RIGHT). The answer is compared with OC and the Z80 does this by effectively performing a subtraction operation in which the value C is subtracted from the contents of the accumulator.

If the answer is zero, the computer is instructed to jump to address 824. If the answer is not zero, a logical OR operation is performed with the accumulator as the operator and also the operand.

Since the accumuator is zero, the answer will be zero. Thus the program will advance via a jump instruction, to 829. If neither of the conditions are met, the CPU will jump to 815 and wait for a key to be pressed.

If the processor advances to location 824, the accumulator will be loaded with the value 1 and told to jump to address 82B. This value 1 is outputed to Port 1 and sets one of the latches ready to display the far right-hand digit.

At address 82D, the accumulator is loaded with 28 so that the segments 'a' and 'b' will illuminate when the value 28 (not twenty eight but two, eight) is outputed to Port 2.

Finally the processor is told to HALT at address 831.

On the other hand, if the program advances to 81E, the processor is told to jump to address 829 where the accumulator is loaded with 20 so that the far left-hand display will be activated when port 2 is given the value 20.

The program can be re-started by pressing ANY key, and the accumulator is loaded with zero at 800 so that odd characters do not appear on the screen.

Here are 6 simple experiments which can be performed on this program to better understand how it operates:

1. Load address 801, 2, 3 and 4 with 00 and play the game a few times. Notice how odd figures appear on the screen. Replace the correct program values and continue:
2. At address 812, load the value 10, or 04 or OF and note the different effects.
3. At address 816 and 817, insert 00 00. What effect does this have?
4. At address 819, insert OD or OE or OF. What is the result?
5. At address 829, load the value 10 and see the result.
6. Finally, insert the value 01 at location 806. Try the value 06, 0A BB or FF. What effect do they have?

difficult to obtain. The other chip on short-supply is the 74c923 as it is only made by one manufacturer.

The FND 500 or FND 560 displays are no longer in production by Fairchild (as they have ceased to produce OPTO devices) however other suppliers have produced identical replacements.

Apart from this, the TEC-1 is straightforward.

Out of the first 300 kits we had reports from only 6 readers who had trouble in getting the computer to work.

Paul had incorrectly placed the "six" button in the keyboard so that the flat was 90° out of position. The TEC-1 came on at address 0800 but the keyboard did not operate.

The 74c923 detected key 6 as being pressed due to the wiring of the contacts and the NMI being activated.

We traced the fault by removing the 74c923 and checking pins 8, 9, 11, 12 and 1, 2, 3, 4, 5 with an ohmmeter. The short between pins 2 and 9 showed key 6 to be at fault and that was when we noticed it!

Another TEC-1 came in with a very faulty + button and a broken PC line under one of the 7-segment displays.

Two computers had shorts between tracks around the memory section where the tracks are very close to one another.

But possibly the worst effort came in the post last week. The 74c923 socket had been made up with a 14 pin and an 8 pin so that 2 pins projected too far. The chip has been inserted so that pin 2 connected with pin 1 on the circuit. The other fault was a jumper missing near the first 8212. Both of these faults showed lack of inspection. When they were repaired, the computer worked perfectly.

Finally a constructor arrived with a TEC-1 under his arm. It gave an occasional display of odd segments and a beep from the speaker when switched on. The trouble was traced to a faulty Z80!

Apart from the above cases, the TEC-1 seems to offer a high degree of success.

If you have any problems with your unit, let us know. We want to present them in the next issue, under FAULT FINDING.

The introduction of the TEC-1 is primarily intended to unlock three areas of microprocessor design. These are:

1. To teach Machine Code programming.
2. To teach the art of creating Video Games and,

3. To access the real world.

MACHINE CODE programming is the skill of telling a microprocessor what to do by writing directly into a memory bank. The memory can either be a tempory storage (RAM) or permanent storage ROM. The main difference between these two is ROM will retain its memory when the power is turned off whereas RAM will lose its contents.

We can use this feature to write into RAM and then erase the program by turning off the power. Alternatively we can simply write over the old program. Both ROM and RAM are used in the TEC-1.

The Z80 has the capability of accepting over 700 instructions, some of which are 2-byte. These take the form of a two digit number which is written in hexadecimal form. The first large table in this installment shows how to write any of these numbers and explains how we arrive at C2 or E5 or D7 as a value in this instruction set.

A typical instruction is 3E. This means "load the accumulator (register A) with the following value . . ." Once you remember some of these instructions you will see why we have concentrated on Machine Code Programming.

Instructions such as 76 for HALT, C9 for RETURN and C2 for JUMP NOT ZERO are quite easy to remember. The meaning of JUMP NOT ZERO needs a little explanation. After C2 you must insert 2 bytes because the computer will interpret whatever is placed as the next two bytes as an address location. For example, 20 08 will tell the processor to jump to 0820. The instruction C2 also infers that if the program IS ZERO, the processor will proceed to the next instruction.

Machine Code is the only approach for video game development. It produces the fast-moving games as seen in the latest coin-in-the-slot machines. Any of the games on cassette are theoretically possible with the TEC-1. Mind you, we will not be advancing to the complexity of colour or the swirling action of V E N U S, but in a developmental way you will have the opportunity to program sections of a video game and watch the result.

PROGRAMMING STARTS HERE:

By now you will have completed the first 7 experiments and possibly tried some additional programming of your own.

But just in case you did not absorb all the facts, we will go through some of the experiments again to make sure everything is understood.

Just before we start, key the following:
Reset, two, +, eight, +, two, +, five, +, fourteen, + four, +, sixteen, +, 1A, +, zero, +, zero, +, zero, + zero, +, zero, +, 1E, ADdress 2, 7, 0, GO, GO.

DO YOU AGREE?

You should remember a few simple programs like this, to impress your friends.

The Z80 instruction-set has 8 special restart instructions which are single-byte instructions. By keying in one of the following: C7, CF, D7, DF, E7, EF, F7, or FF the computer will go to address location 00 or 08, 10, 18, 20, 28, 30, 38 and will receive information to go to the beginning of the 2 tunes and 3 games.

The 2 tunes are accessed by EF and F7. Push Reset, EF, GO. EF tells the Z80 to go to location 28 in the EPROM. At this address is an instruction 21 which tells the Z80 to load register pair HL with the contents of another location which is the beginning of the song table.

F7 directs the Z80 to go to location 30 and this address directs the Z80 to another song table.

We do not use C7, E7 or FF but the other 3: CF, D7 and DF select the commencement of the three games.

Try them.

When the computer is turned on, or the reset button pressed, the first available address is 0800.

This means the address locations from zero to 0800 have been allocated to the 2716 EPROM and most of this has been filled with start-up instructions and games. This includes a music table and letter table which are user accessible and a number table which is only computer accessible.

Actually the MON-1 EPROM has been filled from zero to 05A7, and we will now look at how many locations this represents.

LEARNING HEX

We know the TEC-1 is programmed in hexadecimal. This means the locations commence at 0000 and increase: 0001, 0002, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C . . . . . .9D, 9E, 9F, A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, AA, AB, AC, AD, AE, AF, B0, B1, B2, B3, etc up to 00F2, 00F3,00F4,00F5,00F6, 00F7, 00F8, F9, FA, FB, FC, FD, 00FE, 00FF.

If you fill in all the blanks between 0000 and 00FF, you will find there are 16 lots of 16 addresses. This is equal to 256 locations.

So, between 0000 and 00FF there are 256 address locations.

The next location 0100 (it is best to say oh, one oh, oh as the location is not really one hundred). Between 0100 and 01FF there are another 256 locations.
Between 0200 and 02FF there are another 256 locations.
0300 - 03FF = 256 locations
0400 - 04FF = 256 locations
0500 - 05A7 there are ??? locations.

Let's work it out. The number of locations in A7 is: A x 16 + 7
= 10 x 16 + 7 = 167.

The total number of address locations which have been programmed into the EPROM is:
5 x 256 + 167
= 1447.

We emphasise this aspect of hex numbering as it is often glossed over. Values such as 5A7 do not give us any indication of the value they represent.

To program 1447 address locations would take the best part of an afternoon as it is important to check and double-check the data at each address before running a programme. It only takes one fault in the program to upset its running with the result that hours of work will be ruined.

So, 5A7 is quite a large number and it nearly fills the EPROM. In fact there are only about 600 locations left.

The main reason for locating the beginning of the user-available section at 0800 will become obvious in a moment.

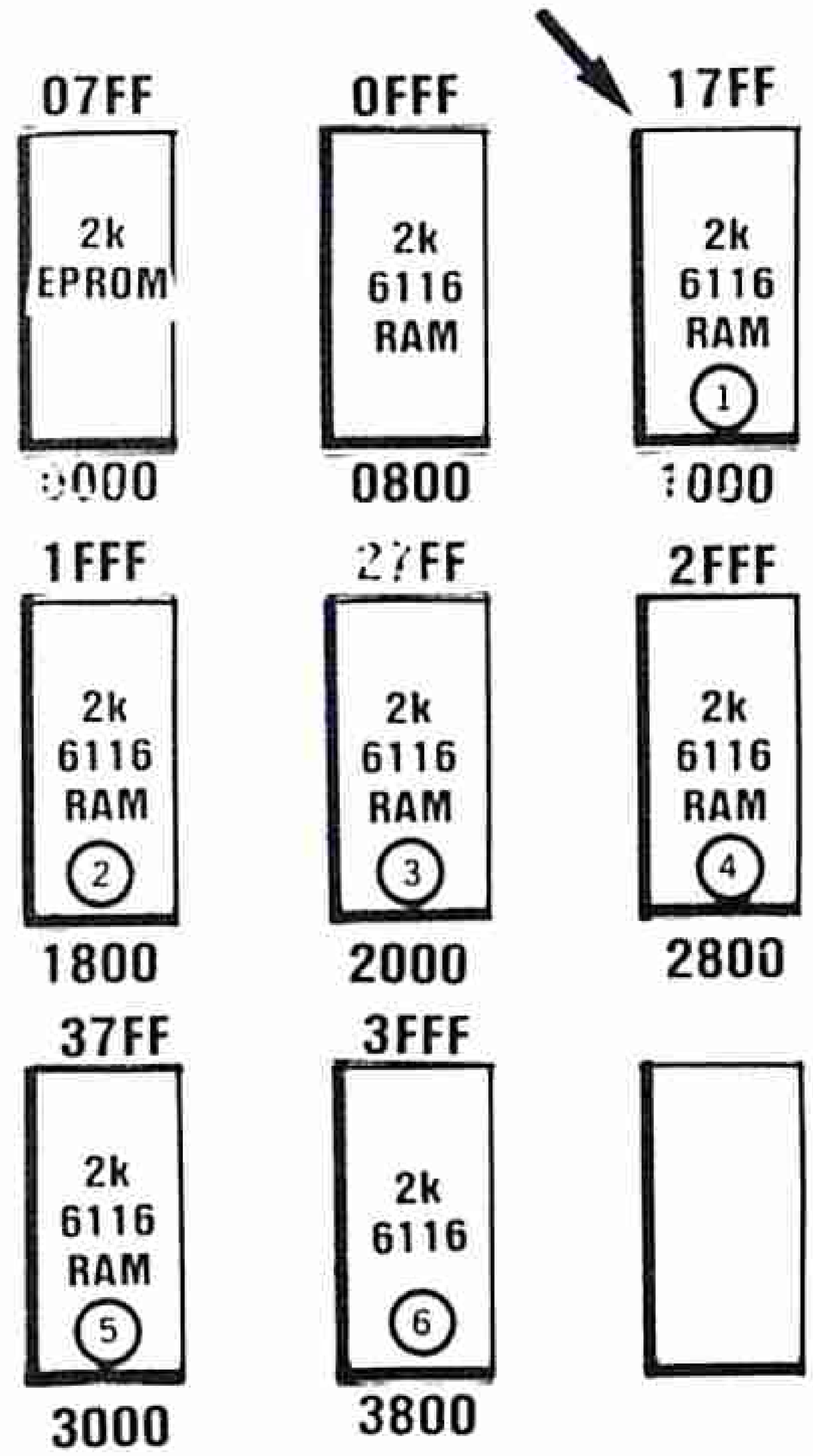
The second and most important reason for having the first user-available location at 0800 is the value it represents.

The EPROM is a 2k byte IC and this means it is capable of storing 2048 addresses. Each address will accept a number as high as 11111111 in binary, which is 255. We casually say it is a 2k EPROM but it is actually a 2k48 EPROM.

The value 2048 is equal to 0000 to ________. Let's work it out.

Divide 2048 by 256 and you will obtain the number of "groups" of locations. This comes to 8. So, 2048 is equal to 0000 to 0800 Or more accurately 0000 to 007FF. Every 0800 in hex represents 2048 locations. The next 2k starts at 0800 to 1000 or more accurately 0800 to 0FFF.

This is the EXPANSION PORT



The Memory Expansion Board, to be presented in a future article will contain 6 RAM chips. These are labelled 1-6 on the diagram and accessed as shown.

The hexadecimal start and finish for each "2k" of RAM on the TEC-1 is shown in the diagram above. The Z80 will access 64k of memory (65,536 bytes) or 32 chips such as 6116 or the N-MOS version 58725 by Mitsubishi. On the TEC-1, the two top address lines have not been decoded and thus the TEC will only address 16k of memory.

Between 05A7 and 07FF, the EPROM has been left blank for additional routines which will appear as MON 1A and MON 1B etc.

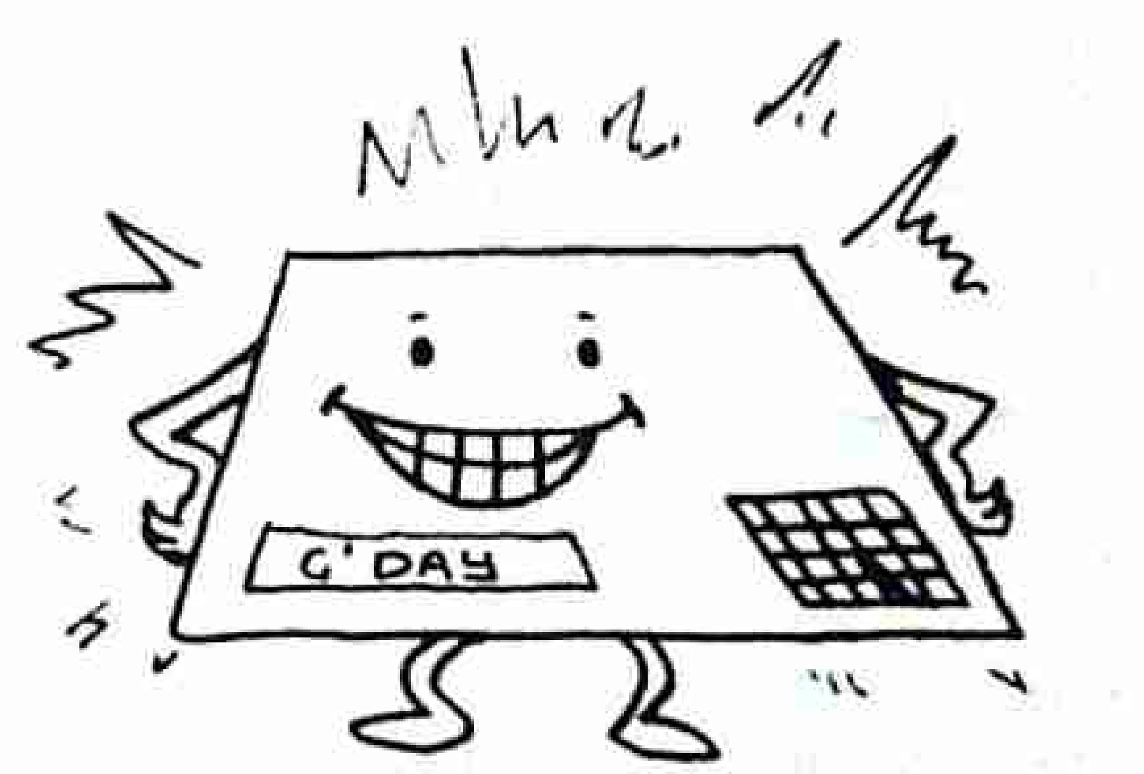
It is always wise to leave empty pockets between one program and the next in case a program needs to be extended. These empty locations can be filled at a later date with the aid of an EPROM BURNER.

Other locations in the EPROM have also been left blank. The most important of these is the "first hundred bytes". Within this section are 8 one-byte subroutine call locations such as 00, 08, 10, 18 where we can place a brief program (up to 7-8 bytes long). We can write a jump or call instruction and maybe a return instruction so that the main program needs ONLY a single byte instruction. (Normally a CALL requires 3 bytes).

You have already keyed a number of instructions and you will be starting to remember what they stand for. **3E,** for instance, means *"Load accumulator A with the following byte"*. The accumulator is one of the registers in the Z80 and there is nothing magic or complex about it.

It is merely a set of 8 flip flops which can be set HIGH or LOW to reflect the number which has been entered.

The reason why register A is so often used in programmes lies in its special feature. It is the accumulator register and this means all logic operations (such as AND, OR ADDITION and SUBTRACTION) will be performed using it.



THE ALL PURPOSE NIFTY TEC-1...

Experiment 7 combines a number of interesting features such as CALL and JUMP and we will explain what these do.

The program looks fairly simple, but this very deceptive. If you were required to program all the information to produce a running letter program, it would be like being asked to buy a bottle of lemonade but firstly manufacture and print the dollar note required to buy the bottle of fizzy. Obviously this would be an enormous task so we use AIDS to make the task easier.

In our case we use John Hardy's letter writing routine and his running or shift routine and put them together to get our sentences.

We will go through the program in the same way as carried out by the Z80.

By pressing the reset button, +, +, the computer will start at location 802. The small amount of operating brains inside the Z80 will instruct it to look at location 802 in the 6116 RAM. It will find the instruction 3E and this will tell it to load the accumulator with 00. This is one way of removing the initial 'rubbish' in a register or accumulator A.

The next instruction 32 means load the contents of accumulator A (which is 00) into the address which follows, which is 0800. This means we have removed any rubbish data and loaded it with 00.

At address location 807, 'A' will be loaded with 09. The previous value, 00 will be written over but it has already done its job of being loaded into location 0800.

At location 809, the Z80 is told to load the value 09 into address 801.

All we have done to date is simply load the first available address with 00 and the second with 09. We could have done this manually but the routine calls for the contents of 800 and 801 to be altered between two different values: 0900 and 0A00 and so the computer has to do the job.

The next instruction is a CALL instruction. The Z80 is asked to look at the address 1B0 in the EPROM. What it will find there is quite considerable.

Firstly it loads register pair DE with the contents of 800 and 801. At these locations it finds the address 0900. Next it takes the first byte at 0900 and plays the note corresponding to this value. This requires another table comprising byte-pairs. One of which is the pitch value and the other time-value. It continues using each byte at 0900 until either a **IF** or **IE** is recognised. With IF the computer goes to line **80F** in the main program and carries out a similar procedure for the Letter Table.

After the letters have scrolled across the screen, the computer returns to the main program (location 817) and this instructs the Z 80 to jump to line 802.

WHAT YOU HAVE LEARNT FROM EXPERIMENTS 1-7:

1. The first available address is 0800.
2.The dots on the display indicate when the address OR the data can be changed.
3.Incrementing the display means to increase the address value.
4.Decrementing the display means to decrease the address location.
5. The TEC-1 is programmed in MACHINE CODE.
6. The key pad is a HEX PAD as it contains the numbers 0-9 and letters A-F.
7. The values 00 to FF represent the numbers ZERO to 255:

HERE IS THE COMPLETE TABLE:

OUR COMPLETE HEX TABLE

0	00								
1	01	51	33	101	65	151	97	201	C9
2	02	52	34	102	66	152	98	202	CA
3	03	53	35	103	67	153	99	203	CB
4	04	54	36	104	68	154	9A	204	CC
5	05	55	37	105	69	155	9B	205	CD
6	06	56	38	106	6A	156	9C	206	CE
7	07	57	39	107	6B	157	9D	207	CF
8	08	58	3A	108	6C	158	9E	208	D0
9	09	59	3B	109	6D	159	9F	209	D1
10	0A	60	3C	110	6E	160	A0	210	D2
11	0B	61	3D	111	6F	161	A1	211	D3
12	0C	62	3E	112	70	162	A2	212	D4
13	0D	63	3F	113	71	163	A3	213	D5
14	0E	64	40	114	72	164	A4	214	D6
15	0F	65	41	115	73	165	A5	215	D7
16	10	66	42	116	74	166	A6	216	D8
17	11	67	43	117	75	167	A7	217	D9
18	12	68	44	118	76	168	A8	218	DA
19	13	69	45	119	77	169	A9	219	DB
20	14	70	46	120	78	170	AA	220	DC
21	15	71	47	121	79	171	AB	221	DD
22	16	72	48	122	7A	172	AC	222	DE
23	17	73	49	123	7B	173	AD	223	DF
24	18	74	4A	124	7C	174	AE	224	E0
25	19	75	4B	125	7D	175	AF	225	E1
26	1A	76	4C	126	7E	176	B0	226	E2
27	1B	77	4D	127	7F	177	B1	227	E3
28	1C	78	4E	128	80	178	B2	228	E4
29	1D	79	4F	129	81	179	B3	229	E5
30	1E	80	50	130	82	180	B4	230	E6
31	1F	81	51	131	83	181	B5	231	E7
32	20	82	52	132	84	182	B6	232	E8
33	21	83	53	133	85	183	B7	233	E9
34	22	84	54	134	86	184	B8	234	EA
35	23	85	55	135	87	185	B9	235	EB
36	24	86	56	136	88	186	BA	236	EC
37	25	87	57	137	89	187	BB	237	ED
38	26	88	58	138	8A	188	BC	238	EE
39	27	89	59	139	8B	189	BD	239	EF
40	28	90	5A	140	8C	190	BE	240	F0
41	29	91	5B	141	8D	191	BF	241	F1
42	2A	92	5C	142	8E	192	C0	242	F2
43	2B	93	5D	143	8F	193	C1	243	F3
44	2C	94	5E	144	90	194	C2	244	F4
45	2D	95	5F	145	91	195	C3	245	F5
46	2E	96	60	146	92	196	C4	246	F6
47	2F	97	61	147	93	197	C5	247	F7
48	30	98	62	148	94	198	C6	248	F8
49	31	99	63	149	95	199	C7	249	F9
50	32	100	64	150	96	200	C8	250	FA
								251	FB
								252	FC
								253	FD
								254	FE
								255	FF

MOVING ON. . . .

What is the next thing you would like to do?
How about turn on one segment of the display?

Key in this program:

```
RESET
3E 01
D3 01
3E 01
D3 02
76
RESET, GO.
```

This is what you have done:
Reset 3E + 1 + D3 + 1 + 3E + 1 + D3 + 2 + 76, Reset, GO.

The top LED in the first display lights up. We say the first display as it is the lowest priority digit.

In English, this is what you have done: Load register A with the value 1. Output this to port 1. Load register A with the value 1 and output it to port 2. Halt.

This is how the program is written:

```
LD A,01      800   3E 01
OUT (1),A    802   D3 01
LD A,01      804   3E 01
OUT (2),A    806   D3 02
HALT         808   76
```

We can read the value of each location by pressing RESET and then stepping through the program by pressing: + + + + + + + + + + + + + +

We can alter the position of the LED which is to be lit, by altering the value of locations 801 and 805.

The whole program does not have to be re-typed. Any location can be altered as follows:

Either press Reset and + or – – – – to get address 801.
Change 801 to 08
Press RESET, GO.

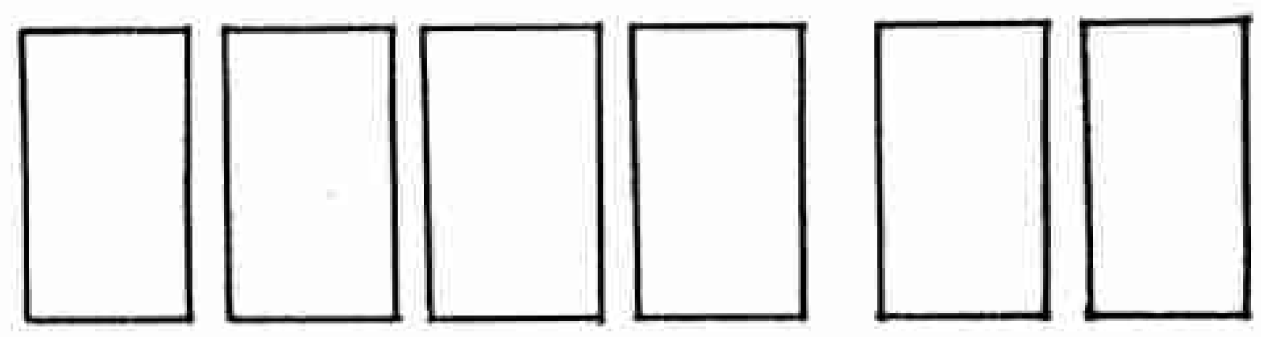
Note the LED has moved to the 4th display.

Try the values: 02, 10, 04, & 20.

You have accessed each display. Return to the first display by changing the value at 801 to: __ __
Increment to 805 and change the value 01 to: 02, 04, 08, 10, 20, 40 and finally 80.

Are you impressed? You have accessed each of the segments including the decimal point. You have become master of the display. With a little more instruction you can illuminate more than one LED in the display. But first let's see what you have learnt.

The displays are accessed as follows:



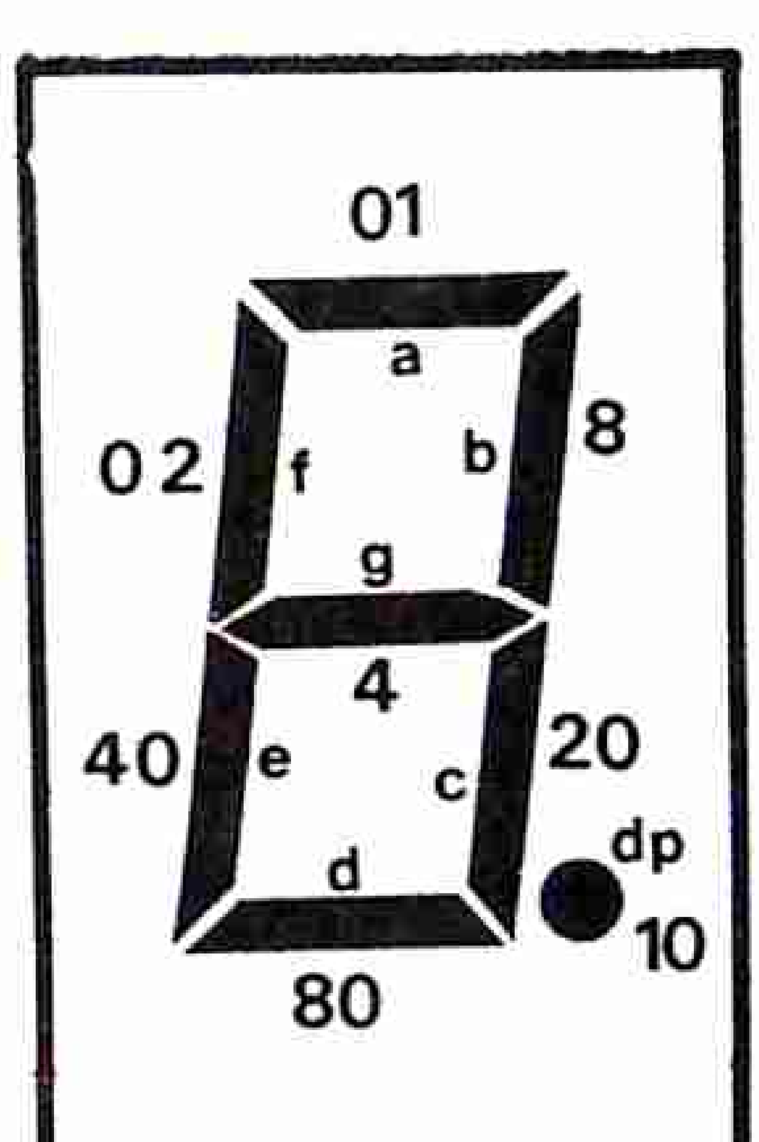
The value of each display is twice the previous and they increase from RIGHT to LEFT.

Port 1 is the cathode port. A 'HIGH' or 'ONE' in the appropriate bit 5 - 0 activates cathode 5 or 4 or 3 or 2 or 1 or 0.

This is how it works: The numbers 20, 10, 8, 4, 2, 1, are converted to binary and the computer sees them as:

```
0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1
```

The last line in the table is easy to read. It is 1. The second lowest line is 2, then the next line has the value 4, then 8. The first and second lines are a little more difficult to explain because they are not 16 and 32. That's the binary number but we are interested in the value we have to punch into the hex keyboard. The answer will be covered in a moment. For the present, we will look into the concept of converting binary numbers into HEX numbers.



Firstly we will give you the answers. Each segment on a display has the following values. The top LED (segment A) is lit with a value 1. The decimal point needs a value 10, the centre LED (segment G) needs the value 4 and so on.

What do you think the following program will produce?

```
3E 01
D3 01
3E 20
D3 02
76
```

Try it. Start at 0800. Enter the program, press reset, GO.

We can illuminate more than one segment at a time and more than one display at a time by changing two locations as shown in the following examples:

Try each of these examples and see a pattern of addition appearing.

At location 805, insert:

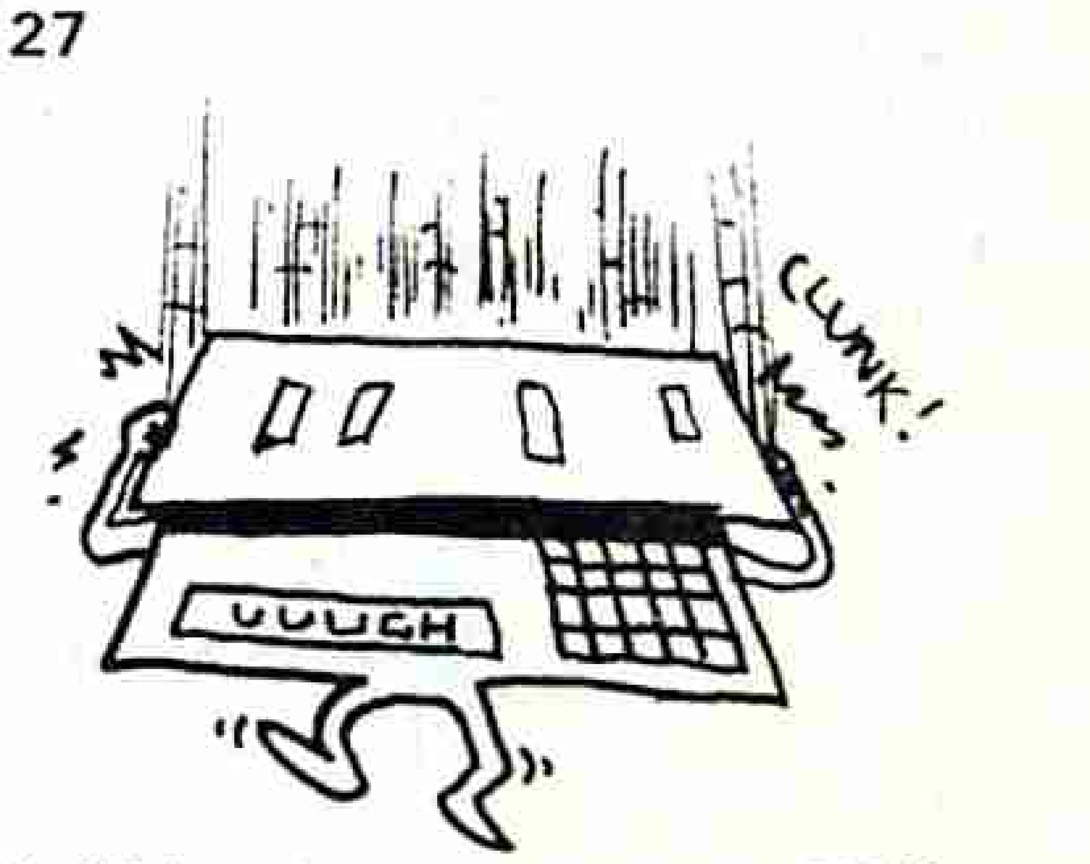
(a) 3
(b) 9
(c) 15
(d) 34
(e) 62
(f) A
(g) D
(h) F

The letters A, D and F will be quite a surprise. They also produce a reading on the display and it is obvious they have a value. Their values will be covered in the section CONVERTING BINARY TO HEX and HEX TO BINARY.

Back to the display.

More than one display can be illuminated at a time by inserting a different value at location 801. Keep say "62" at location 805 and insert the following at 801:

(a) 9
(b) 27

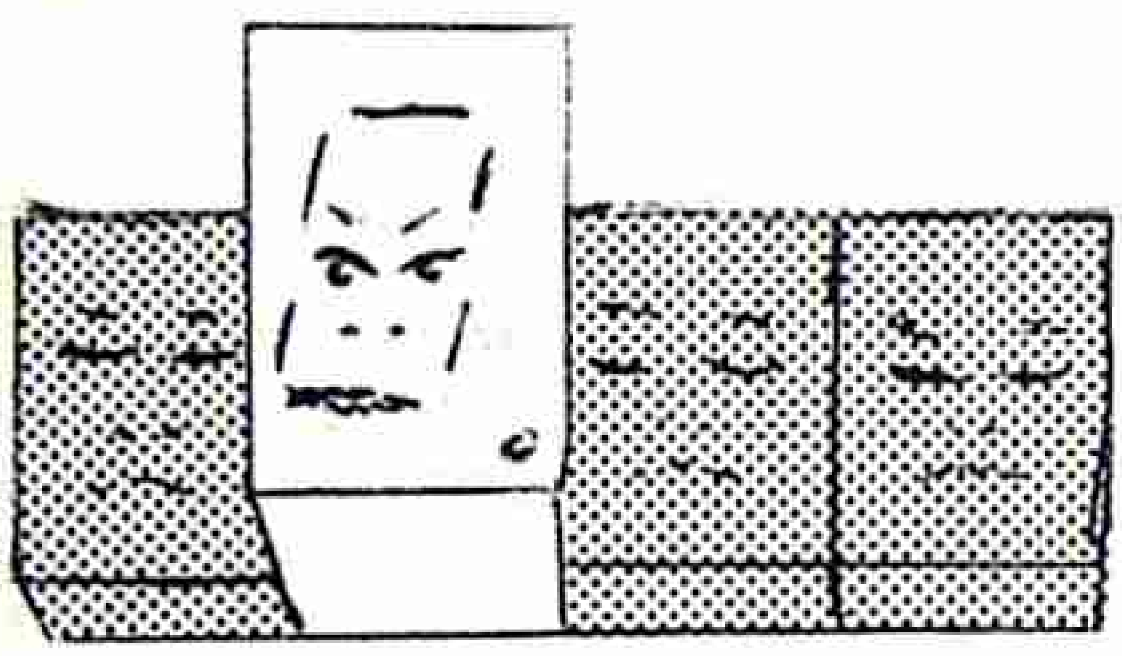


... WITH IN/OUT BOARD....

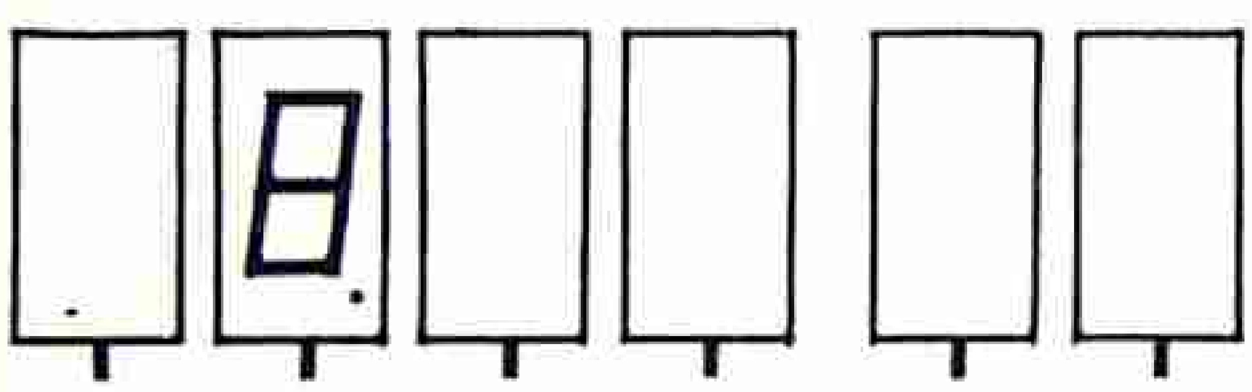
(c) 31
(d) 10
(e) A
(f) 3B
(g) 2F

This is as far as we can go with blind experimenting. We must cover some of the background theory on hex numbers to understand what we are doing.

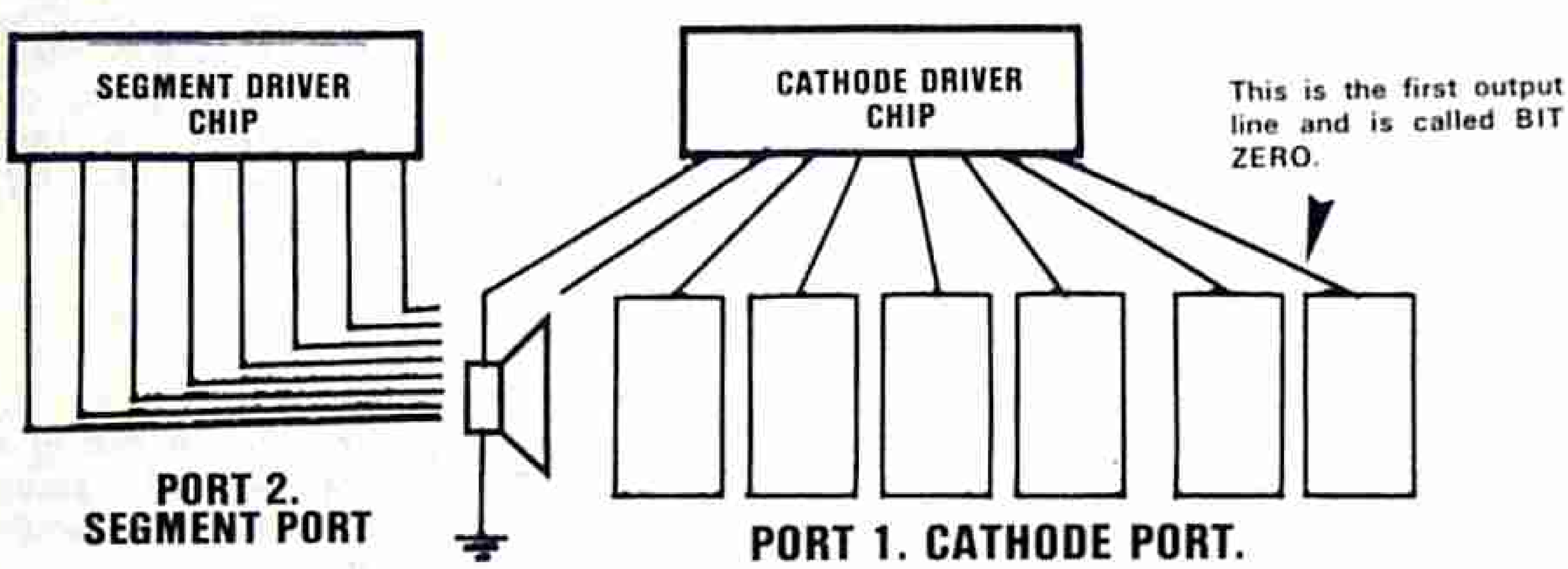
In experiment (d) above, we turned on the fifth display thus:



To do this we must switch the appropriate display transistor ON. (Don't worry about the segment drivers at this stage).



In the TEC-1 the displays are connected to the 8212's as follows:



The display we wish to illuminate is in the fifth output line and its binary number is: 00010000. The keyboard is in hex so to convert this to a hex number we have to break it into two groups of four:

0 0 0 1   0 0 0 0

This represents 1   This represents 0

The answer is 10.

This is not called ten. It is called one, zero or one, oh. Here are some more examples:

1. To illuminate the first, third, and fifth digits, this is the procedure:

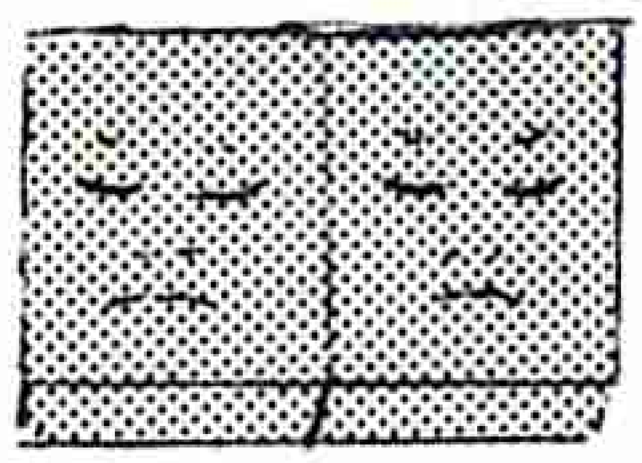
The high lines from the 8212 will be:

0 0 1 0 1 0 1 0

break this into 2 groups of four:

0 0 1 0   1 0 1 0

This is equal to 2   This is equal to A



WAKING UP THE FIFTH DISPLAY

The answer is 2A.

OK, you don't understand how we get A. Look at the table on P 71 of issue 10. The hex number for 1010 is A.

Try these: Write the hex number for:

(a) 1100 =
(b) 1001 =
(c) 1101 =
(d) 1111 =

2. To illuminate the first, second and third displays, the HIGHs must appear in the following places:

0 0 1 1 1 0 0 0

0 0 1 1   1 0 0 0

3   8

Answer: 38 (in hex)

3. To illuminate all the displays:

0 0 1 1 1 1 1 1

0 0 1 1   1 1 1 1

3   F

4. To bip the speaker:

1 0 0 0 0 0 0 0

8   0

5. To click the speaker and turn on displays: one, two, three and four:

1 0 1 1 1 1 0 0

1 0 1 1   1 1 0 0

B   C

Answer: BC

6. To access the vacant cathode:

0 1 0 0 0 0 0 0

0 1 0 0   0 0 0 0

4   0

Answer: 40.   You will notice NOTHING!

Example 3 above shows that the max hex value to illuminate ALL THE DISPLAYS is 3F. Remember this.

This is the program we are using:

This value determines which display(s) are lit and is called CATHODE ACCESSING

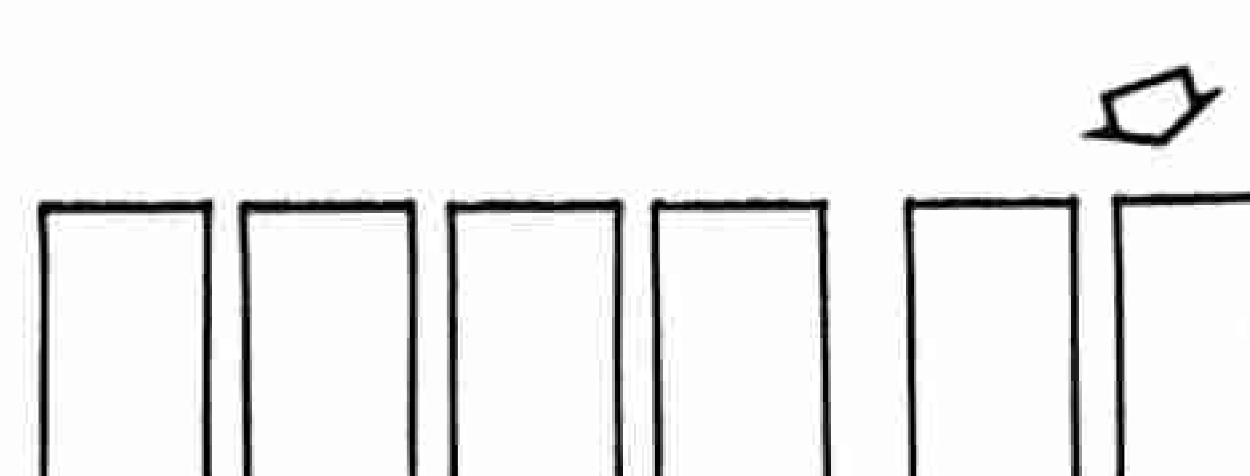
```
3E 01     Max 3F
D3 01
3E 01     Max FF
D3 02
76
```

This value determines the segments which are lit and is called SEGMENT ACCESSING.

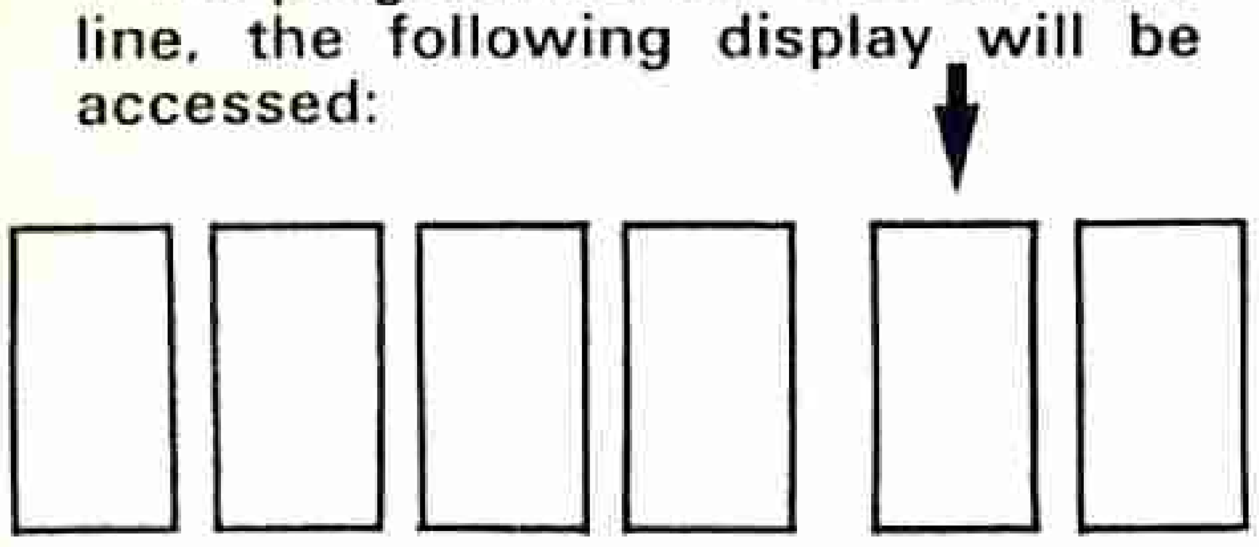
The program has two variables: lines 1 and 3. For line 1, the maximum value is **3F** (this is 64 different possibilities) and line 3 has a maximum of **FF** (this is a maximum of 256 possibilities). These are independent variables and either can be changed at any time to any value in the range as specified above. The result is thousands of different combinations.

Here are some of the possibilities and how they are obtained:

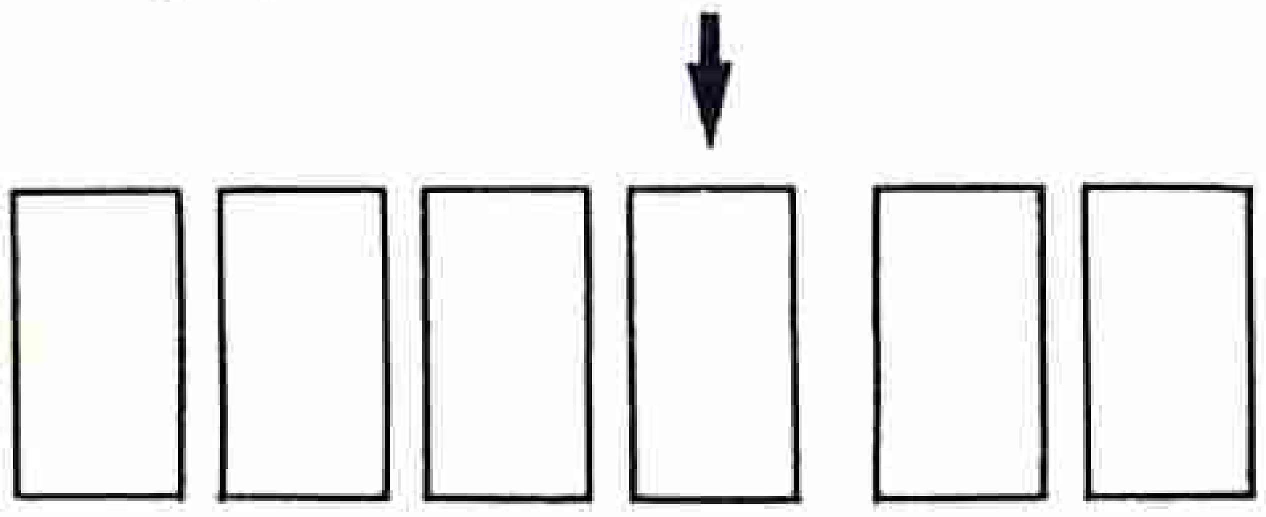
If line 1 has the value 1, we will be accessing this display: (line 3 can be any value from 00 to FF).



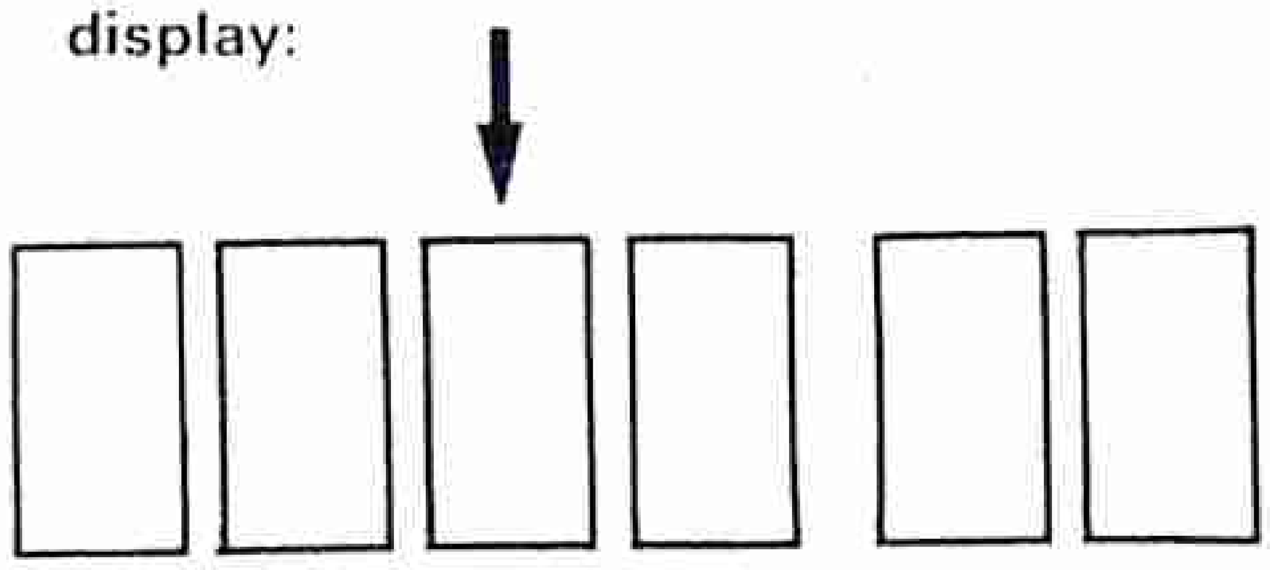
If we program **3E 02** into the first line, the following display will be accessed:



The value **3E 04** will access this display:



The value **3E 08** will access this display:

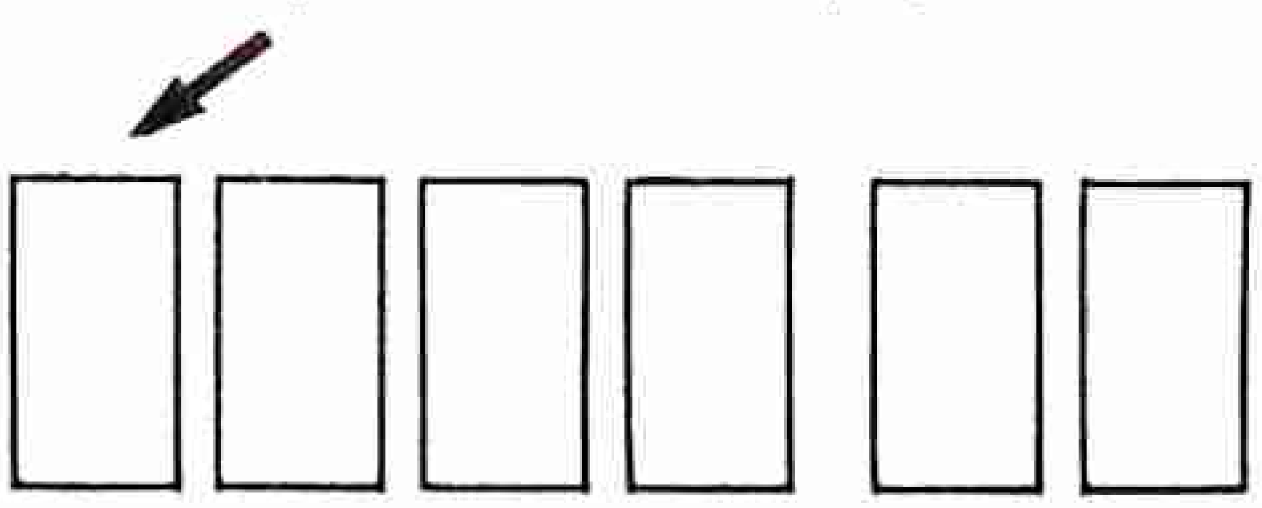


Can you see why?

The computer converts 08 to binary and it becomes 1000.
The "one" or "HIGH" corresponds to the display in the diagram above

3E 20

corresponds to this display:



Note: 20 is the Hex number and is obtained by separating 20 into 2 0.

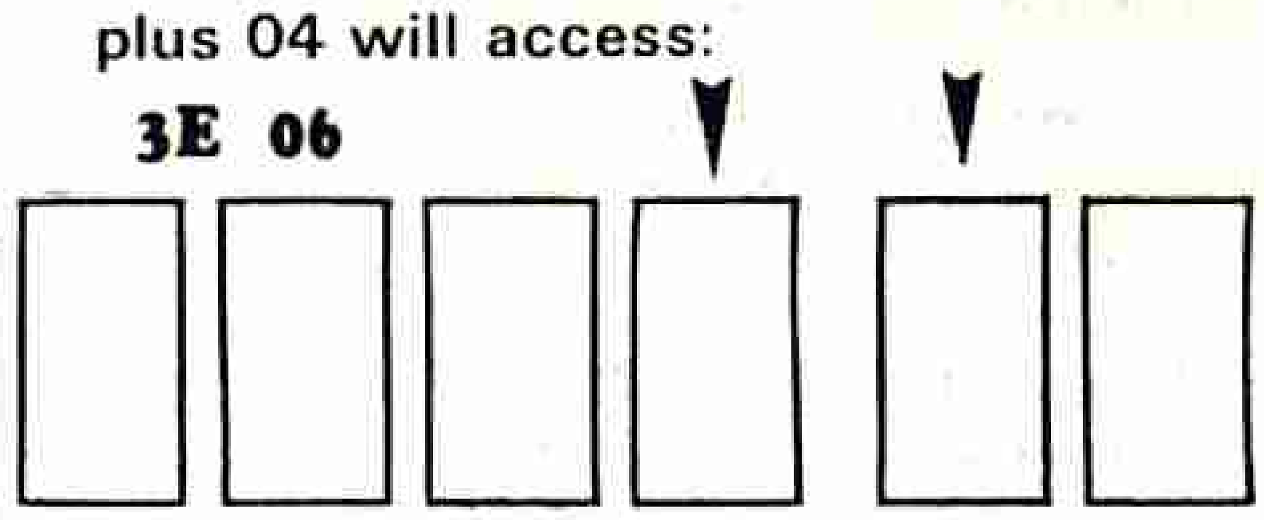
2	0
0010	0000

This gives 0010 0000 as the binary equivalent and shows the HIGH is bit 5 and this will illuminate the first digit on the display. (Note: The first output line is bit 0).

