TALKING ELECTRONICS®

$2.20 ★

$2.95 NZ

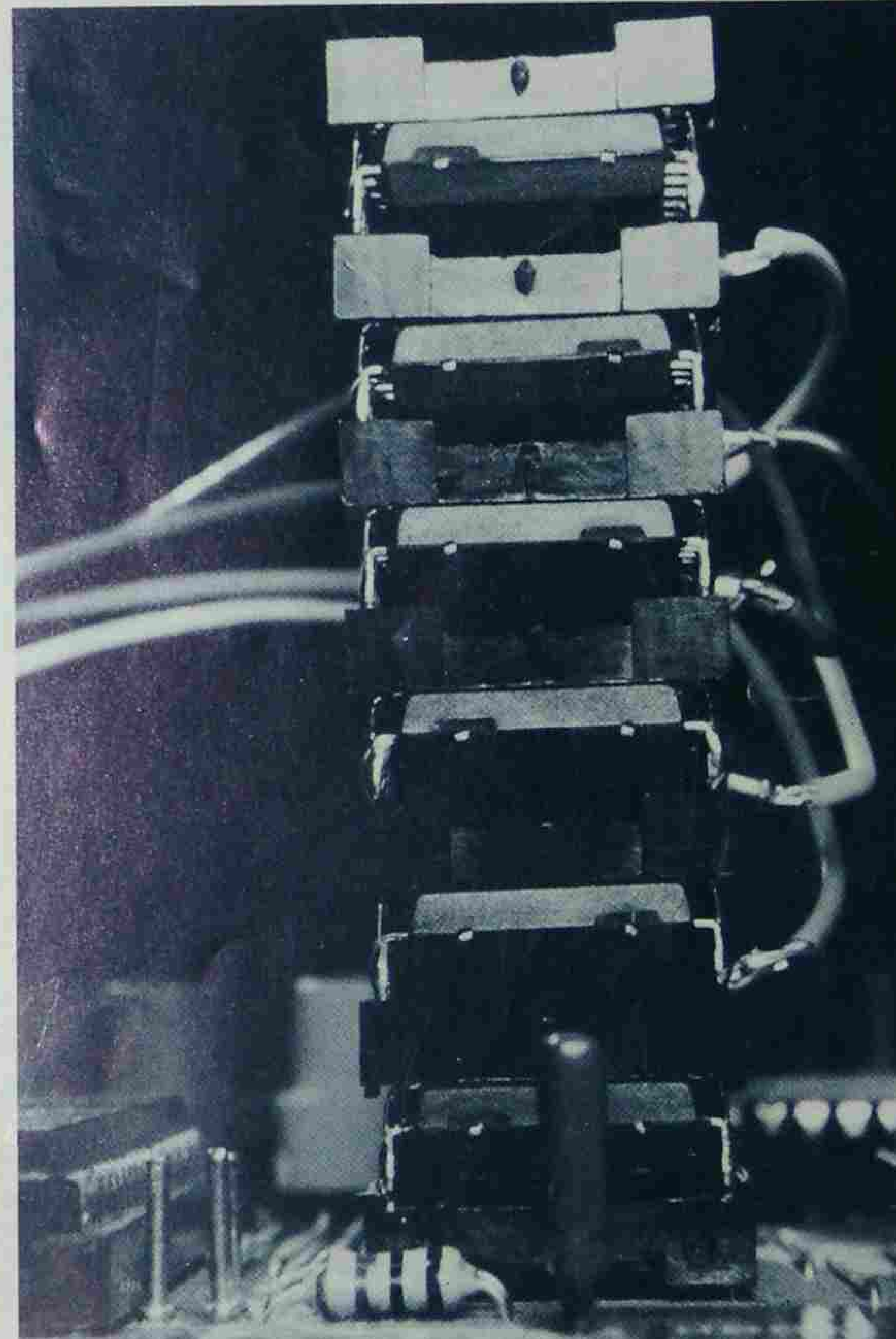
Issue No 12

HEADLIGHT

REMINDER

BIG EAR

TOUCH PUZZLE



2 'Add-ons' for the TEC

★RAM STACK

★PRINTER/PLOTTER

300884-16k

TALKING ELECTRONICS

Editorial...

Vol. 1 No: 12.

At the moment we are seeing a dramatic increase in magazines and books from overseas, dealing with computers and the like. And it may seem to be a bonus for Australia.

But there's a hidden reason for their presence.

Apart from the added sales they generate, many of these are a vehicle for promoting ONE brand of computer or ONE manufacturer of components.

Cleverly consealed within the web of information is an underlying stream of self promotion. Not evident at first, the general bias towards one particular theme gradually emerges.

Undoubtedly this arrangement is one of the cheapest and most efficient methods of promotion but it undermines the whole structure and intent of magazines and books. Supposedly impartial in content, magazines have always been considered to provide an overall unbiased view.

Those titles clearly displaying their association are exempt from this criticism. It's only the devious titles we are referring to.

I take particular exception to these because they are bought by the reader in the hope that they cover a broad spectrum of material. But in the end they are little more than an expanded advertising brochure.

I have been caught 4 times now. One series of magazines leant towards a particular brand of computer, another promoted a range of components from a particular manufacturer, another contained grossly out of date material and the fourth left the reader up in the air at the turn of each page - none of the examples were fully explained.

I won't be caught again.

Of course any market seeks its own level and very few 'false' magazines like this see an active market in Australia. But until their demise they take up valuable shelf space on the newsstand.

And they detract from the sales of more informative magazines.

I think all Australian publishers are suffering from the broad competition rising from these imports.Also from increasing costs and falling readership. But if a title contains valuable editorial, it will survive in the market place.

Fortunately we seem to be in the survival category. With the increase in electronics courses in schools and in industry, more people are realising the value of having an understanding of electronics.

At the moment we are planning an electronics text book for a course which will commence in two year's time. So you can be reassured we will be flat out for the next two years at least!

As I have said before, we have not seen 1% of the potential of electronics. Its impact will be the greatest thing ever to hit mankind.

If you are following electronics, you are going in the right direction. And I am sure we will be staying together.

Colin Mitchell.

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. All material is copyright however up to 30 photocopies is allowed for schools and clubs.

★Maximum recommended retail price only.

INDEX

5 HEADLIGHT REMINDER

10 BIG EAR

12 Z-80 DISASSEMBLY CODES

13 TEC-1 & TEC-1A PART III

22 TONES & TUNES ON THE TEC

23 CLOCK LISTING ON THE TEC

28 RAM STACK

30 PRINTER:PLOTTER

39 PRICE LIST FOR KITS AND PC BOARDS

40 SUBSCRIPTION FORM

42 ORDER FORMS

43 SHOP TALK

47 LETTERS

61 THE TV MAN TELLS

66 DIGITAL ELECTRONICS COURSE

72 TOUCH PUZZLE

75 PC ARTWORK for TEC-1A

76 DATA: Z-80 DISASSEMBLY CODES

TECHNICAL *Ken Stone*

ARTWORK *Ken Stone*

ENQUIRIES *10 minute queries will be answered on* **584 2386** *8am - 6pm.*

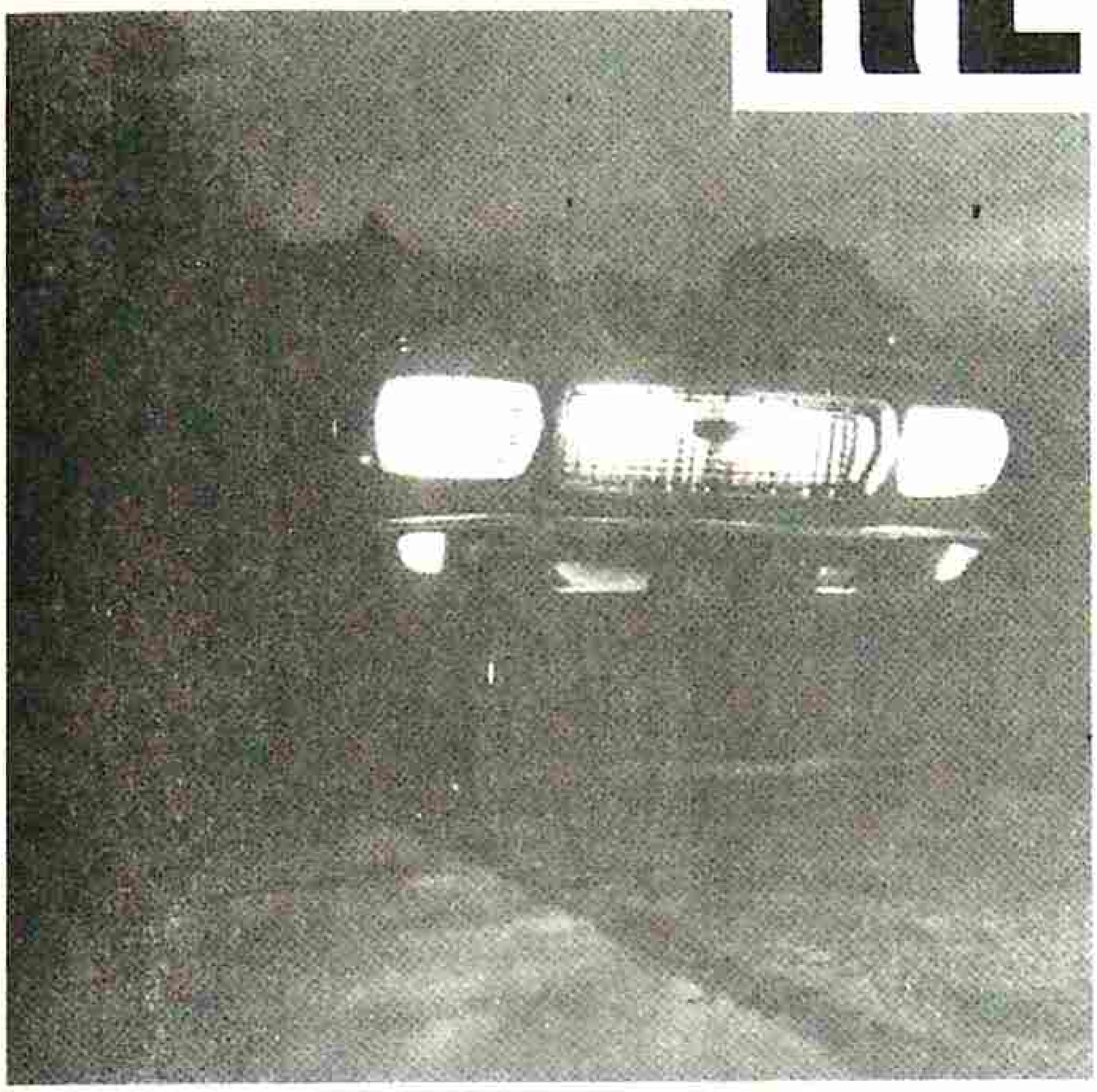
ADVERTISING **(03) 584 2386**

**Printed Web Offset by:
Standard Newspapers Ltd.,
10 Park Road, Cheltenham. 3192.**

Distributed in Australia by Gordon & Gotch.

HEADLIGHT REMINDER

Parts: $9.60
PC board: $3.30
Complete: $12.90



This clever little circuit is a three-in-one design.

It will tell you when to turn your headlights ON, when to turn them OFF and provide a flashing indicator on the dash-board similar to the indicator of a burglar alarm.

For the cost of building and fitting this circuit to your car, you could it save it being tampered with or even stolen!

Anyone looking into a car and seeing a flashing light (beside a sign reading ALARM ACTIVATED), will be sufficiently deterred to move onto an easier target.

Even though the flasher in our unit does not connect to any alarm equipment, how is the would-be thief to know? Some deterrents are silent and kill the ignition after 15 seconds, others sound the horn after a pre-determined time, while others switch off the fuel in the middle of an intersection!

Rather than risk a possible embarrassing situation, anyone intent on stealing your car will prefer to go to an unprotected model.

Our idea of a deterrent is by far the best, as alarms which have to be energised every time you leave the car (and de-activated on entry) often cause an annoyance when they accidently go off. Inevitably the driver tends to leave them disconnected for the rest of their serviceable life to avoid a reoccurance.

This leaves the car without any visible form of protection and the money spent on the alarm system is completely wasted.

Providing you don't add a sticker indicating the type of alarm you have fitted, no thief will know how the alarm works, when it works, or if it works at all!

But let's get back to the real reason for the production of this project.

No doubt, some time in the life of your driving career, you will forget to turn the headlights off when leaving the car. Most likely it won't be you but another member of the family who is less versed in the complexities of driving.

The result will be a flat battery and all the hassles of jump-starting.

On the other hand, this same member may take to the wheel without remembering to turn the headlights ON. And a potential moving death-trap will be created.

Without admitting too much about my driving ability, I must admit I have done both.

It is very easy to drive off without noticing the headlights are not on. Unless another motorist alerts you, it is possible to travel quite a distance, making the situation very dangerous.

The circuit presented here will help to overcome these situations.

It gives both visual and audible warning if you have forgotten to turn your headlights ON or OFF.

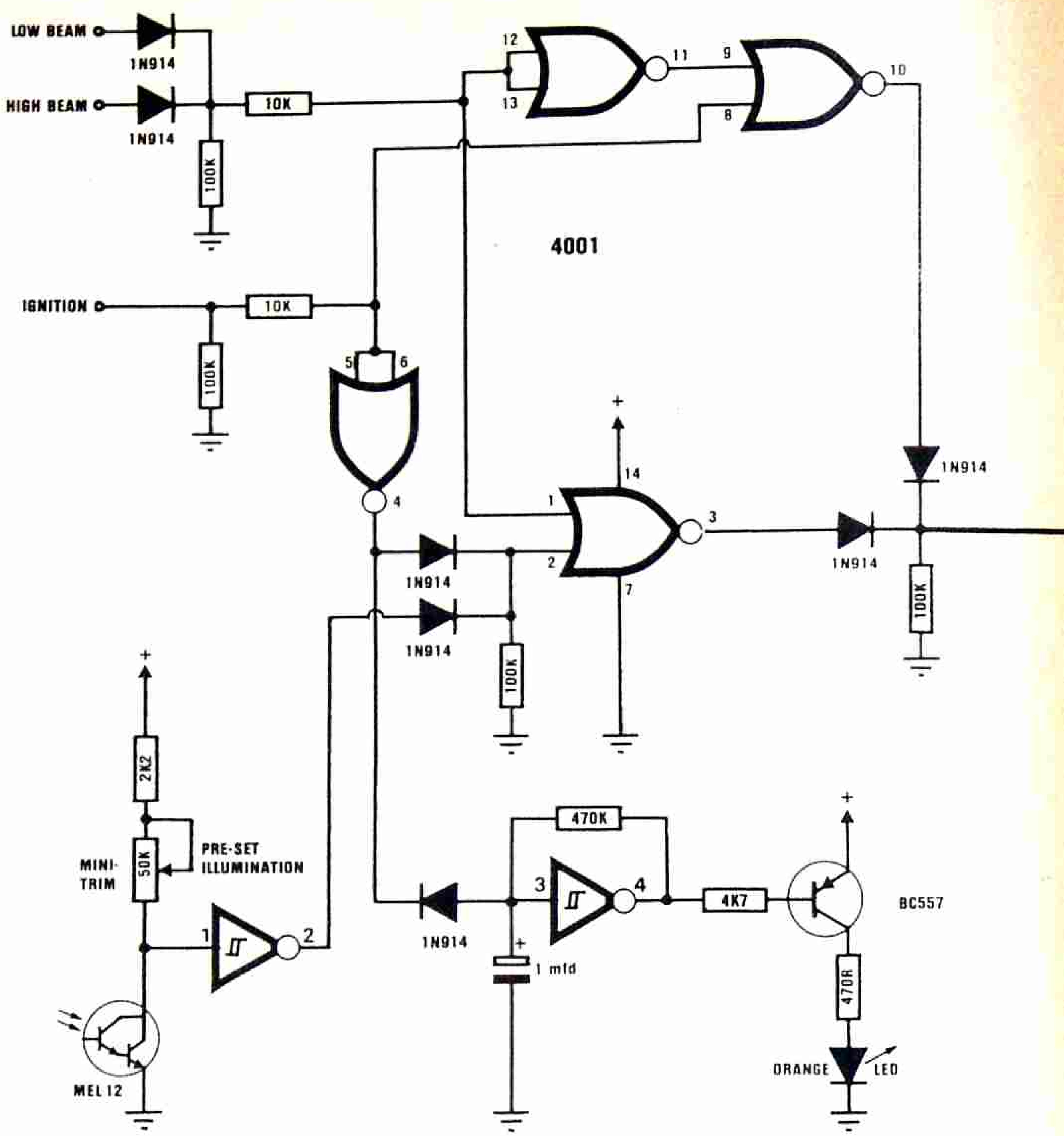
When designing the HEADLIGHT REMINDER, there were a number of points we had to take into account. It had to be cheap and compact, have both visual and audible indication, but most important it had to be easy to wire in.

We had to be very careful with the audible indication as it could be very distracting if allowed to continue for a long time, so we gave it a short burst.

The visual indication was different. It could be designed to stay on until the condition was rectified.

The only other gating condition was the pseudo-alarm LED. It would have to be turned OFF when the ignition was ON to avoid distracting the driver.

With all these conditions put together, we designed the following circuit.



HEADLIGHT

As you can see, it is a set of individual blocks, each gated into operation by one or more diodes. Diode gating saves a lot of IC's and has allowed us to reduce the circuit to two low-cost chips.

The versatile Schmitt Trigger has once again been used and this time its six inverters have been made into three different types of circuits. One is a level detector, two are delay circuits, and three are used as oscillators.

The operation of the circuit provides a good example of simple logic as well as gating arrangements. So, here's a run-down on:

HOW THE CIRCUIT WORKS

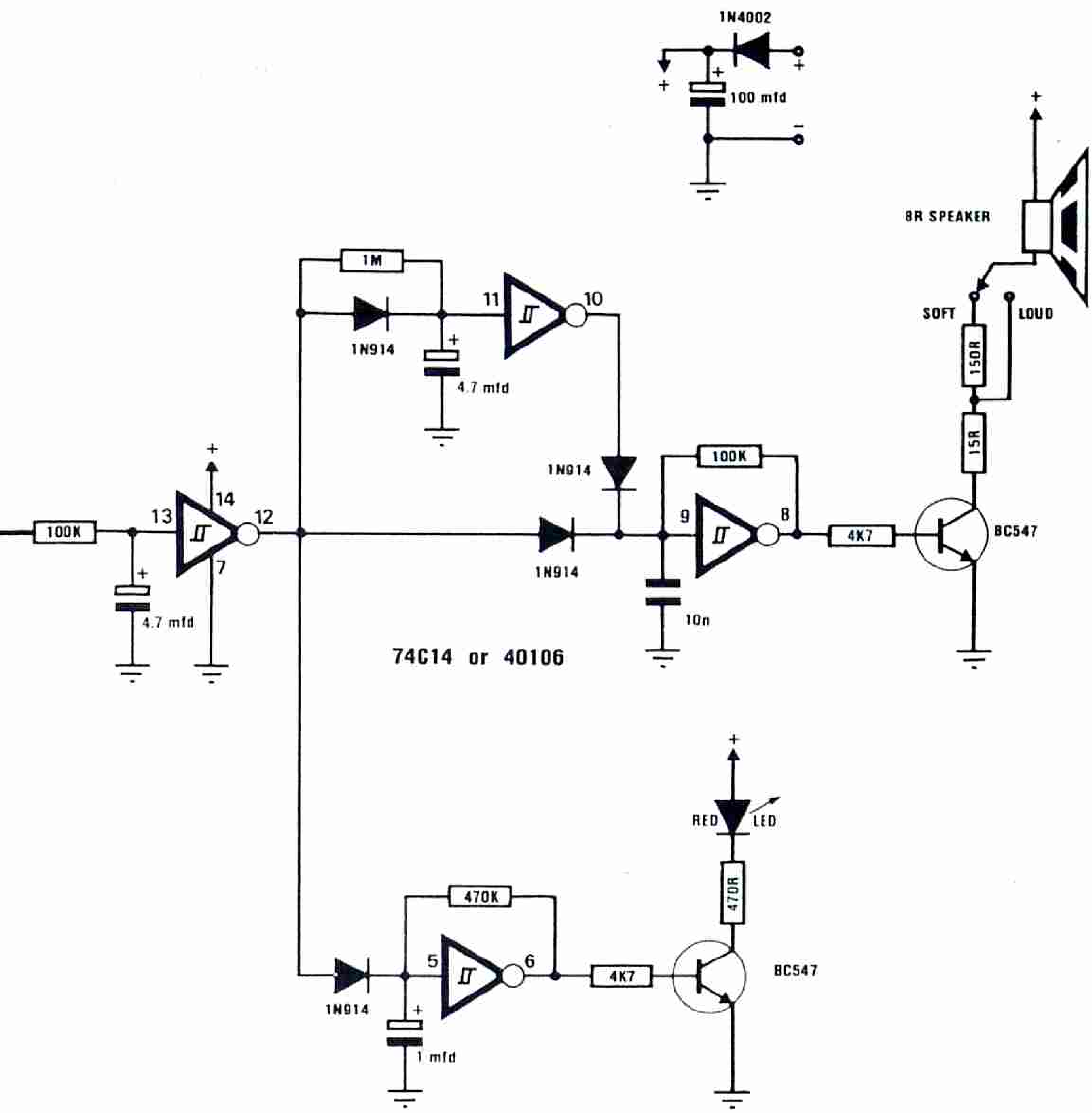
Starting at the top left-hand side of the circuit, we have two diodes detecting the high and low beams. These two diodes form an OR gate so that the circuit responds when either beam is in operation.

It should be noted that the high beam, low beam and ignition lines are ACTIVE HIGH lines. This means the circuit takes a HIGH signal from them as an indication that they are ON.

The output of this gate feeds an inverter so that the second NOR gate has one LOW line and one HIGH line when the circuit is detecting both ignition and beam inputs.

The output of the NOR gate will be LOW and basically this will inhibit the two output oscillators so that the speaker and red LED will not operate.

The only condition to bring this NOR gate circuit into operation is for the ignition to be switched OFF when the headlight line is ON. If this happens, both inputs to the NOR gate will be LOW and thus the output will be HIGH. This will charge the 4.7mfd electrolytic on pin 13 of the Schmitt Trigger and after a period of time, the output of this gate (pin 12) will go LOW.



REMINDER CIRCUIT

The LED oscillator circuit between pins 5 and 6 will immediately cause the indicator LED to flash and the oscillator between pins 9 and 8 will cause the speaker to sound.

The delay circuit made up of the Schmitt trigger between pins 11 and 10 will start timing as the 4.7mfd electrolytic on pin 11 is initially charged and thus keeps the output LOW. But when pin 12 goes LOW, the electrolytic begins to discharge through the 1M resistor and after a short period of time the output pin 10 changes to a HIGH and shuts the speaker OFF. The diode on pin 11 of the delay circuit allows the recharge-time for the delay-circuit electrolytic to be very short.

Next we go to the other input sensor, the light-sensing darlington transistor, and see how it fits into the circuit.

When light is falling on this sensor, it conducts and thus the voltage on the input of the Schmitt trigger between pins 1 and 2 is LOW and the output is HIGH.

This puts a HIGH on pin 2 of the NOR gate via a diode. Also connected to pin 2 of the NOR gate is another diode, which is connected to the ignition line via an inverter. These two diodes and the 2-input NOR gate effectively form a 3-input NOR gate.

The only time when the output of this gate (pin 3) will go HIGH is when the ignition is ON, the headlights are OFF and light is NOT falling on the photo-transistor. The signal from this gate is OR gated with the other gate system (pin 10 of the CD 4001) and fed to the alarm circuit.

The only circuit remaining to be covered is the oscillator between pins 3 and 4. This is the imitation burglar alarm warning indicator. It is in operation when the ignition is NOT on, and shuts OFF when the ignition is turned ON.

The signal diode on pin 3 creates the shut-off condition by preventing the 1mfd from charging more than .6v. This is too low to enable the Schmitt Trigger to change states.

As you can see, a lot of gating is required to fulfill the set of conditions and this has been very cheaply done with diodes and inverting gates.

The delay circuit made up of the Schmitt Trigger between pins 13 and 12 is designed to prevent the alarm going off when the lights are changed from one beam to the other or when shadows fall on the photo-cell, such as when driving under a bridge etc.

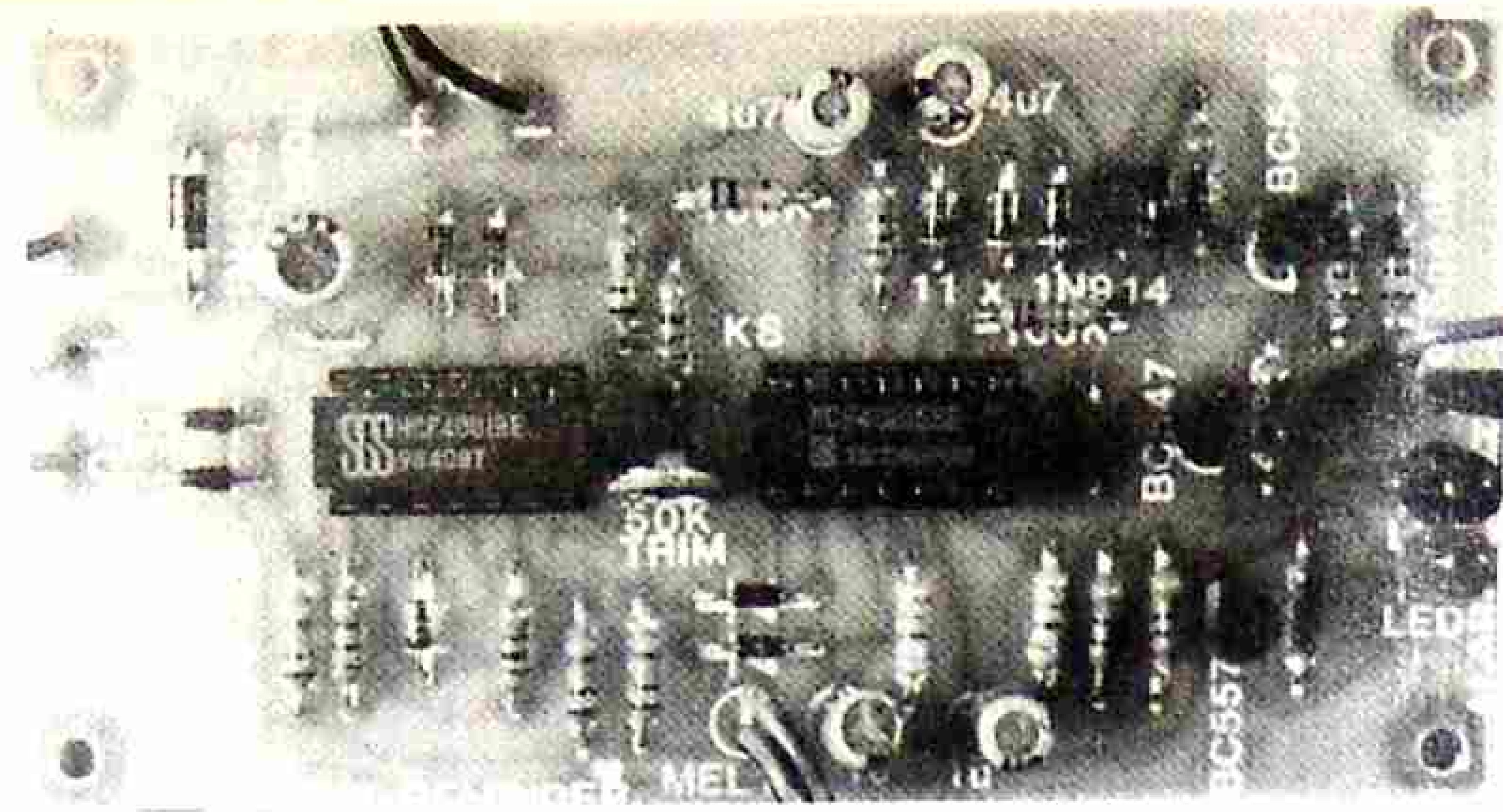
CONSTRUCTION

Like most TE projects, the Headlight Reminder PCB has a component overlay, making it possible to construct the project without even looking at the magazine.

Firstly solder all the diodes and resistors. One diode is a 1N 4002 so make sure you get it in the correct location. The holes for the trim pot may have to enlarged a little to take the legs.

Insert the IC sockets, making sure all the pins go down the holes in the PC. Next solder in the 10n greep cap, the electrolytics and the transistors. One transistor is a BC 557. This is soldered onto the board in the location marked by the solid white shape. The BC 547 transistors are soldered into the positions marked by an outline.

Enlarge the three holes marked 'L', 'S', and 'C'. Insert matrix pins into these holes. These three terminals are for the connection of the speaker. The hole marked 'C' is common and 'L' signifies Loud while 'S' indicates Soft. Connect the speaker to the pin best suited to your 'attention-getting'.



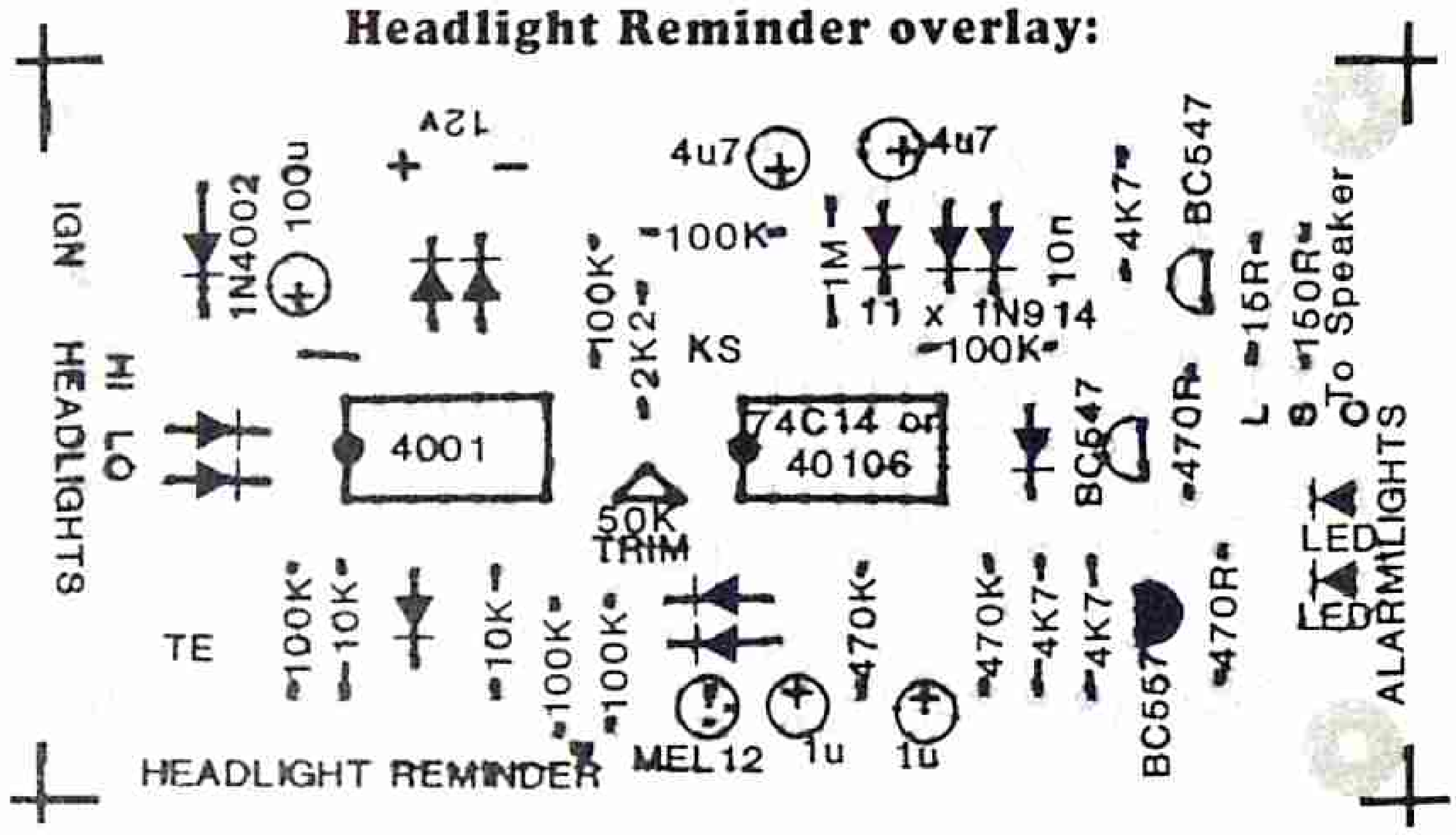
Complete unit showing parts and overlay.

Insert the IC's in the sockets making sure they are in the correct sockets and facing the right direction.

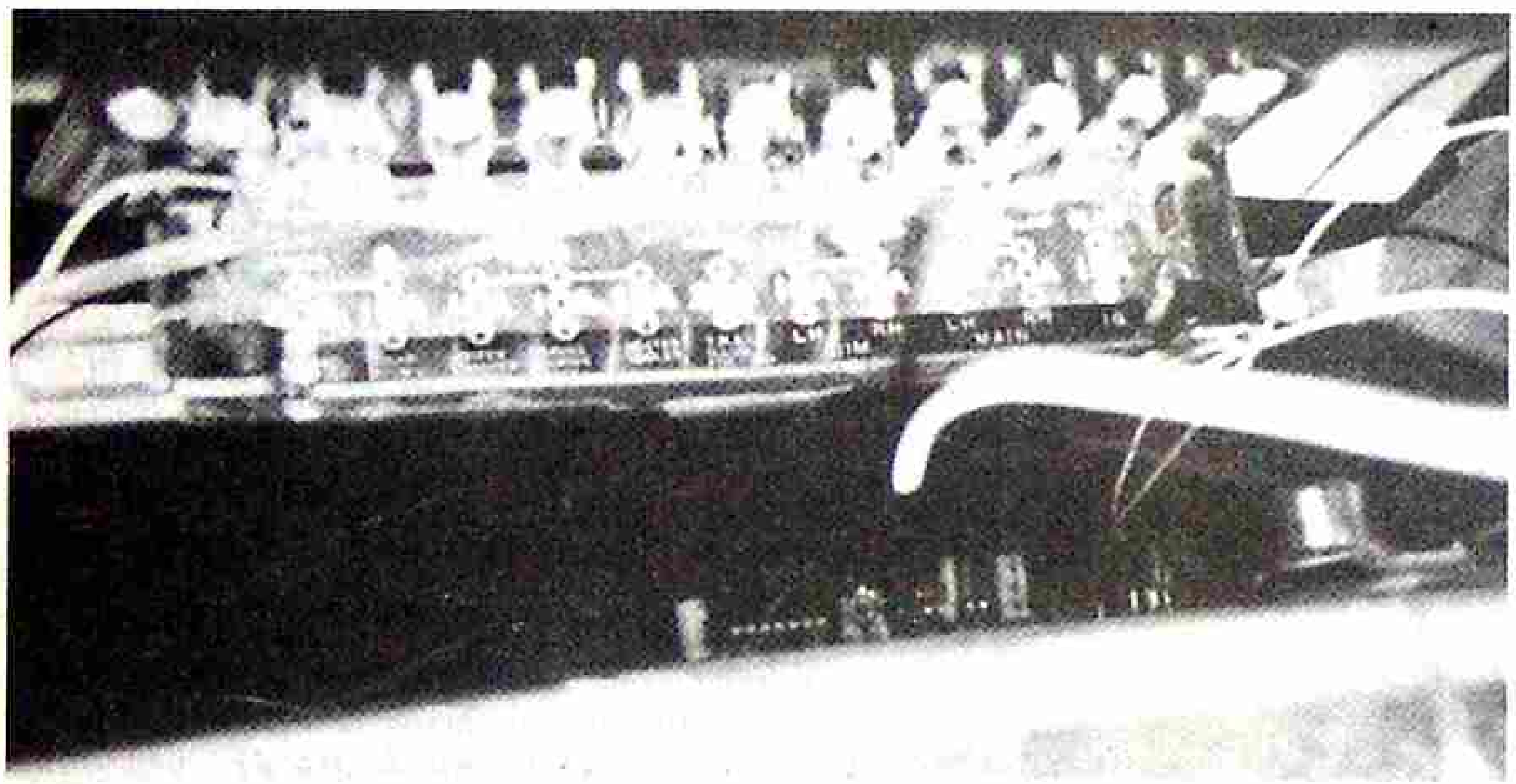
The two LEDs are mounted on the dash in a position that is visible to the driver. We used an orange LED for the pseudo alarm and a red LED for the 'lights on' reminder.

Mount the circuit board near the fuse box of the car. Wire the circuit to this box using lengths of hook-up flex.

Headlight Reminder overlay:

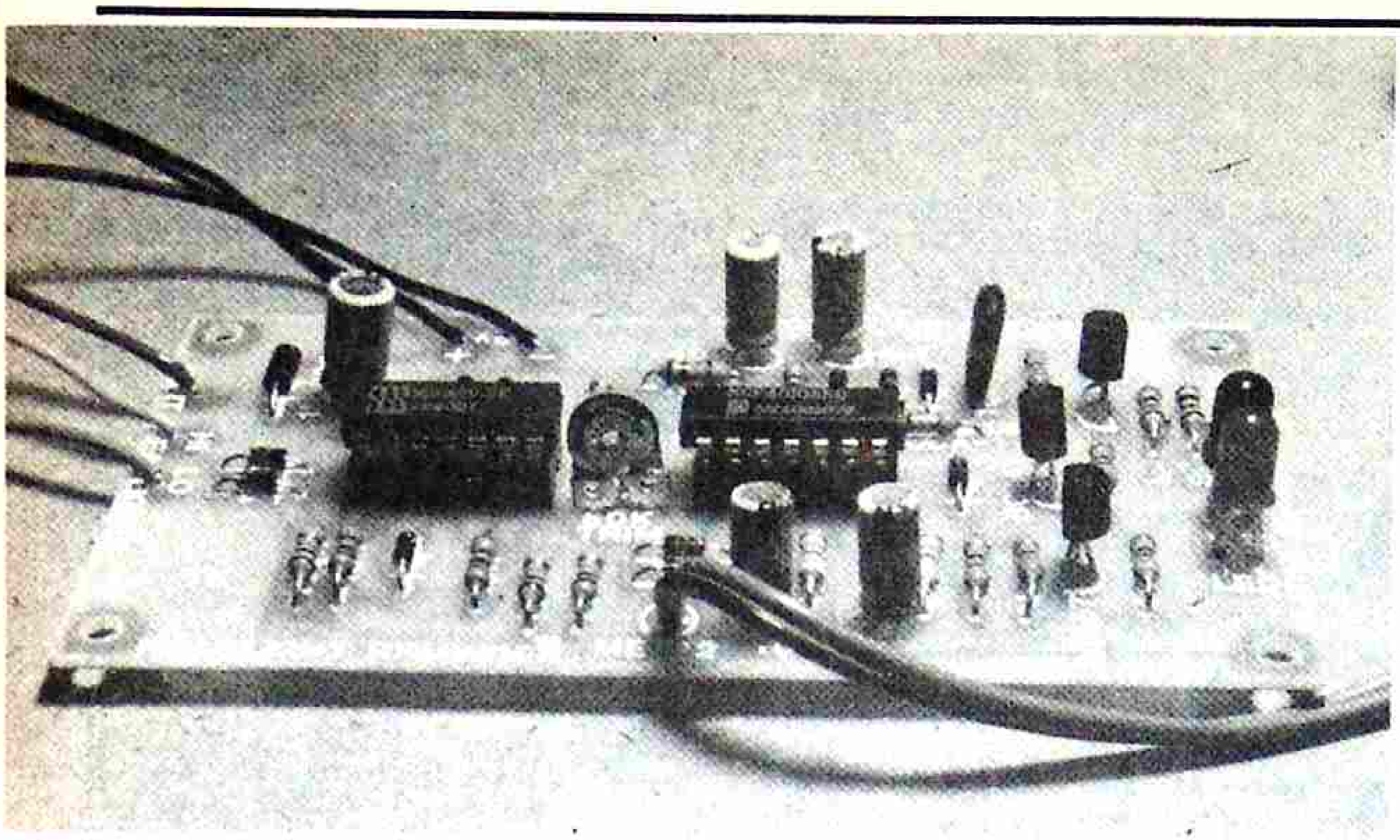


Connections under the dash:



You can make connection by twising the wire around the end of the fuse and pushing it back into the holder but the best way is to solder onto the fuse and fit them back into the holder.

Look for fuses marked Headlamp L.H. and L.D. (left high beam and left dim). Connect the headlamp inputs on the PCB to these fuses. Note that no connection has been made to the parking lamps. If you connect to these, the circuit will mistake the parkers for the headlamps on the second function of the unit, with the result that you could drive around with the parkers on and the head-lamps off, when it is dark, without the alarm sounding.



PARTS LIST:

1 - 15R ¼watt
1 - 150R
2 - 470R
1 - 2k2
3 - 4k7
2 - 10k
6 - 100k
2 - 470k
1 - 1M

1 - 50k mini trim pot

1 - 10n greencap

2 - 1mfd 25v PC electrolytics
2 - 4.7mfd 16v PC electrolytics
1 - 100mfd 16v PC electrolytic

11 - 1N 914 or 1N 4148 diodes
1 - 1N 4002 diode

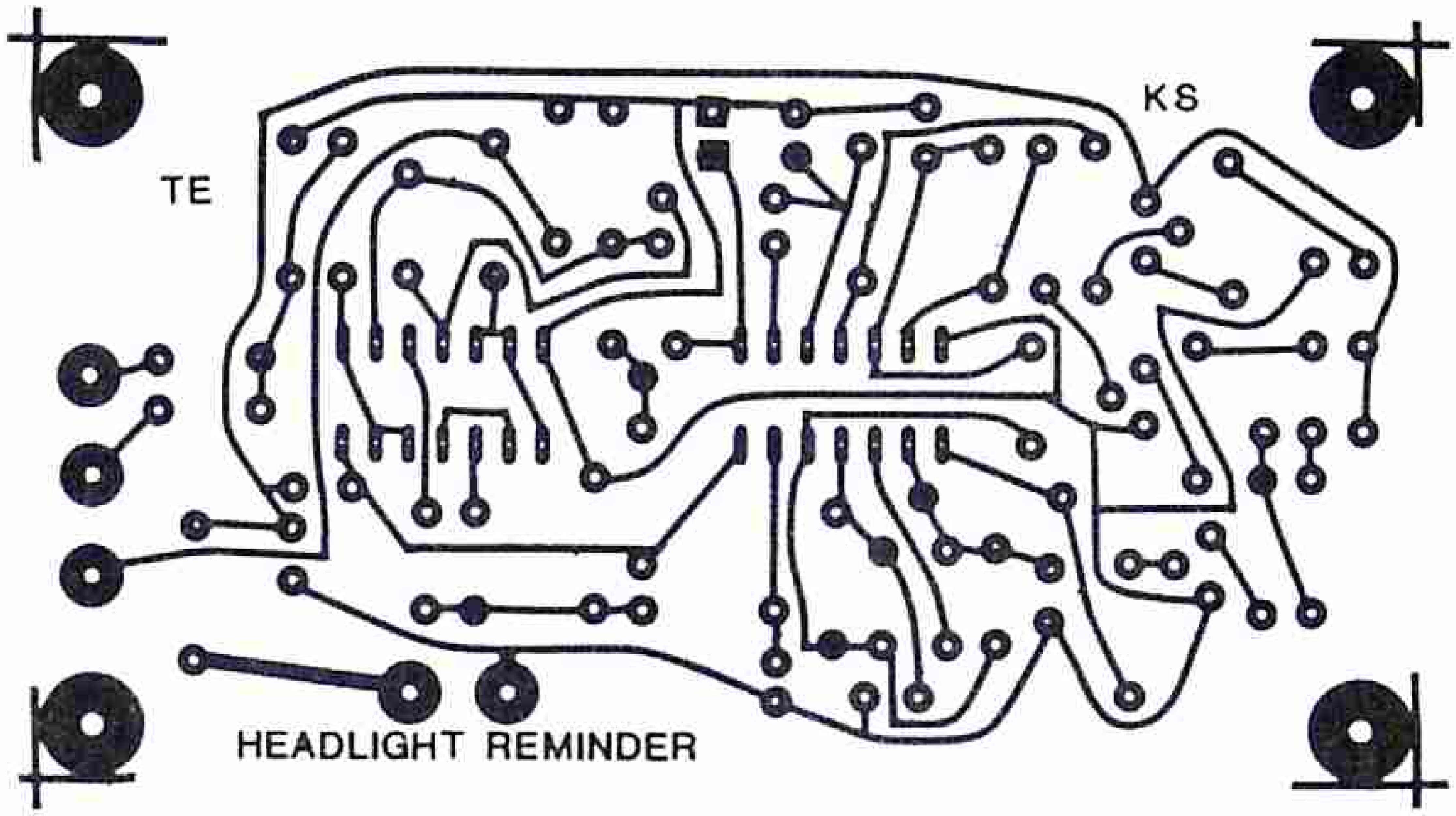
1 - 5mm red LED
1 - 5mm orange LED

2 - BC 547 transistors
1 - BC 557 transistor

1 - CD 4001 IC
1 - 74c14 or 40106 IC
1 - MEL-12 photo transistor

2 - 14 pin IC sockets
3 'Quick Connect' pins and connectors
1 - 8R speaker
Lengths of hook-up flex

1 - HEADLIGHT REMINDER PC



PC track-work.