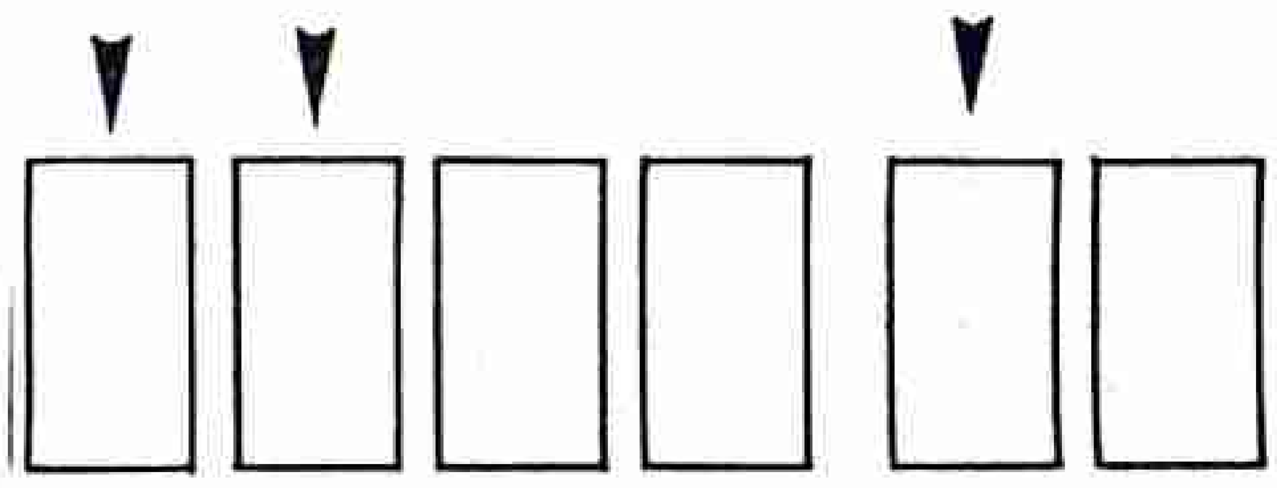
[The actual combination of segments which will be illuminated will depend on line 3 of the program and this will be explained in a moment.]

More than one display can be turned on at the same time by adding CATHODE ACCESS values. Thus 02 plus 04 will access:

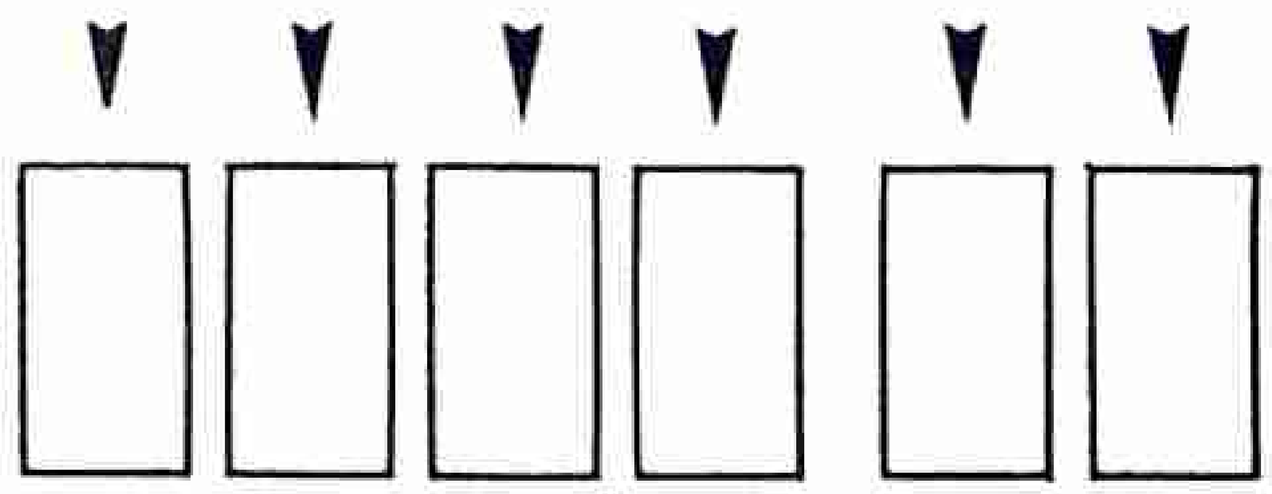
3E 06



3E 32 will access:



3E 3F will access:

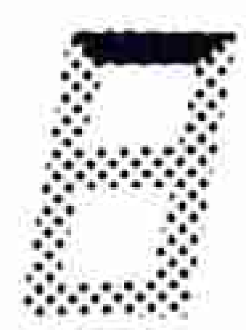


SEGMENT ACCESS

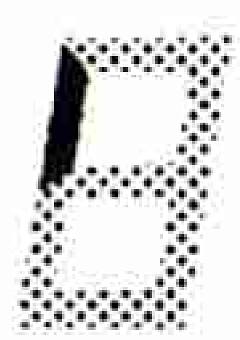
Line 3 of the program determines the letter, number or pattern which will appear in the display(s). We will take the simple case of the lowest priority display being accessed (line 1 will be 3E 01) and we will produce some interesting patterns, numbers and letters in (or on) the display.

Here are some of the results of changing the data in line 3:

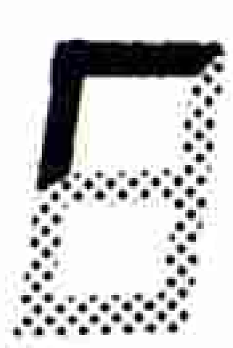
3E 01



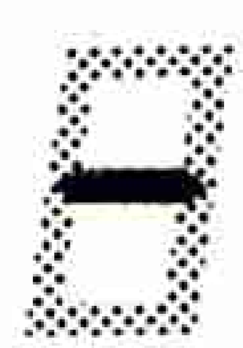
3E 02



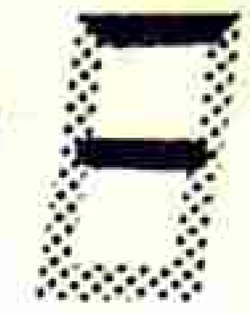
3E 03



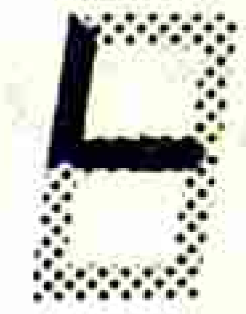
3E 04



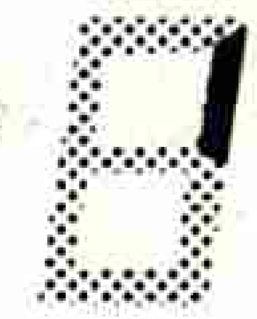
3E 05



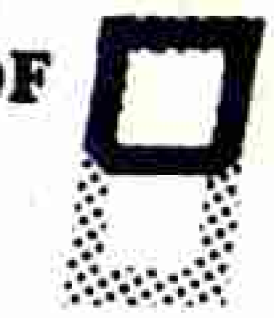
3E 06



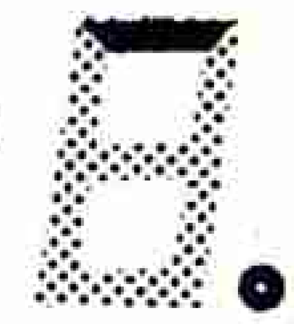
3E 08



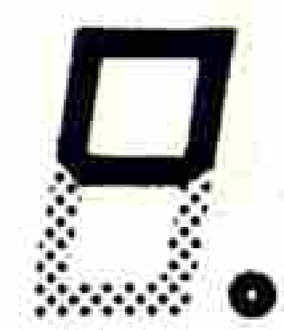
3E 0F



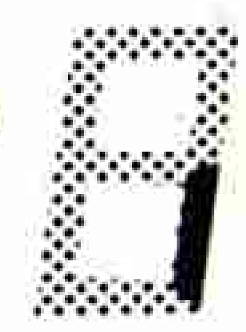
3E 11



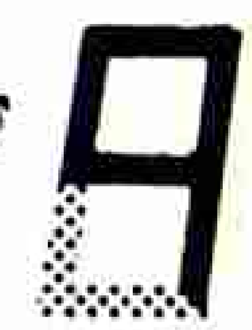
3E 1F



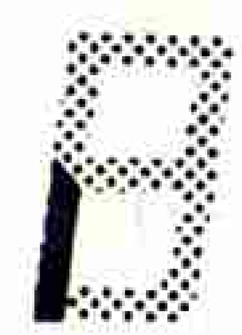
3E 20



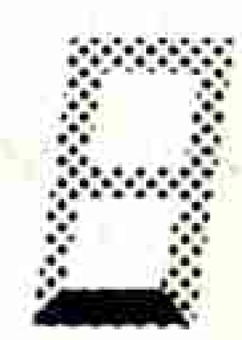
3E 2F



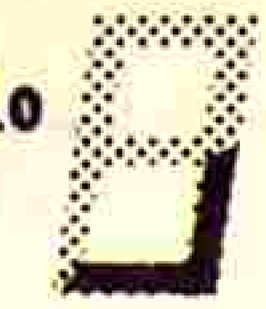
3E 40

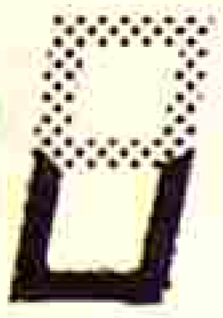


3E 80



.... 8X8 MATRIX DISPLAY BOARD...

3E A0 

3E E0 

3E FF 

Here is an equally interesting program to automatically increment the SEGMENT ACCESS value. The result will be to produce every combination possible on the display. Watch the results carefully and see if you can predict the next segment to appear.

At address 0800, type the following program:
(And the DELAY ROUTINE at 0A00)

```
LD A,01        800   3E 01
OUT (1),A      802   D3 01
LD B,00        804   06 00
LD A,B         806   78
OUT (2),A      807   D3 02
INC B          80A   04
CALL 0A00      80B   CD 00 0A
JP 0806        80E   C3 06 08
```

0A00

```
                 A00   11 FF FF
This is the      A03   1B
DELAY            A04   7B
ROUTINE          A05   B2
                 A06   C2 03 0A
                 A09   C9
```

By reducing the DELAY TIME, the display will cycle at a faster rate. Try the value 06, A0, C0, 10 and 02 for the value A02 and determine which value is the most suitable.

SUMMARY

This is the program you have been entering into the TEC-1 to illuminate the segments. The two variables are located at 801 and 805.

```
800  3E  04
802  D3  01
804  3E  10
806  D3  02
808  76
```

The data at address 801 can range from 01 to 3F and it determines the combination of digits which will be illuminated.

The data at address 805 can range from 01 to FF and it determines the combination of segments which are illuminated.

Expt 8: ILLUMINATING ONE SEGMENT

Aim: To illuminate one segment on the display .

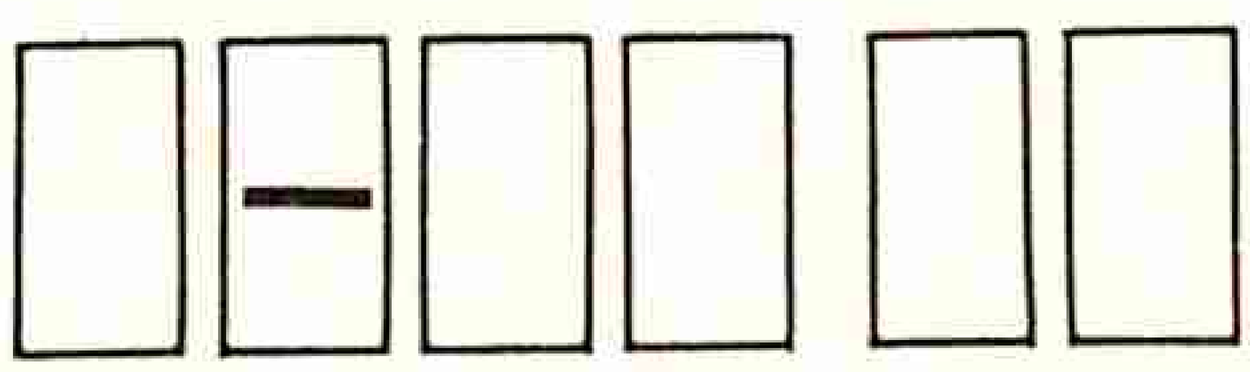
Theory: To master the display you must be able to access (locate) any segment.
The TEC-1 display has 6 digits and each has 8 light emitting diodes. This produces 6x8 = 48 locations.

The aim of this experiment is to illuminate one of these segments.

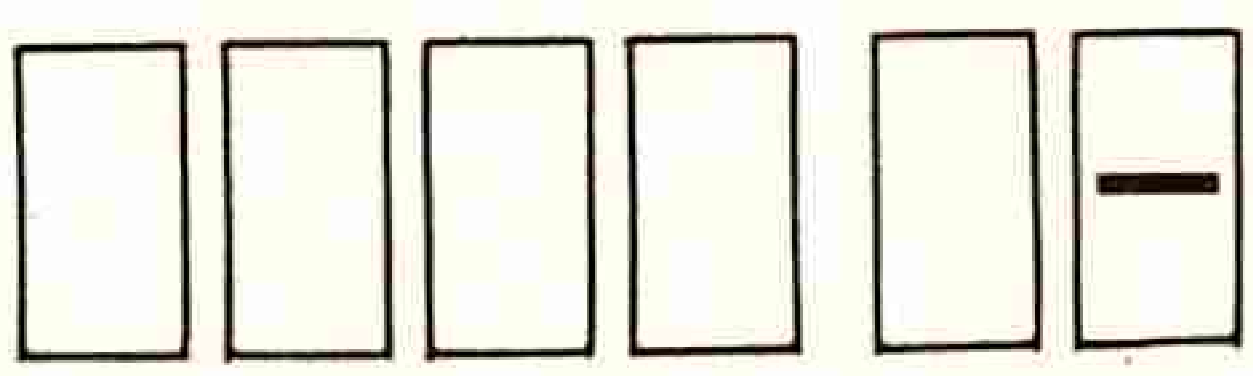
Key the following:

Reset, 3E + 10 + D3 + 01 + 3E + 04 + D3 + 02 + 76 Reset Go.

The display will show one centre segment in the 5th display thus:

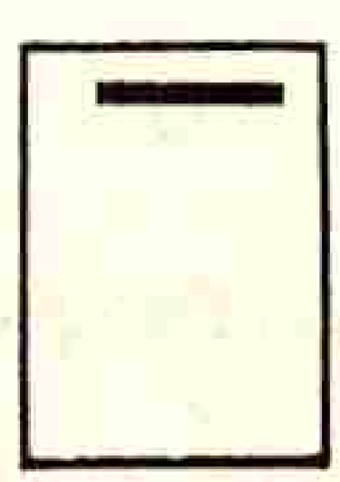


Now key in this program:
Reset 3E + 01 + D3 + 01 + 3E + 04 + D3 + 02 + 76 Reset, Go.
The result is:

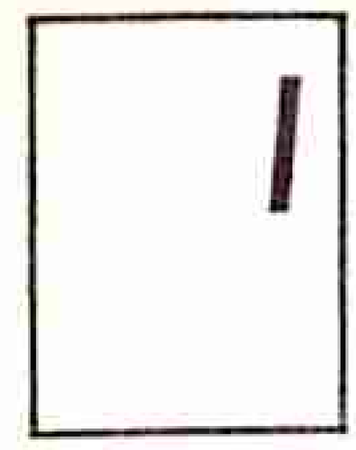


The only difference between the two programs is the byte at 801.
You are now going to make a discovery yourself:
This is how to do it:

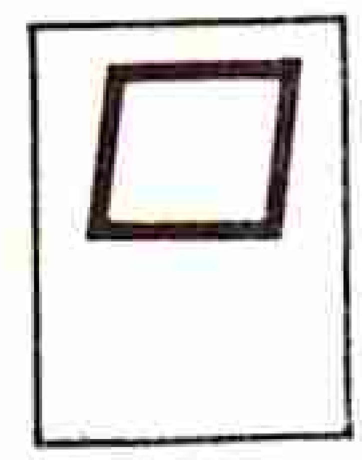
Press Reset.
Press + to 805. Change the data (now showing 04) to 01.
Press Reset, Go.
The segment 'a' illuminates thus:



Press RESET. Press + + to get address location 805.
Change data to 08.
Press RESET, GO.
Only segment b illuminates thus:

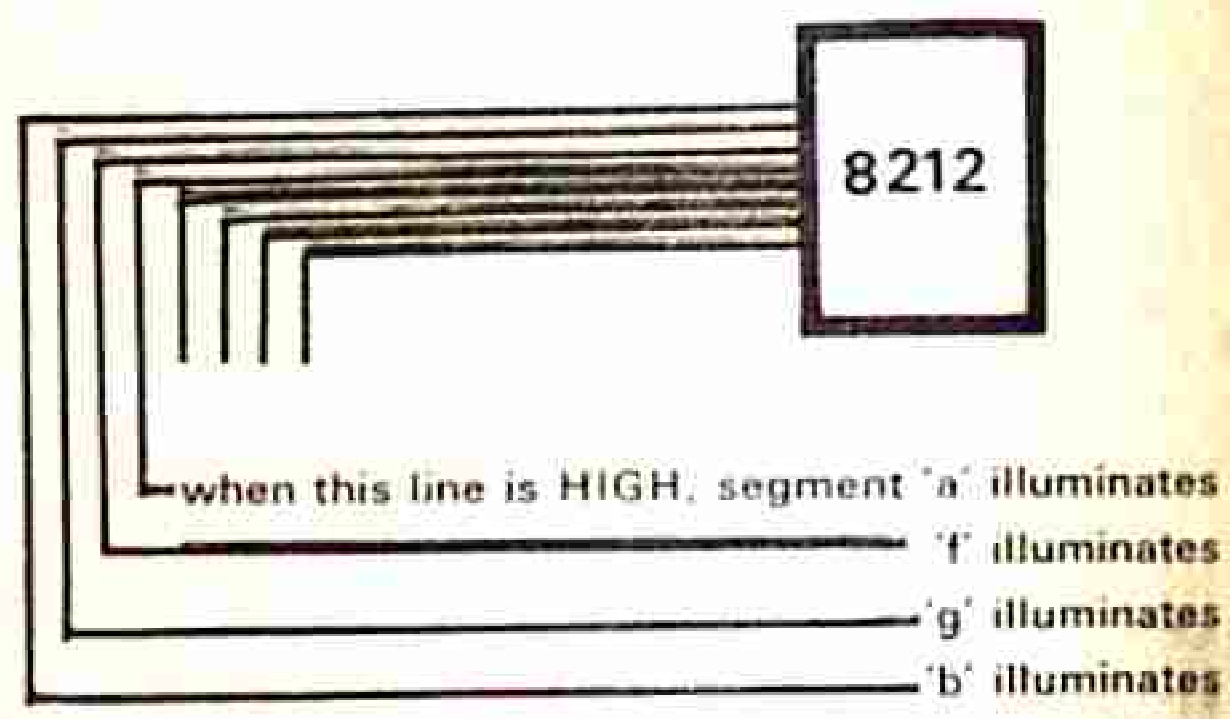


Press RESET. Press + + to 805.
Change data to OF.
Press RESET, GO.
Four segments illuminate thus:



Let's see how this comes about.

The seven-segment display is labelled a-g and the decimal point h. Each segment is illuminated via a binary number sent from the Z80 to the 8212 latches.



So far, so good.

You can see we have activated the 4 lowest value lines and this has produced a rectangle made up of segments a, b, f and g.

Now look at how we programmed a, b, f and g.
The numbers we used were 1, 2, 4, and 8.
Can you see the connection?
It's binary.

The computer converts our keyboard number to a binary number thus:

a = 1
b = 10
c = 100
d = 1000

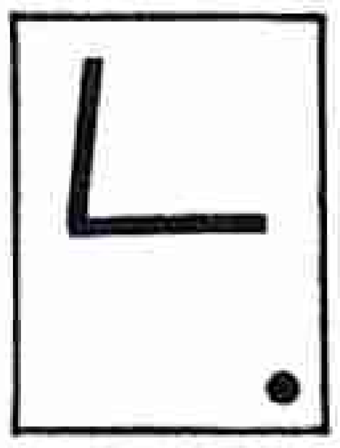
The next segment in the series is 10000, which is 16 in binary terms.

So, let's see what happens when we put 16 into the program at 801.

Key this sequence:

RESET 3E + 16 + D3 + 02 + 3E + 10 + D3 + 01 + 76 RESET GO.

Has something gone wrong? We get this result:

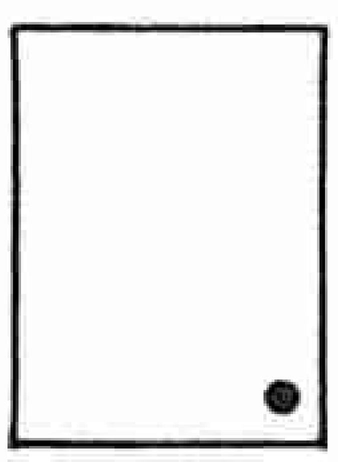


The one fact we have omitted is the TEC-1 is programmed in hexadecimal numbers. This means 16 should be typed as 10 (one, oh or one zero) since the computer counts: 1,2, 3,4,5,6,7,8,9,A,B,C,D,E,F,10. Thus 10 is equal to 10000 in binary.

Put 10 into the program thus:

RESET 3E + 10 + D3 + 02 + 3E + 10 + D3 + 01 + 76. RESET GO.

You will find the decimal point will illuminate thus:



Now add the value of segments a, b, f and g to the value of the decimal point:

This is done by adding 1 + 08 + 02 + 04 + 10 to get 1F.

Place this value in the program:

RESET *3E + 1F + D3 + 02 + 3E + 10 + D3 + 01 + 76* RESET GO.

Success.

We have just added five Hex numbers.

Let us illuminate the next segment in the series. It will have the binary value 100000.
This is equivalent to 32 in binary, but the binary value is not important. It is the hexadecimal value we are interested in. What is 100000 in Hex?
Break the value into 10 0000 so that the four last numbers form a group. The value 10 is 2 in binary and 0000 is zero. Thus the hex value is 2 0 = 20.

Place 20 in the program. Segment 'c' will illuminate.

To combine segment c with a, b, f, and g, we add their Hex values together: *20 + 01 + 08 + 02 + 04 = 2F.*

Insert 2F into the program.
Press RESET, GO.

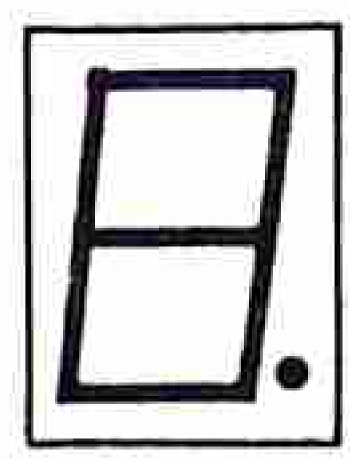
The result is a figure **NINE!**

You can now see that each number and letter in the display is produced by a set of HIGHs on the appropriate lines.

To access the next segment in the series, we must place a HIGH on the 7th line: 0 1 0 0 0 0 0 0. This is 40 in Hex terms and will result in segment 'e' iluminating. When the 8th line is HIGH, segment 'd' will be illuminated.

To illuminate ALL the segments on the display (including the decimal point) we must add the following values: *1 + 2 + 4 + 8 + 10 + 20 + 40 + 80. This is eqial to FF*
(1 + 2 + 4 + 8 + = F) (10 + 20 + 40 + 80 = F0)

Program FF into the sequence at location 801. The result is:



If you programmed our SEGMENT ACCESS routine on P 20 and watched the display carefully, you will have been amazed at the number of recognisable letters and numbers which can be created on a simple 7-segment display.

Our requirement at this stage is to be able to produce some, if not all, of the characters using the information we have learnt in experiment 8.

These are the facts we need:

1. Segment values:
   a = 01
   b = 08
   c = 20
   d = 80
   e = 40
   f = 02
   g = 04
   h = 10

2. The value of each segment is added in hex form when we need to illuminate two or more segments.

Answers to questions in col 3.

1. *2 = Cd, 3 = Ad, 4 = 2E, 5 = A7, 6 = E 7, 7 = 29, 8 = EF, 9 = 2F.*

2. *A = 6F, b = E6, C = C3, d = EC, E = C7, F = 47.*

Example: To produce the number **ONE** on a display we must acces segments b and c. This requires the value 28 (in HEX) to be inserted into the following program at location 805:

```
800  3E 01
802  D3 01
804  3E 28
806  D3 02
808  76
```

QUESTION: What would result if **28** were placed at location 801 instead of 805?

Answer: The first and third displays would illuminate with segments "a".

This means you must take care to present the data to the correct output port. In fact every program must be ABSOLUTELY CORRECT as the computer will not be able to correct any of your mistakes.

PROBLEMS:

1. What Hex value must be inserted into the program above to produce the following numbers:

1 = 28
2 = Cd
3 =
4 =
5 =
6 =
7 =
8 =
9 =

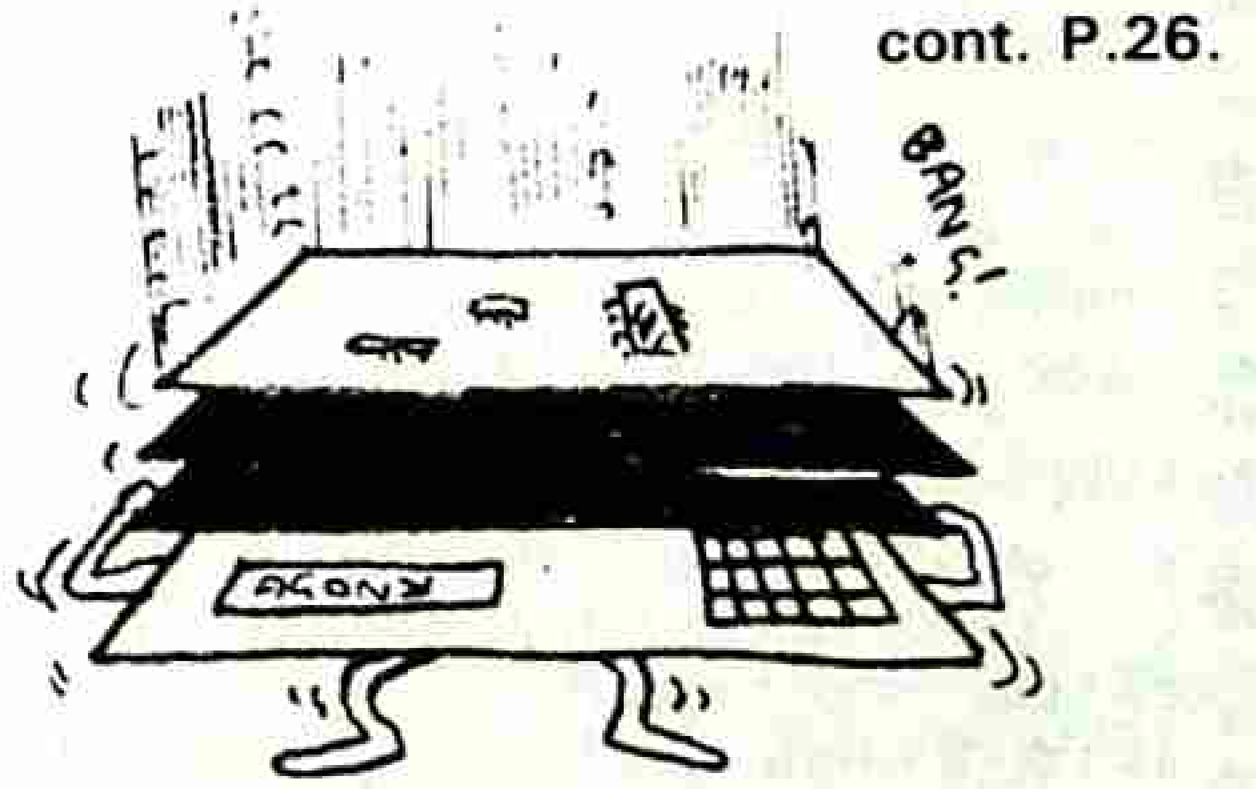
2. Work out the value to illuminate these letters:

A =
b =
C =
d =
E =
F =

3. *Draw the result of entering these Hex numbers into the program:*

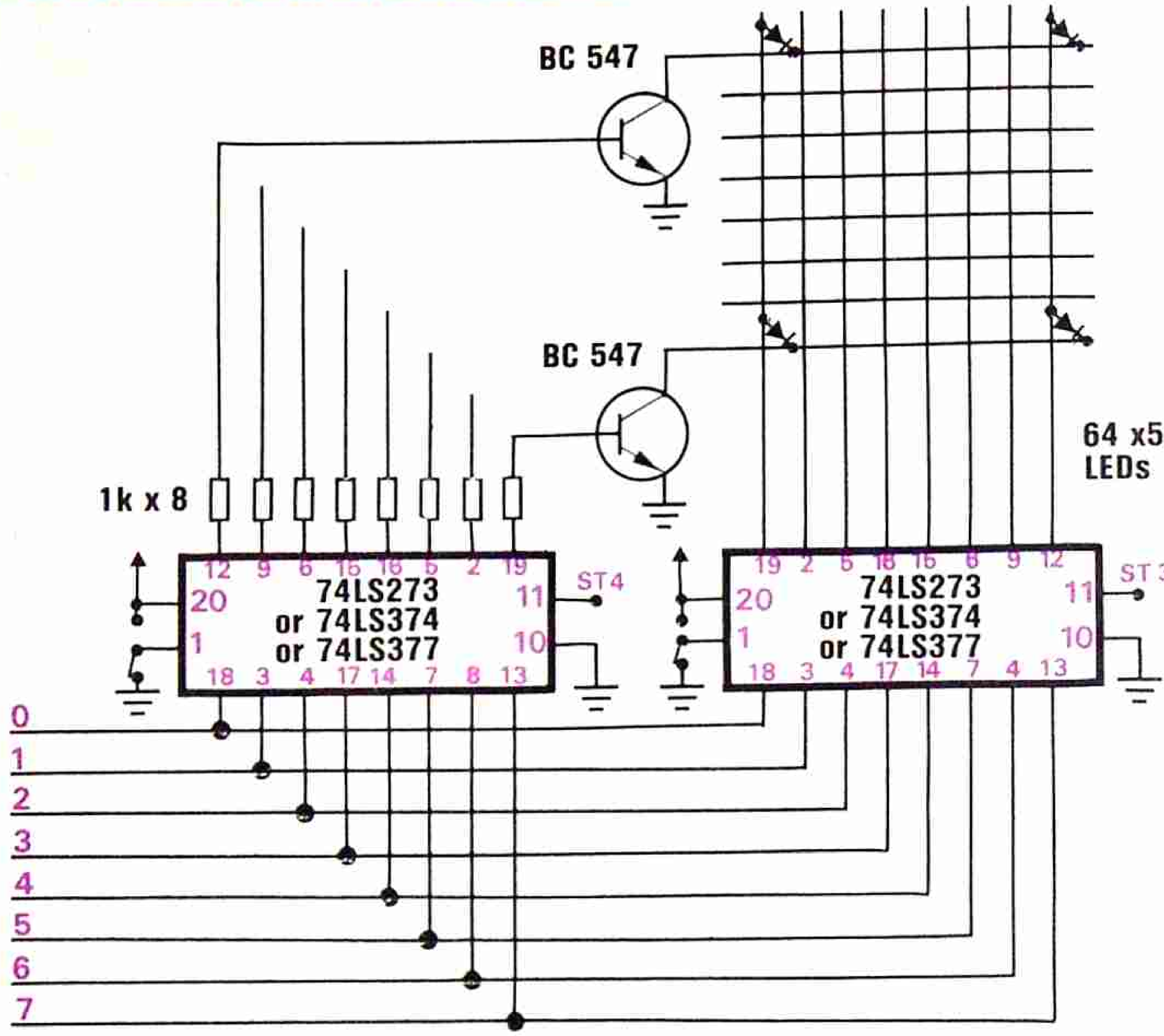
(a) 32 =
(b) 14 =
(c) 49 =
(d) 88 =
(e) A4 =
(f) C3 =
(g) DD =
(h) F1 =
(i) FE =

cont. P.26.



.. AND SPEECH SYNTHESIZER....

8x8 MATRIX



This is our first "add-on" for the TEC-1. It is an array of 64 LEDs arranged in a matrix of 8 LEDs by 8 LEDs. Actually it has almost the same number of LEDs as the display on the TEC but in this design the LEDs are arranged in ROWS to create a very interesting display.

The whole concept of the 8 x 8 matrix is to produce the equivalent of a WINDOW ON A VIDEO SCREEN.

Each LED represents one pixel and this will enable us to produce characters, letters and movement equal to 8 pixels by 8 pixels.

This may be only a small fraction of the area of video screen but it is the best place to start. If you can produce effects and movement on a small scale, a full-size VDU screen is only an enlargement of our 'window'.

If you have seen the advertising signs composed of thousands of LEDs or globes on which moving letters and characters are displayed, you will be interested to know the same effect can be produced with this project.

Modules of the 8x8 display can be placed side-by-side to create a long display. The PC board is designed to be cut so that the pattern runs as a continuous display.

At this stage it is not out intention to promote the extended display as it requires a slightly different driving circuit. To achieve a readable brightness with more than 8 columns of LEDs, it is necessary to introduce blocks of columns which are latched or even latching for each column. This will enable each LED to be turned on to full brightness and produce a bright display.

PARTS LIST

8 - 1k ¼watt

8 - BC 547 transistors

2 - 74LS273 or
2 - 74LS374 or
2 - 74LS377

64 - 5mm red LEDs

2 - matrix pins
2 - matrix connectors

30 cm hook-up wire, 12 colours.
30cm tinned copper wire

15cm - Heat-shrink tubing

1 - 24 pin DIP HEADER

1 - 8x8 DISPLAY PC BOARD

In our design, the LEDs are multiplexed and this means they are being turned on for one-eighth of the time during one cycle. The result is a dull display but one which can be read under normal lighting conditions.

We are presenting this project slightly ahead of time so that it will be ready when needed.

There are a number of MACHINE CODE instructions which can only be investigated on a display format and this project is a necessary part to understanding the Z80.

The greatest visual impact for this type of display revolves around programmed lighting and effects such as COCA-COLA signs and some of the background effects at discos and TV shows.

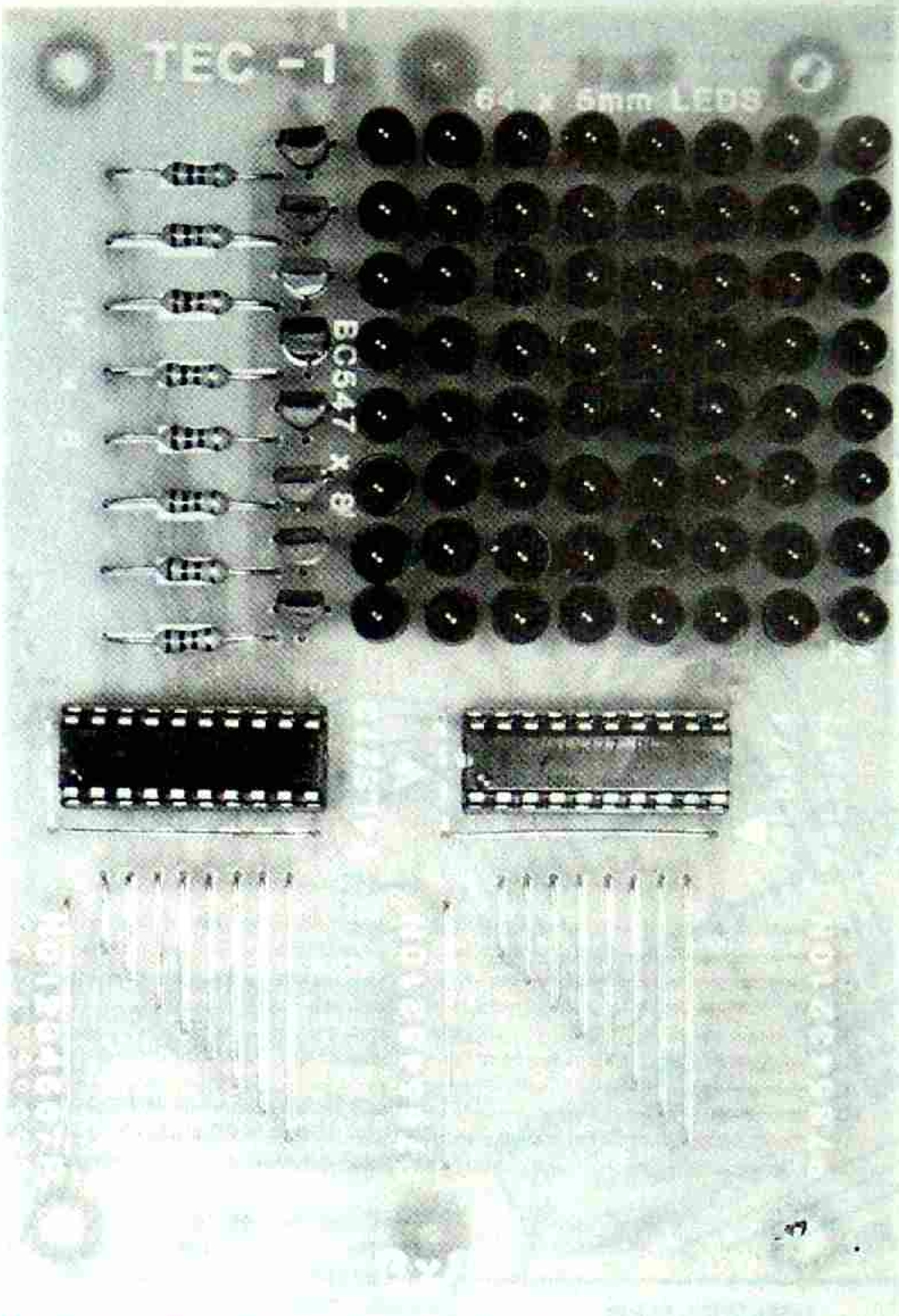
Most of the dazzling effects behind singers and dancers on TV have some form of micro-processor controlled lighting. The effects that can be produced are limitless.

We have chosen LEDs in our design for cheapness and simplicity but they could quite easily be replaced with miniature 6v or 12v globes. The only extra components would be the addition of one extra transistor in each line as an emitter follower. This will enable the extra current to be supplied to the globes.

Our 8x8 display is most effective when flashing blocks of LEDs (or globes) and having them jump from one position to another. In addition, bands of LEDs can be made to pass across or up the screen . These effects are very effective and very simple to produce.

But first you must understand the Machine Code instructions involved and how to include them in a program. This will be our endevour in the latter part of the course in this issue and the time is right to prepare the display project so that it can be plugged into the TEC-1 when the time comes.

Believe me, you will be most impressed with the results.



A full-size view of the display showing the neatness of the rows of LEDs. This is necessary if you want the best effect when the display is operating.

CONSTRUCTION

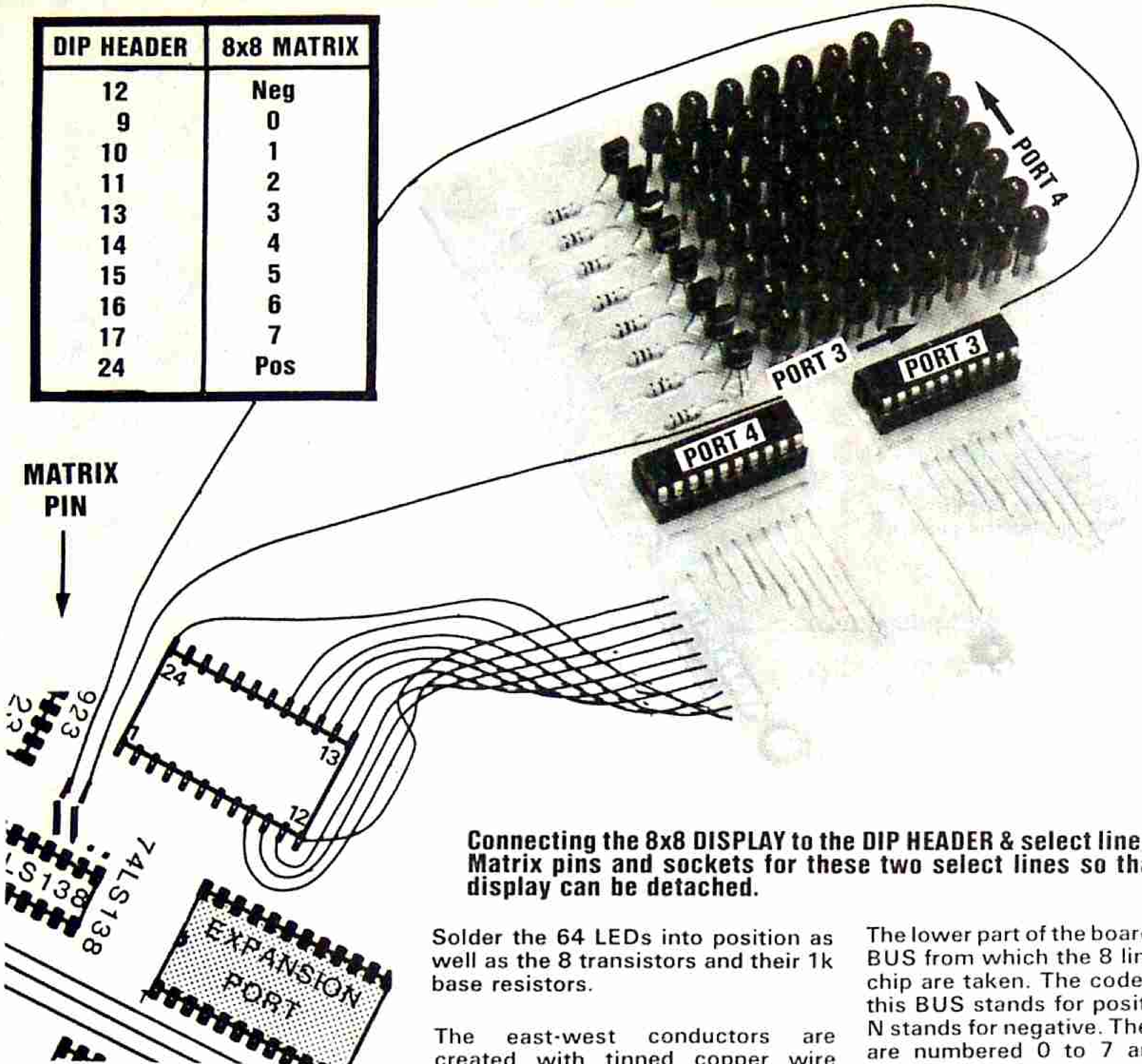
Before starting any of the construction, it is absolutely essential that you know which lead of the light emitting diode is the cathode. There is only one guaranteed way of determining this. You need a 3v to 6v battery and a 100 ohm or 220ohm resistor. Place the LED and resistor in a series circuit connected to the battery and check the degee of illumination. The cathode lead will be the one nearest the negative terminal of the battery. There are no other sure-fire methods of determining this as some LEDs have their long lead cut differently to the accepted practice.

We have seen some LEDs with the outline (inside the LED) around the opposite way to the general rule. So, it can be quite confusing. You must test each LED or at least a sample from the batch.

Next you must be certain which way they are to be inserted in the PC board. A mistake will take a very long time to rectify. The cathode lead is nearest to the row of transistors. When soldering the LEDs to the board, you must take special care to keep them all the same height and perpendicular to the board. The neatness of the dispay will depend entirely on how well you position the LEDs.

At first you may think one lead of the LEDs is not connected to the circuit. But this is where we have had to improvise. Multiplexing requires one line of conductors to travel north-south and the matching line to travel east-west. This would normally require a double-sided PC board, but since they are very expensive and difficult to solder, we have opted for the cheaper approach.

DIP HEADER	8x8 MATRIX
12	Neg
9	0
10	1
11	2
13	3
14	4
15	5
16	6
17	7
24	Pos



Connecting the 8x8 DISPLAY to the DIP HEADER & select lines. Use Matrix pins and sockets for these two select lines so that the display can be detached.

DIP HEADER

A DIP HEADER is a plug which has thin pins similar to the pins on an IC, on the underside. On the top are cup-shape (or 'Y' shape) terminals to which you can make a solder connection.

Leads can be soldered to the terminals to create a low-cost adaptor.

It is suggested that heat-shrink tubing be placed over each lead before soldering to the Dip Plug. When all the leads are attached, the sleeving is slid over each terminal so that the conductor is strengthened.

This will prevent fine wiskers of wire shorting from one pin to the other and creating havoc.

MATRIX PINS & SOCKETS

These are the cheapest and best way of connecting a single line to a printed circuit board.

Solder the 64 LEDs into position as well as the 8 transistors and their 1k base resistors.

The east-west conductors are created with tinned copper wire running along the ends of the LED leads and this connects to the collector of the driver transitor via the PC circuit. The only lead which has to be cut short is the anode lead, to prevent it touching the tinned copper wire.

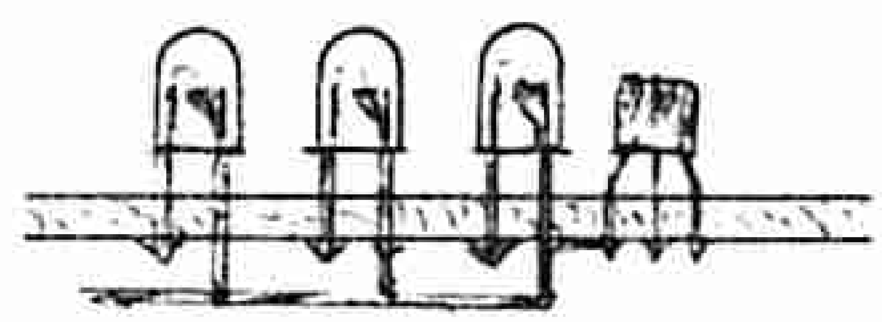


Diagram showing how the 'COMMON' line is created on the underside of the board. Both leads of each LED are soldered to the PC board. But only the ANODE lead is cut short. The CATHODE leads are either joined with a length of tinned copper wire which runs below the board, or each lead is bent over and soldered to the next lead to produce a rigid conductor which runs at right-angles to the copper tracks on the board.

The lower part of the board contains a BUS from which the 8 lines for each chip are taken. The code letter P on this BUS stands for positive and the N stands for negative. The other lines are numbered 0 to 7 and this co-incides with the data lines on the latches.

The two output latches can be: 74LS347, or 74LS377 or 74LS273 or a combination of any two. The information on the overlay shows which jumper link must be included for the type of latch you choose.

Eighteen jumper links connect between the data bus and the latches to complete the assembly. The only wiring left is the connecting wires between the DIP HEADER plug and the PC board. This plug is designed to fit into the expansion port socket on the TEC-1.

The 10 lines from the bus on the display board connect to the DIP plug and the two spare lines connect to the chip select outputs near the 74LS138 (near the keyboard encoder). These lines are for ports 3 and 4. Solder two matrix pins to these output holes and use a matrix-pin connector soldered to the hook-up wire to connect to these pins.

The 8x8 display is now ready for testing and we will give 3 simple programs to test the operation of the LEDs. This will check their illumination, their OFF response and the correct wiring of the data lines and chip select lines.

To check the brightness of the LEDs, insert this program at 800:

```
3E FF
D3 03
3E FF
D3 04
76
Reset
GO
```

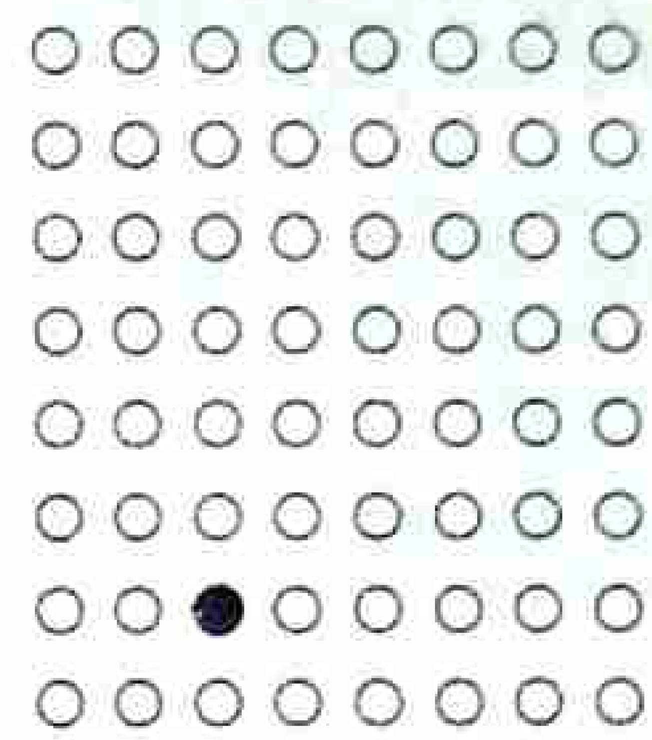
Replace any dull LEDs.

To make sure all the LEDs are extinguished when they are not being accessed, change the program above to:

```
3E 00
D3 03
3E 00
D3 04
76
```

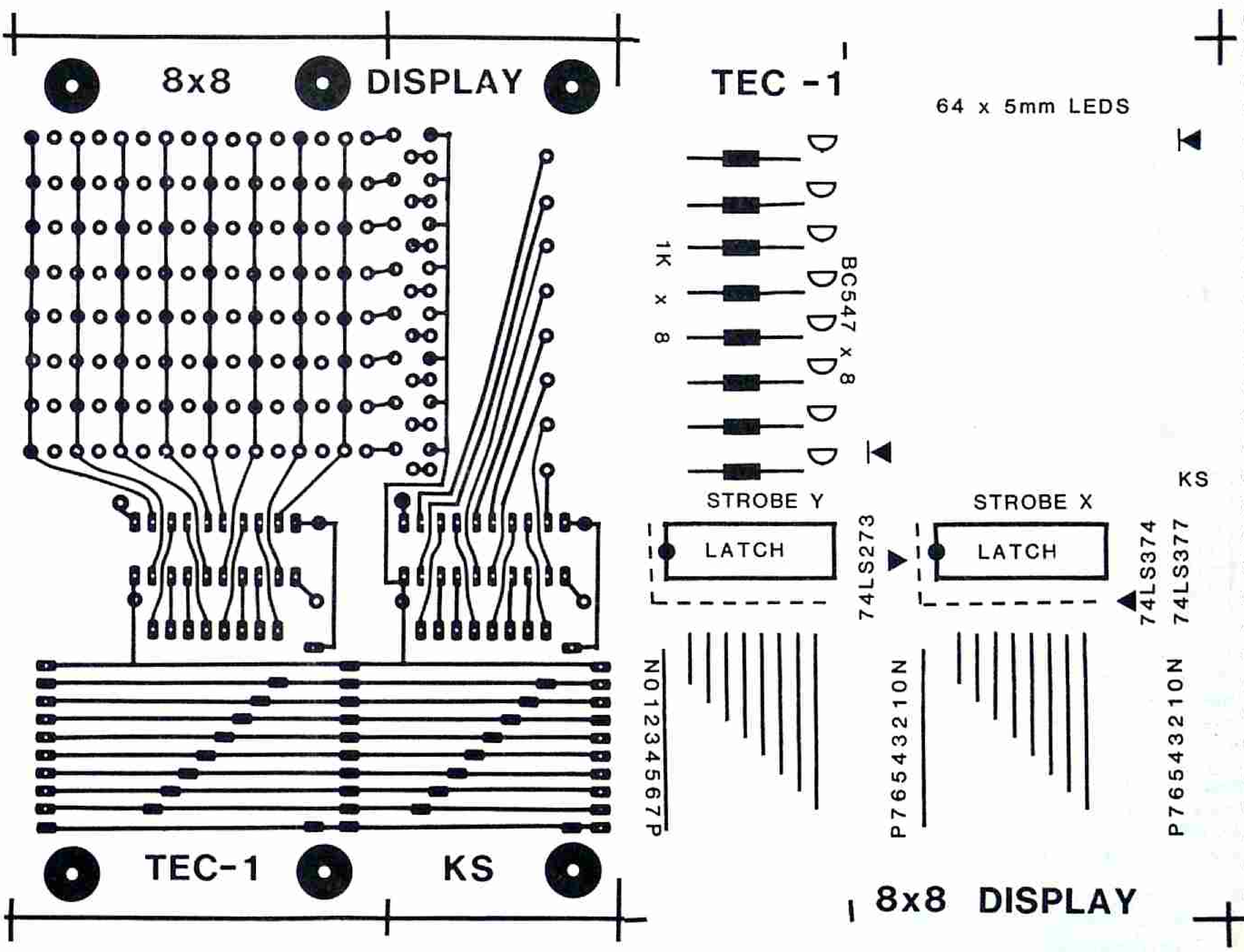
To check the data and chip select lines, insert this program:

```
3E 04
D3 03
3E 02
D3 04
76
Reset
GO
```



The LED which will illuminate is shown in the diagram. If any other LED illuminates, check the select lines.

Now go back to experimenting with the display on the TEC-1. When you get to p.30, you will be able to use the 64 LED display to create some startling effects.



The PC layout and overlay for the 8x8 Matrix. All LEDs face one direction and must be soldered as explained in the text. Apart from the jumpers connecting the BUS line, ONE link is required for each of the latch chips and this must be placed according to the type of chip you use.

....from P. 21.

ILLUMINATING TWO OR MORE DIGITS

More than one display can be illuminated at the same time and this is achieved by changing the value at 801 in the program above.

Example: To fill the six displays with the letter **A** we program the following:

```
3E 3F
D3 01
3E 6F
D3 02
76
```

Problems:

4. Fill the six displays with the following:
(a) 1's
(b) 5's
(c) b's
(d) E's

5. Place a 'C' at each end of the display.

6. Fill the first and last two displays with the value '8'.

7. Fill only the address displays with 4's.

8. Illuminate only segments a and d on the six displays.

TEC-1 AS A PROGRAMMABLE LOGIC DEVICE

· by John Hardy

The truly unique thing about computers is not that they can preform arithmetic in a twinkling of an eye but it is the way they can be used to simulate any digital (and almost any analog) circuit under the sun.