The ignition line is taken from any appliance such as the heater or wind-screen wipers. You will find at least three suitable fuses.

The positive line can be taken from the 'hazard' lights, horn or cigarette lighter.

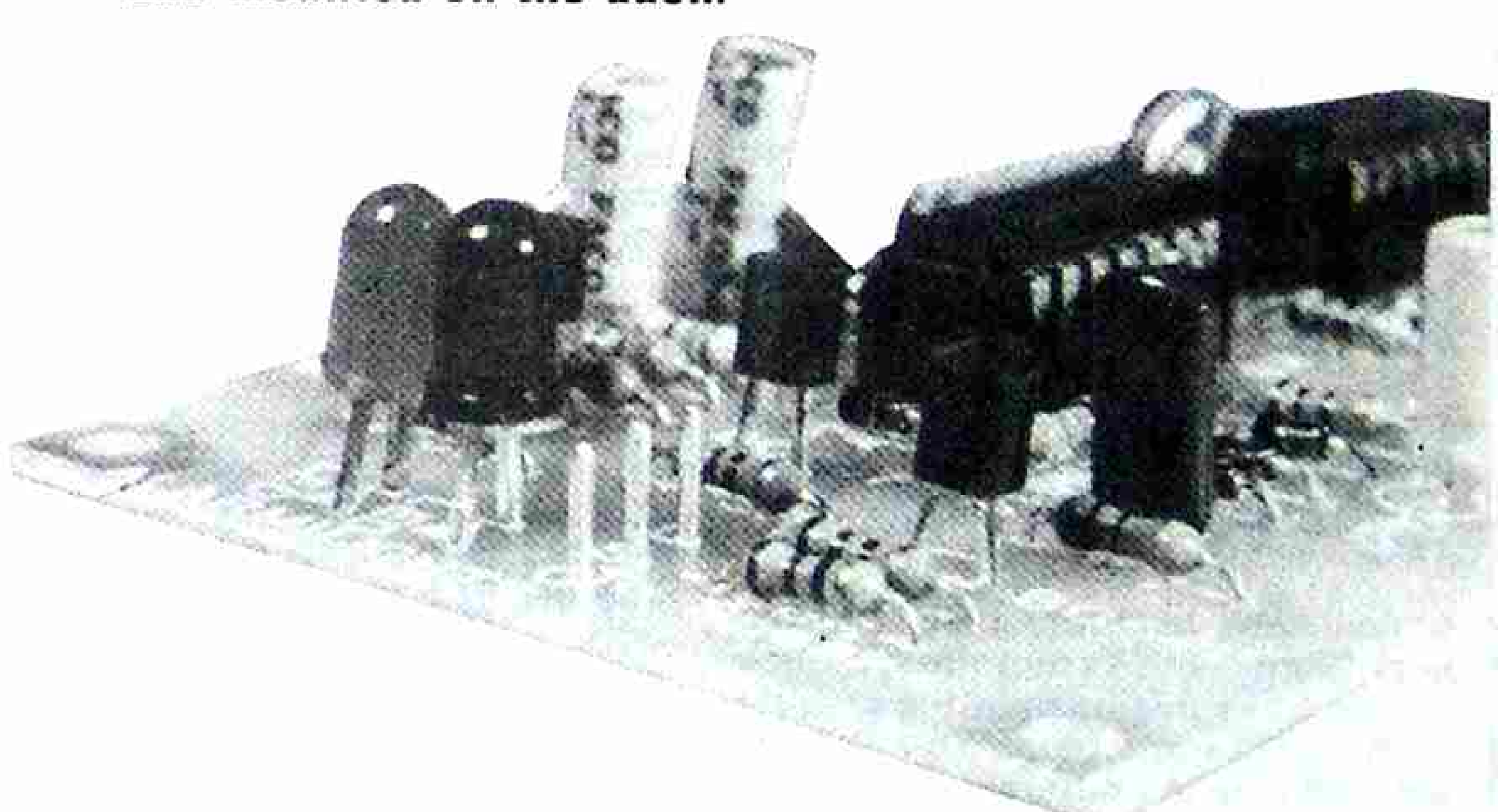
The earth line is simply bolted to any earthed part of the car.

The photo-transistor is mounted on the top of the dash, facing so that it is lit from the outside. Make sure you don't put the street directory on top of it or you will spend hours looking for the fault.

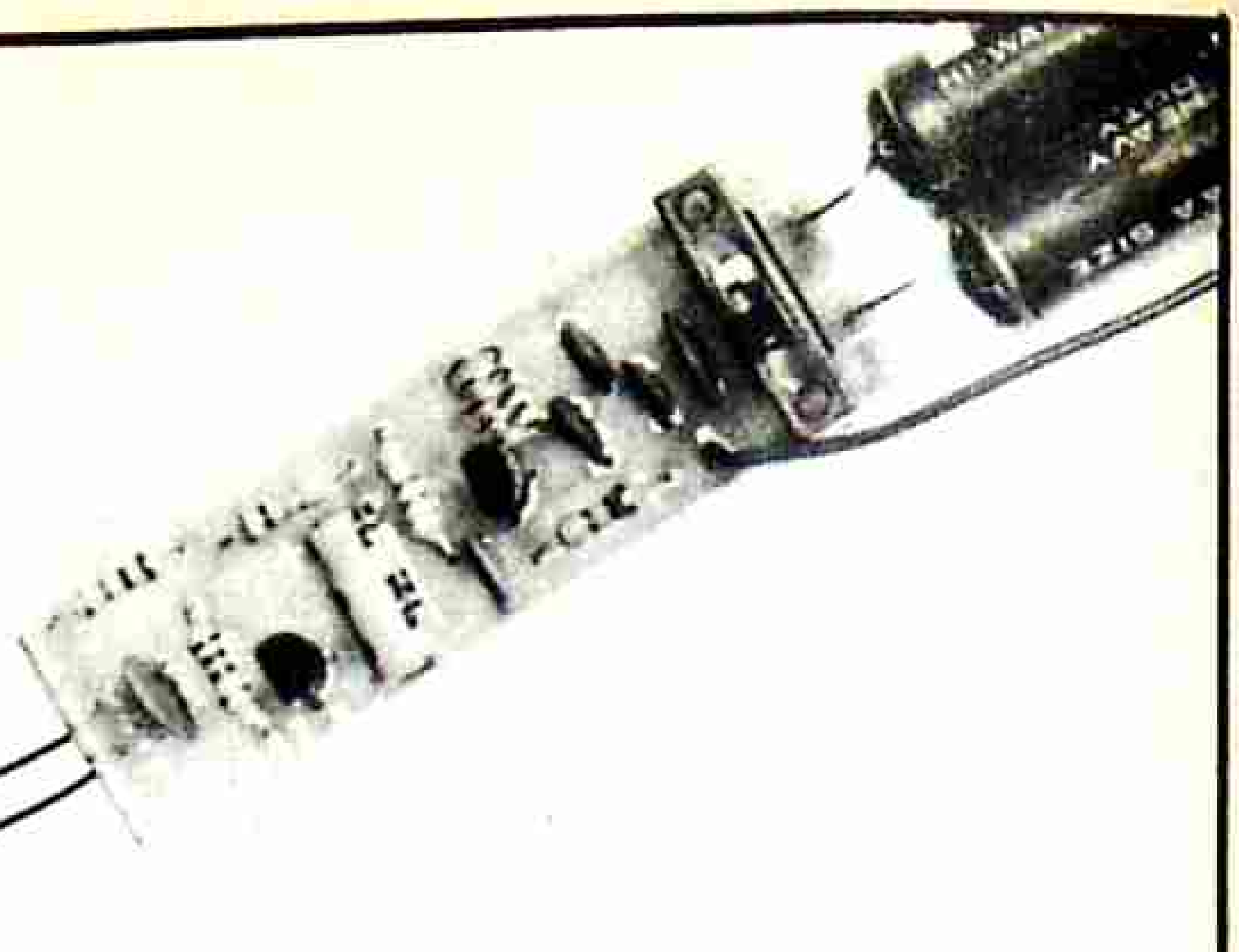
ALIGNMENT

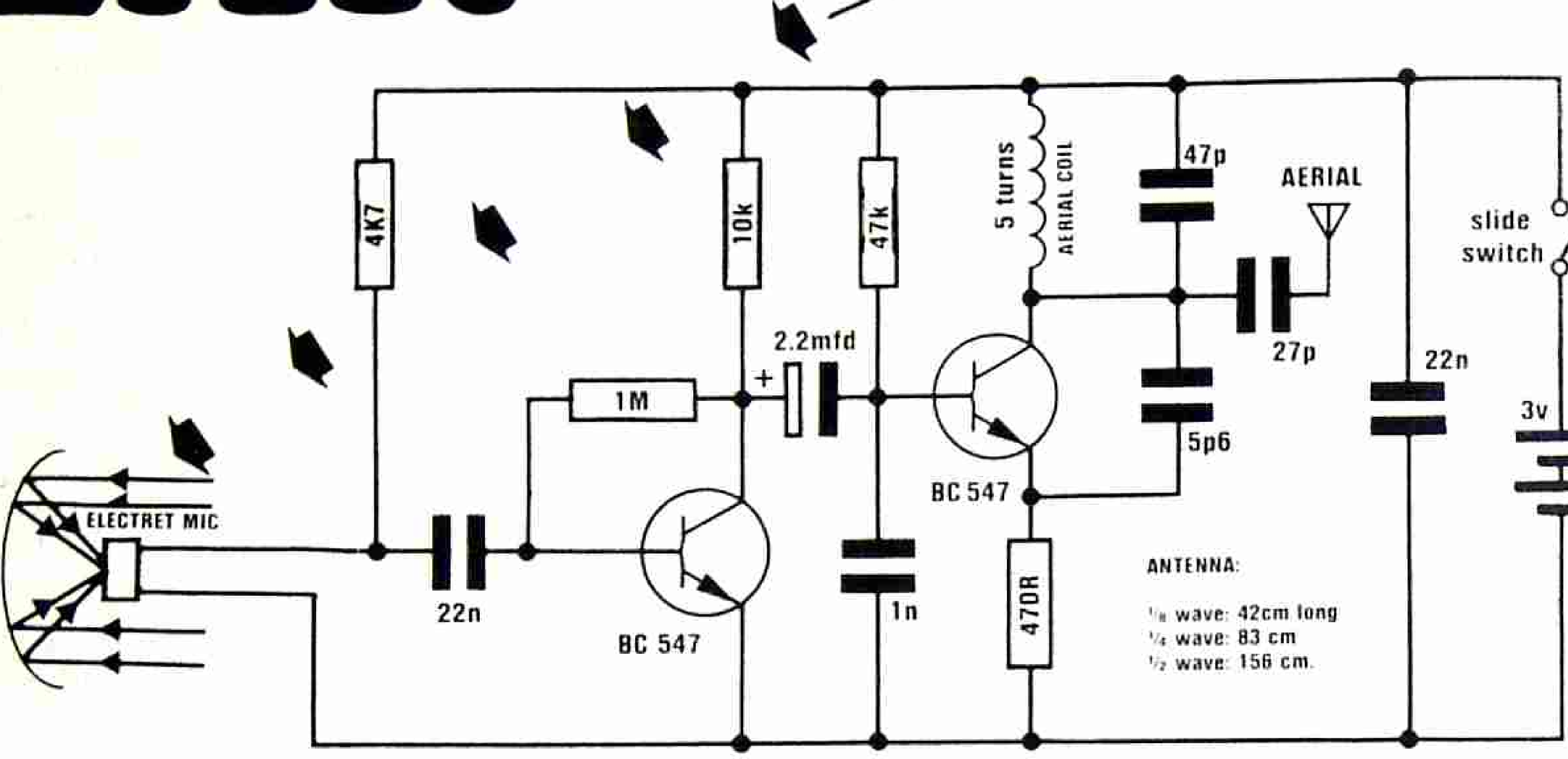
When it starts to get dark, start the car. Leave the headlights OFF. Now adjust the trim pot until the alarm sounds. The unit is now ready for use.

Close-up of the components on the PC. The 2 indicator LEDs should be connected to leads and mounted on the dash.



BIG EAR





The BIG EAR project is made up on the FM BUG PC. The only new arrangements are the 4k7 resistor in the circuit and the mounting of the electret microphone at the focus of a parabolic dish so that the sound-gathering capabilities are increased enormously.

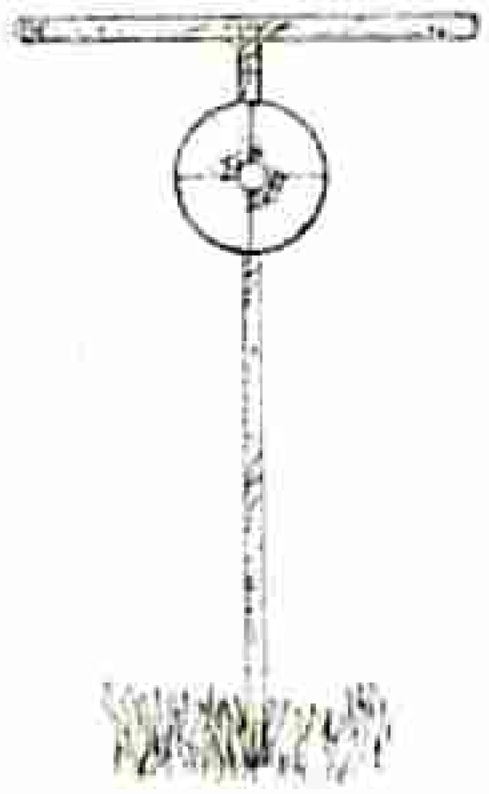
The FM BUG in issues 11, was one of our successes. Not only did it out-perform all the other kit-models on the market, but operated better than one fully built model costing as much as $49.95!

The only trouble we had was with the SGS transistors. The broad range of specifications within a batch meant that some of them failed to work at all. They failed only in the RF oscillator section and means that the high frequency is the limiting factor.

It seems incredible that a transistor of this type would fail at 100MHz but we had three reports that this was the case.

If you have built the BUG and not been happy with its performance, you should try replacing the RF transistor with a different type. You will be amazed with the difference it makes.

We have sold the BUG in bulk lots to schools as a first-year project. If you are not lucky enough to be in one of these schools, we suggest you look into building our super-sensitive version. It is called BIG EAR.



The BIG EAR mounted on a 3M pole with the sound-collecting reflecting dish.

PARTS LIST:

1 - 470R ¼watt
1 - 4k7
1 - 10k
1 - 47k
1 - 1M

1 - 5.6pf ceramic
1 - 22pf or 27pf or 33pf ceramic
1 - 47pf ceramic
1 - 1n ceramic
2 - 22n ceramic
1 2.2mfd 16v electrolytic

2 - BC 547 transistors
1 - mini slide switch
1 - electret microphone

2 - AAA cells
2 metres aerial wire

FM BUG PC BOARD.

Basically our BIG EAR is the FM BUG circuit as presented in issue 11 with one change to the circuit and one improvement to the layout.

Both these two improve the audio pick-up to a level where it is better than the human ear!

Everyone knows the effect of cupping your hands behind your ears to improve the reflector and thus increase the sound pick-up. The BIG EAR works in the same way.

A parabolic dish behind the microphone serves to reflect microscopic sounds to the focal point or FOCUS of the dish and this is where the electret microphone is situated.

Thus the capture area for the microphone is increased many times, making those elusive, minute sounds come into range.

We have received four requests along these same lines and I think it has wide possibilities.

One reader has a very large property and the front gate is about 250 metres from the house.

It seems a number of cars turn and/or park in front of the gate during the evening hours and it was needed to know if any of the passengers entered the property.

The FM BUG was mounted atop a 3 metre pole and arranged with a dipole antenna directed towards the house.

Two nicads and a set of solar cells were fitted to the top of the pole to power the unit and also charge the batteries. This provided power for continuous day/night operation.

This proved to be a great idea as the FM radio could be turned on whenever a threatening set of headlights was seen at the front gates.

The performance was as good as a land-line system and at only a fraction of the cost.

A small dish reflector concentrated the sounds so that the slightest whispers could be detected. This type of arrangement is trouble-free and requires no maintenance. It could be used in paddocks, out–buildings or even your neighbours house to detect any prowlers etc.

The same set-up would be a good idea for a security organisation. They could place a number of these units around a building and by tuning in an FM receiver, monitor each and every part of the building without having to go the high cost of wiring each unit.

Idea number two came from a reader requiring to tape bird calls and the general sounds in his garden.

He did not explain his reason for this but was very pleased with the clarity and range of the bug when transmitting back to his lounge room. From there he could monitor the sounds and tape them directly from the receiver.

Idea number three came from a piano player who wanted to tape his practicing. The piano was remote from his recorder and the BUG proved to be an ideal transmitting medium.

Again the clarity and performance was stated to be be well beyond a similar low-priced kit and even better than a $49.95 unit which was purchased in built-up form.

He was so impressed with the BUG that he bought another.

● ○ ● ○ ● ○ ● ○ ● ○

Picks up sounds you never thought existed!

● ○ ● ○ ● ○ ● ○ ● ○

The story from reader number four we cannot print. He used his bug to tape his friends!

So, you can see, the ideas are endless for such a compact unit and if you want to listen to something beyond ear-shot, this improved design is the answer.

Gino thought of it first. He added a reflector to the microphone on his BUG and reduced the load resistor to the electret microphone so that it would have increased sensitivity.

The idea actually came from a toy he bought a few years ago. It consisted of a long trumpet or cone containing a microphone. This fed into a two-transistor amplifier and powered a pair of headphones.

With the toy long since defunct, he used the tumpet to house the electret microphone and placed the BUG circuitry in the cavity left by the transistor amplifier.

The idea worked so well that he thought it would be a good extension for the FM BUG. So here it is.

The parabolic reflector in our design can be made from an aluminium pie dish. These can be purchased in packs of three from the super market or removed from your mum's hot pie!

The parabolic shape is created by pressing the dish into a bowl and rubbing with a wooden spoon. The formation needed is a fairly shallow parabola and without being too technical or critical, almost any curved shape will be acceptable.

Solder two long tinned copper wire leads to the microphone and attach the ends to the PC board. This will enable the mocrophone to be positioned at the focus of the dish to achieve maximum sensitivity.

The other improvement to the circuit is to reduce the load resistor to the electret microphone.

This will produce a slightly higher voltage across the microphone and increase the current through it. Inside the microphone is a field effect transistor with the gate lead connected to one plate of a capacitor. That's why only two leads emerge from the microphone.

The other plate of the capacitor is made of a very thin membrane which is a conducting plastic material.

When the sound waves enter the microphone they vibrate the thin membrane and the charge sitting on the two plates interact with each other. This modifies the charge on the gate lead of the FET. This is amplified by the FET and fed to the first BC 547 in the circuit.

By increasing the voltage across the microphone, the output swing will increase for the same sound level and thus fainter sounds will be able to be detected.

Obviously there is a limit to this and using a 4k7 resistor with a 3v supply will be about optimum.

The only other way to increase the sensitivity and range is to increase the supply voltage. By increasing this to 9v, an increased transmitting distance will be achieved with a corresponding increase in sensitivity.

Try experimenting with various values for each of the components in the circuit to determine the effect of each component on the overall performance.

High Frequency transmission is a very interesting field and one which could see a lot more use of the band.

Z80 Machine Codes FOR DISASSEMBLY

SHEET 12

This is a Z-80 MACHINE CODE disassembly table. Use it in conjunction with the Z-80 Machine Codes presented previously, for the creation of your own programs.

These lists make programming and disassembly easy. Fit them into a plastic sleeve and keep them handy.

00	NOP
01	LD BC,dddd
02	LD (BC),A
03	INC BC
04	INC B
05	DEC B
06	LD B,dd
07	RLCA
08	EX AF,A'F'
09	ADD HL,BC
0A	LD A,(BC)
0B	DEC BC
0C	INC C
0D	DEC C
0E	LD C,dd
0F	RRCA
10	DJNZ dis
11	LD DE,dddd
12	LD (DE),A
13	INC DE
14	INC D
15	DEC D
16	LD D,dd
17	RLA
18	JR dis
19	ADD HL,DE
1A	LD A,(DE)
1B	DEC DE
1C	INC E
1D	DEC E
1E	LD E,dd
1F	RRA
20	JR NZ dis
21	LD HL,dddd
22	LD (ADDR),HL
23	INC HL
24	INC H
25	DEC H
26	LD H,dd
27	DAA
28	JR Z,dis
29	ADD HL,HL
2A	LD HL,(ADDR)
2B	DEC HL
2C	INC L
2D	DEC L
2E	LD L,dd
2F	CPL
30	JR NC,dis
31	LD SP,dddd
32	LD (ADDR),A
33	INC SP
34	INC (HL)
35	DEC (HL)
36	LD (HL),dd
37	SCF
38	JR C,dis
39	ADD HL,SP
3A	LD A,(ADDR)
3B	DEC SP
3C	INC A
3D	DEC A
3E	LD A,dd
3F	CCF
40	LD B,B
41	LD B,C
42	LD B,D
43	LD B,E
44	LD B,H
45	LD B,L
46	LD B,(HL)
47	LD, B,A
48	LD C,B
49	LD C,C
4A	LD C,D
4B	LD C,E
4C	LD C,H
4D	LD C,L
4E	LD C,(HL)
4F	LD C,A
50	LD D,B
51	LD D,C
52	LD D,D
53	LD D,E
54	LD D,H
55	LD D,L
56	LD D,(HL)
57	LD D,A
58	LD E,B
59	LD E,C
5A	LD E,D
5B	LD E,E
5C	LD E,H
5D	LD E,L
5E	LD E,(HL)
5F	LD E,A
60	LD H,B
61	LD H,C

62	LD H,D
63	LD H,E
64	LD H,H
65	LD H,L
66	LD H,(HL)
67	LD H,A
68	LD L,B
69	LD L,C
6A	LD L,D
6B	LD L,E
6C	LD L,H
6D	LD L,L
6E	LD L,(HL)
6F	LD L,A
70	LD (HL),B
71	LD (HL),C
72	LD (HL),D
73	LD (HL),E
74	LD (HL),H
75	LD (HL),L
76	HALT
77	LD (HL),A
78	LD A,B
79	LD A,C
7A	LD A,D
7B	LD A,E
7C	LD A,H
7D	LD A,L
7E	LD A,(HL)
7F	LD A,A
80	ADD A,B
81	ADD A,C
82	ADD A,D
83	ADD A,E
84	ADD A,H
85	ADD A,L
86	ADD A,(HL)
87	ADD A,A
88	ADC A,B
89	ADC A,C
8A	ADC A,D
8B	ADC A,E
8C	ADC A,H
8D	ADC A,L
8E	ADC A,(HL)
8F	ADC A,A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
96	SUB (HL)
97	SUB A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
9E	SBC A,(HL)
9F	SBC A,A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L
A6	AND (HL)
A7	AND A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
AE	XOR (HL)
AF	XOR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
B6	OR (HL)
B7	OR A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H
BD	CP L
BE	CP (HL)
BF	CP A
C0	RET NZ
C1	POP BC
C2	JP NZ ADDR
C3	JP ADDR
C4	CALL NZ ADDR
C5	PUSH BC
C6	ADD A,dd
C7	RST 00
C8	RET Z
C9	RET
CA	JP Z ADDR
CB	★
CC	CALL Z,ADDR
CD	CALL ADDR
CE	ADC A,dd
CF	RST 08
D0	RET NC
D1	POP DE
D2	JP NC ADDR
D3	OUT port,A
D4	CALL NC ADDR
D5	PUSH DE
D6	SUB dd
D7	RST 10
D8	RET C
D9	EXX

DA	JP C ADDR
DB	IN A,port
DC	CALL C ADDR
DD	★
DE	SBC A,dd
DF	RST 18
E0	RET PO
E1	POP HL
E2	JP PO ADDR
E3	EX (SP),HL
E4	CALL PO ADDR
E5	PUSH HL
E6	AND dd
E7	RST 20
E8	RET PE
E9	JP (HL)
EA	JP PE ADDR
EB	EX DE,HL
EC	CALL PE ADDR
ED	★
EE	XOR dd
EF	RST 28
F0	RET P
F1	POP AF
F2	JP P ADDR
F3	DI
F4	CALL P ADDR
F5	PUSH AF
F6	OR dd
F7	RST 30
F8	RET M
F9	LD SP,HL
FA	JP M ADDR
FB	EI
FC	CALL M ADDR
FD	★
FE	CP dd
FF	RST 38
CB 00	RLC B
CB 01	RLC C
CB 02	RLC D
CB 03	RLC E
CB 04	RLC H
CB 05	RLC L
CB 06	RLC (HL)
CB 07	RLC A
CB 08	RRC B
CB 09	RRC C
CB 0A	RRC D
CB 0B	RRC E
CB 0C	RRC H
CB 0D	RRC L
CB 0E	RRC (HL)
CB 0F	RRC A
CB 10	RL B
CB 11	RL C
CB 12	RL D
CB 13	RL E
CB 14	RL H
CB 15	RL L
CB 16	RL (HL)
CB 17	RL A
CB 18	RR B
CB 19	RR C
CB 1A	RR D
CB 1B	RR E
CB 1C	RR H
CB 1D	RR L
CB 1E	RR (HL)
CB 1F	RR A
CB 20	SLA B
CB 21	SLA C
CB 22	SLA D
CB 23	SLA E
CB 24	SLA H
CB 25	SLA L
CB 26	SLA (HL)
CB 27	SLA A
CB 28	SRA B
CB 29	SRA C
CB 2A	SRA D
CB 2B	SRA E
CB 2C	SRA H
CB 2D	SRA L
CB 2E	SRA (HL)
CB 2F	SRA A
CB 38	SRL B
CB 39	SRL C
CB 3A	SRL D
CB 3B	SRL E
CB 3C	SRL H
CB 3C	SRL L
CB 3E	SRL (HL)
CB 3F	SRL A
CB 40	BIT 0,B
CB 41	BIT 0,C
CB 42	BIT 0,D
CB 43	BIT 0,E
CB 44	BIT 0,H
CB 45	BIT 0,L
CB 46	BIT 0,(HL)
CB 47	BIT 0,A
CB 48	BIT 1,B
CB 49	BIT 1,C
CB 4A	BIT 1,D
CB 4B	BIT 1,E
CB 4C	BIT 1,H
CB 4D	BIT 1,L
CB 4E	BIT 1,(HL)
CB 4F	BIT 1,A
CB 50	BIT 2,B
CB 51	BIT 2,C
CB 52	BIT 2,D
CB 53	BIT 2,E
CB 54	BIT 2,H
CB 55	BIT 2,L
CB 56	BIT 2,(HL)
CB 57	BIT 2,A
CB 58	BIT 3,B
CB 59	BIT 3,C
CB 5A	BIT 3,D

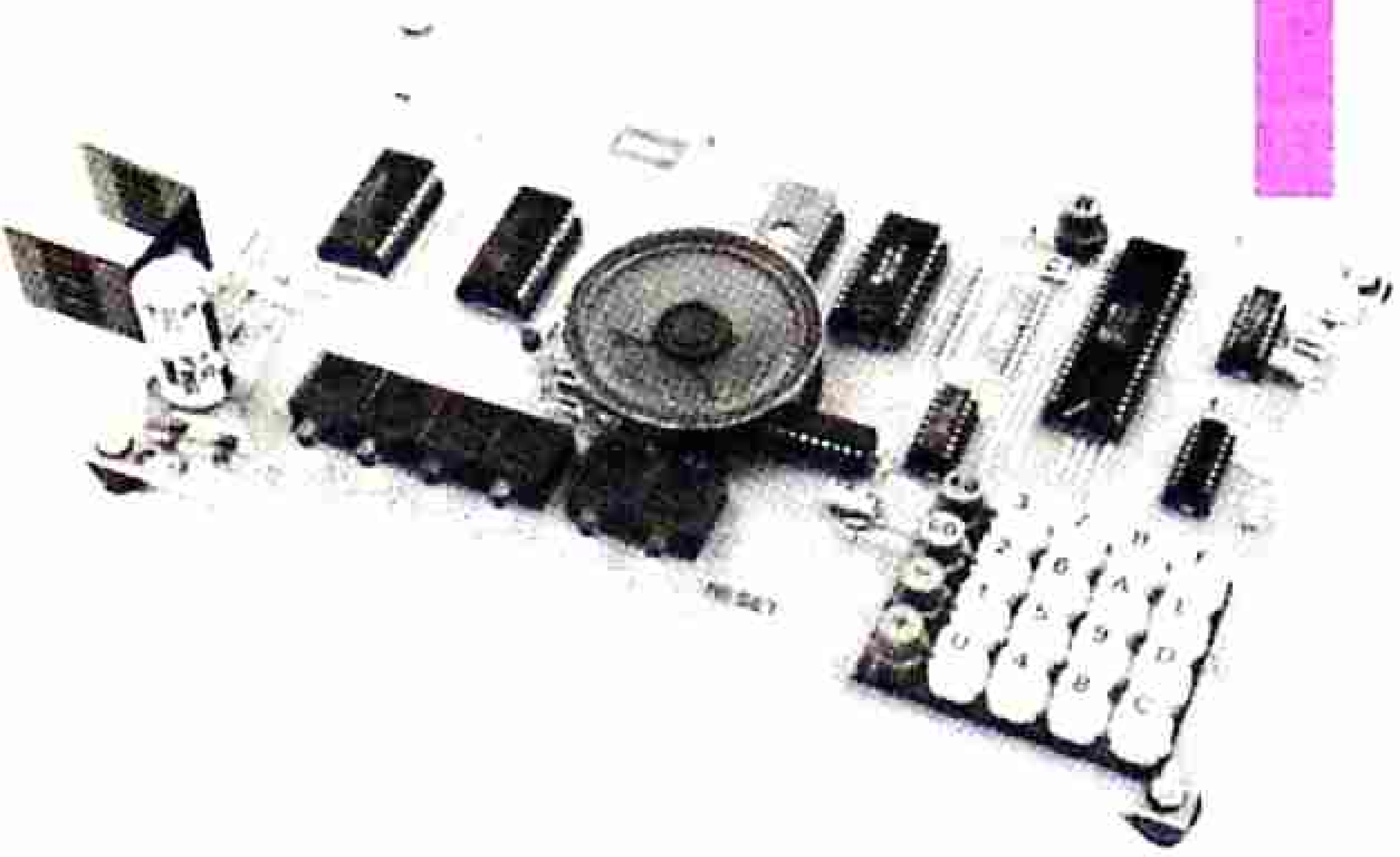
CB 5B	BIT 3,E
CB 5C	BIT 3,H
CB 5D	BIT 3,L
CB 5E	BIT 3,(HL)
CB 5F	BIT 3,A
CB 60	BIT 4,B
CB 61	BIT 4,C
CB 62	BIT 4,D
CB 63	BIT 4,E
CB 64	BIT 4,H
CB 65	BIT 4,L
CB 66	BIT 4,(HL)
CB 67	BIT 4,A
CB 68	BIT 5,B
CB 69	BIT 5,C
CB 6A	BIT 5,D
CB 6B	BIT 5,E
CB 6C	BIT 5,H
CB 6D	BIT 5,L
CB 6E	BIT 5,(HL)
CB 6F	BIT 5,A
CB 70	BIT 6,B
CB 71	BIT 6,C
CB 72	BIT 6,D
CB 73	BIT 6,E
CB 74	BIT 6,H
CB 75	BIT 6,L
CB 76	BIT 6,(HL)
CB 77	BIT 6,A
CB 78	BIT 7,B
CB 79	BIT 7,C
CB 7A	BIT 7,D
CB 7B	BIT 7,E
CB 7C	BIT 7,H
CB 7D	BIT 7,L
CB 7E	BIT 7,(HL)
CB 7F	BIT 7,A
CB 80	RES 0,B
CB 81	RES 0,C
CB 82	RES 0,D
CB 83	RES 0,E
CB 84	RES 0,H
CB 85	RES 0,L
CB 86	RES 0,(HL)
CB 87	RES 0,A
CB 88	RES 1,B
CB 89	RES 1,C
CB 8A	RES 1,D
CB 8B	RES 1,E
CB 8C	RES 1,H
CB 8D	RES 1,L
CB 8E	RES 1,(HL)
CB 8F	RES 1,A
CB 90	RES 2,B
CB 91	RES 2,C
CB 92	RES 2,D
CB 93	RES 2,E
CB 94	RES 2,H
CB 95	RES 2,L
CB 96	RES 2,(HL)
CB 97	RES 2,A
CB 98	RES 3,B
CB 99	RES 3,C
CB 9A	RES 3,D
CB 9B	RES 3,E
CB 9C	RES 3,H
CB 9D	RES 3,L
CB 9E	RES 3,(HL)
CB 9F	RES 3,A
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 4,L
CB A6	RES 4,(HL)
CB A7	RES 4,A
CB A8	RES 5,B
CB A9	RES 5,C
CB AA	RES 5,D
CB AB	RES 5,E
CB AC	RES 5,H
CB AD	RES 5,L
CB AE	RES 5,(HL)
CB AF	RES 5,A
CB B0	RES 6,B
CB B1	RES 6,C
CB B2	RES 6,D
CB B3	RES 6,E
CB B4	RES 6,H
CB B5	RES 6,L
CB B6	RES 6,(HL)
CB B7	RES 6,A
CB B8	RES 7,B
CB B9	RES 7,C
CB BA	RES 7,D
CB BB	RES 7,E
CB BC	RES 7,H
CB BD	RES 7,L
CB BE	RES 7,(HL)
CB BF	RES 7,A
CB C0	SET 0,B
CB C1	SET 0,C
CB C2	SET 0,D
CB C3	SET 0,E
CB C4	SET 0,H
CB C5	SET 0,L
CB C6	SET 0,(HL)
CB C7	SET 0,A
CB C8	SET 1,B
CB C9	SET 1,C
CB CA	SET 1,D
CB CB	SET 1,E
CB CC	SET 1,H
CB CD	SET 1,L
CB CE	SET 1,(HL)
CB CF	SET 1,A
CB D0	SET 2,B
CB D1	SET 2,C
CB D2	SET 2,D
CB D3	SET 2,E

CB D4	SET 2,H
CB D5	SET 2,L
CB D6	SET 2,(HL)
CB D7	SET 2,A
CB D8	SET 3,B
CB D9	SET 3,C
CB DA	SET 3,D
CB DB	SET 3,E
CB DC	SET 3,H
CB DD	SET 3,L
CB DE	SET 3,(HL)
CB DF	SET 3,A
CB E0	SET 4,B
CB E1	SET 4,C
CB E2	SET 4,D
CB E3	SET 4,E
CB E4	SET 4,H
CB E5	SET 4,L
CB E6	SET 4,(HL)
CB E7	SET 4,A
CB E8	SET 5,B
CB E9	SET 5,C
CB EA	SET 5,D
CB EB	SET 5,E
CB EC	SET 5,H
CB ED	SET 5,L
CB EE	SET 5,(HL)
CB EF	SET 5,A
CB F0	SET 6,B
CB F1	SET 6,C
CB F2	SET 6,D
CB F3	SET 6,E
CB F4	SET 6,H
CB F5	SET 6,L
CB F6	SET 6,(HL)
CB F7	SET 6,A
CB F8	SET 7,B
CB F9	SET 7,C
CB FA	SET 7,D
CB FB	SET 7,E
CB FC	SET 7,H
CB FD	SET 7,L
CB FE	SET 7,(HL)
CB FF	SET 7,A
DD 09	ADD IX,BC
DD 19	ADD IX,DE
DD 21	LD IX,dddd
DD 22	LD (ADDR),IX
DD 23	INC IX
DD 29	ADD IX,IX
DD 2A	LD IX,(ADDR)
DD 2B	DEC IX
DD 34	INC,(IX + dis)
DD 35	DEC,(IX + dis)
DD 36	LD (IX + dis)dd
DD 39	ADD IX,SP
DD 46	LD B,(IX + dis)
DD 4E	LD C,(IX + dis)
DD 56	LD D,(IX + dis)
DD 5E	LD E,(IX + dis)
DD 66	LD H,(IX + dis)
DD 6E	LD L,(IX + dis)
DD 70	LD (IX + dis),B
DD 71	LD (IX + dis),C
DD 72	LD (IX + dis),D
DD 73	LD (IX + dis),E
DD 74	LD (IX + dis),H
DD 75	LD (IX + dis),L
DD 77	LD (IX + dis),A
DD 7E	LD A,(IX + dis)
DD 86	ADD A,(IX + dis)
DD 8E	ADC A,(IX + dis)
DD 96	SUB (IX + dis)
DD 9E	SBC A,(IX + dis)
DD A6	AND (IX + dis)
DD AE	XOR (IX + dis)
DD B6	OR (IX + dis)
DD BE	CP (IX + dis)
DD CB XX 06	RLC (IX + dis)
DD CB XX 0E	RRC (IX + dis)
DD CB XX 16	RL (IX + dis)
DD CB XX 1E	RR (IX + dis)
DD CB XX 26	SLA (IX + dis)
DD CB XX 2E	SRA (IX + dis)
DD CB XX 3E	SRL (IX + dis)
DD CB XX 46	BIT 0,(IX + dis)
DD CB XX 4E	BIT 1,(IX + dis)
DD CB XX 56	BIT 2,(IX + dis)
DD CB XX 5E	BIT 3,(IX + dis)
DD CB XX 66	BIT 4,(IX + dis)
DD CB XX 6E	BIT 5,(IX + dis)
DD CB XX 76	BIT 6,(IX + dis)
DD CB XX 7E	BIT 7,(IX + dis)
DD CB XX 86	RES 0,(IX + dis)
DD CB XX 8E	RES 1,(IX + dis)
DD CB XX 96	RES 2,(IX + dis)
DD CB XX 9E	RES 3,(IX + dis)
DD CB XX A6	RES 4,(IX + dis)
DD CB XX AE	RES 5,(IX + dis)
DD CB XX B6	RES 6,(IX + dis)
DD CB XX BE	RES 7,(IX + dis)
DD CB XX C6	SET 0,(IX + dis)
DD CB XX CE	SET 1,(IX + dis)
DD CB XX D6	SET 2,(IX + dis)
DD CB XX DE	SET 3,(IX + dis)
DD CB XX E6	SET 4,(IX + dis)
DD CB XX EE	SET 5,(IX + dis)
DD CB XX F6	SET 6,(IX + dis)
DD CB XX FE	SET 7,(IX + dis)
DD E1	POP IX
DD E3	EX (SP),IX
DD E5	PUSH IX
DD E9	JP (IX)
DD F9	LD SP,IX
ED 40	IN B,(C)
ED 41	OUT (C),B
ED 42	SBC HL,BC
ED 43	LD (ADDR),BC
ED 44	NEG
ED 45	RETN
ED 46	IM 0

ED 47	LD I,A
ED 48	IN C,(C)
ED 49	OUT (C),C
ED 4A	ADC HL,BC
ED 4B	LD BC,(ADDR)
ED 4D	RETI
ED 4F	LD R,A
ED 50	IN D,(C)
ED 51	OUT (C),D
ED 52	SBC HL,DE
ED 53	LD (ADDR),DE
ED 56	IM 1
ED 57	LD A,I
ED 58	IN E,(C)
ED 59	OUT (C),E
ED 5A	ADC HL,DE
ED 5B	LD DE,(ADDR)
ED 5E	IM 2
ED 5F	LD A,R
ED 60	IN H,(C)
ED 61	OUT (C),H
ED 62	SBC HL,HL
ED 63	LD (ADDR),HL
ED 67	RRD
ED 68	IN L,(C)
ED 69	OUT (C),L
ED 6A	ADC HL,HL
ED 6B	LD HL,(ADDR)
ED 6F	RLD
ED 72	SBC HL,SP
ED 73	LD (ADDR),SP
ED 78	IN A,(C)
ED 79	OUT (C),A
ED 7A	ADC HL,SP
ED 7B	LD SP,(ADDR)
ED A0	LDI
ED A1	CPI
ED A2	INI
ED A3	OUTI
ED A8	LDD
ED A9	CPD
ED AA	IND
ED AB	OUTD
ED B0	LDIR
ED B1	CPIR
ED B2	INIR
ED B3	OTIR
ED B8	LDDR
ED B9	CPDR
ED BA	INDR
ED BB	OTDR
FD 09	ADD IY,BC
FD 19	ADD IY,DE
FD 21	LD IY,dddd
FD 22	LD (ADDR),IY
FD 23	INC IY
FD 29	ADD IY,IY
FD 2A	LD IY,(ADDR)
FD 2B	DEC IY
FD 34	INC,(IY + dis)
FD 35	DEC,(IY + dis)
FD 36	LD (IY + dis)dd
FD 39	ADD IY,SP
FD 46	LD B,(IY + dis)
FD 4E	LD C,(IY + dis)
FD 56	LD D,(IY + dis)
FD 5E	LD E,(IY + dis)
FD 66	LD H,(IY + dis)
FD 6E	LD L,(IY + dis)
FD 70	LD (IY + dis),B
FD 71	LD (IY + dis),C
FD 72	LD (IY + dis),D
FD 73	LD (IY + dis),E
FD 74	LD (IY + dis),H
FD 75	LD (IY + dis),L
FD 77	LD (IY + dis),A
FD 7E	LD A,(IY + dis)
FD 86	ADD A,(IY + dis)
FD 8E	ADC A,(IY + dis)
FD 96	SUB (IY + dis)
FD 9E	SBC A,(IY + dis)
FD A6	AND (IY + dis)
FD AE	XOR (IY + dis)
FD B6	OR (IY + dis)
FD BE	CP (IY + dis)
FD CB XX 06	RLC (IY + dis)
FD CB XX 0E	RRC (IY + dis)
FD CB XX 16	RL (IY + dis)
FD CB XX 1E	RR (IY + dis)
FD CB XX 26	SLA (IY + dis)
FD CB XX 2E	SRA (IY + dis)
FD CB XX 3E	SRL (IY + dis)
FD CB XX 46	BIT 0,(IY + dis)
FD CB XX 4E	BIT 1,(IY + dis)
FD CB XX 56	BIT 2,(IY + dis)
FD CB XX 5E	BIT 3,(IY + dis)
FD CB XX 66	BIT 4,(IY + dis)
FD CB XX 6E	BIT 5,(IY + dis)
FD CB XX 76	BIT 6,(IY + dis)
FD CB XX 7E	BIT 7,(IY + dis)
FD CB XX 86	RES 0,(IY + dis)
FD CB XX 8E	RES 1,(IY + dis)
FD CB XX 96	RES 2,(IY + dis)
FD CB XX 9E	RES 3,(IY + dis)
FD CB XX A6	RES 4,(IY + dis)
FD CB XX AE	RES 5,(IY + dis)
FD CB XX B6	RES 6,(IY + dis)
FD CB XX BE	RES 7,(IY + dis)
FD CB XX C6	SET 0,(IY + dis)
FD CB XX CE	SET 1,(IY + dis)
FD CB XX D6	SET 2,(IY + dis)
FD CB XX DE	SET 3,(IY + dis)
FD CB XX E6	SET 4,(IY + dis)
FD CB XX EE	SET 5,(IY + dis)
FD CB XX F6	SET 6,(IY + dis)
FD CB XX FE	SET 7,(IY + dis)
FD E1	POP IY
FD E3	EX (SP),IY
FD E5	PUSH IY
FD E9	JP (IY)
FD F9	LD SP,IY

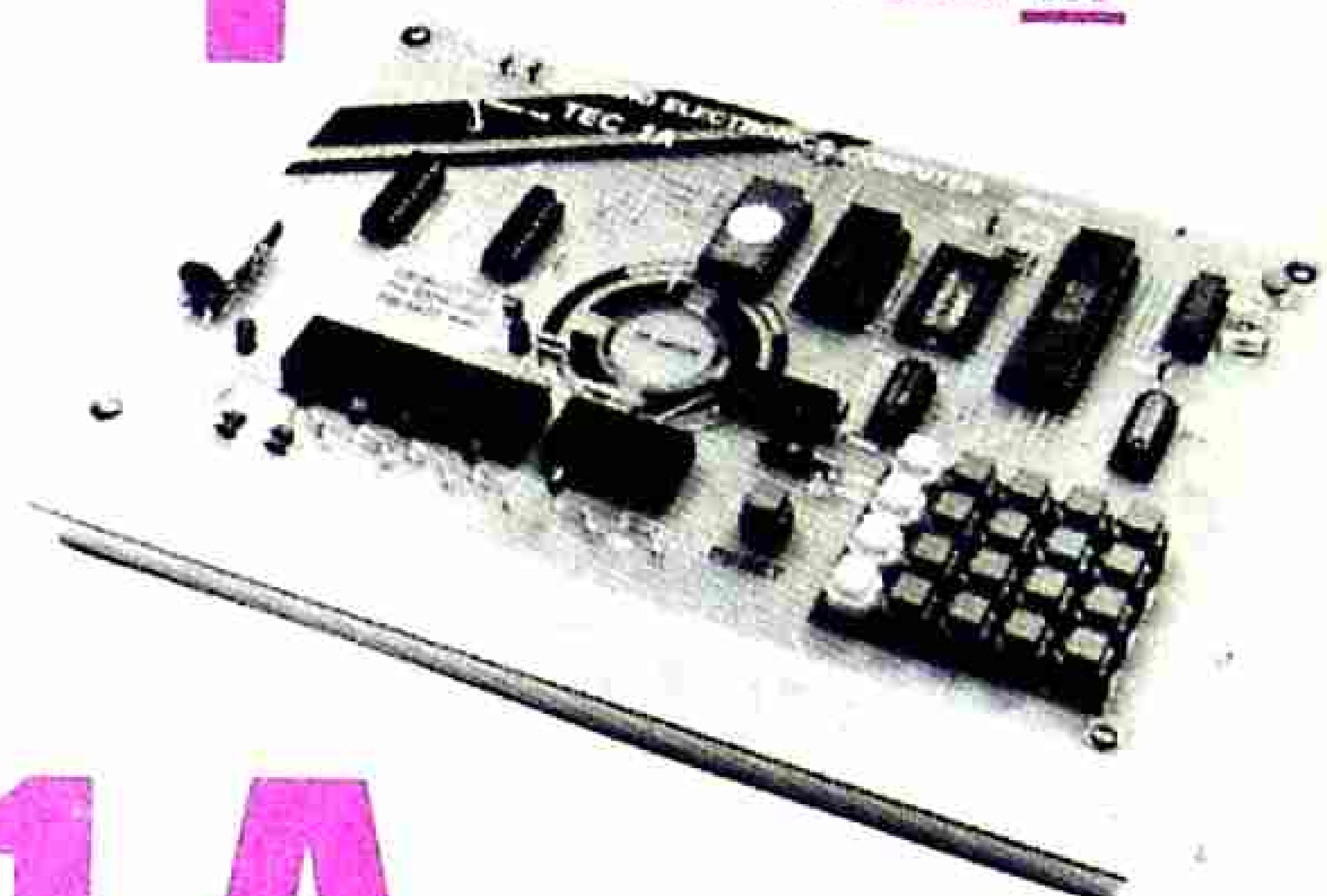
TALKING ELECTRONICS COMPUTER

TEC-1



PART III

&



TEC-1A

The TEC-1A is an update of the TEC-1. For all programming, both are the same. Only TEC-1A kits are available from the kit suppliers.

FEATURES IN THIS ISSUE:

★ Programs for the 7-segment display.
★ Programs for SOUND and TONES
★ Programs for the 8x8
★ RAM STACK Project
★ Interface for PRINTER: PLOTTER

- Text mode
- Graphics mode
- Computer Graphics.

Parts: $74.00
PC board: $21.00
Case: $19.00
Post: $5.00 MAX.
'Add-ons' as required . . .

With over 1,000 TEC's in the field, we have had a lot of feedback on the success of this project.

Most of them worked first go and this was very encouraging, considering the complexity of the unit.

Five schools bought class sets and this included a University in New Zealand.

More schools and TAFE colleges will be including them in their microprocessor courses and we will see more of this type of demonstrator in the near future.

It's undeniable that the Z80 is one of the most brilliantly designed microprocessors and it is unfortunate that so little has been written about it.

This is most probably due to the designers and manufacturers of the chip. Until recently, the entire production has enjoyed advanced orders. They were extensively used in controlling applications, computers and intelligent games.

In fact the original manufacturer, Zilog, could not keep up with supplies for its own orders and franshised SGS to make the shortfall. With the down-turn of sales, we have seen some of the chips come on the market and in the initial stages, nobody knew what to do with them.

A few Z80 programming books came on the market but were generally hard to follow and severely lacking in information. Most of them didn't explain how the programming went together. This is why we have designed this series.

In the first two parts we covered some experiments to show how the display was accessed and how to create simple movement and sound effects.

In this part we are continuing along the same lines with slightly more advanced material and introduce a couple of 'add-ons' for those who are advancing faster than the midstream.

Try them all, in the order presented and you will see the concept of MACHINE CODE programming fall into place.

TEC-1A

After the first run of 1,000 boards we found the 8212 display driver chips were fairly difficult to get. So we decided to update the board and include a few small improvements at the same time. Both the TEC-1 and TEC-1A operate with the same software however the TEC-1A has 3 small changes.

The regulator is mounted under the PC board so that it cannot be bent over and broken off, the 2,200uf electrolytic has been changed to 1,000uf and the output latches have been changed to 74LS273 (or 74LS 374 or 74LS377). In all other respects, the boards are identical.

TEC CASE

Either board can be mounted on a **RETEX BOX,** size:RA-2 as shown on P.3 of issue 11. From that small photo we sold hundreds of cases, proving that the magazine is read from cover to cover.

These boxes not only neaten your project but strengthen the board near the keys. There is sufficient room inside the case to fit a power transformer however we suggest using an external plug pack connected to the computer via a plug and socket on the side of the case.

If you want a TEC-CASE, they are available from us for $19.50 including post and packing.

THE ADVENTURES OF Mr WRAGG:

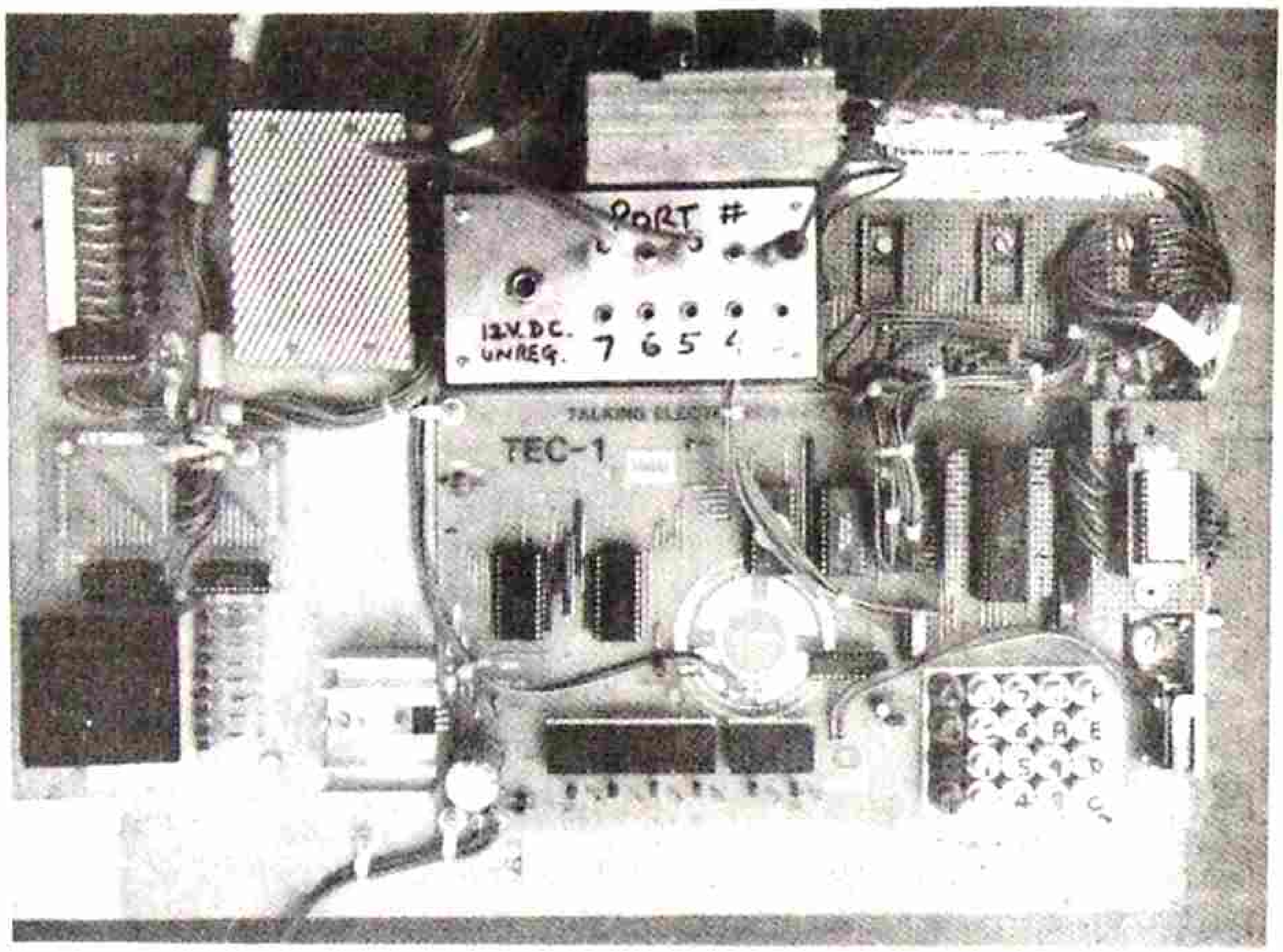
Some constructors have used their own method of mounting the PC board and others have strengthened the weaker parts of the design such as the PC board near the keys, the 7805 regulator mounting, the expansion socket etc.with a variety of ideas. But none have done more than a teacher from Brighton High School. Mr Wragg has made his computer STUDENT-PROOF. He has strengthened every lead and plug by using clips, clamps, thick hook-up wire and fasteners where ever possible to reduce the possibility of damage.

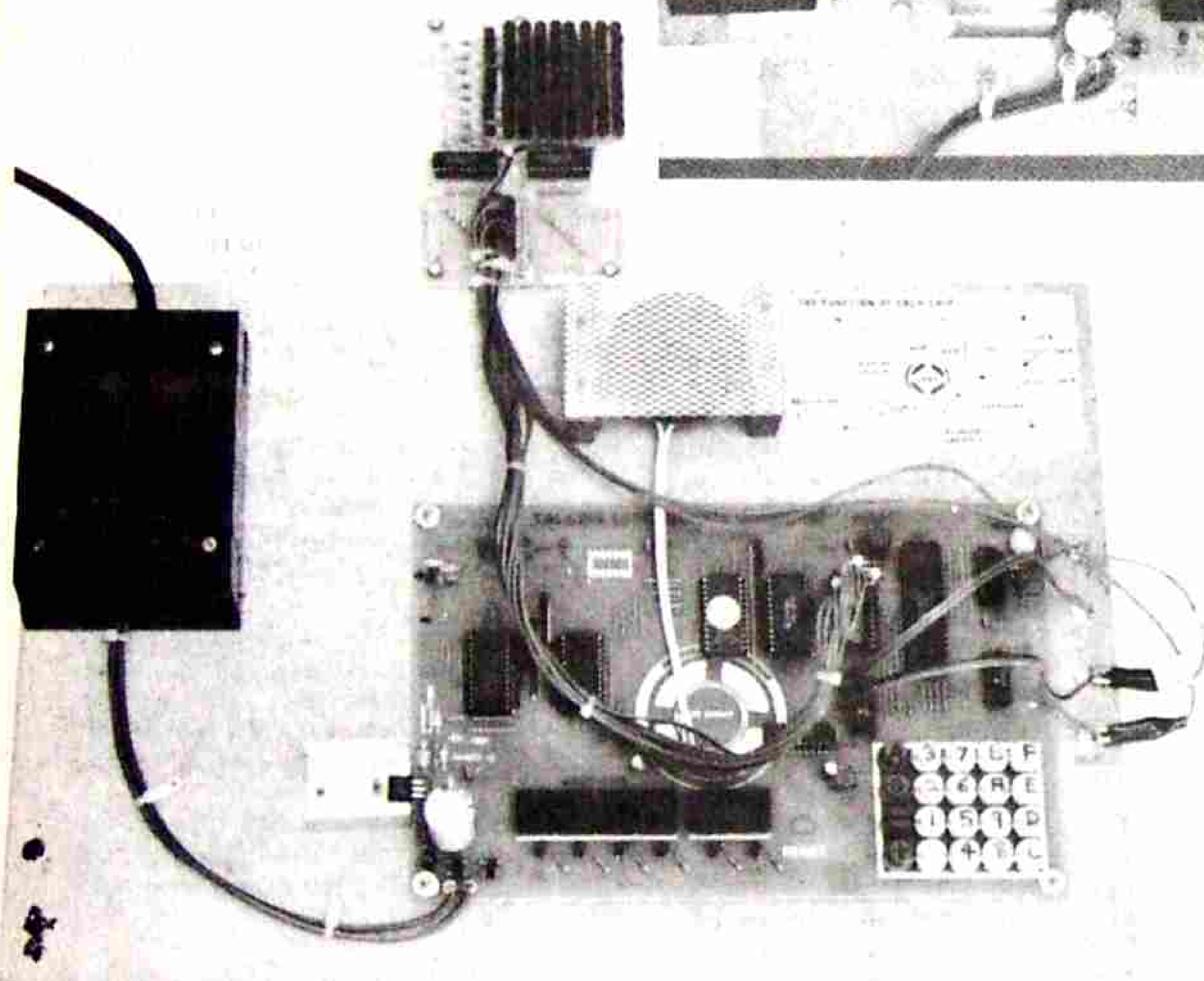
The results are shown in the photos. It looks like a before-and-after case.

The lower photo shows the TEC-1, screwed to a base-board, with the 8x8 matrix connected to the expansion socket - before he got the bug. The upper photo shows how things grew. It shows only 3 add-ons, out of the 8 he has designed and built.

Inspired by the TEC, he now holds early morning classes for electronics enthusiasts.

He covers both the practical and programming aspects of computers and some of his teaching aids will be presented in the next issue.





The beginning stages of Mr Wragg's TEC. He now has a whole batch of support devices including a Velocity-Acceleration recording interface & a programmed ROM.

We will be passing on may of these developments in future issues.

Programming on the 7-Segment Displays:

We continue from P.29 of issue 11 with more programs on the TEC display.

For this, you will need the TEC-1 or TEC-1A. No 'add-ons' are used at this stage. The displays on the PC provide the readout.

The next program, "**Back and Forth**" runs the 'g' segment across the 6 displays and back again without shifting to the 7th and 8th outputs. Thus the blank output and speaker are not activated.

The number of 'shifts' is determined by the value loaded into register C and 1, 2, 3, 4, 5, or 6 displays can be activated.

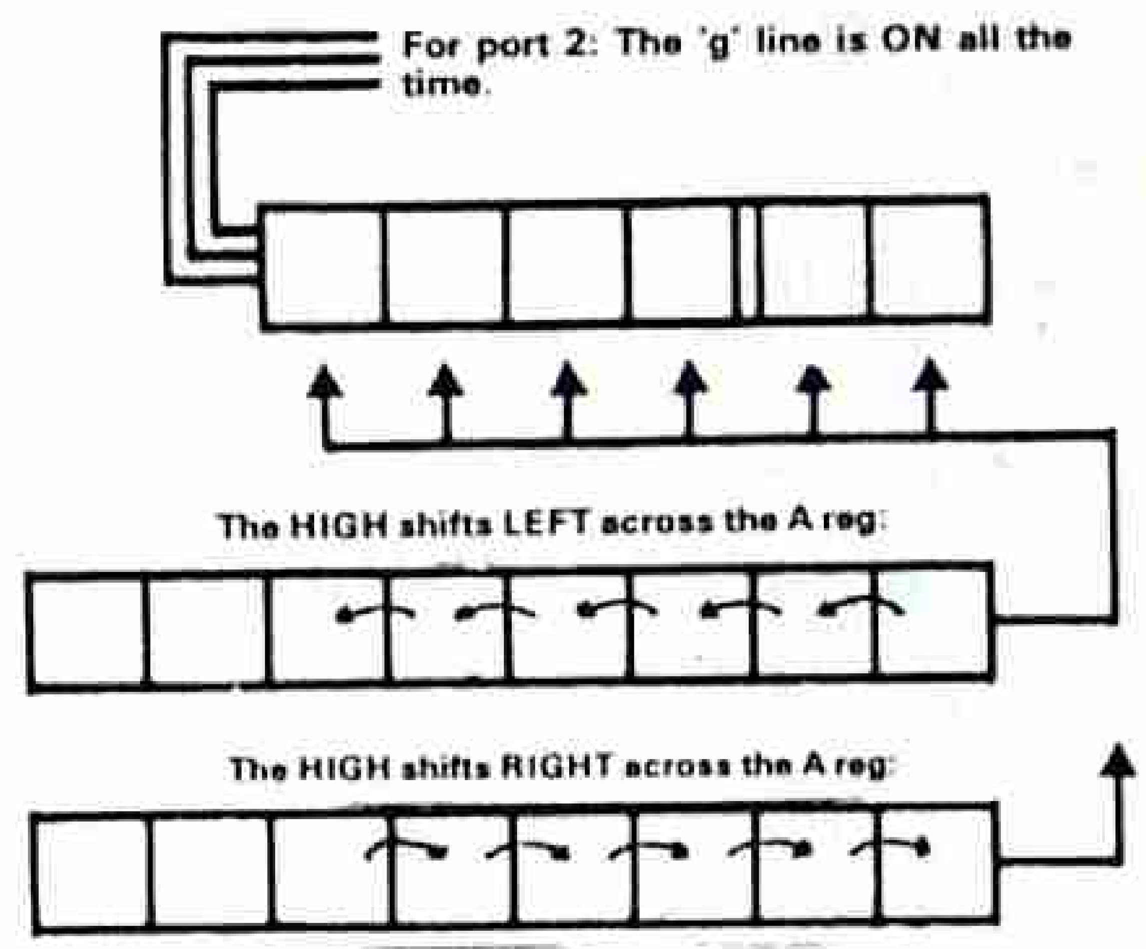
"BACK AND FORTH"

```
LD A,04 ('g' segment)       800   3E 04
OUT (2),A (seg. port)       802   D3 02
LD C,05                     804   0E 05
LD A,01 1st display         806   3E 01
OUT (1),A (cath. port)      808   D3 01
LD B,A                      80A   47
CALL DELAY                  80B   CD 00 09
LD B,A                      80E   78
RLC A                       80F   CB 07
DEC C                       811   0D
JP NZ LOOP                  812   C2 08 08
LD C,06                     815   0E 06
OUT (1),A                   817   D3 01
LD B,A                      819   47
CALL DELAY                  81A   CD 00 09
LD A,B                      81D   78
RRC A                       81E   CB 0F
DEC C                       820   0D
JP NZ LOOP 2                821   C2 17 08
JP START                    824   C3 00 08
```

Delay at 0900:

```
11 FF 07
1B
7B
B2
C2 03 09
C9
```

For Port 1, the cathode of the first display is activated, then the 2nd display, then the third display, etc.



The diagram shows the shifting of the HIGH across the accumulator (the A register). This is output to port 1 and will turn on one transistor at a time. These transistors feed the common cathode lines of the displays.

Keep this program in the computer for the next experiment.

QUESTIONS:

1. At 805, why do we insert 05 into register C? Try the value 06 and see what happens. (slow the clock rate to see the effect.)

2. At 805, insert the value 03. What happens?

3. At 805, insert the value 01. Can you see what is happening on the screen and in register C. (The bit is being shifted through the register via the speaker and is appearing on the left hand side of the display.)

Another interesting TEC-1 repair came in this week. The constructor had purchased a set of components from us in 'short-form'. He had bought some of the items himself and the remainder he purchased from us - things like the EPROM, Z80 and display drivers.

But what was interesting, he had made the printed circuit board himself. Not being satisfied with photocopying the layout on the back of the magazine, he reproduced the entire artwork using tape and stick-on lands. It was copied so exactly that it took us a few minutes to realize the situation. And when it finally dawned, it was quite a shock! It was like looking at a forgery! The work entailed in its creation must have been enormous. And now we had to repair it.

Normally there is not a single firm under the sun which would entertain a repair which had not been made with components and PC board as supplied by the organisation. You can imagine the assortment of repairs which would be sent in.

But because we have had so much success with repairing the computer, we accepted it and went to work.

Theoretically, every track had to be inspected because the board was a 'one-off'.

After quite a few minutes, we noticed two tracks did not connect to the appropriate places. When we joined these with short lengths of tinned copper wire, all the chips received power, but still the computer failed to work.

Then we noticed the clock chip was a Fairchild 4049 and this was promptly replaced.

The display immediately lit up and on going through the keyboard, we found keys 4 and 7 did not give the correct display reading.

Back to more inspection. More detective-work located a track on the common line which should not have been there. When this was cut, the computer worked perfectly.

Here's the lesson:

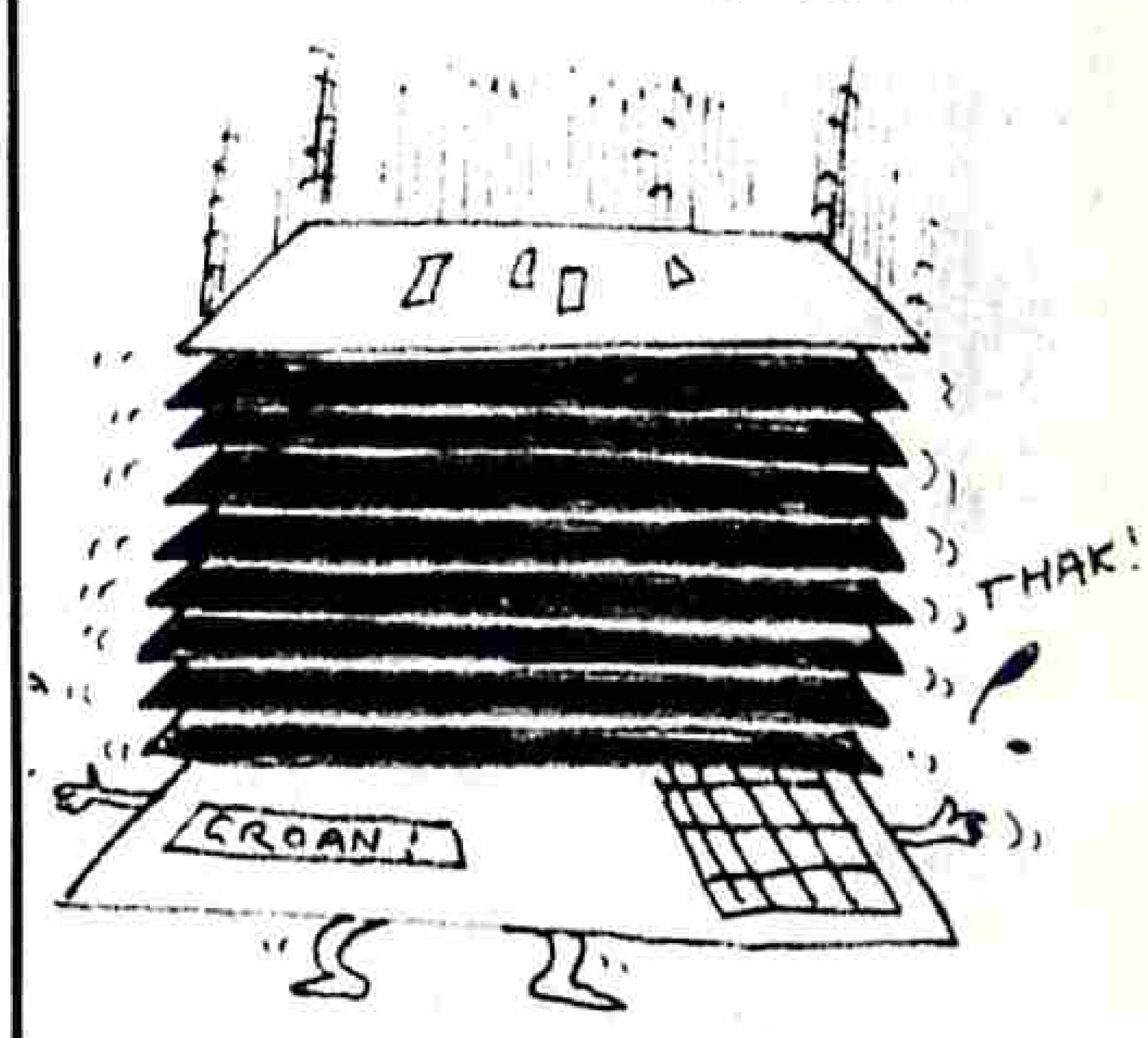
If you are going to reproduce artwork, you must check, re-check and then triple check the layout. The only effective way to do this is to make the artwork identical in size and design so that the two layouts can be placed together and held up to the light. Looking from both sides, you will be able to detect any discrepancy in the wiring.

When the board is etched and drilled, go over the entire board with a multimeter and check each of the lines for continuity. Mark each line as you check it so that none are missed.

Only this way can you be sure every line goes to its correct destination.

I know it is fun to produce everything yourself, but for some items the result is not economical. Making PC boards is one of the most time-consuming and costly endevours. After its all finished, it will probably cost nearly as much as a bought one and won't have an overlay or solder mask.

If you have made your own PC board and it doesn't work, please don't send it in for repair. We have met the challenge and don't need it again. It will involve too much of our time and delays the production of the next issue of the magazine.



AND OUTPUT LATCHES & DRIVE

"ALL THE VALUES"

This program produces all the combinations for the 7-segment display. This will include many unusual effects as well as all the known letters and numbers.

It is basically an extension to the **BACK and FORTH** program with an additional listing at **0A00** and a change at **824** and **801.**

Fill in this section:

```
3E 00
D3 02
0E 05
3E 01
D3 01
47
CD 00 09
78
CB 07
0D
C2 08 08
0E 06
D3 01
47
CD 00 09
78
CB 0F
0D
C2 17 08
C3 02 0A
```

At 0A00:

```
LD L,01    A00   2E 01
INC L      A02   2C
LD A,L     A03   7D
JP 0802    A04   C3 02 08
```

Delay at 0900:

```
11 FF 07
1B
7B
B2
C2 03 09
C9
```

Push RESET, GO:

Write the ASSEMBLY CODE for the program above and also the address listings, starting at 0800.

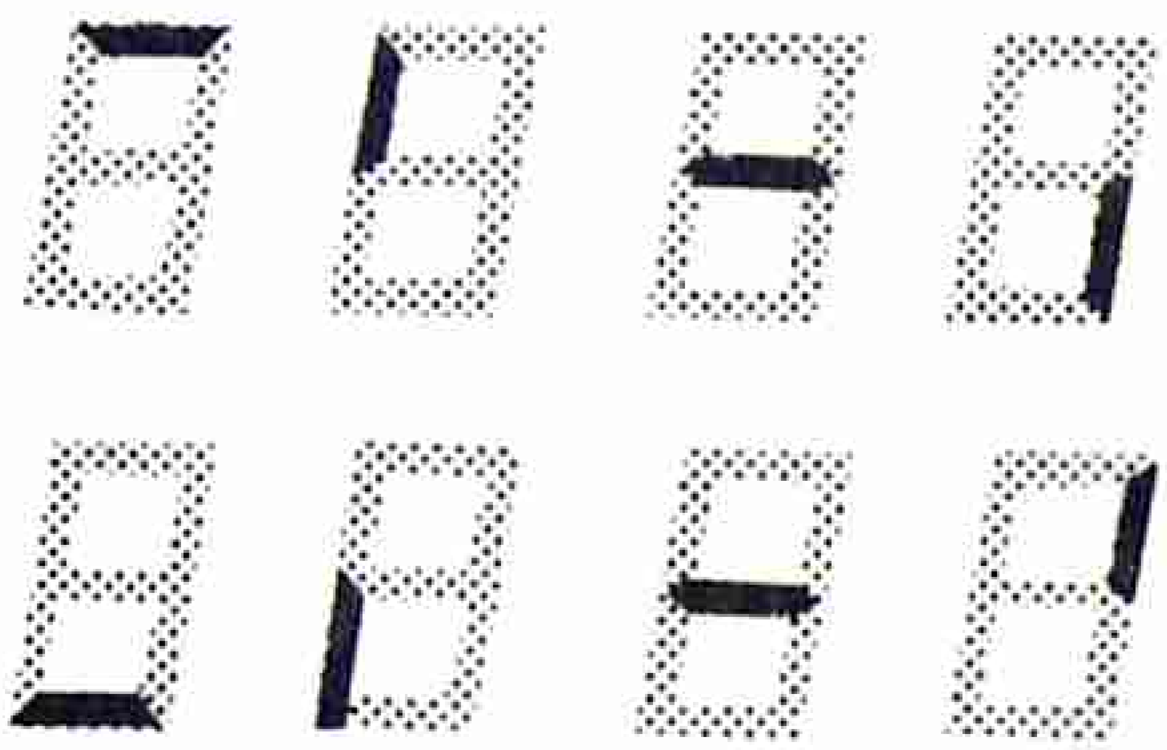
MOVEMENT AROUND A SINGLE 7-SEGMENT DISPLAY:

The following program produces movement around a single 7-segment display which can be increased in speed to produce a novel effect.

```
LD A,20      800   3E 20
OUT (1),A    802   D3 01
LD A,01      804   3E 01
CALL 0900    806   CD 00 09
LD A,02      809   3E 02
CALL 0900    80B   CD 00 09
LD A,04      80E   3E 04
CALL 0900    810   CD 00 09
LD A,20      813   3E 20
CALL 0900    815   CD 00 09
LD A,80      818   3E 80
CALL 0900    81A   CD 00 09
LD A,40      81D   3E 40
CALL 0900    81F   CD 00 09
LD A,04      822   3E 04
CALL 0900    824   CD 00 09
LD A,08      827   3E 08
CALL 0900    829   CD 00 09
JP 0800      82C   C3 00 08
```

at 0900:

```
OUT (2),A    900   D3 02
             902   11 FF 07
             905   1B
             906   7B
             907   B2
             908   C2 05 09
             90B   C9
```



This program has not been efficiently written. It contains a repetition of **LD A'** and **'CALL'** instructions. It should have a **BYTE TABLE.** This will be shown in a later program.

However it does contain one byte-saving feature. The statement **OUT (2),A** is used at each stage and has been placed in the **CALL ROUTINE.** This saves 14 bytes of program.

1. Replace **(07)** at **904** with **00** and watch the screen.

2. Replace **(FF)** at **903** with **0F**.

3. At address **801** replace the data with **01, 04,** and then **10.** What happens to the display?

4. Replace data at **801** with **05, 0F, 2D.** What happens to the display in each of these cases?

THE DELAY ROUTINE

We have used a delay routine in many of the programs we have investigated.

We have also seen how it can be adjusted from a few milliseconds to a few seconds in length.

For this time-delay to occur, many thousands of clock cycles must be involved in its execution. In fact, up to one million or more cycles can be involved.

Let us look at how this comes about and how the delay operates.

The delay program we will be investigating is:

1. **11 FF 07**
2. **1B**
3. **7B**
4. **B2**
5. **C2**

The meaning of each line is:

1. **LD DE 07FF** FF is loaded into E and 07 into D
2. **DEC DE** Register E is decremented by one. If an **underflow** occurs, register D is decremented.
3. **LD A,E** Register E is loaded into the accumulator.
4. **OR D** The D register is OR-ed with the accumulator.
5. **JP NZ** to: The program jumps to line 2 if the result is not zero.

The number of cycles to perform each operation is as follows:

LD DE	**11 FF 07**	10 cycles
DEC DE	**1B**	6 cycles
LD A,E	**7B**	7 cycles
OR D	**B2**	7 cycles
JP NZ Line 2	**C2 03**	10 cycles

One loop consists of:

1B	- 6
7B	- 7
B2	- 7
C2	- 10

total: 30 cycles.

To produce a delay of 256 loops, the instruction is:

11 FF 00
FF is loaded into the E register and 00 into the D register.

E is decremented by one on each loop of the program and when it gets to 00, the result of OR-ing the accumulator (which will contain the value of the D register - 00) will be zero and the microprocessor will jump out of the delay routine and back to the main program.

Total clock time for the 256 loops is:
256 x 30
= 7680 cycles.

If the D register is loaded with FF the delay time will be:

7680 x 256
= 1,966,000 cycles.
This is about 2 million cycles!

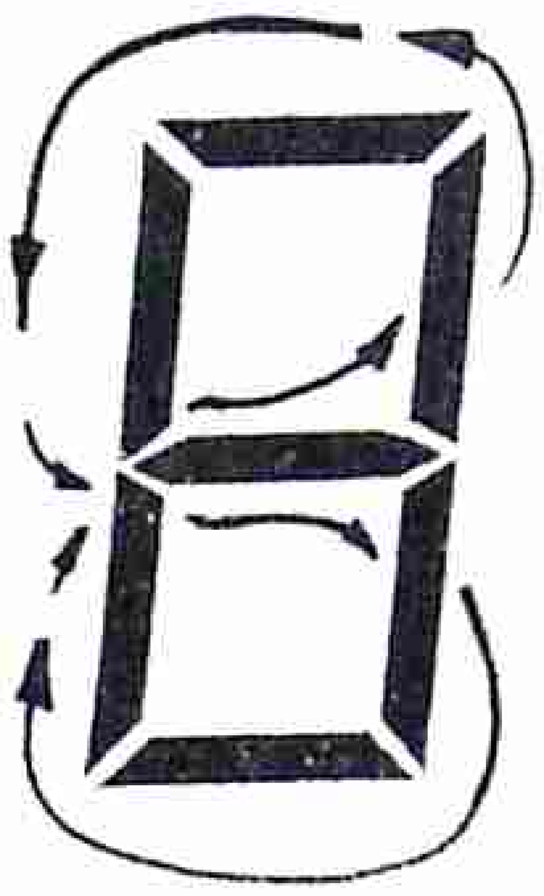
When deciding upon a delay of suitable length, try various values in this location:

11 FF 07

This location has only a small effect on the delay time and can be considered to have a 'trimming effect'.

FIGURE 8's ACROSS THE SCREEN:

The next program introduces a table of values which the program 'looks up' during the execution of each cycle. These are called **'data bytes'** or bytes of data, which are used one at a time.



The LED turn-on sequence around the display.

```
LD C,20         800    0E 20
LD A,C          802    79
OUT (1),A       803    D3 01
CALL 0900       805    CD 00 09
RRC C           808    CB 09
BIT 7, C        80A    CB 79
JP Z LOOP       80C    CA 02 08
JP 800          80F    C3 00 08
```

```
LD HL,0B00      900    21 00 0B
LD B,9          903    06 09
LD A(HL)        905    7E
OUT (2)A        906    D3 02
CALL DELAY      908    CD 00 0A
INC HL          90B    23
DEC B           90C    05
JP NZ 905       90D    C2 05 09
RET             90F    C9
```

At 0A00

```
A00    11 FF 07
A03    1B
A04    7B
A05    B2
A06    C2 03 0A
A09    C9
```

at 0B00:

```
B00    01
B01    02
B02    04
B03    20
B04    80
B05    40
B06    04
B07    08
B08    01
```

On the first pass, the program places 0B00 into a register pair such that the 0B goes into the H register (meaning the High order byte register) and 00 in to the L register (Low order byte register).

At 905 the contents of the byte at 0B00 will be loaded into the Accumulator because this is the address specified by the HL instruction.

On each subsequent pass, the HL register pair in incremented by ONE. Since the value 0B00 is contained in this pair, the result will be to add 1 to 00 to get 01, 02 etc. Thus the program will look up 0B01, 0B02 etc and finds the relevant byte of data.

The second interesting part of this program is the counting of the **DATA BYTE** table. The computer must know how many data bytes are to be accessed.

Thus it is given an intial value of **09** at **904** and decremented the value by one on each pass. When the result is zero, the program jumps to **0800** and starts again.

The segment can be made to travel across the screen in the opposite direction by changing 3 values:

The starting value at 801 must be changed to 01, RRC must be changed to RLC **(CB 01)** and bit 6 must be tested for zero **(CB 71)**

The changes are:

```
800   0E 01
808   CB 01
80A   CB 71
```

CONTROL VIA THE KEYBOARD

Movement on the screen can be controlled by the keyboard by introducing a **HALT** or wait function. This causes the program to halt and wait for an input via the interrupt line.

When a key is pressed, the non-maskable interrupt line is activated and allows the Z80 to accept data from the keyboard encoder via the data bus.

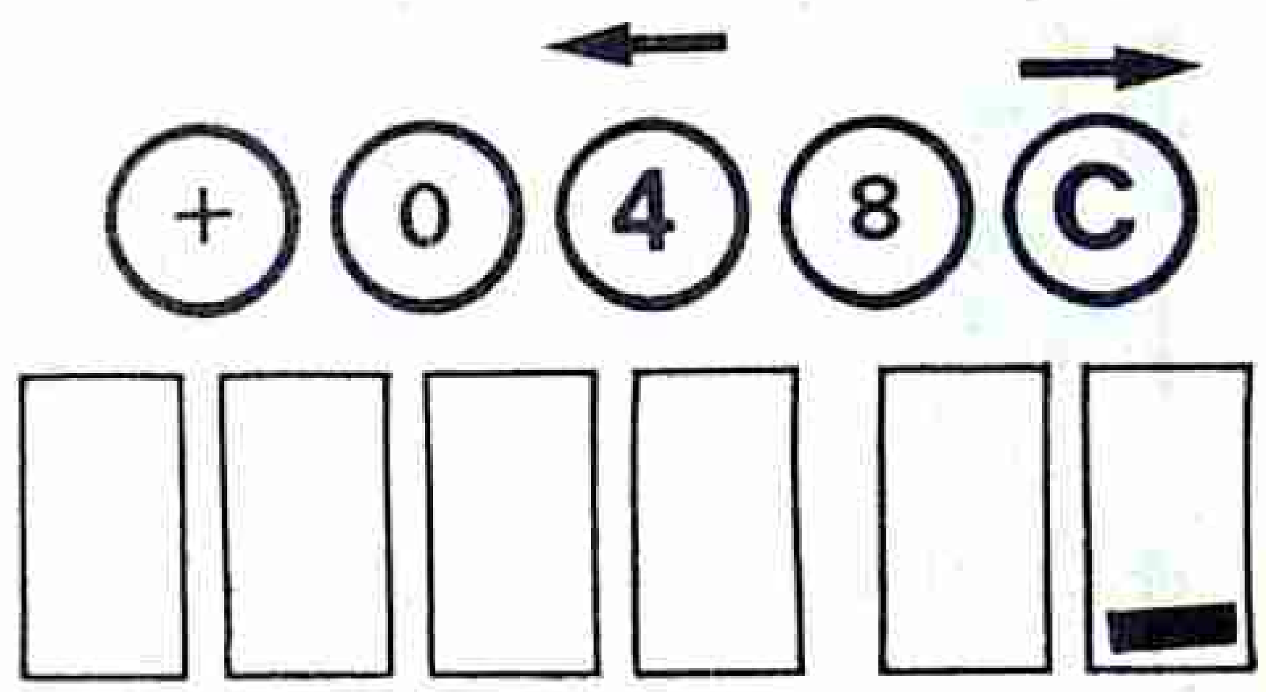
The data is loaded into the accumulator and compared with a value in the program. If the two values are the same, the output is zero and as determined by the next instruction, the program advances.

This program moves a LED across the bottom row.

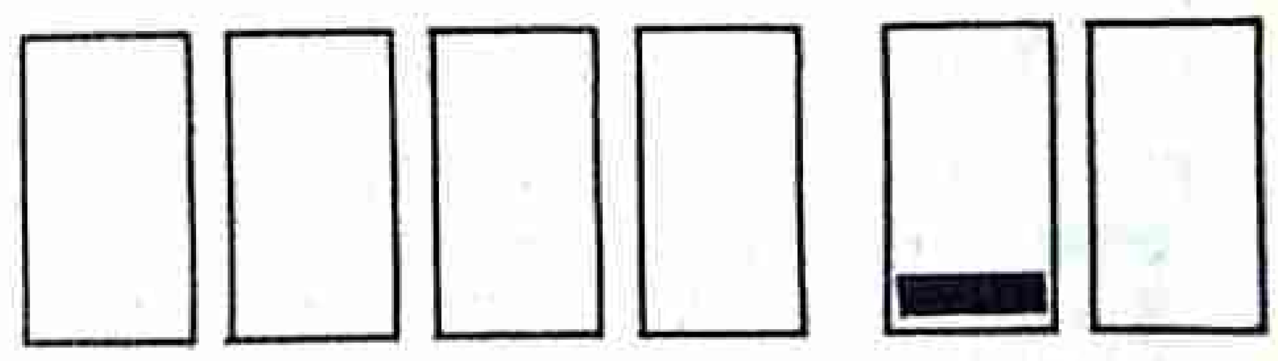
Key '4' shifts the LED left and 'C' shifts it right.