The microprocessor (Z80) can be likened to a bag of AND and OR gates, a thousand flip flops and tens of thousands of inverters.

You have an 8-bit data bus which means that you can simulate an 8 input NAND (with the aid of a program) and you can output 8 bits of data on this bus.

You are not simply restricted to 8 of this and 8 of that. You can output 16, 32, 64, or even 1024 bits, as long as you break it up into 8-bit groups.

By systematically dealing with 'bits', you can perform a multitude of digital functions.

The TEC-1 is first and foremost a binary computer. While superficially it appears that the computer operates on hexadecimal numbers (9B, 3E, 2C etc.) deep in the heart of the computer binary numbers are the norm.

The problem with binary numbers is their unfamiliarity to humans. Imagine if you wrote a program in binary and made a mistake. It would be very difficult to spot. Take the following example. Can you see the difference between the two?

```
01011010    01011010
10110101    10110101
11001111    11001111
10110101    10110111
11101011    11101011
```

It is possible to check for binary errors but they don't show up easily. Hexadecimal is a short-hand way of representing binary. It is based on breaking up an 8-bit binary number into two 4-bit numbers and converting these into two hexadecimal digits.

Comparing the two sets of numbers above, the difference is quickly spotted when they are converted to hex values as shown below:

```
5A    5A
B5    B5
CF    CF
B5    B7
EB    EB
```

Hex was therefore chosen for use in the TEC-1 but we must never forget that ALL DIGITAL COMPUTERS WORK IN BINARY.

When dealing with computer problems, we should always visualize the inner registers as holding 'bits' and that the computer performs BINARY operations. We then convert to Hex after this. While this might seem awkward, the conversion between Hex and binary can be done quite quickly after a little practice.

- John.

CREATING MOVEMENT!

All the programs up to now have been static.

We will now create some life and movement!

This will introduce a SHIFT or ROTATE function into the program. The rotate function we have selected is located at 80C in the program which follows. This is a two-byte instruction and tells the Z80 to rotate a HIGH bit left circular through the B register. You will understand what we mean by this statement in a few minutes.

This shift operation will take 8 DELAY PERIODS to complete one cycle and will include toggling or clicking the speaker.

RUNNING SEGMENT 'a' ACROSS THE SCREEN

The Program: **CB 00** runs segment ← **CB 08** runs segment →

at 0800:

```
LD A,01        800   3E 01
OUT (2),A      802   D3 02
LD B,01        804   06 01
LD A,B         806   78
OUT (1),A      807   D3 01
CALL DELAY     809   CD 00 0A
RLC B          80C   CB 00
JP LOOP        80E   C3 06 08
```

at 0A00:

```
0A00   11 FF FF
       1B
       7B
       B2
       C2 03 0A
       C9
```

This is what the program is saying and instructing the Z80 to do:

The first instruction is to load register A with the value 1.

This is then passed to the SEGMENT PORT latch and this value remains fixed for the whole program.

The remainder of the program concerns port 1, the CATHODE PORT and as the different cathode are accessed, the effect is to run a pattern across the screen.

The next instruction is to load register B with the value 1. This value is then loaded into register A via the instruction 78. The reason for this will be explained in a moment.

The contents of A are now outputted to port 1 with the result that segment 'a' on the lowest priority display will be lit.

We now call a DELAY ROUTINE so that this display will be illuminated for about half a second.

The HIGH bit in register B is then shifted RIGHT. This is performed within register B by the Z80. The program is then incremented to the next instruction and this tells the Z80 to jump to address 806.

The output of the DELAY ROUTINE appears in register A and when this value is zero, the delay routine returns to address 80C.

This means we must use another register to provide our shift routine and in this case we have chosen register B.

Quite a number of variations can be produced with this program by changing the data at some of the locations. These can be carried out after the main program has been entered.

The main program starts at 800 and the delay routine is located at 0A00.

Now try these variations:

1. To run the segment left-to-right, change location **80d** from **00** to **08.**

2. To increase the SPEED of the display: Change A02 from FF to 0F or 06.

3. To run segments 'a' and 'd' across the screen: Change location 801 from 01 to 81.

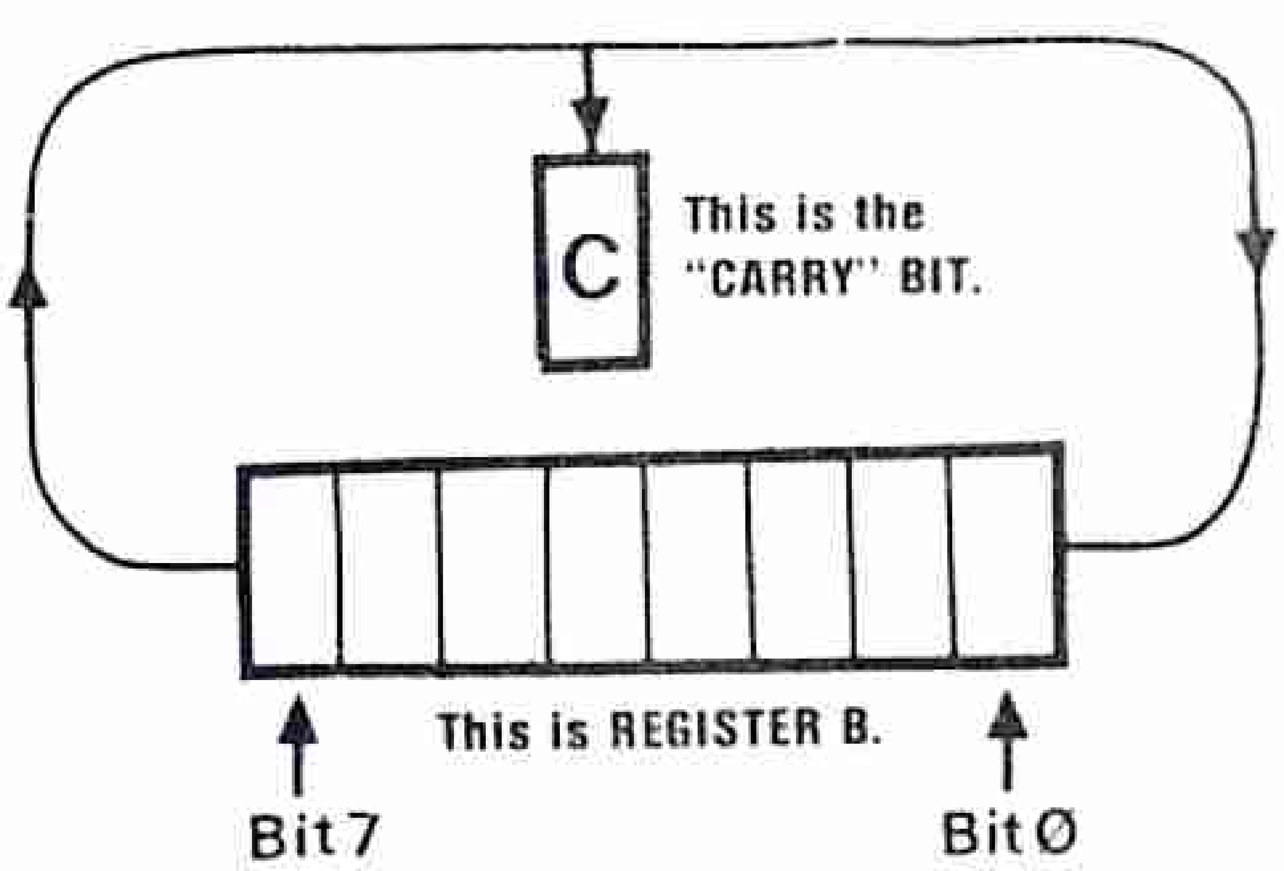
4. To run the number 7 across the screen, change location 801 to **29.**

5. To run the letter A across the screen, change location 801 to **6F.**

The program we have investigated introduced the ROTATE REGISTER B LEFT instruction **CB 00** and ROTATE REGISTER B RIGHT instruction **CB 08.**

RRC B = CB 08

RLC B = ROTATE LEFT CIRCULAR REGISTER B.



The diagram shows register B as 8 boxes. These can be considered as flip flops. The lowest value flip flop is at the right hand end of the row and is labelled Bit zero (Bit 0). This is the Least Significant Bit (LSB).

The Most Significant Bit (MSB) is called Bit 7.

The instruction RLC B has a Machine Code instruction **CB 00** and this causes the most significant bit to emerge from the register and enter it again to become the least significant bit. In this process it does not pass through the CARRY bit but does set the C flag to the original status of the register's most significant bit.

In other words, if the bit in question is a HIGH, the C flag becomes HIGH, if the bit is LOW, the C flag goes LOW.

RLC B = CB 00

RRC B = ROTATE RIGHT CIRCULAR REGISTER B.

This instruction is a reversal of the path shown above. The C flag, however, is altered as above. The ONLY difference between the two instructions is the direction of rotation.

The point to remember in these Machine Code operations is RLC and RRC can be performed on registers A, B, C, D, E, H and/or L and are 8-stage shift operations.

In the next program, on P.28, the instruction which will produce a shift operation across the screen is the instruction **RRA** or **RLA.**

After each shift is performed, the contents of the 'A' register must be 'hidden' or SAVED to prevent it being destroyed.

To do this we must load the contents of register A into another register before calling the DELAY ROUTINE. We could load it into B, C, D or even E register and load it back again when required.

However this will tie up one more of our valuable registers and a better solution is to call upon 2 interesting instructions which load the contents of A into an area of RAM in the 6116 chip.

The code word for saving the contents of a register is called PUSH and recalling it is POP.

The PUSH instruction will take the contents of register A to an area called the STACK.

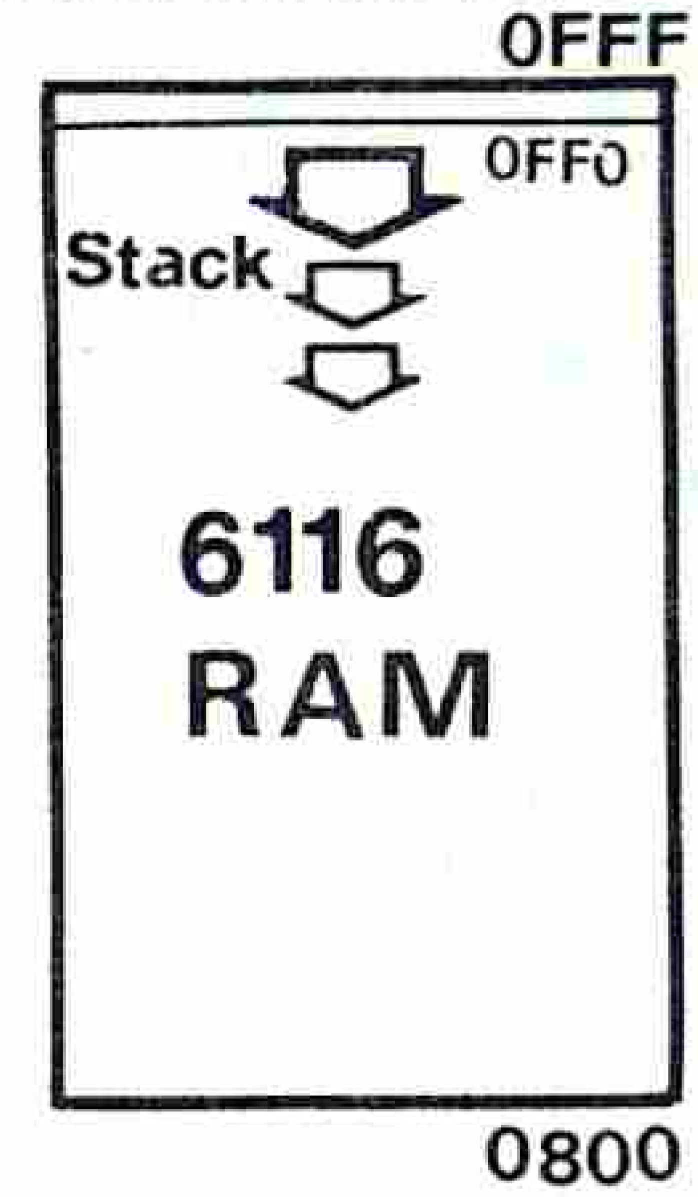
This area is located in the 6116 RAM at address OFFO. (This is only 16 bytes from the end of this chip's memory and is usually considered to be the unused end of the RAM.)

The highest 16 bytes are used as a scratch-pad area.

The PUSH and POP instructions are similar to stacking plates or trays in a pile. Trays are "pushed" or piled onto the top of the stack and are "popped" or removed from the top.

In the computer the area called the STACK is filled DOWNWARDS. This is an ideal way of using the top part of the RAM and it can be increased in size until it meets the program.

Thus we start with address 0FF0 and work downwards thus: 0FEF, 0FEE, 0FED etc. To keep track of the last address, the Z80 has a register called SP. This is the STACK POINTER register and always points to the byte with the lowest address.



The STACK starts at 0FF0 and heads DOWNWARDS in the 6116 RAM. The data in the EPROM decides this and is 0FE0 when using MON-1B EPROMS.

The Z80 has two instructions for operating on the stack. These are PUSH and POP (or Pull). Both instructions require a register PAIR (such as HL, AF, BC, DE) to be specified as the SOURCE for PUSH and the DESTINATION for POP.

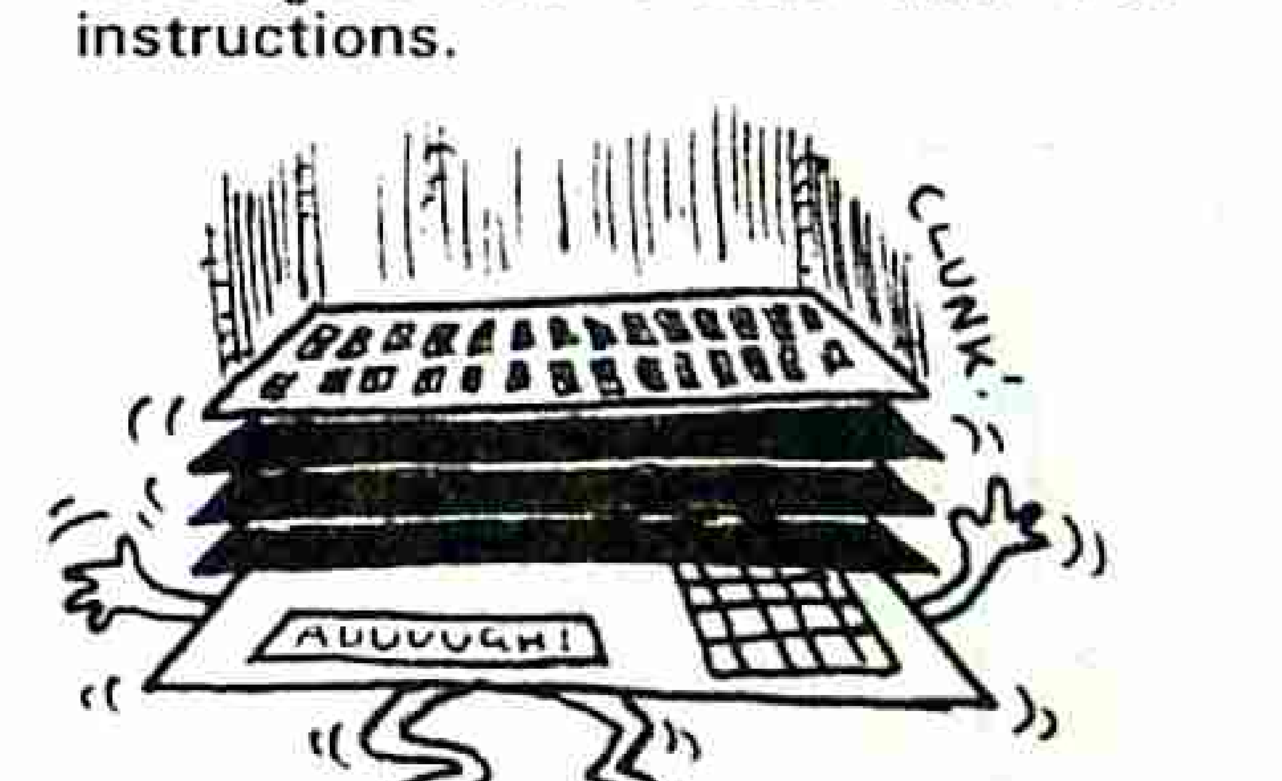
We PUSH new bytes onto the stack and POP bytes off the top.

The Z80 processes this operation TWO BYTES AT A TIME and results in a new byte on the top of the stack with either operation.

The top byte has the lowest address and the memory is filled downwards. The STACK POINT register decreases with a **PUSH** instruction and increases with a **POP** instruction.

Bytes are entered onto the stack, HIGH byte first, then LOW byte. The bytes are removed LOW byte first, then HIGH byte.

In the next program we will investigate the PUSH and POP instructions.



.... AND MEMORY EXPANSION ..

To run the 'g' segment from Left-to-Right:

at 800:

LD A,04	800	3E 04
OUT (2),A	802	D3 02
LD A,01	804	3E 01
OUT (1),A	806	D3 01
RRA	808	1F
Push AF	809	F5
CALL DELAY	80A	CD 00 09
POP AF	80D	F1
JP 806	80E	C3 06 08

at 0900:

THIS IS THE DELAY ROUTINE	11 FF 06
	1B
	7B
	B2
	C2 03 09
	C9

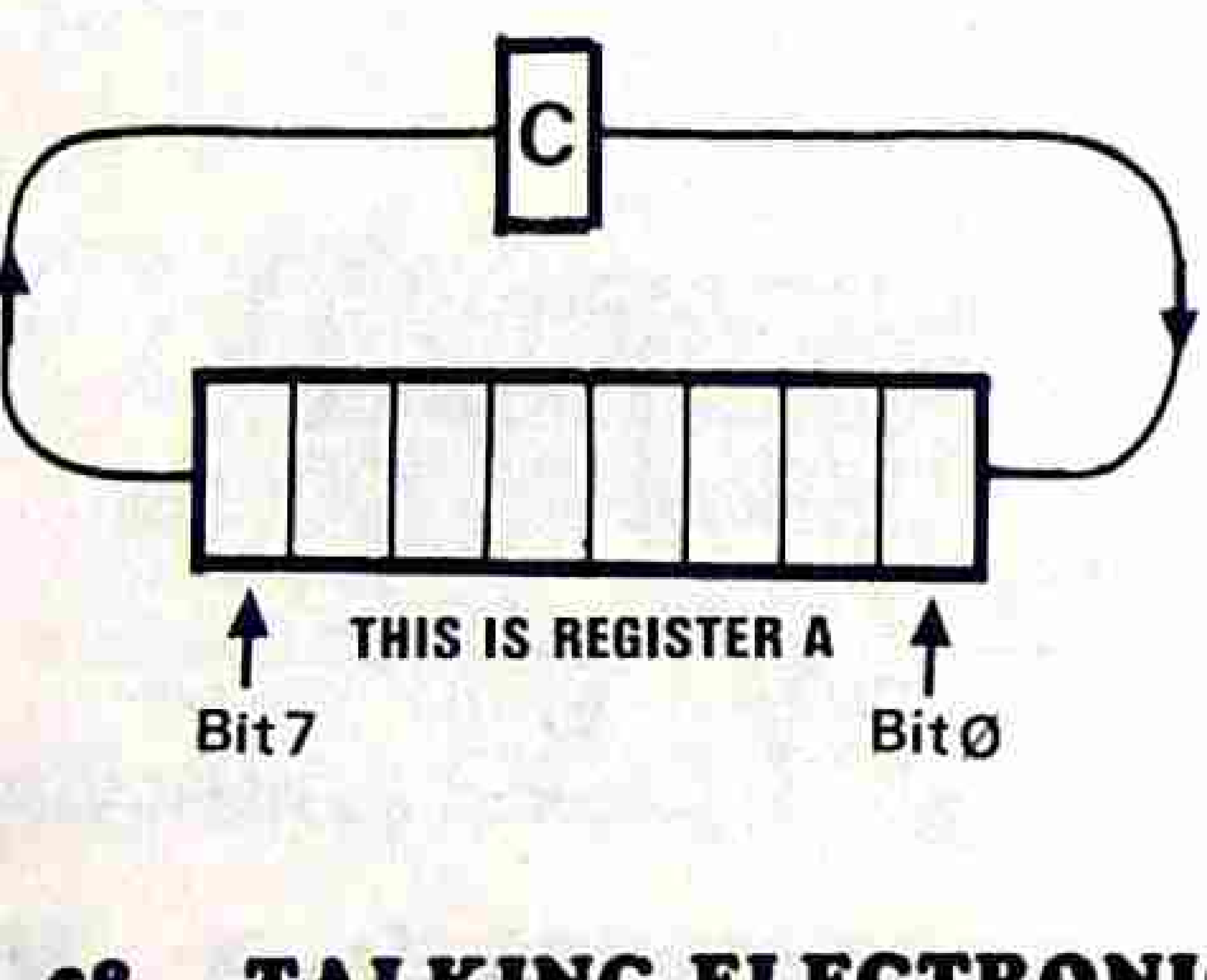
Using the program above, change the address location 808 to 17. This is the machine code instruction for rotating the accumulator left. (RLA), through the carry.

Machine codes covered:

RRA = 1F
RLA = 17

RRA & **RLA** have nothing to do with moving left or right on the display. They refer to shifting the information through the accumulator via the carry. This means it is a nine-stage shift in which no output is activated when the bit is shifted into the 'carry'. This effect can be seen on the displays when a long delay routine is employed. The illumination travels across the 6 displays, the next output is not used and then the speaker/LED combination is activated. A delay is then noticed before the illumination re-appears on the screen.

The main difference between the two programs is the number of SHIFT STAGES. The first program produced an 8-stage shift while the second produced a 9-stage shift due to the carry bit becoming loaded with each bit from the register as it circulated as shown in the diagram.



To create a FLASHING SEGMENT

Routine at 800:

LD A,01	800	3E 01
OUT (2),A	802	D3 02
LD A,01	804	3E 01
OUT (1),A	806	D3 01
CALL DELAY	808	CD 00 09
LD A,00	80B	3E 00
OUT (1),A	80D	D3 01
CALL DELAY	80F	CD 00 09
JP LOOP	812	C3 04 08

Delay Routine at 0900:

900	11 FF 07
903	1B
904	7B
905	B2
906	C2 03 09
909	C9

The exact operation of the delay routine is not important at this stage. It is enough to know that it creates a delay of length determined by the number loaded into register-pair DE. If this number is 01 00, the delay will be only a few microseconds. The first byte refers to register E and this is the lower register while the second byte is the higher register and has the greater effect on the delay.

Try putting different values into location 902 to vary the length of the delay. A value such as **02** will increase the flash-rate while **FF** will create the slowest flashing.

The format of the main routine is very simple. It is an ENDLESS LOOP which means it executes part of the program over and over again.

The 'BIT' patterns for the segments to be lit are loaded into the segment register (port 2). Cathode 1 is then turned on and the delay routine is called.

The cathode register is then cleared and the delay routine is called again.

This creates the OFF cycle.

The program then jumps back to address 804 where it is instructed to turn on cathode 1. This causes segment 'a' to come on once again.

You can flash segment 'g' by loading **04** into the program at 802 thus:

3E 04
D3 02
etc...

Create flashing numbers and letters in the display by inserting the appropriate hex numbers as discovered in questions 1 & 2 on P 21.

You can also use this program to alternate from one number or letter to another. This is achieved by the second letter taking the place of the blanking routine in the program above.
Insert the value **28** at location 80C and run the program. What happens?

The segment 'a' alternates between one display and two other displays. Turn the speed of the computer down to observe this. But this is not what we wanted. We want different segments of the same display to be turned on. We have forgotten to change location **80E** from **01** to **02.**

Run the program and note segment 'a' changes once to a figure '1' and appears to be stuck on this figure.

There is a second fault in the program. Only the second part of it is being cycled.

Change location **813** to **00.** The program will now alternate between the 'a' segment and the figure '1'.

This is the introduction to simple cartooning on the screen. Try changing locations **801** and **80C** to get some interesting effects.

RUNNING AROUND THE DISPLAY

To run a single illuminated segment around the display takes a considerable amount of programming. There are a number of ways of doing this and we will use a program which uses some of the features we have covered so far.

Basically what we are doing is defining our start co-ordinates, shifting a 'bit' six places to the left and halting.

The next part of the program loads the co-ordinates of the side segment (at the top of the display) and then the lower end segment is lit.

We then define the co-ordinate on the bottom row and run the illuminated LED across the bottom of the display.

Finally we define the bottom side segment and the top side segment to arrive back at the starting point.

This will create an endless run around the display.

We will produce this program in 4 stages and check its operation at each stage.

"AROUND THE DISPLAY"

```
LD A,01          800   3E 01
OUT (2),A        802   D3 02
LD C,06          804   0E 06
LD A,01          806   3E 01
OUT (1),A        808   D3 01
LD B,A           80A   47
CALL DELAY       80B   CD 00 09
LD A,B           80E   78
RLC A            80F   CB 07
DEC C            811   0D
JP NZ,LOOP 1     812   C2 08 08
HALT             815   76
```

Push RESET, GO.

If the LED runs across the top of the display and HALTS, everything is working.

Press RESET,ADdress 815 +

Now insert the following program so that the **HALT** instruction is written over and is removed from the program.

```
LD A,02          815   3E 02
OUT (02),A       817   D3 02
CALL DELAY       819   CD 00 09
LD A,40          81C   3E 40
OUT (02),A       81E   D3 02
CALL DELAY       820   CD 00 09
HALT             823   76
```

Check the program at this stage by running it. If the LED travels across the top and down one side, it is working. Over-type 3E at address 823 and continue with the 3rd stage:

```
LD A,80          823   3E 80
OUT (02),A       825   D3 02
LD C,06          827   0E 06
LD A,20          829   3E 20
OUT (01),A       92B   D3 01
LD B,A           82D   47
CALL DELAY       82E   CD 00 09
LD A,B           831   78
RRC A            832   CB 0F
DEC C            834   0D
JP NZ,LOOP 2:    835   C2 2B 08
HALT             838   76
```

If all is ok, type the last part of the program:

```
LD A,20          838   3E 20
OUT (02),A       83A   D3 02
CALL DELAY       83C   CD 00 09
LD A,08          83F   3E 08
OUT (02),A       841   D3 02
CALL DELAY       843   CD 00 09
JP START         846   C3 00 08
```

Delay Routine at 0900:

```
11 FF 06
1B
7B
B2
C2 03 09
C9
```

Don't forget to add the DELAY ROUTINE.

The overall speed of the sequence can be varied by adjusting the SPEED control on the TEC-1.

More programs for the TEC-1 using its own display will be presented in the next issue.

MON-1A

Some of the latest kits of the TEC-1 have included a monitor EPROM marked Mon 1A. This EPROM will work in both the TEC-1 and TEC 1A as both are software compatible with each other.

The difference between Mon 1 and Mon 1A is a small additional routine at 05B0. This program was originally designed for use with music synthesisers but can also be used for a number of other applications.

The routine is a simple sequencer. It reads the data stored in RAM and deposits it at a fixed rate into the output latches.

The overall speed of the sequence can be varied by adjusting the SPEED control on the TEC.

There are two sequencing functions being performed in this program, one depositing information to its relavent latch (04) at TWICE the speed of the other (03).

The two sequences are synchronised and one output falls mid-way between the other. However the sequence-length is independent.

The end of the sequence is marked by placing an FF after the last piece of data. The sequence will then reset itself to the beginning. The other sequence will continue unaffected until it also hits an FF.

Because FF has been used to indicate the end of the sequence, you cannot use FF as a piece of data. In our application, this presents no problem, but when used with the relay board, it means all 8 relays cannot be activated at the one time.

We can go as high as FE without upsetting the program and this will turn on 7 relays, but not the lowest priority relay.

The slower sequence outputs to latch 03 and reads its data from address 0800 until it encounters FF and then resets.

The faster sequence outputs to latch 04 and reads its data from address 0B00 until it encounters FF and then it resets.

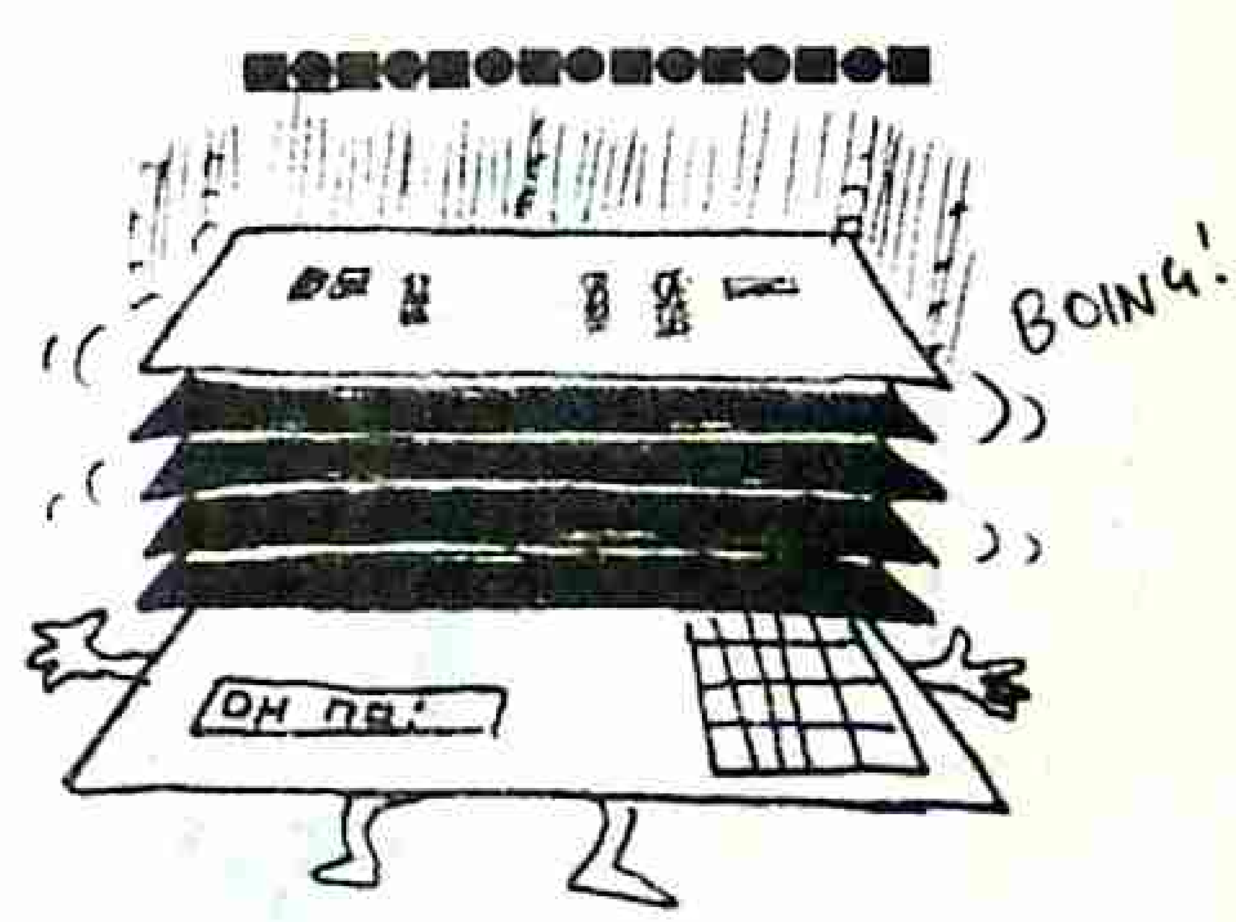
It should be noted that high memory is used by the Z80 to store its stack and thus memory above 0F00 should not be used.

A disassembly and Hex listing for this routine is given below:

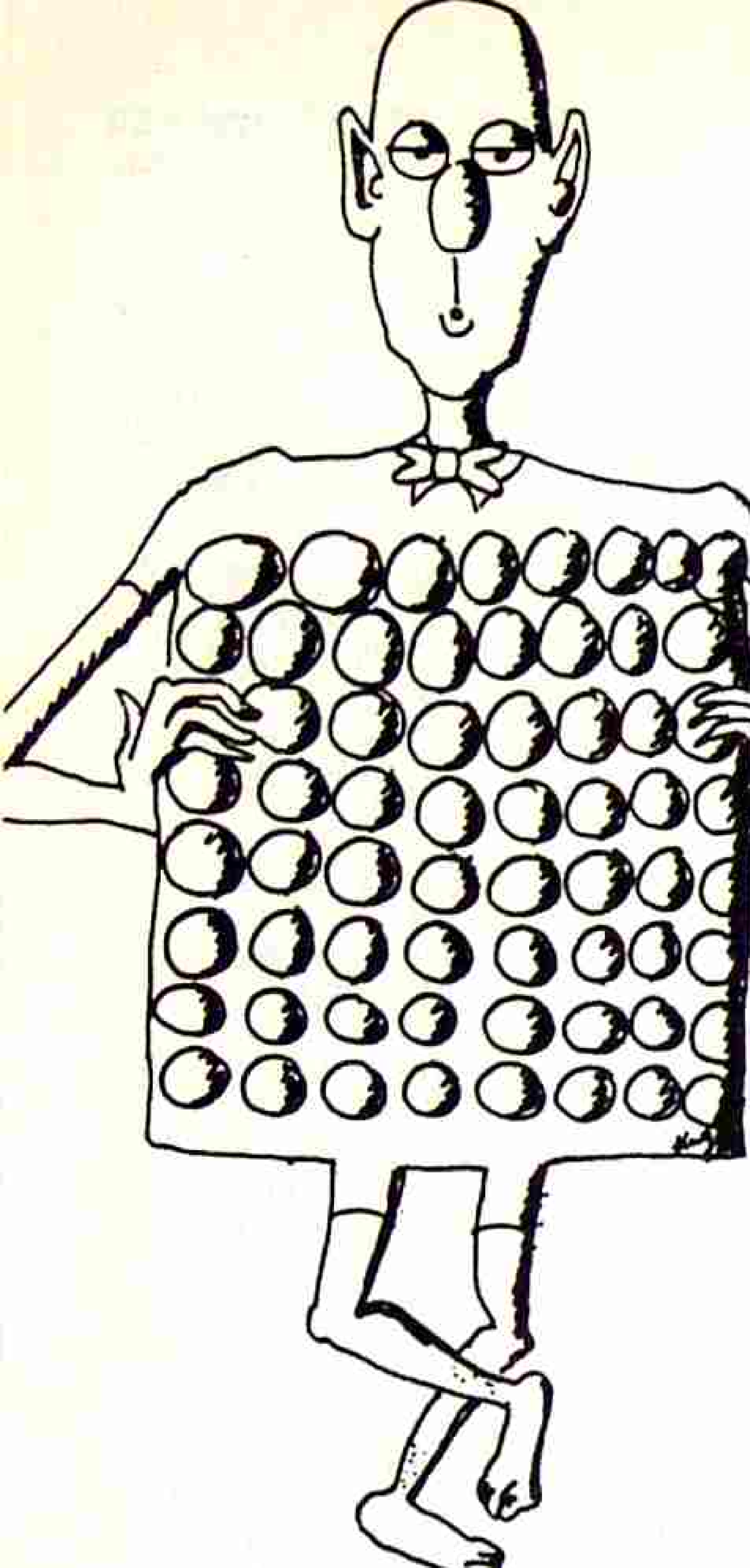
```
05B0   21   LD HL,0800
05B3   11   LD DE,0B00
05B6   7E   LD A,(HL)
05B7   FE   CP FF
05B9   C2   JPNZ,05C2
05BC   21   LD HL,0800
05BF   C3   JP 05B6
05C2   D3   OUT (03),A
05C4   1A   LD A,(DE)
05C5   FE   CP FF
05C7   C2   JPNZ,05D0
05CA   11   LD DE,0B00
05CD   C3   JP 05C4
05D0   D3   OUT (04),A
05D2   CD   CALL 05E1
05D5   13   INC DE
05D6   1A   LD A,(DE)
05D7   D3   OUT (04),A
05D9   CD   CALL 05E1
05DC   13   INC DE
05DD   23   INC HL
05DE   C3   JP 05B6
05E1   01   LD BC,03FF
05E4   0B   DEC BC
05E5   78   LD A,B
05E6   B1   OR C
05E7   C2   JPNZ, 05E4
05EA   C9   RET
```

Hex Listing:

```
05B0   21 00 08 11
05B4   00 0B 7E FE
05B8   FF C2 C2 05
05BC   21 00 08 C3
05C0   B6 05 D3 03
05C4   1A FE FF C2
05C8   D0 05 11 00
05CC   0B C3 C4 05
05D0   D3 04 CD E1
05D4   05 13 1A D3
05D8   04 CD E1 05
05DC   13 23 C3 B6
05E0   05 01 FF 03
05E4   0B 78 B1 C2
05E8   E4 05 C9
```



.... AND DIGITAL TO ANALOG INTERFACE



8x8 MATRIX

The possibilities and effects on a MATRIX layout are infinite. We will allocate the next few pages to showing some interesting visual effects.

Firstly we will show how each of the LEDs is accessed.

As with any matrixing system, each location has a set of co-ordinates. If we compare our display with the x and y axes in geometry, we find the x-axis has the lower output port number and the y-axis the higher number.

The output ports allocated to this display are 3 and 4 and this is determined by the chip access lines on the main board. Each line from the 74LS138 has a particular number and we have selected lines 3 and 4.

On the display board, each of the LEDs has a particular co-ordinate value which must be in the form of a Hex number. Each successive row or column has a hex number which is DOUBLE the previous number. The following diagram shows this:

The lowest priority LED has the value 01, 01 and the highest LED 80, 80. The value of each LED between these limits is also given, as well as the value for 4 individual LEDs, as a guide.

Placing these hex values into a simple program will illuminate any particular LED on the screen.

Here is the general program:

```
3E Hex value: ←→
D3 03
3E  Hex value: ↕
D3 04
76
```

IF THE 8x8 MATRIX DOESN'T WORK

On P.28 of this issue we described the construction of the 8x8 matrix and presented 3 short programs to test the LEDs in the display.

Hopefully you will have put the project together by now and will be ready to explore its capabilities.

The main difference between this project and the display on the TEC-1 is not so much the number of LEDs, but the way in which they are arranged.

We have created a regular matrix of 8 LEDs by 8 LEDs and this produces a screen very similar to a window on a video display.

The most common fault will be one or two of the LEDs failing to illuminate when the whole screen is accessed.

If this is the case, or if one is dull, the fault will be a damaged LED. LEDs are temperature sensitive, and excess heat when soldering will damage them. On the other hand, it may be a poor quality LED in the batch.

If any of the LEDs are particularly dull, they should be replaced at this stage to produce a good display.

Here are some of the possible faults and their remedies:

If a row or column fails to light, the fault will be in one of the output lines of a latch or one of the driver transistors. Make sure it is not a dry joint or a missing link and then check the orientation of the transistors and the LEDs.

If a row and column is failing to illuminate, the fault will lie in a shorted LED at the intersection.

Remove the LED and turn on the remainder of the screen. If the remainder of the LEDs come on, the fault is a short.

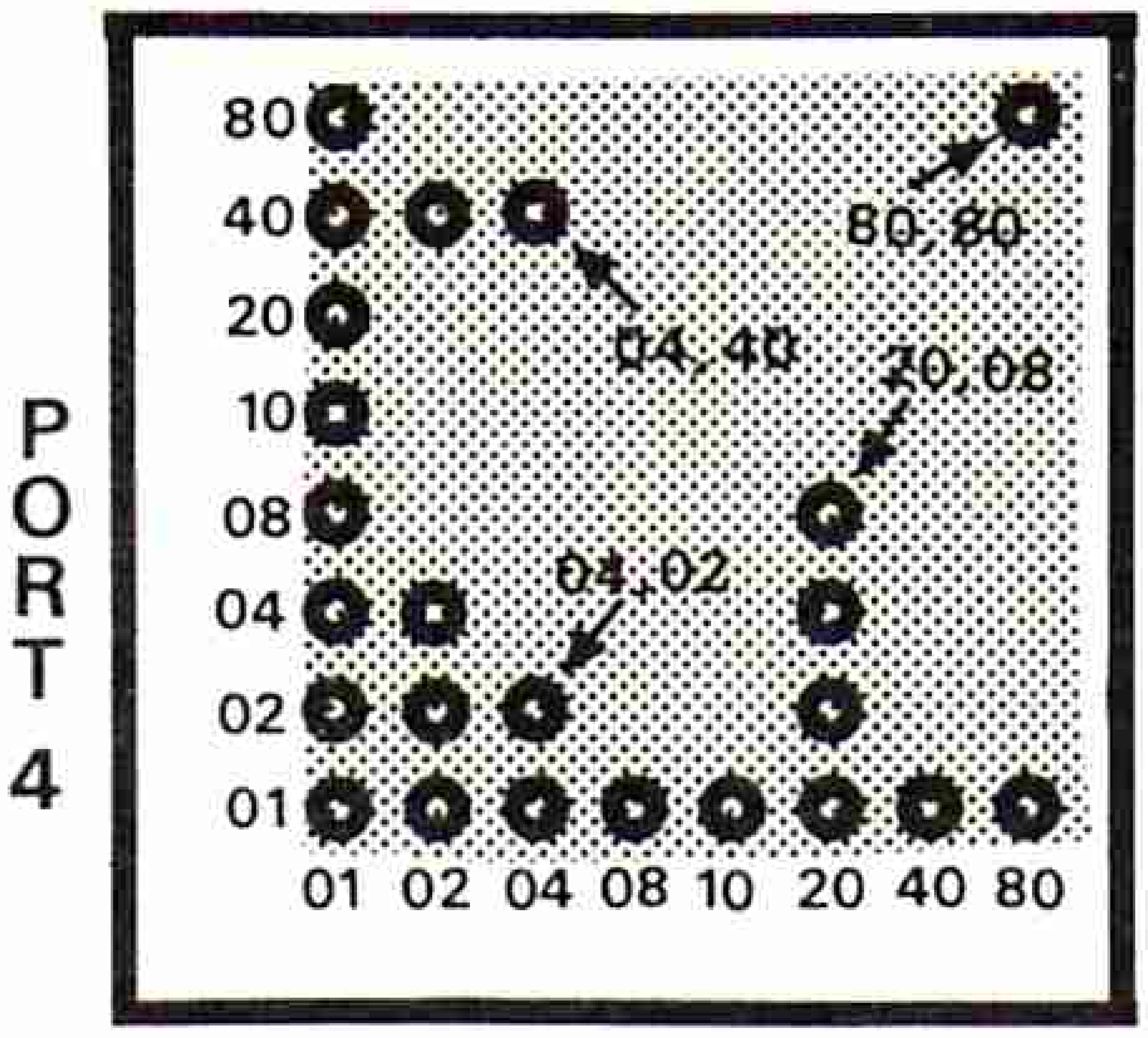
The only other fault we have seen is one row glowing brighter than the rest. This can be due to one of the transistors shorting between collector and emitter. A short to base may cause the row to be extinguished.

If all these suggestions fail to locate the fault, turn the TEC-1 off and re-program the set of instructions. Check to see that you have loaded FF into both port 3 and port 4.

Check both ends of the connecting leads and make sure they are connected correctly to the pins on the dip plug.

Since the expansion port socket is effectively in parallel with the other memory chips, it is very unlikely the the PC tracks will have shorts between them.

This means you should look mainly on the display board itself.



PORT 3

Diag 1: The ports and their Hex values.

If we take a particular case and load the co-ordinates 04, 02 into the program:

```
3E 04
D3 03
3E 02
D3 04
76
```

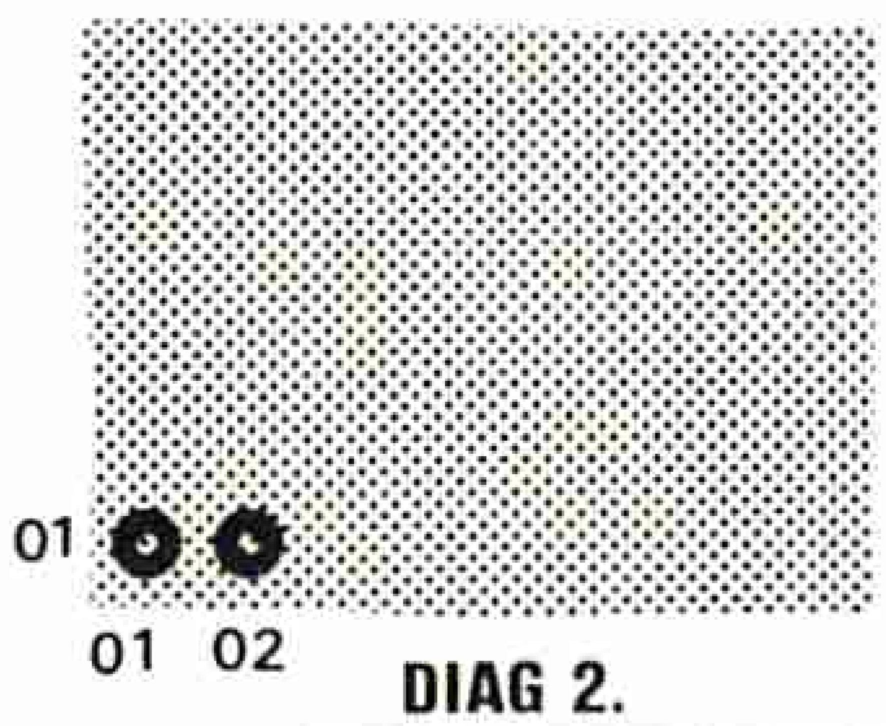
As you type the program, this is what you should be saying: Load the accumulator with 4, output it to port 3. Load the accumulator with 2 and output it to port 4. Halt.

Problems:

Illuminate 3 of the other LEDs by inserting the following data into the program:
1: 04,40
2: 20,08
3: 80,80.

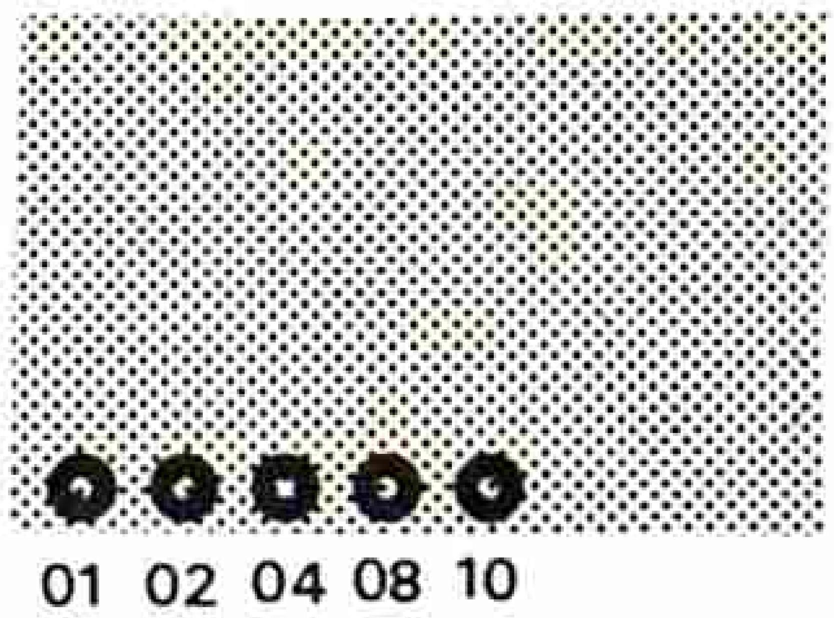
TWO OR MORE LEDs

More than one LED can be illuminated in any row or column by adding the Hex value of each LED. We will start with the simplest case but absolutely any LEDs in any row or column can be illuminated.



DIAG 2.

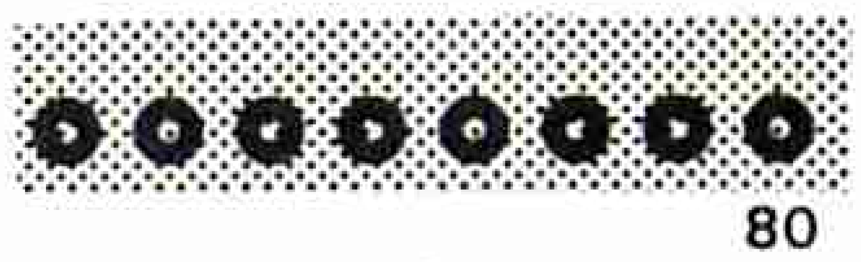
In diagram 2, two LEDs are shown illuminated. These have co-ordinates 01,01 and 01,02. To turn on both of these LEDs we add the bottom Hex numbers. The result is 03. Place this value into the program at address 801.



Diag 3.

Diagram 3 shows five LEDs illuminated. Add the Hex numbers together and insert it into the program and see if you are correct.

Did you get 1F?



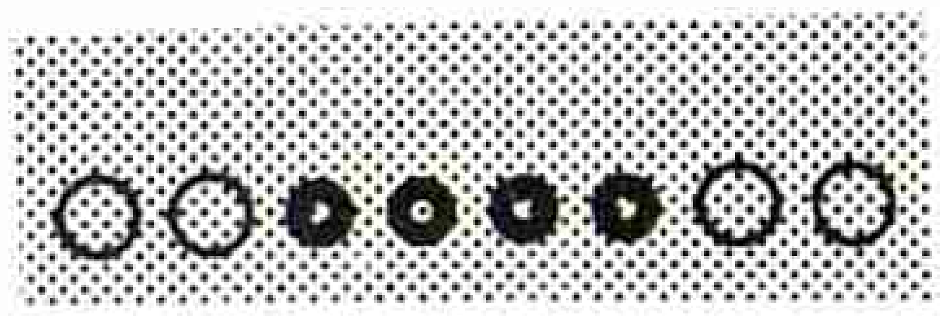
Diag 4.

The fourth diagram shows ALL the LEDs on the bottom row illuminated. What value must be placed in the program at 801 to access these LEDs?

The answer if FF. This is obtained by adding 01, 02, 04, 08 10, 20, 40, 80, this gives: 0F + F0 = *FF*

Problem:

Load the program with a hex value which will illuminate the four LEDs in the centre of the bottom row:

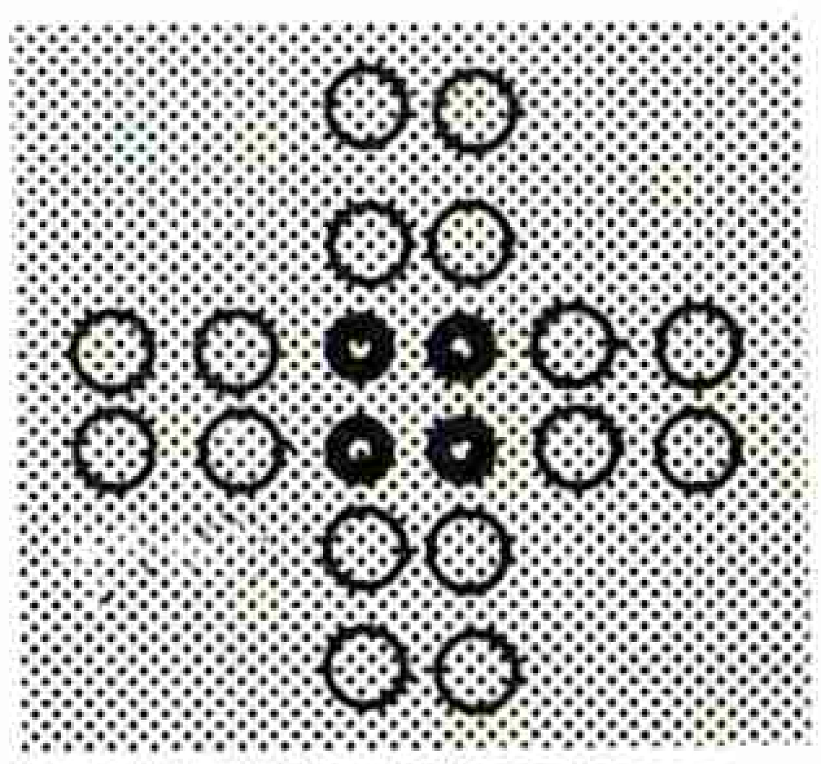


Diag 5.

Firstly look up which values are allocated to each LED then add these values.

Place this into the program and observe the result. You will be correct with the value 3C.

The program for accessing the LEDs in the 8X8 Marix is identical to that for the display on the TEC-1. The only difference is in appearance. A regular array makes the effect more dramatic and the overall possibilities are much greater.

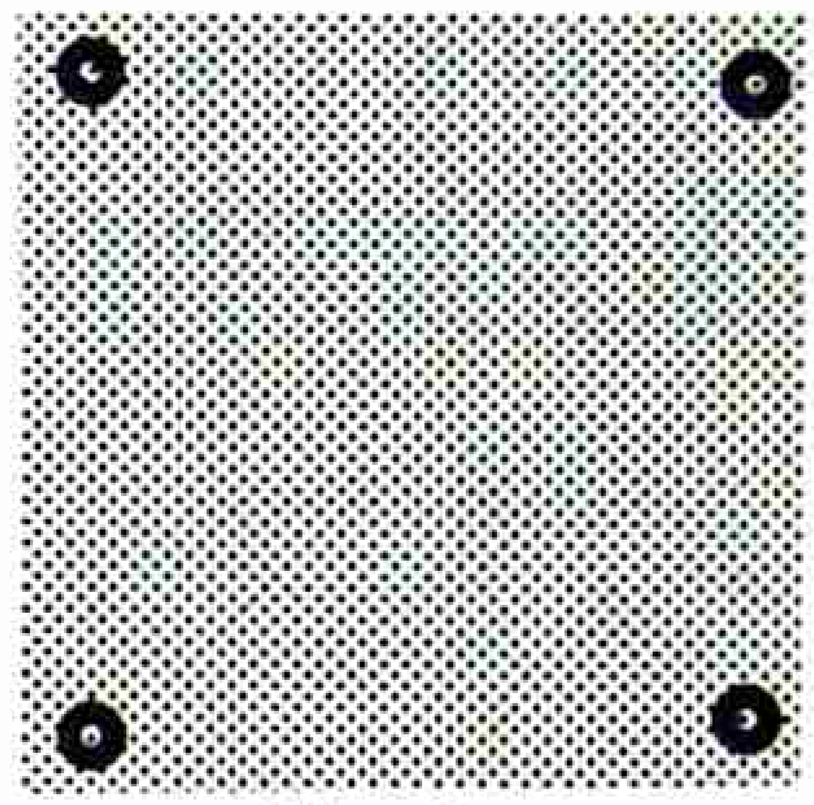


Diag 6.

To turn on the four centre LEDs we must insert the value 08 + 10 into the program for both outputs.

Problem:

What value must be inserted into the program to illuminate the four corner LEDs?

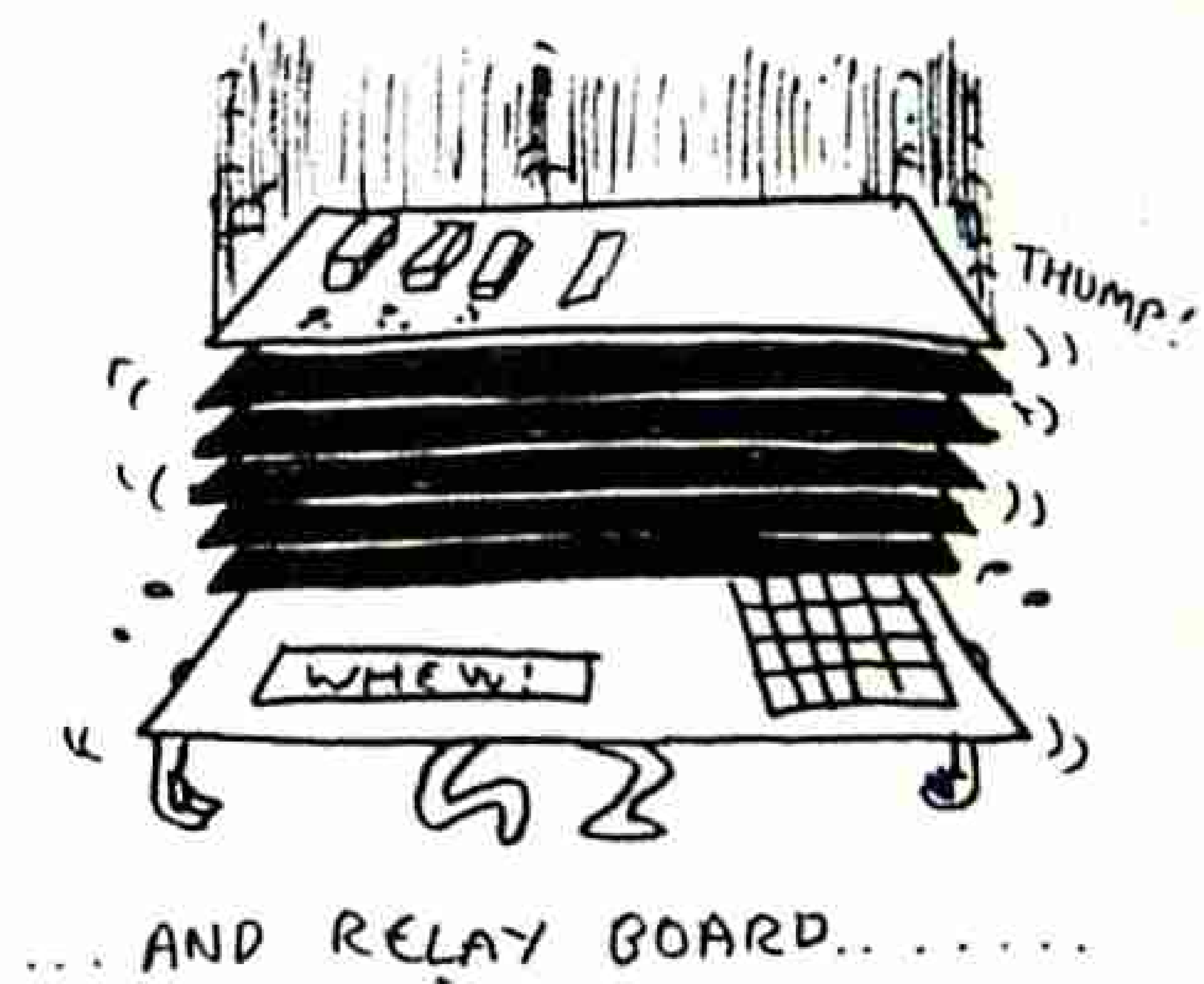


Diag 7.

It is now your turn to illuminate a LED. Select a LED on the matrix and mark it with a pen. Determine its co-ordinates and put them into the program. Execute the program and see it the marked LED comes on. Try two more of these routines and confirm the program by illuminating the LED.

Now illuminate two or three LEDs in any row or column by adding the relevant Hex values together and observe the LEDs on the display.

With this simple program it is not possible to illuminate any combination of LEDs on the whole screen because we are using the outputs in the static mode. To illustrate this, try to illuminate one column and one row at the same time. You know the Hex value for a complete row is FF. Place this into the program and see what happens. The result is a completely-filled screen. The closest effect to producing an intersecting row and column is a non-illuminated row and column produced by inserting a value such as EF into the program.



PROBLEMS:

Demonstrate your understanding of addressing the matrix display by solving the following:

1. Illuminate the whole screen.
2. Illuminate the whole screen except for the outer row and column of LEDs.
3. Illuminate the four centre LEDs as well as the next row and column on each side.
4. Illuminate any quarter of the display.
5. Leave the two centre rows and columns non-illuminated.
6. Place FF in port 3 and 00 in port 4. What appears on the screen? Why?

MAKING A FLASHING LED

We know the general formula for turning on a LED on the matrix:

```
3E (data)  <- Single Byte.
D3 03
3E (data)
D3 04
76
```

"FLASHING LED"

To FLASH the LOWEST priority LED we insert data into the program as follows:

```
LD A,01       800   3E 01
OUT (3),A     802   D3 03
LD A,01       804   3E 01
OUT (4),A     806   D3 04
CALL DELAY    808   CD 00 0A
LD A,00       80B   3E 00
OUT (3),A     80D   D3 03
LD A,00       80F   3E 00
OUT (4),A     811   D3 04
CALL DELAY    813   CD 00 0A
JP 0800       816   C3 00 08
```

DELAY ROUTINE AT 0A00:

```
11 FF 06
1B
7B
B2
C2 03 0A
C9
```

Press **RESET, GO** and the lowest LED will blink ON and OFF. The program is basically loading data into ports 3 and 4 then calling the delay so that the information will be displayed on the screen for a short period of time. The output latches are then loaded with 00 data which will produce a non-illuminated display and the delay routine is called. This produces the 'OFF' period. The program is cycled in an endless loop to produce the flashing.

With this program it is easy to flash any number of LEDs or even the whole screen.

TO BLINK THE WHOLE SCREEN

To blink the whole screen, change the data at addresses 801 and 805 to FF. This has the effect of filling the screen for one delay period and then non-illuminating the screen for one delay period.

To alternately blink the left-hand side of the screen and then the right-hand side:
Insert the following data:

at address:

801 insert FF
805 insert 0F
80C insert FF
810 insert F0

You can make the flash move in the up/down motion by programming:

801 insert 0F
805 insert FF
80C insert F0
810 insert FF

An overlap can be created by inserting the following data:

801 insert 1F
805 insert FF
80C insert F8
810 insert FF

You will notice the two centre rows remain ON for the whole period of time as shown by this table:

TOP	80	
	40	
	20	
	10	
	8	BOTTOM
	4	
	2	
	1	

An interlocking effect can be created by programming the following:

801 insert AA
805 insert FF
80C insert 55
810 insert FF

To make a block of 4 LEDs jump diagonally and back again, the following information is inserted into the program:

change 801 to 0F
change 805 to 0F
change 80C to F0
change 810 to F0

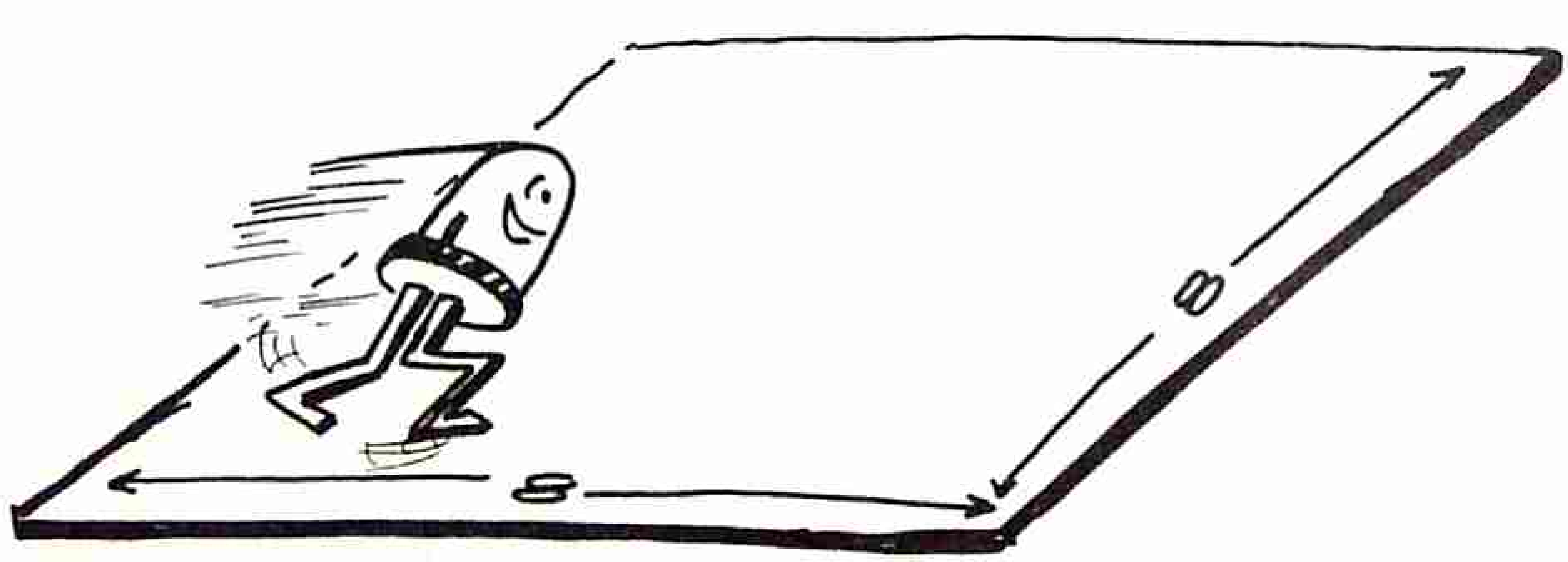
You can experiment with the length of the delay to produce a faster or slower flash rate.

For a slow flash insert: 11 FF 0A
Medium flash: **11 FF 08**
fast flash: **11 FF 06**

TO RUN A SINGLE LED ACROSS THE DISPLAY

This program will run a single LED across the bottom of the display, from left to right and HALT.

```
LD A,01       800   3E 01
OUT (4),A     802   D3 04
LD C,08       804   0E 08
LD A,01       806   3E 01
OUT (3),A     808   D3 03
LD B,A        809   47
CALL DELAY    80A   CD 00 0C
LD A,B        80D   78
RLC A         80E   CB 07
DEC C         810   0D
JP NZ LOOP    812   C2 08 08
HALT          815   76
```



"RUNNING LED ON AN 8X8 MATRIX"

To regulate the speed at which the LED crosses the display, we need a delay routine. (Exactly the same as the previous delay routine.)

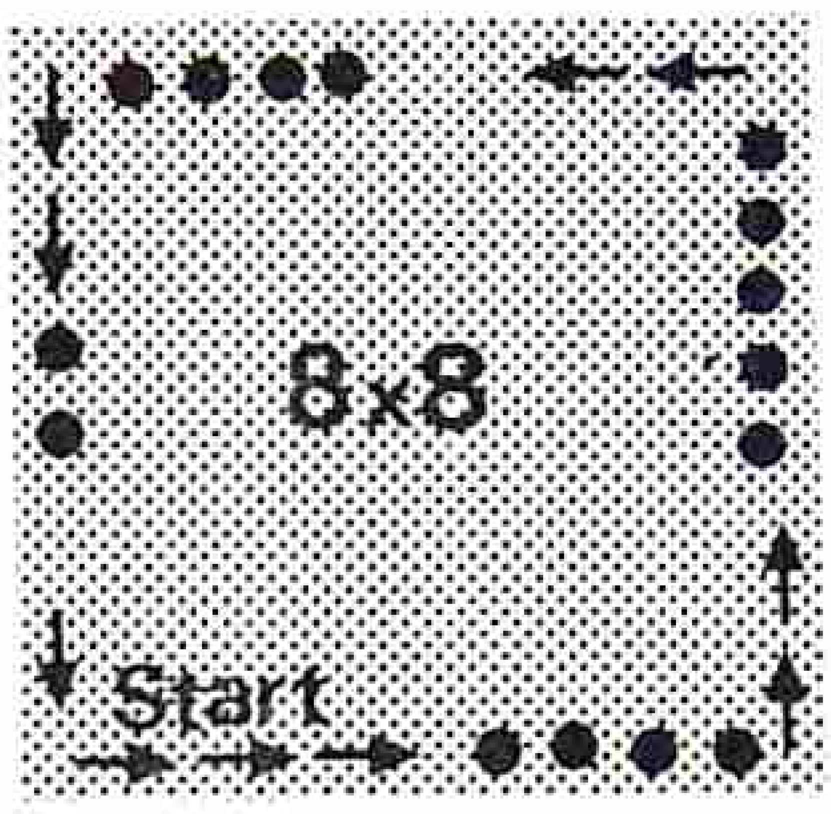
Delay routine at 0C00:

```
11 FF 06
1B
7B
B2
C2 03 0C
C9
```

For a full column to move across the screen, change the data at **801** to **FF.**

To create a REPEAT, change the Halt at **815** to **C3 00 08.**

To make a single LED run around the perimeter of the display, we must create a program for each of the four sides. The program above is suitable for the first side and three more



programs are needed. At location **815** we remove the **HALT** function (or the return function) and add the following:
Press RESET,ADdress 0815, +. *Now continue:*

```
LD A,80        815   3E 80
OUT (4),A      817   D3 04
LD C,07        819   0E 07
LD A,02        81B   3E 02
OUT (3),A      81D   D3 03
LD B,A         81F   47
CALL DELAY     820   CD 00 0C
LD A,B         823   78
RLC A          824   CB 07
DEC C          826   0D
JP NZ LOOP     827   C2 1D 08
HALT           82A   76
```

Press RESET, GO. The LED will travel along 2 sides of the display and Halt.

Program the third side as follows:
Press RESET, ADdress, 082A, +
Add the following:

```
LD A,80        82A   3E 80
OUT (3),A      82C   D3 03
LD C,07        82E   0E 07
LD A,40        830   3E 40
OUT (4),A      832   D3 04
LD B,A         834   47
CALL DELAY     835   CD 00 0C
LAD A,B        838   78
RRC A          839   CB 0F
DEC C          83B   0D
JP NZ LOOP     83C   C2 32 08
HALT           83F   76
```

Press RESET,GO and watch the LED travel the 3 sides of the display. If everything is correct, program the last side as follows:

```
LD A,01        83F   3E 01
OUT (4),A      841   D3 04
LD C,07        843   0E 07
LD A,40        845   3E 40
OUT (3),A      847   D3 03
LD B,A         849   47
CALL DELAY     84A   CD 00 0C
LAD A,B        94D   78
RRC A          84E   CB 0F
DEC C          850   0D
JP NZ LOOP     851   C2 47 08
JP 0800        854   C3 00 08
```

Two adjustments must be made to the first section of the program to eliminate the double exposure on the lowest priority LED. Change location **805** to **07** and **807** to **02.** The led will now travel evenly around the display.

To view the effect, press RESET, GO.

The previous program is long because each direction of travel must include the commencement location.

The next program is just as interesting but much shorter because it generates its own new set of values at the end of each cycle via the **INC H** operation.

It moves a LED across the screen and increases its value on each pass.

```
LD A,01        800   3E 01
LD H,01        802   26 01
LD A,H         804   7C
OUT (3),A      805   D3 03
LD C,08        807   0E 08
LD A,01        809   3E 01
OUT (4),A      80B   D3 04
LD B,A         80D   47
CALL DELAY     80E   CD 00 0C
LD A,B         811   78
RLC A          812   CB 07
DEC C          814   0D
JP NZ LOOP     815   C2 0B 08
INC H          818   24
JP 804         819   C3 04 08
```

At 0C00:

```
11 FF 06
1B
7B
B2
C2 03 0C
C9
```

At the beginning of the previous routine, the first instruction **LD A,01** is not needed as the second and third instruction performs this task. Your requirement is to re-write the whole listing, beginning at 0800, with this instruction removed. This requires the instruction at **819** to be changed to **C3 02 08** as all the instructions have been shifted two locations.

Run the new listing and make sure it works.

Increase the speed of the program by decreasing location **0C02** to **03.**

How can we make it run slower?

Ans: Insert FF into location 0C02 and reduce the CLOCK speed on the computer.

MAKING THE LEDS RUN FROM RIGHT-TO-LEFT

We can add an instruction to this program to make the LEDs run from right-to-left.

The two locations to change are:

change **809** to **3E 80**
change **812** to **CB OF**

Try these variations:

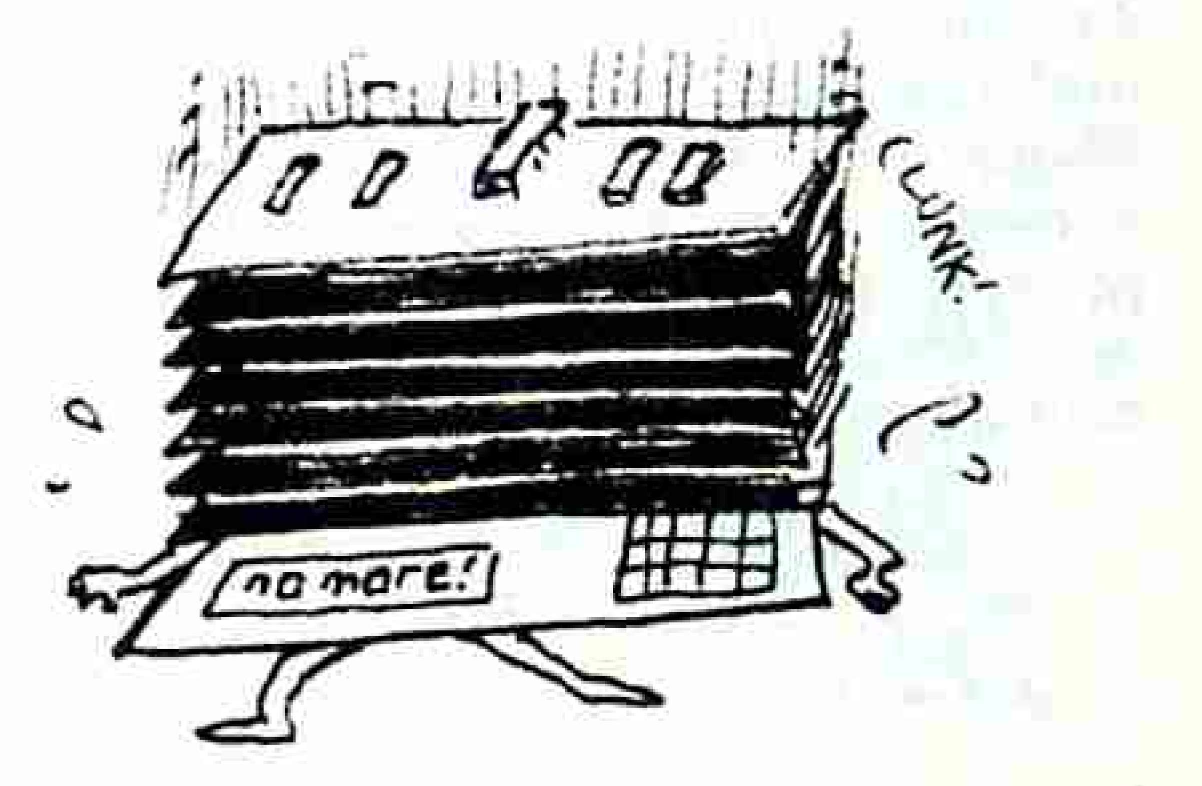
change **802** to **26 FF**
change **818** to **25**

To make the LEDs run from left to right and back again or from top to bottom and down again, requires the combining of a SHIFT-LEFT program with a SHIFT-RIGHT program.

Key in the following listing and push RESET, GO. Watch the effect.

Don't forget the delay routine at 0C00.

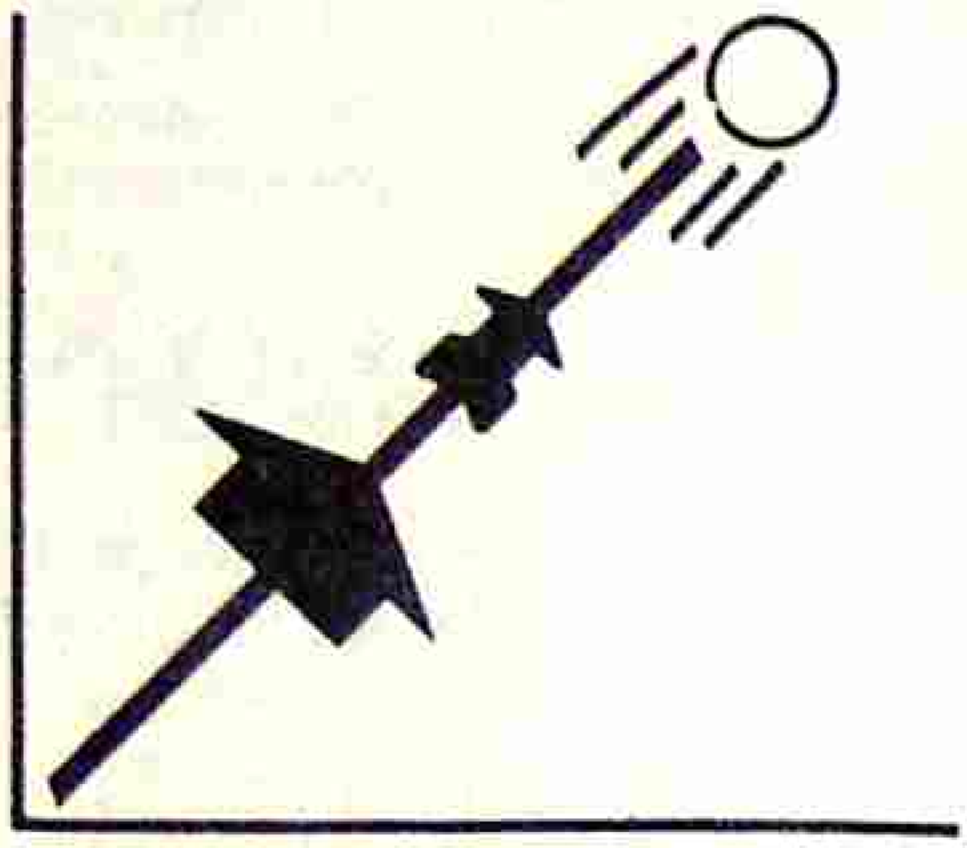
```
LD H,01        800   26 01
LD A,H         802   7C
OUT (3),A      803   D3 03
LD C,08        805   0E 08
LD A,01        807   3E 01
OUT (4),A      809   D3 04
LD B,A         80B   47
CALL DELAY     80C   CD 00 0C
LD A,B         80F   78
RLC A          810   CB 07
DEC C          812   0D
JP NZ LOOP     813   C2 09 08
LD C,08        816   0E 08
LD A,80        818   3E 80
OUT(4),A       81A   D3 04
LD B,A         81C   47
CALL DELAY     81D   CD 00 0C
LD A,B         820   78
RRC A          821   CB 0F
DEC C          823   0D
JP NZ LOOP     824   C2 1A 08
INC H          827   24
JP 0802        828   C3 02 08
```



.... AND CLOCK TIMER BOARD..

"TAKE-OFF!"

This program produces a single LED which runs diagonally across the display. The angle at which the LED



moves is the result of increasing the value of both outputs AT THE SAME TIME. This can lead to some interesting effects.

At 800:

```
LD A,01        800    3E 01
OUT (3),A      802    D3 03
OUT (4),A      804    D3 04
RRA            806    17
PUSH AF        807    F5
CALL DELAY     808    CD 00 09
POP AF         80B    F1
JP 802         80C    C3 02 08
```

At 900:

```
11 OF 00
1B
7A
B3
C2 03 09
C9
```

At address **806** the instruction **17** will cause the LED to travel up the screen. If we insert the instruction **1F** the LED will travel down the screen.

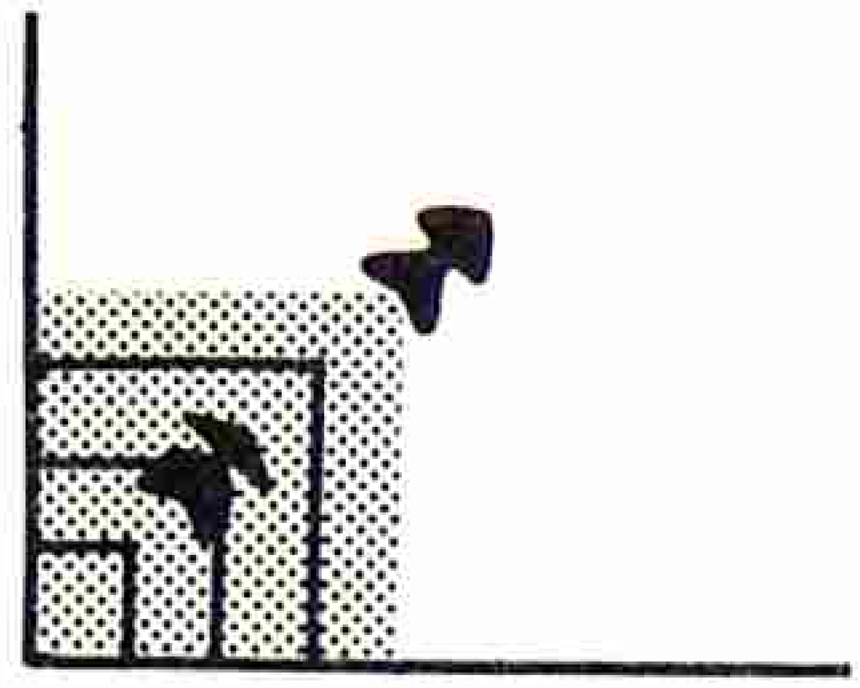
At location **801** insert the value 90. Try both directions of travel and watch the different effects.

Both ROTATE instructions **17 & 1F** cause the 'bits' in the accumulator to rotate through the 'carry' and this creates a 'hole' or zero in the output. This forms the non-illuminated band which passes across the screen.

At location 801, the value 01 can be replaced by 02, 4, 8, 10, 20, 40 or 80. These will not alter the effect on the screen as they will merely define the starting point for the program and it will run through its cycle in the normal manner.

"FAN - OUT"

This program is almost identical to the previous. But by adding one new instruction, we can change the effect



on the display to produce a completely different effect.

```
LD A,01        3E 01
OUT (3),A      D3 03
OUT (4),A      D3 04
RLA            07
PUSH AF        F5
CALL DELAY     CD 00 09
POPAF          F1
INC A          3C
JP 802         C3 02 08
```

Delay at 900:

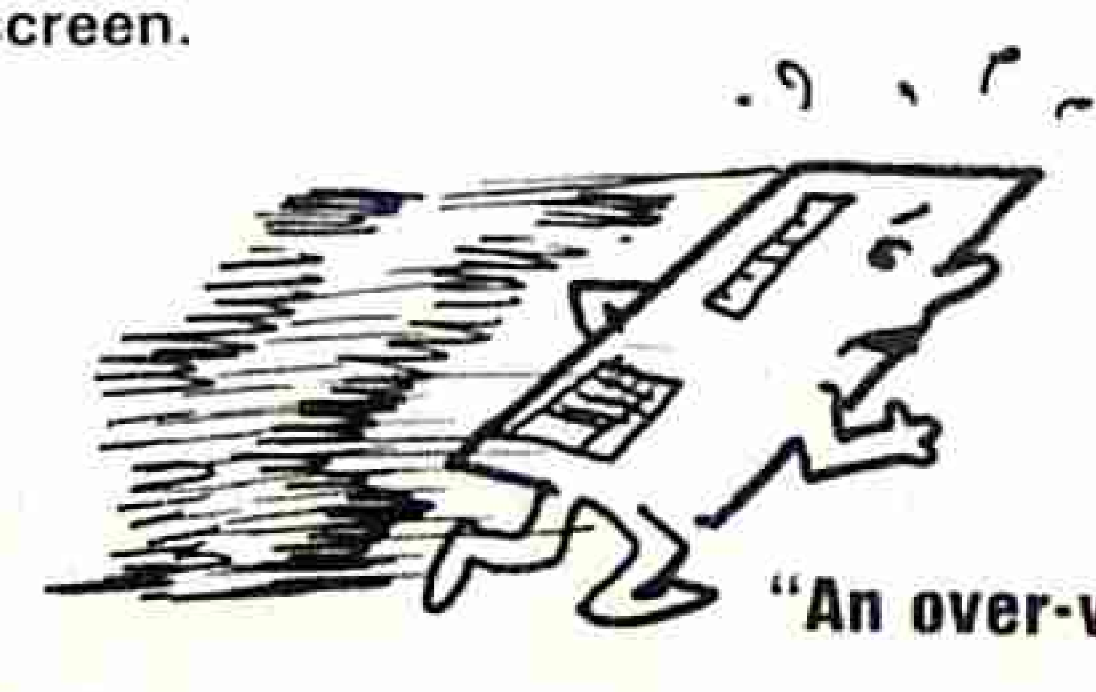
```
11 FF 06
1B
7A
B2
C2 03 09
C9
```

The new instruction is **INC A**. It makes the least significant bit HIGH. The result is to produce an increasing row of LEDs. This is how it happens:

Initially a HIGH is programmed as the Least Significant Bit. The operation **RLA** transfers this HIGH to the second location. When **INC A** is executed, a HIGH is placed in the lowest position. This gives two HIGHs in the register. These two HIGHs shift up the register when **RLA** is executed. **INCA** produces another HIGH in the lowest position and thus the whole register is gradually filled.

The program is producing its own NEW set of data each time the listing is cycled.

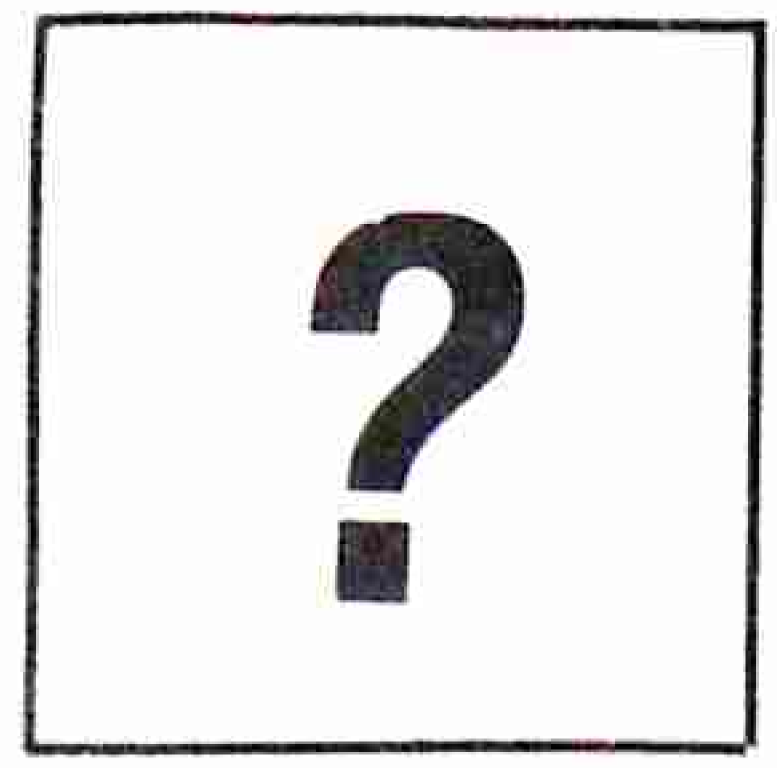
The final result is most impressive. The display fans out from the lower left-hand corner to fill the entire screen.



"An over-worked TEC-1"

OUR MYSTERY EFFECT

I call this our mystery effect as I have forgotten how it appears on the screen. All I remember is producing



it. It took about an hour or so to get the program together and I will leave it for you to type into the TEC-1 and see what appears.

Here is the listing:

```
LD C,40        800    OE 40
LD A,01        802    3E 01
OUT (3),A      804    D3 03
LD A,01        806    3E 01
OUT (4),A      808    D3 04
RLCA           80A    07
CALL 900       80B    CD 00 09
DEC C          80E    OD
JP NZ 808      80F    C2 08 08
LD C,20        812    OE 20
LD A,01        814    3E 01
OUT (4),A      816    D3 04
LD A,01        818    3E 01
OUT (3),A      81A    D3 03
RLCA           81C    07
Call 900       81D    CD 00 09
DEC C          820    OD
JP NZ 824      821    C2 1A 08
LD C,40        824    OE 40
LD A,01        826    3E 01
OUT (3),A      828    D3 03
OUT (4),A      82A    D3 04
RLCA           82C    07
CALL 900       82D    CD 00 09
DEC C          830    OD
JP NZ 832      831    C2 28 08
JP 0800        834    C3 00 08
```

At 0900:

```
900    F5
901    CD 00 0C
904    F1
905    3C
906    CB 47
908    CA OC 09
90B    C9
90C    CD 00 0C
90F    CD 00 0C
912    C3 00 09
```

Delay at 0C00:

```
11 FF 06
1B
7A
B2
C2 03 0C
C9
```

USING THE KEYBOARD

The next area of learning is to include a keyboard input for the 8x8 matrix.

Whenever the HALT function is placed in a program, the Z80 stops the program and waits for an input via the interrupt line.

In our case, this comes from the keyboard and the non-maskable interrupt line is activated to allow the Z80 to accept the data from the keyboard encoder via the data bus.

This data is loaded into the accumulator and compared with a value in the program. If the two values are the same, the output is zero and the program advances.

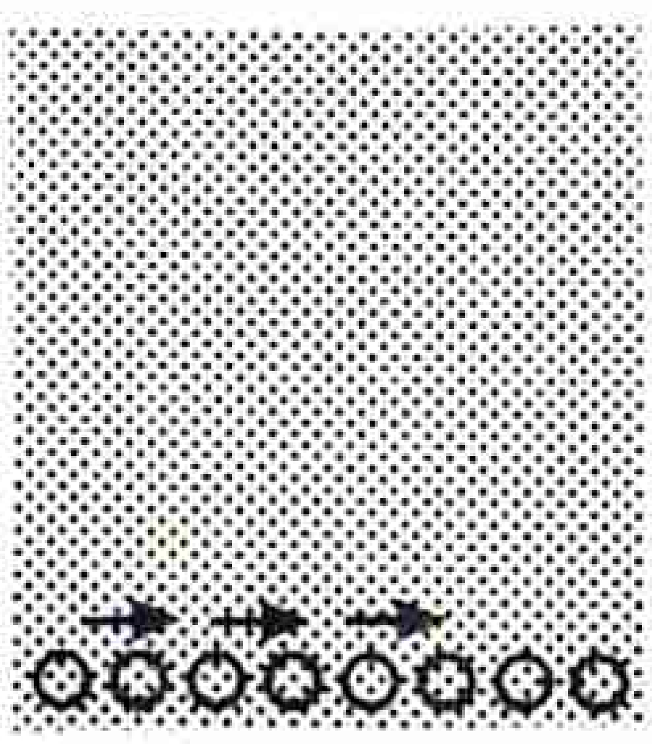
This is the basis of the next set of programs. The correct key must be pressed for the program to be executed. Otherwise the program will return to the HALT instruction and the outputs will not change.

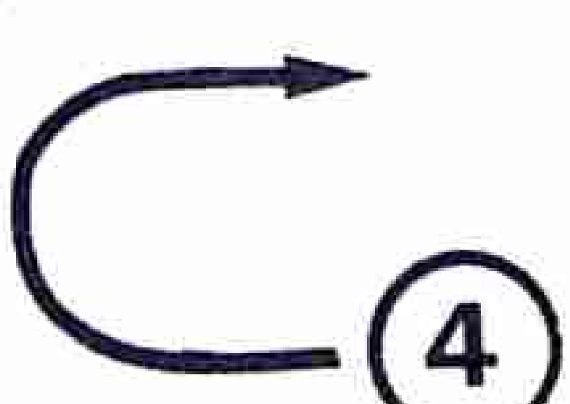
MOVING A LED VIA KEY '4'.

This program moves a LED across the bottom row. It advances one position each time the '4' key is pressed.

No delay routine is employed and the LED will shift at a speed determined by pressing the key.

When the LED reaches one side of the display it re-appears at the opposite side. This can be a distinct advantage when playing some of the games we have devised. At the moment the shift in this program is only left-to-right.





LD A,01	**800**	**3E 01**
OUT (4),A	**802**	**D3 04**
LD B,A	**804**	**47**
LD A,B	**805**	**78**
OUT (3),A	**806**	**D3 03**
HALT	**808**	**76**
LD A,I	**809**	**ED 57**
CP 04	**80B**	**FE 04**
JP NZ 808	**80D**	**C2 08 08**
RLC B	**810**	**CB 00**
JP 805	**812**	**C3 05 08**

Accumulator A is loaded with 01 and passed to segment port 4 where it is latched. The contents of A are loaded into register B so that it can be operated upon by the **ROTATE LEFT CIRCULAR** function and also be in a "safe" register, so it is not written over.

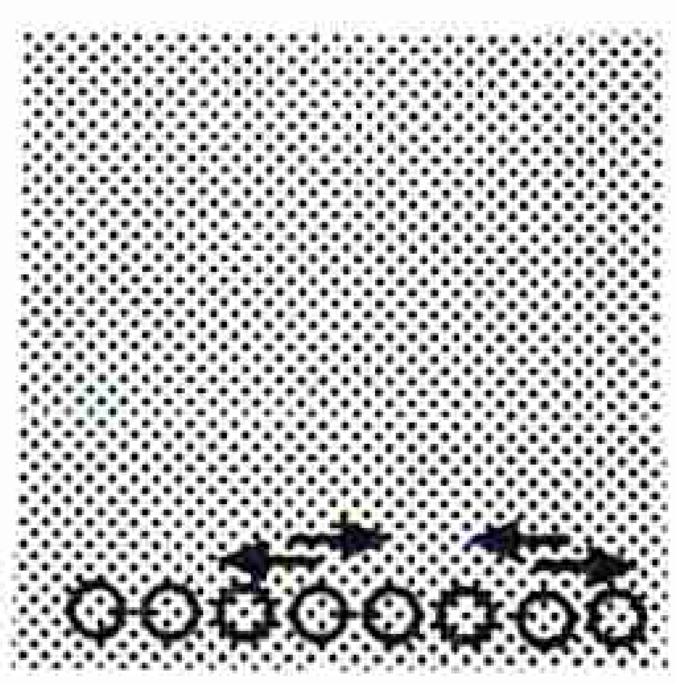
The program is HALTED at 808 and the Z80 waits for a keyboard instruction. When a key is pressed, the NMI line is activated and the data is sent to the Z80 and initializes the INTERRUPT VECTOR REGISTER 'I'. The keyboard data is placed in the accumulator register and compared with the value 04. If the answer is ZERO, the program is incremented to address 810, which instructs the Z80 to ROTATE REGISTER B LEFT. This causes the HIGH bit to shift from bit 0 position to bit 1 position and this will make the LED shift one place to the right on the display when operations at 81C, 81D, 81E, 805, 806, 807 and 808 have been performed.

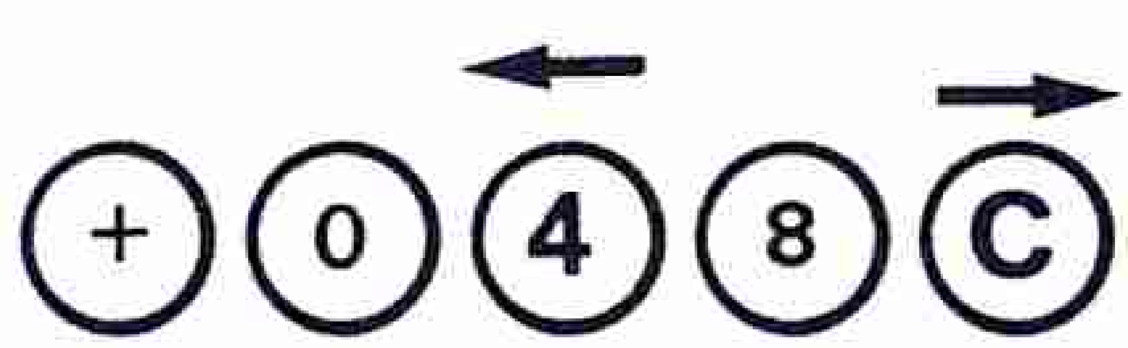
The new data-value in register B is loaded into register A at 805 and is passed to the display latch port 3 at 806 and 807.

The important feature of this program is the use of the interrupt vector register I to detect the input from the keyboard and to enable a compare function to be performed.

SHIFTING A LED ⟷

This program expands on the previous and adds a shift in the opposite direction. We now have a forward and reverse shift.





Key '4' shifts left and 'C' shifts the LED to the right.

The direction of shift is governed by **RLC B** and **RRC B** and these can be swapped to give the opposite effect.

If you require the LED to travel up and down the screen, the output ports 3 and 4 must be reversed in the program.

LD A,01	**800**	**3E 01**
OUT (4),A	**802**	**D3 04**
LD B,A	**804**	**47**
LD A,B	**805**	**78**
OUT (3),A	**806**	**D3 03**
HALT	**808**	**76**
LD A,I	**809**	**ED 57**
CP 04	**80B**	**FE 04**
JP NZ 815	**80D**	**C2 15 08**
RLC B	**810**	**CB 00**
JP 805	**812**	**C3 05 08**
CP 0C	**815**	**FE 0C**
JP NZ 808	**817**	**C2 08 08**
RRC B	**81A**	**CB 08**
JP 805	**81C**	**C3 05 08**

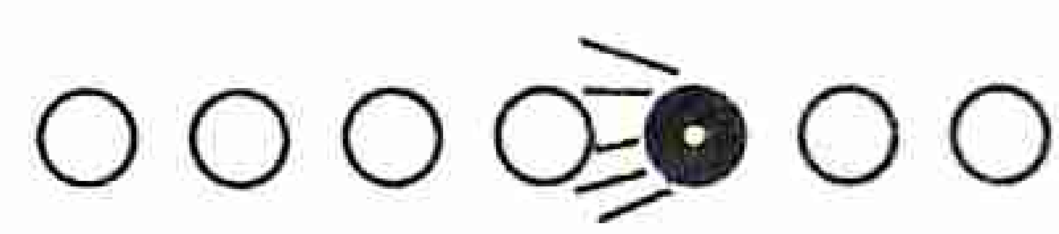
This, and many other features can be altered to suit your own requirements. It is a matter of experimenting and determining which instruction should be altered. If you discover these changes yourself, you will have a much greater understanding of how the program is put together.

The values at 80C and 816 determine which buttons are operative. These can be changed to any pair you choose, simply by inserting the correct data into the program.

The data corresponds to the value which appears on the key, for 0 to F. Keys +, −, GO and AD have the values 10, 11, 12, and 13.

ADDING AUTO REPEAT

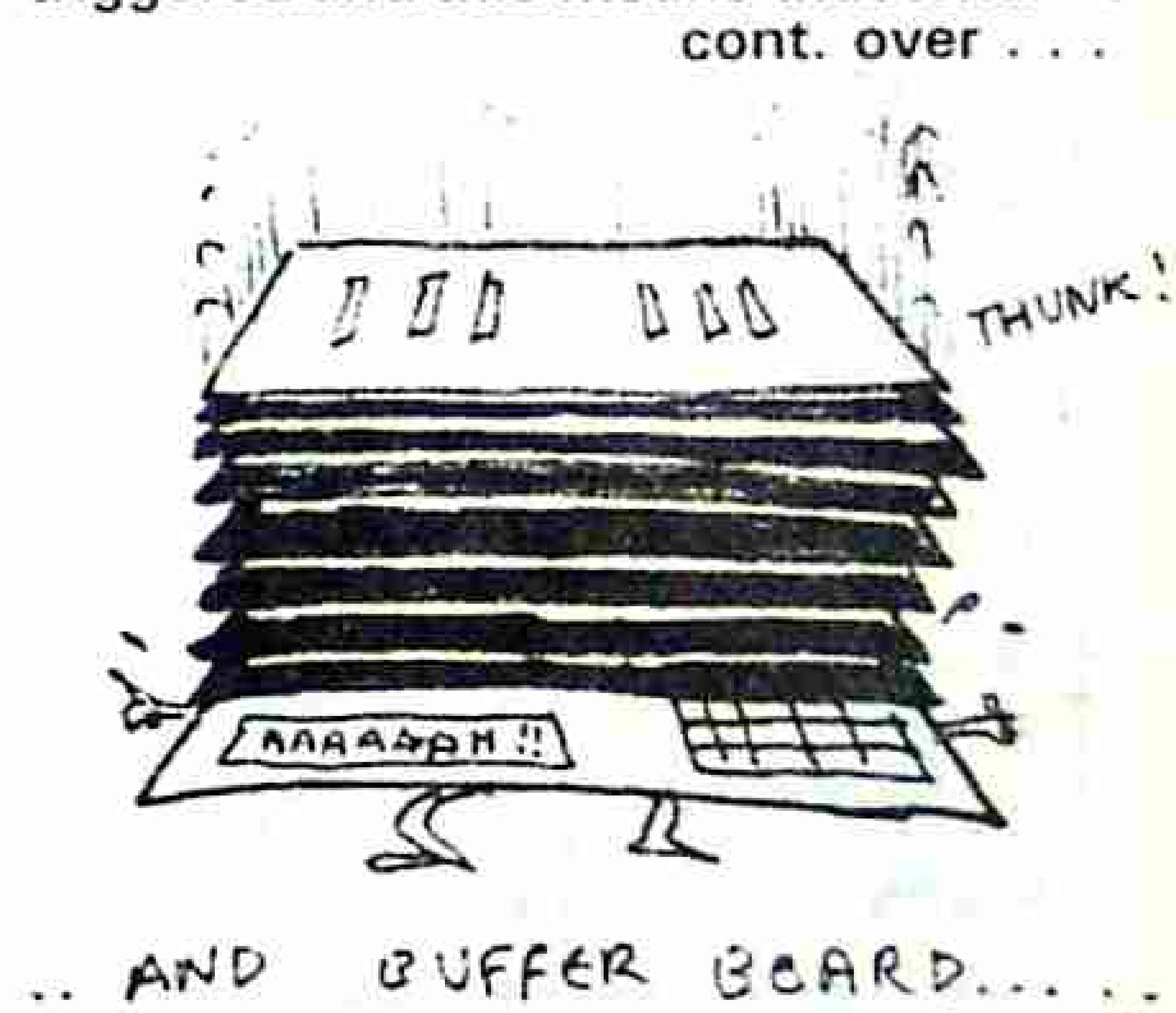
A simple addition to the previous program will enable the LED to run across the display in an auto repeat mode, when the correct key is pressed.



This repeat operation is not capable of detecting when the key has been released as the keyboard encoder contains a latch which retains the last value outputted from the key pad.

The NMI line operates a flip flop inside the Z80 which is edge triggered and this means that when it

cont. over . . .



cont. next issue!!

is reset, after dealing with the value from the keyboard encoder, it cannot be set again without physically pressing the key AGAIN.

Thus a key pressed for a long time can only be recorded ONCE.

The following program will detect key 4 and run the LED across the screen via a loop in the program and continue to do so until another key is pressed. This is the only way of halting the run.

```
LD A,01         800    3E 01
OUT (4),A       802    D3 04
LD A,01         804    3E 01
OUT (3),A       806    D3 03
LD B,01         808    06 01
HALT            80A    76
LD A,I          80B    ED 57
CP 04           80D    FE 04
JP NZ HALT      80F    C2 0A 08
RLC B           812    CB 00
LD A,B          814    78
OUT (3),A       816    D3 03
CALL DELAY      817    CD 00 0C
JP 80B          81A    C3 0B 08
```

At 0C00:

```
11 FF 0A
1B
7B
B2
C2 03 0C
C9
```

Press RESET, GO.
Press **Key 4** to shift LED.
Press any other key to **HALT** LED.

AUTO RETURN AND STOP

The following program detects 3 keys. The + key shifts the LED left, the 'O' key stops the LED and key '4' shifts it right.

The speed of travel across the display is controlled by the length of time of the DELAY ROUTINE.

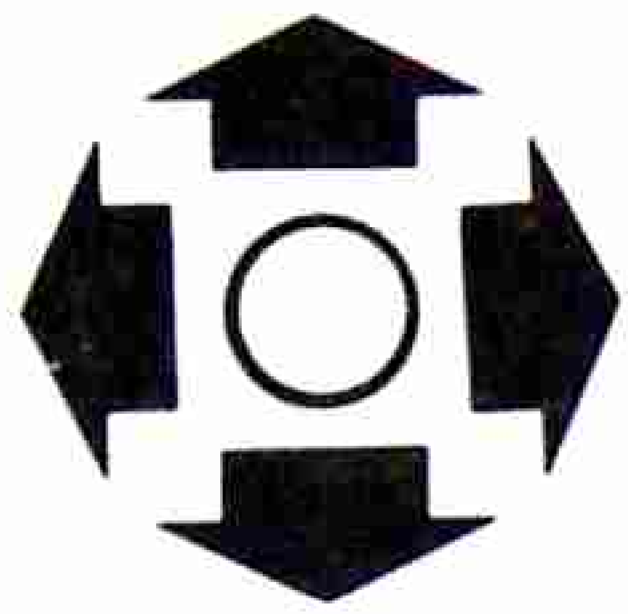
```
LD A,01         800    3E 01
OUT (4),A       802    D3 04
LD A,01         804    3E 01
OUT (3),A       806    D3 03
LD B,01         808    06 01
HALT            80A    76
LD A,I          80B    ED 57
CP 04           80D    FE 04
JP NZ 81A       80F    C2 1A 08
RLC B           812    CB 00
LD A,B          814    78
OUT (3),A       815    D3 03
CALL DELAY      817    CD 00 0C
CP 10           81A    FE 10
JP NZ HALT      81C    C2 0A 08
RRC B           81F    CB 08
JP 80B          821    C3 0B 08
```

At 0C00:

```
11 FF 0A
1B
7B
B2
C2 03 0C
C9
```

4-DIRECTION SHIFT

This program is an extension to the previous listing to obtain a 4-direction shift.



The four buttons we have chosen for controlling the LED are: –, 5, 2 and 0. There is no auto repeat feature in this listing and the LED can be moved around the entire display by using the keys mentioned.

```
LD A,01         800    3E 01
OUT (3),A       802    D3 03
LD B,A          804    47
LD A,01         805    3E 01
OUT (4),A       807    D3 04
LD C,A          809    4F
HALT            80A    76
LD A,I          80B    ED 57
CP 11           80D    FE 11
JP NZ 81A       80F    C2 1A 08
RRC B           812    CB 08
LD A,B          814    78
OUT (3),A       815    D3 03
JP 80A          817    C3 0A 08
CP 05           81A    FE 05
JP NZ 827       81C    C2 27 08
RLC B           81F    CB 00
LD A,B          821    78
OUT (3),A       822    D3 03
JP 80A          824    C3 0A 08
CP 02           827    FE 02
JP NZ 834       829    C2 34 08
RRC C           82C    CB 01
LD A,C          82E    79
OUT (4),A       82F    D3 04
JP 80A          831    C3 0A 08
CP 00           834    FE 00
JP NZ 80A       836    C2 0A 08
RLC C           839    CB 09
LD A,C          83B    79
OUT (4),A       83C    D3 04
JP 80A          83E    C3 0A 08
```

This program is the basis of a game we will be presenting in the next issue. Basically it is a **HUNT THE FOX** game in which a secret co-ordinate is selected and the object of the game is to locate the fox in the MINIMUM NUMBER OF MOVES. The LED is the pack of hounds and when they coincide with the fox, the screen will flash a victory or produce a hunting tune.

The completion of the game is up to you. Try your hand at writing a game along these lines and send it in for publishing in the next issue.

In the little space left I would like to include a program from one of our readers.

Inspired by the content of issue 9, a TEC-1 and a Z80 Machine Code book, he has written a sound effects program which will really amaze you. It is a complex sound generator which is fully programmable and it is only when you start to change some of the data bytes, that you will see how it goes together.

ALIENS ATTACK RUN

-by M J Allison, 3095

```
LD HL,0903      800    21 03 09
LD A,01         803    3E 01
LD HL,A         805    77
LD C,30         806    06 30
CALL 0903       808    CD 0E 09
INC (HL)        80B    34
DJNZ CALL       80C    10 FA
JP 0800         80E    C3 00 08

PUSH AF         900    F5
PUSH DE         901    D5
LD DE, 0020     902    11 20 00
DEC DE          905    1B
LD A,D          906    7A
OR A,E          907    B3
JP NZ 905       908    C2 05 09
POP DE          90B    D1
POP AF          90C    F1
RETURN          90D    C9
PUSH AF         90E    F5
PUSH BC         90F    C5
LD BC,00AA      910    01 AA 00
LD A,80         913    3E 80
OUT (1)A        915    D3 01
LD A,00         917    3E 00
OUT (1),A       919    D3 01
CALL 0900       91B    CD 00 09
DEC BC          91E    0B
LD A,B          91F    78
OR A,C          920    B1
JP NZ 913       921    C2 13 09
POP BC          924    C1
POP AF          925    F1
RETURN          926    C9
```

I have run out of room for this issue and still have lots more programs and ideas. Next issue will contain another 20 pages of programming and include 2 more programs from Mr Allison.

Turn to P.50 for 6 pages on the RELAY DRIVER BOARD project and type in the program for operating the relays.

The projects for next issue . . . . .I'll keep them a secret, but you'll be very pleased; I assure you.

KITS FOR ISSUE 11.

( ) Black Jack		7.20
( ) " "	PC Board	2.10
( ) Dual PSU (Ken's)		6.90
( ) " "	PC Board	3.30
( ) FM Bug		4.40
( ) " "	PC Board	1.60
( ) 8x8 Display		14.60
( ) " "	PC Board	5.50
( ) Relay Driver Board		16.80
( ) " "	PC Board	8.80

NZ READERS!

We have been informed that a NEW firm has taken over the supply of TE kits and magazines in New Zealand.
Prices will appear in our next issue and in the meantime you can get a price list by writing to:
Clan Electronics Ltd,
P.O. Box 254,
Rangiora,
N. Canterbury, 8254,
New Zealand.

MODIFICATIONS AND CORRECTIONS

"5 FEED-THROUGHS"

The LOGIC PROBE was our first introduction to double sided PC boards. And it seems we confused some of our readers. We had 8 boards returned because "some of the holes were not drilled".

It seems we forgot to mention the fact that the components are mounted on the top side of the board and this means some lands do not need holes. Another point we failed to mention was the need to solder both sides of some components and also the need for 5 feed-through links to join the top side of the board with the bottom. If your unit does not work, we suggest you look over the board. The layout below shows the position of the feed-throughs:

THE 1MFD DELAY ELECTRO:

The 1mfd electrolytic in the delay circuit is around the wrong way on the schematic, the overlay diagram on P6 of issue 10 and is possibly wrong in YOUR model. Check it now. The positive of the electro should go to the output (pin 12) of the 4049 inverter IC. The negative should go to the 470k delay resistor. Please amend all diagrams.