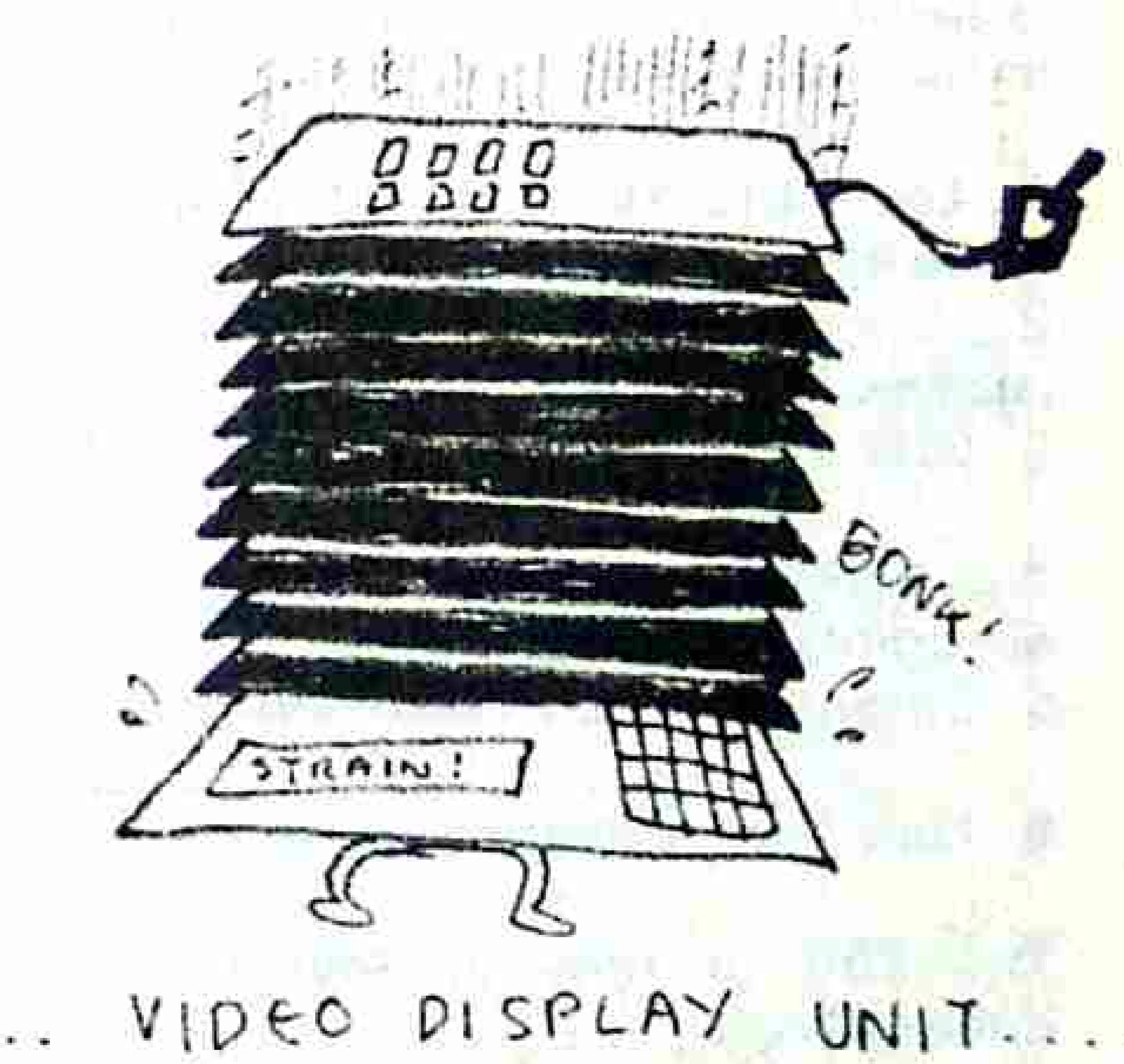
The direction of shift is determined by **RLC B** and **RRC B.** Each press of a button moves the LED one place. No delay routine is required in this program.

```
LD A,04         800    3E 04
OUT (2),A       802    D3 02
LD B,A          804    47
LD A,B          805    78
OUT (1),A       806    D3 01
HALT            808    76
LD A,01         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
```



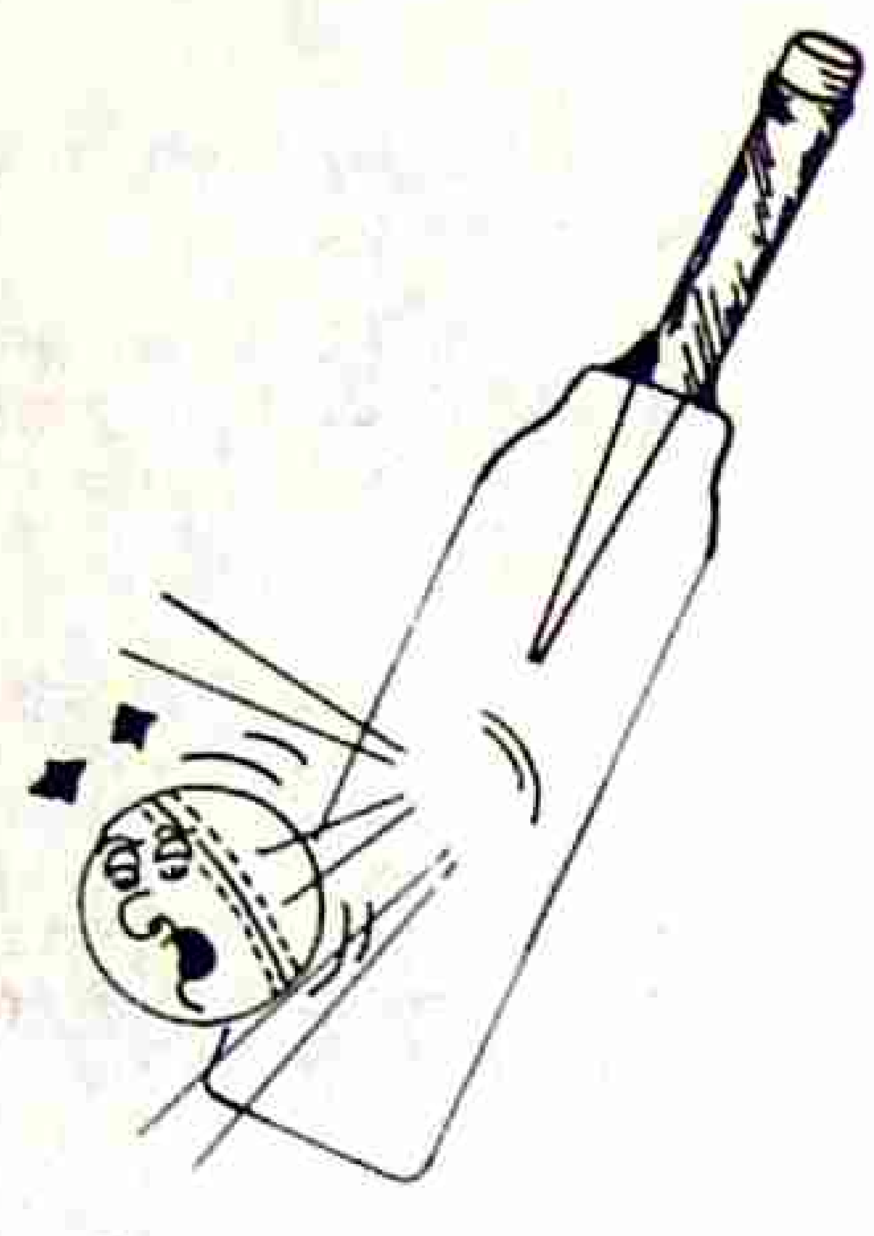
The 'd' segment shifts across the display. The direction is determined by buttons '4' and 'C'.





CREATING A BAT

This program produces a **2-segment** bat capable of travelling across the lower segment of the display. The '+' key moves the bat to the left and the 'C' key moves it to the right.



```
LD A,80        800   3E 80
OUT (2),A      802   D3 02
LD B,03        804   06 03
LD A,B         806   78
OUT (1),A      807   D3 01
HALT           809   76
LD A,I         80A   ED 57
CP 10          80C   FE 10
JP NZ 0816     80E   C2 16 08
RLC B          811   CB 00
JP 806         813   C3 06 08
CP C           816   FE 0C
JP NZ 809      818   C2 09 08
RRC B          81B   CB 08
JP 806         81D   C3 06 08
```

WRITING A PROGRAM

This is a written exercise requiring YOU to write a program. Our aim will be to write a BAT program exactly like the previous program and you can refer to it if a problem arises.

For each line, the MACHINE-CODE value should be obtained from the Z80 CODE SHEET on the back page of issue 11. It should then be placed in the space provided.

1. Load the accumulator with the value 80. Answer: ______
2. Output the contents of the accumulator to port 2. ______
3. Load register B with the value 3: ______
4. Load register B into the accumulator: ______
5. Output the accumulator to port 1: ______
6. Halt the program: ______

The Z80 is now waiting for an interrupt.

7. Load the index register into the accumulator: ______
8. Compare the accumulator with the value 10: ______
9. Jump to 'COMPARE C' (below) if the answer to line 8 is NOT zero: ______
10. Rotate register B LEFT CIRCULAR: ______
11. Jump to the address which states: Load B into A (above): ______
12. Compare the accumulator with the value 'C': ______
13. Jump to HALT (above) if NOT zero: ______
14. Rotate register C Right Circular: ______
15. Jump to: Load register B into A (above): ______

Complete the following listing by adding the values you have obtained from the statements above:

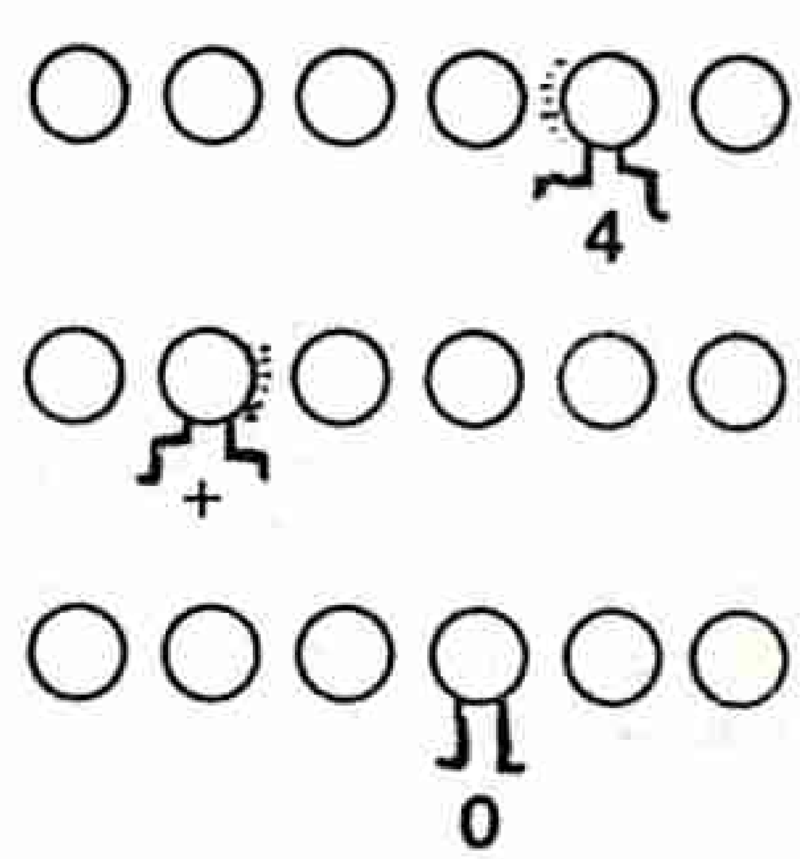
```
800
802
804
806
808
80A
80B
80D
80F
812
814
815
817
81A
81D
81F
822
824
825
827
82A
```

AUTO MOVEMENT & HALT

S Riley,
Guildford, 2161.

The following program detects 3 keys. The '+' key shifts the LED left, the '0' key stops the LED and key '4' shifts it right.

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




```
LD A,01         800   3E 01
OUT (2),A       802   D3 02
LD A,01         804   3E 01
OUT (1),A       806   D3 01
LD B,01         808   06 01
HALT            80A   76
LD A,I          80B   ED 57
CP '+'          80D   FE 10
JP NZ 081D      80F   C2 1D 08
RLC B           812   CB 00
LD A,B          814   78
OUT (1),A       815   D3 01
CALL DELAY      817   CD 00 0C
JP 80B          81A   C3 0B 08
CP 04           81D   FE 04
JP NZ 80D       81F   C2 0D 08
RRC B           822   CB 08
LD A,B          824   78
OUT (1),A       825   D3 01
CALL DELAY      827   CD 00 0C
JP 808          82A   C3 0B 08

at 0C00:              11 FF 0A
                      1B
                      7B
                      B2
                      C2 03 0C
                      C9
```

So far we have turned on one segment or LED at a time in the display or more than one segment or LED within the same digit. But not 2 LEDs in different displays, in different positions.

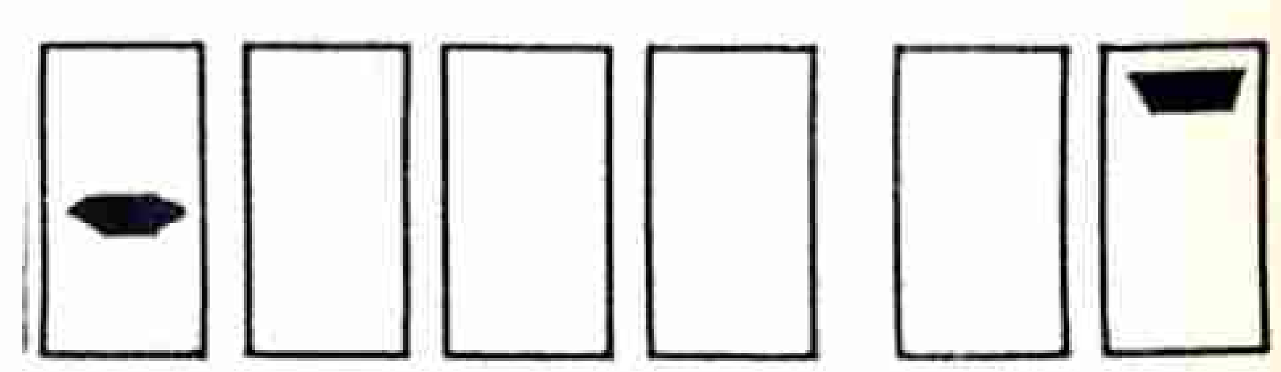
This seems impossible but by using a clever pulsing technique we can alternately access one then the other to produce the effect of both being on at the same time.

In this program we will alternately access segment 'g' in the first display and segment 'a' in the 6th display to give the appearance that they are both on at the same time.

SWITCHING 2 PIXELS INDEPENDENTLY:

Run the following program and observe the effect:

```
LD A,20         800   3E 20
OUT (1),A       802   D3 01
LD A,01         804   3E 04
OUT (2),A       806   D3 02
CALL DELAY      808   CD 00 0B
JP 0A00         80B   C3 00 0A
```



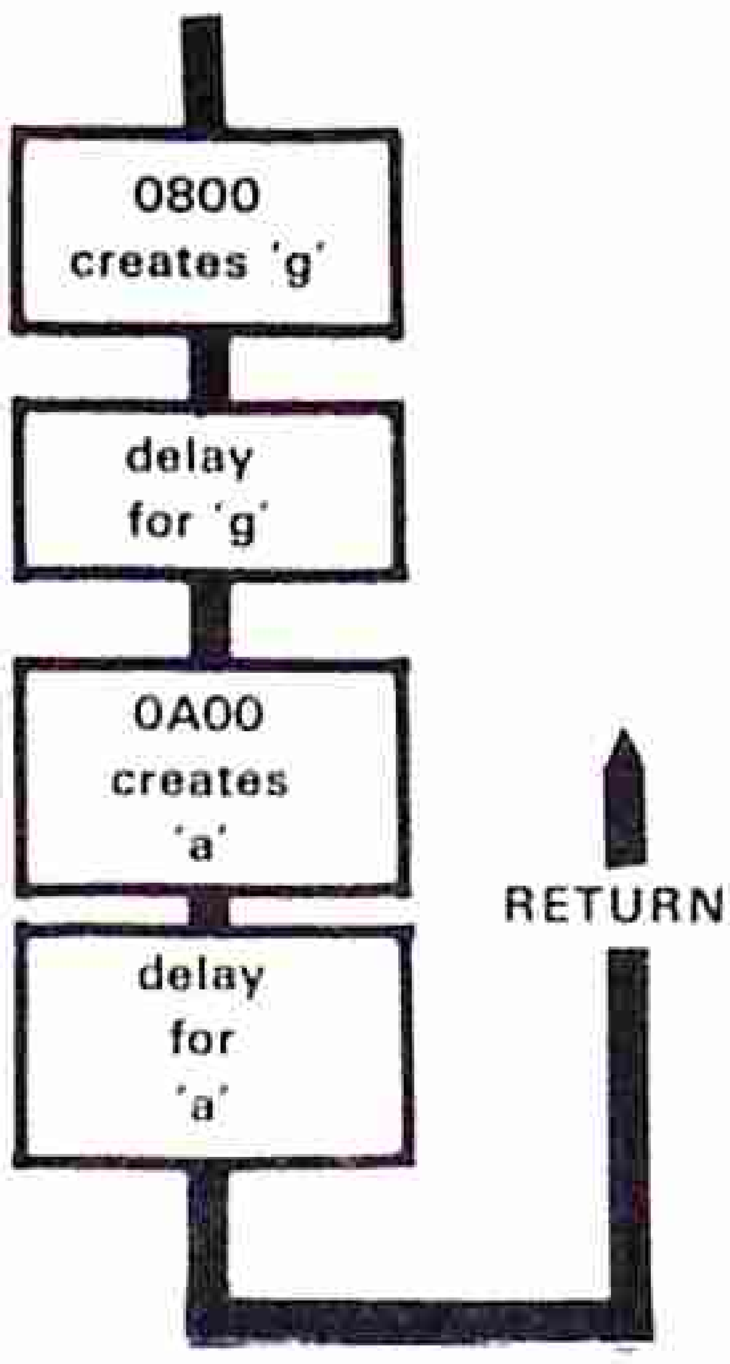
```
LD A,01         A00   3E 01
OUT (1),A       A02   D3 01
LD A,01         A04   3E 01
OUT (2),A       A06   D3 02
CALL DELAY      A08   CD 00 0B
JP 0800         A0B   C3 00 08
```

at 0B00:

```
11 FF 00
1B
7B
B2
C2 03 0B
C9
```

Turn the speed control up and the effect is two different LEDs being lit at the same time. Turn the speed control down and the alternating effect becomes more noticeable.

The flow diagram for this is:



Insert **05** into the delay routine at **0B02** and watch the display. The alternating effect is more obvious. This is the basis for all the letters and writing on the display. Each digit is being turned on and off very quickly.

PROBLEMS:

Turn the speed control up and keep the delay routine short for the following problems:

1. Change values in the program to turn on segment 'd' in the first display and 'a' in the sixth display.

2. Create the figure 1 in the first display and '0' in the last display. Which locations in the program must be altered to achieve this?

TO CONTROL 2 PIXELS. One with movement.

This program produces two pixels. One is fixed and the other moves up and down.

In this experiment, the main program is at **0A00** and it calls the delay at **0B00** and a short routine at **0800**. When the program has been entered, push, RESET, ADdress, 0A00, GO, GO to execute the program. The main task with this experiment will be to rewrite the main program so that it appears at 0800. This will involve changing a number of machine code values to suit the new location.

```
LD C,BB  (or any value)  0A00  0E BB
LD A,01                  0A02  3E 01
OUT (1),A                0A04  D3 01
LD A,01                  0A06  3E 01
OUT (2),A                0A08  D3 02
CALL DELAY               0A0A  CD 00 0B
CALL 0800                0A0D  CD 00 08
DEC C                    0A10  0D
JP NZ 0A02               0A11  C2 02 0A
LD C,BB                  0A14  0E BB
LD A,01                  0A16  3E 01
OUT (1),A                0A18  D3 01
LD A,04                  0A1A  3E 04
OUT (2),A                0A1C  D3 02
CALL DELAY               0A1E  CD 00 0B
CALL 0800                0A21  CD 00 08
DEC C                    0A24  0D
JP NZ 0A16               0A25  C2 16 0A
LD C,BB                  0A28  0E BB
LD A,01                  0A2A  3E 01
OUT (1),A                0A2C  D3 01
LD A,80                  0A2E  3E 80
OUT (2),A                0A30  D3 02
CALL DELAY               0A32  CD 00 0B
CALL 0800                0A35  CD 00 08
DEC C                    0A38  0D
JP NZ 0A2A               0A39  C2 2A 0A
JP 0A00                  0A3C  C3 00 0A
```

Delay at 0B00:

```
11 FF 00
1B
7B
B2
C2 03 0B
C9
```

```
LD A,20      800  3E 20
OUT (1),A    802  D3 01
LD A,01      804  3E 80
OUT (2),A    806  D3 02
CALL DELAY   808  CD 00 0B
RETURN       80B  C9
```

Problem:

1. Rewrite the MAIN PROGRAM to start at 0800:

```
LD C,BB      800
LD A,01      802
OUT (1),A    804
LD A,01      806
OUT (2),A    808
CALL DELAY   80A
CALL 0A00    80D
DEC C        810
JP NZ 0802   811
LD C,BB      814
LD A,01      816
OUT (1),A    818
LD A,04      81A
OUT (2),A    81C
CALL DELAY   81E
CALL 0A00    821
DEC C        824
JP NZ 0816   825
LD C,BB      828
LD A,01      82A
OUT (1),A    82C
LD A,80      82E
OUT (2),A    830
CALL DELAY   832
CALL 0A00    835
DEC C        838
JP NZ 082A   839
JP 0800      83C
```

Fill in the MACHINE CODE values:

at 0B00:
Same Machine code values for delay.

```
0B00
0B03
0B04
0B05
0B06
0B09
```

at 0A00:

```
LD A,20      A00
OUT (1),A    A02
LD A,01      A04
OUT (2),A    A06
CALL DELAY   A08
RETURN       A0B
```

Run the program by pressing RESET, GO. Does it work? (It should)

2. Insert the following data into the program you have written, to produce the name of a semiconductor:

at 0806: 3E 47

at 081A: 3E C7

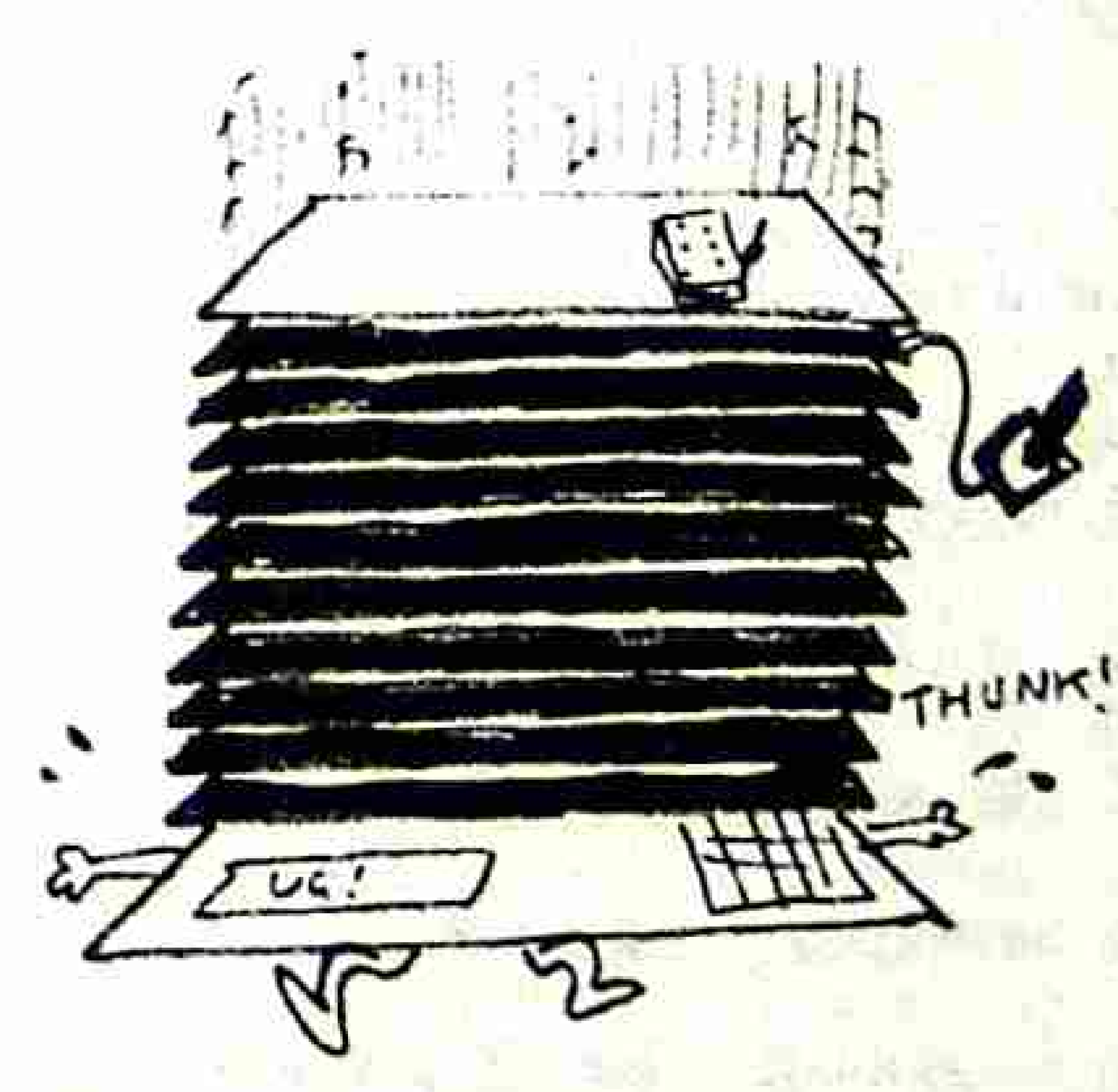
at 082E: 3E C6

What is the name of the device?

3. Create the name of another semiconductor device by inserting the following information into the program:

```
802   3E 10
806   3E 4F
816   3E 08
81A   3E EA
82A   3E 04
82E   3E C6
```

Remove the value **20** at **0A01**.
Push RESET, GO.
What is the name of the device?
Reduce the delay of **BB** in the MAIN PROGRAM (at 3 locations) to **05**.
The result will be your first readable multiplexed word.



AND AN EPROM BURNER.

JUMPS AND CALL INSTRUCTIONS

JUMP and CALL instructions are called **BRANCH INSTRUCTIONS.**

They cause the program to branch to another location in memory and execute the instruction contained at that location.

The 6 instructions we will investigate are:

JP Address	**C3 XX XX**
JP NZ Address	**C2 XX XX**
JR dis	**18 XX**
JR NZ dis	**20 XX**
CALL Address	**CD XX XX**
CALL NZ Address	**C4 XX XX**

The meaning of each instruction is as follows:

JP Address. This is an unconditional instruction . It means **Jump: Address.** The program will jump to a new address as determined by the next two bytes XX XX.

JP NZ Address. This means **Jump, non-zero: address.** The program will only jump to a new address if the result of the previous instruction is **NOT** zero. (If the result is zero, the program will neglect this 3-byte instruction and advance to the next instruction).

JR dis. This is an unconditional statement. It means: **Jump relative: displacement.**

In simple terms a relative jump means the program will jump to an address of plus 129 bytes or minus 126 bytes of the address of the JR op-code byte.

For instance, the value **FB** will cause a jump to **1B** in the following program.

11 FF 07
1B
7B
B2
18 FB
C9

For a forward jump, **03**will cause the program to jump to **D3** in the following:

3E 01
18 03
3E 20
D3 01
3E 28

JR NZ dis. This is a conditional statement. It means **Jump relative, non-zero: displacement.** The displacement is given by a hex value such as **D7, EE, F8** for a backward jump or **07, 18, 44, 76,** for a forward jump.

When determining the displacement value, this is an easy method:

★★ ★★	**F9 FA**
★★	**FB**
★★ ★★	**FC FD**
20 XX	**FE FF**
★★ ★★ ★★	**00 01 02**
★★ ★★	**03 04**
★★	**05**
	etc.

can be **18** or **20**

CALL Address. This is an unconditional instruction. It means **CALL the address given by the next two bytes XX XX.**

When using this instruction, it must be the intention of the programmer to call a sub-routine and then return to the instruction which immediately follows, as this is the requirement of the microprocessor.

For this reason, the sub-routine must conclude with a return instruction **C9.** The address of the byte immediately following **CD XX XX** will be saved in the stack. At the conclusion of the sub-routine it will be **popped** off the stack, looked at, and cause the program to return to the instruction after **CD XX XX.**

CALL NZ Address. This is a conditional instruction and will only be executed if the result of the previous instruction is **NOT** zero.
All other features of this instruction as per CALL Address above.

The main differences between these three sets of jump instructions are:

A **JP** instruction causes the program to go to a sub-routine but does not call it back again.

A **JP** instruction can make the program go to any location in memory. It is not restricted to a displacement value.

A **JP** instruction cannot be re-located without changing or looking at the two-byte jump address to see if the sub-routine is still at the same address.

A **JR** can only operate within +127 and -128 bytes (approx.)

JR can be easily re-located as it relates only to relative memory. This type of instruction is ideal when large portions of a program need to be shifted.

CALL instructions are used when a sub-routine is required to be executed (such as a delay) followed by a return to the main program.

QUESTIONS

1. Write the meaning of these, in words:
   (a) **JP**
   (b) **JP NZ**
   (c) **JR dis**
   (d) **JR NZ dis**
   (e) **CALL**
   (f) **CALL NZ**

2. Which instruction would you use for the following:
(a) You require to go to a sub-routine and then return to the main program.
(b) You require to go to another routine if the answer to the previous line is NOT zero.
(c) You require to go to the beginning of the program.
(d) You require to go to a location about 15 bytes further down the program.
(e) You require to go to a sub-routine on the condition NON-zero, and return.
(f) You require to go to a location back 8 bytes.

3. Give one advantage of a **JUMP RELATIVE** instruction, compared to a **JUMP** instruction.

4. To produce a loop in a program, which of the following should be used: **JR dis or JR NZ dis.**

5. At the end of a program, which instruction should be used: **CALL, JR NZ, JP.**

6. What is the difference between **CALL** and **JUMP**?

Answers:

1. Jump.
Jump Non-zero.
Jump Relative displacement.
Jump relative non-zero displacement
Call.
Call Non-zero.

2. (a) **CALL**
(b) **JP NZ**
(c) **JP**
(d) JR 14
(e) **CALL NZ**
(f) **JR F6**

3. The program can be transferred to another location without affecting the **JUMP RELATIVE** instruction.

4. It must have a non-zero condition.

5. It must be a **JUMP** instruction with no other conditions.

6. **CALL** transfers the program to another location and requires that it be returned to the next instruction after the sub-routine has been performed. **JP** transfers the program to another location without any return requirement.

USING JR's

To show how we can substitute a **JR** instruction for a **JUMP** instruction, we will consider a simple program containing a delay routine.

We will choose the program: **RUNNING SEGMENT 'a' ACROSS THE SCREEN.** This can be found on P. 26 of issue 11 and is repeated here:

Type this into the computer and **RUN.**

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

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

1. We will change the instruction at **80E** to **JR 806** . Change the address values to **18 F6.**

Place this into the program and **RUN.** Is any difference observed? (There should be no difference).

2. The delay routine at **0A00** can also be changed to include a **JR** instruction.

at **A06:**
change **C2 03 0A** to **20 FB 00**

Run the program and note the result. No difference should be detected. Both instruction perform the same in this case.

3. The **DELAY PROGRAM** can be placed immediately below the main program so that a **JR** instruction can be used at **809** and also at the end of the delay.

at **809:** insert **JR 820.**

Start the delay routine at **0820.**

At the end of the delay routine, insert: **JR 80C.**

The displacement values will have to be worked out by you. Follow through the steps as shown and write the complete program. Use the TEC as a counter to work out the displacement values (by pushing +,+,+,+,+, etc.)

Do not look at the answer at the bottom of the next column until you have finished.

Next issue we will give a **JR table** and explain how it is used.

QUICK DRAW

In this final exercise we will change a number of **JUMP** instructions to **JR** instructions. See the **QUICK DRAW** program on P. 13 of issue 11.

This is how to change the program:

1. Copy all the assembly code, replacing **JP** with **JR.**

2. Copy the machine code listing, remembering that the 3-byte **JP** instructions will become 2-byte **JR** instructions. At this stage do not insert the displacement values - this will be the final job.

3. Insert the displacement values for each of the **JR** instructions.

This is what your program should look like:

QUICK DRAW

```
         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
         JR NZ Delay      80A   20 FB
         LD A,E3          80C   3E E3
         OUT (2),A        80E   D3 02
         LD A,08          810   3E 08
         OUT (1),A        812   D3 01
LOOP 1   HALT             814   76
         AND OF           815   E6 0F
         CP 0C            817   Fe 0C
         JR Z,Right       819   28 05
         OR A             81B   B7
         JR Z,Left        81C   28 06
         JR Loop 1        81E   18 F4
RIGHT    LD A,01          820   3E 01
         JR Finish        822   18 02
         LD A,20          824   3E 20
         OUT (1),A        826   D3 01
LEFT     LD A,20          828   3E 28
FINISH   OUT (2),A        82A   D3 02
         HALT             82C   76
         JR Start         82D   18 D1
```

4. Fill in the memory locations, starting at **0800**.

Push RESET, GO and play the Quick Draw game. Does everything work correctly? It should.

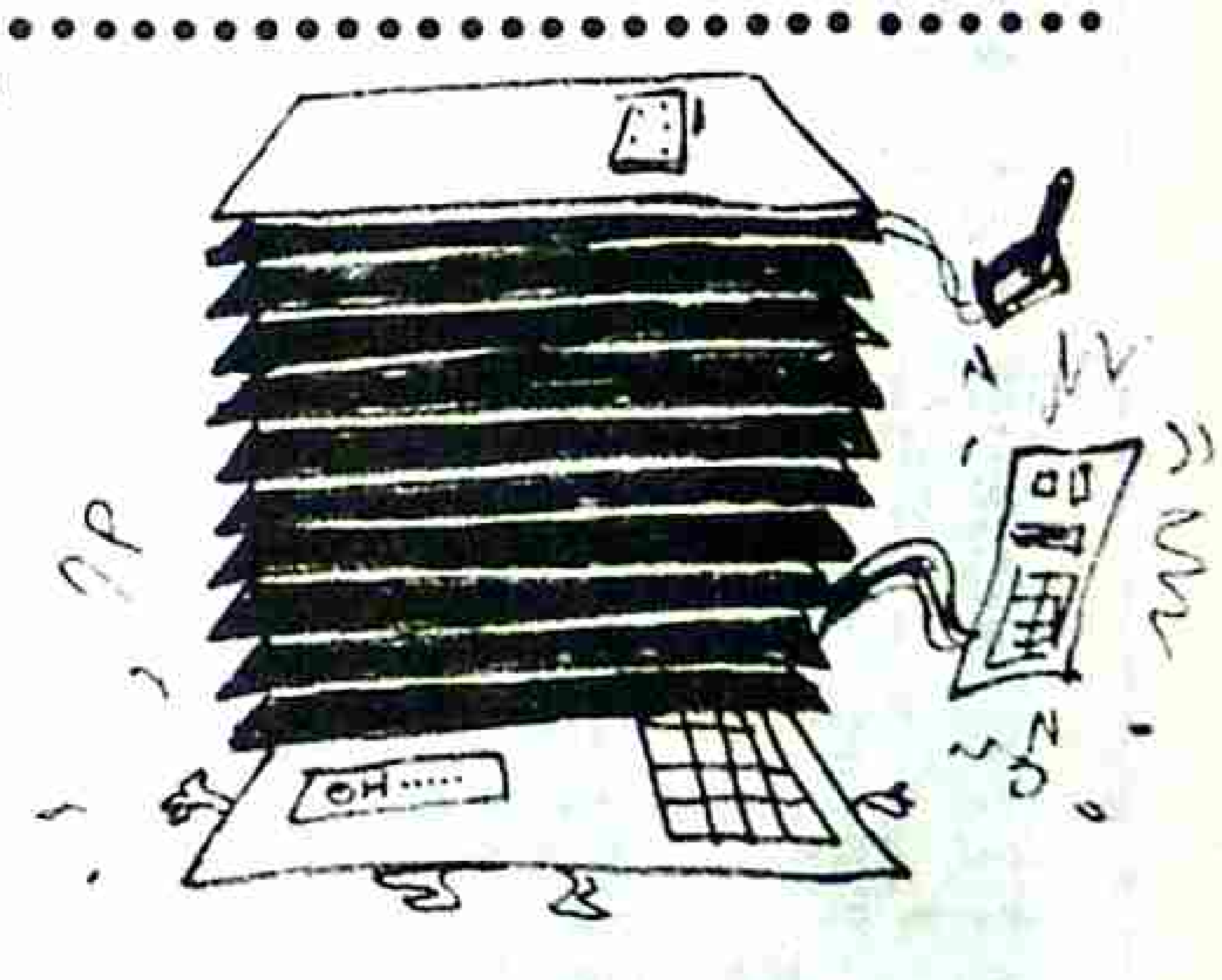
We have learnt the major advantage of a **JR** instruction. It enables a program to be transferred to another location without having to alter any of the data.

See the effectiveness of this. Move the whole Quick Draw game to **0900** or **0A00** making sure you wipe the program at**0800** before starting at the new location.

To start the game, push RESET, GO, GO. Is it a success?

Your final program will look like this:

```
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
JR 820           809      18 14
RLC B            80B      CB 00
JR 806           80D      18 F7

                 820      11 FF FF
                 823      1B
                 824      7B
                 825      B2
JR NZ 823        826      20 FB
JR 80B           828      18 E0
```



...AND REMOTE CONTROL...

TONES & TUNES

Here is a selection of tones and tunes for the computer. These have been submitted by readers and are presented in various formats to get you acquainted with the different ways of presenting a program.

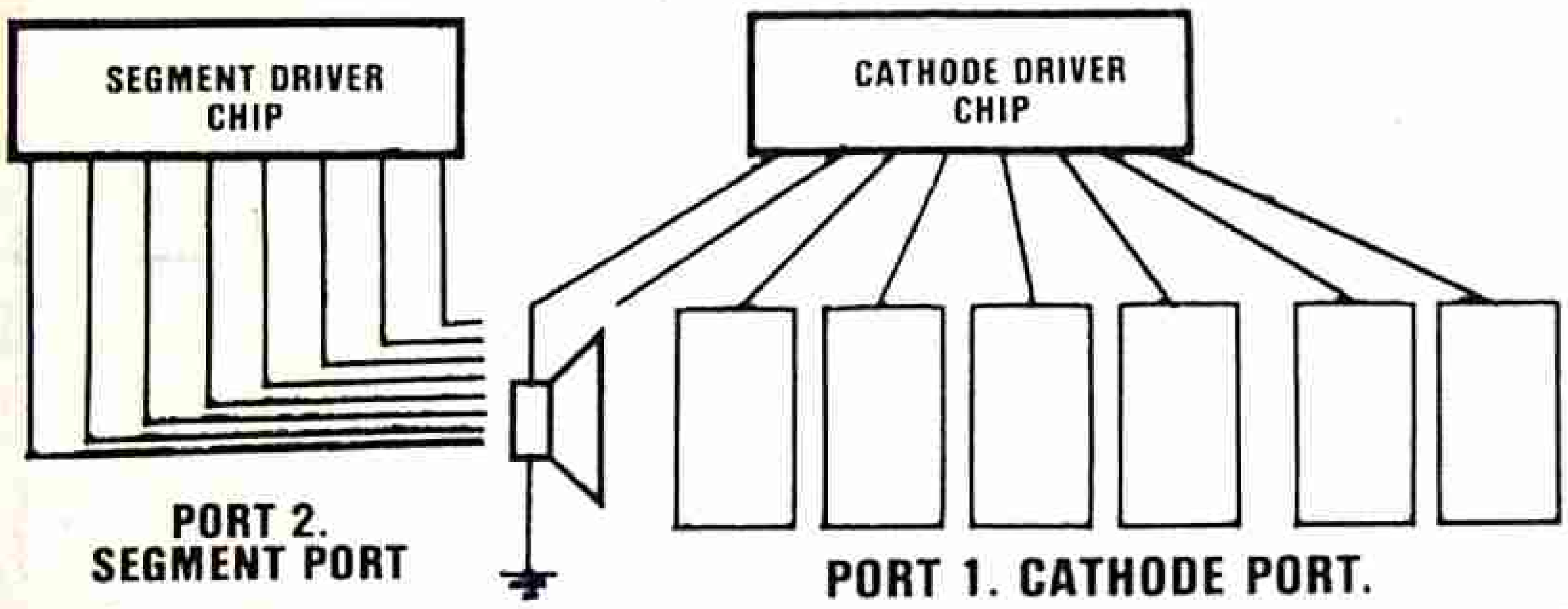
OSCILLATOR

by Peter Aleksejevs

```
800   3E  LD A,80
801   80
802   D3  OUT (01),A
803   01
804   3E  LD A,00
805   00
806   D3  OUT (01),A
807   01
808   C3  JP 800
809   00
80A   08
```

The principle of operation of this program can be seen in the diagram. We are accessing the speaker via port 1 and this is the 8th line of the driver chip. Thus the value 80 is inserted in the program.

We load a HIGH into this line for a number of clock cycles and then a LOW. This produces a CLICK which sounds like an oscillator when the speed control is increased.



TOCCATA

-by Stephen Clarke, 2774.

```
800: 00 00 3E 00 32 00 08 3E 09 32 01 08 CD B0 01 CD
810: B0 01 3E 21 32 00 08 CD B0 01 3E 62 32 00 08 CD
820: B0 01 C3 02 08

900: 0A 0F 11 12 0F 11 12 14 11 12 14 16 12 14 16 17
910: 14 16 12 14 11 12 0F 11 0E 0F 0A 0B 08 0A 0A 00
920: 1F 00 17 17 17 17 16 16 16 16 14 14 14 14 12 12
930: 12 12 11 11 11 11 0F 0F 0F 0F 0E 0E 0E 0E 0E 0E
940: 0E 0E 0B 0B 0B 0B 0A 0A 0A 0A 08 08 08 08 06 06
950: 06 06 05 05 05 05 03 03 03 03 02 02 02 02 02 02
960: 00 1F 0A 0F 11 12 0F 11 12 14 11 12 14 16 12 14
970: 16 17 14 16 12 14 11 12 0F 11 0E 0F 0A 0B 08 0A
980: 08 06 05 03 03 03 03 03 03 03 03 03 00 00 00 00
990: 00 00 00 00 00 00 1F
```

FREQUENCY SWEEP

by Peter Aleksejevs

This program gives an effect similar to a phaser gun. By changing the value of the second byte, different effects can be generated.

This program can be placed anywhere in memory as it consists entirely of **JR** instructions.

```
LD H,FF       26 FF
LD B,H        44
LD A,00       3E 00
OUT (1),A     D3 01
LD A,80       3E 80
OUT (1),A     D3 01
LD A,B        78
DEC A         3D
JR NZ FD      20 FD
DJNZ F2       10 F2
LD B,00       06 00
LD A,00       3E 00
OUT (1),A     D3 01
LD A,80       3E 80
OUT (1),A     D3 01
INC B         04
LD A,B        78
DEC A         3D
JR NZ FD      20 FD
LD A,H        7C
SUB B         90
JR NZ EF      20 EF
JP DA         18 DA
```

PHONE RING

By Cris Cogdon.

This program generates a ring similar to that of a new phone. It would make an ideal trick if you have one of these phones!

```
START   CALL RING      800   CD 14 08
        LD HL,1000     803   21 00 10
        CALL PAUSE     806   CD 1E 08
        CALL RING      809   CD 14 08
        LD HL,8000     80C   21 00 80
        CALL PAUSE     80F   CD 1E 08
        JR START       812   18 EC

RING    LD B,10        814   06 10
XRING   PUSH BC        816   C5
        CALL 081E      817   CD 8E 01
        POP BC         81A   C1
        DJ NZ XRING    81B   10 F9
        RETURN         81D   C9

PAUSE   DEC HL         81E   2B
        LD A,H         81F   7C
        OR L           820   B5
        RET Z          821   C8
        JR PAUSE       822   18 FA
```

THE STRIPPER

```
800: 00 00 3E 00 32 00 08 3E 09 32 01 08 CD B0 01 C3
810: 02 08

900: 01 01 03 03 03 06 06 06 06 06 06 0A 08 08 06 06
910: 06 02 02 02 02 02 02 01 01 03 03 03 06 06 06 06
920: 06 06 0A 0D 0D 0C 0C 0C 0B 0B 0B 0B 0B 0B 0B 0A
930: 0A 01 01 01 01 0A 0A 01 01 01 01 0A 0A 09 0A 0A
940: 0A 0B 0B 0A 0B 0B 0C 0D 0D 0C 0D 0D 0D 0E 0E 0D
950: 0E 0E 0E 0F 0F 0E 0F 0F 11 11 11 0F 11 11 12 12
960: 12 12 12 12 12 00 00 00 00 00 00 00 00 00 00 1F
```

TEC CLOCK

This program will allow the TEC to be used as a CLOCK. The display is used as the readout and the time can be set as shown opposite.