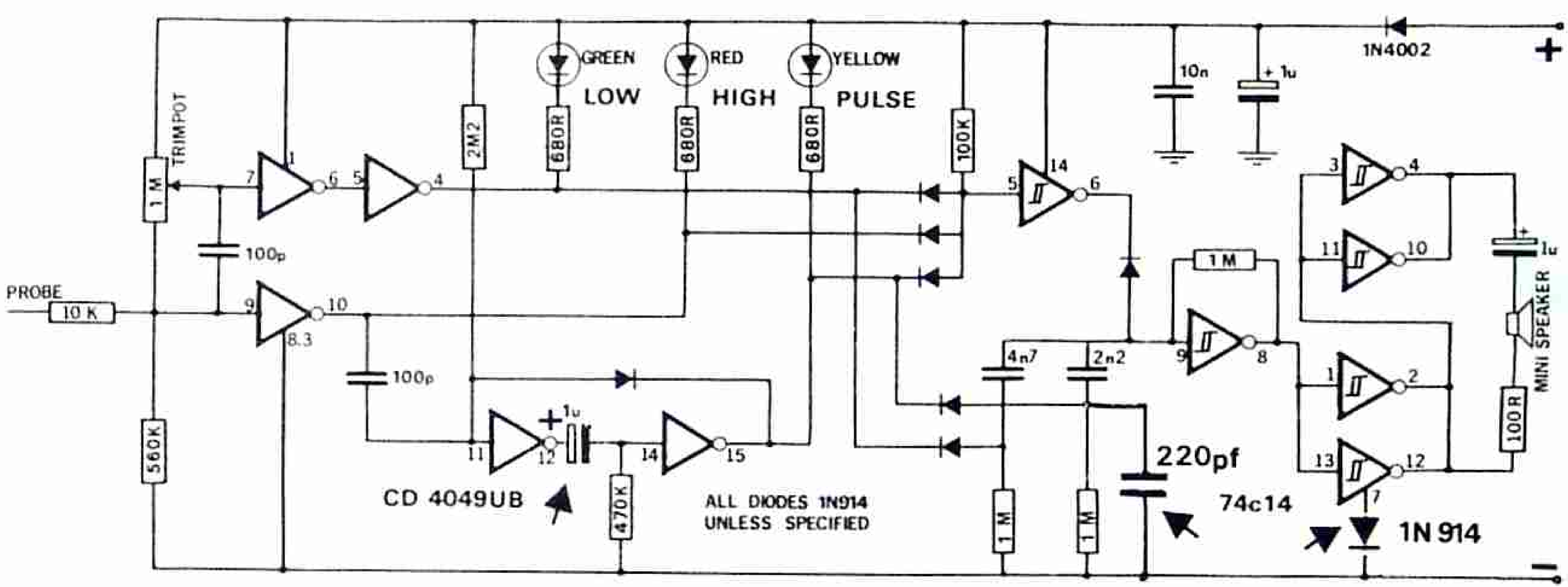
MODS:

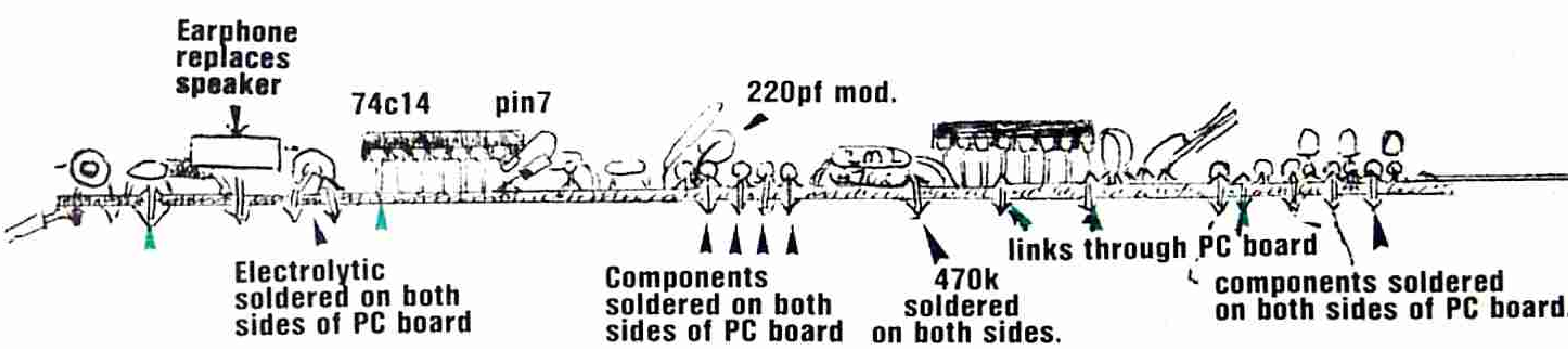
If the tone fails to be emitted from your Logic Probe, the fault will lie in the earthy ends of the 4n7 and 2n2 capacitors not going LOW enough to trigger the Schmitt Trigger.

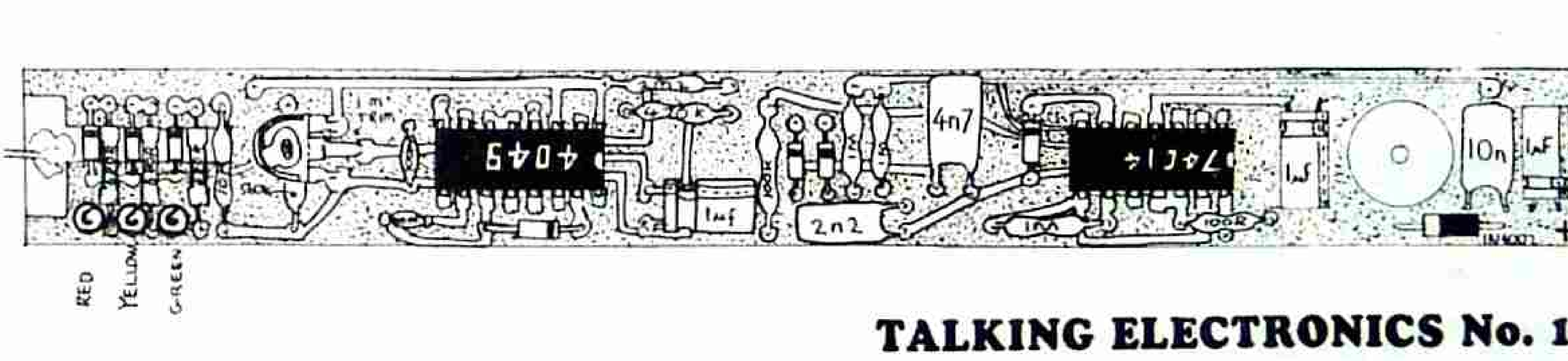
To overcome this we suggest you add a diode between pin 7 of the 74c14 (HCF 40106 or CD 40014) and earth with cathode facing earth as shown in the diagram below.

A 220pf (100pf to 330pf) capacitor can also be added across either of the 1M resistors which are in series with the 4n7 and 2n2 capacitors.

You can also try varying the frequency of the HIGH and LOW tones by altering the value of the 4n7 and 2n2 capacitors.







SHOP TALK

MORE DETAILS ON:
ELECTRONICS Stage-1.

With each new publication we bring out, the digital arena gets a little more coverage.

Eventually our goal is to cover all the basics of electronics but that is a long way off yet. **ELECTRONICS Stage-1** has helped a little in this regard and has found favour from those just starting out.

Some electronics lecturers have mentioned it to students and this has helped them, and us, considerably.

Along with the Stage-1 publication comes a complete package of components to build all the models described in the book. This package is advertised a little further through this issue.

If you are interested in learning via the PRACTICAL, construction, approach, this book will involve you in many hours of construction-work.

And you can really only learn by CONSTRUCTION. All the theory in the world will come unstuck when the prototype fails to operate.

That's where the Stage-1 book comes on its own. It is PRACTICAL.

The need for a text book came to a head about 12 months ago. At that time TE was sent a number of text books for review. At first we thought they must be fairly good as they were already in use in a number of technical schools.

On closer examination, we came to the frightening conclusion that they were far from ideal. In fact two of them were little more than useless.

Granted, they all contained circuit diagrams and construction projects, but the circuit diagrams in two books were laid out according to the plan-view of the chip so that each pin number was shown in a rotational direction. This may be ideal for assembly but as a presentation of a circuit diagram, it provides little indication of how a circuit works. A circuit diagram should be instantly understandable, like a photo, and the student should be required to refer to it during the course of construction.

To prove my point, I have an example of construction - without - understanding:

A teacher visited our premises recently to buy a range of kits for his electronics group. He had taken over one of the electronics classes part-way through the term and was reviewing one of the projects constructed by the previous teacher. He mentioned to one of his pupils that the resistors must touch the PC board. The student thought for a moment and replied: "Which are the resistors? Sir."

It seems the construction was demonstrated in a very clear manner at the beginning of the lesson and the students were required to copy the teacher, ape fashion.

And believe me, this can happen. A very clear demonstration can be performed without the students ever knowing what is going on.

But we hope to change all this.

Our new book is intended to fill the gap. The whole concept of the book is different. It approaches electronics in a way which makes sense to the student.

It follows a hypothetical case in which a beginner wishes to add a LIGHT EMITTING DIODE to an amplifier, to show when it is on.

He very soon finds out that the LED needs a dropper resistor, must be connected around the correct way and the value of the resistor needs to be worked out.

This is the same in any practical field. One thing leads to another. But there is one advantage in learning electronics. It is a field which is growing and people with qualifications are in big demand.

Obviously a book of 100 pages could not cover everything. We have aimed at introducing as many building blocks as possible and each chapter concentrates on one component or topic and these are introduced when they are needed. This coincides with a real-life situation.

By careful presentation, complex mathematics and theory have been kept to a minimum.

Even the introduction of resistors in Parallel and Series has been covered with only the use of simple mental arithmetic. After all, it is amazing the number of values of resistance which can be created with 2, 3 or 4 equal-value resistors in series or parallel.

Construction projects have been introduced as early as possible and they create basic pieces of test equipment for use in later digital circuits.

All the components for the experiments and projects are available from the supplier listed in the front of the book. This enables the course to be commenced immediately by a student at school or at home, knowing that every circuit can reproduced. Any kit or kits can be bought separately if required.

Here is an outline of the chapters:

THE LED Determining the cathode lead. Voltage drop across a red LED and a green LED. The flashing LED.

THE RESISTOR Colour Code. The meaning of tolerance. Resistors in Parallel and Series. Identifying 1% Tolerance Resistors.

THE DIODE Forward and Reverse conduction. Measuring the 'resistance' of a diode with a multimeter. The diode as a rectifier.

THE MULTIMETER Measuring Voltage, Resistance and Current. The meaning of Sensitivity. Presenting the GOLDEN RULE for using a multimeter.

THE TRANSISTOR (The transistor is described in three of the chapters). Biasing an NPN transistor. Cut-off, Partial Conduction and Saturation.

CONTINUITY TESTER This project uses the components studied in the first 5 chapters. A very simple project which measures resistance up to 10Meg ohms.

THE CAPACITOR The Greencap and Electrolytic. How a capacitor works. Identifying Greencaps marked in nanofarad. Capacitors in Parallel and Series. Tantalum Capacitors. Uses for a capacitor.

SOLDERING How to solder correctly.

DRAWING CIRCUIT DIAGRAMS A large digital circuit is required to be copied onto a grid page.

AUDIBLE LOGIC PROBE A 3-Transistor project designed for use in testing digital circuits. How the circuit works.

THE DRY CELL A coverage of basic electricity as presented in a booklet by Eveready Batteries. Cells in Parallel and Series. Ohm's Law. An experiment to compare the capacitance of a set of cheap AA cells and an expensive set. The Nicad Cell.

THE TRANSISTOR The transistor as a SWITCH. The transistor in HIGH-LOW mode so that heat dissipation is minimal. The Constant Current Generator.

ZN414 IC, POCKET RADIO. A project using a ZN 414 radio IC. Selectivity and Sensitivity.

THE TRANSISTOR as a multivibrator. The Monostable, Bistable and Astable. How they function. Construction projects.

THE 555 IC A full coverage of the most versatile timer chip to come on the market.

THE 4017 DECADE COUNTER. Running Light Effect. Freezing. Resetting and Clocking the counter.

DOOR CHIME. A project using an Astable Multivibrator, the 4017 Decade Counter, the transistor as a SWITCH and the 555 as an oscillator.

LM 380 4-WATT AMPLIFIER. A power amplifier in a single chip which is heat-sinked by the printed circuit board. Can be duplicated to make a stereo amplifier.

LOGIC CIRCUITS. Covering the **AND, OR,** and **BUFFER** gates and their inverse functions: **NAND, NOR** and **INVERTER.** Combining two gates together.

THE SCHMITT TRIGGER. How the Schmitt Gate works. Its use as an oscillator and as a delay.

THE CLOCK A look at clock circuits and how to gate them ON and OFF.

UP/DOWN COUNTER A brief look at a driver chip, a 7-segment display and a Decade Counter which is coded in binary. (A BCD Counter).

The text book concludes with a project which uses many of the BUILDING BLOCKS covered in the chapters.

The best advertisement for any of our publications is YOU. By showing the book to your teacher or friend, you have done yourself a favour. Not only do our sales remain high, but you ensure that we will be around to bring you additional interesting publications.

This text book is only stage-1. We have already thought about stage-2 and are accepting advance orders. The book will be sent to you as soon as it is printed. The cost will be $3.30 plus 80¢ postage.

If you have any ideas or suggestions, send them to Talking Electronics and your name may appear in the next title.

If you have not yet seen **STAGE-1**, you can order your copy directly from TE. See the order-form pages for details.

Our most recent release is: **DIGITAL ELECTRONICS REVEALED.** This is basically a compilation of the 10 MINUTE DIGITAL COURSE and includes pages which will be appearing in the next three issues.

The new sections of the book deal with the flip flop family. It covers the 4modes of operating a J-K flip flop and the effect of PRESET and CLEAR.

The lengthy descriptions and clear diagrams, in laboratory-note format, makes learning a pleasure and improves your retention enormously.

After reading the book, you will say "It's so easy, why didn't I know that?"

All our publications are worth adding to your library and, as you know, books go off the market very quickly. Keep up with the new releases and you will be ahead of everyone else.

THE DRY JOINT

Years of practical experience in locating dry joints in TV sets paid dividends this month. We had to locate two dry joints in TEC-1's. Here's a little story on how we do it.

As consumer products become more complex, the only feature which dogs the electronics industry is the intermitttent dry joint! Almost every other aspect of electronics has been conquered or improved out-of-sight. Resistors are extremely reliable, IC's have an almost negligible breakdown rate and the prices have tumbled to an all-time low.

In fact the retail price for most electronic products makes a repair unthinkable. You can turn around and buy the latest model for about the same price as the cost of a repair.

This is the case with most faults, but not all.

The one exception is a dry joint. It doesn't cost money to fix - just time. And time is on your side. If you can carry out the repair yourself, you can save a fortune. All it takes is a little know-how. And this is what we hope to impart.

One of the most common faults with electronic items will be an intermittent fault. This comes about through a number of causes, of which the major is rough handling and overheating.

Obviously portable TV's and the like must be moved around during the course of their life, and this causes a certain amount of vibration between the components and the PC board.

This is the situation with large components such as transformers, electrolytics and transistors mounted on heat-sinks. The jarring and vibration can cause the leads to come adrift from tbe PC board and create an intermittent connection.

There is another factor which contributes to the dry joint and this is heat. Heat generated in the component, such as a transistor, diode or even electrolytic, is transferred to the PC tracks via the lead. This heats up the solder connection and can even lead to the component dropping off the board!

The third contribution is via expansion and contraction. As the transistor heats up, the leads increase in length. The body of the transistor is generally restrained via a mounting screw and the only direction the lead can move is towards the PC board. This may be only a microscopic increase but repeated heating and cooling will eventually fracture the connection.

If you have ever seen a dry joint, you will be fascinated. It consists of a very fine ring or gap around the lead. By moving the board relative to the lead you will see the two are completely separate. Its only when certain conditions are met that the connection becomes open.

No matter how the fracture occurs, the end result is the same. All of a sudden the equipment fails. A hit or a bang may get it operating or it may take a rest period before it will come to life.

If it is only a very delicate dry joint, a bang may fix it for weeks. This is a dangerous solution as you will never know when it will fault again.

THE GOLDEN RULE

Locating dry joints is one of the most frustrating and complex of sciences. There is a definite method of approach and if you follow the rules, you will achieve a high degree of success.

The golden secret to fixing an intermittent fault is: **DETERMINE WHICH COMPONENTS ARE NOT AT FAULT AND "MOVE-IN" ON THE SUSPECT AREA.**

This is how you do it.

The first stage to locating a dry joint is to operate the equipment until the fault occurs. If it takes 10 minutes or three hours, be patient. You must see the effect of the fault before starting.

This type of fault is called THERMAL and usually occurs when the inside temperature of the equipment reaches a pre-determined value. A knock or bump may rectify the problem but that is not the solution. We have to go about it scientifically.

When the set faults, give it a little tap on the side. This will determine how delicate the dry joint is.

This is an important preliminary test since we need to remove the chassis from the cabinet and want to know if this is likely to heal the fault. If the slightest tap will cure the problem, you take the risk of finding the set will operate when it is removed.

At least you know the possibility before-hand.

Remove the chassis and power the set until it fails once again. At this stage the skill of a surgeon is required. With the use of only a screwdriver, it is necessary to move over the underside of the PC board and locate the most sensitive area. This is done with a light tapping or rubbing of the plastic end of the driver over the sharp end of the component leads.

This will produce sufficient vibration to spark a very delicate joint into operation when the driver is near. Surprisingly, our aim is not to locate the joint immediately but to determine which areas of the board are NOT sensitive. This is a form of double-check and will be appreciated later.

Depending on the degree of sensitivity, the rubbing and tapping can be increased in strength until you are fairly sure the fault is contained within a small area.

At this stage you can move and wiggle each of the components individually with your fingers and try to pick out the faulty contact. If this is not conclusive, you can proceed to solder ALL the connections in the general vicinity.

But before you heat up the soldering iron, it is a nice feeling to be able to see the faulty joint before it is fixed. If you intend to probe any of the joints, make sure the tools you use are non-metallic. To use anything metal is very dangerous. One short betwen two conductors may damage the set completely. And this, on top of a dry joint, may bring you undone. So, only use plastic tools.

Look for the suspect joint, and if you think you have found it, turn the set off and solder the joint. Switch the set on again and see if it is cured. This is a slow method but it is very smart.

By soldering one joint at a time, you know which is to blame if the set gets worse. Otherwise, if a whole area is soldered at once, you will not know where to look.

If everything else fails, you will have to go over the complete board.

I have done this on about two occasions and although it took about half an hour to complete, and a metre of solder, it can quite oftern be the quicker approach.

Once you are fairly certain the joint has been located, tap the board a few times and replace the chassis into the cabinet.

You should run the set for a few hours to make sure your handiwork has been successful. And then you can rest peacefully.

This procedure applies equally to fault-finding the TEC-1. And also any of the smaller projects we present.

In fact it is universal.

Now to the two specific cases of dry joints in the TEC-1. Both appeared on the address bus and as the keyboard was operated, the display would change to undecipherable characters.

We used the principles described above to 'home-in' on the suspected area and found that the slightest movement on the Z80 socket would cause the computer to go wild.

Even careful inspection of the solder connections around the socket failed to locate anything suspect so we had to carefully re-solder each of the pins.

Fortunately this solved the problem and the units are now back in the field.

The main cause of this type of dry joint is lack of flux. If the flux has evaporated by the time the soldering iron has reached the joint, the solder cannot possibly flow over the copper land and around the other side of the lead to create a good connection.

This is very important as one faulty joint will render a project useless or involve hours of searching .

Talking about searching, If you are searching around for components to build a project, you will be well advised to stop and think. The price for a kit of components is generally considerably less than if the components were bought separately.

Take the case of the TEC-1.

Two readers from Sydney have rung requesting the EPROM for the TEC-1, as they had already purchased the components individually over the past few weeks. By their reckoning, they had spent nearly $15 more for the components than if they had bought the kit. Apart from this, they could not get a uniform style of component and this detracted from the neatness of the finished product.

Some of the sockets were old-style and not low-line, the cermet pot did not fit the board, the 100pf cap was silver-mica type and the resistors were ½watt types. We have also found a mixing of FND 500's with FND 560's. These have different brightness outputs and the 560 is a far superior performer.

The same applies to our small projects. We only use ¼watt resistors and if you can't buy them at your local shop, send to a larger supplier. Why spoil a project for the sake of a couple of weeks or a few cents? Nothing looks worse than a big resistor, standing on end, and crammed into a tight space.

And finally we recommend IC sockets. If you are not 100% proficient with a soldering iron, it is always wise to use sockets. Some constructors have found the secret to damaging IC's. They blow them up time-after-time. The cost of removal and replacement far out-weighs fitting a socket in the first place.

One word of warning. All kits will be 15% more expensive from issue 12 due to the increased costs of components and postage.

LETTERS...

A good range of letters this month with the funniest one kept for last. In fact a lot of letters have had to be kept over for the next issue for lack of space. If you have a query, send it in. It's nice to see your own name in print.

The complexity of the requests in the letters is reflected in the articles we are describing.

The TEC-1 brought a lot of requests for help, mainly via the phone, from constructors who were having last minute problems with the display. For some it flashed, for others it didn't light up at all.

Some were not sure about the 58725 NMOS version of the 6116 while others were wondering if the Z80A would take the place of the Z80.

And rightly so. There is insufficient technical information in books, manuals and magazines and the 'A' could refer to anything from reverse polarity to a minimum clock frequency of 3MHz. Fortunately we now know it means a Max frequency of 4MHz but if you don't know, then it's a mystery. When you know, it seems so trivial.

So that's why we encourage letters and phone calls. It let's us know what is troubling you and what points need more explanation.

Simple things to us can be enormous hurdles to the beginner. The LOGIC PROBE board is a point in question. It being a double-sided board suited for a special purpose, many of the holes were not drilled because the components were mounted on the top side.

We had numerous boards sent back to us for re-drilling and lots of phone calls asking for help in positioning the components.

Via the columns in TE and its data pages we hope to provide technical information on most of the common range of chips and components.

Also, the letter pages let you know what is in the mind of other readers. This could save you writing a letter or encourage you to seek advise on your own particular problem.

In any case we like these pages. It brings everybody close together and puts a personality into the magazine.

Here we go with the first letter:

I am writing to give some feed-back on the TEC-1. I found it a very satisfying project, the first time I have worked on a board with solder-mask, and it certainly tidied up my soldering.

It worked first go and I got just as big a thrill as when my first crystal set made a noise in the earphones, fifty years ago. Or forty years ago when my first valve receiver worked.

I am now waiting impatiently for the next instalment, so don't stall it!

Over the intervening years I have tried to keep up with the latest in electronics (it is not my vocation) and I am sure I would have been left way behind if it hadn't been for your magazine and method of teaching.

One thing I ask as a hobbyist with no formal education in electronics is to include a list of the pronunciation of the new terms.

One could appear a little more knowledgable if asking for a 'triple-5' timer IC instead of a 'five-five-five', as I learn't from your Digital Course. But what about CMOS, EPROM, MOSFET, BYTE, PUT etc? The meanings are always made clear in the text but no-where have I seen how they are sounded.

Yours are the first kits I have ever built. The inclusion of the PC board on the magazine set me going and I am impressed with your efficiency. I think of the hours I spent shopping around for bits. I always thought it was an inevitable part of the hobby.

You may be interested in the use I have put your EGG TIMER to. During the last summer, to relieve the tedium of hand-watering, I listened to the radio with earphones. I removed the speaker and in its place fitted the Egg Timer board, adjusted to time for 45 seconds. The output tone was fed into the amplifier of the radio.

I hung the radio around my neck, tilted it to start timing and directed the hose onto a shrub or tomato plant. When the beep occured, I reset the timer and moved on. The 45 seconds represented one bucket full of water, so all plants got their share and I was not bored to tears with the monotony. Looks like it may be in use again this summer.

N A Cave,
Pascoe Vale, 3044

Nice to see some novel ideas for our projects. I know one reader who times his toast with the timer and another who times his phone call. Doesn't anyone use it as an egg timer?

Here are some electronic words and the way we say them. We will add more later.

Bezel - Bez-el. Not Bee-zel.
EPROM - Eee-PROM.
CMOS - SEA - MOS.
MOSFET - MOSS - FET.
PUT - PEA - YOU - TEA.
555 - Triple - 5.
LED - LEAD (as the metal) **or EL - EE - DEE.**
MFD - 'Mickey'.
pF - PUFF.
BYTE - BITE.

For the past eight months the ACT VIC-20 Users Association has been producing a bi-monthly magazine with articles, programs, reviews and the like, for VIC-20 owners. The magazine is distributed by subcription and through local shops. The cost per copy is $1.50 posted or a 6-issue subscription for $8.00 posted. It has an average of 34 pages and 12 programs ready to type in. There are sections including Utility programming, Games Music, Graphics, Input/Output, Education and General Programming.

Readers can also purchase any number of the group's public domain programs from its library for only $2.00 each plus $2.00 for the tape to put them on. The library contains over 400 programs from places as far away as the Netherlands and Canada.

If anyone is interested in joining the VIC-20 club, they can write to me:

Chris Groenhout,
The ACT VIC-20 Users Association,
25 Kerferd St.,
Watson, ACT, 2602.

Thank you for the sample copies.

We have decided to discontinue the VIC-20 column, mainly through lack of interest and were very pleased to hear from a group of readers who had formed a club magazine.

If you own a VIC-20 computer, you will be very interested in the magazine as well as the program library. This is an ideal way of getting new input at a reasonable cost.

You requested feedback and I submit the following points for thought:

I found the photographs for the Probe difficult to follow and much prefer a line drawing for the layout. No-where in the text does it state that the components passing through the board must be soldered on both sides. Nor does it indicate which holes need a piece of wire to connect the top of the board to the bottom.

W R Symms,
Redcliffe, 4020.

We greatly appreciate these comments and have produced a line drawing for the probe on the centre pages of this issue. I realize we took too much for granted on our first double-sided board. A number of constructors had similar problems putting the probe together.

Due to the fact that a probe must fit into a tube, the available space for the PC board and components is limited. This means a double-sided PC board must be used to the greatest advantage. All the components are mounted on top of the board so that copper lines can run on the other side of the board and reduce the need for jumper links.

But this also requires fitting jumper links or FEED-THROUGHs and sometimes you have to solder the component leads on both sides of the board. See P 39 for diagrams to help construction.

I recently came across TE in the newsagents and was attracted by the fact that a PC board came with the magazine. From a country point of view, this is important.

I have made the Dual Tracking Power Supply... it worked from the moment I turned it on, the Logic Designer and the Logic Probe. All these projects worked first-off.

Now I have built the Z80 computer and again it worked straight away.

I am now waiting for the next issue of the magazine. There are so many things you can do with a Z80 that I am eager to start. I have a number of simple queries about computers but I think they will be answered in forthcoming articles.

F. M. Measday,
Millicent, 5280

I would like to offer some thoughts on ways your already fine magazine may be improved.

I would like to draw your attention to a system utilized in 'ELEKTOR'. Whenever possible, this magazine uses the term TUN or TUP for low power transistors. This indicates a universal NPN or PNP transistor such as BC 547 or BC 107 for the NPN and BC 177 or BC 557 for the PNP.

Diodes are labelled DUG for Diode Universal Germanium and DUS for Diode Universal Silicon.

I might also add that this publication is now being produced in Australia and is an ideal 'ideas' magazine for those who are getting to grips with electronics.

In the same vein, 74 series IC's are very plentiful and cheap on old computer boards and bags of factory-floor sweepings. The non-LS is especially plentiful and therefore it would be appreciated if some details could be given for converting circuits to use these chips.

How about giving workable circuits to match your theory notes or are some them workable?

In conclusion, I must say that I find your magazine most interesting and my knowledge of digital electronics is gradually improving.

P. Mumford,
Cammeray, 2062

I'm glad you have noticed ELEKTOR has recently introduced an Australian presentation of the overseas material. It has always been a thoroughly worthwhile publication and is strongly recommended to all readers.

Even if you don't understand its content at the moment, it is a non-ageing publication in which the circuits will be just as useful in 10 years or 2 years or when-ever you need to build something more advanced.

I buy ELEKTOR regularly and have seen their presentation of DUS, DUG, TUN and TUP. I think it is a good idea, especially for a magazine intended for world-wide distribution.

For the beginner, these terms are a little off-putting. They like to see the component values and the transistor types. Even the letters such as BC 547B or 547C confuse a lot of constructors. Imagine what DUS would do?

After all, if you are an advanced constructor, you can mentally perform the same reasoning with our circuits as BC 547 is ANY low power NPN transistor and BC 557 is ANY low power PNP transistor.

That's why we have used these transistors in all our circuits. We have carried out your request without you realizing it!

In answer to you next point: CMOS technology and TTL technology are completely different. The impedance factors for the circuits are at opposite ends of the scale. We have not used any TTL chips in our projects due to their need for accurate rail voltages and high current consumption.

However we will be producing a text book in the near future using a TTL TRAINER to look at GATING, SHIFTING and COUNTING. It will have about 25 experiments and 25 projects using the DESIGNER BOARD. It will be a very informative text book and the projects will be worth investigating.

And to answer your last point: I consider we are already backing up the theory sections of the magazine with plenty of projects and if you can expose a large gap in any section, please let me know.

I am a keen reader of Talking Electronics but I have one major problem. My regular newsagent has not received a single copy of TE for months. Are you only selling to subscribers or have you stopped producing the magazine.

I know you still exist because I found your latest project book in a shop.

I think you should keep the PC boards attached to your magazine because many people don't have the resources or time to make their own. For people in New Zealand, like myself, making your own can prove to be quite expensive if you do it the way suggested in the first issue. It involves chemicals which aren't available in NZ and to send away for them is very costly.

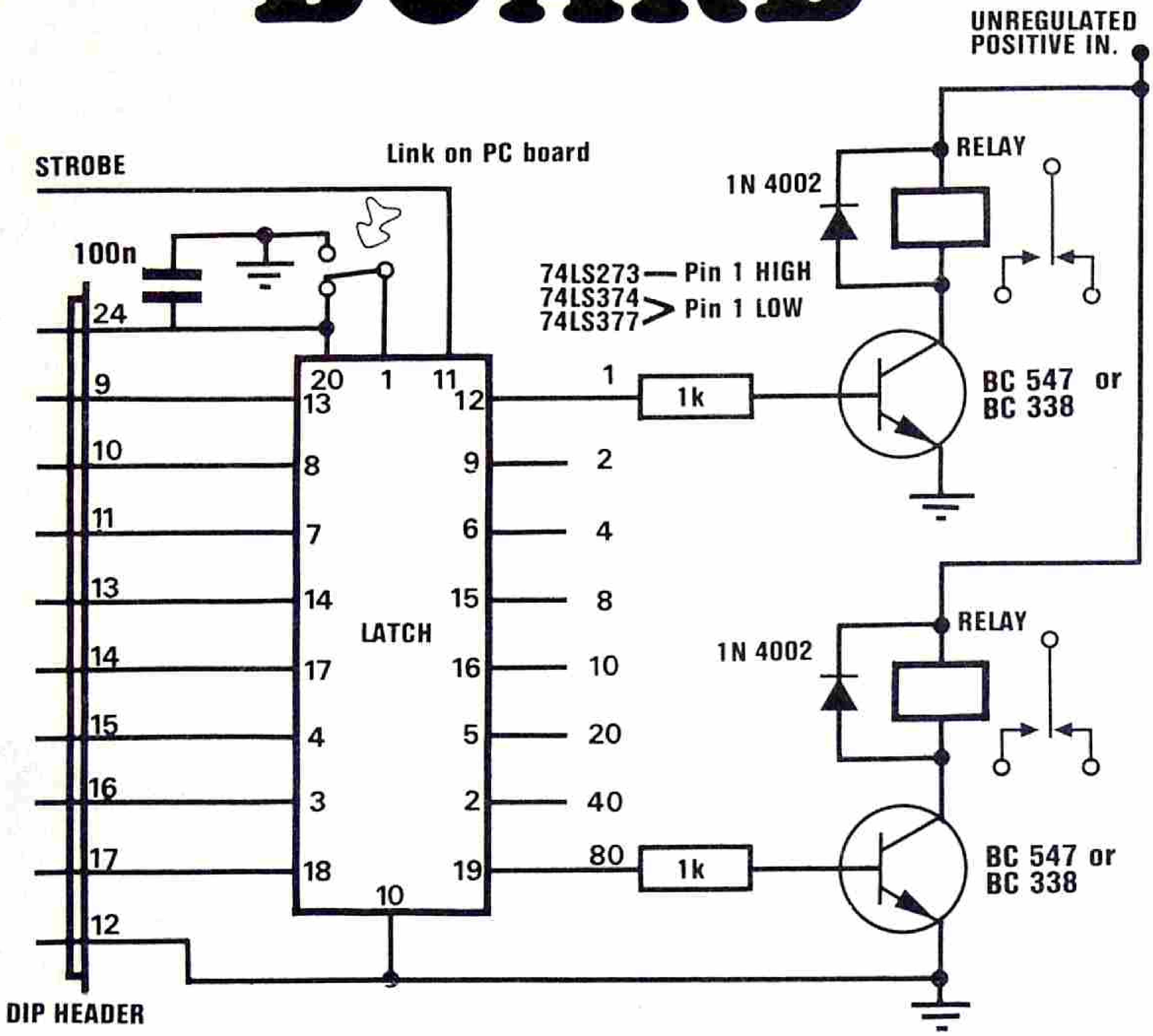
So, please keep the boards attached.

M Young,
Dunedin, NZ.

The part I like is "I know you still exist. . . ."

RELAY DRIVER BOARD

$25.60 Complete



TEC-1 RELAY BOARD

The TEC-1 Relay Board has been designed to give the TEC-1 an interface with the real world.

It can be used in such applications as the control of model railways or complex light sequencers. It can also be used to implement timing and control functions for machinery.

The project will drive 8 on-board relays or external relays with suitable coil resistances and/or it is possible to drive small low-voltage lamps directly.

We do not recommend switching 240v via the relays on the PC board as the terminations are very near the other TEC-1 components.

If you require to switch 240v, the on-board relay can become a slave to power the higer-voltage relay.

Each of the relay contacts has been brought to the edge of the PC board where it is labelled 'NO' for Normally Open contact and 'NC' for Normally Closed contact. The centre 'C' is the Common contact.

The Hexadecimal value for each relay has also been printed on the board to make programming easier.

If relays are not required, small light bulbs can be connected between the two holes normally used for the relay coil.

CONSTRUCTION

Three types of latch chips can be used on the board. These are: 74LS273, 74LS374, or 74LS377.

Each chip is capable of performing the latch function and the only pin which requires a different logic level is pin 1.

On the 74LS273, pin 1 is master RESET, active LOW. So this pin must be tied HIGH to stop this chip from remaining in the reset condition.

On the 74LS374, pin 1 is an output enable, active LOW. When pin 1 is HIGH, the outputs will go to a high impedance state, effectively disconnecting them from circuit. When pin 1 is LOW, the outputs are normal, being either HIGH or LOW, depending on the contents of the latch.

Pin 1 on the 74LS377 is an ENABLE, active LOW. When this pin is HIGH, the contents of the latch cannot be changed by any of the input signals. When the pin is LOW, the latch operates as normal.

None of these special funtions are required on our project so pin 1 can be tied HIGH or LOW and remain in this condition.

For the 74LS273, pin 1 is tied HIGH and this requires a very short jumper as shown on the overlay.

For the 74LS374 and 74LS377, pin 1 is tied LOW and this requires a jumper link the full length of the IC as shown on the overlay.

Solder in the necessary link for pin 1 and another link on the other side of the IC.

Next fit the eight 1k resistors, the eight NPN transistors and eight protection diodes. A 100n decoupling capacitor fits above the input socket.

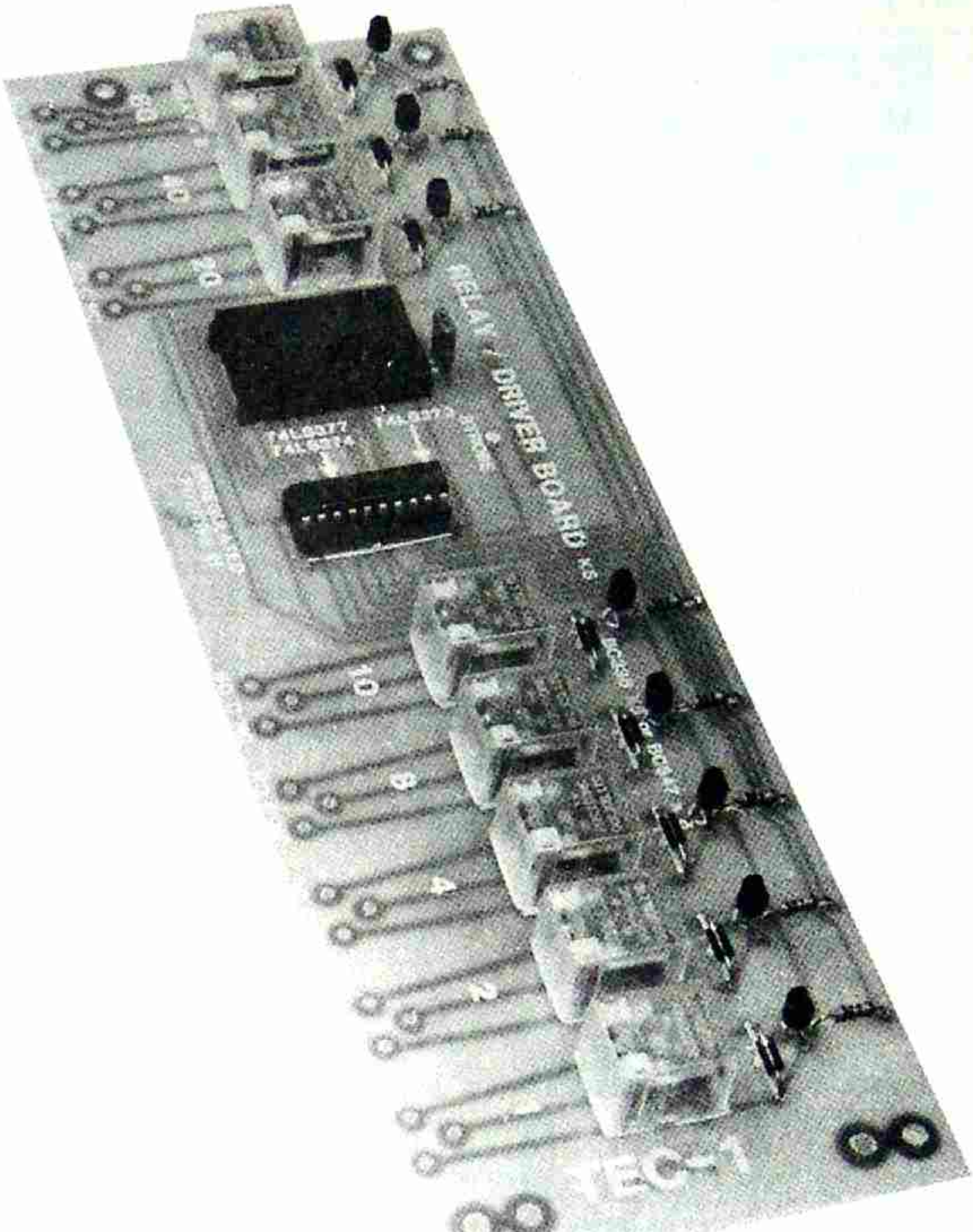
The board is designed to take two types of relays. They are both single pole double throw types and one has a contact rating of 1 amp, while the other has a 3amp contact rating. Select the type your require and install them.

This unit is designed to plug into the EXPANSION PORT SOCKET on the TEC-1 and there is two methods of connecting them together.

One is to individually wire the leads onto a dip header or dip plug and plug it into the expansions socket. The other is to use a wire wrap socket. The socket is inserted into the holes on the PC board and pushed home. The pins are now carefully soldered. Next a dip header is soldered to the end of the long leads of the wire-wrap socket. Solder the corner pins first, then align the rest of the pins and solder them. Care must be taken not to over-heat the dip header as the plastic melts easily.

Finally insert a 20 pin IC socket and fit the latch IC.

Two jumper wires are also needed to complete the wiring. These are taken from the 'STROBE' hole and the UNREGULATED POSITIVE IN hole.



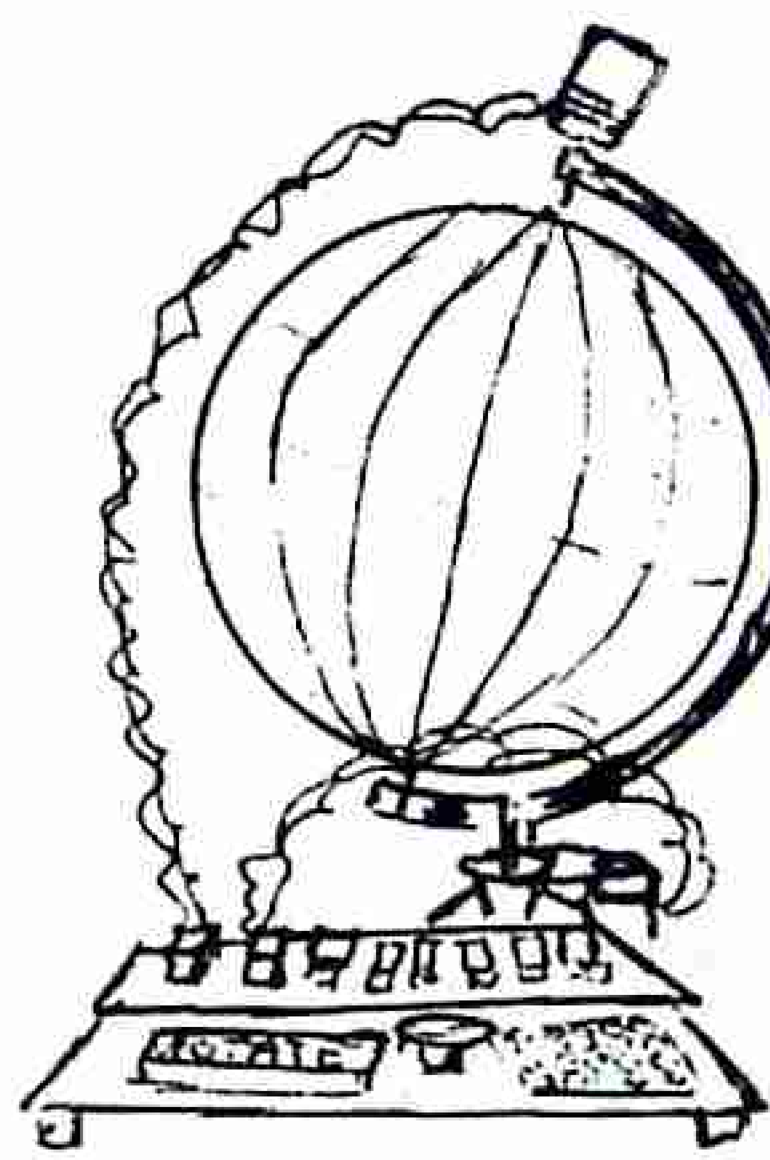
COMPLETED RELAY DRIVER BOARD

The Unregulated IN line goes to the INPUT terminal of the 7805 regulator as 12v is needed to operate the relays.

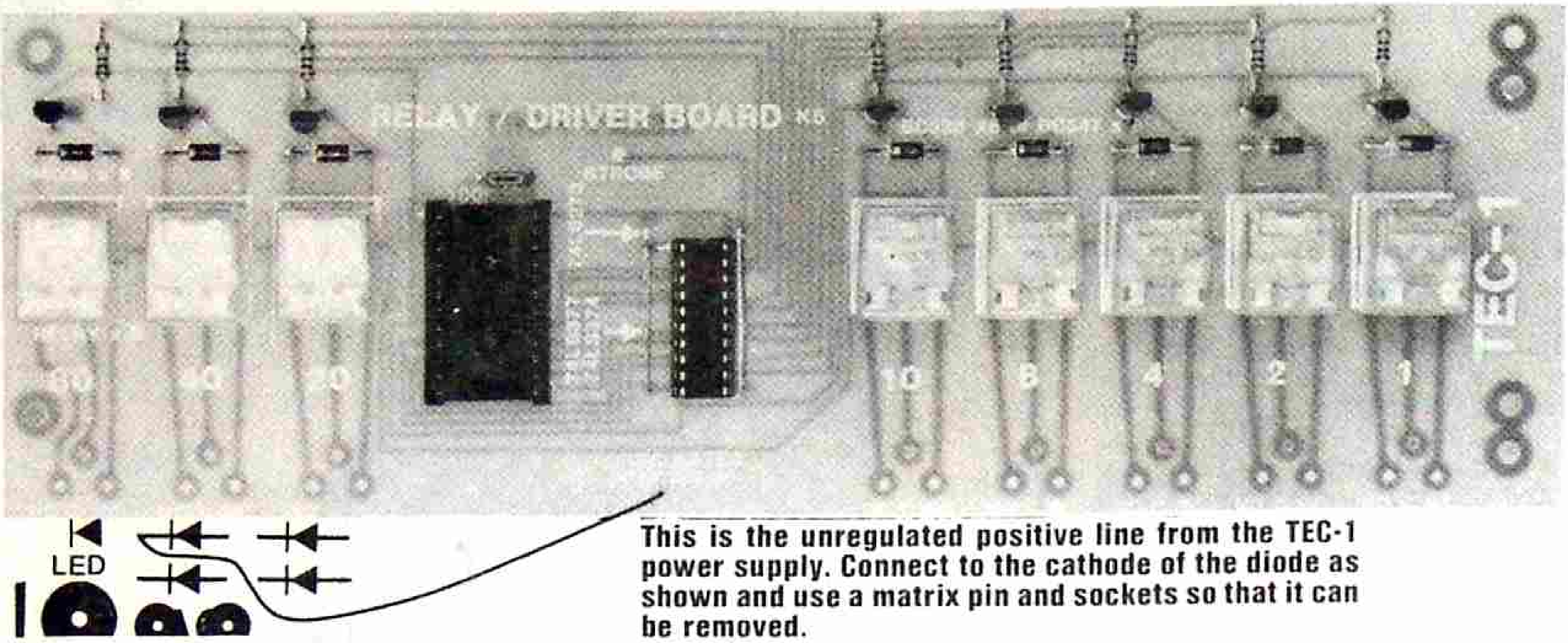
The other line, from the STROBE hole, is taken to pin 12 of the 74LS138 IN-OUT decoder nearest the 74C923 chip. This is output 03.

Solder a 24 pin IC socket to the EXPANSION PORT on the TEC-1 and plug the RELAY BOARD dip header socket into this expansion port socket. The board should fit neatly above the TEC-1. The printing on the RELAY board will be up-side-down because all wiring to the relays is done from the back.

The unit is now ready for use.



I TOLD THEM I WANTED TO CONTROL THE WORLD!



This is the unregulated positive line from the TEC-1 power supply. Connect to the cathode of the diode as shown and use a matrix pin and sockets so that it can be removed.

HOW IT WORKS

The latch board works in the same manner as the display latches and displays.

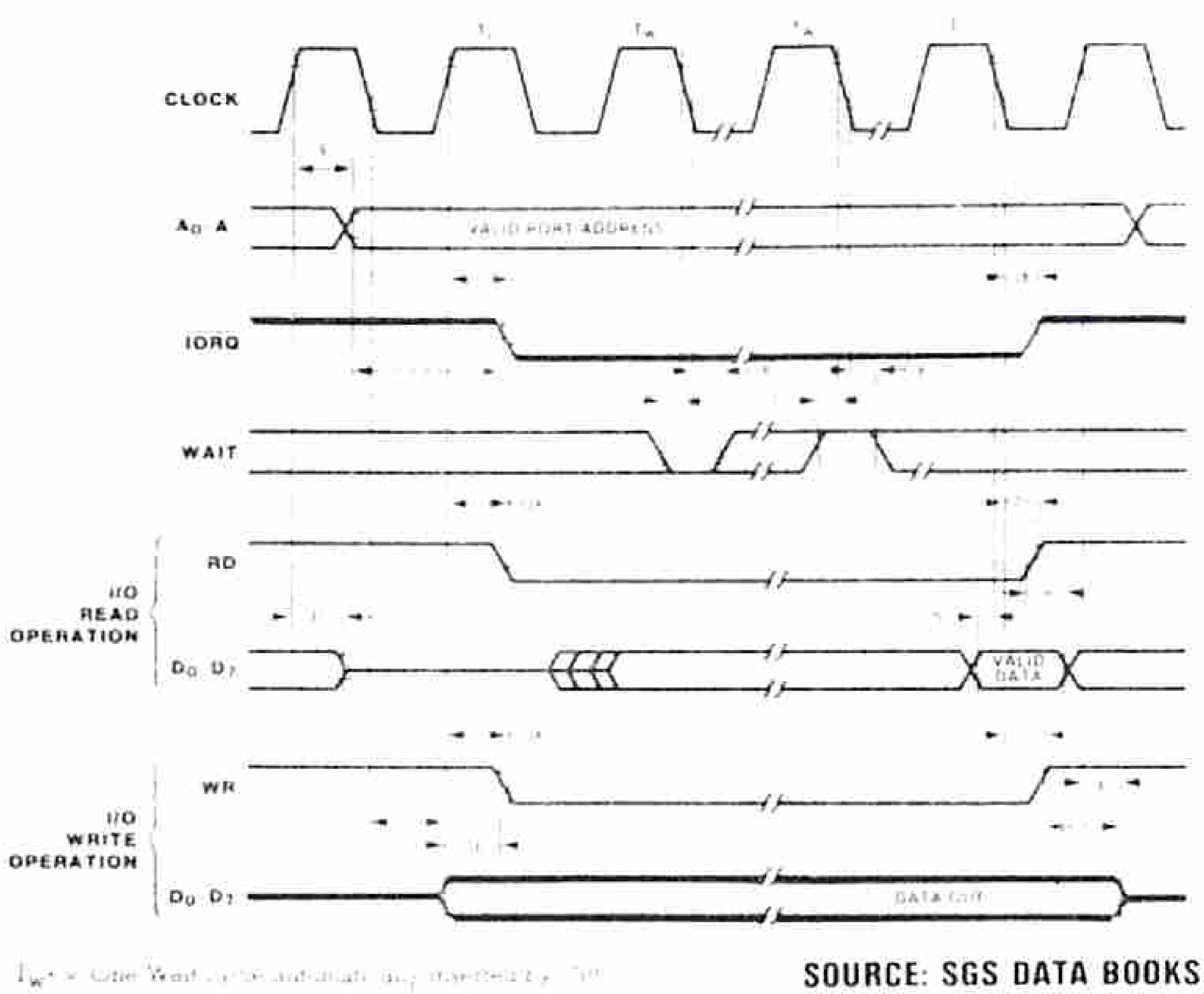
Look at the timing diagram for the Z80 INPUT/OUTPUT cycle.
Compare the IORQ signal to the DATA BUS signal. It can be seen that the DATA BUS stabilizes first, then the IORQ signal goes LOW and HIGH again.

During this the DATA BUS has remained stable and now that the cycle is complete, it is no longer required.

The latch used is a positive-edge-triggered device. It can be seen that the positive-going edge of the IORQ, which is passed to the latch via the 74LS138, comes late in the cycle, but as the data is still stable, the latch is able to record this information and energise a relay.

The port which is to receive the data is selected by the 74LS138, according to the information on the address bus. There are 256 ports addressable by the Z80, but only 8 are decoded on the TEC-1. This still gives us the availablility of using 40 relays!

Up to 5 boards can be added to the TEC-1. These are stacked on top of each other and accessed via ports 03, 04, 05, 06, and 07. Ports 00, 01 and 02 are used by the TEC-1 for the keyboard, and displays.



SOURCE: SGS DATA BOOKS.

HOW TO USE THE RELAY BOARD

If you have a Mon 1A EPROM, try the following:

Start at address 0800, program the following: 01, 02, 04, 08, 10, 20, 40, 80, FF ADdress 05B0, GO, GO.

The relays will sequence in order.

If you have the MON-1 EPROM, enter the following at 0800: 3E, 01, D3, 03, C7. RESET, GO.

The first relay will switch on and the computer will reset. Substitute different values for 01 in the program and watch the operation of the relay or relays.

With this RELAY BOARD your TEC-1 can access the REAL WORLD.

We will now show you how to use it and produce delay routines of varying length so that you can turn lights, motors and pieces of equipment on and off.

PARTS LIST

8 - 1k ¼ watt

1 - 100n greencap

8 - BC 547 or BC 338 transistors

8 - 1N 4002 diodes

8 - relays SPDT type S 4060

1 - 74LS273 Latch IC

1 - wire-wrap socket 24 pin
1 - dip header 24 pin
1 - 20 pin IC socket

The RELAY BOARD takes up to 8 relays and these can be turned ON and OFF in absolutely ANY order, by the program we have developed.

The only limitations are very heavy currents and high voltages. The relays are somewhat exposed and are close to the components on the TEC-1, so we do not suggest 240v switching.

Secondly the relays are designed for 1 amp operation and should be limited to around this value for reliable operation.

We suggest only low voltage operation (12v) with currents up to 1 amp for the relays and transistors. We have a high current version (3 amp), S 4066 and this will enable loads up to 36 watts to be handled by each output. This is quite considerable in electronic terms.

Higher currents and voltages will have to be handled via a relay which is remotely positioned so that it can be provided with the insulation it needs. Don't forget, any relays switching high voltages can still have a 12v coil. This means that you can combine relays and transistors on the one RELAY BOARD.

For fast sequencing, a set of transistors will be the best choice as relays have a low operating frequency.

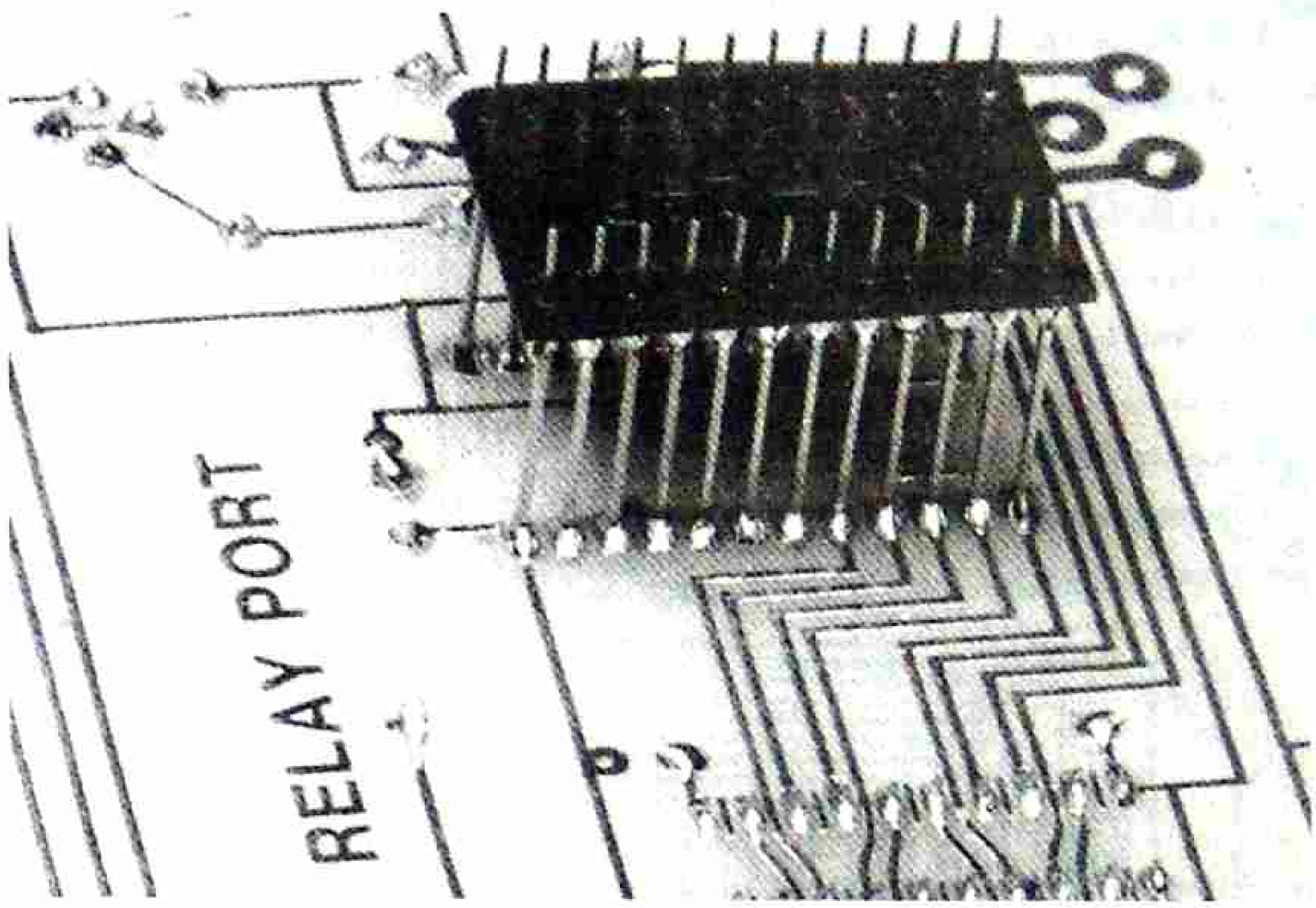
The choice of using a relay or transistor will depend on the type of load you are driving. You will need to know such factors as current requirements, voltage, speed of operation and if any spikes are present.

Our main concern at this stage is to test the RELAY BOARD and get it working.

To do this you will will need some form of indication on each of the outputs to show when they are operating. This can be a miniature lamp or a light emitting diode. Fit 8 of these and keep them near each relay to show which relay they are indicating.

The program that we have developed for this project controls the 8 outputs and will switch them ON and OFF in any combination, as per the program.

It is merely a matter of changing the data at **0A00** to produce the required sequence. The data governs the time of the delay between each operation and this can be set for a millisecond delay or up to a 100-hour delay. To do this requires 4 bytes of information. The next byte governs how many relays are activated at the end of each delay. These 5 bytes form a 'UNIT' or 'STEP'.



The diagram shows a close-up of the plug which connects the relay board to the TEC-1. It is made up of a wire-wrap socket and a dip-header plug. The wire-wrap socket is mounted on the components side of the board with its long leads extending out the side shown in the photo. The dip plug is soldered onto these 'thick' leads, effectively turning them into thin pins suitable for inserting into the EXPANSION PORT socket.

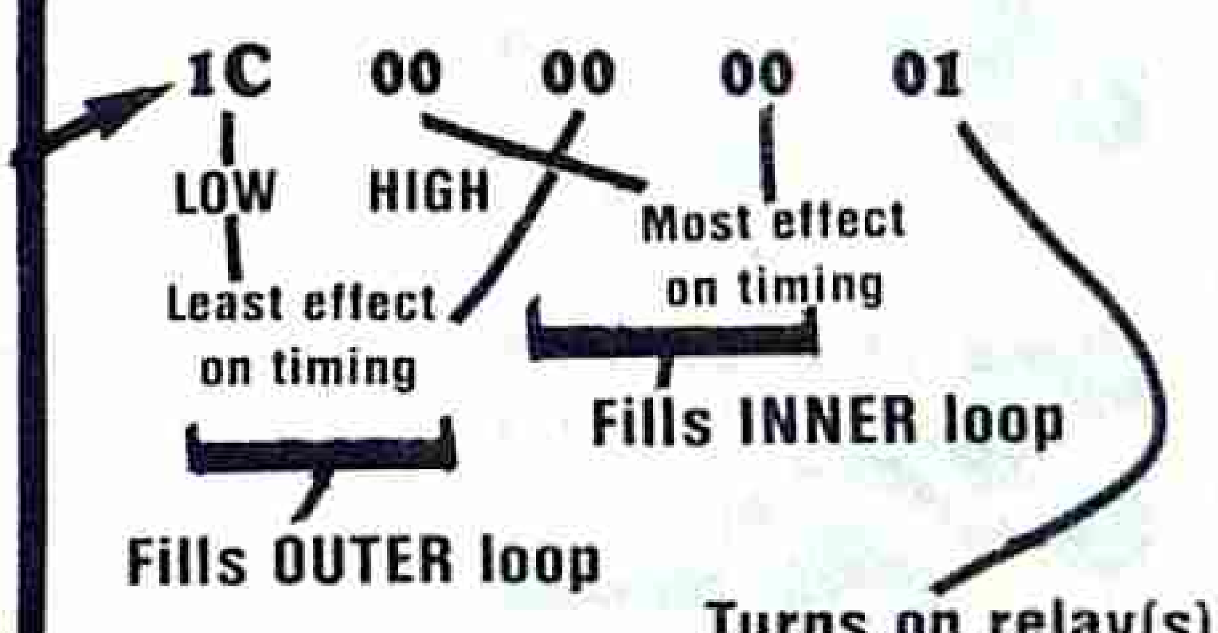
THE PROGRAM:

```
LD B,dd            800  06 08 (No of steps)
LD HL,A00          802  21 00 0A
LD IX,904          805  DD 21 04 09
LD A,(HL)          809  7E
LD (IX +00),A      80A  DD 77 00
INC HL             80D  23
LD A,(HL)          80E  7E
LD (IX + 01),A     80F  DD 77 01
INC HL             812  23
LD A,(HL)          813  7E
LD (IX+ 03),A      814  DD 77 03
INC HL             817  23
LD A,(HL)          818  7E
LD (IX + 04),A     819  DD 77 04
INC HL             81C  23
LD A,(HL)          81D  7E
INC HL             81E  23
OUT (03),A         81F  D3 03
CALL 0900          821  CD 00 09
DEC B              824  05
LD A,B             825  78
OR A               826  B7
JP NZ, 0809        827  C2 09 08
JP 0800            82A  C3 00 08

PUSH AF            900  F5
PUSH BC            901  C5
PUSH DE            902  D5
LD BC, FF FF       903  01 FF FF
LD DE, FF FF       906  11 FF FF
DEC DE             909  1B
LD A,D             90A  7A
OR E               90B  B3
JP NZ 909          90C  C2 09 09
DEC BC             90F  0B
LD A,B             910  78
OR C               911  B1
JP NZ, 906         912  C2 06 09
POP DE             915  D1
POP BC             916  C1
POP AF             917  F1
RET                918  C9

DATA               1C 00 00 00 01
at 0A00:           38 00 00 00 23
                   1C 00 00 00 86
                   1C 00 00 00 2C
                   38 00 00 00 38
                   1C 00 00 00 32
                   1C 00 00 00 62
                   1C 00 00 00 C0
```

The theory of operation of this program is quite complex. It will take two or more chapters before we are up to describing its features. In the meantime we will describe it in a simplified way.

What each byte does:

At **0A00** the data bytes are taken in groups of 5:




TEC-1
RELAY / DRIVER BOARD
KS
NO C NC
80
RELAY x 8
1N4002 x 8
1K
40
20
RELAY / DRIVER BOARD
KS
100n
EXPANTION PORT
74LS377
74LS374
74LS273
STROBE
UNREGULATED POSITIVE IN
LATCH
10
8
4
2
1
BC338 X8 or BC547 x 8
TEC-1

. . . . .from p. 53.

Here is an overall view of the program:

The program consists of 3 sections: The main program is at 800. This can be likened to a cook book. They both give an overall picture of what you are trying to achieve.

The second part of the program is the DELAY ROUTINE. This is at 900 and is similar to the mixing and cooking times for a recipe. They both involve TIME, and this is the important element of this section.

The third part is the DATA. This is similar to the ingredients: flour, milk, eggs, etc. The more flour you add, the longer you have to mix.

That's a simple analogy. This is how it works:

The main program (the procedure for the recipe. Such as getting out the bowls and equipment) contains all the 'setting-up' data and instructions with the first instruction, at 801, telling the computer how many cakes (steps) you require.

At 900 the length of time for the mixing will depend on quantity of flour and milk is going to be used.

0A00 contains the quantity of these ingredients.

The program can make up to 256 cakes (steps) before it starts to repeat.

The main program may look complex but most of it is a set-up routine, like getting everything together to make a cake.

To extend this program is very easy, all you need do is add a 5-byte block of data to the end of the data at 0A00. Each block will produce one more delay and instruct a set of relays to come on at the end of the delay. The value of 'B' at 801 must correspond to the number of steps in the program otherwise it will be forgotten!

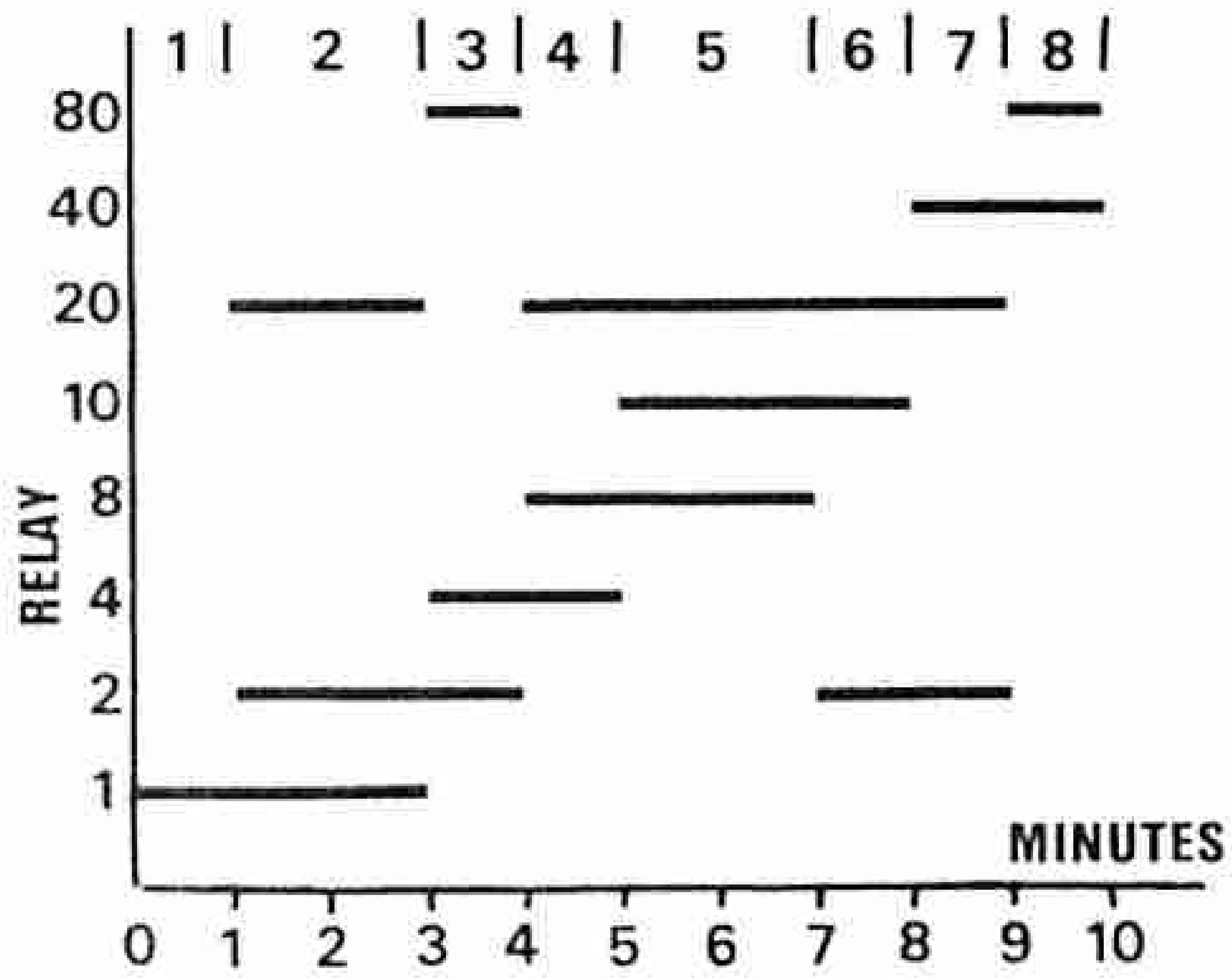
The first three bytes in the delay (at 0900) PUSH the registers onto the STACK making the delay 'transparent' to the main program.

At the end of the delay, the registers are POPPED off the stack. This means that they will contain the same value after the delay, as they had before it.

The main part of the delay program consists of two routines, one inside the other. The inside or 'nested' delay starts at 906 and uses the register pair DE.

The delay-value is loaded into the register pair and decremented by one. The accumulator is then loaded with the contents of the D register and the E register is ORed with the accumulator. The result appear in the accumulator. The accumulator will be zero if both D and E are completely zero and this will set the zero flag.

The next instruction JP NZ 909 checks if the zero flag has been set and if it has, the computer will execute the next instruction. If it has not been set, the computer will jump to 909 and perform another DECrement instruction.



The graph of our program. It shows the operating time for each of the relays.

When the nested delay is zero, the computer will decrement the outer delay, check that it is not zero and proceed to decrement the inner delay. When the two delays are BOTH zero, the original values of the six registers are pulled (POPed) off the stack and the program returns to the main program.

DELAY LENGTHS

The values 00 00 in the data at **0A00** may look like a short delay length, but they are not. If fact we have tried to trick you! **00** is, in fact the longest delay you can get! If you load the register with **00 00** it is decremented by one to get **FF FF** and then checked to see if the value is zero. As you can see, it is not! The shortest delay is **01 00** as this is immediately decremented to **00 00**, then checked.

To give an example of the time taken to execute **00 00**, we place **00 00,** in the nested delay and **01 00** in the outer delay. This will produce a time delay of 2 seconds, when the computer is turned to the high position. The value **1C 00** for the outer delay will produce a time of about 1 minute.

If you require a delay of 10 minutes, the value of the outer delay is increased 10-fold by inserting the value 1C 0A. For a 100-minute delay, the value is 1C 64. See the HEX table on P.16 for additional values.

RUNNING THE ROUTINE

Enter the program into the TEC-1 and set it into operation. It will take 10 minutes to execute and has 8 steps. There are six 1-minute steps and two 2-minute steps.

The length of the delay for each step can be individually programmed as can the data for the number of relays which are to be activated. The accompanying diagram shows the various relays which will be activated and by referring to this, you will be aided in designing your own program. More details will be given in the next issue.

Complete Relay Board positioned on the TEC-1 computer.

OF ALL THE LUCK!

Its 10.30 Friday night. The telephone rings. A rather dissatisfied customer blurts out a complaint.

"I bought four of your kits today and none of them . . . .work!"

"I've spent three hours putting them together and none of them work properly. I'm going to come around to your door-step tomorrow and demand my money back. In fact I don't think any of your kits work and if I don't get satisfaction, I'll . . blah . . blah . . . blah . . . "

After getting it all of his shoulder, I thought I would shock him.

In a calm voice I said, "Would you like to bring them around now and I'll fix them.

"Aw, no, it's too late, and anyhow you couldn't fix 'em."

"No . . .come around now. We work till 2am and I bet I'll get them going in 10 minutes."

"I bet you couldn't!" "I work for a television rental company and I fix video cassette recorders and I know all about . .blah . . blah . . blah . ."

Anyway, after a few minutes of persuasion, I talked him into coming around in the dead of night to get his new-bought gadgets going.

He arrived at 11.30

I was eager to see what had gone wrong with four-out-of-four projects. I had not had the opportunity to see a faulty project for such a long time and was keen to see if any of the kits were failing to operate due to the spread of parameters on components.

The transformation from an aggressive, aggrieved customer on the phone, to a quiet, polite, passive person at the door, was quite remarkable.

He came with a friend who had initially purchased the kits. Since they didn't work, the *electronic wizzard* had been asked to get them going.

Here were two of the quietest customers one could wish for.
Possibly they were skeptical about the 10 minutes, or it could have been the lateness of the night. Never-the-less, they had arrived.

The first project on the bench was the LED Zeppelin. When the battery was connected, the mini LED and the first LED in the row flashed for a few pulses and died. I immediately thought of the 4.7mfd electrolytic. The clock oscillator circuit using a CD 4001 is not a particularly good design and we found that reversing the electrolytic would sometimes get the oscillator working.

Unfortunately this did not help in our case. So I had to look further. Using a test resistor and a 6v battery, I tried to turn on the row of LEDs. But only the first two LEDs came on.

After careful inspection, I found two dry joints on the board. The remainder of the soldering was good but it was obvious that insufficient solder was used to create most of the connections. This resulted in two poor joints on the collector and base leads of the second and third transistors.

With these rectified, the seven LEDs came on when the PNP transistor was touched with the test resistor.

But still the IC did not give a good flash rate to the mini LED.

It had to be the IC. So it had to be removed. By holding onto the IC with my fingers, and running the soldering iron down one side of the pins, I was able to lift one side of the IC nearly off the board. By repeating the same procedure down the other side I was able to remove the chip.

To be on the safe side, I fitted a socket to the board and inserted another IC. It worked perfectly. It took a little longer than 10 minutes but at least we got it going!

The next project to appear out of his pocket was the LED DICE. Only two of the LEDs were coming on. By looking into the body of the LEDs, I noticed that one of the outer LEDs was inserted around the wrong way. When this was corrected, four of the LEDs came on.

The last two LEDs were tested with a test lead and one was found to be damaged, possibly due to soldering. So it was changed. When we turned the DICE on, it worked perfectly. The LEDs flickered and slowed down to a six!

Now to the third project, HANGMAN.

By touching the TOUCH SWITCH a number of times, only the first few LEDs would light up. The first section to head for was the BOOST. On the positive lead of the 100mfd electrolytic, I got a reading of 7 volts. It was obvious that the oscillator was not working. Taking the previously damaged components into account, I took a punt and reasoned that the top CD 4001 could have suffered the same fate.

As a double check, I took a CRO and sampled the output waveform. Sure enough, it was nothing like a square wave and was appearing at a very high frequency. The amplitude was well below that expected and this is why it failed to drive the buffer transistors.

With the IC changed, the output waveform became a clean square wave and the boost voltage appeared.

The next fault appeared to be with LEDs 12 and 13. Testing the transistor stair-case, all the LEDs except LEDs 12 and 13 would light up. This turned out to be one LED incorrectly inserted and the other LED damaged during soldering.

Everything now seemed to be working well. Only one slight adjustment was needed. Led 8 did not flash when all the LEDs were illuminated. Even when the trim pot was turned around to the most sensitive position. To bring the pot into mid range, the 390k resistor between the trim pot and the bases of the transistors was reduced to 330k.

The same effect could be achieved by increasing the 390k between the trim pot and earth, to 470k. Now the trim pot could be set to its mid position and the 8th LED would flash when the 'feet' were lit.

"There you are . . .these are finished, what about the fourth project?"
"Oh, I haven't started that yet!"
"When these didn't work, I got so annoyed, I rang you."
"Now I can go back and make up the COMBINATION LOCK."

As I ushered them out, I thought "I hope my rented video cassette recorder never needs a repair!!"

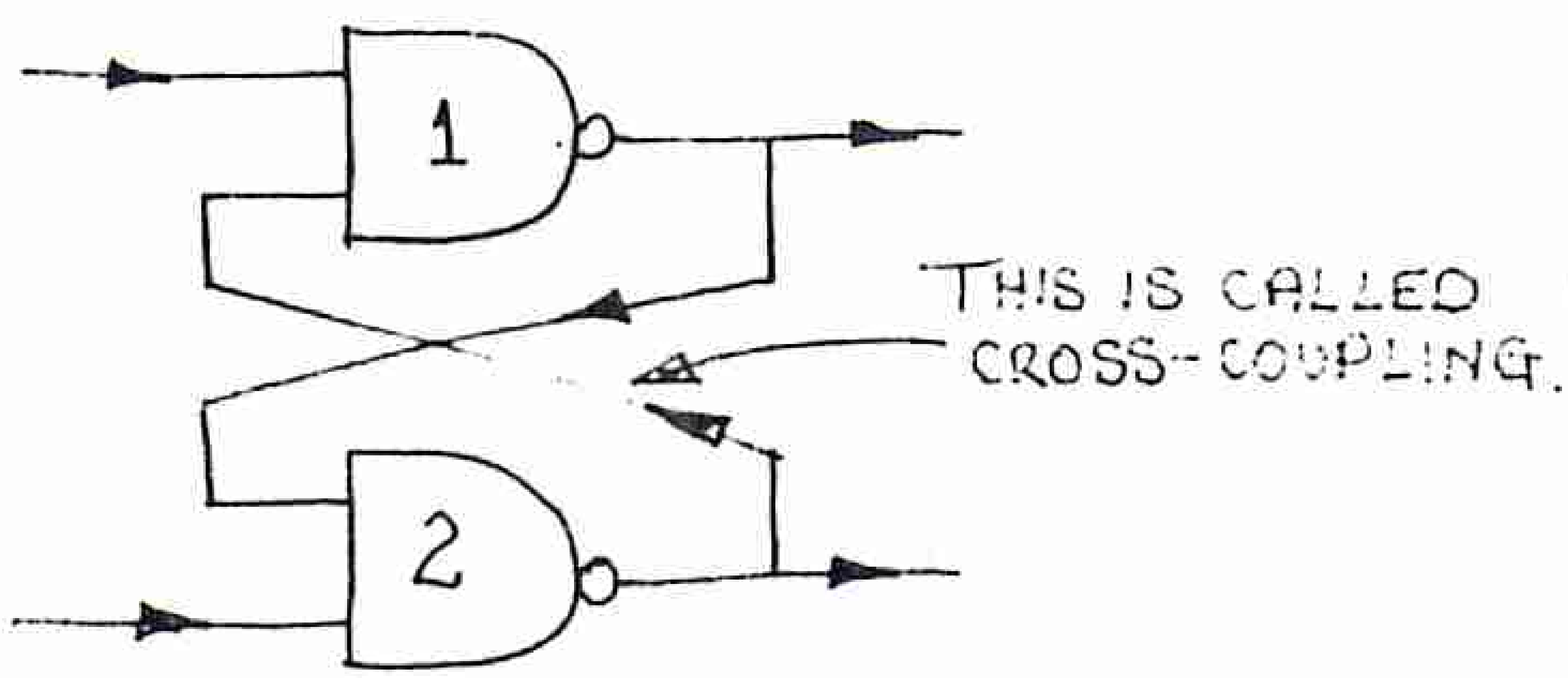
FLIP FLOPS

THE FLIP FLOP WAS INTRODUCED IN FRAMES 17 TO 21 AND THEY SHOWED HOW A NOR GATE & NAND GATE COULD BE WIRED TO PRODUCE A FLIP FLOP.

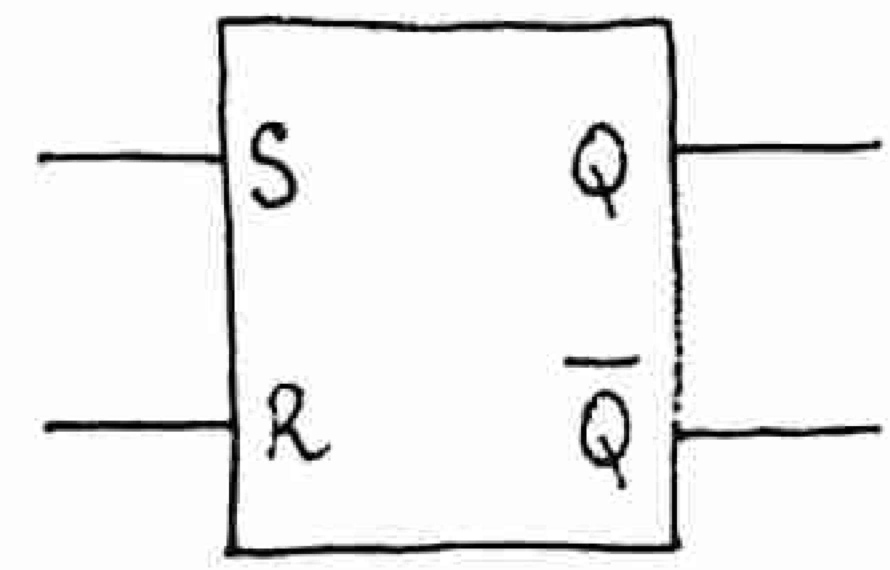
A FLIP FLOP IS AN IMPORTANT BUILDING BLOCK AS IT IS A BASIC STORAGE CELL — IT HAS A MEMORY.

IN THIS SECTION WE WILL EXTEND FLIP FLOP THEORY AND INTRODUCE SPECIALIZED TYPES OF FLIP - FLOPS.

A FLIP FLOP CAN BE MADE FROM 2 NAND GATES OR 2 NOR GATES FIRSTLY WE WILL DISCUSS THE NAND GATE FLIP FLOP.



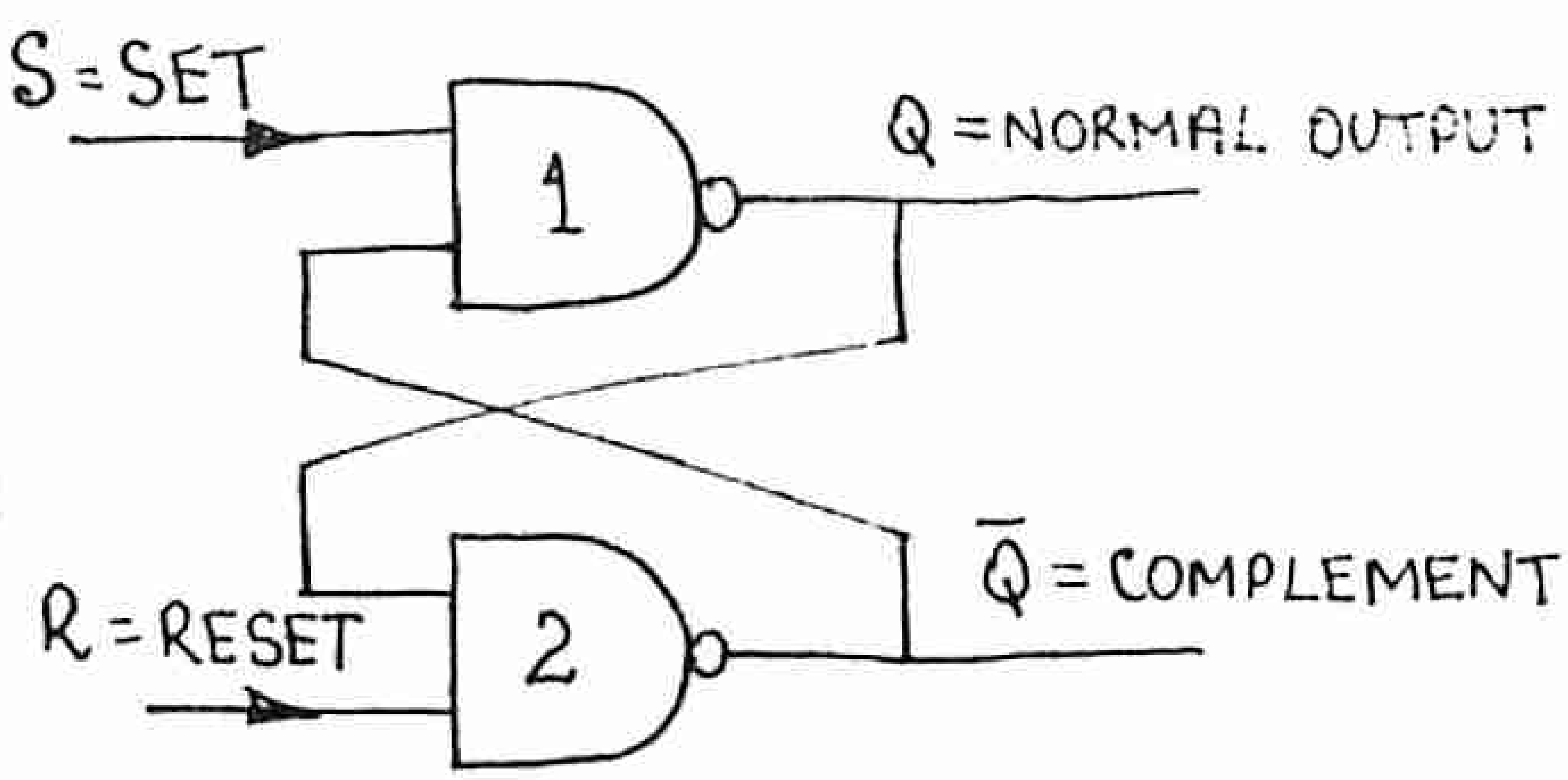
NAND FLIP FLOP



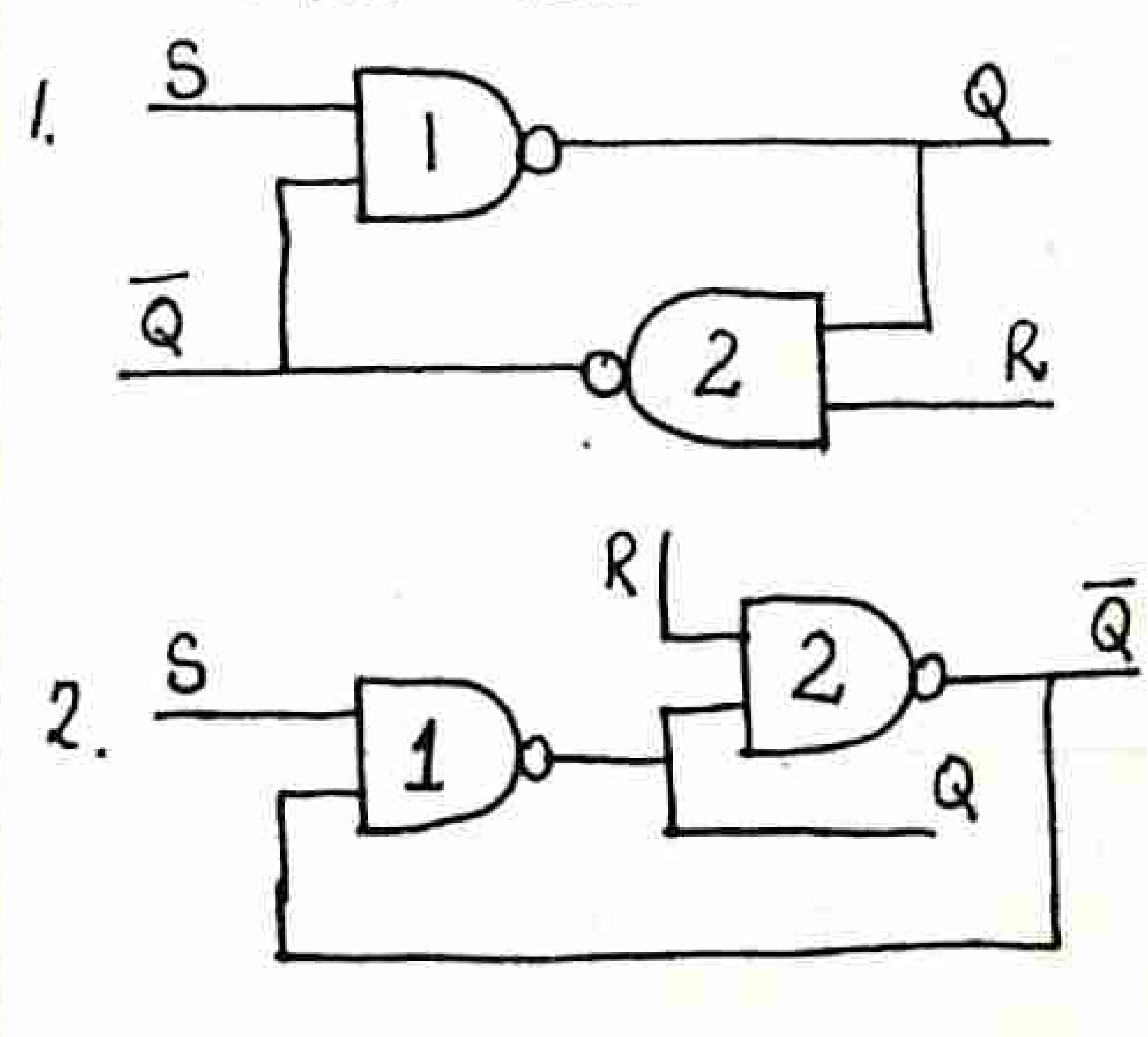
R-S FLIP FLOP

THE 2 GATES ARE WIRED SO THAT EACH OUTPUT FEEDS INTO ONE INPUT OF THE OPPOSITE GATE.

THE INPUT & OUTPUT LINES ARE IDENTIFIED WITH LETTERS:



2 OTHER WAYS OF DRAWING A FLIP FLOP:



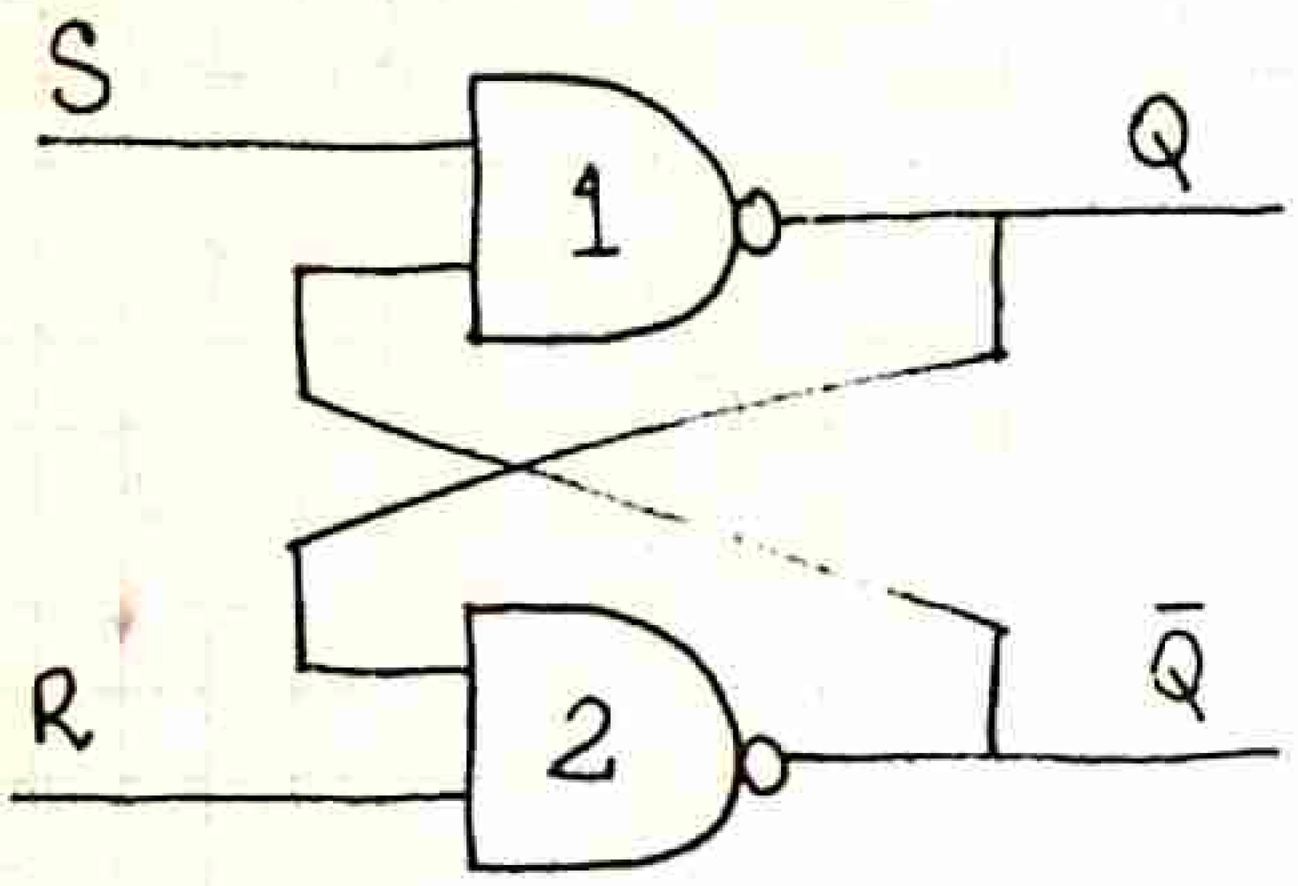
A FLIP-FLOP HAS ONLY 2 STABLE STATES

1. GATE 1 ON : GATE 2 OFF.
2. GATE 1 OFF : GATE 2 ON.

DURING THESE STATES THE FLIP FLOP IS CAPABLE OF STORING A PIECE OF BINARY INFORMATION.

IN ONE OF THE STATES THE FLIP FLOP WILL STORE A BINARY 1 & IN THE OTHER STATE IT WILL STORE A BINARY 0. IT WILL HOLD THIS INFORMATION AS LONG AS THE POWER IS APPLIED TO THE AND GATES.

THE SIMPLEST TYPE OF FLIP FLOP IS THE LATCH. WE WILL DRAW THE LATCH WITH THE CROSS-COUPLING LINES TO SHOW HOW BOTH GATES ARE INTER-CONNECTED.



A SIMPLE LATCH.

THE LATCH HAS 2 INPUTS, LABELED S FOR SET & R FOR RESET. THE S INPUT IS USED TO SET THE LATCH & THUS STORE A BINARY 1, IN THE FLIP FLOP.

THE R INPUT IS USED TO RESET THE FLIP FLOP & THUS STORE A BINARY 0.

THESE LETTERS GIVE THE FLIP FLOP ITS MOST COMMON NAME:

RS FLIP FLOP [FOR RESET-SET FLIP FLOP]

THE OUTPUT ARE LABELED Q & $\overline{Q}$ (Q-BAR OR Q COMPLEMENT) AND REFER TO THE NORMAL OUTPUT (Q) AND THE COMPLEMENT OUTPUT ($\overline{Q}$)

THE VALUE STORED IN THE FLIP FLOP IS GIVEN BY THE READING ON THE NORMAL OUTPUT.

THE $\overline{Q}$ IS AN OUT-OF-PHASE VALUE & IS USED WHEN DESIGNING COMPLEX LOGIC DIAGRAMS. IT CAN, HOWEVER, BE USED TO DETERMINE THE STATE OF THE FLIP FLOP:

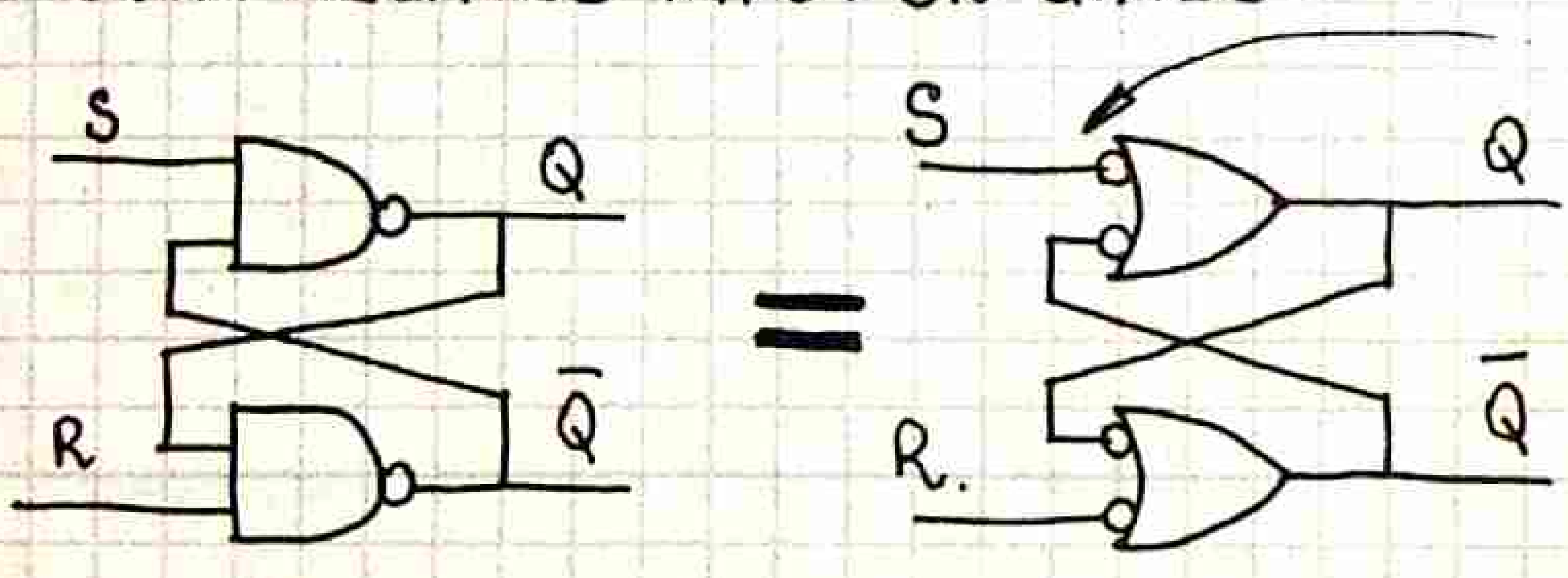
IF $\overline{Q}$ IS HIGH (BINARY 1) THE FLIP FLOP IS RESET.
IF $\overline{Q}$ IS LOW (BINARY 0) THE FLIP FLOP IS SET.

THE MAIN FEATURE OF A NAND GATE IS THIS: THE OUTPUT GOES LOW WHEN BOTH INPUTS ARE HIGH. THIS MEANS THE LOW IS THE CONTROLLING INFLUENCE FOR A NAND GATE FLIP FLOP AND EACH GATE REQUIRES A LOW FOR ITS STATE TO CHANGE.

NAND

INPUTS		OUTPUT
0	0	1
0	1	1
1	0	1
1	1	0

FROM THE ARTICLE ON "BUBBLES" WE CAN REDRAW THE NAND GATE FLIP FLOP AS DUAL NEGATED INPUT OR GATES:

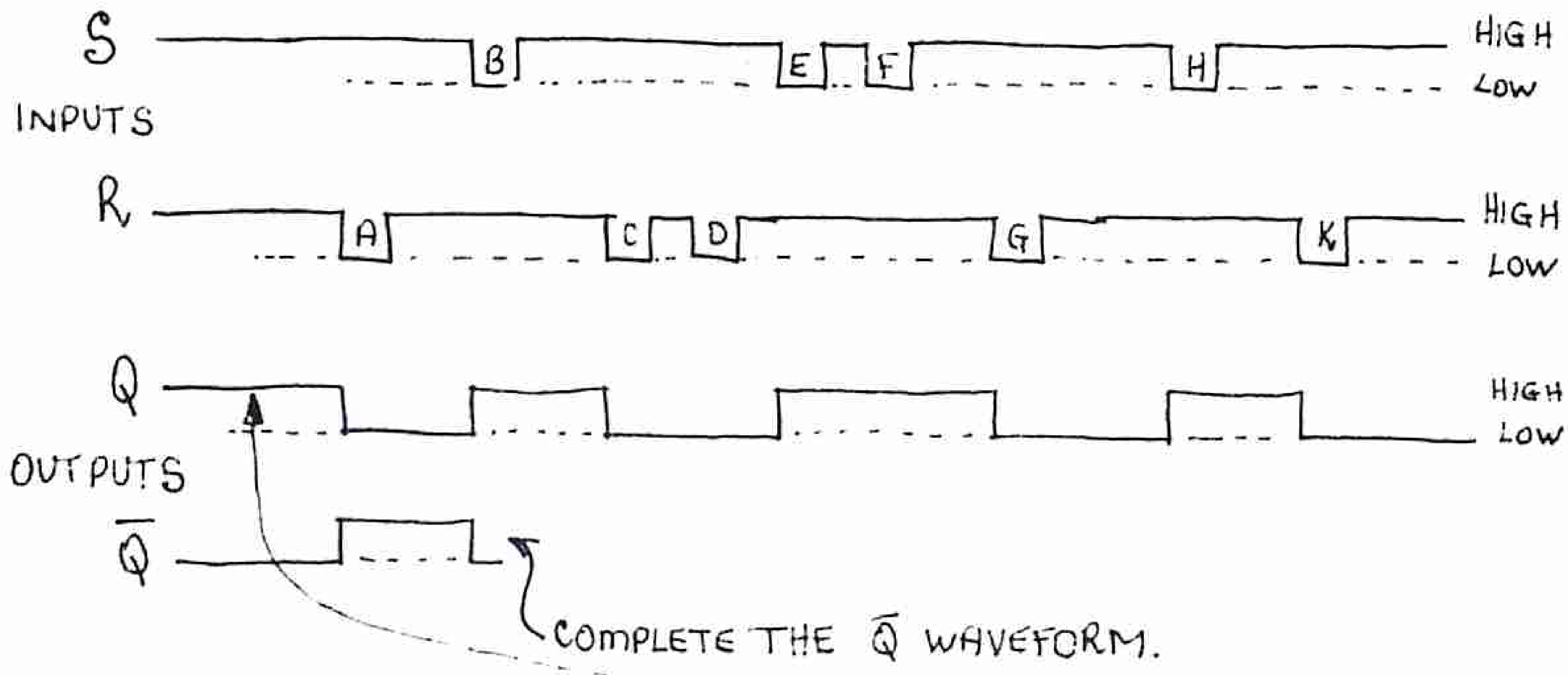


THIS BUBBLE INDICATES THE GATE IS ACTIVE WHEN THE INPUT LINE IS LOW.

THIS WILL HIGHLIGHT THE FACT THAT A LOW IS REQUIRED TO CHANGE THE STATE OF A NAND GATE R-S FLIP FLOP

A LOW ON THE S INPUT WILL SET THE FLIP FLOP.
A HIGH ON THE S INPUT WILL DO NOTHING AS THE INPUT IS ONLY ACTIVATED BY A LOW SIGNAL (THAT IS A SIGNAL CHANGING FROM A HIGH VALUE TO A LOW VALUE)
A LOW ON THE RESET LINE WILL RESET THE FLIP FLOP & THIS LINE CANNOT SET THE FLIP FLOP BY GOING HIGH. ONLY LOWS ACTIVATE THE FLIP FLOP

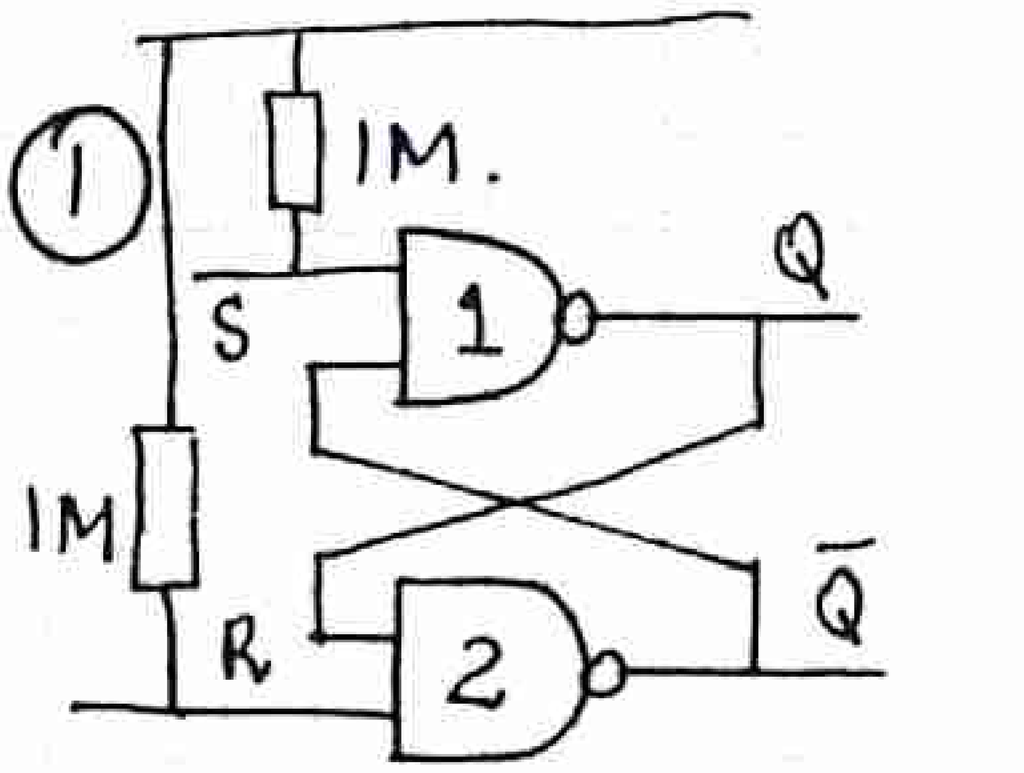
THE FOLLOWING DIAGRAM SHOWS HOW THE FLIP FLOP RESPONDS TO THE INPUT PULSES: THE FLIP FLOP IS INITIALLY SET (IN OTHER WORDS IT IS PRODUCING A HIGH ON OUTPUT Q).



THE FLIPFLOP IS INITIALLY SET. THE LOW ON THE RESET LINE AT A RESETS THE F-F. THIS CONDITION CONTINUES UNTIL THE SET PULSE B IS RECEIVED TO SET THE F-F. RESET PULSE C RESETS THE F-F & PULSE D HAS NO EFFECT AS IT IS ALSO A RESET PULSE. PULSE E SETS THE F-F & PULSE F IS ALSO A SET PULSE AND DOES NOT ALTER THE F-F. PULSE G RESETS THE F-F AND PULSE H SETS IT AGAIN. FINALLY PUSE K RESETS F-F.

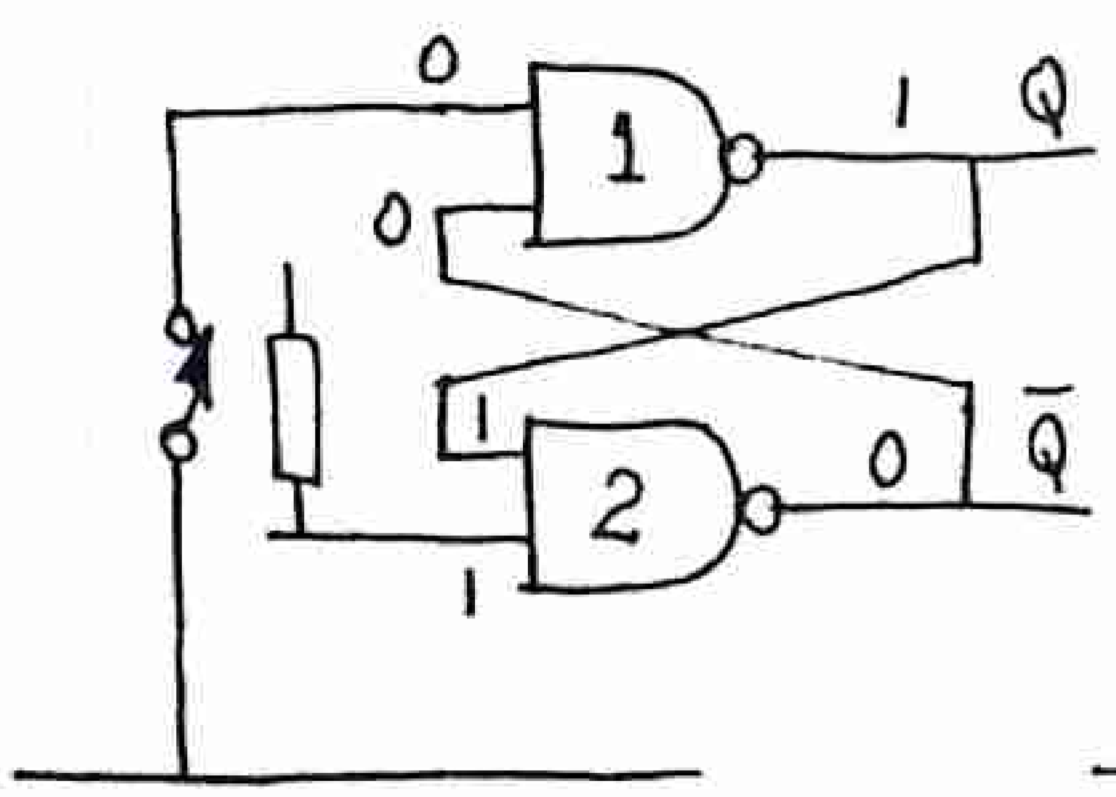
THIS SIMPLE TYPE OF FLIP FLOP SUFFERS FROM ONE TYPE OF PROBLEM WHICH WE HAVE NOT SHOWN IN THE ABOVE DIAGRAM.

FOLLOW THROUGH THESE DIAGRAMS AND SEE THIS LIMITATION:



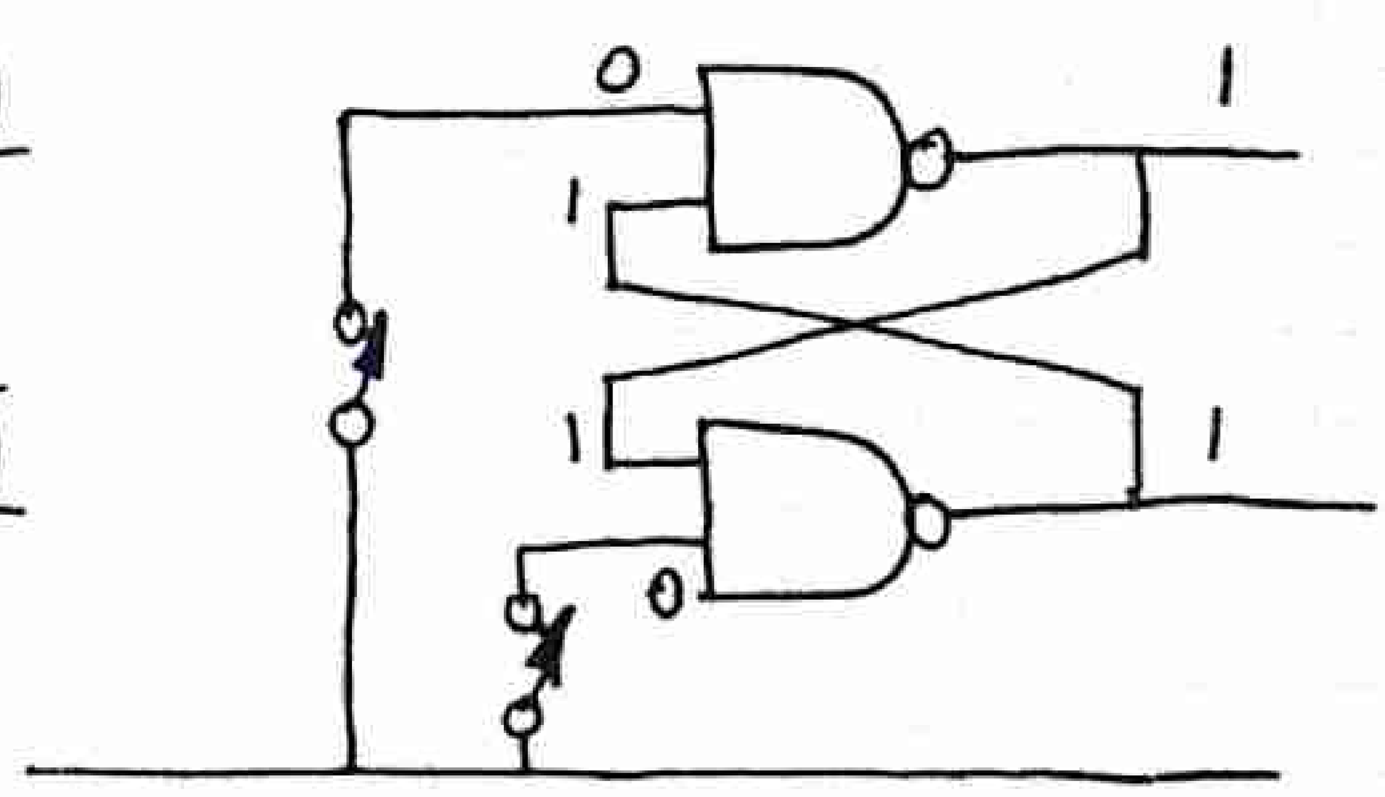
THE INPUTS ARE NORMALLY HIGH & WE HAVE ASSUMED THE FLIP FLOP IS IN THE RESET STATE

2 TAKE THE SET INPUT LOW



FLIP FLOP 1 IS SET AND THE NOMAL OUTPUT IS HIGH.