This is a 24 hour clock and its accuracy depends on the setting of the SPEED CONTROL. In a future issue we will present a crystal oscillator to take the place of the 4049 to turn the TEC into an accurate time-piece.

To set CLOCK:

at 989: insert seconds
at 98A: insert minutes.
at 98B: insert hours.

Example: 7:45:32

989: 32 98A: 45 98B: 07

START	**LD IY, Clock Buffer**	**900**	**FD 21 89 09**	Load pointer to clock counting buffer
	LD B,2	**904**	**06 02**	load number of 60's to be tested
	LD A,(IY +0)	**906**	**FD 7E 00**	Read first clock buffer value
	ADD A,01	**909**	**C6 01**	add 1 to the value
	DAA	**90B**	**27**	decimal adjust the accumulator
	CP 60	**90C**	**FE 60**	TEST A=60 sec/min
	JR NZ,DISP	**90E**	**20 13**	GOTO 'DSP' if not equal
	XOR A	**910**	**AF**	ZERO the accumulator
	LD (IY+0),A	**911**	**FD 77 00**	Store A in clock buffer
	INC IY	**914**	**FD 23**	Advance pointer
	DJNZ EE	**916**	**10 EE**	complete LOOP if B is not zero
	LD A,(IY +0)	**918**	**FD 7E 00**	Read hours value
	ADD A,01	**91B**	**C6 01**	Increment hours value
	DAA	**91D**	**27**	Decimal adjust the accumulator
	CP 24H	**91E**	**FE 24**	TEST hours =24
	JR NZ,DISP	**920**	**20 01**	If not GOTO 'DSP'
	XOR A	**922**	**AF**	ZERO A
DISP	**LD (IY + 0),A**	**923**	**FD 77 00**	Store hours in clock buffer
	LD B,03	**926**	**06 03**	Load number of bytes to be converted
	LD HL,DISP BUF +6	**928**	**21 92 09**	Load pointer to display buffer
	LD IX,CLK BUF	**92B**	**DD 21 89 09**	Load pointed to clock buffer
LOOP 1	**LD A,(IX + 0)**	**92F**	**DD 7E 00**	Read CLOCK BUFFER value
	INC IX	**932**	**DD 23**	Advance pointer by 1
	PUSH BC	**934**	**C5**	Save BC contents
	PUSH AF	**935**	**F5**	Save contents of A
	AND 0F	**936**	**E6 0F**	Get least significant 4 bits
	LD B,A	**938**	**47**	Transfer A to B
	CALL LOOK	**939**	**CD 73 09**	Get pattern for B
	POP AF	**93C**	**F1**	Restore AF
	SRL A	**93D**	**CB 3F**	Shift A one place to the right
	SRL A	**93F**	**CB 3F**	
	SRL A	**941**	**CB 3F**	
	SRL A	**943**	**CB 3F**	
	LD B,A	**945**	**47**	Load A into B
	CALL LOOK	**946**	**CD 73 09**	Get bit pattern for B
	POP BC	**949**	**C1**	Restore BC
	DJNZ LOOP	**94A**	**10 E3**	Complete LOOP if B is not zero.
	LD B,0FFH	**94C**	**06 FF**	Load LOOP value
LOOP 2	**LD IX,DISP BUF**	**94E**	**DD 21 8C 09**	Load pointer to digit patterns
	PUSH BC	**952**	**C5**	Save BC contents
	LD B,07	**953**	**06 07**	Load number of digits to be displayed
	LD C,40H	**955**	**0E 40**	Load bit pattern for display cathodes
	LD A,(IX +0)	**957**	**DD 7E 00**	Read display pattern
	OUT (2),A	**95A**	**D3 02**	Output pattern to port 2
	LD A,C	**95C**	**79**	Load C into A
	OUT (1),A	**95D**	**D3 01**	Output cathode pattern to port 1
	SRL C	**95F**	**CB 39**	Move cathode bit one place for MUX effect
	XOR A	**961**	**AF**	Clear A
	LD E,10H	**962**	**1E 10**	Load TIME DELAY value
	DEC E	**964**	**1D**	Decrement E
	JR NZ FD	**965**	**20 FD**	LOOP if not equal to zero
	OUT (1),A	**967**	**D3 01**	Turn off anode bits
	INC IX	**969**	**DD 23**	Advance to next pattern
	DJNZ loop 2	**96B**	**10 EA**	LOOP if not zero
	POP BC	**96D**	**C1**	Restore BC
	DJNZ LOOP 2	**96E**	**10 DE**	LOOP if all digits not displayed
	JP START	**970**	**C3 00 09**	Jump to START
LOOK	**LD DE, DISP**	**973**	**11 7F 09**	Load DE with display pattern
	PUSH AF	**976**	**F5**	Save AF
	LD A,E	**977**	**7B**	Load E into A
	ADD A,B	**978**	**80**	Calculate pattern address
	LD E,A	**979**	**5F**	Load A into E
	LD A,(DE)	**97A**	**1A**	Read pattern
	LD (HL),A	**97B**	**77**	Store pattern in display buffer
	DEC HL	**97C**	**2B**	Decrement HL
	POP AF	**97D**	**F1**	Restore AF
	RETURN	**97E**	**C9**	End of sub-routine
	DISP PATTERN: EB, 28, CD, AD, 2E, A7, E7, 29, EF, AF.	**97F**		
	CLOCK BUFFER	**989**		
	DISP BUFFER	**98C**		

SPIROID ALIENS

-by M Allison, 3095

This is quite a long program and shows the length of listing required to achieve a degree of realism. The game uses all of page **0800** and portions of **0900, 0A00, 0B00** and **0D00.**

The main program is at **0800** with calls at the other pages.

The game consists of unusual-shaped aliens passing across the display. Each game consists of 16 passes and you must shoot down the arrivals by pressing buttons 1, 2 or 3. To win, you must shoot down at least 11.

In the initial stages of the game, you must acquaint yourself with the connection betweeen the spiroid shapes and buttons 1, 2, 3. After this you will be ready to launch an attack.

Here's the listing:

```
Reserved for
message.        800  00 08
Blank           802  00
LD HL,903       803  21 03 09
LD A,80         806  3E 80
LD (HL),A       808  77
INC HL          809  23
LD A,00         80A  3E 00
LD C,A          80C  4F
LD (HL),A       80D  77
LD HL,0911      80E  21 11 09
LD A,20         811  3E 20
LD (HL),A       813  77
INC HL          814  23
LD A,00         815  3E 00
LD (HL),A       817  77
LD HL, 0849     818  21 49 08
LD A,01         81B  3E 01
LD (HL),A       81D  77
LD B,10         81E  06 10
LD D,00         820  16 00
LD IY,0865      822  FD 21 65 08
CALL 0D00       826  CD 00 0D
LD A,(0865)     829  3A 65 08
CP C            82C  B9
JR Z, 0826      82D  28 F7
LD HL,0849      82F  21 49 08
                832  00
CP 01           833  FE 01
JR Z,083F       835  28 08
CP 02           837  FE 02
JR Z,0843       839  28 08
LD A,61         83B  3E 61
JR, 0845        83D  18 06
LD A,0F         83F  3E 0F
JR 0845         841  18 02
LD A,26         843  3E 26
LD (084D),A     845  32 4D 08
LD A( )         848  3E ( )
OUT A,(01)      84A  D3 01
LD A,(SYMBOL)   84C  3E ( )
OUT A,(02)      84E  D3 02
CALL 0900       850  CD 00 09
CALL 090E       853  CD 0E 09
XOR,A           856  AF
IN A(00)        857  DB 00
```

```
LD E,A          859  5F
LD A,72         85A  3E 72
CP E            85C  BB
JR Z,0861       85D  20 02
LD E,00         85F  1E 00
LD A,E          861  7B
AND 03          862  E6 03
CP (CODE)       864  FE ( )
JR Z,0871       866  28 09
SLA (HL)        868  CB 26
LD A,40         86A  3E 40
CP (HL)         86C  BE
JR Z,0878       86D  28 09
JR 0848         86F  18 D7
LD C,A          871  4F
INC D           872  14
CALL 0B00       873  CD 00 0B
JR 087D         876  18 05
LD C,00         878  0E 00
CALL 0A00       87A  CD 00 0A
LD A,01         87D  3E 01
LD (HL),A       87F  77
DJNZ 0822       880  10 A2
LD HL,0807      882  21 07 08
LD A,0B         885  3E 0B
CP D            887  BA
JR C,08A5       888  38 1B
LD A,(HL)       88A  7E
CP F0           88B  FE F0
JR Z,0842       88D  28 03
ADD A,10        88F  C6 10
LD (HL),A       891  77
LD HL,0800      892  21 00 08
LD A,F8         895  3E F8
LD (HL),A       897  77
PUSH HL         898  E5
CALL 01B0       899  CD B0 01
POP HL          89C  E1
LD A,E8         89D  3E E8
LD (HL),A       89F  77
CALL 0270       8A0  CD 70 02
JR 08BE         8A3  18 19
LD A,(HL)       8A5  7E
CP 10           8A6  FE 10
JR Z,08AD       8A8  28 03
SUB 10          8AA  D6 10
LD (HL),A       8Ac  77
LD HL,0800      8AD  21 00 08
LD A,DE         8B0  3E DE
LD (HL),A       8B2  77
PUSH HL         8B3  E5
CALL 01B0       8B4  CD B0 01
POP HL          8B7  E1
LD A,CA         8B8  3E CA
LD (HL),A       8BA  77
CALL 0270       8BB  CD 70 02
LD A,3F         8BE  3E 3F
OUT A,(01)      8C0  D3 01
LD A,8A         8C2  3E 8A
OUT A,(02)      8C4  D3 02
HALT            8C6  76
JP 0802         8C7  C3 02 08
Messages:       8CA  00
                8CB  01
                8CC  0C
                8CD  09
                8CE  05
                8CF  0D
                8D0  12
                8D1  00
                8D2  04
                8D3  05
                8D4  12
                8D5  13
                8D6  11
                8D7  0E
                8D8  16
                8D9  05
                8DA  04
                8DB  1A
                8DC  00
                8DD  1F
                8DE  04
                8DF  00
                8E0  04
                8E1  00
                8E2  04
                8E3  00
                8E4  01
                8E5  01
                8E6  01
                8E7  1F
                8E8  00
                8E9  05
                8EA  0D
                8EB  04
                8EC  00
                8ED  0E
                8EE  06
                8EF  00
                8F0  05
                8F1  01
                8F2  11
                8F3  13
                8F4  08
                8F5  1A
                8F6  00
                8F7  1F
                8F8  01
                8F9  1A
                8FA  01
                8FB  1A
                8FC  01
                8FD  1A
                8FE  1F
```

```
PUSH AF         B00  F5
PUSH BC         B01  C5
PUSH DE         B02  D5
PUSH HL         B03  E5
LD HL(0903)     B04  2A 03 09
PUSH HL         B07  E5
LD HL(0911)     B08  2A 11 09
                B0B  E5
LD HL,0912      B0C  21 12 09
LD A,00         B0F  3E 00
LD(HL),A        B11  77
DEC HL          B12  2B
LD A,20         B13  3E 20
ld (HL),A       B15  77
LD HL,0904      B16  21 04 09
LD A,00         B19  3E 00
LD(HL),A        B1B  77
DEC HL          B1C  2B
LD A,24         B1D  3E 24
LD (HL),A       B1F  77
LD HL,0B35      B20  21 35 0B
LD A,01         B23  3E 01
LD (HL),A       B25  77
EXX             B26  D9
LD DE,0904      B27  11 04 09
LD C,00         B2A  0E 00
LD HL,0B68      B2C  21 68 0B
LD B,06         B2F  06 06
LD A,01         B31  3E 01
LD (DE),A       B33  12
LD A,01         B34  3E 01
OUT (01),A      B36  D3 01
LD A,(HL)       B38  7E
OUT (02),A      B39  D3 02
```

CALL 0900	B3B	CD 00 09
LD A,00	B3E	3E 00
LD (DE),A	B40	12
CALL 090E	B41	CD 0E 09
INC HL	B44	23
DEC DE	B45	1B
EX DE,HL	B46	EB
DEC (HL)	B47	35
EX DE,HL	B48	EB
INC DE	B49	13
DJNZ 0B31	B4A	10 E5
EXX	B4C	D9
SLA (HL)	B4D	CB 26
EXX	B4F	D9
INC C	B50	0C
LD A,06	B51	3E 06
CP C	B53	B9
JP Z,0B5A	B54	CA 5A 0B
JP 082C	B57	C3 2C 0B
Exx	B5A	D9
POP HL	B5B	E1
LD (0911),HL	B5C	22 11 09
POP HL	B5F	E1
LD (0903),HL	B60	22 03 09
POP HL	B63	E1
POPDE	B64	D1
POPBC	B65	C1
POPAF	B66	F1
RETURN	B67	C9
LOOK-UP TABLE FOR SPIRAL	B68	01
	B69	09
	B6A	29
	B6B	A9
	B6C	E9
	B6D	EB

PUSH AF	A00	F5
PUSH BC	A01	C5
PUSH HL	A02	E5
LD HL(0903)	A03	2A 03 09
PUSH HL	A06	E5
LD HL(0911)	A07	2A 11 09
PUSH HL	A0A	E5
LD B,09	A0B	06 09
LD HL,0911	A0D	21 11 09
LD A,05	A10	3E 05
LD (HL),A	A12	77
INC HL	A13	23
LD A,00	A14	3E 00
LD (HL),A	A16	77
LD HL,0903	A17	21 03 09
LD A,1F	A1A	3E 1F
LD (HL),A	A1C	77
INC HL	A1D	23
LD A,00	A1E	3E 00
LD (HL),A	A20	77
DEC HL	A21	2B
CALL 090E	A22	CD 0E 09
DEC (HL)	A25	35
LD A,01	A26	3E 01
CP (HL)	A28	BE
JP Z 0A2F	A29	CA 2F 0A
JP 0A22	A2C	C3 22 0A
DJNZ 0A1A	A2F	10 E9
POP HL	A31	E1
LD (0911),HL	A32	22 11 09
POP HL	A35	E1
LD (0903),HL	A36	22 03 09
POP HL	A39	E1
POP BC	A3A	C1
POP AF	A3B	F1
RETURN	A3C	C9

LD A,R	C00	ED 5F
CALL 03B5	C02	CD B5 03
AND 03	C05	E6 03
LD E,A	C07	5F
LD A,00	C08	3E 00
CP E	C0A	BB
JR Z 0C00	C0B	28 F3
LD A,E	C0D	7B
CP C	C0E	B9
JR Z 0C00	C0F	28 EF
RETURN	C11	C9

PUSH HL	D00	E5
PUSH BC	D01	C5
LD HL,0D06	D02	21 06 0D
LD B,01	D05	06 01
LD A,R	D07	ED 5F
DJNZ, 0D07	D09	10 FC
AND 08	D0B	E6 08
PUSH HL	D0D	E5
LD LH,0D33	D0E	21 33 0D
ADD A,L	D11	85
LD L,A	D12	6F
LD E,HL	D13	5E
LD HL,0D33	D14	21 33 0D
LD B,08	D17	06 08
LD C(HL)	D19	4E
INC HL	D1A	23
LD A,(HL)	D1B	7E
DEC HL	D1C	2B
LD (HL),A	D1D	77
INC HL	D1E	23
DJNZ,0D1A	D1F	10 F9
LD (HL),C	D21	71
POP HL	D22	E1
INC (HL)	D23	34
LD A,20	D24	3E 20
CP (HL)	D26	BE
JR Z,0D2F	D27	28 06
	D29	FD 73 00
	D2C	C1
	D2D	E1
	D2E	C9
	D2F	36 01
	D31	18 F6
LOOK-UP TABLE FOR RANDOM NUMBERS	D33	01
	D34	02
	D35	03
	D36	01
	D37	02
	D38	03
	D39	01
	D3A	02
	D3B	03

Finally, the listing at **0900** must be inserted. This listing can be found in issue 11 P 36, under the heading **ALIENS ATTACK RUN.** This will provide the sound for the game.

This completes the listing. Before pushing **RESET, GO**, it is a very good idea to go through the complete listing again and double-check each of the machine code values. The reason for this is to prevent the program **SELF DESTRUCTING**. This could happen if you placed the wrong value in one of the locations which caused the computer to write over some of the contents of the program.

PUSH & POP

PUSH and **POP** are very much like **PUSH** and **PULL**. They are operations which transfer the contents of a register-pair to a holding area so that the registers can be used for other operations. This holding area is called the **STACK.**

We say register PAIR because the operations **PUSH** and **POP** require that 2 registers be specified. Thus, if the accumulator (Register A) is required to be pushed onto the stack, we combine it with the FLAGS register to get the register pair: AF.

There are a few technical complications concerning the placement of bytes onto the stack but these will not concern us at this stage. It is sufficient to say that the stack is located at the top end of the RAM, (about 8 - 10 bytes from the top)and as each new set of bytes is placed on the stack, the pile grows DOWNWARDS, towards the program we are executing.

We have already seen the effect of placing (PUSHING) more and more bytes onto the stack (issue 11, P. 12) and for this reason we must use the stack very carefully. Otherwise it will increase downwards and and crash into our program!

Basically we PUSH one pair of bytes onto the stack (from say register-pair AF) then push another pair of bytes onto the stack from say register pair HL. This will leave the accumulator and HL registers free for other operations.

If we want to get the 2 bytes of AF from the stack, we must firstly POP the two bytes from HL and then we can get the AF pair. It is a simple principle of **LAST ON, FIRST OFF.**

Pushing and popping are very handy instructions. By using a PUSH instruction at the start of a routine and a POP at the end, we can place a routine such as a delay routine, which will not affect the registers at all. This routine is said to be TRANSPARENT.

PUSHING and POPPING can take place between the stack and register pairs including the index registers. This group consists of the following: AF, BC, DE, HL,IX and IY.

It is interesting to note that the bytes are pushed onto the stack HIGH BYTE first, then LOW BYTE. They come off the stack LOW BYTE then HIGH BYTE. But because the stack is increasing DOWNWARDS, each byte placed onto the stack will have a lower address!

In the programs we have presented you can see PUSH and POP in operation. The stack is a temporary holding area and only the top pair can be accessed.

MORE PROGRAMS FOR THE 8x8 DISPLAY:

The 8x8 matrix was a very popular 'add-on', with nearly every TEC owner building up a display.

Here are some more programs for the matrix, commencing with a simple routine similar to the **FAN OUT** on P.34 of issue 11.

FAN OUT MK II

```
LD A,01         800   3E 01
OUT (3),A       802   D3 03
OUT (4),A       804   D3 04
RLA             806   07
PUSH AF         807   F5
CALL DELAY      808   CD 00 09
POP AF          80B   F1
INC A           80C   3C
JP NZ 802       80D   C2 02 08
LD A,FE         810   3E FE
OUT (3),A       812   D3 03
OUT (4),A       814   D3 04
RLA             816   07
PUSH AF         817   F5
CALL DELAY      818   CD 00 09
POP AF          81B   F1
DEC A           81C   3D
JP NZ 812       81D   C2 12 08
JP 802          820   C3 02 08
```

Delay at 0900:

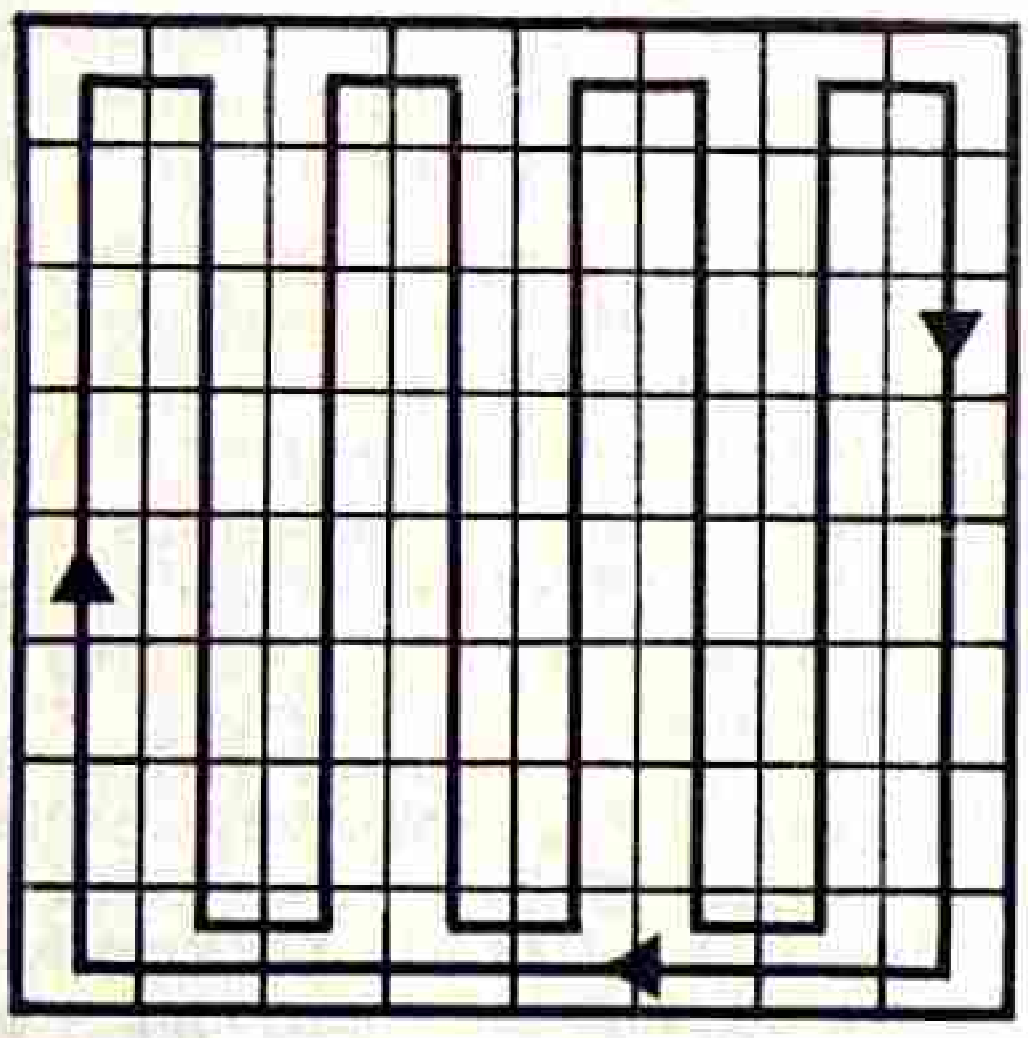
```
11 FF 06
1B 7B B2
C2 03 09
C9
```

BOUNCING BALL

by G L Dunt, 3219.

Bouncing Ball is an extension of **'AROUND THE DISPLAY'** (issue 11, P.29).

The diagram below shows the effect produced by this program and by varying the delay, it will appear as if two or more LEDs are circulating the display.



```
800   3E 01
802   D3 03
804   0E 08
806   3E 01
808   D3 04
80A   47
80B   CD 00 0C
80E   78
80F   CB 07
811   0D
812   C2 08 08
```

DELAY AT 0C00:

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

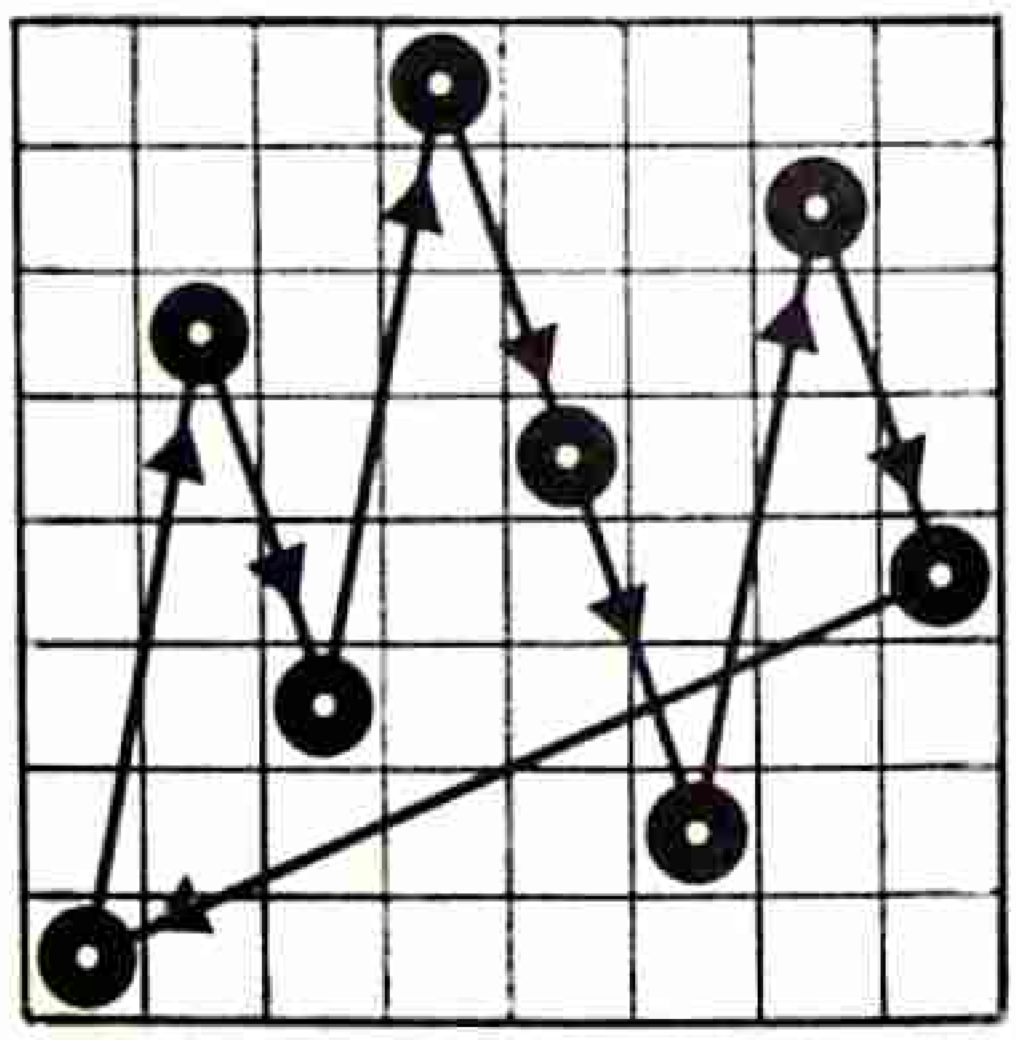
Type the first section into the TEC and **RUN**. This will check the code-values and prevent a major mistake. Type the second stage and **RUN**. Continue this way until the whole program has been inserted.

```
815   3E 02
817   D3 03
819   0E 08
81B   3E 80
81D   D3 04
81F   47
820   CD 00 0C
823   78
824   CB 0F
826   0D
827   C2 1D 08

82A   3E 04
82C   D3 03
82E   0E 08
830   3E 01
832   D3 04
834   47
835   CD 00 0C
838   78
839   CB 07
83B   0D
83C   C2 32 08

83F   3E 08
841   D3 03
843   0E 08
845   3E 80
847   D3 04
849   47
84A   CD 00 0C
84D   78
84E   CB 0F
850   0D
851   C2 47 08

854   3E 10
856   D3 03
858   0E 08
85A   3E 01
85C   D3 04
85E   47
95F   CD 00 0C
862   78
863   CB 07
865   0D
866   C2 5C 08

869   3E 20
86B   D3 03
86D   0E 08
86F   3E 80
871   D3 04
873   47
874   CD 00 0C
877   78
878   CB 0F
87A   0D
87B   C2 71 08

87E   3E 40
880   D3 03
882   0E 08
884   3E 01
886   D3 04
888   47
889   CD 00 0C
88C   78
88D   CB 07
88F   0D
890   C2 86 08

893   3E 80
895   D3 03
897   0E 08
899   3E 80
89B   D3 04
89D   47
89E   CD 00 0C
8A1   78
8A2   CB 0F
8A4   0D
8A5   C2 9B 08

8A8   3E 01
8AA   D3 04
8AC   0E 06
8AE   3E 40
8B0   D3 03
8B2   47
8B3   CD 00 0C
8B6   78
8B7   CB 0F
8B9   0D
8BA   C2 B0 08
8BD   C3 00 08
```

JUMPING LEDs.

- by G L Dunt, 3219.

This program demonstrates multiplexing in an easily understood manner.



By adjusting the SPEED CONTROL, the flickering effect of each LED will be speeded-up to give a steady pattern.

```
LD A,01        800   3E 01
OUT (3),A      802   D3 03
OUT (4),A      804   D3 04
CALL DELAY     806   CD 00 0C
LD A,02        809   3E 02
OUT (3),A      80B   D3 03
LD A,20        80D   3E 20
OUT (4),A      80F   D3 04
CALL DELAY     811   CD 00 0C
LD A,04        814   3E 04
OUT (3),A      816   D3 03
OUT (4),A      818   D3 04
CALL DELAY     81A   CD 00 0C
LD A,08        81D   3E 08
OUT (3),A      81F   D3 03
LD A,80        821   3E 80
OUT (4),A      823   D3 04
CALL DELAY     825   CD 00 0C
LD A,10        828   3E 10
OUT (3),A      82A   D3 03
OUT (4),A      82C   D3 04
CALL DELAY     82E   CD 00 0C
LD A,20        831   3E 20
OUT (3),A      833   D3 03
LD A,02        835   3E 02
OUT (4),A      837   D3 04
CALL DELAY     839   CD 00 0C
LD A,40        83C   3E 40
OUT (3),A      83E   D3 03
OUT (4),A      840   D3 04
CALL DELAY     842   CD 00 0C
LD A,80        845   3E 80
OUT (3),A      847   D3 03
LD A,08        849   3E 08
OUT (4),A      84B   D3 04
CALL DELAY     84D   CD 00 0C
JP 0800        850   C3 00 08
```

DELAY at 0C00:

```
11 0F 0F
1B
7B
B2
C2 03 0C
C9
```

Change delay to these values to create the full multiplexing effect.

at 0C00:

```
11 0D 01
11 6F 00
```

PRODUCING A LETTER

This extension to **JUMPING LEDs** program produces a letter of the alphabet. It will show the flexibility of multiplexing. Any figure or shape can be created on the screen.

The letter we will produce is the letter 'A'. This will be somewhat dimmer than when displaying one or two LEDs due to the current limitation of the latch at port 3. It cannot supply sufficient current to turn on 8 LEDs at the same time. A set of emitter-follower transistors would cure the problem.

```
LD B,08        800   06 08
LD A,01        802   3E 01
LD HL,0B00     804   21 00 0B
OUT (3),A      807   D3 03
PUSH AF        809   F5
LD A,(HL)      90A   7E
OUT(4),A       80B   D3 04
CALL DELAY     80D   CD 00 0A
INC HL         810   23
POP AF         811   F1
RLC A          812   CB 07
DEC B          814   05
JP NZ 807      815   C2 07 08
JP 800         818   C3 00 08
```

Delay at 0A00:

```
11 0F 01
1B
7B
B2
C2 03 0A
C9
```

at 0B00:

BYTE TABLE for letter A:

```
00
1F
3F
64
64
3F
1F
00
```

PODUCING A SHORT DELAY

When running the letter program above, you will find a disturbing flickering produced by the scan routine. This is basically due to the number of operations which must be carried out by the Z80 for each complete cycle of the program.

This takes a lot of clock cycles and the scan speed cannot be increased without increasing the clock frequency.

The solution is to provide a delay routine which requires less clock cycles for each loop.

This can be done by using the B register and an auto decrement function **DJNZ.** This will automatically decrement register B until it becomes zero.

At **0A00** the following delay routine is inserted:

```
PUSH BC        C5
LD B,FF        06 FF
DJNZ           10 FE
POP BC         C1
Return         C9
```

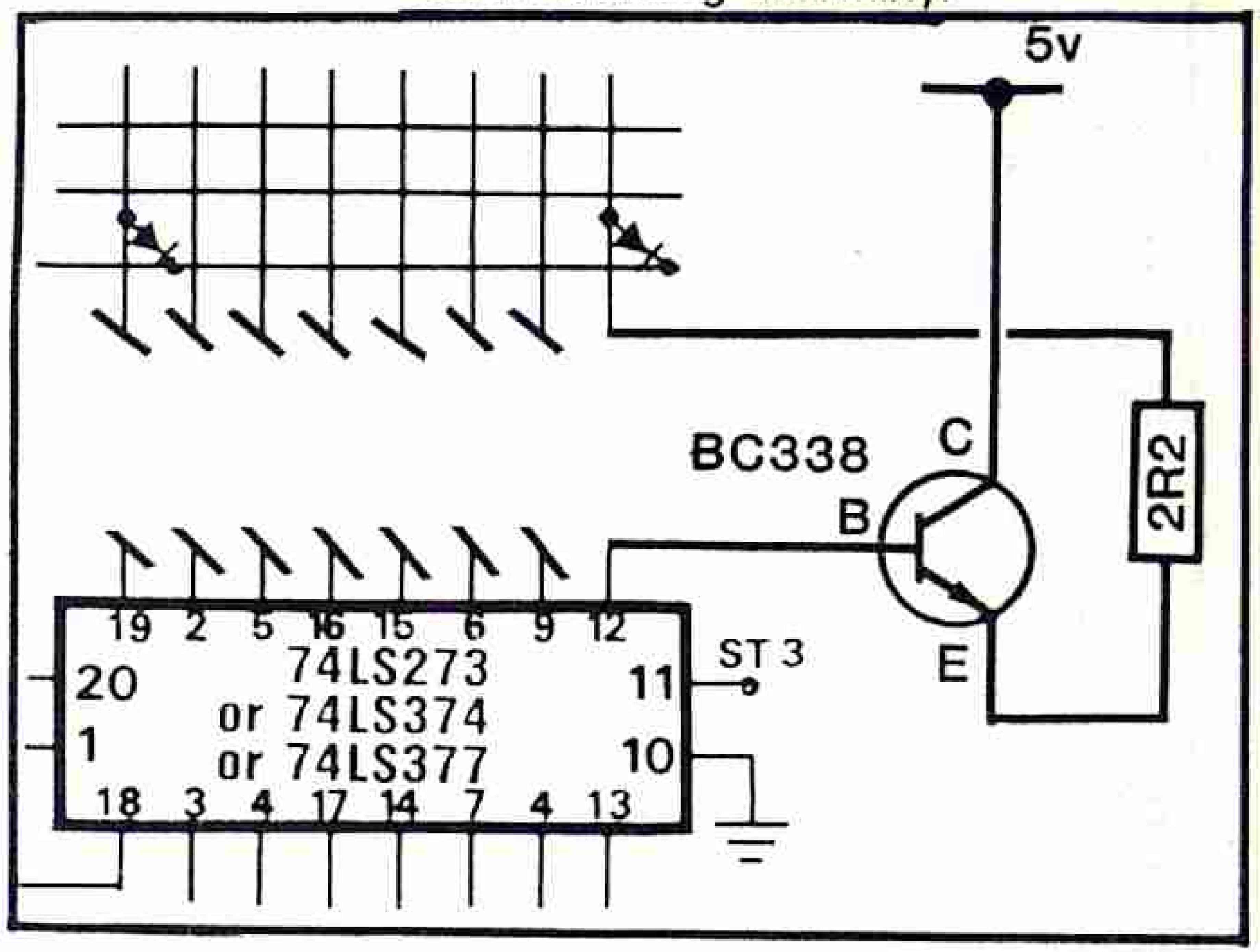
Note: The B register must be pushed onto the stack before it can be used as a decrementing register as it is alreay used in the main program to count the number of DATA BYTES.

To reduce the flicker even more, change the value of B for **FF** to **50**(or similar value). If the display is too dim, try our next modification:

INCREASING THE BRIGHTNESS OF THE 8x8

The brightness of the 8x8 can be dramatically improved by sourcing the display with a set of transistors.

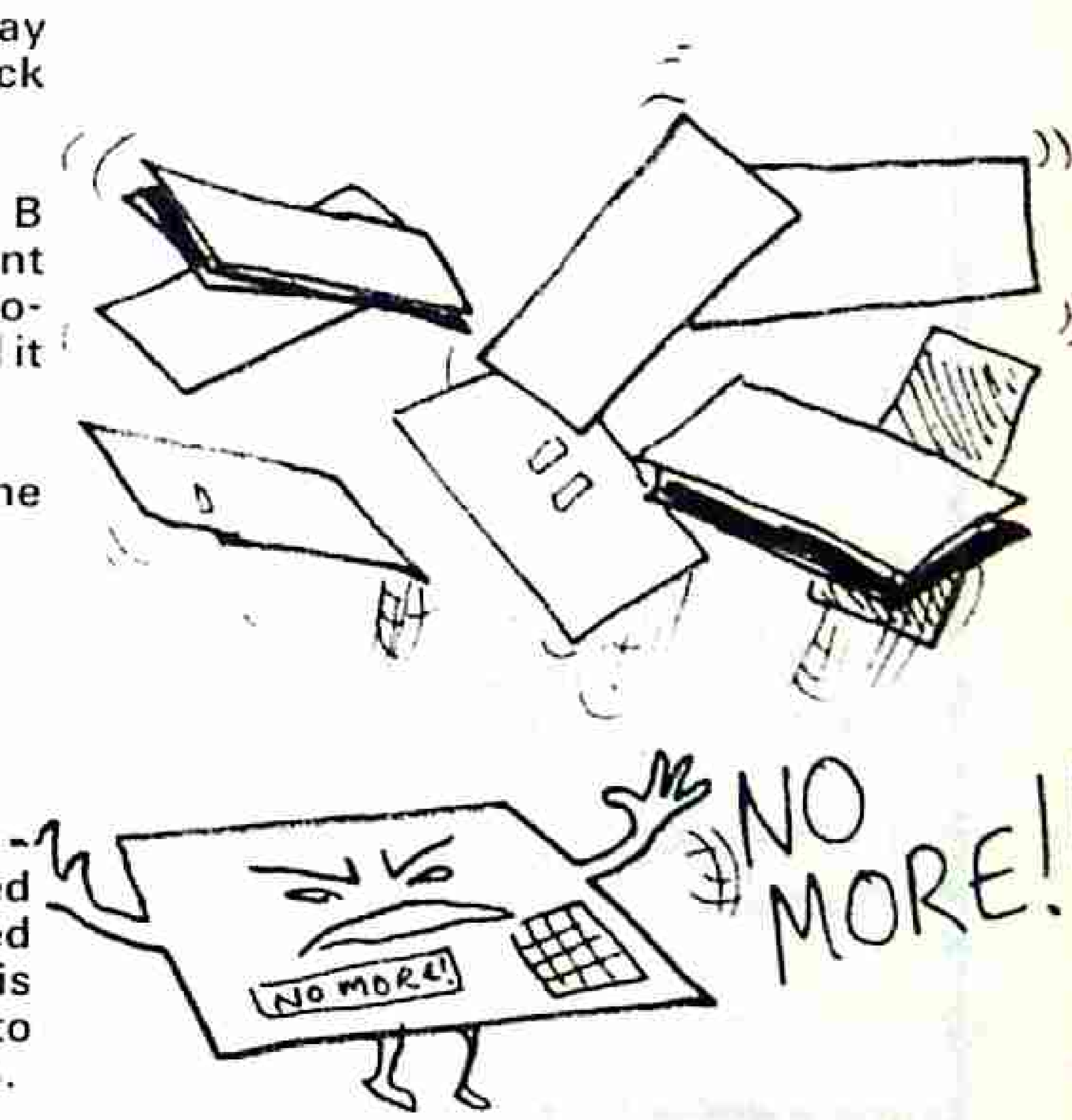
These are soldered under the PC in a row similar to the 8 sinking transistors. Don't forget to cut the PC tracks to each of the columns of LEDs before starting assembly.



This will enable you to start experimenting with different letters and shapes on the display and allow you to see them in a brightly lit room.

We will continue next issue with running these letters across the display in a similar manner to the running signs in shop windows etc.

AND NEXT MONTH.....



PRESENTING:

$9.70 per 2k
No PC board required.

OUR RAM STACK

ADD 12K TO THE TEC. OR ANY NUMBER OF 2K BLOCKS FOR RUNNING LARGER PROGRAMS.

Some of the best ideas are discovered by accident while others are over-looked for years because of their sheer simplicity.

Such is the case with our RAM STACK.

We have been thinking of a RAM PACK for a long time but never came up with an idea we really liked. Most ideas revolved around a PC board and trying to simplify the accompanying complexity of track-work.

Due to the parallel wiring requirement of memory chips, it is necessary to have PC tracks running between each of the pins.

This produces a very FINE set of tracks and consequently a number of problems arise. The most troublesome of these is the chance of a land being cut-in-two when the holes are being drilled. This causes a fine break in the track-work which must be repaired with solder when the components are being mounted on the board.

Failure to see any of these breaks will render some of the chips inoperative.

In addition, the closeness of the tracks highlights the need for a solder mask, all contributing to increasing the cost of the project.

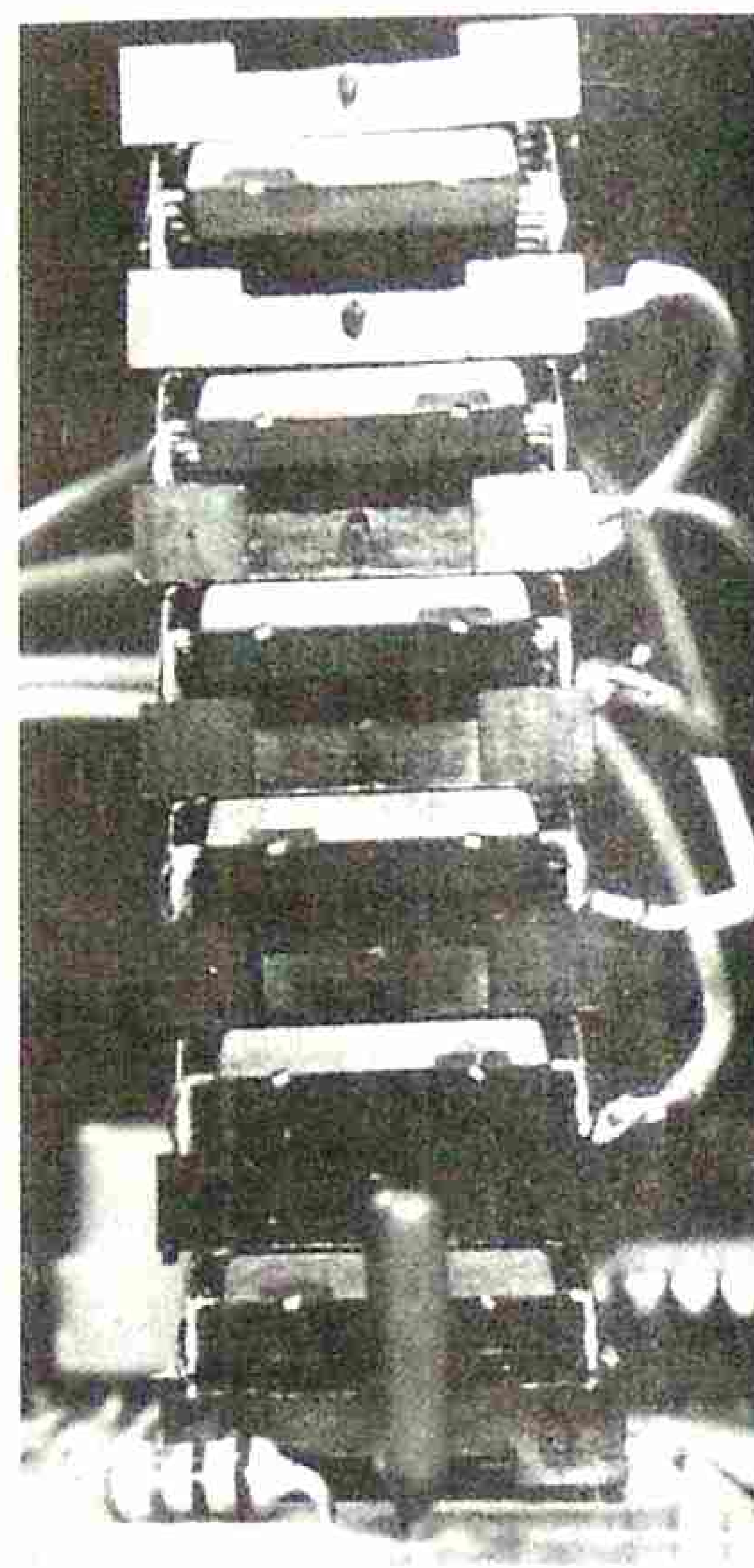
But a PC board can actually be eliminated.