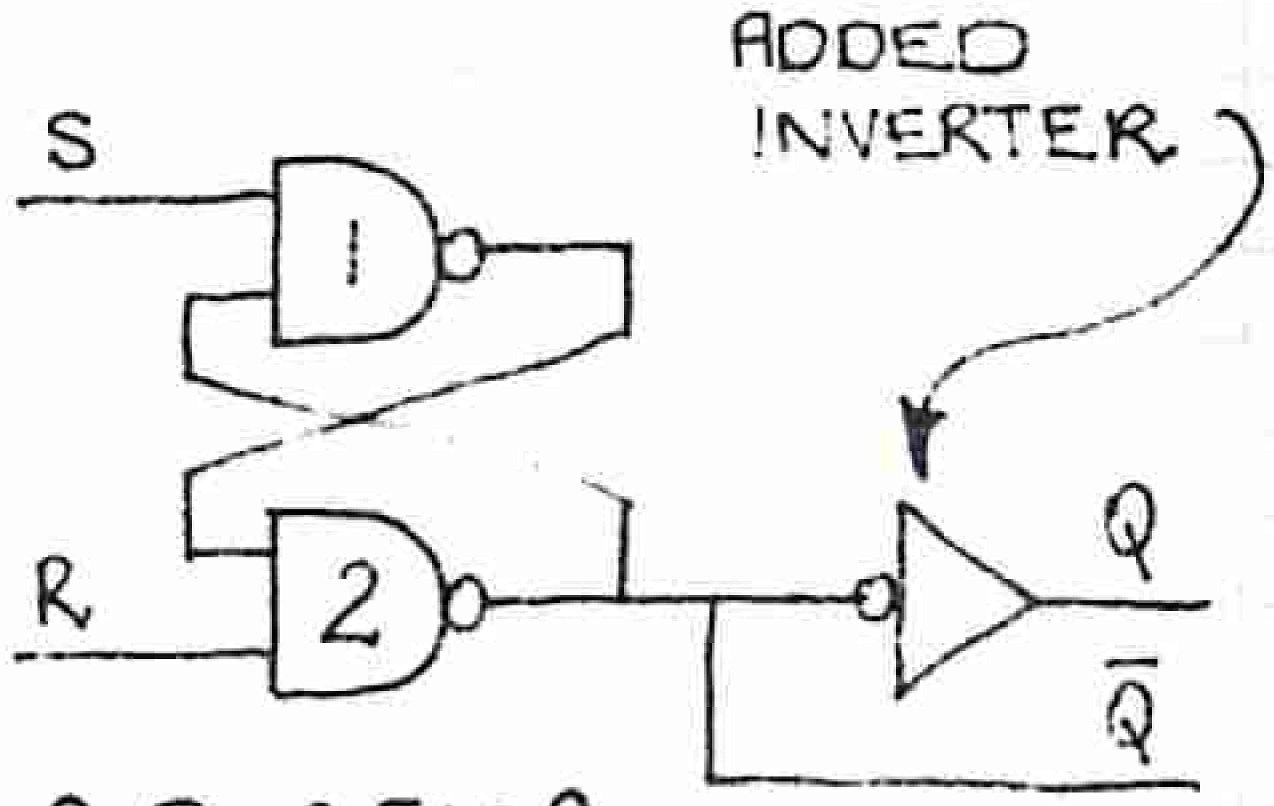
3 TAKE BOTH SET & RESET INPUTS LOW



BOTH OUTPUTS GO HIGH!

IF BOTH INPUTS ARE TAKEN LOW, THE OUTPUTS PRODUCE A HIGHLY UNDESIRABLE CONDITION. THEY BOTH BECOME HIGH! WE LOSE OUR FEATURE OF Q & $\overline{Q}$ BEING COMPLEMENTARY & WE MUST INTRODUCE MEASURES TO AVOID THIS SITUATION.

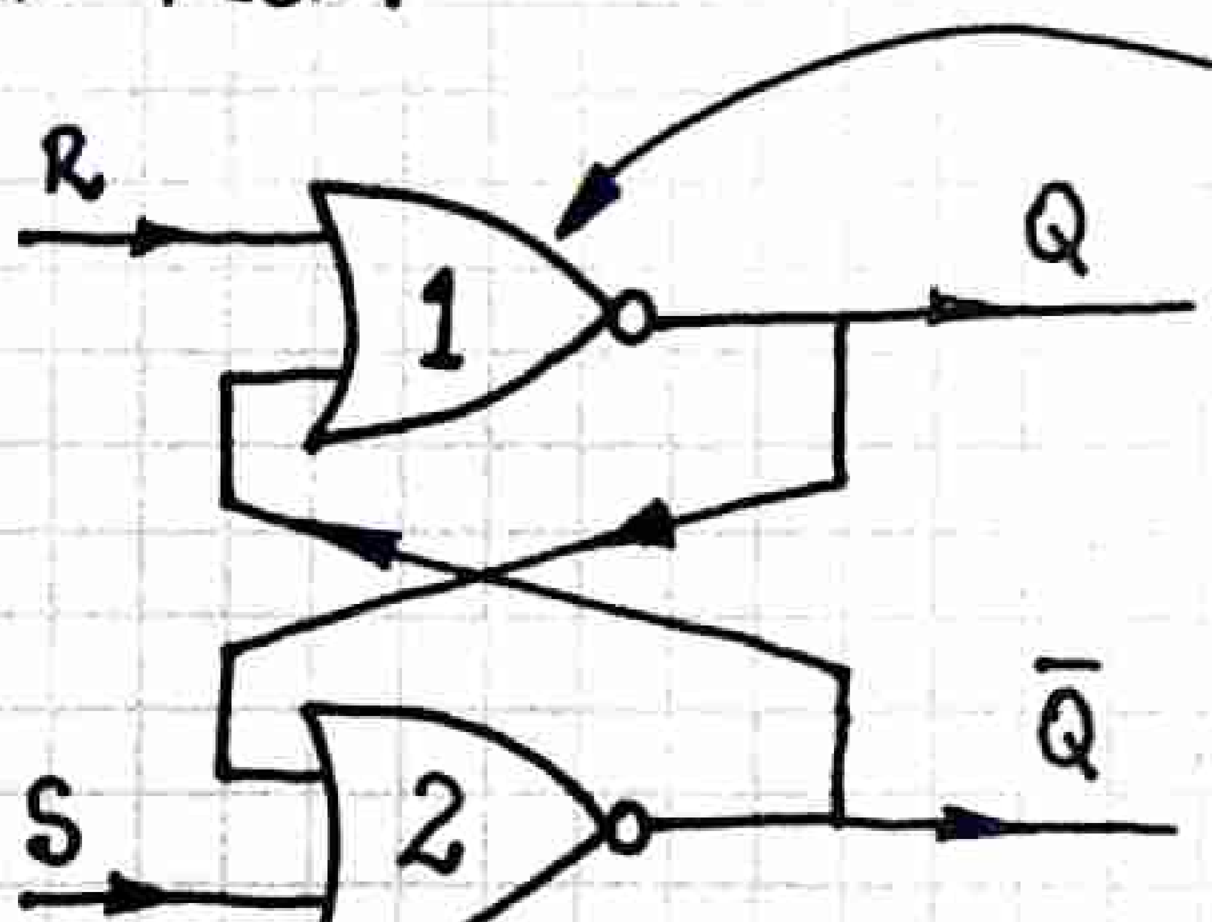
THIS THIRD CONDITION IS CALLED "LIMBO" & ONE METHOD OF ELIMINATING IT IS TO PLACE AN INVERTER ON ONE OF THE OUTPUTS. THIS ENSURES THE Q & $\overline{Q}$ WILL BE COMPLEMENTARY EVEN WHEN BOTH INPUTS GO LOW. THE INVERTER CAN BE ADDED TO EITHER Q OR $\overline{Q}$ OUTPUT WITH THE SAME RESULT.



SUMMARY OF THE NAND LATCH – ALSO R-S FLIP FLOP

	INPUTS		OUTPUTS		
	SET	RESET	Q	$\overline{Q}$	
	0	0	1	1	PROHIBITED OR LIMBO CONDITION
THIS IS A LOW TO THE SET LINE	0	1	1	0	SET CONDITION
THIS IS A LOW TO THE RESET LINE	1	0	0	1	RESET CONDITION
	1	1	X	$\overline{X}$	THE X MEANS EITHER SET OR RESET & THE $\overline{X}$ MEANS RESET OR SET.

A SECOND TYPE OF R-S FLIP FLOP CAN BE CONSTRUCTED WITH NOR GATES. IT OPERATES IN A SIMILAR MANNER EXCEPT THAT IT REQUIRES A HIGH ON THE SET OR RESET LINES TO CHANGE THE STATE OF THE FLIP FLOP.



A NOR-GATE LATCH.

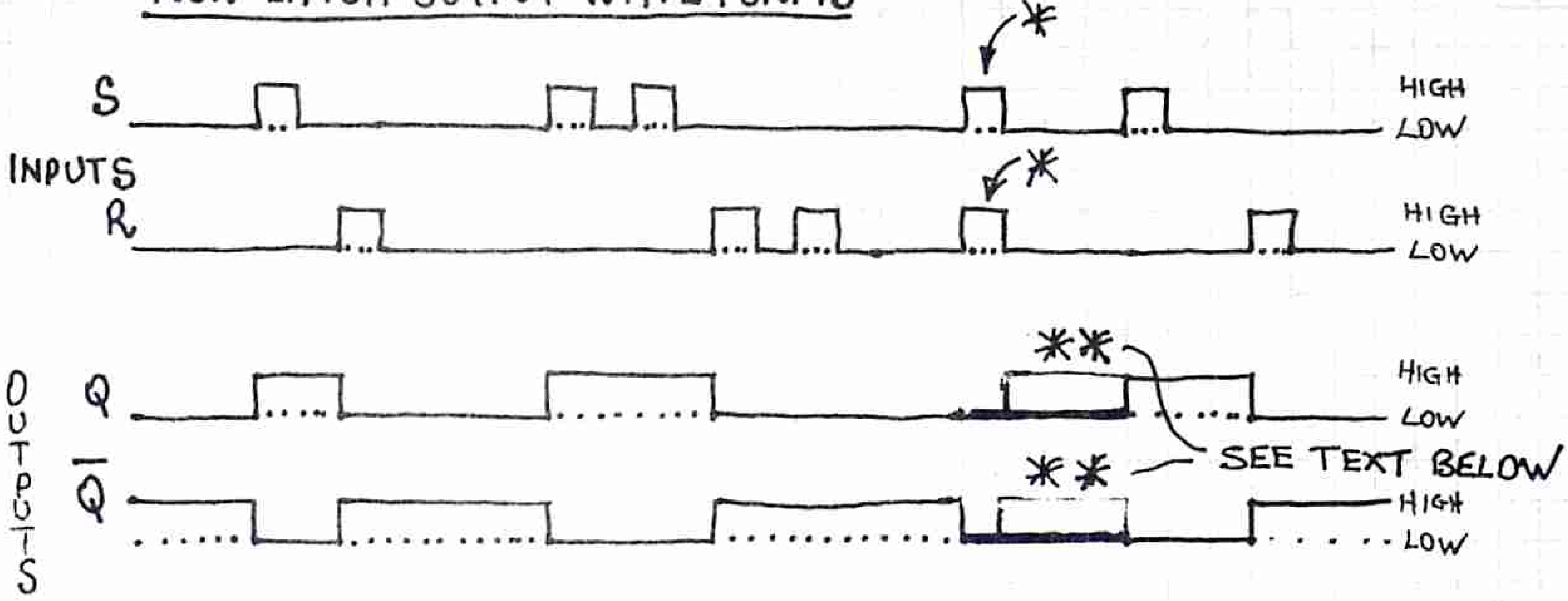
NOR

INPUTS		OUTPUT
0	0	1
0	1	0
1	0	0
1	1	0

TRUTH TABLE – NOR GATE

THE FIRST POINT TO NOTE IS THE LABELING OF THE R & S LINES. THEY ARE IN REVERSE TO THE NAND FLIP FLOP. IN THE NOR LATCH A HIGH ON THE SET LINE WILL PRODUCE A HIGH ON THE Q OUTPUT. A HIGH ON THE RESET LINE WILL PRODUCE A LOW ON THE Q OUTPUT.

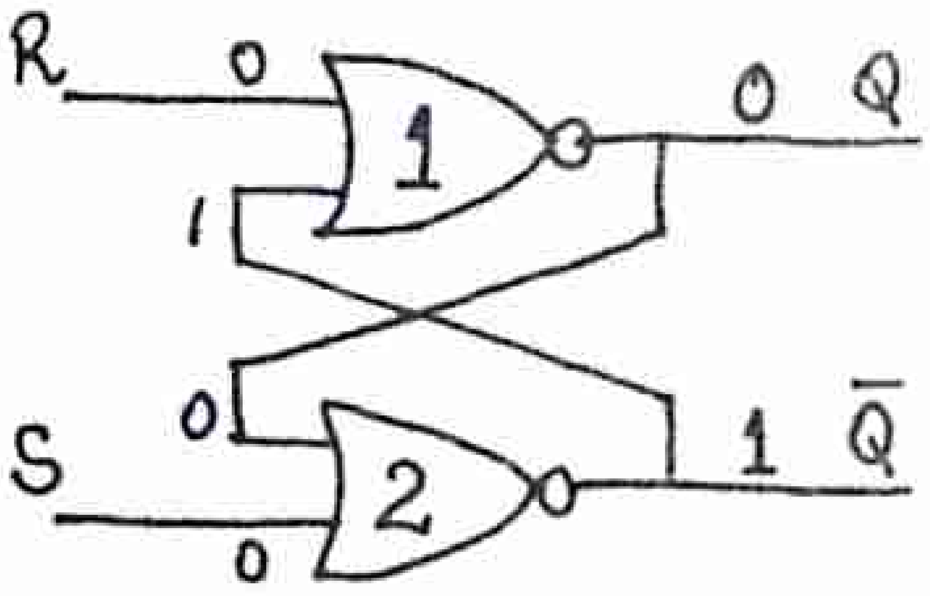
NOR LATCH OUTPUT WAVEFORMS



* THE TWO PULSES OCCUR ON THE SET & RESET LINES AT THE SAME TIME. THE RESULT IS THE Q & $\overline{Q}$ OUTPUTS GO LOW AT THE SAME TIME TO PRODUCE THE UNWANTED RESULT WE DISCUSSED PREVIOUSLY. THIS IS THE ONE LIMITATION OF THE R-S LATCH & INTRODUCES THE NEED FOR AN IMPROVED LATCH.

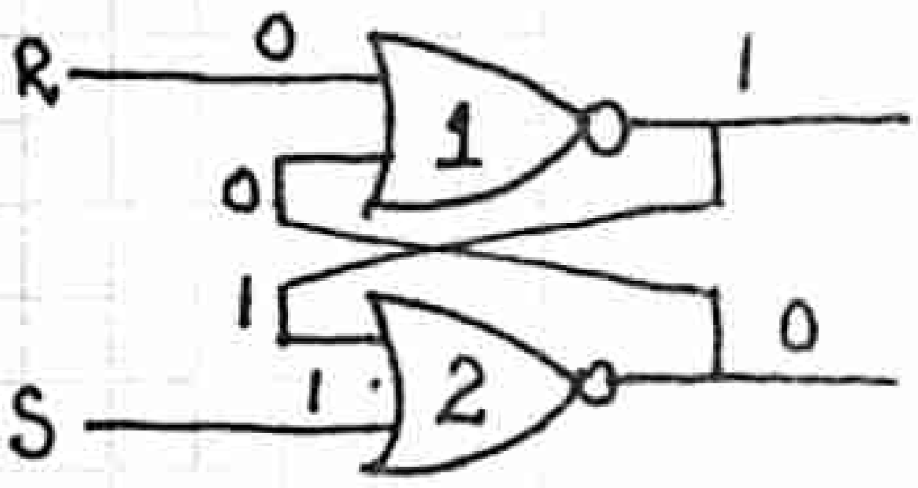
** THE UNWANTED STATE ABOVE IS SHOWN IN THE OUTPUT BY **. AFTER THE SET & RESET PULSES GO LOW TOGETHER, A RACE OCCURS IN THE FLIP FLOP FOR THE FIRST OUTPUT TO GO HIGH. THIS COULD BE EITHER THE NORMAL OR COMPLEMENTARY OUTPUT. THIS UNKNOWN STATE CAN BE CLEARED UP BY PROVIDING A SET PULSE OR RESET PULSE, AFTER WHICH THE LATCH FUNCTIONS PREDICTABLY.

(1) THE NOR LATCH CAN START UP IN ANY STATE WHEN BOTH INPUTS ARE LOW. — ASSUME IT STARTS UP RESET.

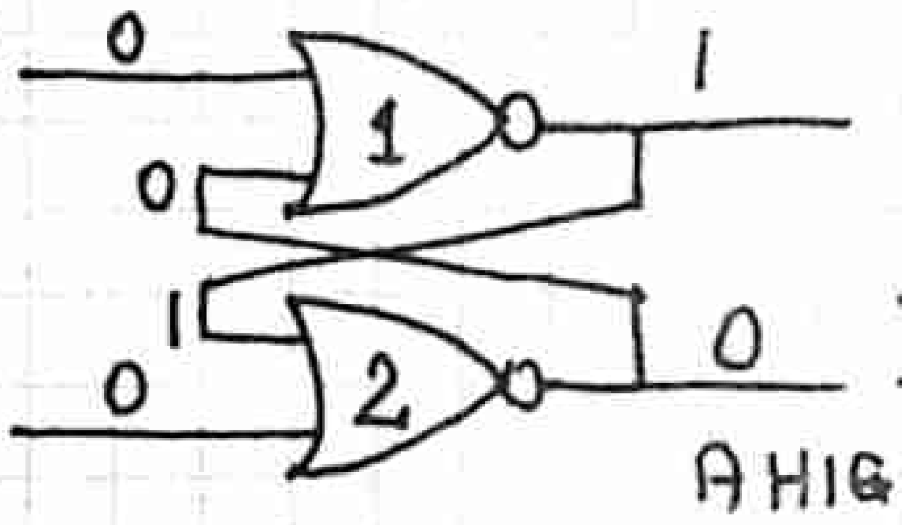


THE "0's" & "1's" ARE HIGHS & LOWS.

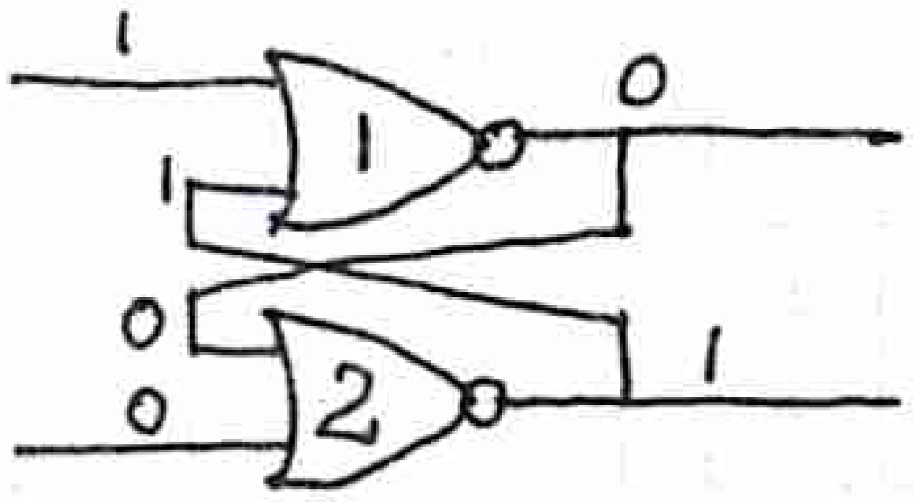
(2) A SET PULSE WILL SET THE FLIP FLOP:



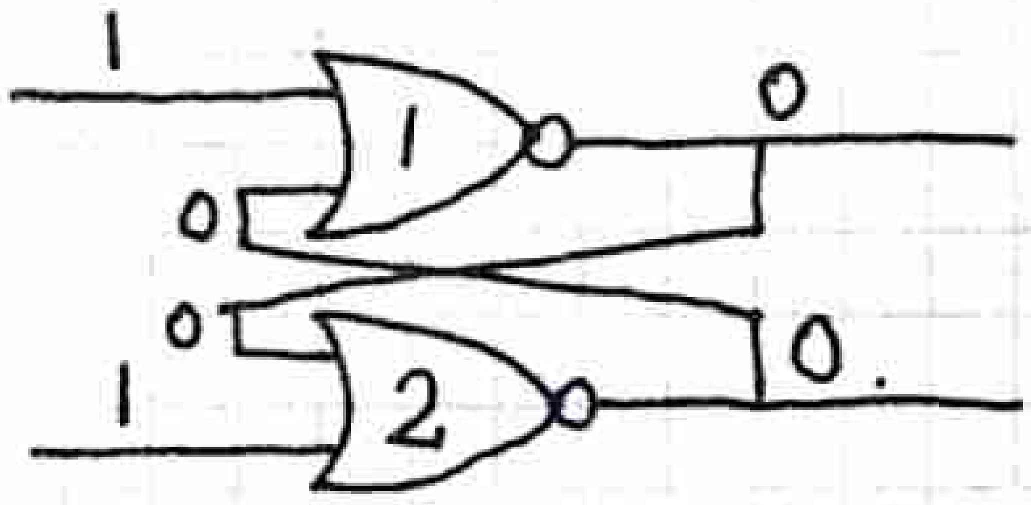
(3) REMOVING THE SET PULSE WILL NOT AFFECT THE F-F BECAUSE THE FEEDBACK FROM THE Q OUTPUT TAKES OVER WITH A HIGH INTO F-F #2.



(4) A RESET PULSE (HIGH) WILL RESET THE FLIP FLOP.

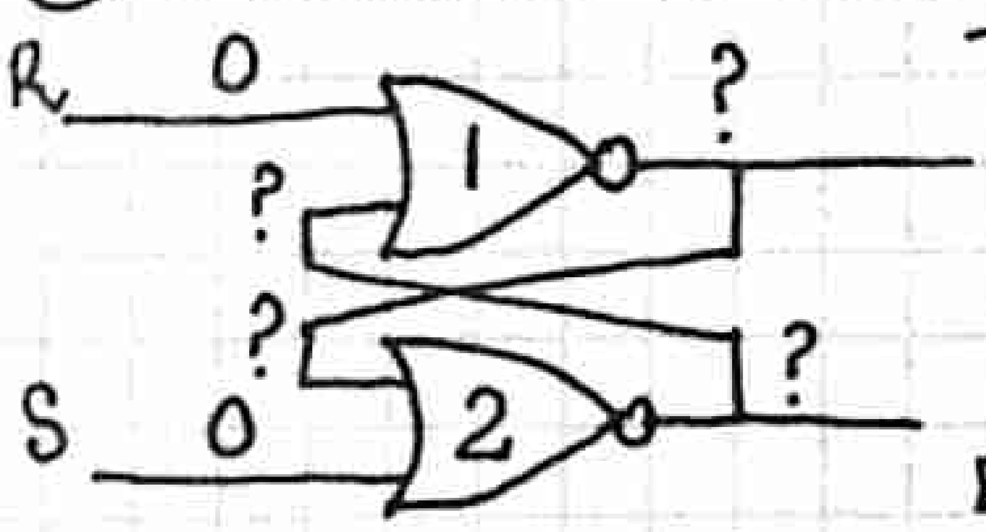


(5) IF THE RESET PULSE REMAINS & A SET PULSE IS APPLIED:



BOTH OUTPUTS GO LOW!

(6) IF BOTH S&R PULSES ARE REMOVED TOGETHER, A RACE EXISTS FOR THE FIRST OUTPUT TO GO HIGH — & EITHER CAN GO HIGH.



SUMMARY OF THE NOR LATCH (R-S FLIP FLOP)

INPUTS		OUT PUTS		
SET	RESET	Q	$\overline{Q}$	
0	0	X	$\overline{X}$	— EITHER SET OR RESET
0	1	0	1	— RESET CONDITION
1	0	1	0	— SET CONDITION
1	1	0	0	— UNDESIRABLE CONDITION.

A HIGH ON THE RESET LINE

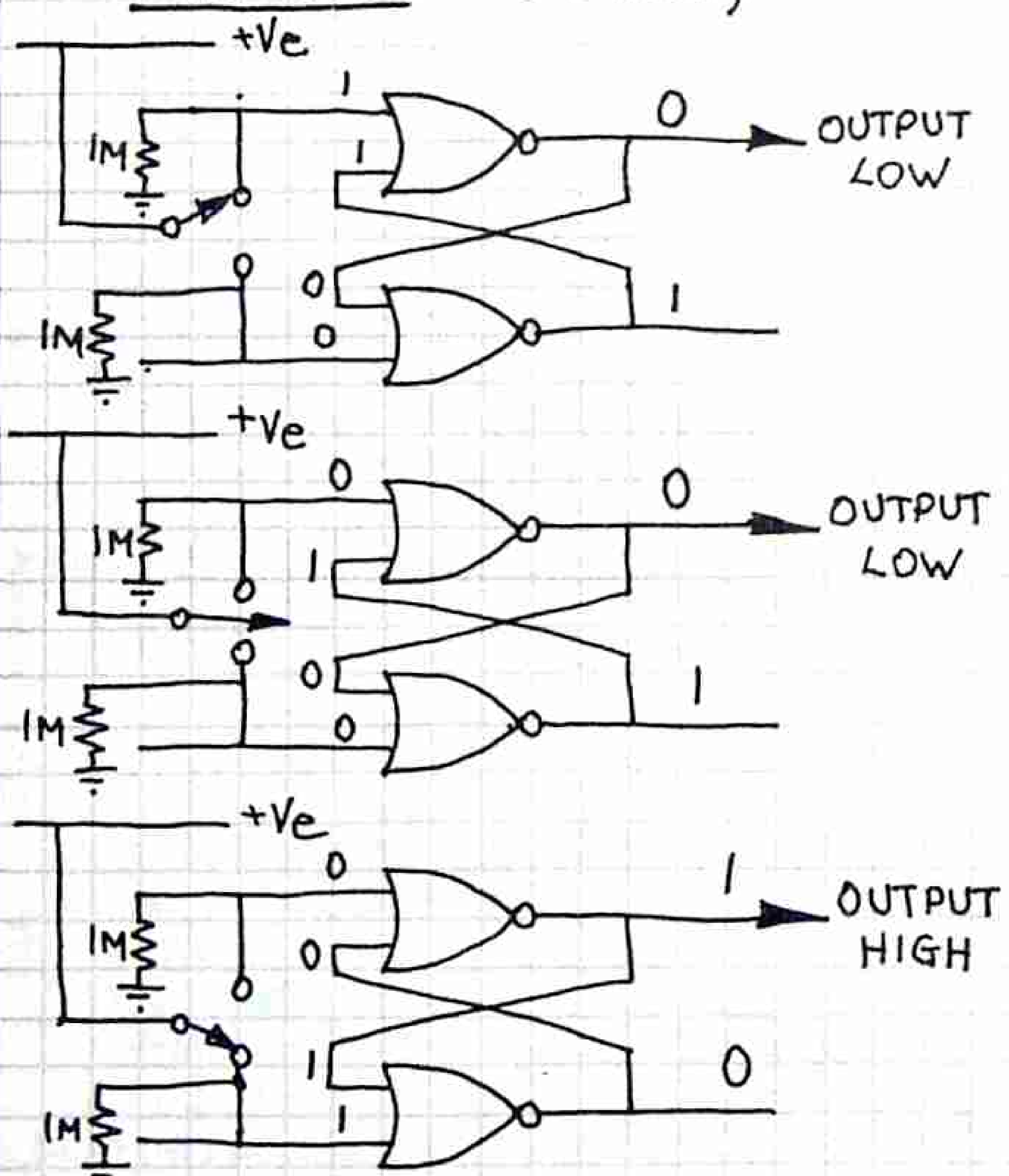
A HIGH ON THE SET LINE.

DEBOUNCING A SWITCH

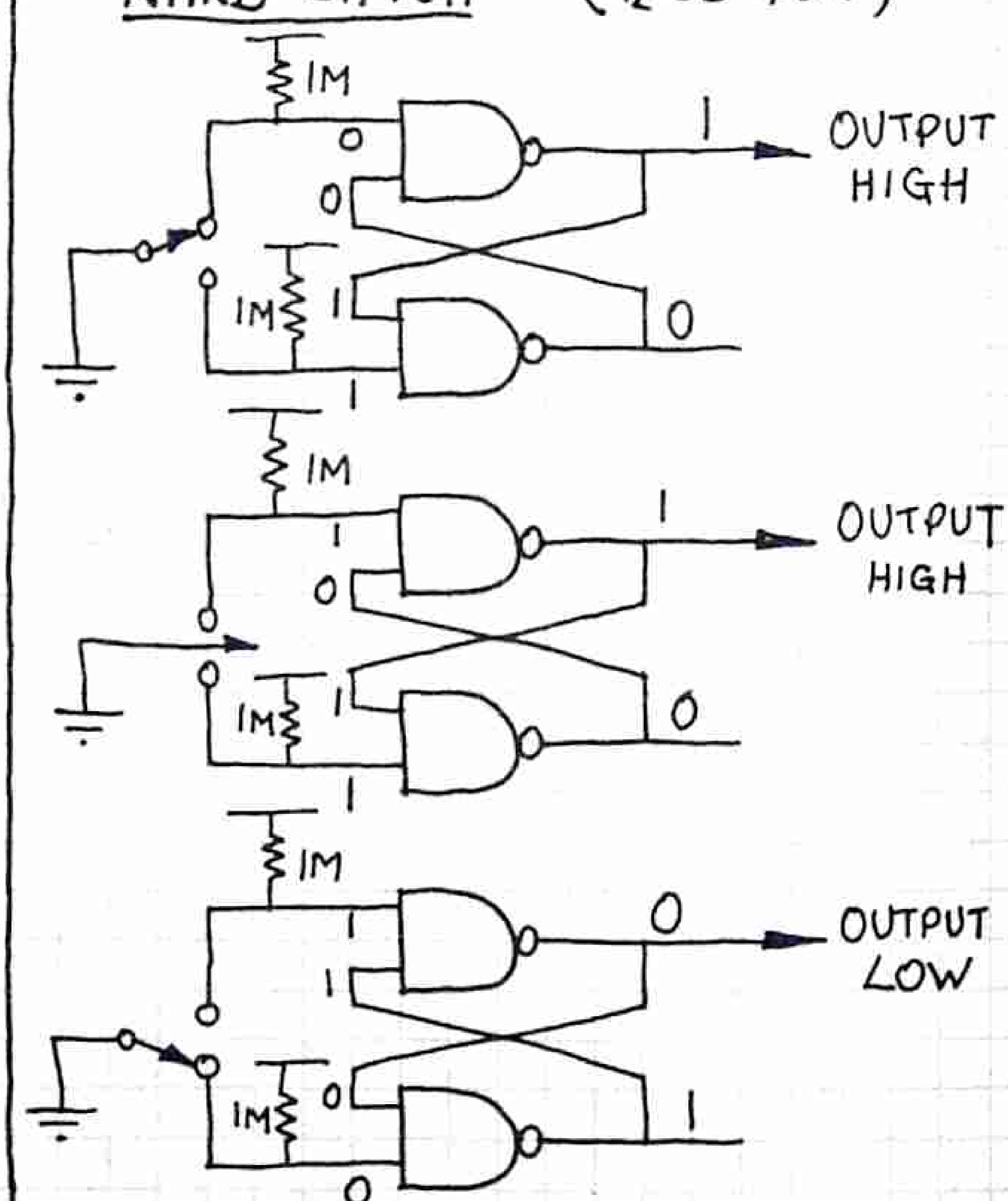
THE NAND & NOR LATCHES CAN BE SUCCESSFULLY USED AS DEBOUNCE CIRCUITS FOR MECHANICAL SWITCHES. THE IMPORTANT FEATURE OF THESE CIRCUITS IS THE FACT THAT THEY DO NOT CHANGE STATE WHEN THE SWITCH IS OPENED BUT ONLY ON CLOSING THE CONTACTS.

NEITHER DOES IT CHANGE STATE ON RE-CLOSING THE CONTACTS BACK TO THE SAME GATE.

NOR LATCH — (CD4001)



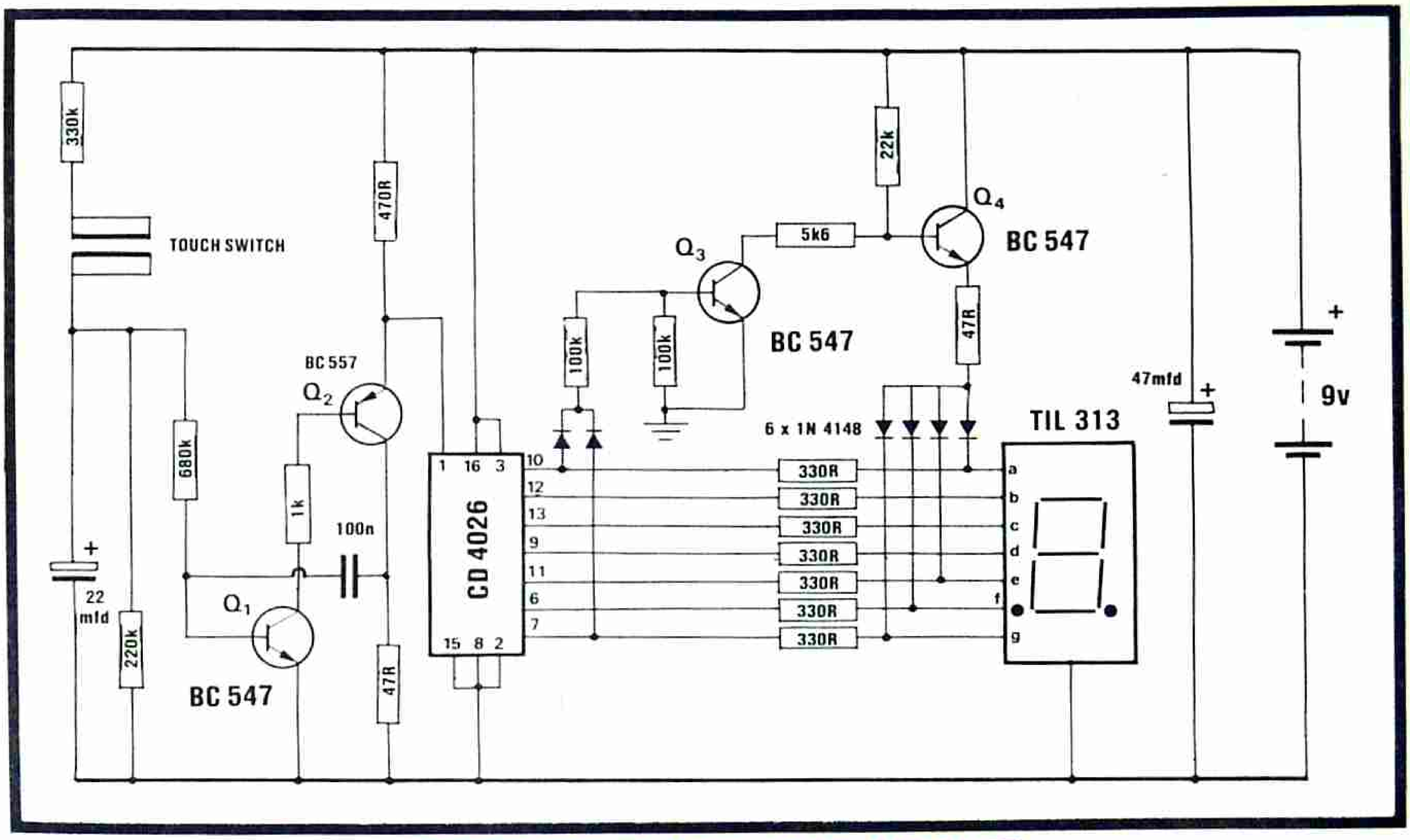
NAND LATCH — (½ CD 4011)



A2345678910JQKA2345678910JQK-A2345678910JQKA2345678910JQK-A234567

BLACK JACK

$9.30
COMPLETE



COMPLETE BLACK JACK CIRCUIT

Along with our previous range of party games and puzzles, we add our latest project: BLACK JACK.

Designed by our new member of staff, Paul, this project employs a part digital design and part linear design.

By this we mean the circuit has a transistor or two to 'get around' a digital limitation.

As you know, the CD 4026 is a 7-segment display driver and is capable of producing the numbers 0-9.

For our arrangement, we require the '1' to be replaced with the letter 'A'. And so to achieve this we needed to add a transistor detector and transistor segment driver.

This makes an interesting circuit design and shows the advantage of thinking 'transistor' as well as digital.

The only slight problem you may encounter with this circuit is its voltage dependence. The circuit is designed for 9v operation and at this level all segments illuminate with equal brightness. If the voltage is decreased to below 6v, the letter 'A' will not have equal brightness in all segments.

Our party games receive a lot of interest from constructors who like to bring out a new game or puzzle at club meetings and the like. Sometimes they modify old game to give them a new slant. Like the Rouled game in which one reader used RED and GREEN LEDs in the circle. Or the Digi Chaser which was increased to 1k of memory by Mr Ohlson and coded with his CW call sign.

The best is the Cube Puzzle which was made impossible by leaving out a link!

Maybe you will be able to think of a variation for this game. If so, let us know.

I don't know much about card games and so I'll pass you over to Paul to describe his project.

PARTS LIST

2 - 47R ¼ watt
7 - 330R
1 - 470R
1 - 1k
1 - 5k6
1 - 22k
2 - 100k
1 - 220k
1 - 330k
1 - 680k

1 - 100n 50v greencap
1 - 22mfd 16v PC electro
1 - 47mfd 16v PC electro

6 - 1N 4148 diodes

3 - BC 547 transistors
1 - BC 557 transistor

1 - TIL 313 display
1 - CD 4026 driver IC

1 - 16 pin IC socket
1 - battery snap

BLACK JACK PC board.

8910JQKA2345678910JQK-A2345678910JQKA2345678910JQK-A2345678910JQKA

BLACK JACK, or 21, as most people prefer to call it, is a game in which players have to get as close to 21 as possible without going over.

Although it is 90% chance, each player can increase his chance of winning by making a few simple decisions.

First, each player is dealt 2 cards. The object of the game is to reach 21 by adding up the value of the cards. The spot cards count as indicated and the COURT cards count as 10. The ACE has a dual value and can count as 1 or 11. This provides about the only mathematical manipulation in the game. The few remaining rules are simple.

If you have a low score, you can pick up another card. On the other hand, if you have a score of 14, you can elect to throw down your hand and pick up one, two or three cards.

A couple of small bonuses also apply to the game. If you are betting, and holding 5 cards with a total score less than 21, you receive a payout which is twice the amount of your bet. If you are holding three 7's, the payout is three times your bet.

This game has always been very popular and we have created an electronic version. The display represents the face of the cards and the following table will show how they are equated. The letter A appears for the ACE. The court cards and the 10-spot are all represented by the number 0 and unfortunately the game is not weighted to take the court cards into account in the same percentage as they appear in the pack. That's the only limitation of this version.

But it has two advantages.

It is much quicker and easier to use, especially for young children, and can accommodate any number of players.

Build the unit and you will soon understand how to play the game.

CONSTRUCTION

All the components are mounted on a printed circuit board.

The first item to solder onto the board is the IC socket. We suggest a socket because the chip is expensive and some constructors have damaged them in the Music Colour project.

You will find a small mark at one end of the socket to identify pin 1. Place this mark over the dot on the PC board to help in later identification.

Add the 18 resistors to the board in the positions shown in the overlay and cut the excess leads off, close to the board. Keep some of the leads for the jumper links and TOUCH WIRES.

CARD:	DISPLAY	SCORE:
ACE	A	1or 11
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10		
JACK	0	10
QUEEN		
KING		

THE DIGITAL READOUT AND THE SCORE IT REPRESENTS.

The two electrolytics and 100n capacitor are added to the board, followed by the 6 diodes. These must be inserted around the correct way for them to function correctly. Double check them before continuing. Fit the jumper links and TOUCH WIRES.

The circuit requires 3 NPN and 1 PNP transistor. Any type of transistors will be suitable and we have suggested our standard BC series.

Insert the PNP transistor first, making sure it fits into the holes correctly. Then solder the 3 NPN transistors.

The display is not polarised and it is important to insert it around the correct way. The two decimal points can be seen through the diffusing screen and these must be placed downwards.

Solder the battery snap with red lead to positive and black lead to negative. Fit the 4026 IC into the socket with the dot on the chip covering the dot on the board.

The project is now ready for operation. Touch the two links marked TOUCH SWITCH and watch the display.

IF IT DOESN'T WORK

Since the circuit consists of 3 separate sections, the first stage is to find out which stage is at fault.

Obviously you will have to check the soldering, the positioning of all the components and the quality of the battery.

Don't forget the BC 557 transistor. It is a PNP type and must be positioned as Q2 in the circuit.

The display must be type TIL 313 by Texas Instruments. Other displays may work but must be checked first.

Here is our approach to fixing the project:

If the display fails to illuminate it may be the wrong type, inserted around the wrong way or the common line may not be connected.

To test the display, the collector and emitter of Q4 are bridged with a jumper lead. This should illuminate the letter F. If this does not occur, the 47R resistor may be the wrong value or the diodes may be the wrong way around. If the F appears, the display will be OK.

The next area to look at will be the 330R dropper resistors. With one lead lifted from the board, switch your multimeter to the ohms range and check the value.

Next, move to the 4026 IC. For this test, the input line (pin 1), must have all input pulses removed. This is done by removing Q2 from the board so that pin 1 becomes connected directly to the positive rail.

The chip will now be in a static condition and you can check the output pins.

With a multimeter switched to the 10v range, detect a HIGH on all or some of the following pins: 6, 7, 9, 10, 11 and 12. Take pin 10 for instance. When it is HIGH, segment should illuminate. If it doesn't, the 330R resistor may not be connected or it may be an incorrect value. The only other possibility is a cut in the PC track or a dry joint.

Follow this procedure through to segment g. If some of the outputs are not HIGH, place a jumper lead from the negative rail, and touch the other end onto pin 1 of the 4026 chip. This action will cause it to clock. Repeat this a few times and you will eventually clock the chip through its complete cycle and cause each of the output lines to go HIGH.

Under normal conditions no other components would be needed in the display circuit. The 4026 is capable of driving the display without the need for any dropper resistors, provided the voltage is kept to about 5 - 6v. It has its own internal current limiting. But since we need to produce the letter A whenever '1' normally appears on the screen, we need to introduce a lot of additional circuitry.

And the way it works is very clever.

The circuit is basically detecting 2, 3, 4, 5, 6, 7, 8, 9 or 0 appearing on the screen. It ASSUMES that the number 1 will be appearing if the above is not the case.

When the transistor circuit is operating, it puts an F on the display. This is combined with the letter 1 and results in the letter A appearing.

The detection circuit consists of 2 diodes forming an OR gate.

The function of the two 100k resistors is to keep the voltage on the base of Q3 below .6v when the circuit is not required.

The first 100k resistor also separates the base of Q3 from the diodes so that the drive-lines from the chip can rise above 1.2v and illuminate a segment on the display.

If the voltage-dividing circuit were not present, the natural voltage drop across the diode, combined with the slight voltage on the output of the driver chip, (when it is in the LOW condition), would turn the transistor ON slightly.

This would be sufficient to create a glow in the display, which is undesirable.

The 100k resistor to ground halves this voltage and prevents the transistor from turning ON.

The second transistor is an emitter-follower. This means the top of the 4 diodes (connected to the segments) are in-phase with the collector of the first transistor. If the collector is HIGH, the top of the diodes is HIGH and this will turn on the segments. When the collector is LOW, the voltage on the diodes is LOW and the segments will not be lit.

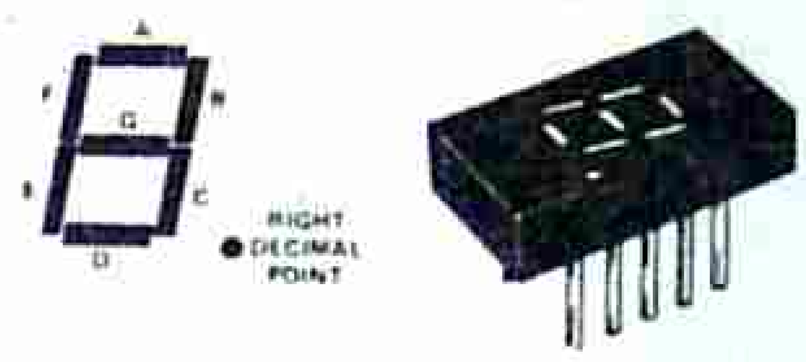
The first transistor is an inverting stage and we can describe how the circuit works in one simple sentence.

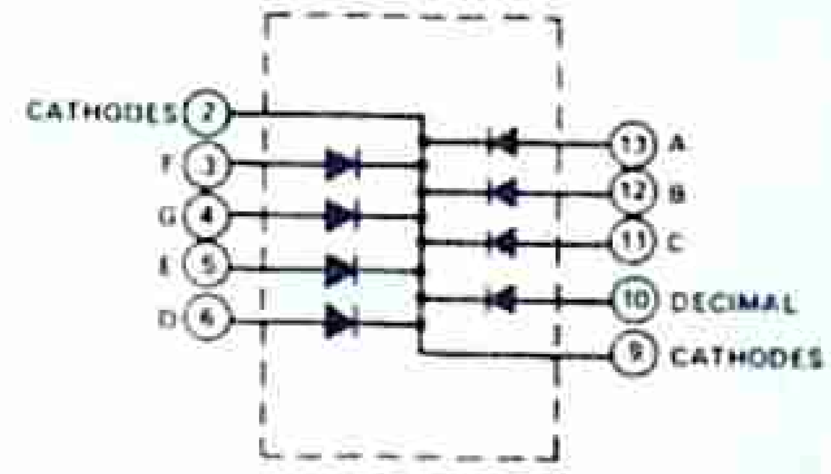
Only when both detecting diodes are LOW, will the collector of the first transistor be HIGH and thus pull the top of the 4 diodes HIGH and illuminate the letter F.

The 47R resistor is a current limiting resistor to adjust the brightness of the letter F to be equal to the number 1 and thus give an evenly-illuminated letter A.

The seven 330R resistors have two functions. Firstly they act as current limiting for the 4026 when driving the display from 9v and secondly they separate the input of the transistor section from the output.

THE TIL 313 DISPLAY

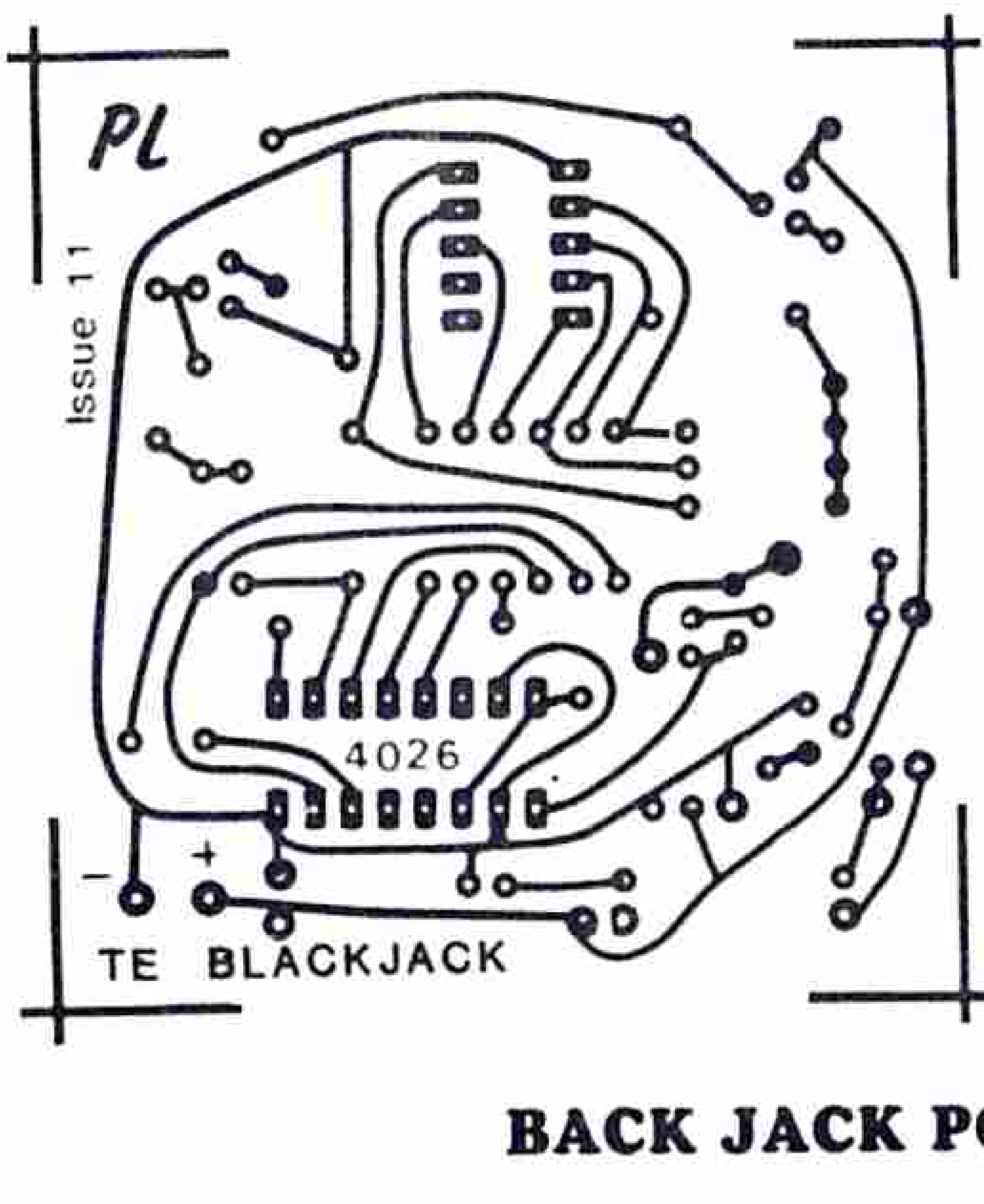


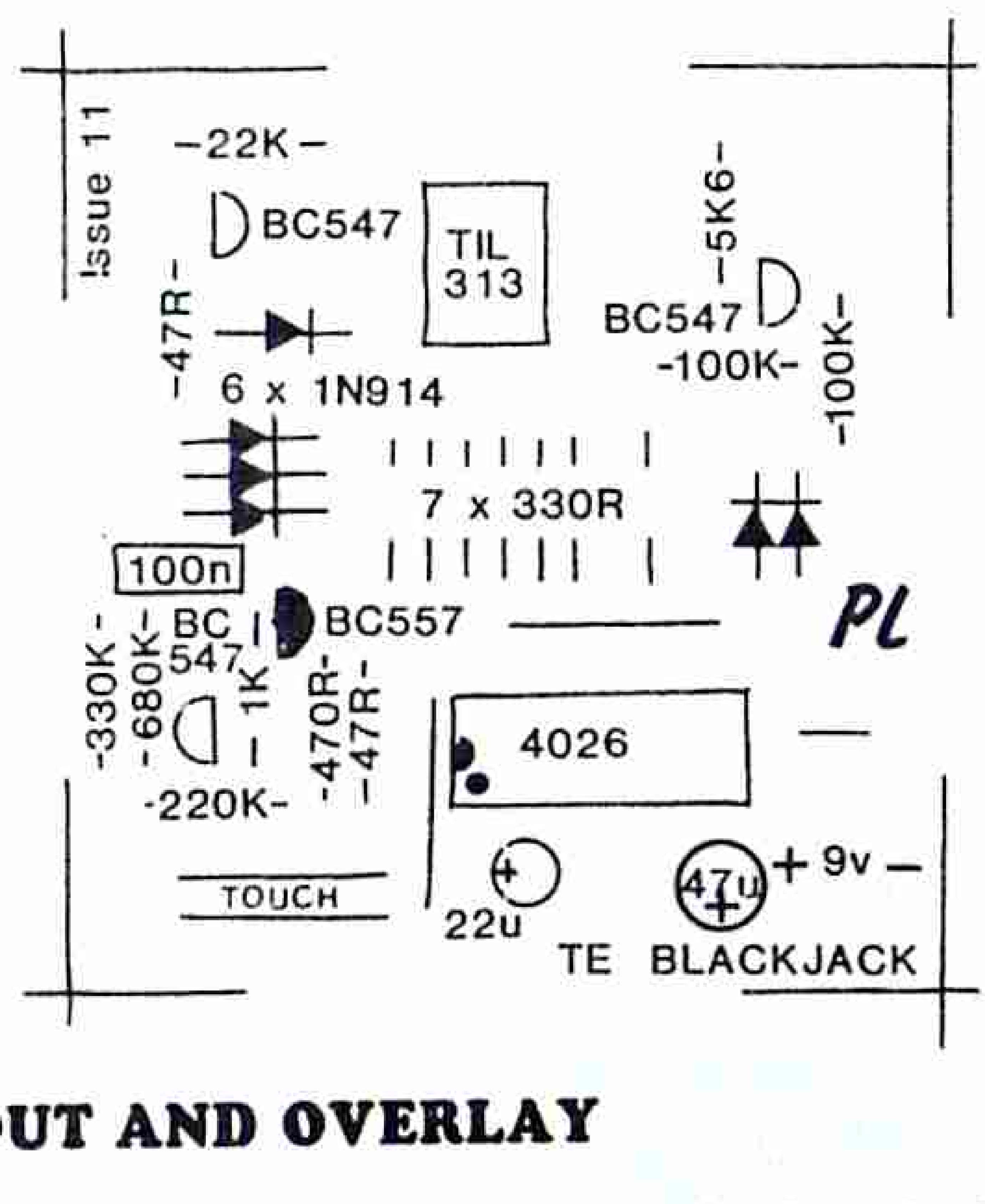


TIL 313 is a COMMON CATHODE display in which pins 1, 7, 8 and 14 are NOT on the unit. Only pins 2, 3, 4, 5, 6, 8, 9, 10, 11, 12 and 13 are present.