The simplest and best design for a RAM pack requires no more than a set of IC's and sockets.

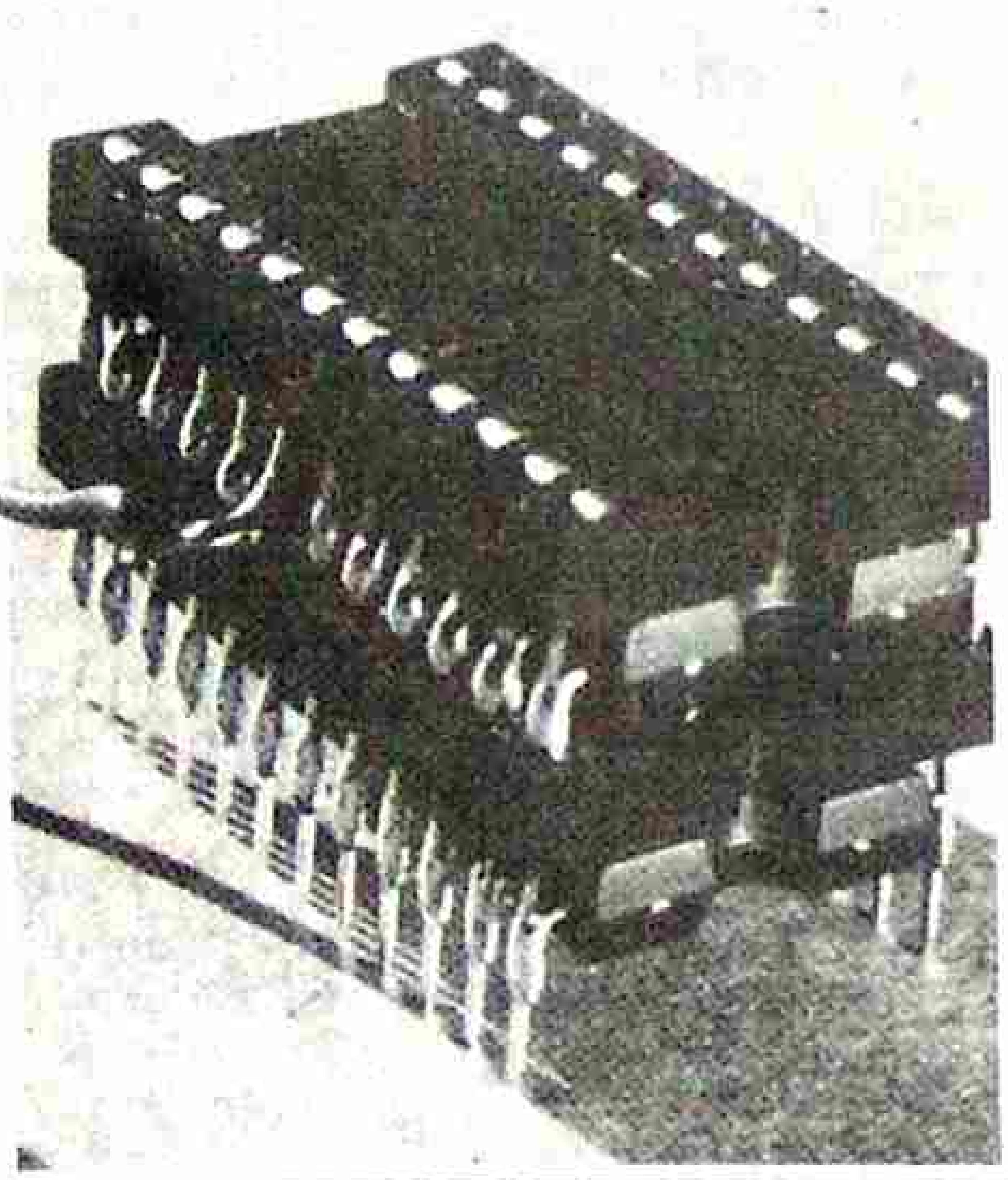
And that's when we struck upon our brilliant idea - a RAM STACK.

Not only is this design the cheapest arrangement possible, but it also incorporates a number of advantages. The best of these is memory can be increased or decreased in blocks of 2k for little more than the cost of an IC and socket. This will enable 2k or 4k to be an economical addition.

With our design, if a fault develops, each chip can be tested individually and removed if found to be defective.



Two 'units' piggy-backed together. The lower chip is accessed via the PC board; the top chip via the jumper lead.



PARTS LIST:

for each 2k:
1 - 6116 or equiv.
1 - 24 pin IC socket
1 - length of hook-up flex
1 - matrix pin & connector.

Putting all these advantages together you can see why we are pleased with this design. The accompanying photos shows how it goes together.

There isn't much to explain about construction. It's just a matter of placing an IC socket over a RAM chip and soldering each of the chip-pins to the socker pins.

Make sure the solder does not flow down any of the IC's pins otherwise you will not be able to insert the chip into a socket when putting the whole thing together.

The only other connection to each of the RAM chips is at pin 18. This is the **CHIP ENABLE** pin and an individual line is taken from one of the outputs of the 74LS 138 (near the oscillator chip), to this pin.

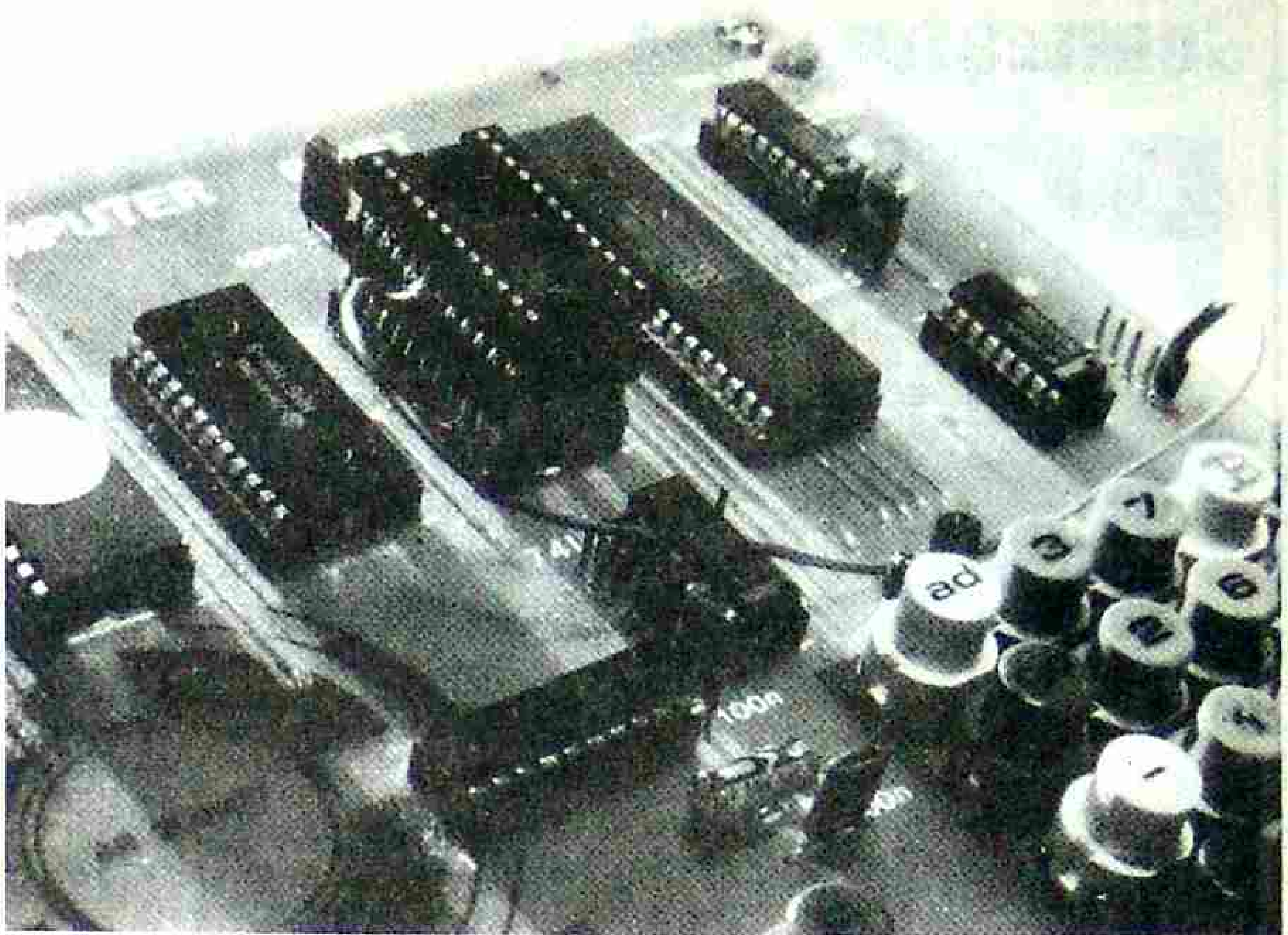
Each pin 18 of a memory chip must be kept separated from the others so that any chip can be individually selected.

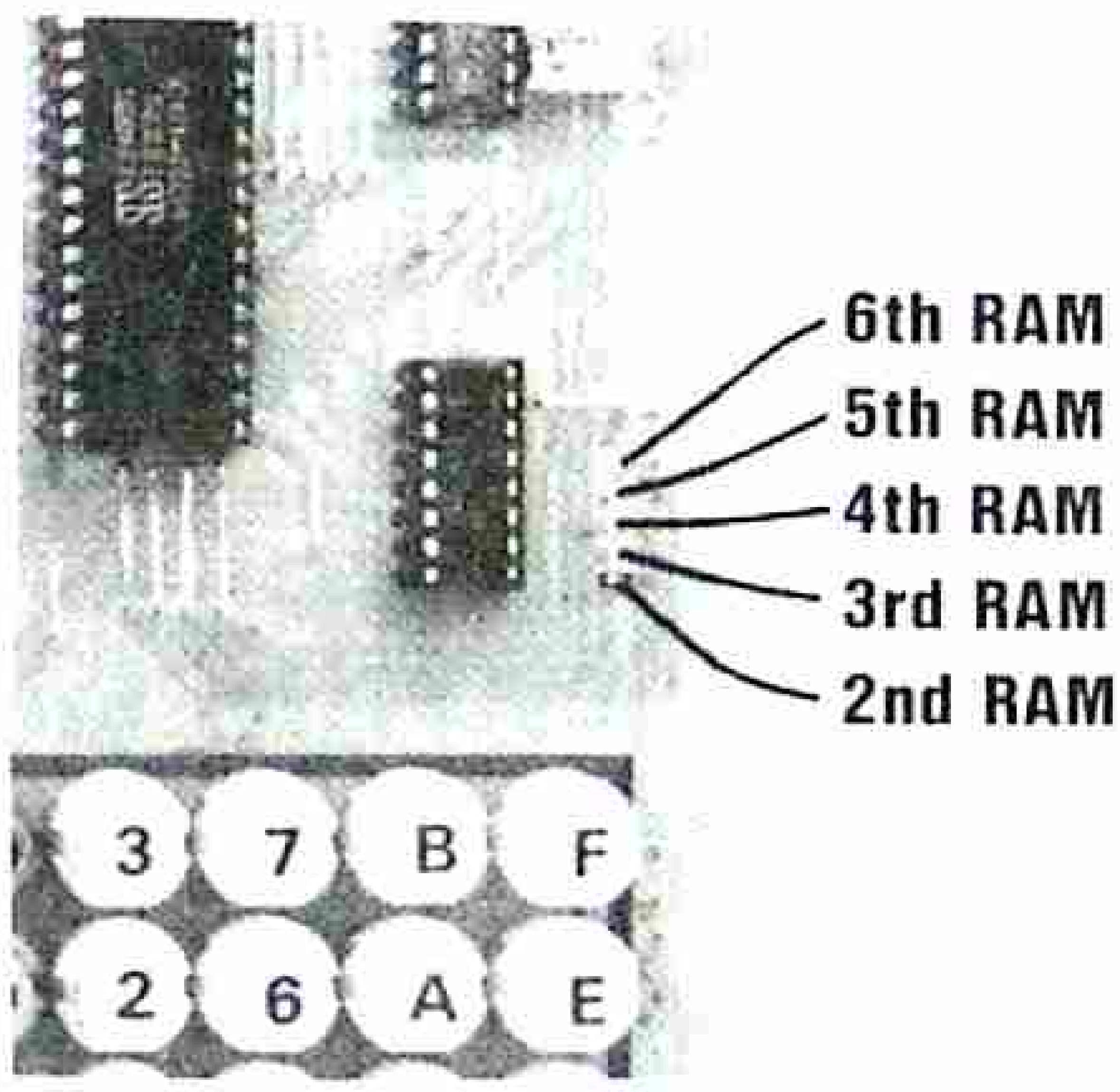
The close-up photos show how this pin is bent away from the rest so that it does not make contact with the lower IC socket.

Only the lowest RAM chip in the stack is selected by the track under the PC board. All others are connected via jumper leads, directly to the relevant output of the 74LS138 mentioned before.

Without any additional decoding, we can add a stack of 6 chips to the EXPANSION PORT SOCKET making a total of 14k for the TEC.

The lowest chip will have address values starting at 1000H to 17FFH. The others will have values as shown in the diagram below.

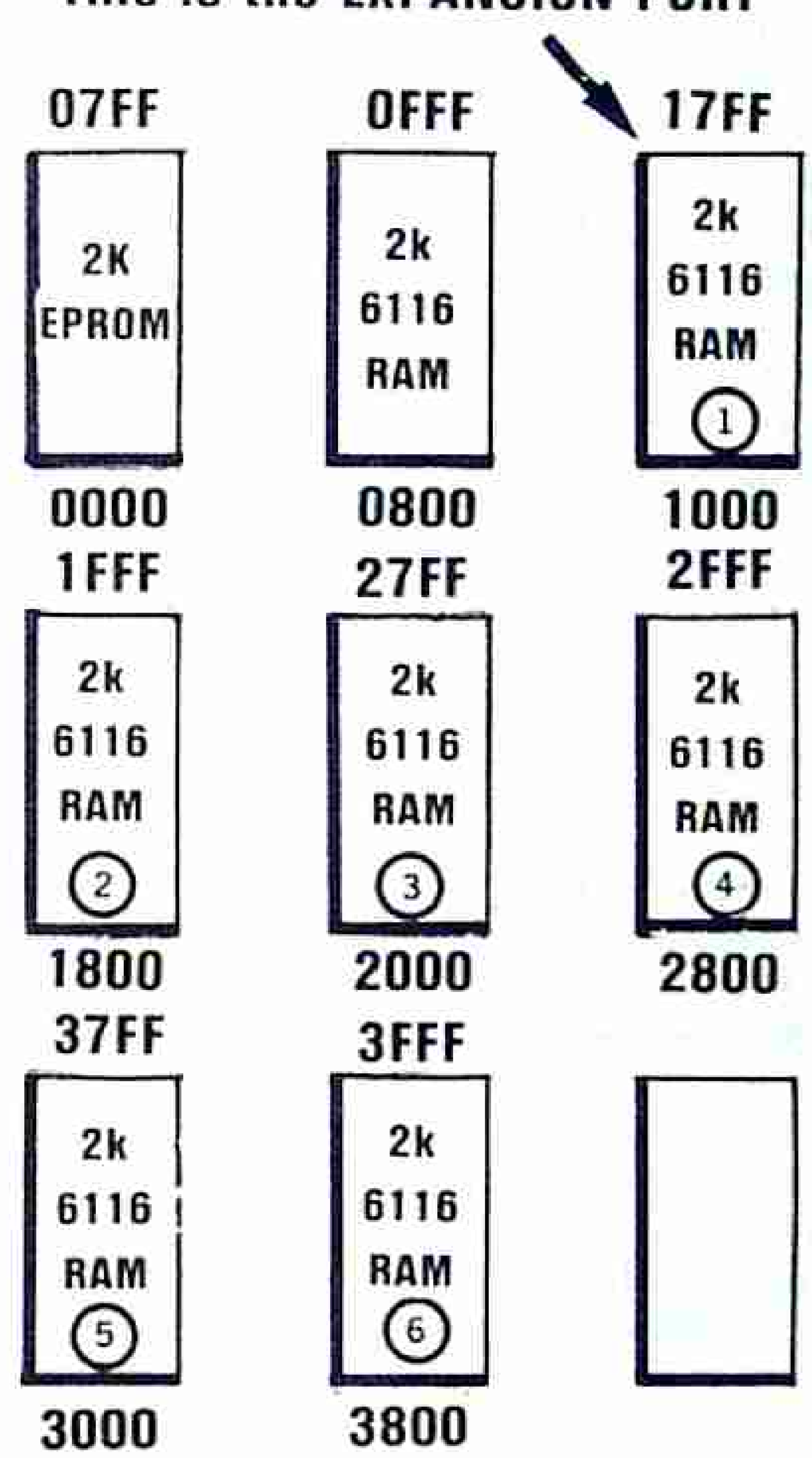




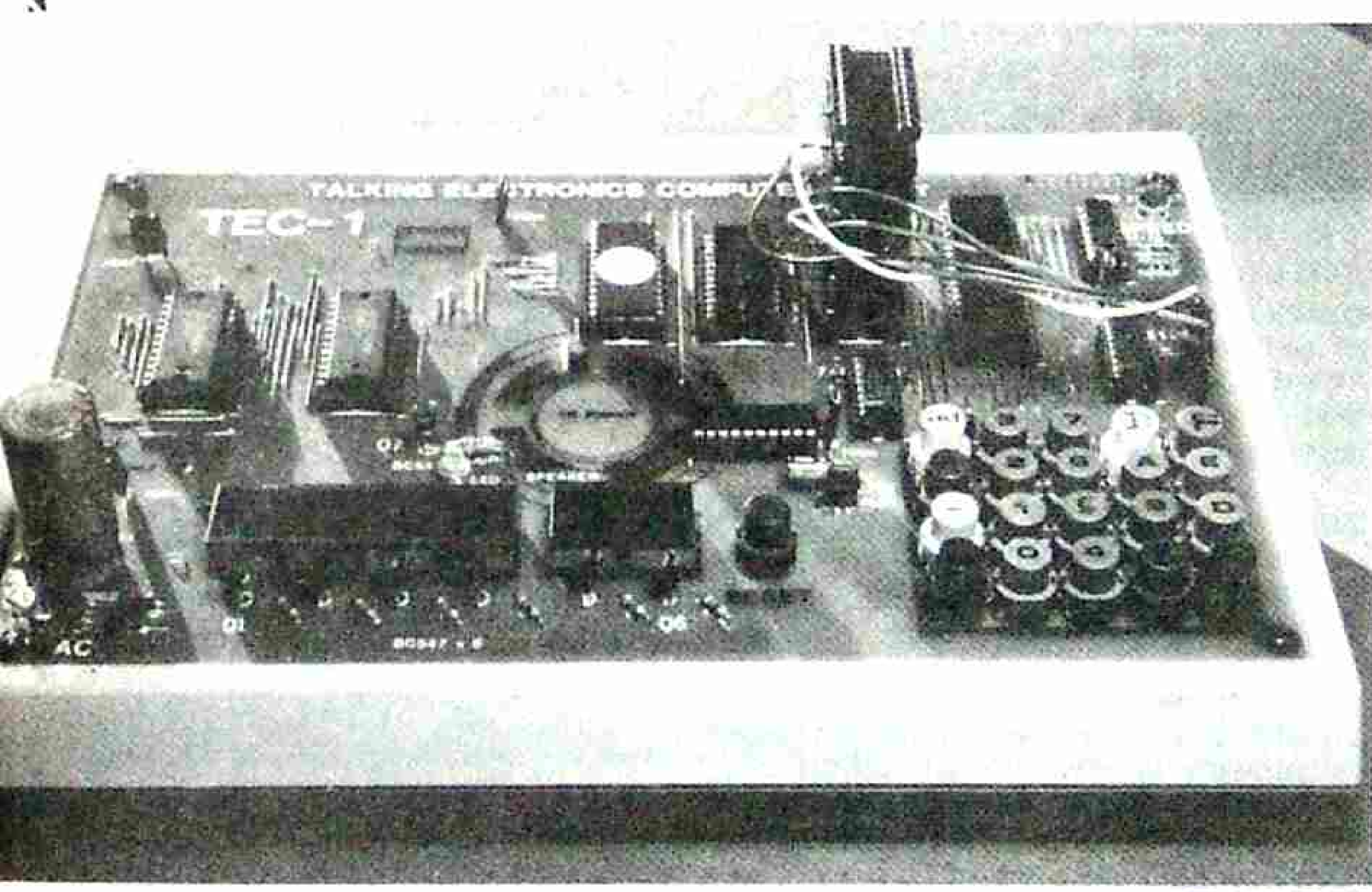
The first chip in the stack is enabled via the Expansion Port socket wiring. The other chips are enabled by connecting a jumper lead from pin 18 to one of the pins as shown above.

Of course you can **ENABLE** the chips 'out-of-order', by mixing up the jumper leads. This may fill the bottom chip, then the top chip, then number 3, then the fifth etc. No damage will result, it's just not a systematic way to do it.

This is the EXPANSION PORT

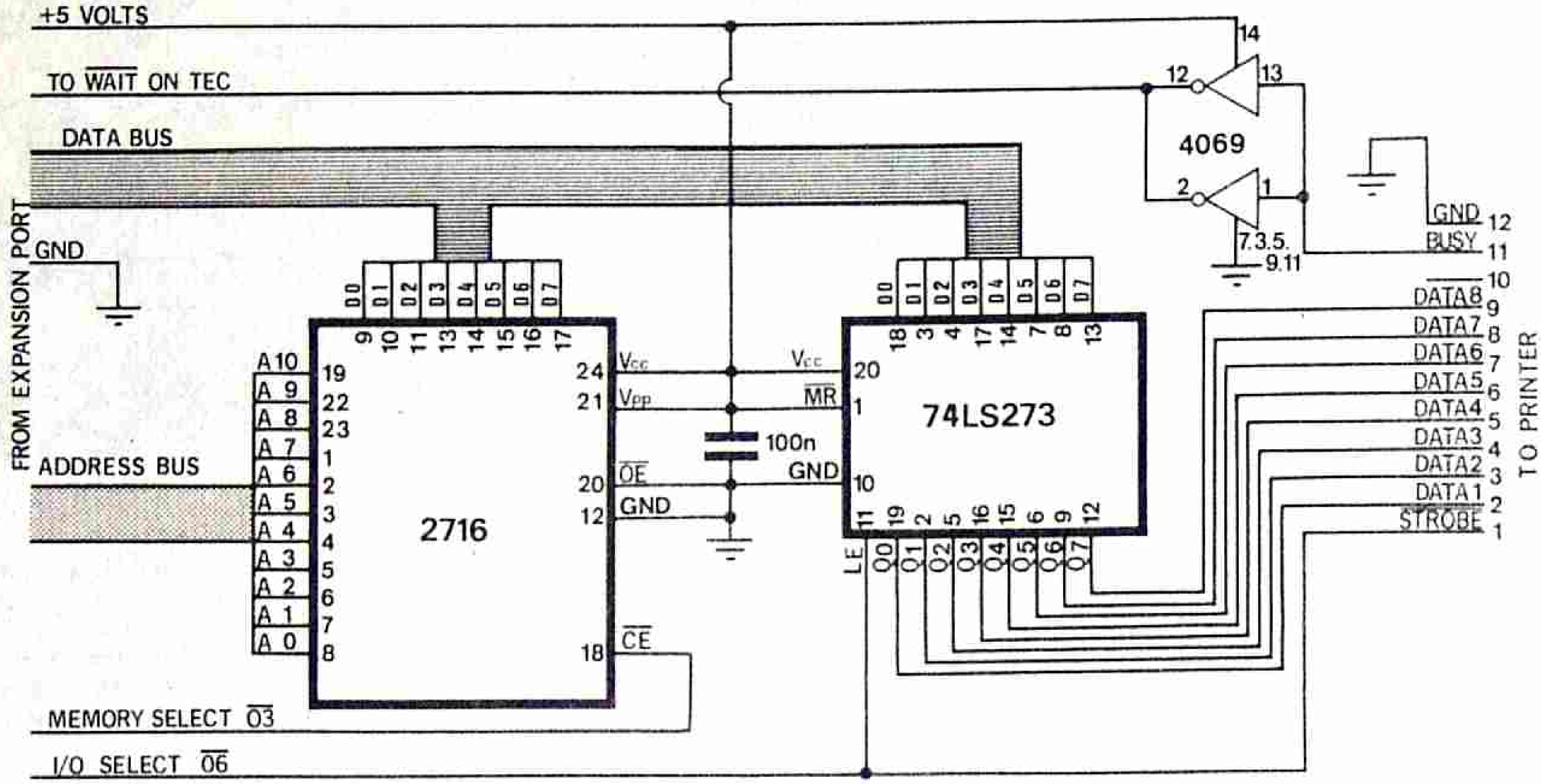


The start and finish address for the first 6 RAM chips.



Connecting the TEC to a: PRINTER/PLOTTER

Parts: $24.80 PC board: $3.80



Printer/Plotter circuit

Buy a Dick Smith VZ 200 Printer-Plotter.

This project explains how to directly access (talk to) a PRINTER/ PLOTTER. We have used the most readily available printer/plotter as it is not only the cheapest, but can be obtained from a number of suppliers.

Talking to one of these clever little performers is not very involved when you know how. But without the correct information it will remain completely DEAD. When you know how to supply it with the right stuff, it will do practically anything bar talk to you.

Actually you only have to send it the necessary codes to produce the character, all the creation of the shape of the symbol is done by the chip within the printer.

Not only do these printer/plotters accept instructions to produce numbers letters and symbols, but they can also be told to rotate, plot, vary the size of the characters and move in almost any direction.

There is a two-way interaction between printer and computer. Data is sent to the printer faster than it can be executed and to save holding up the computer, it is deposited in a FIFO register in blocks of about 4 bytes (in our case). Larger computers can be instructed to go away and execute other work while the FIFO register empties.

Bursts of data are transmitted like this until the whole program is executed.

As we have used a standard printer, it is obvious that it has been designed to connect to any computer which has a normal, full-size, key-board so that each key will produce the corresponding letter on the paper.

But this luxury is not absolutely necessary as the computer merely produces a code number which is sent to the printer.

The code number (or value) is called an ASCII number or ASCII CODE and fortunately is identical for all types and models of personal computers.

The secret to getting the printer to work on the TEC is the latch chip. It holds the data long enough for the printer to read it.

PARTS LIST

1 - 100n mono block

1 - 4049
1 - 74LS273
1 - 2716 (programmed)

1 - 14 pin IC socket
1 - 20 pin IC socket
1 - 24 pin IC socket
1 - 24 pin wire-wrap socket
1 - 24 pin DIP HEADER

1 - 36 pin Centronics type plug.

tinned copper wire
hook-up flex
3 - 'quick connect' pins and sockets

PRINTER INTERFACE PC BOARD

This means all we have to do is produce the same set of ASCII numbers (or codes) and the printer will produce the correct set of shapes on the paper.

Thus we don't need a full-size computer at all.

It may be a bit slow pressing the keys on the TEC, but all the printing capabilities will be possible, and that's all we want.

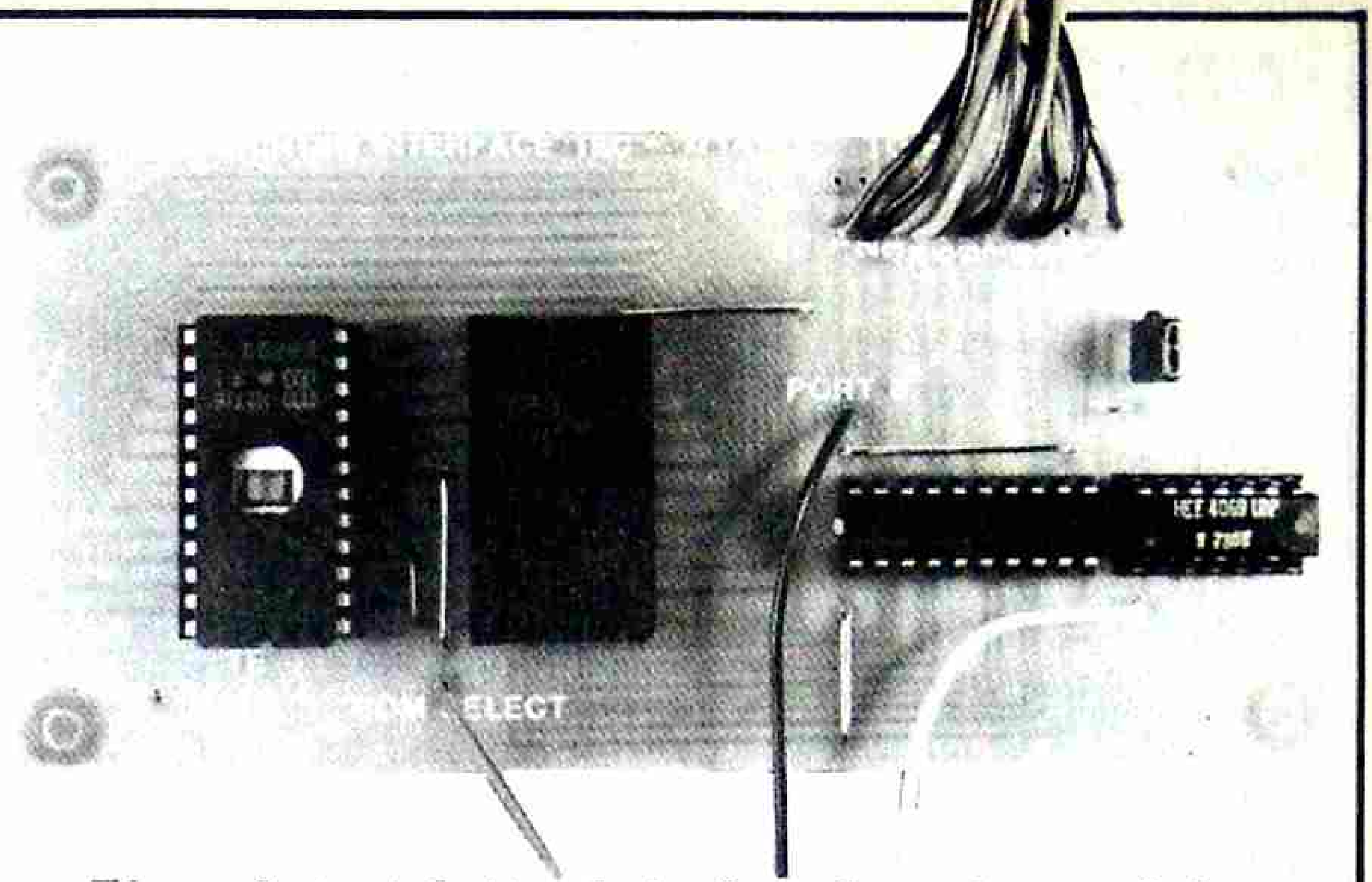
In this series of articles, we will explore the functions of the printer/plotter and create some amazing effects.

The most important aspect of this is realizing you can create a CONTROL PROGRAM with machine code listings and thus fill the minimum amount of memory for any given effect.

In this way you can produce you own system and expand it as much as you like without having to resort to buying a ready-made console. This will produce a cheaper and more compact system and will gain you much more respect from your boss or customer.

The first part of this project requires assembly of the printer interface board. This board contains a latch and EPROM (filled with a number of handy programs). This will give you a run-up program to test the interface board and provide instant transfer of data from computer to paper to reduce the amount of button-pushing.

The other chip on the board provides an inverted WAIT signal to halt the Z80. This basically keeps the two units in synchronisation.



The printer/plotter interface board complete and ready for plugging into the TEC & printer.

Set out all the parts on your bench and check everything. Solder the sockets, cap and 6 jumper links to the board. Mount the wire-wrap socket through the board so that the long pins act as 'stand-offs' for the component header plug. See the RELAY DRIVER BOARD article and photos for details of how this is done.

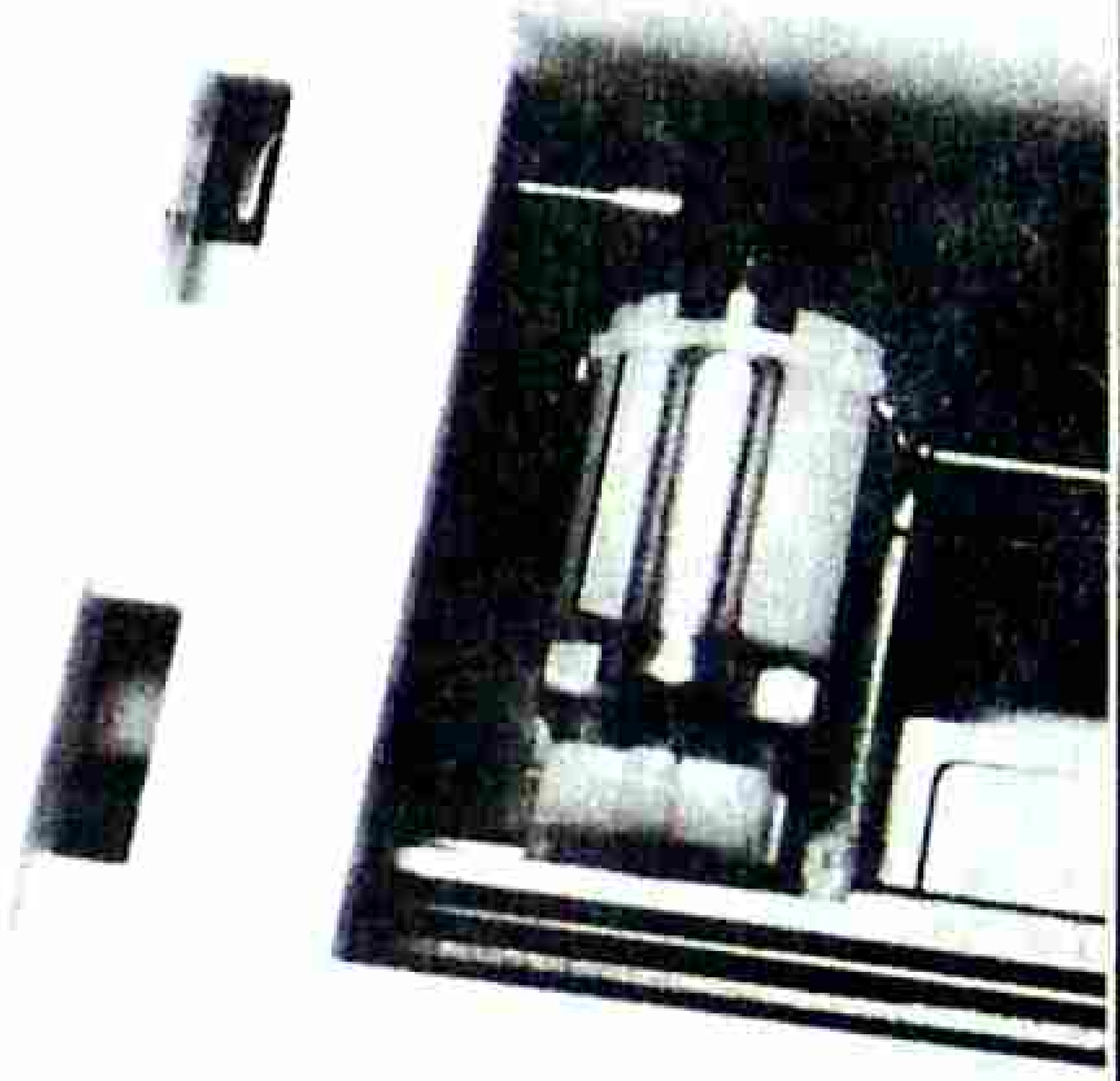
The final task involves connecting the board to the 36 pin Centronics plug.

WIRING THE PLUG

Wiring the Centronics-types plug to the printer interface is very easy. On the printer interface PC board there are 24 holes. Twelve of these are numbered. These numbers correspond to the numbers on the Centronics plug. Solder a length of hook-up wire between each hole and a corresponding hole on the connector plug. Pin 10 is not used, so no lead is needed. It is not necessary to use special connecting flex such as twisted pairs or screened lead. Our prototype worked perfectly with ordinary hook-up flex. It's best to use different coloured flex for each line to make tracing easier. These leads can be about 50 cm long and kept together with ties or tape at regular intervals.

We used a VZ 200 printer/plotter but there are other units with the same internal workings on the market. But they may not have the same input instruction set.

The pens use water-based ink and tend to dry-out fairly easily. If they fail to start: open them up, add a drop of water, heat them up and fit them back into the printer.



A close-up of the 4-pen print head.

The only remaining wires left are the 3 control lines. These are:

Memory select 03,
I/O select 06, and
WAIT.

These are fitted with 'quick connect' terminals which push onto matrix pins on the main PC board. Heat-shrink tubing can be placed over the terminals to strengthen the solder joint and make them easier to handle.

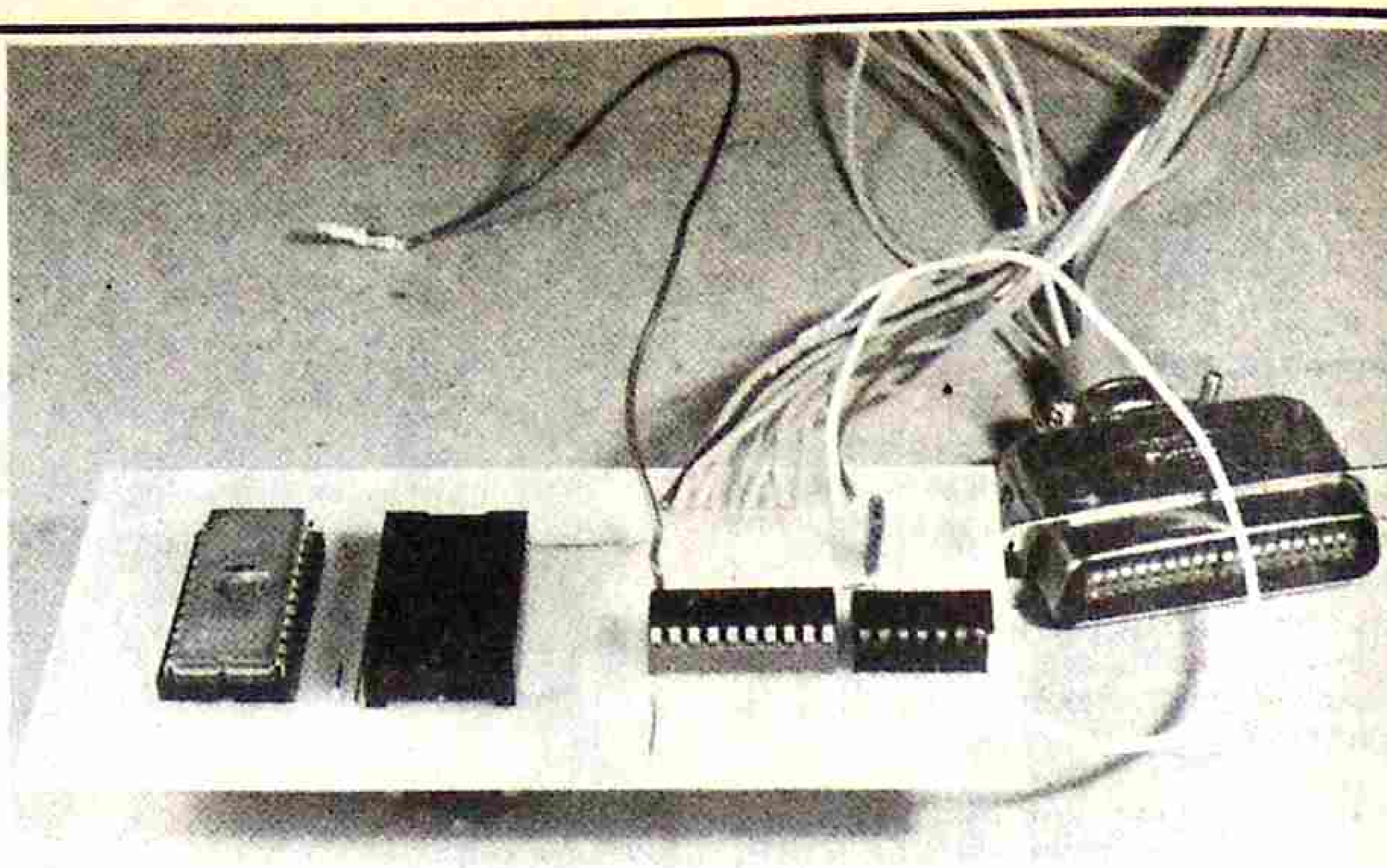
When the printer is first turned on it runs through an initial program (from its internal memory) which feeds the paper, sets the pen colours and starts the ink flowing by producing a box with each pen.

After this, there is very little else you can do via the buttons on the unit, except forward feed, change the colour of the pens and/or remove them.

All the rest of the action must come in the form of data from an outside source.

This is why we need the TEC. It supplies data at high speed to get the print-head moving.

Connect the centronics plug into the rear of the printer and fit the PRINTER INTERFACE PC BOARD to the computer. Connect the 3 flying leads as shown in the diagram:



All the parts shown are included in the kit.

To get something interesting out of the printer you will need to send it a program. The first of these is:

KEN's START-UP PROGRAM:

Make sure the print-head is to the left of the printer as when the printer has been switched on.

Push **ADdress 18A0 GO GO.**

Watch the result.

This type of program is beyond us at the moment but you will be capable of similar effects after reading this article.

For now, the next step is to be able to get letters and characters onto the paper.

PRODUCING LETTERS etc. . .

All information is fed to the printer in ASCII code. If you want a particular character, the correct code must be sent to the printer. Even if you want to send a number to the printer, such as 150, you must send it in the form of ASCII. This means 150 translates to 31 35 30, as you will see later from the table.

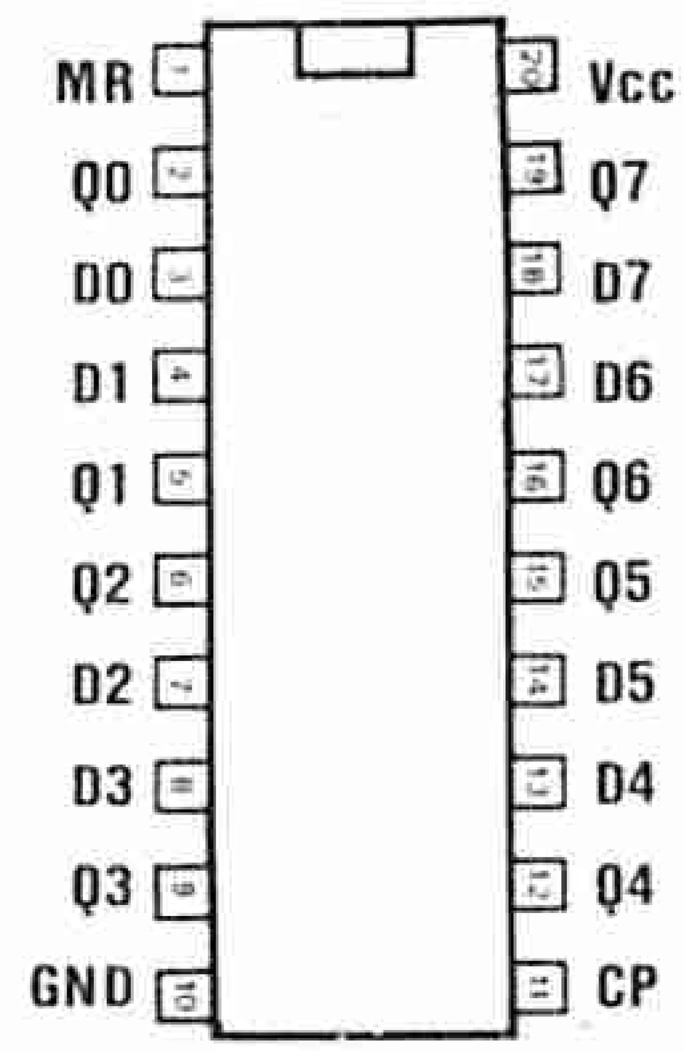
A small program is required to interpret your button pushing and send it to an output port. This is similar to making a segment on the display illuminate and the program for this is contained in the PRINTER/PLOTTER EPROM at 1980.

To use this program:

Press **ADdress 1980 GO GO**

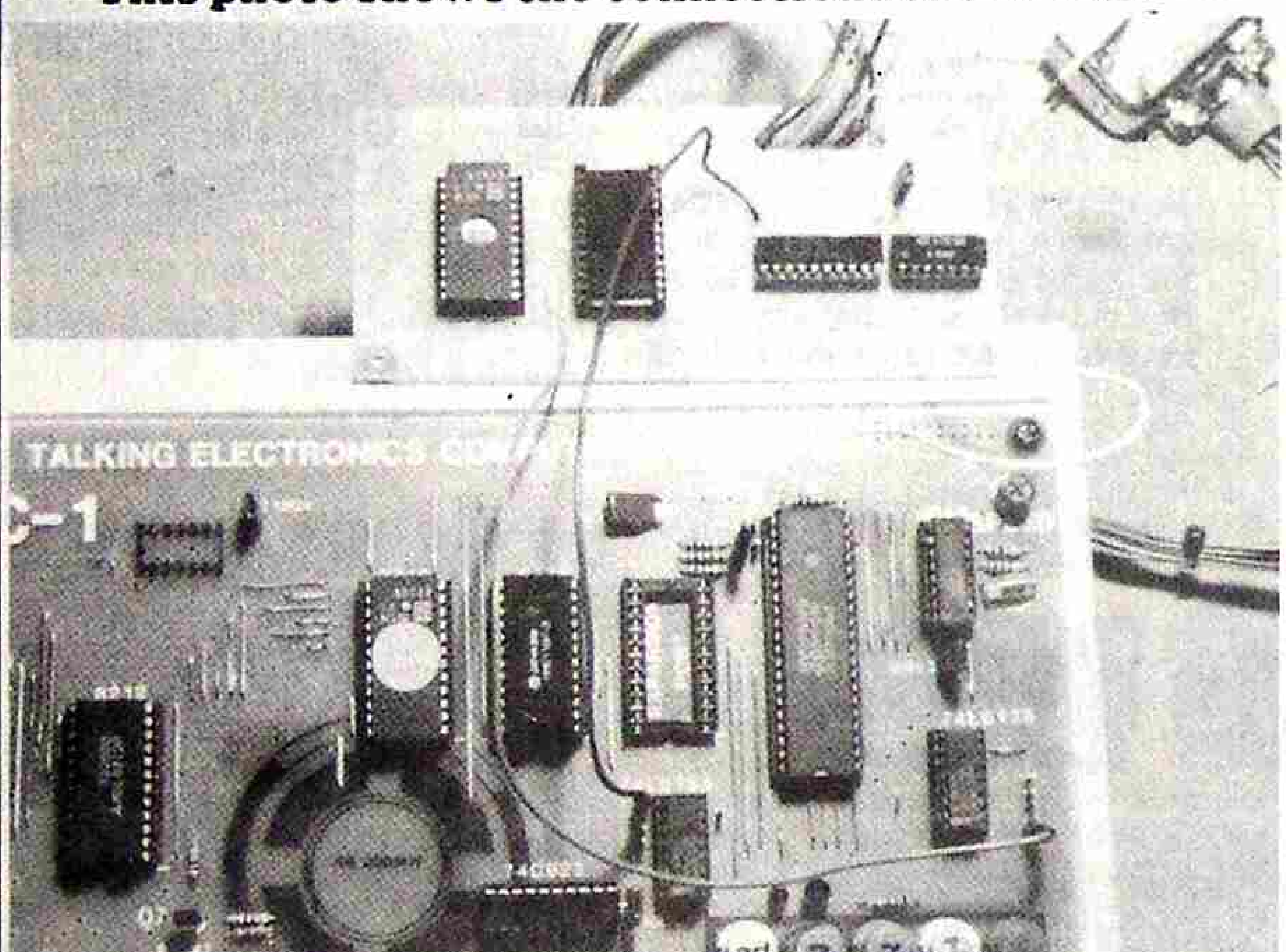
The display will go blank and the TEC will be ready for conveying your keyboard instructions directly to the printer.

74LS 273



Pin-out for 74LS273

This photo shows the connections to the TEC.



Each of the letters, numbers and symbols is shown in the table below and the corresponding hex value must be used for the symbol to appear on the paper.

Try obtaining all the letters, numbers and characters by following through the table.

PRINTER/PLOTTER ASCII VALUES:

SPACE	20	0	30	@	40	P	50	`	60	p	70		
BACK SPACE	08	!	21	1	31	A	41	Q	51	a	61	q	71
LINE FEED	0A	"	22	2	32	B	42	R	52	b	62	r	72
CR	0D	#	23	3	33	C	43	S	53	c	63	s	73
DC1	11	$	24	4	34	D	44	T	54	d	64	t	74
DC2	12	%	25	5	35	E	45	U	55	e	65	u	75
NEW COLOUR	1D	&	26	6	36	F	46	V	56	f	66	v	76
		'	27	7	37	G	47	W	57	g	67	w	77
		(	28	8	38	H	48	X	58	h	68	x	78
		)	29	9	39	I	49	Y	59	i	69	y	79
		*	2A	:	3A	J	4A	Z	5A	j	6A	z	7A
LINE BEFORE	0B	+	2B	;	3B	K	4B	[	5B	k	6B	{	7B
		,	2C	<	3C	L	4C	\	5C	l	6C	¦	7C
		-	2D	=	3D	M	4D	]	5D	m	6D	}	7D
		.	2E	>	3E	N	4E	^	5E	n	6E	~	7E
		/	2F	?	3F	O	4F	_	5F	o	6F	⊠	7F

Try the following sequence and you will see a word appear:
49 4E 43 52 45 44 49 42 4C 45

For the hex value **49**, the letter I will be printed. Press each number only ONCE. The first press will appear to have no effect, but as soon as the second button is pressed, the letter I will be printed.

Be very careful not to press button-sequence **11** or **12** as this will cause the mode to change and everything will appear to 'lock-out'.

Try writing a sentence using the hex key pad. It's slow but eventually gets you there. A space between words is created by typing **20.**

Any sentence you send to the printer via the keyboard can be re-presented again and again if placed into memory before-hand. It can also be corrected and adjusted (within limits). To do this, place the data at **0800** and call a program at **1880.**

Insert the following at **0800:**

49 4E 43 52 45 44 49 42 4C 45 20 20 48 55 4C 4B 0D 0A 1D FF.

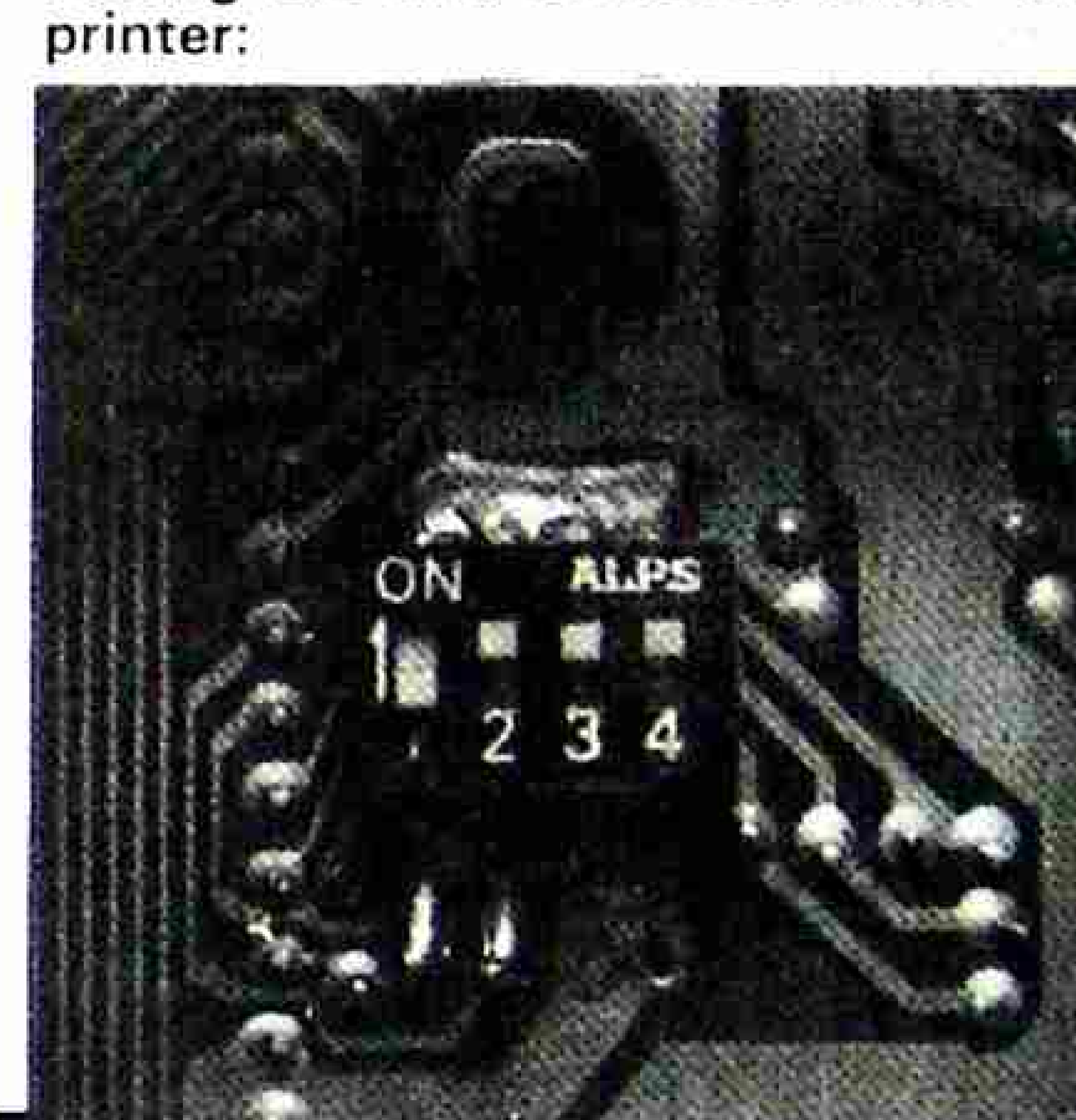
Push **ADdress 1880 GO GO.**

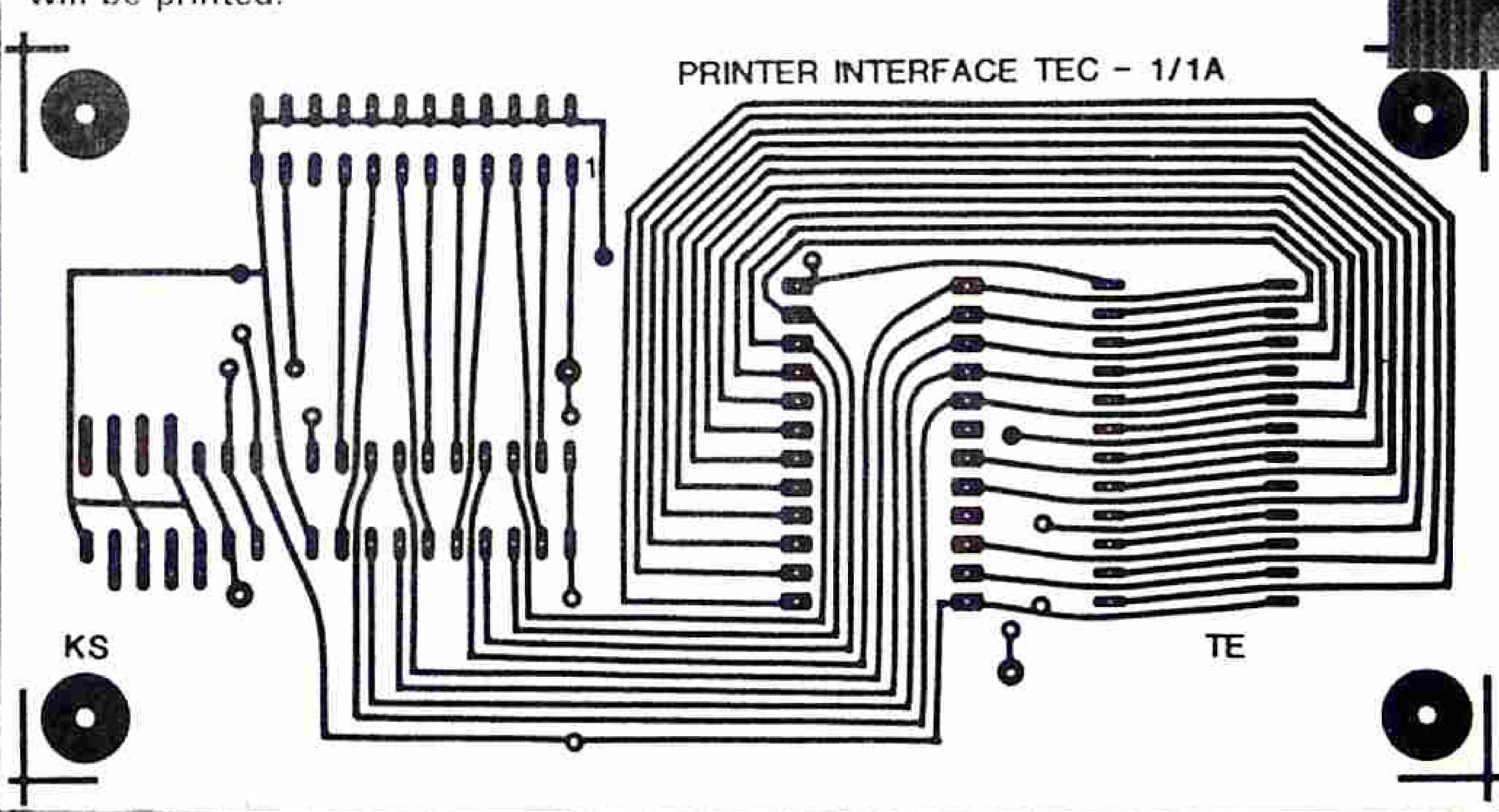
Recall it again by pressing:
ADdress 1880 GO GO.

THE LIST PROGRAM

This program lists any part of the EPROM, RAM or any additional memory you add to your TEC. In fact the first thing you can do is get a print-out of your MONitor ROM. Many readers have written requesting a listing of the MONitor and now they can produce it themselves.

But before you can get a listing, you must make a modification to the operation to the printer. This involves setting the two switches under the printer:





The PC layout for the Printer/Plotter. The overlay and parts positioning can be gained from the photo on P 31.

This is how to do it.

On the bottom of the printer is a small plate. Undo the screws and remove the plate. Inside you will find a bank of 4 switches. Switch 1 should be in the OFF position and switch 2 in the ON position. Don't worry about switch 3 and 4.

When the switches are set like this, CR (carriage Return) will set the print-head to the left of the paper without feeding the paper forward. The paper can then be fed forward by using LF (Line Feed). The switches should be set like this because the program in ROM automatically line feeds after each carriage return. If the switches are not set like this, the typing will be double line spaced.

Enter the following into the TEC:
ADdress **1800 GO GO:**

The display will go blank and the printer will CR and LF. Now enter **0000** and the printer will start printing out characters in pairs. This is a listing of the contents of your monitor ROM.

If you want a listing of any of the programs you have typed into memory, start at **0800** or where your program starts, and enter a 4-digit number into the keyboard. It must be 4 digits, so don't forget the leading 0.

The text mode is not very interesting. After all, we have seen electric/electronic typewriters for years, But for a print-head to produce GRAPHICS! That's different!

GRAPHICS MODE

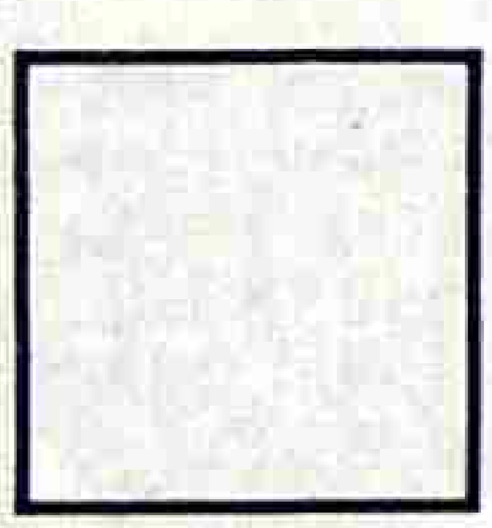
The program at **1880** can also be used to generate graphics on the printer.

Remember, all information must be programmed into the printer in ASCII.

Type the program below into the TEC's memory at **0800.** An **FF** is placed after the last piece of data to signify the end of a program. Now run the program at **1880** by pressing **AD**dress **1880 GO GO.**

at 0800: 0A 0D 12 49 2C 44 38 30 2C 30 2C 38 30 2C 2D 38 30 2C 30 2C 2D 38 30 2C 30 2C 30 0D FF.

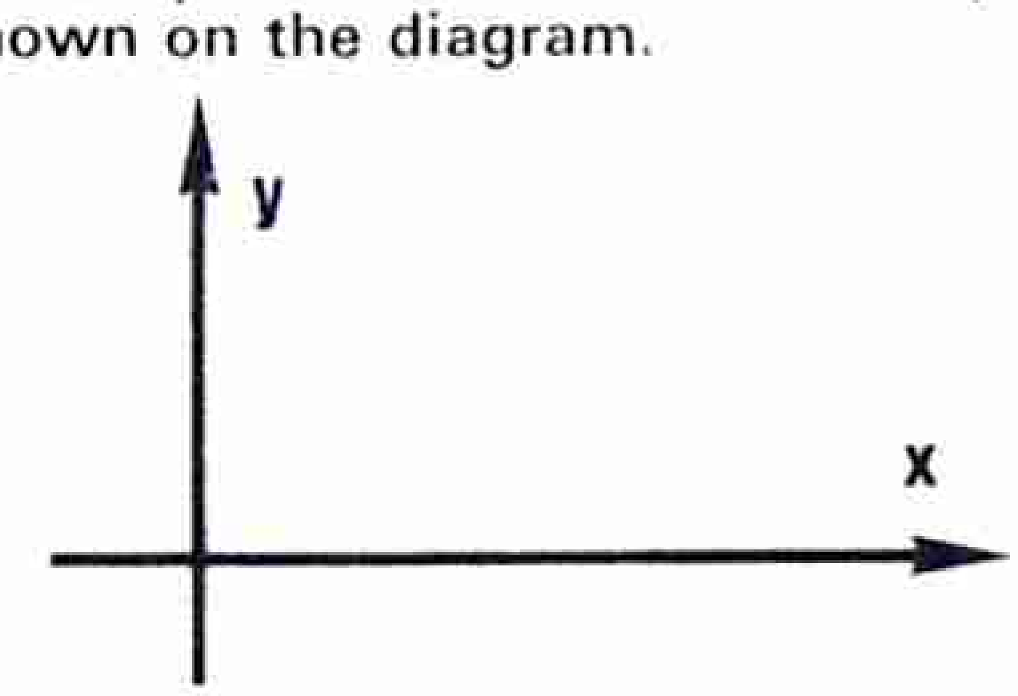
The printer will draw a square.

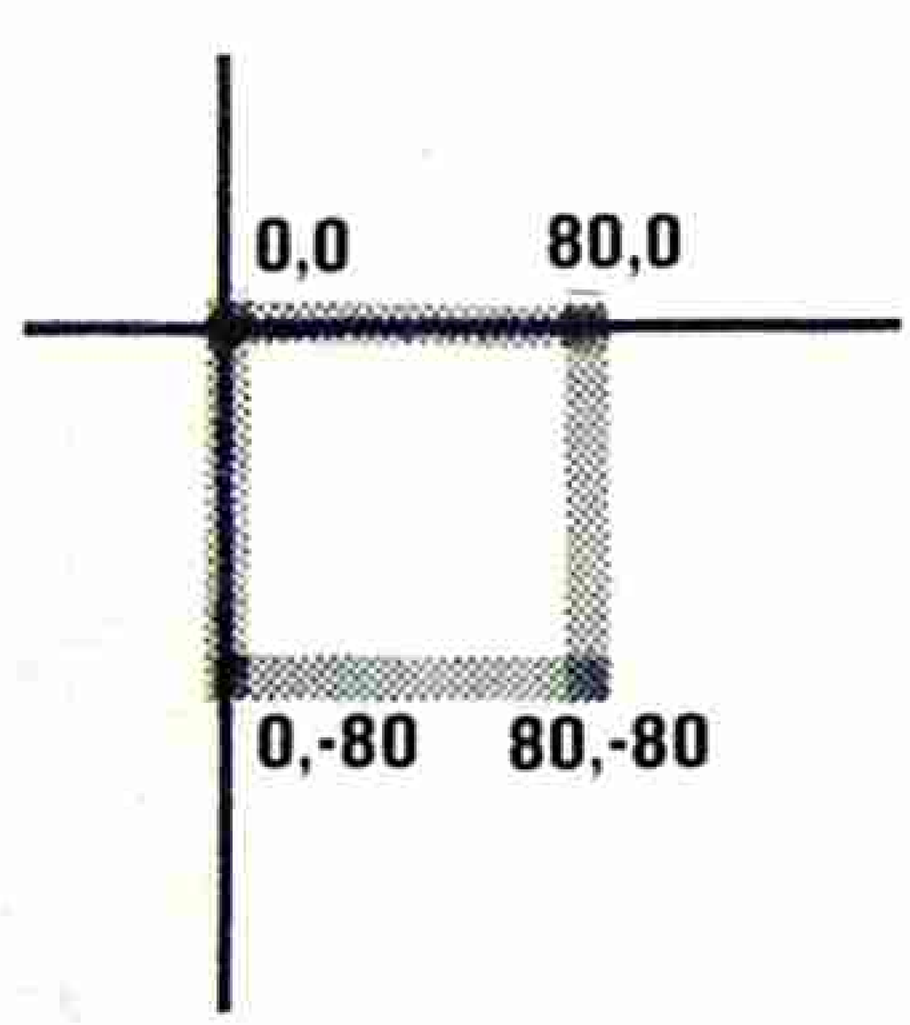


Look at the listing. It may look complex but can be easily decoded using the table. It will decode to this:

0A = LF = Line Feed.
0D = CR = Carriage Return
12 = DC2 = Graphic Mode
49 = I = sets the pen's location as co-ordinates 0,0.
2C = , =Separates I from D
44 = D = draw from present location to the co-ordinate given by the next byte(s) of data.
38 = 8
30 = 0
2C = ,
30 = 0
2C = ,
38 = 8
30 = 0
2C = ,
2D = -
38 = 8
30 = 0
2C = ,
30 = 0
2C = ,
2D = -
38 = 8
30 = 0
2C = ,
30 = 0
2C = ,
30 = 0
0D = CR = carriage return
FF = signifies end of program.

The printer uses a co-ordinate system exactly like the x,y axis used to draw graphs. The origin is 0,0 (or 00,00) and the positive direction of x and y is shown on the diagram.





The co-ordinates of the corners of the box are shown in this diagram. This clearly shows how the values are obtained.

The program can be separated into 4 sections, each drawing one side of the box. This will show how the program goes together.

The following program produces the top of the square:

at 0800 type: **0A 0D 12 49 2C 44 38 30 0D FF.**

ADdress **1880 GO GO.**

The result will be:

Let us produce a line the full width of the paper. For this you will need a 3-digit value. The printer is capable of accepting a value as high as 999 (also -999) but this will be too high for our width of paper. Try 300.

The ASCII value is 33 30 30.

at **0800:**
0A 0D 12 49 2C 44 33 30 30 2C 30 0D FF.
Press **AD**dress **1880 GO GO.**

The final **0D** is important to get the printer to execute the graphics command.

The value **300** will not quite reach the far side of the paper. Try **450.** This will be about the longest line possible and don't forget to use the ASCII values in the program.

Shorten the side of the box to 80 and continue with the experiment.

The second side of the box will be produced at an angle other than 90° by inserting the following co-ordinates: 50, -80

at **0800:**
0A 0D 12 49 2C 44 38 30 2C 30 2C 35 30 2C 2D 38 30 0D FF.

Run the program. Does it produce two sides of an irregular figure?

The next side will be produced as follows:

0A 0D 12 49 2C 44 38 30 2C 30 2C 35 30 2C 2D 38 30 2C 31 35 30 2C 2D 38 30 0D FF.

Run the program and see the result.

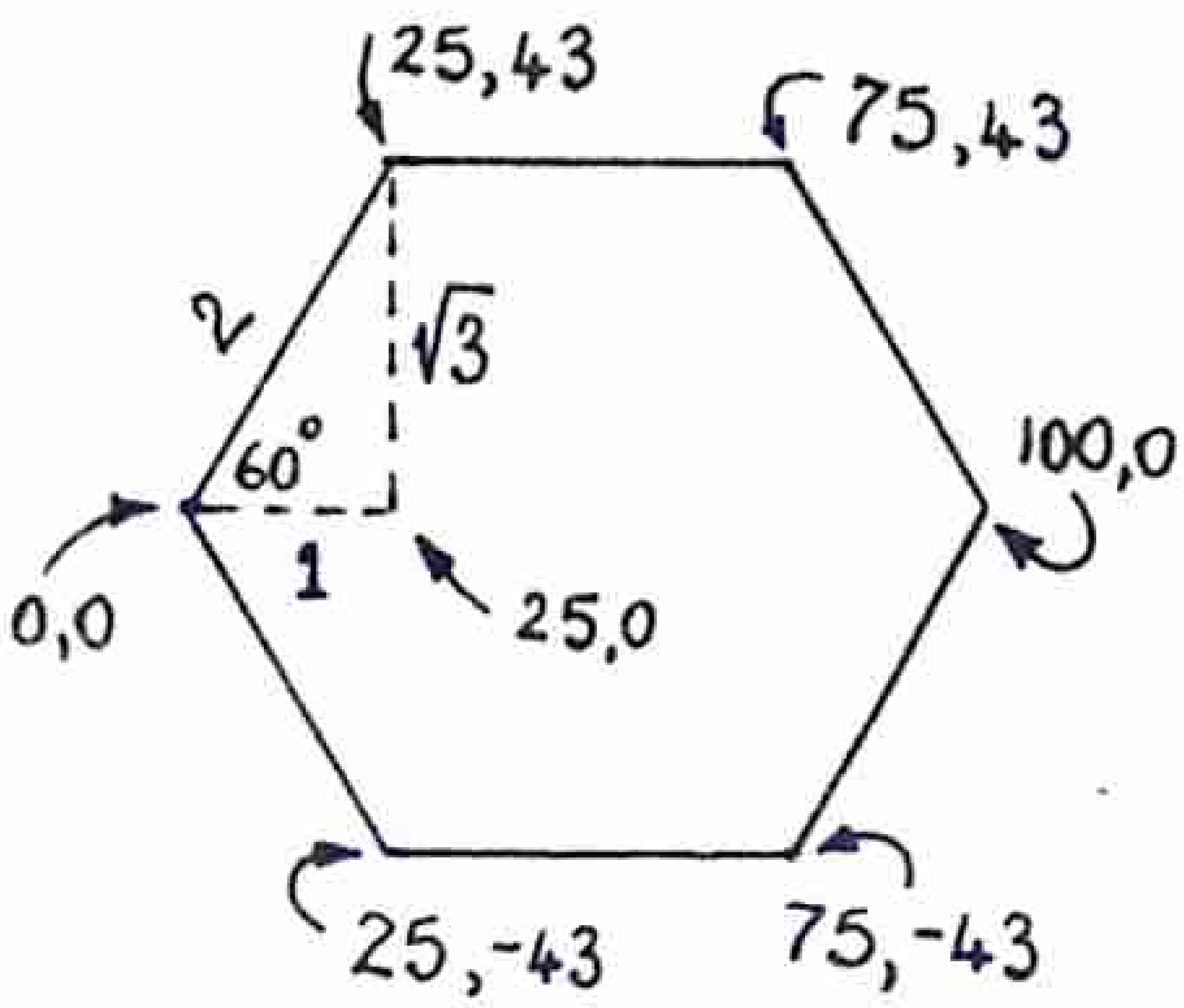
Finally:
0A 0D 12 49 2C 44 38 30 2C 30 2C 35 30 2C 2D 38 30 2C 31 35 30 2C 2D 38 30 2C 0A 0D 11 1D 0D FF.

Produce other shapes and you will understand how to plot co-ordinates.

HEX

The second shape we will investigate is a HEXAGON.

To produce this shape you need to know the value of the internal angle and produce a 30° 60° 90° triangle as shown. This will give you the length of the sides of the triangle and from this the first set of co-ordinates can be obtained (25,43) These values are 1/4 of 100, 173, which are the lengths of the sides of the triangle.

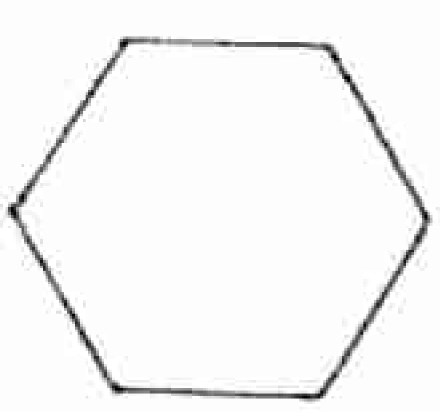


The second co-ordinate, 75,43 is found by adding 50 to the value 25. Continue around the hex shape until the figure is closed.

This is the listing for the printer:

at 0800:

12 49 0D 44 32 35 2C 34 33 2C 37 35 2C 34 33 2C 31 30 30 2C 30 2C 37 35 2C 2D 34 33 2C 32 35 2C 2D 34 33 2C 30 2C 30 0D FF.



0's and X's

The new instruction with this shape is the MOVE command. — 4D

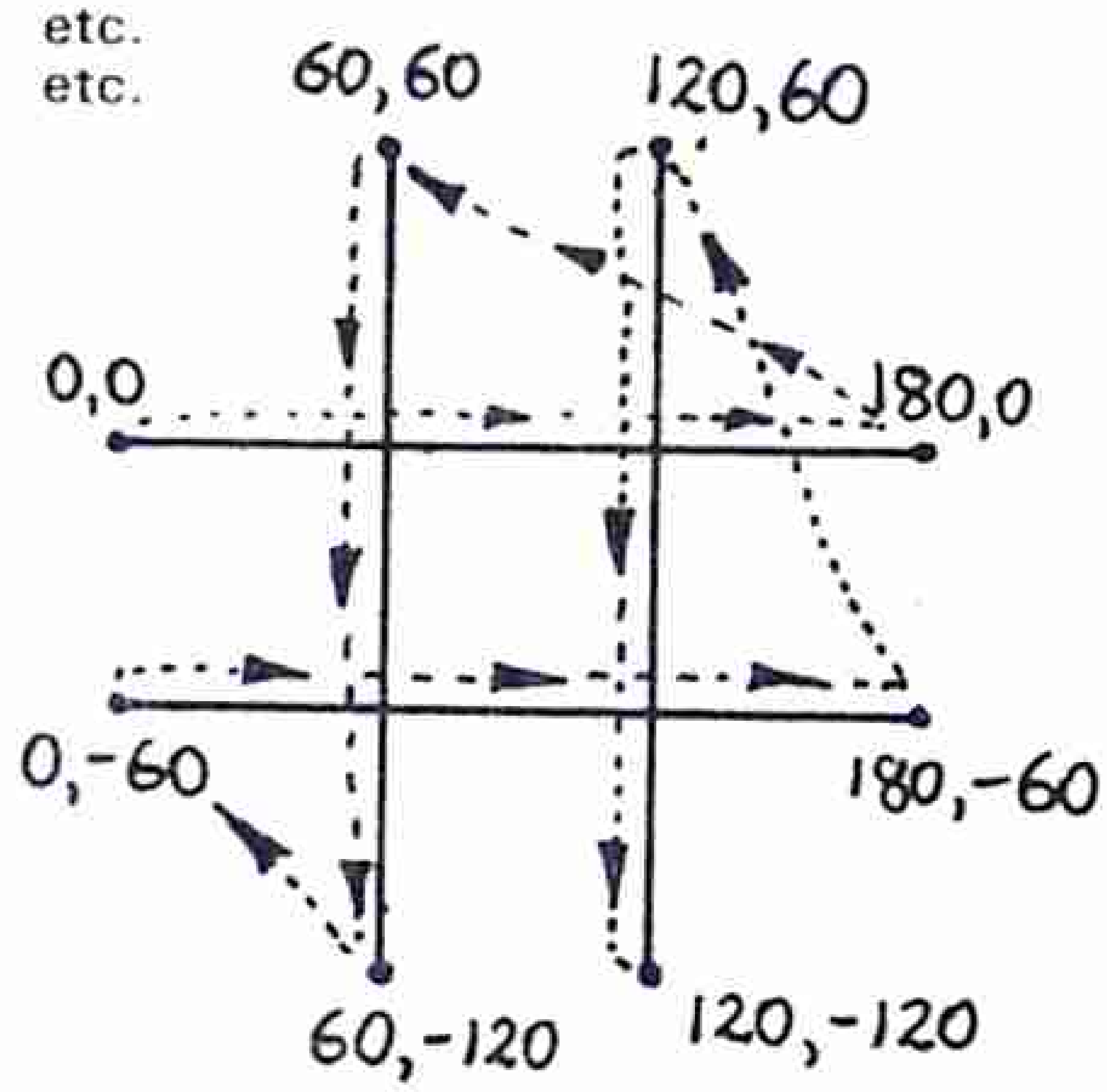
This instructs the pen to lift from the page and move to a specified location without drawing on the paper.

Here is the listing and the shape which will be drawn:

0800	**0A**	**0D**	**12**	**49**	**0D**	**44**	**31**	**38**
0808	**30**	**2C**	**30**	**0D**	**4D**	**36**	**30**	**2C**
0810	**36**	**30**	**0D**	**44**	**36**	**30**	**2C**	**2D**
0818	**31**	**32**	**30**	**0D**	**4D**	**30**	**2C**	**2D**
0820	**36**	**30**	**0D**	**44**	**31**	**38**	**30**	**2C**
0828	**2D**	**36**	**30**	**0D**	**4D**	**31**	**32**	**30**
0830	**2C**	**36**	**30**	**0D**	**44**	**31**	**32**	**30**
0838	**2C**	**2D**	**31**	**32**	**30**	**0D**	**FF**	

This is a decoding of the first part of the listing. This will be sufficient to understand how the program is written.

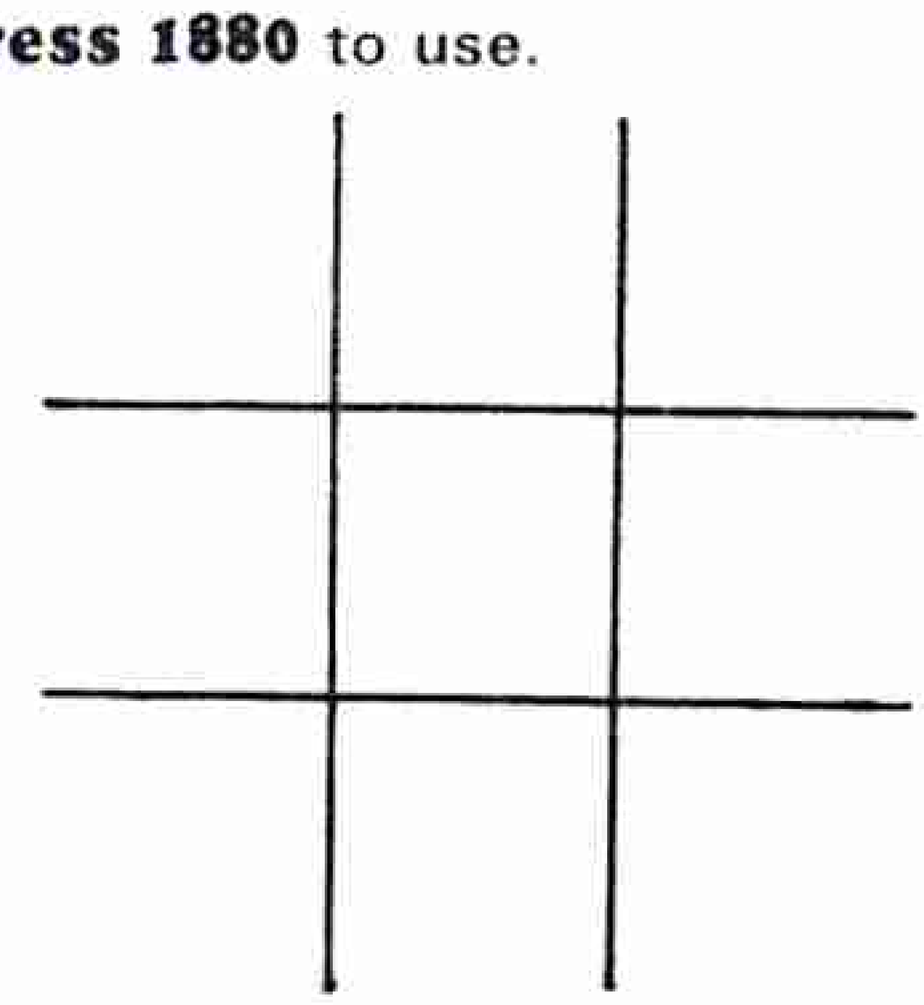
0A = LF =Line Feed
0D = CR =Carriage Return
12 = DC2 = Graphics Mode
49 = I = initialize the co-ords 0,0
0D = CR = signifies the end of the previous command. It does not cause the carriage to return but enables the previous command to be carried out.
44 = D = Draw
31 = 1
38 = 8
30 = 0
2C = ,
30 = 0
0D = CR = end of draw statement.
4D = M = Move. The pen is instructed to move without drawing.
36 = 6
30 = 0
2C = ,
36 = 6
30 = 0
0D = CR End of Move statement.
44 = D = Draw
36 = 6
30 = 0
etc.
etc.



This diagram shows the value of the co-ordinates required to draw the shape.

Copy out the complete listing and decode it to prove that the path taken by the print-head is as shown in the diagram.

ADdress 1880 to use.



WAR GAMER'S DELIGHT

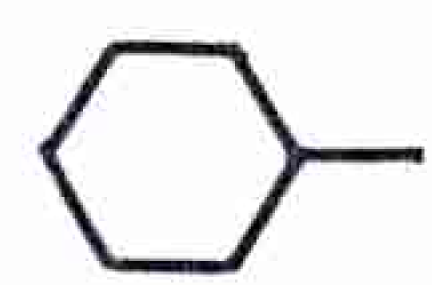
The full impact of this effect is shown on the next page.

The first thing you notice about the program is a set of values at the beginning which the printer does not recognise. This means they must be Machine Code values for an 'operations' program for the Z80. And they are.

The program produces a honey-comb pattern.

Anyone into war games will soon recognise the possibilities of the honey-comb as a playing board. The reason is each block has 6 borders, increasing the possible moves and thus the strategy, over a regular field of squares.

This shape is created using a picture element of a hexagon attached to a straight line thus:



This pattern is repeated 4 times across the paper and then a move to a new starting point -450,-86 down the paper.

The co-ordinates of the new starting point can be explained as follows:

After each picture element is drawn, the printer is initialized. This means that the present co-ordinate of the pen is taken as 00,00.

This gives us a value of 450,00 for the commencement of the 4th picture element with reference to the origin.

The next row of hexagons commence at the left-hand edge, which is -450 with reference to the above X co-ordinate and a y value of -86, with reference to the y value above.

The only way to understand how the honey-comb has been produced is to decode the listing. It contains two loops, one to draw the picture element and the other to count-to-4 across the screen.

Write each of the ASCII codes in a single file and alongside it place the printer value it represents.

You can experiment further by making the hexagons smaller. This will use a 2-digit ASCII value for the length of the sides. In the program, the original 3-digit ASCII values have been converted to 2-digit by using 00 for the 3rd value.

The first 18H bytes (31 bytes is the MAIN program and this contains the instructions to fetch one byte of data from the printer program and send it to the printer.

Data for the printer is stored in the form of a BYTE TABLE and starts at **0820.**

The main program is divided into two separate parts. **0800 - 080F** is a loop which loads the printer program and runs it 4 times.

811 - 81D loads the data for the MOVE commands and each piece of data is sent to the printer until **FF** is detected.

The count-to-4 operation is performed by **DJNZ** (at **80F**) which automatically decrements register B by ONE on each pass of the loop until it becomes zero.

The program then advances to loading HL register-pair with the contents of memory location **0860** and this instruct the print-head to move to the left-hand edge and down the paper to a new starting point. The main program then jumps to the start (**0800**) via instruction **JR Z E7** (at **817**).