No other displays will work in the same set of holes as the TIL 313 as every display has a different pin-out arrangement.

You will have to drill 4 more holes and re-wire the PC board if you intend to use another display. Remember, only a common cathode display will work.





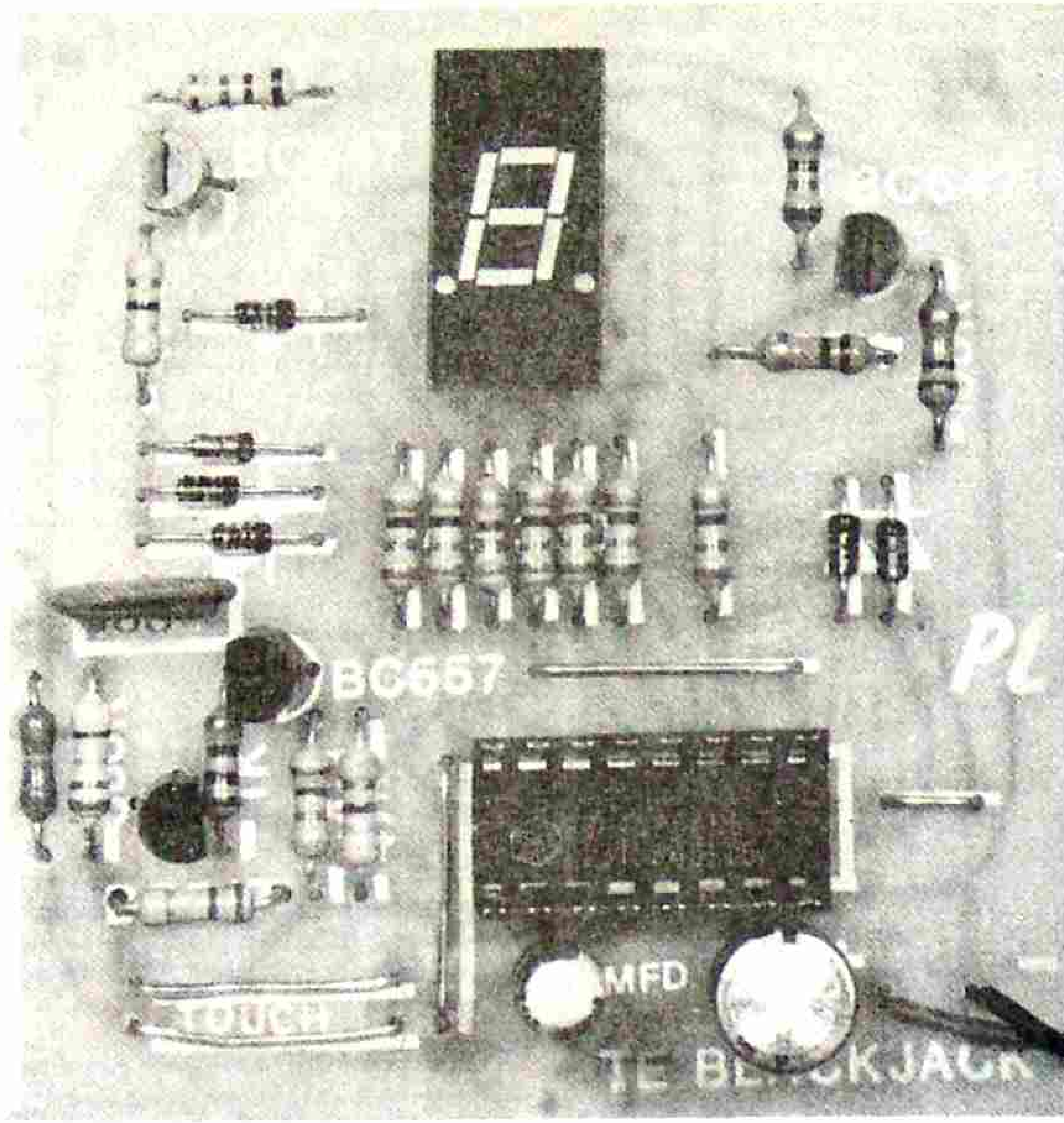
BACK JACK PC LAYOUT AND OVERLAY

If this part of the circuit is operating correctly, we will be able to manually clock the chip through its full cycle and watch the display change. You cannot prevent the chip jumping a few counts every time pin 1 is touched as the chip will pick up switch-noise. Even so, we will be able to see all the numbers.

If the number 1 appears on the screen, the transistor section will be faulty as it will not be generating the letter F.

To operate the transistor section manually, put a 10k on jumper leads and connect one end to the base of Q3. Take the other end LOW and the display should show the letter F as well as whatever number is being generated by the 4026. Take the base HIGH and the F will not show on the display. If this result does not occur, check the voltage on the collector of Q3 and repeat the operation.

If the collector voltage does not alter, Q3 will be at fault. If the voltage changes, place the probe of your multimeter on the base of Q4. It should swing from nearly full rail voltage to about 2 - 3v. If it does not hve this large swing, Q4 will be faulty, (as we have ascertained that the collector of Q3 has a large swing).



A CLOSE-UP VIEW OF THE COMPONENTS

Note the neat placing of the components. Use only ¼watt resistors and small capacitors to keep everything in 'balance'. You can use molex pins to create a socket for the display.

Finally, if the emitter of Q4 does not closely follow the base, the transistor itself will be faulty.

The last section to trouble-shoot is the oscillator.

Re-connect the BC 557 transistor and place a LED in series with the 470R resistor (in the collector circuit). This is done by removing one end of the 470R and connecting the light emitting diode with the cathode facing towards the negative terminal of the battery.

To check the oscillator it is necessary to test the two transistors in a DC state. This requires removing the 100n capacitor. The circuit now becomes a 2 transistor DC coupled amplifier. Touch the TOUCH SWITCH and the LED will gradually come on. If this does not occur, you will have to test each transistor individually.

A 4k7 resistor placed between collector and emitter of Q1 will turn the LED on. If this does not occur, Q2 may be faulty. If the LED comes on, the problem may lie with Q1. To check the DC operation of Q1, connect a 10k resistor between its base and the positive rail. If this does not turn the LED on, Q1 may be faulty.

If these tests are successful, the only remaining area is the feedback section. This is more difficult to test as the AC voltages are small and the pulse durations are very short.

To turn the oscillator on, re-fit the 100n capacitor and measure the voltage on the base of Q1 while shorting across the TOUCH SWITCH. The voltage range for the base is .55v to .65v. Above this, the oscillator jams. The function of the 680k resistor is to separate the voltage on the 22mfd electrolytic from the base and allow the oscillator to function.

To test the oscillator, firstly discharge the 22mfd electrolytic by shorting across its terminals.
Bridging across the TOUCH SWITCH will turn on the oscillator and increase its speed until it jams. Repeat this operation but use your finger across the switch. If it fails to start-up, your finger may have a very high resistance and fail to charge the electrolytic.

If you are careful, your BLACK JACK will work as soon as the battery is connected but if you encounter a 'bug', let's hope you find it and get it working quickly.

Black Jack is an ideal project for parties. It can be passed around more conveniently than a pack of cards.

We hope you build it and get as much pleasure as we did.

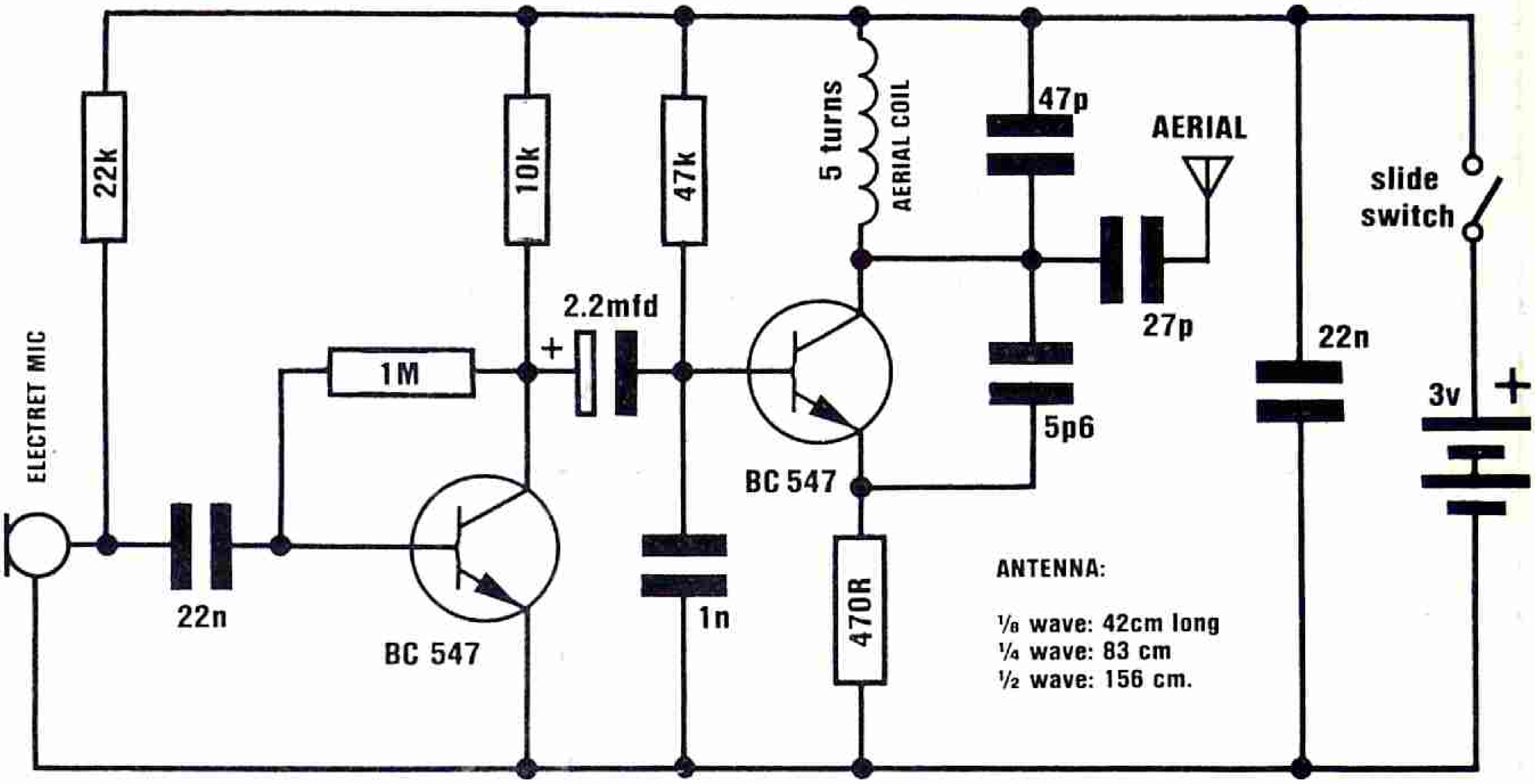
Paul.

PARTS $4.40

FM BUG

$6.00 COMPLETE

No Bugs with this BUG. A guaranteed performer using readily available parts.



COMPLETE FM BUG CIRCUIT

Corporate espionage is reaching new heights in sophistication. The latest information to be released shows the depths firms will go to pry into a rival firm's operations.

By using the latest in electronic bugging, they have stolen information, secrets and even formulas known only to the inventors themselves.

Take the example of one firm:

Leaks from Top Management level remained a mystery until, one day, a bug was discovered inside the Managing Director's office.
Sitting prominently on his desk was a gift box of imported cigars!

Cleverly concealed in the lower part of the box was a miniature FM transmitter . . all a gift from a phoney sales rep.

This is just one of the many bugging devices available on the eavesdropping market. The range includes pen and pencil holders, trophies, framed pictures and office furniture with false bottom drawers.

These products are readily sold to fledgling companies, eager to nestle into big brother's market.

And for a while these bugging devices worked. Few firms knew of their existence, and even less on how to sniff them out.

But that has all changed now. If a corporation suspects a leak at any level, the first thing they order is an investigation into security. Not only personnel, but information and electronic security.

Debugging has grown into big business. Most large security organisations have a section concentrating on electronic surveillance including bugging and debugging.

Everyone has been absolutely amazed at the performance of this bug.

They use scanners to detect hidden devices and can locate absolutely anything, anywhere, and on any frequency.

It was only after the firm above had commissioned a scan of the entire floor, that the cigar box was discovered. Its innocence had deceived everyone. And cost them a small fortune!

Bugging of this kind is completely illegal and we don't subscribe to this type of application at all.

PARTS

1 - 470R ¼watt
1 - 10k
1 - 22k
1 - 47k
1 - 1M

1 - 5.6pf ceramic = 5p6
1 - 22pf ceramic or 27pf or 33pf
1 - 47pf ceramic
1 - 1n ceramic = *1000pf or 102*
2 - 22n ceramic = .022 or 223

1 - 2.2mfd 16v or 25v PC electro

2 - BC 547 transistors (Not SGS type)

1 - mini slide switch spdt.

1 - electret microphone (insert)

2 - AAA cells

10cm tinned copper wire
2 - metres aerial wire

1 - FM BUG PC board

1 - Toothbrush case.

But the uses for our SUPER-SNOOP FM WIRELESS MICROPHONE can be harmless, helpful and a lot of fun.

Our unit is both compact and very sensitive and can be used to pick up even the faintest of conversations or noises and transmit them 20 or so metres to any FM receiver.

When you build the FM BUG you will see why we consider the design to be very clever. We have used only low priced components and they are all easy to obtain.

No air trimmer capacitor is required as the coil is squeezed slightly to obtain the desired frequency. This has allowed us to fit the bug into a tooth-brush case so that it can be carried around or placed on a shelf.

If it is set between two books it will be hidden from view or as a supervision accessory it can be placed on a small child, etc. The transmitted signal will over-ride the background noise and the the output will be clean. If the child wanders beyond the range of the transmitter, the background noise will come up and signal that the tot is out of range.

As an added bonus, you can listen to the chatterings and squabbles as the children amuse themselves in the back yard.

It is also great for picking up the first signs of a child awakening from his afternoon sleep ar it can be used as an indicator from a bed-ridden patient.

The great advantage of the bug is the absence of wires. And since it draws only about 5 - 10 milliamps, the pair of AAA cells will last for many months.
The success of this FM BUG is the use of TWO transistors in the circuit. To create a good design, like this, each transistor should be required to perform only one task. In any type of transmitter, there is a minimum of two tasks.

One is to amplify the signal from the microphone and the other is to provide a high frequency oscillator.

The amplified microphone signal is injected into the oscillator to modify its frequency and thus produce a FREQUENCY MODULATED oscillator. If an aerial is connected to the output of the oscillator, some of the energy will be radiated into the atmosphere.

To increase the output of our design, an RF amplifier would be needed but this gets into legal technicalities with maximum transmitting power.

It may be of interest to know that a record distance of 310 miles was achieved with a 350 micro-watt transmitter in the USA, some 15 years ago. This equates to an astounding ONE MILLION miles per watt!

In simple terms, an RF amplifier becomes a LINEAR amplifier. This can be seen as per the second transistor in the Polykit design as presented in issue 4.

We have opted for sensitivity and the first transistor is employed as a pre-amplifier. This will enable you to pick up very low-level sounds and transmit them about 20 to 50 metres.

MAKING THE OSCILLATOR COIL

The only critical component in the FM BUG is the oscillator coil. When I say critical, I am referring to its effect on the frequency. Its critical nature only means it must not be touched when the transmitter is in operation as this will detune the circuit completely.

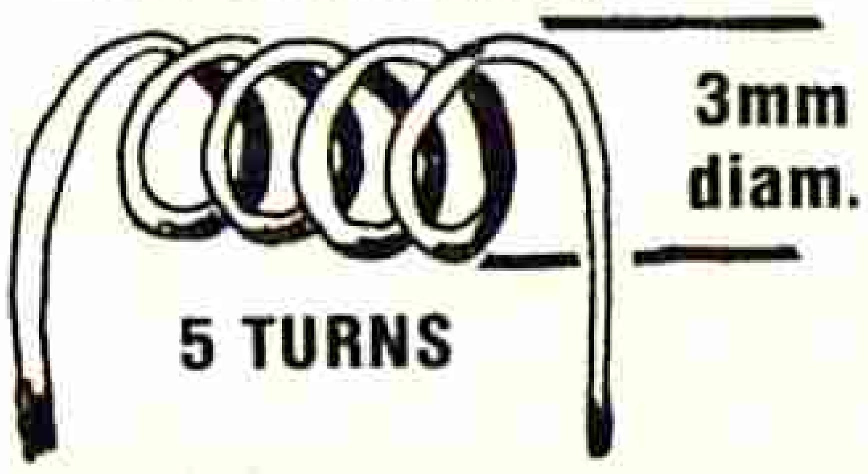
It is the only component which needs to be adjusted or aligned and we will cover its winding and formation in detail.

The oscillator coil is made out of tinned copper wire and does not need any insulation. This is not normal practice but since the coil is small and rigid, the turns are unable to touch each other and short-out.

The coil is made by winding the tinned copper wire over a medium-size phillips screw-driver. The gauge of wire, the diameter of the coil and the spacing between turns is not extremely important and it will be adjusted in the alignment stage. However when the project is fully aligned, it must not be touched at all.

Don't be over-worried at this stage. Just follow the size and shape as shown in the diagram and everything will come out right in the end.

THE DETAILS:



The coil has 5 turns. To be more specific, it has 5 loops of wire at the top and each end terminates at the PC board. The coil must be wound in a clock-wise direction to fit onto the board and if you make a mistake, rewind the coil in the opposite direction.

A pre-wound coil comes with the kits supplied by the magazine however you can make you own very easily in a few minutes.

Collect all the necessary components and lay them on the work bench ready for the next stage:

CONSTRUCTION

Construction is quite straight-forward as everything is mounted on the printed circuit board. The only point to watch is the height of some of the components. The capacitors and electrolytic must be folded over so that the board will fit into the case.

Positioning of the parts is not as critical as you think as the final frequency is adjusted by squeezing the coil together or stretching it apart.

However it is important to keep the component leads as short as possible and the soldering neat due to the high frequencies involved. The components must be soldered firmly to the board so that they do not move when the transmitter is being carried.

Even the poorest of soldering will work but who wants to see poor soldering on a project? Especially when it is housed in a clear perspex case.

The soldering may not affect the resulting frequency but poor layout of the components certainly will.

All the resistors must be pressed firmly against the PC board before soldering and the two transistors must be pushed so that they are shorter than the opening in the case.

Some BC 547 transistors will not work in the circuit. Maybe the frequency is too high. SGS BC 547 transitors did not work at all. The other two types: fBC 547 and Philips BC 547 worked perfectly.

All the small-value capacitors are ceramic as they are not critical in value and do not need to be high stability. But you must be careful when identifying them. It would be a very simple mistake to buy a 56pf instead of 5p6 because there is no difference in the size. 22n may be identified with 223 or 22n or .022. A capacitor marked 22k will be a 22pF cap and will not be suitable. The 1n capacitor may be marked 1n or .001 or 102. These are all the same value. The value 101 or 103 is NOT 1n so be careful, the caps may be about the same size. The rule is: don't use a capacitor unless its markings are clear and you are sure of the value.

The switch is mounted on the PC board with its three terminals fitted into the large holes.

Later, a square cut-out will be made in one half of the plastic case so that the slide of the switch protrudes through.

The final items to add to the board are the two AAA cells. These are available at Tandy stores and we have chosen them for slenderness so that they can be fitted side-by-side in the case. A small piece of tape will keep them together ready for connecting into circuit.

It is very difficult to solder to the zinc case but if you roughen the surface with a file and use a large, HOT, soldering iron, the job can be done very quickly. Use a piece of tinned copper wire to join the positive of one to the negative of the other. At the other end, solder longer lengths of wire so that they can be connected directly to the PC board. Make sure the positive terminal connects to the plus on the PC board.

AAA cells are also obtainable at photographic shops. The only alternative is an 'N' cell which is nearly as thin as an AAA cell but only half the length. If all this fails, you can use 2 AA cells in a long tooth-brush tube by connecting them end-to-end.

The terminal marked A on the board is the antenna output. For a frequency of 90MHz, the antenna should be 165cm long. This is classified as a half-wave antenna and provides one of the most effective radiators. If you find the antenna gets in the way you can opt for a quarter-wave antenna and this will be 83cm long. If you only require to transmit 10 to 20 metres the antenna can be as short as 42cm or even as low as 5 or 10 cm.

The most suitable length will depend on the sensitivity of the FM radio used to pick up the signal and the obstructions between the transmitter and receiver. It will be a good experiment for you to 'cut' your own antenna and determine which is the most suitable for your application.

HOW THE CIRCUIT WORKS

The circuit consists of two separate stages. The first is an audio pre-amplifier and the second is a 90MHz oscillator.

The first stage is very simple to explain. It is a self-biasing common-emitter amplifier capable of amplifying minute signals picked up by the electret microphone. It delivers these to the oscillator stage. The amplification of the first stage is about 100 and it only operates at audio frequencies. The 22n capacitor isolates the microphone from the base voltage of the transistor and allows only AC signals to pass through. The transistor is automatically biased via the 1M resistor which is fed from the voltage appearing at the collector. This is a simple yet very effective circuit. The output from the transistor passes through a 2.2mfd electrolytic. This value is not critical as its sole purpose is to couple the two stages.

The 47k, 1n, 470R and 22n components are not critical either. So, what are the critical components in this circuit?

The critical components are the coil and 47pF capacitor. These determine the frequency at which the bug will transmit. In addition, the effective capacitance of the transistor plays a deciding factor in the resulting frequency.

This stage is basically a free-running 90MHz oscillator in which the feedback path is the 5p6 capacitor.

When the circuit is turned on, a pulse of electricity passes through the collector-emitter circuit and this also includes the parallel tuned circuit made up of the oscillator coil and the 47pf capacitor. This pulse of electricity is due to the transistor being turned on via the 47k resistor in the base circuit.

When ever energy is injected into a tuned circuit, the energy is firstly absorbed by the capacitor. The electricity will then flow out to the coil where it is converted to magnetic flux. The magnetic flux will cut the turns of wire in the coil and produce current and voltage which will be passed to the capacitor.

In theory, this current will flow back and forth indefinitely, however in practice, there are a number of losses which will cause the oscillations to die down fairly quickly.

If a feedback circuit is provided for the stage, the natural RESONANT frequency of the coil/capacitor combination will be maintained. The 5p6 provides this feedback path and keeps the transistor oscillating.

The 5p6 feeds a small sample of the voltage appearing at the collector, to the emitter and modifies the emitter voltage. The transistor sees its base-to-emitter voltage altering in harmony with the resonant frequency of the tuned circuit and turns the collector on and off at the same frequency.

Thus there is a degree of stability in the oscillator frequency.

The actual frequency of the stage is dependent upon the total capacitance of the circuit and this includes all the other components to a minor extent.

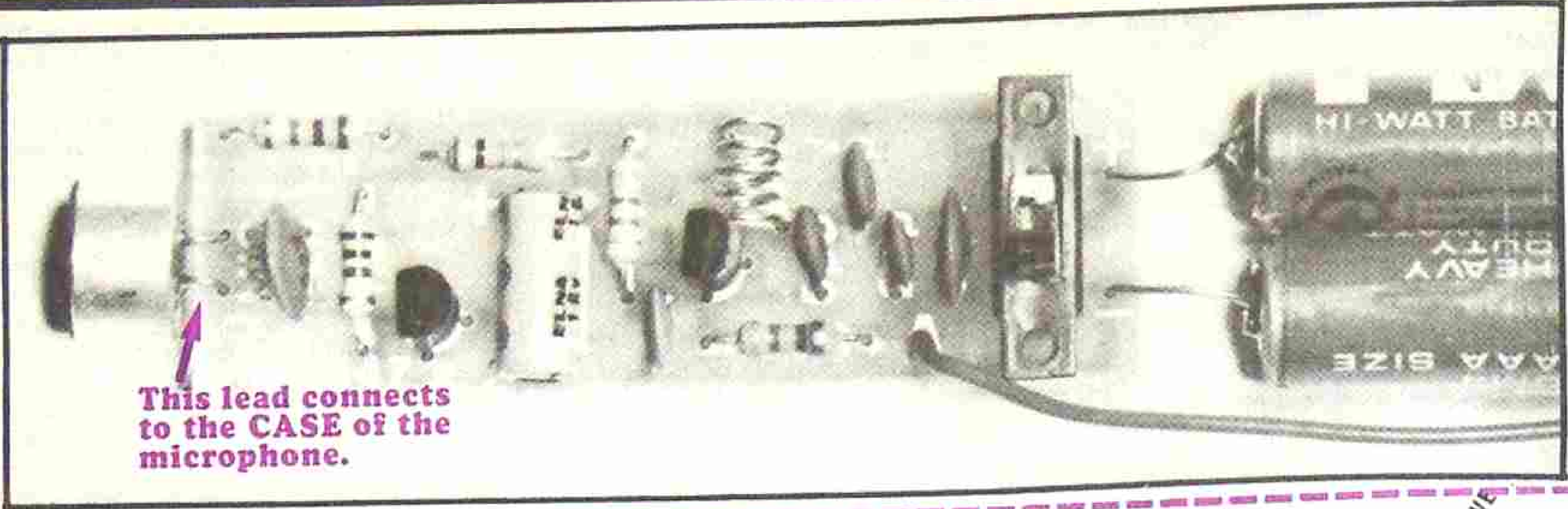
Once the basic frequency of 90MHz is set, the variations in frequency are produced by the changes in effective capacitance of the transistor. This occurs when its base voltage is increased and reduced. The electret microphone picks up the sound waves which are amplified by the first transistor and the resulting frequency is passed to the base of Q2 via the 2.2mfd electrolytic.

This alters the gain of the transistor and changes its internal capacitance. This junction capacitance modifies the oscillator with a frequency equal to the sound entering the microphone thus FREQUENCY MODULATING the circuit. A short length of antenna wire is connected to the collector of the oscillator via a coupling capacitor and some of the energy of the circuit will be radiated to the surroundings.

Any FM receiver will pick up this energy and decode the audio portion of the signal.

MEMO from Gino: If you decrease the 22k to 10k or 4k7, the BUG becomes even more sensitive! See his BIG EAR project in the next issue.

The completed BUG in the clear plastic case. The aerial wire supplied is sufficient for a 165cm half-wave antenna and can be cut to 83cm or shorter, depending on the range you require.

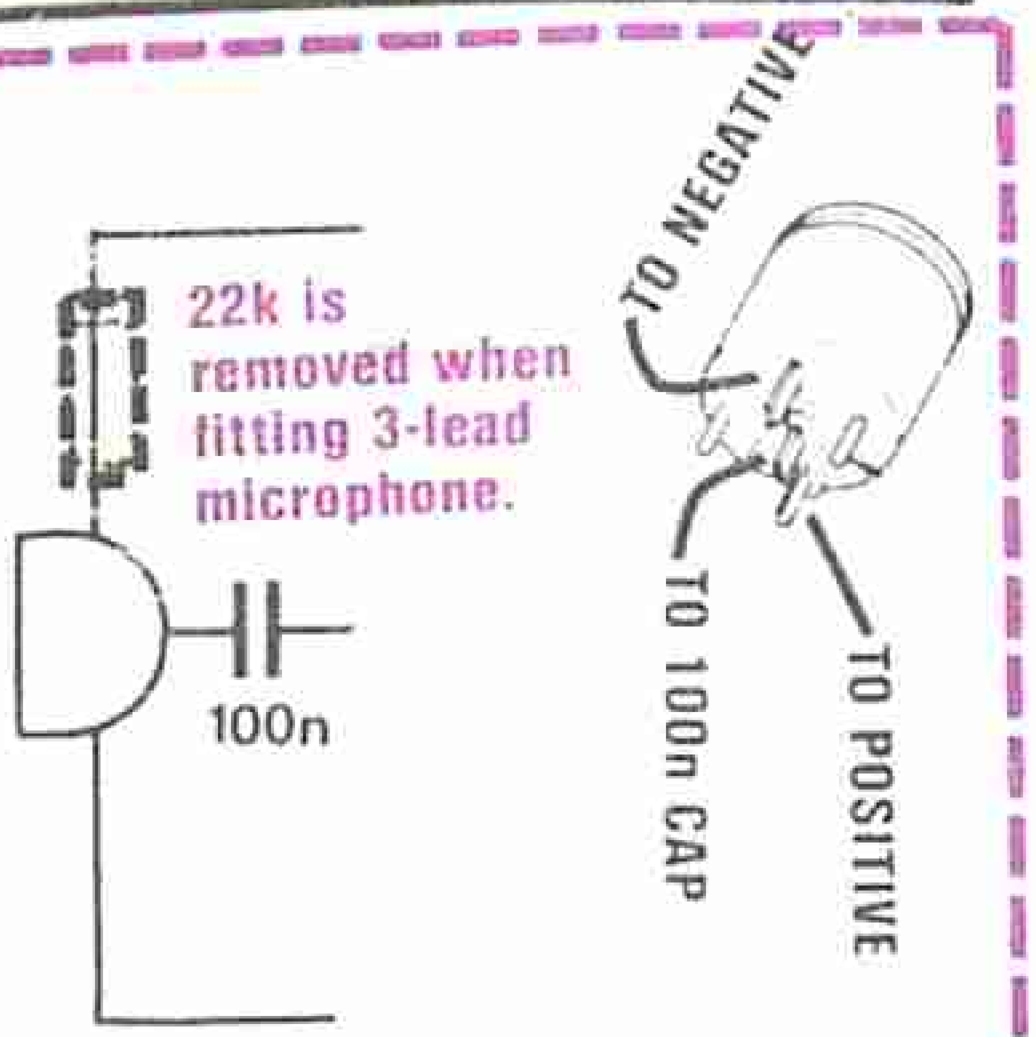


3-LEADED ELECTRET MIC:

If you have a 3-leaded electret microphone, it can be used in the circuit provided a simple modification is made. Three-leaded mics have internal dropping resistors and thus the 22k resistor in our circuit is not needed.

All microphones are polarity sensitive and must be inserted into the circuit around the correct way. See the accompanying diagram.

The 2 holes in the PC take the negative pin and the output pin. A jumper wire is taken from the positive pin on the mic to the positive line on the PC, at a point where the 22k meets the positive rail.



SETTING UP THE TRANSMITTER

When the FM BUG is complete, checked and ready for insertion into its case, there is one slight adjustment which must be made to align it to the correct frequency.

As we have said, the only critical component is the oscillator coil. It is the only item which is adjustable.

Since we are working with a very high freqency, the proximity of your hand or even a metal screw-driver will tend to de-tune the oscillator appreciably.

For this reason you must use a plastic aligning stick to make the adjustment. Any piece of plastic will do. A knitting needle, pen barrel or plastic stirring stick can be used.

Place the bug about a metre from the FM radio and switch both units on. Tune the radio to an unused portion of the band and use the alignment stick to push the turns of the coil together. Make sure none of the turns touch each other as this will short out the operation of the oscillator.

All of a sudden you will hear the background noise diminish and you may even get feed back. This amount of adjustment is sufficient. Place the BUG in its case and tape up the two halves.

The fine tuning between radio and transmitter is done on the radio. Peak the reception and move the BUG further away. Peak the fine tune again and move the BUG into another part of the house and see how far it will transmit.

IF THE BUG FAILS

If the bug fails to operate, you have a problem. Simple digital tests will not fix it nor will ordinary audio procedures. The frequency at which the BUG operates is too high.

You have to use a new method called comparison.

This involves the comparing of a unit which works, with the faulty unit.

This means it is ideal for a group of constructors to build a number of units and compare one against the other.

This will not be possible with individual constructors and they will have to adapt this fault-finding section.

The first fact you have to establish is the correct operation of the FM receiver.

If you have another BUG and it is capable of transmitting through the radio you know the radio is tuned to the correct frequency. Otherwise you will have to double-check the tuning of the dial and make sure the radio is switched to the correct setting.

The next stage is to determine if the BUG is functioning AT ALL. The only voltage measurements you can make are across the collector-emitter terminals of the first transistor (1v to 1.5v) and across the collector-emitter terminals of the second transistor (1.3v to 1.5v) These values won't tell you much, except that the battery voltage is reaching the component.

Tune the radio to about 90MHz and lay the radio antenna very close to the antenna of the BUG. Switch the BUG on and off via the slide switch. You should hear a click in the radio if the BUG is on a frequency NEAR 90MHz. Move the turns of the aerial coil together or apart with a plastic stick as you switch the unit ON and OFF.

If a click is heard but no feed-back, the oscillator will be operating but not the pre-amp stage. This could be due to the electret microphone being around the wrong way, the transistor around the wrong way, a missing component or an open 2.2mfd electro.

If the fault cannot be located, compare your unit with a friend's. You may have made a solder bridge, connected the batteries around the wrong way, made the coil too big or used the wrong value capacitor for one of the values.

If all this fails, put the unit aside and start again. This time buy a complete kit and see how much more success you have.

Z80 Machine Codes

This table contains over 700 Machine Code instructions for the Z80.
It has been compiled from Zilog Data sheets, SGS Data books, Z80 Programming by P. Levison (now out of print) and Micro-Professor Programming Handbooks.
Two books to help with the interpretation of this table are: **Z80 ASSEMBLY LANGUAGE** by Lance A. Leventhal (Mc Graw Hill) and **PROGRAMMING the Z80** by Rodnay Zaks (Sybex).

ADC A,(HL)	8E
ADC A,(IX+dis)	DD 8E XX
ADC A,(IY+dis)	FD 8E XX
ADC A,A	8F
ADC A,B	88
ADC A,C	89
ADC A,D	8A
ADC A,dd	CE dd
ADC A,E	8B
ADC A,H	8C
ADC A,L	8D
ADC HL,BC	ED 4A
ADC HL,DE	ED 5A
ADC HL,HL	ED 6A
ADC HL,SP	ED 7A
ADD A,(HL)	86
ADD A,(IX+dis)	DD 86 XX
ADD A,(IY+dis)	FD 86 XX
ADD A,A	87
ADD A,B	80
ADD A,C	81
ADD A,D	82
ADD A,dd	C6 dd
ADD A,E	83
ADD A,H	84
ADD A,L	85
ADD HL,BC	09
ADD HL,DE	19
ADD HL,HL	29
ADD HL,SP	39
ADD IX,BC	DD 09
ADD IX,DE	DD 19
ADD IX,IX	DD 29
ADD IX,SP	DD 39
ADD IY,BC	FD 09
ADD IY,DE	FD 19
ADD IY,IY	FD 29
ADD IY,SP	FD 39
AND (HL)	A6
AND (IX+dis)	DD A6 XX
AND (IY+dis)	FD A6 XX
AND A	A7
AND B	A0
AND C	A1
AND D	A2
AND dd	E6 dd
AND E	A3
AND H	A4
AND L	A5
BIT 0,(HL)	CB 46
BIT 0,(IX+dis)	DD CB XX 46
BIT 0,(IY+dis)	FD CB XX 46
BIT 0,A	CB 47
BIT 0,B	CB 40
BIT 0,C	CB 41
BIT 0,D	CB 42
BIT 0,E	CB 43
BIT 0,H	CB 44
BIT 0,L	CB 45
BIT 1,(HL)	CB 4E
BIT 1,(IX+dis)	DD CB XX 4E
BIT 1,(IY+dis)	FD CB XX 4E
BIT 1,A	CB 4F
BIT 1,B	CB 48
BIT 1,C	CB 49
BIT 1,D	CB 4A
BIT 1,E	CB 4B
BIT 1,H	CB 4C
BIT 1,L	CB 4D
BIT 2,(HL)	CB 56
BIT 2,(IX+dis)	DD CB XX 56
BIT 2,(IY+dis)	FD CB XX 56
BIT 2,A	CB 57
BIT 2,B	CB 50
BIT 2,C	CB 51
BIT 2,D	CB 52
BIT 2,E	CB 53
BIT 2,H	CB 54
BIT 2,L	CB 55
BIT 3,(HL)	CB 5E
BIT 3,(IX+dis)	DD CB XX 5E
BIT 3,(IY+dis)	FD CB XX 5E
BIT 3,A	CB 5F
BIT 3,B	CB 58
BIT 3,C	CB 59
BIT 3,D	CB 5A
BIT 3,E	CB 5B
BIT 3,H	CB 5C
BIT 3,L	CB 5D
BIT 4,(HL)	CB 66
BIT 4,(IX+dis)	DD CB XX 66
BIT 4,(IY+dis)	FD CB XX 66
BIT 4,A	CB 67
BIT 4,B	CB 60
BIT 4,C	CB 61
BIT 4,D	CB 62
BIT 4,E	CB 63
BIT 4,H	CB 64
BIT 4,L	CB 65
BIT 5,(HL)	CB 6E
BIT 5,(IX+dis)	DD CB XX 6E
BIT 5,(IY+dis)	FD CB XX 6E
BIT 5,A	CB 6F
BIT 5,B	CB 68
BIT 5,C	CB 69
BIT 5,D	CB 6A
BIT 5,E	CB 6B
BIT 5,H	CB 6C
BIT 5,L	CB 6D
BIT 6,(HL)	CB 76
BIT 6,(IX+dis)	DD CB XX 76
BIT 6,(IX+dis)	FD CB XX 76
BIT 6,A	CB 77
BIT 6,B	CB 70
BIT 6,C	CB 71
BIT 6,D	CB 72
BIT 6,E	CB 73
BIT 6,H	CB 74
BIT 6,L	CB 75
BIT 7,(HL)	CB 7E
BIT 7,(IX+dis)	DD CB XX 7E
BIT 7,(IY+dis)	FD CB XX 7E
BIT 7,A	CB 7F
BIT 7,B	CB 78
BIT 7,C	CB 79
BIT 7,D	CB 7A
BIT 7,E	CB 7B
BIT 7,H	CB 7C
BIT 7,L	CB 7D
CALL ADDR	CD XX XX
CALL C,ADDR	DC XX XX
CALL M,ADDR	FC XX XX
CALL NC,ADDR	D4 XX XX
CALL NZ,ADDR	C4 XX XX
CALL P,ADDR	F4 XX XX
CALL PE,ADDR	EC XX XX
CALL PO,ADDR	E4 XX XX
CALL Z,ADDR	CC XX XX
CCF	3F
CP (HL)	BE
CP (IX+dis)	DD BE XX
CP (IY+dis)	FD BE XX
CP A	BF
CP B	B8
CP C	B9
CP D	BA
CP dd	FE dd
CP E	BB
CP H	BC
CP L	BD
CPD	ED A9
CPDR	ED B9
CPI	ED A1
CPIR	ED B1
CPL	2F
DAA	27
DEC (HL)	35
DEC (IX+dis)	DD 35 XX
DEC (IY+dis)	FD 35 XX
DEC A	3D
DEC B	05
DEC BC	0B
DEC C	0D
DEC D	15
DEC DE	1B
DEC E	1D
DEC H	25
DEC HL	2B
DEC IX	DD 2B
DEC IY	FD 2B
DEC L	2D
DEC SP	3B
DI	F3
DJNZ dis	10 XX
EI	FB
EX (SP),HL	E3
EX (SP),IX	DD E3
EX (SP),IY	FD E3
EX AF,AF'	08
EX DE,HL	EB
EXX	D9
HALT	76
IM 0	ED 46
IM 1	ED 56
IM 2	ED 5E
IN A,(C)	ED 78
IN A,port	DB XX
IN B,(C)	ED 40
IN C,(C)	ED 48
IN D,(C)	ED 50
IN E,(C)	ED 58
IN H,(C)	ED 60
IN L,(C)	ED 68
INC (HL)	34
INC (IX+dis)	DD 34 XX
INC (IY+dis)	FD 34 XX
INC A	3C
INC B	04
INC BC	03
INC C	0C
INC D	14
INC DE	13
INC E	1C
INC H	24
INC HL	23
INC IX	DD 23
INC IY	FD 23
INC L	2C
INC SP	33
IND	ED AA
INDR	ED BA
INI	ED A2
INIR	ED B2
JP (HL)	E9
JP (IX)	DD E9
JP (IY)	FD E9
JP ADDR	C3 XX XX
JP C,ADDR	DA XX XX
JP M,ADDR	FA XX XX
JP NC,ADDR	D2 XX XX
JP NZ,ADDR	C2 XX XX
JP P,ADDR	F2 XX XX
JP PE,ADDR	EA XX XX
JP PO,ADDR	E2 XX XX
JP Z,ADDR	CA XX XX
JR C,dis	38 XX
JR dis	18 XX
JR NC,dis	30 XX
JR NZ,dis	20 XX
JR Z,dis	28 XX
LD (ADDR),A	32 XX XX
LD (ADDR),BC	ED 43 XX XX
LD (ADDR),DE	ED 53 XX XX
LD (ADDR),HL	ED 63 XX XX
LD (ADDR),HL	22 XX XX
LD (ADDR),IX	DD 22 XX XX
LD (ADDR),IY	FD 22 XX XX
LD (ADDR),SP	ED 73 XX XX
LD (BC),A	02
LD (DE),A	12
LD (HL),A	77
LD (HL),B	70
LD (HL),C	71
LD (HL),D	72
LD (HL),dd	36 dd
LD (HL),E	73
LD (HL),H	74
LD (HL),L	75
LD (IX+dis),A	DD 77 XX
LD (IX+dis),B	DD 70 XX
LD (IX+dis),C	DD 71 XX
LD (IX+dis),D	DD 72 XX
LD (IX+dis),dd	DD 36 XX dd
LD (IX+dis),E	DD 73 XX
LD (IX+dis),H	DD 74 XX
LD (IX+dis),L	DD 75 XX
LD (IY+dis),A	FD 77 XX
LD (IY+dis),B	FD 70 XX
LD (IY+dis),C	FD 71 XX
LD (IY+dis),D	FD 72 XX
LD (IY+dis),dd	FD 36 XX dd
LD (IY+dis),E	FD 73 XX
LD (IY+dis),H	FD 74 XX
LD (IY+dis),L	FD 75 XX
LD A,(ADDR)	3A XX XX
LD A,(BC)	0A
LD A,(DE)	1A
LD A,(HL)	7E
LD A,(IX+dis)	DD 7E XX
LD A,(IY+dis)	FD 7E XX
LD A,A	7F
LD A,B	78
LD A,C	79
LD A,D	7A
LD A,dd	3E dd
LD A,E	7B
LD A,H	7C
LD A,I	ED 57
LD A,L	7D
LD A,R	ED 5F
LD B,(HL)	46
LD B,(IX+dis)	DD 46 XX
LD B,(IY+dis)	FD 46 XX
LD B,A	47
LD B,B	40
LD B,C	41
LD B,D	42
LD B,dd	06 dd
LD B,E	43
LD B,H	44
LD B,L	45
LD BC,(ADDR)	ED 4B XX XX
LD BC,dd	01 dd dd
LD C,(HL)	4E
LD C,(IX+dis)	DD 4E XX
LD C,(IY+dis)	FD 4E XX
LD C,A	4F
LD C,B	48
LD C,C	49
LD C,D	4A
LD C,dd	0E dd
LD C,E	4B
LD C,H	4C
LD C,L	4D
LD D,(HL)	56
LD D,(IX+dis)	DD 56 XX
LD D,(IY+dis)	FD 56 XX
LD D,A	57
LD D,B	50
LD D,C	51
LD D,D	52
LD D,dd	16 dd
LD D,E	53
LD D,H	54
LD D,L	55
LD DE,(ADDR)	ED 5B XX XX
LD DE,dddd	11 dd dd
LD E,(HL)	5E
LD E,(IX+dis)	DD 5E XX
LD E,(IY+dis)	FD 5E XX
LD E,A	5F
LD E,B	58
LD E,C	59
LD E,D	5A
LD E,dd	1E dd
LD E,E	5B
LD E,H	5C
LD E,L	5D
LD H,(HL)	66
LD H,(IX+dis)	DD 66 XX
LD H,(IY+dis)	FD 66 XX
LD H,A	67
LD H,B	60
LD H,C	61
LD H,D	62
LD H,dd	26 dd
LD H,E	63
LD H,H	64
LD H,L	65
LD HL,(ADDR)	ED 6B XX XX
LD HL,(ADDR)	2A XX XX
LD HL,dddd	21 dd dd
LD I,A	ED 47
LD IX,(ADDR)	DD 2A XX XX
LD IX,dddd	DD 21 dd dd
LD IY,(ADDR)	FD 2A XX XX
LD IY,dddd	FD 21 dd dd
LD L,(HL)	6E
LD L,(IX+dis)	DD 6E XX
LD L,(IY+dis)	FD 6E XX
LD L,A	6F
LD L,B	68
LD L,C	69
LD L,D	6A
LD L,dd	2A dd
LD L,E	6B
LD L,H	6C
LD L,L	6D
LD R,A	ED 4F
LD SP,(ADDR)	ED 7B XX XX
LD SP,dddd	31 dd dd
LD SP,HL	F9
LD SP,IX	DD F9
LD SP,IY	FD F9
LDD	ED A8
LDDR	ED B8
LDI	ED A0
LDIR	ED B0
NEG	ED 44
NOP	00
OR (HL)	B6
OR (IX+dis)	DD B6 XX
OR (IY+dis)	FD B6 XX
OR A	B7
OR B	B0
OR C	B1
OR D	B2
OR dd	F6 dd
OR E	B3
OR H	B4
OR L	B5
OTDR	ED BB
OTIR	ED B3
OUT (C),A	ED 79
OUT (C),B	ED 41
OUT (C),C	ED 49
OUT (C),D	ED 51
OUT (C),E	ED 59
OUT (C),[illegible]	[illegible] 61
OUT (C),[illegible]	[illegible] 69
OUT port,A	[illegible] port
OUTD	ED AB
OUTI	ED A3
POP AF	F1
POP BC	C1
POP DE	D1
POP HL	[illegible]
POP IX	DD E1
POP IY	FD E1
PUSH AF	F5
PUSH BC	C5
PUSH DE	D5
PUSH HL	E5
PUSH IX	DD E5
PUSH IY	FD E5
RES 0,(HL)	CB 86
RES 0,(IX+dis)	DD CB XX 86
RES 0,(IY+dis)	FD CB XX 86
RES 0,A	CB 87
RES 0,B	CB 80
RES 0,C	CB 81
RES 0,D	CB 82
RES 0,E	CB 83
RES 0,H	CB 84
RES 0,L	CB 85
RES 1,(HL)	CB 8E
RES 1,(IX+dis)	DD CB XX 8E
RES 1,(IY+dis)	FD CB XX 8E
RES 1,A	CB 8F
RES 1,B	CB 88
RES 1,C	CB 89
RES 1,D	CB 8A
RES 1,E	CB 8B
RES 1,H	CB 8C
RES 1,L	CB 8D
RES 2,(HL)	CB 96
RES 2,(IX+dis)	DD CB XX 96
RES 2,(IY+dis)	FD CB XX 96
RES 2,A	CB 97
RES 2,B	CB 90
RES 2,C	CB 91
RES 2,D	CB 92
RES 2,E	CB 93
RES 2,H	CB 94
RES 2,L	CB 95
RES 3,(HL)	CB 9E
RES 3,(IX+dis)	DD CB XX 9E
RES 3,(IY+dis)	FD CB XX 9E
RES 3,A	CB 9F
RES 3,B	CB 98
RES 3,C	CB 99
RES 3,D	CB 9A
RES 3,E	CB 9B
RES 3,H	CB 9C
RES 3,L	CB 9D
RES 4,(HL)	CB A6
RES 4,(IX+dis)	DD CB XX A6
RES 4,(IY+dis)	FD CB XX A6
RES 4,A	CB A7
RES 4,B	CB A0
RES 4,B	CB A1
RES 4,C	CB A2
RES 4,D	CB A3
RES 4,E	CB A4
RES 4,H	CB A5
RES 5,(HL)	CB AE
RES 5,(IX+dis)	DD CB XX AE
RES 5,(IY+dis)	FD CB XX AE
RES 5,A	CB AF
RES 5,B	CB A8
RES 5,C	CB A9
RES 5,D	CB AA
RES 5,E	CB AB
RES 5,H	CB AC
RES 5,L	CB AD
RES 6,(HL)	CB B6
RES 6,(IX+dis)	DD CB XX B6
RES 6,(IY+dis)	FD CB XX B6
RES 6,A	CB B7
RES 6,B	CB B0
RES 6,C	CB B1
RES 6,D	CB B2
RES 6,E	CB B3
RES 6,H	CB B4
RES 6,L	CB B5
RES 7,(HL)	CB BE
RES 7,(IX+dis)	DD CB XX BE
RES 7,(IY+dis)	FD CB XX BE
RES 7,A	CB BF
RES 7,B	CB B8
RES 7,C	CB B9
RES 7,D	CB BA
RES 7,E	CB BB
RES 7,H	CB BC
RES 7,L	CB BD
RET	C9
RET C	D8
RET M	F8
RET NC	D0
RET NZ	C0
RET P	F0
RET PE	E8
RET PO	E0
RET Z	C8
RETI	ED 4D
RETN	ED 45
RL (HL)	CB 16
RL (IX+dis)	DD CB XX 16
RL (IY+dis)	FD CB XX 16
RL A	CB 17
RL B	CB 10
RL C	CB 11
RL D	CB 12
RL E	CB 13
RL H	CB 14
RL L	CB 15
RLA	17
RLC (HL)	CB 06
RLC (IX+dis)	DD CB XX 06
RLC (IY+dis)	FD CB XX 06
RLC A	CB 07
RLC B	CB 00
RLC C	CB 01
RLC D	CB 02
RLC E	CB 03
RLC H	CB 04
RLC L	CB 05
RLCA	07
RLD	ED 67
RR (HL)	CB 1E
RR (IX+dis)	DD CB XX 1E
RR (IY+dis)	FD CB XX 1E
RR A	CB 1F
RR B	CB 18
RR C	CB 19
RR D	CB 1A
RR E	CB 1B
RR H	CB 1C
RR L	CB 1D
RRA	1F
RRC (HL)	CB 0E
RRC (IX+dis)	DD CB XX 0E
RRC (IY+dis)	FD CB XX 0E
RRC A	CB 0F
RRC B	CB 08
RRC C	CB 09
RRC D	CB 0A
RRC E	CB 0B
RRC H	CB 0C
RRC L	CB 0D
RRCA	0F
RRD	ED 6F
RST 00	C7
RST 08	CF
RST 10	D7
RST 18	DF
RST 20	E7
RST 28	EF
RST 30	F7
RST 38	FF
SBC A,(HL)	9E
SBC A,(IX+dis)	DD 9E XX
SBC A,(IY+dis)	FD 9E XX
SBC A,A	9F
SBC A,B	98
SBC A,C	99
SBC A,D	9A
SBC A,dd	DE dd
SBC A,E	9B
SBC A,H	9C
SBC A,L	9D
SBC HL,BC	ED 42
SBC HL,DE	ED52
SBC HL,HL	ED 62
SBC HL,SP	ED 72
SCF	37
SET 0,(HL)	CB C6
SET 0,(IX+dis)	DD CB XX C6
SET 0,(IY+dis)	FD CB XX C6
SET 0,A	CB C7
SET 0,B	CB C0
SET 0,C	CB C1
SET 0,D	CB C2
SET 0,E	CB C3
SET 0,H	CB C4
SET 0,L	CB C5
SET 1,(HL)	CB CE
SET 1,(IX+dis)	DD CB XX CE
SET 1,(IY+dis)	FD CB XX CE
SET 1,A	CB CF
SET 1,B	CB C8
SET 1,C	CB C9
SET 1,D	CB CA
SET 1,E	CB CB
SET 1,H	CB CC
SET 1,L	CB CD
SET 2,(HL)	CB D6
SET 2,(IX+dis)	DD CB XX D6
SET 2,(IY+dis)	FD CB XX D6
SET 2,A	CB D7
SET 2,B	CB D0
SET 2,C	CB D1
SET 2,D	CB D2
SET 2,E	CB D3
SET 2,H	CB D4
SET 2,L	CB D5
SET 3,(HL)	CB DE
SET 3,(IX+dis)	DD CB XX DE
SET 3,(IY+dis)	FD CB XX DE
SET 3,A	CB DF
SET 3,B	CB D8
SET 3,C	CB D9
SET 3,D	CB DA
SET 3,E	CB DB
SET 3,H	CB DC
SET 3,L	CB DD
SET 4,(HL)	CB E6
SET 4,(IX+dis)	DD CB XX E6
SET 4,(IY+dis)	FD CB XX E6
SET 4,A	CB E7
SET 4,B	CB E0
SET 4,C	CB E1
SET 4,D	CB E2
SET 4,E	CB E3
SET 4,H	CB E4
SET 5,(HL)	CB EE
SET 5,(IX+dis)	DD CB XX EE
SET 5,(IY+dis)	FD CB XX EE
SET 5,A	CB EF
SET 5,B	CB E8
SET 5,C	CB E9
SET 5,D	CB EA
SET 5,E	CB EB
SET 5,H	CB EC
SET 5,L	CB ED
SET 6,(HL)	CB F6
SET 6,(IX+dis)	DD CB XX F6
SET 6,(IY+dis)	FD CB XX F6
SET 6,A	CB F7
SET 6,B	CB F0
SET 6,C	CB F1
SET 6,D	CB F2
SET 6,E	CBF3
SET 6,H	CB F4
SET 6,L	CB F5
SET 7,(HL)	CB FE
SET 7,(IX+dis)	DD CB XX FE
SET 7,(IY+dis)	FD CB XX FE
SET 7,A	CB FF
SET 7,B	CB F8
SET 7,C	CB F9
SET 7,D	CB FA
SET 7,E	CB FB
SET 7,H	CB FC
SET 7,L	CB FD
SLA (HL)	CB 26
SLA (IX+dis)	DD CB XX 26
SLA (IY+dis)	FD CB XX 26
SLA A	CB 27
SLA B	CB 20
SLA C	CB 21
SLA D	CB 22
SLA E	CB 23
SLA H	CB 24
SLA L	CB 25
SRA (HL)	CB 2E
SRA (IX+dis)	DD CB XX 2E
SRA (IY+dis)	FD CB XX 2E
SRA A	CB 2F
SRA B	CB 28
SRA C	CB 29
SRA D	CB 2A
SRA E	CB 2B
SRA H	CB 2C
SRA L	CB 2D
SRL (HL)	CB 3E
SRL (IX+dis)	DD CB [illegible]
SRL (IY+dis)	FD CB XX [illegible]
SRL A	CB 3F
SRL B	CB 38
SRL C	CB 39
SRL D	CB 3A
SRL E	CB 3B
SRL H	CB 3C
SRL L	CB 3D
SUB (HL)	96
SUB (IX+dis)	DD 96 XX
SUB (IY+dis)	FD 96 XX
SUB A	97
SUB B	90
SUB C	91
SUB D	92
SUB dd	D6 dd
SUB E	93
SUB H	94
SUB L	95
XOR (HL)	AE
XOR (IX+dis)	DD AE XX
XOR (IY+dis)	FD AE XX
XOR A	AF
XOR B	A8
XOR C	A9
XOR D	AA
XOR dd	EE dd
XOR E	AB
XOR H	AC
XOR L	AD

DESIGNED and PUBLISHED by COLIN MITCHELL at 35 Rosewarne Ave., CHELTENHAM, VICTORIA. 3192 (03) 584 2386