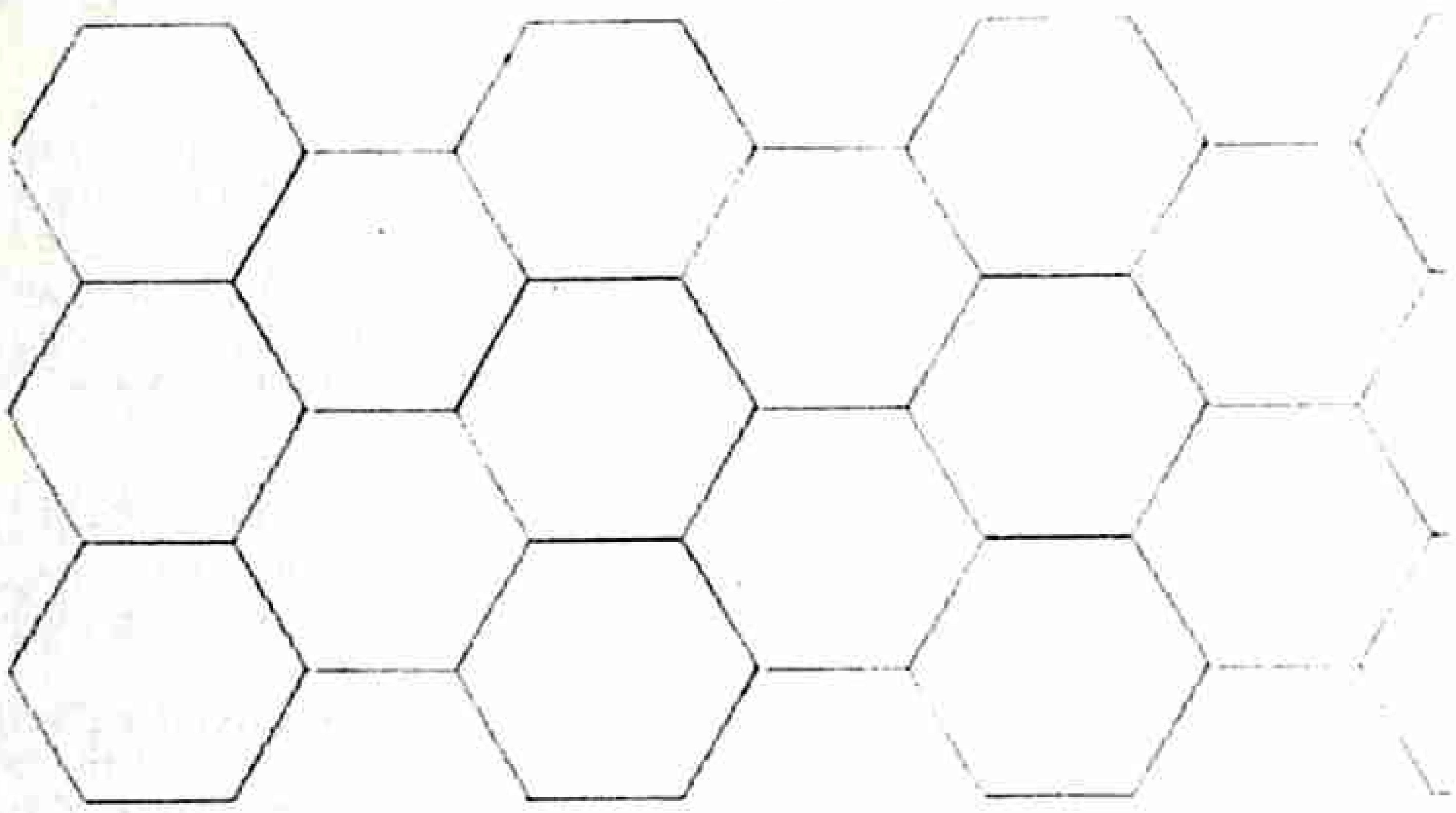
800	06	LD B,04
801	04	
802	21	LD HL,0820
803	20	
804	08	
805	7E	LD A,(HL)
806	FE	CP FF
807	FF	
808	28	JR Z 05
809	05	
80A	D3	OUT (06),A
80B	06	
80C	23	INC HL
80D	18	JR F6 (to 805)
80E	F6	
80F	10	DJNZ F1 (to 802)
810	F1	
811	21	LD HL 0860
812	60	
813	08	
814	7E	LD A,(HL)
815	FE	CP FF
816	FF	
817	28	JR Z E7 (to start)
818	E7	
819	D3	OUT (06),A
81A	06	
81B	23	INC HL
81C	18	JR F6 (to 814)
81D	F6	
81E	FF	NOT USED
81F	FF	NOT USED

DATA FOR PRINTER:

820	12	Graphics Mode
821	49	Initialize
822	0D	CR
823	44	Draw
824	32	2
825	35	5
826	2C	,
827	34	4
828	33	3
829	2C	,
82A	37	7
82B	35	5
82C	2C	,
82D	34	4
82E	33	3
82F	2C	,
830	31	1
831	30	0
832	30	0
833	2C	,
834	30	0
835	2C	,
836	37	7
837	35	5
838	2C	,
839	2D	-
83A	34	4
83B	33	3
83C	2C	,
83D	32	2
83E	35	5
83F	2C	,
840	2D	-
841	34	4
842	33	3
843	2C	,
844	30	0
845	2C	,
846	30	0
847	0D	CR
848	4D	Move
849	31	1
84A	30	0
84B	30	0
84C	2C	,
84D	30	0
84E	0D	CR
84F	44	Draw
850	31	1
851	35	5
852	30	0
853	2C	,
854	30	0
855	0D	CR
856	FF	End

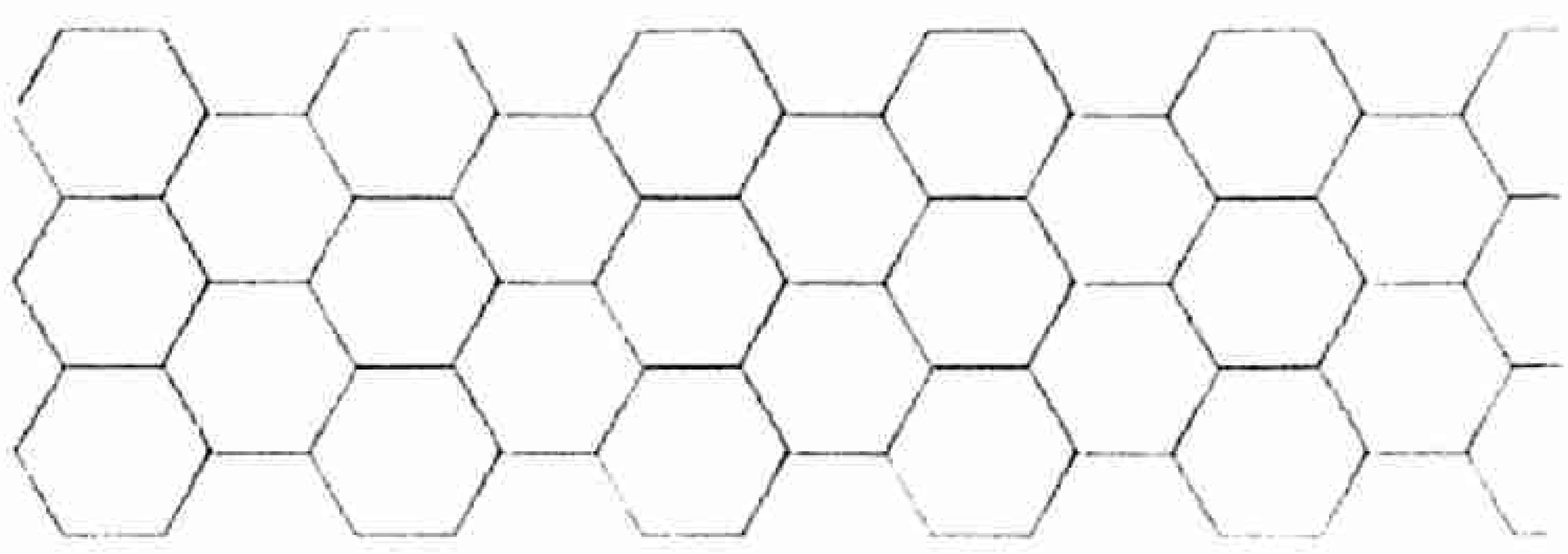
Data for MOVE COMMANDS:

860	4D	Move
861	2D	-
862	34	4
863	35	5
864	30	0
865	2C	,
866	2D	-
867	38	8
868	36	6
869	0D	CR
86A	FF	End

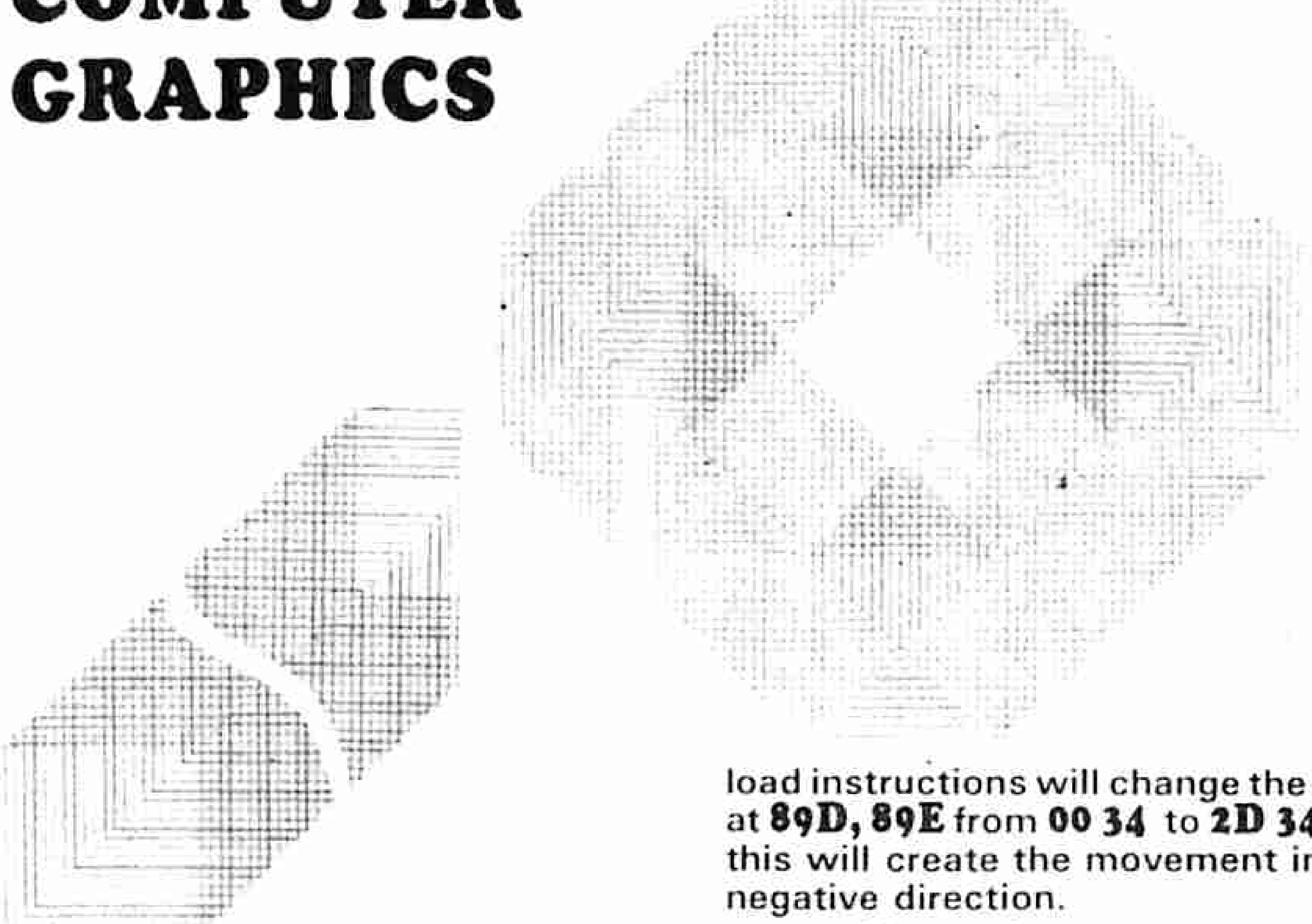


```
0800 06 04 21 20 08 7E FE FF
0808 28 05 D3 06 23 18 F6 10
0810 F1 21 60 08 7E FE FF 28
0818 E7 D3 06 23 18 F6 FF FF
0820 12 49 0D 44 32 35 2C 34
0828 33 2C 37 35 2C 34 33 2C
0830 31 30 30 2C 30 2C 37 35
0838 2C 2D 34 33 2C 32 35 2C
0840 2D 34 33 2C 30 2C 30 0D
0848 4D 31 30 30 2C 30 0D 44
0850 31 35 30 2C 30 0D FF FF
0858 FF FF FF FF FF FF FF FF
0860 4D 2D 34 35 30 2C 2D 38
0868 36 0D FF
```

```
0800 06 06 21 20 08 7E FE FF
0808 28 05 D3 06 23 18 F6 10
0810 F1 21 60 08 7E FE FF 28
0818 E7 D3 06 23 18 F6 FF FF
0820 12 49 0D 44 31 35 2C 32
0828 36 2C 34 35 2C 32 36 2C
0830 00 36 30 2C 30 2C 34 35
0838 2C 2D 32 36 2C 31 35 2C
0840 2D 32 36 2C 30 2C 30 0D
0848 4D 00 36 30 2C 30 0D 44
0850 39 30 00 2C 30 0D FF FF
0858 FF FF FF FF FF FF FF FF
0860 4D 2D 34 35 30 2C 2D 35
0868 32 0D FF
```



COMPUTER GRAPHICS



Being able to draw some of the basic shapes (as we have shown), opens up a whole new world of computer graphics.

If we take the box-shape, we can produce a very effective pattern simply by re-defining the start co-ordinates and repeating the shape many times. The result can be anything from a 'check-tie' to an irregular octogon.

The colourful patterns which can be obtained (of which we can only see the result in black and white) is produced by a combination of drawing, shifting and colour-changing. The first of these to be investigated will be an irregular octogon or DIAMOND.

We have already outlined the structure of the program and briefly it is a set of instructions which are loops. Each sets a particular condition and then decrements on each pass.

For the diamond shape, a square is generated at the origin, 00,00 via the program at **0880. The lengths of the sides of the square are 80.** When the 4 sides have been drawn, the pen lifts off the paper and moves to a new origin with co-ordinates 04,04. The program is now up to location **083C.** It then jumps to **0842.** The contents of the accumulator (which is the value at location **0860** i.e. **00**) is loaded into **089A.** Register pair DE is incremented and now looks at location **0861.** The value **34** is loaded into the accumulator. At **0847** the contents of the accumulator is loaded into location **089B.**

So far, the program at **0880** has not been altered but the next two sets of load instructions will change the data at **89D, 89E** from **00 34** to **2D 34** and this will create the movement in the negative direction.

800	06	06	3E	0A	D3	06	10	FC
808	3E	12	D3	06	11	60	08	06
810	08	C5	06	04	21	80	08	7E
818	FE	FF	CA	23	08	D3	06	23
820	C3	17	08	10	EF	3E	11	D3
828	06	3E	1D	D3	06	AF	D3	06
830	3E	12	D3	06	C1	10	DA	1A
838	FE	FF	C2	42	08	3E	11	D3
840	06	C7	32	9A	08	13	1A	32
848	9B	08	13	1A	32	9D	08	13
850	1A	32	9E	08	13	C3	0F	08
858	00	FF	00	FF	00	FF	00	FF
860	00	34	2D	34	2D	34	2D	34
868	2D	34	00	34	FF	FF	FF	FF
870	FF	FF	FF	FF	FF	FF	FF	FF
878	FF	FF	FF	FF	FF	FF	FF	FF
880	49	2C	44	38	30	2C	30	2C
888	38	30	2C	2D	38	30	2C	30
890	2C	2D	38	30	2C	30	2C	30
898	0D	4D	00	34	2C	00	34	0D
8A0	FF	FF	FF	FF	FF	FF	FF	FF

The Machine Code listing required to produce the DIAMOND.

Complete decoding of the above listing, with explanations.

800	06	06	LD B,06	Load B with 6
802	3E	0A	LD A,0A	Load A with the Forward Feed instruction
804	D3	06	OUT (06),A	OUT to the printer port
806	10	FC	DJNZ 0804	create 6 loops of forward feed
808	3E	12	LD A,12	Select the Graphics Mode
80A	D3	06	OUT (06),A	OUT to the printer
80C	11	60 08	LD DE,0860	Load DE with start of Direction Change TABLE
80F	06	08	LD B,08	Sets number of colour changes before a direction change
811	C5		PUSH BC	Save B. B must be paired with C to be saved
812	06	04	LD B,04	Sets number of squares for each colour
814	21	80 08	LD HL,0880	Load HL with start of DRAWING TABLE
817	7E		LD A,(HL)	Load the data at 880 into A
818	FE	FF	CP FF	detects end of TABLE
81A	CA	23 08	JP Z,0823	At end of table, jump to 823
81D	D3	06	OUT (06),A	OUT data value at 880 to printer
81F	23		INC HL	Increment to 881, 882 etc
820	C3	17 08	JP 0817	Jump to 817 to increment through Drawing Table
823	10	EF	DJNZ 0814	Loop DRAWING TABLE 4 times
825	3E	11	LD A,11	Change to TEXT MODE
827	D3	06	OUT (06),A	OUT to printer
829	3E	1D	LD A,1D	NEXT COLOUR
82B	D3	06	OUT (06),A	OUT to printer
82D	AF		XOR A	Clear A
82E	D3	06	OUT (06),A	OUT to printer
830	3E	12	LD A,12	Select GRAPHICS MODE
832	D3	06	OUT (06),A	OUT to printer
834	C1		POP BC	Get B from STACK. Actually BC.
835	10	DA	DJNZ 0811	Decrement B and jump to 811 for 6 loops
837	1A		LD A,(DE)	Load A with data at 860 etc
838	FE	FF	CP FF	Detects end of DIRECTION CHANGE program
83A	C2	42 08	JP NZ,0842	If not zero, jumps to 842
83D	3E	11	LD A,11	If zero, change to TEXT MODE
83F	D3	06	OUT (06),A	OUT to printer
841	C7		RST 0	END OF PROGRAM ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★
842	32	9A 08	LD (089A),A	Load first byte of Dircetion Change table into location 089A
845	13		INC DE	Increment DIRECTION CHANGE table
846	1A		LD A,(DE)	Load next byte of Direction Change table into A
847	32	9B	LD (089B),A	Load this byte into location 089B
84A	13		INC DE	Increment the DIRECTION CHANGE table
84B	1A		LD A,(DE)	Load the third byte into the accumulator
84C	32	9D 08	LD (089D),A	Load this third byte into loaction 089D
84D	13		INC DE	Increment the DIRECTION CHANGE TABLE
84E	1A		LD A,(DE)	Load the fourth byte of the direction change table into A
84F	32	9E 08	LD (089E),A	Load this fourth byte into location 089E
850	13		INC DE	Increment the DIRECTION CHANGE table ready for next
851	C3	0F 08	JP 080F	Jump to 80F to commence the next direction

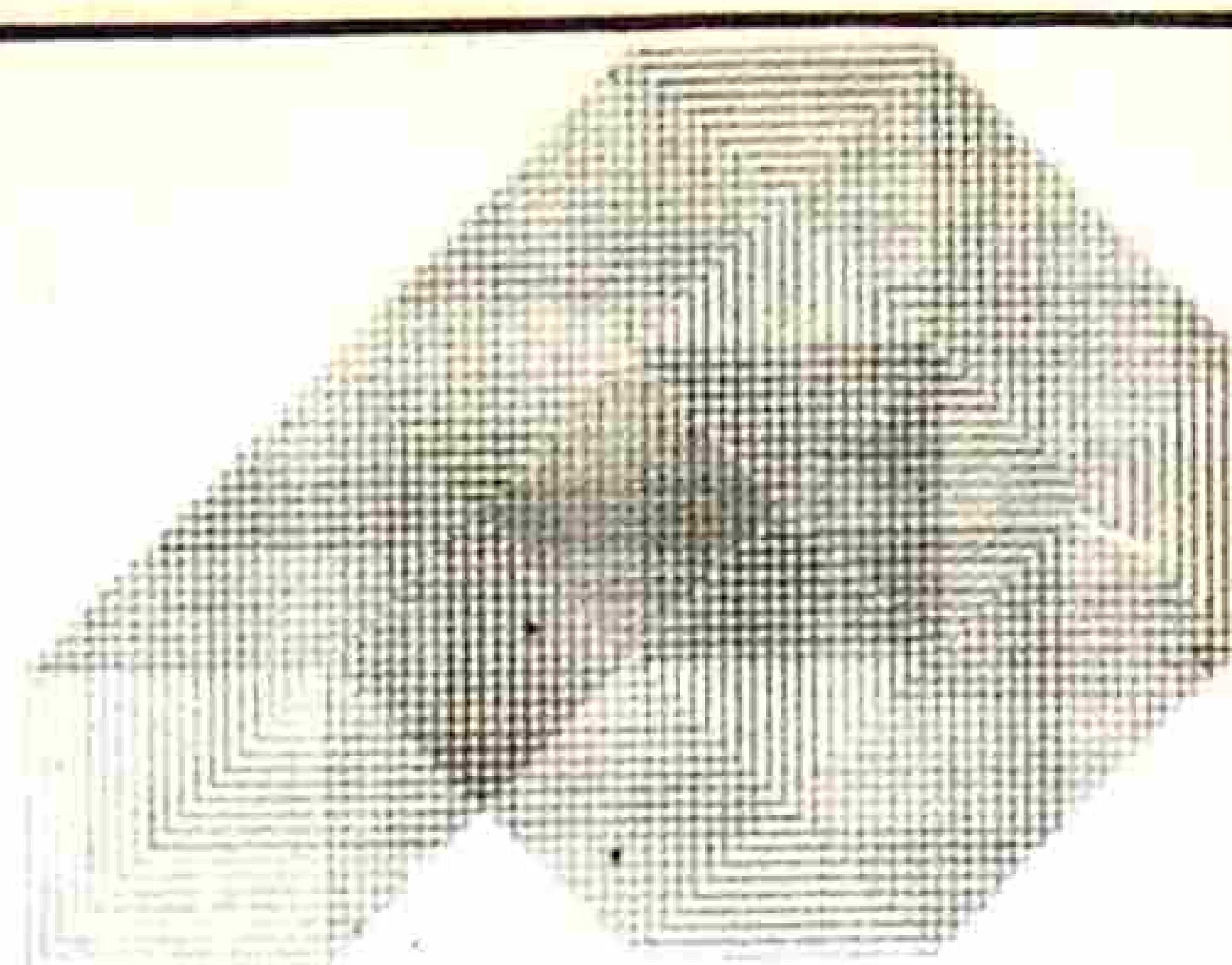
The program will then jump to **80F** and draw the second side of the diamond.

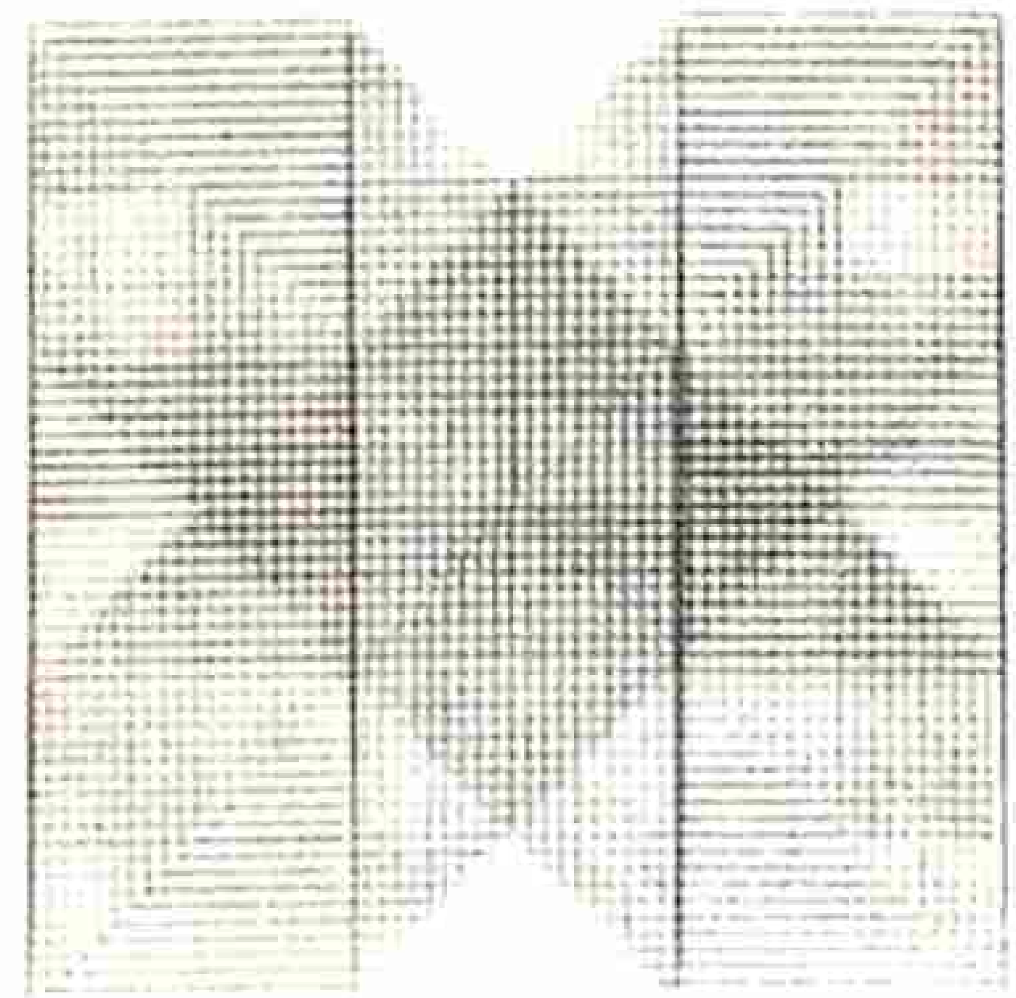
On the next pass, the register pair DE will be looking at locations **0864, 0865, 0866,** and **0867.** This will change locations **089A, 089B, 089D** and **089E** to **2D 34 2D 34** and thus the third side of the diamond will be drawn.

Via the same reasoning, the 4th side of the diamond will be completed.

Try experimenting and changing this program to produce other patterns. We have included two examples on the right and will be continuing with these and more 'add-ons' in the next issue.

The aim of COMPUTER GRAPHICS is to be able to produce 'forms' and ruled work for invoices etc. and place information in correct locations. It also gives you an understanding of ROBOT movement, an area we all would like to investigate.





860	**00**	**0**	**880**	**49**	**= I**	**88C**	**38**	**= 8**	**898**	**0D**	**= CR**
861	**34**	**4**	**881**	**2C**	**= ,**	**88D**	**30**	**= 0**	**899**	**4D**	**= Move**
862	**2D**	**—**	**882**	**44**	**= Draw**	**88E**	**2C**	**= ,**	**89A**	**00**	**= 0**
863	**34**	**4**	**883**	**38**	**= 8**	**88F**	**30**	**= 0**	**89B**	**34**	**= 4**
864	**2D**	**—**	**884**	**30**	**= 0**	**890**	**2C**	**= ,**	**89C**	**2C**	**= ,**
865	**34**	**4**	**885**	**2C**	**= ,**	**891**	**2D**	**= —**	**89D**	**00**	**= 0**
866	**2D**	**—**	**886**	**30**	**= 0**	**892**	**38**	**= 8**	**89E**	**34**	**= 4**
867	**34**	**4**	**887**	**2C**	**= ,**	**893**	**30**	**= 0**	**89F**	**0D**	**= CR**
868	**2D**	**—**	**888**	**38**	**= 8**	**894**	**2C**	**= ,**	**8A0**	**FF**	**= end**
869	**34**	**4**	**889**	**30**	**= 0**	**895**	**30**	**= 0**			
86A	**00**	**0**	**88A**	**2C**	**= ,x**	**896**	**2C**	**= ,**			
86B	**34**	**4**	**88B**	**2D**	**= —**	**897**	**30**	**= 0**			

The Direction Change table and Drawing table with decoded values.

MON-1B

LOOKING AT THE REGISTERS

The MONITOR ROM for the TEC-1A (it can also be fitted to the TEC-1) is a MON-1B. This ROM has the facility for looking at the registers.

This ROM is the result of a number of requests from readers who needed to look at the contents of the various registers during the running of a program.

If you would like one of these updated ROMs, send your MON-1 or MON-1A plus $3.00 postage and we will re-burn your EPROM to include the additional instructions.

There is a limit to when and where you can use the register facility but it can help enormously with debugging programs.

For instance, it can let you know the progress of a program or delay routine simply by interrupting it part way through.

The way this facility works is as follows:

If you reset the computer while it is executing a program, by pressing the reset button ONCE, the contents of each of the registers is pushed onto a stack.

This stack starts at **0FF0** and increases downwards to **0FD8.**

To look at any of the registers, press reset once and key the address of the register you want to look at.

The following list identifies the location of each register:

Mem ADD:	Reg:
0FF0	
0FEF	A
0FEE	F
0FED	B
0FEC	C
0FEB	D
0FEA	E
0FE9	H
0FE8	L
0FE7	IX MSB
0FE6	IX LSB
0FE5	IY MSB
0FE4	IY LSB
0FE3	A'
0FE2	F'
0FE1	B'
0FE0	C'
0FDF	D'
0FDE	E'
0FDD	H'
0FDC	L'
0FDB	I
0FDA	—
0FD9	Stack MSB
0FD8	Pointer LSB

Note: Reset clears the I register and thus it will always equal 00.

Use **JP 0000** if you wish to look at **I.**

An alternate method of saving the registers is to insert a **JP 00 00** instruction in the program at the position you wish to investigate. This will cause a JUMP to the beginning of the MONITOR ROM where it will find a jump to the register-save routine.

This will enable you to exit a program at a pre-determined point and look at the registers. The contents will be shown in the data displays.

Pushing Reset twice will destroy the information.

This is the program at **05F0** which performs the 'REGISTER-SAVE' operation. Don't forget the Monitor ROM has an instruction at 0000 to Jump to **05F0.**

```
05F0   ED   LD (0FD8),SP
05F4   31   LD SP,0FF0
05F7   F5   PUSH AF
05F8   C5   PUSH BC
05F9   D5   PUSH DE
05FA   E5   PUSH HL
05FB   DD   PUSH IX
05FD   FD   PUSH IY
05FF   08   EX AF,AF'
0600   D9   EXX
0601   F5   PUSH AF
0602   C5   PUSH BC
0603   D5   PUSH DE
0604   E5   PUSH HL
0605   ED   LD A,I
0607   F5   PUSH AF
0608   C3   JP 0580
060B   FF   RST 38H
```

SHOP TALK

A few pages on how we think, and what we do in-between issues.

After each issue of TE, everyone wonders where we disappear to! From about the fifth week after its release, we get phone calls each day asking: 'when is the next issue coming out?" "What's in the next issue?" "Are you still in business?"

It's only after you come along and see our operations, that your worries are satisfied. We operate from three rooms with a staff of three and everything pertaining to the magazine is carried out in these three rooms. This includes designing the projects, construction, testing, writing the articles, making the drawings, taking the photos, typesetting the pages, pasting them up and getting everything ready for the printer.

We then have to make up the kits, process the orders and send it off to the schools and readers.

But the magazine is only part of the story. The most encouraging response comes from the text books we produce.

So far, we have released DIGITAL ELECTRONICS REVEALED, ELECTRONICS Stage-1, and two ELECTRONICS NOTEBOOKS. We have already sold out of two of these. Schools and clubs are getting the message and ordering complete sets for their classes. With a sizeable discount, these books are cheaper than photocopying half their pages!

As amazing as it may seem, we are still getting lots of enquiries from hobbyists who have only just seen one of our publications.

Distribution has always been a big problem as electronics magazines rate very low on the display area of a newsstand.

They tend to be cramped together at the far end of the shop and even then, many issues never see the light of day. They get hidden in a drawer or lay in a pile at the back of a shop. That's the problem we have to face and the most disconcerting point of all is the in-rush of overseas magazines which take pride of place on the shelf.

Most of them deal with computers and the like, promoting one brand or another by supporting it with a few fiddly programs.

At the moment the Personal Computer is enjoying a rush of sales but this will be short lived as the real truth seeps into the market. The fact is the PC is a long way from being a useful tool and although they may be promoted as being a daily wonder, they are little more than a games machine.

The truth about video games has finally come home with a big bang. Sales of video games has come to a crashing halt with importers being left with an almighty warehouse of unsalable machines.

Even the offer of a handful of free games cartridges (valued at more than the console) has not made any impact on the sales.

Prototion for our ever-popular additional publications.

The only solution to emptying the warehouse is to offer the whole kit at less than imported price!

It is inevitable that this will happen and the only wise thing for us to do is to sit back and watch it happen.

By that time video games will be so out of date that you wouldn't want to buy one anyway.

Things are very short lived in the electronics industry. Six months is about the time for a product to rise to a peak then start to wane. By that time a new, cheaper, better product will be launched by a competitor and the market potential will be halved - if not destroyed!

The next product to see a dip will be the Personal Computer. They have alrady dived to below $100 and we will see a clearing out of at least three other makes before the introduction of a more acceptable design.

Sales are tending to go more 'up market' where the capability of adding peripherals enables the computer to expand into a more useful piece of equipment.

The back-up for the PC market, and the support of magazines is also at a peak at the moment but this will sort itself out in the very near future and with a little bit of luck TE will regain the shelf space it deserves.

Until that time comes, we will have to do a little promoting of some of our other publications.

Many readers are quite upset when they realize they have missed out on one of our 'one-shots'. In fact the majority of readers have every publication we have ever released.

To keep you up to date, here is a brief run down on an experiment we tried recently.

We produced two books called ELECTRONICS NOTEBOOK 1 and NOTEBOOK 2. They consisted of hand written pages similar to the digital electronics course but mainly dealing with simple transistor theory. The idea behind the series is to cover all those questions which are constantly asked: "How this. . ." and "Why that. . ." pertaining to transistor circuits.

Although these have been presented without any specific course in mind, the content has closely aligned with some of the courses being taught. Through this we have been able to pick up lots of school orders and if you are currently doing an introductory electronics course, you will find these books very helpful.

Notebook 1 is $2.40 plus 90¢ post.
Notebook 2 is $2.60 plus 90¢ post.

In a later issue, we will give you a few sample pages, but in the meantime you can try your local newsagent for a copy. If they are no longer available, you can send to us.

Our publications are supposed to get a 3 months shelf life to get the maximum amount of exposure. But some shops take them off the shelf after one month and this means many readers do not get the opportunity to see them.

This is the case with anyone living remote from a major town and that's why we get an enormous number of subscribers who live in the remoter parts of Australia. They find the issues have gone before they get there!

NEW PUBLICATIONS

Between producing issues of TE, we are kept busy designing other publications. The latest of these is a data book for CMOS chips. These books will already be in the news-agents and we suggest you get in quick for your copy. At a cost of $3.00 we know this represents very good value for money. We have taken 500 pages from the SGS data book CMOS B-series and condensed them into 100 larger pages. By simply photo-reducing the artwork, we have saved an enormous amount of paper and weight. These types of data books normally cost between $9 and $12 and are always in short supply.

This is the first in a series of similar data books and we intend to provide every one with a library of data books at a fraction of their normal price.

Data is one of the most expensive commodities to acquire and yet it is the most essential item for anyone building or designing.

We hope we have turned the tide on this anomaly.

The foundation supplier behind this arrangement is SGS. Although they are a large supplier of semi-conductors on the world market, they are relatively new to Australia.

In the past 12 months I have learned a little about this firm, its attitude, range and prices. I think it is worth relating.

SGS is basically a government company situated in Italy. Normally you would think anything 'government' would be slow, inefficient and lacking any drive.

But in this case it's the opposite.

Their export drive, efficiency, and price structure have been the main influence in keeping prices low in Australia.

As you are possibly aware, prices for integrated circuits have more than doubled in the past two years on the world market. This would also be the case in Australia if it were not for SGS and ELLISTRONICS.

Ellistronics have been the main distributing company for SGS and have maintained a very competitive price structure for the SGS range.

By my simple reasoning, the combination of these two firms have not only kept prices down, but reduced the prices of many of the exotic chips. And I think we all have a lot to thank them for. I have seen some of the projected prices for the 4000 series CMOS chips from other manufacturers and the rise is enormous. Quite frankly, if these were to come into operation, it would put many experimenters off construction.

The mere fact that SGS has put Australia first on their list for supplies, has removed the possible profiteering of rival firms.

Taking the range of semiconductors we use in TE, SGS prices have been 10% to 50% cheaper than competitors. This is quite considerable when you buy a large quantity and has enabled us to keep prices fixed for the past two years. But the time has come for an accross-the-board increase. We have had to lift prices by 15% (as from this issue) to cover the increases.

● ● ● ● ● ● ● ● ● ●

Even though TE magazines sell well, the Project books and Notebooks sell even better. Why?

● ● ● ● ● ● ● ● ● ●

SUBSCRIPTIONS

We at TE, produce a number of publications aimed at the electronics experimenter. These include Project Books, Text Books, and Note Books. We alternate production between these and this means TE appears only very occassionally.

This time-delay is causing a lot of worry from out subscribers. They think something is wrong.

To improve deliveries we have decided to make a change to the subscription arrangement.

Since everyone enjoys our full fange of publications, we have decided to make the subscription valid for each PUBLICATION as it comes off the production line.

This means you will be getting slightly more value for money over the full range of publications and will save you missing out on any of our releases. And you will be getting more frequent issues.

The new arrangement will be 4 publications for $15 while those on existing subscriptions will get 4 publications for their $14.00. Those with a NO PC subscription will be sent issues according to the value of their subscription.

I think this will be a very good arrangement but will not include the PROJECT BOOK series as this is already a separate series. But all our new ideas will be.

If you like the idea, send for your subscription today - especially if you live in the outer parts of the cities, where newsagents seem to only supply issues under an 'advance-order' system.

NZ READERS

It's an unfortunate state of affairs but we have had no luck at all with our NZ representatives.

We got a constant stream of complaints about BC Electronics not answering letters etc and so we changed to Clan Electronics.

Both these firms wrote to us requesting that they be our NZ representatives and so there was no coercion involved.

But unfortunately, up to now, we have exactly the same result with firm number two. Readers have informed us that they have had no response to letters and requests, from this firm either!

If this is the general rule for businesses in NZ, it shows very poor regard for their fellow country-men. On the other hand, if we have struck two very inefficient firms, let me save you the cost of sending them a letter!

New Zealand readers should send directly to us where they will get the same 24hr (actually 18hr) service we provide the Australian reader - along with a quickly written note, if required.

With the recent devaluation of the NZ dollar, and the wage structure in NZ, it costs a New Zealand hobbyist more than twice as much as an Australian reader, for any electronic component or publication. And then they are only allowed to send $100 overseas per year! Such is life.

Maybe we won't be so lucky in the future. Prices in Australia are set to rise up to 40% in the next few months. How can we ever win!!

I hope you are keeping up with our other publications. They come and go from some newsagents so fast that you may miss them. See our list of releases and keep them in mind when sending for kits.

Our binders are also a good investment as they keep all the issues together. We have now filled 2 binders with publications and already have thoughts and ideas for the next series. I hope you will be staying for these too.

LETTERS...

Lots of good, constructive comments from readers who are progressing rapidly and wanting to know more.

Most of the letters we get each day are short and specific. They can be answered immediately and everything is settled.

Others are of general interest and can be put in our LETTERS column. This way more of us can keep up with what is happening and any modifications etc to projects.

If you write to us with a request, don't expect a long, type-written reply; we havn't go time for that. All you'll get is a roughly scrawled note on a slip of paper which will be only just readable. But at least you'll get an answer the next day!

Here are some of the letter of general interest and one particularly in-depth one on the Dual Power Supply from issue 11.

Sir,
I would like to congratulate you on the excellence of TE as a learning journal. I have never before seen such easy to follow explanations of electronic theory.

When I retired in 1976, I decided that I was going to devote a lot more time to my major interest, radio and electronics. I have managed to do that, despite many other demands on my time.

However with nobody living near with the slightest interest in things electronic, acquiring knowledge and skills without the cross-fertilisation of discussion with like-minded persons has made it a slow process.

So much so that I regard a lot of the books I have bought as a collection of myths and fairy tails.

Quite obviously, in a lot of cases, the authors have never tried any of the circuits they discuss.

It is very evident to me that the staff of TE have a vast background of the very necessary practical experience.

I am learning so much that I wish that this teaching had been available years ago.

You have a knack, which I have found to be rare in teachers and lecturers, of clearly explaining the fundamentals and backing this with comments which obviously come from your own observations and experience.

I will quote one example: In your article on the 555 timer in issue 8, you mention that, under certain conditions, the IC can go into self-oscillation. I have never seen this mentioned before but I know it occurs, and was much perplexed, when I saw it on the screen of my oscilloscope. I am very pleased I lashed out and bought a CRO. It has proved to be a wonderful learning tool.

At the moment I am working on a number of devices to aid old people. They will be radio controlled and I have spoken to the appropriate department and been told that I can use a range of frequencies commencing at 26.95MHz. Can you recommend a transistor which will oscillate at that frequency and put out about 100 milliwatts.

Because of the possibly interference from CB, static and car ignition systems, I have decided that the signal must be encoded.

If I modulate the 27MHz carrier by, say 20kHz, the receiver would accept this signal, square the modulation into pulses and feed the result into a 4040 counter. When a certain number of pulses had been counted, the output would drive a relay which would latch ON.

To avoid the accumulation of pulses from spurious sources, a 555 could inhibit the counter at a rate of say twice a second. Does this sound a practical way of proceeding?

R Barnes,
Bribbaree, 2594.

Without actually seeing the whole design I cannot give any comment. The only thing I can say is the danger of using a 555. They quite often create more trouble than they solve. They are extremely noisy devices and can cause false clocking in the surrounding chips. I would suggest something like a Schmitt Trigger oscillator made from a 74C14 and, at the same time, the current demand will be reduced appreciably.

While on the topic of coded transmitters, we will be presenting a 27MHz short-range transmitter using a single chip to provide over 1,000 possible output codes. This is still in the planning stage but will be a possibility for a future article.

Sir,
Could you please inform me if a BFO metal detector is being planned.

WJ Chapman,
Bowen, 4805.

Metal detectors and the like have been fully covered by other magazines. One publisher produced a magazine completely devoted to this topic. Try some of the larger magazine distributors for a copy of "Constructing Metal Detectors".

The gold boom has almost but died and I don't think there is much call for that type of project at the moment.

Sir,
Re: your computer. This seems to be a computer with a lot of potential. At present I have a 16k Microbee and I am only just getting into computers and Machine Code programming.

But in issue 10, I learnt more about the code than in the past 6 months of ploughing through an array of other books on the subject.

I am considering buying a TEC for a couple of reasons. To learn Machine Code, and to produce a simple yet powerful computer to operate a robot.
Before I picked up an issue of TE, I was all fired up to purchase a computer from a South Australian firm. But two things swung me in your direction. The neatness of the project and also the back-up it will get in the form of soft and hardware from the magazine.

Also the fact that the CPU is a versatile Z80.

However I have a few questions to ask before I go ahead with the purchase.

1. Firstly, can the Z80 be substituted with the faster Z80A?

2. Will the project be supported with regular columns and projects?

3. What additional software is planned for the future?

4. Will you be designing a project with input/output ports so that data from sensors can be fed into the computer?

5. Is is possible to expand the TEC above 12k?

6. Will it be possible to fit an RS-232C serial port to the TEC so that printers and speech synthesisers can be used.

7. And finally, will there be a project that will provide connection to a video monitor?

D Hughes,
Howrah, 7018.

The Z80 operates at a maximum of 1.5MHz and can be substituted with a Z80A, enabling the computer to operate at 4MHz. At this frequency a crystal oscillator is suggested and will be the subject of a future article.

Each issue will present further projects for the TEC and these will include input/output and speech. New speech chips are coming on the market all the time and the price is coming down all the time. When they are available for less than $20, we will be presenting an article.

The video board is still in the designing stage and the major problem here is the complexity and cost.

Some of your needs are provided in this issue and we have six more add-ons in the completed stage.

Most of your requirements will be possible with the TEC and you will see how it all goes together after the next issue.

Because of my age(67), I grew up and spent most of my working life in the valve era. My knowledge of Digital Logic Theory is therefore rather inadequate. This prompted me to build your Logic Designer (Project Book 2) to gain experience.

I found it a most absorbing project and upon completion it performed as specified, counting both decimal and binary accurately, and yielding the correct results in the experiments.

However I have a problem. The 4026 has burnt out segment 'd' in the display on two occassions. None of the others are affected.

Can you shed some light on this problem?

N. Hoffman,
Mt Kuring-gai, 2080.

The problem lies in the 4026 driver chip. These chips are supposed to be AUTOMATIC CURRENT LIMITING and this means no dropper resistors are required in the segment lines. Chips from some manufacturers have this feature of full current limiting and will operate up to 9v, others provide very poor current limiting, even at 5v. The result is either a burnt out segment in the display or a very hot chip.

The fact that some chips fail to current-limit was not realized until the project began to be constructed by readers in other states, where different brands of 4026's were available. A simple way to overcome this problem is to add a 47R dropper resistor between pin 3 and earth on the FND 500 display. This will require cutting the track and placing the resistor on the underside of the board.

Writing to us helps both yourself and us. We don't know what trouble you are having without this valuable feedback.

Sir,
I refer to certain errors and omissions in the Dual Power Supply article in issue 11.

Firstly, the circuit diagram on P.5. has pins 2 and 3 reversed on both IC's and pin out of the components. The pins are correct on the PC board, however.

I will answer your points, one at a time as presented in your letter.

The first point you mention is incorrect. The pin numbers on the schematic and pin-out diagram are correct. The pin numbering does not go 1, 2, 3 but 1, 3, 2. This causes some confusion both in our workshop and with constructors. The only consolation is that a 7805 cannot be insered in a 7905 position. It will simply fail to work.

Secondly, on the same diagram, the LED dropping resistors are shown as 1k. This gives sufficient current to operate them at 12v (10mA) and 15v (13mA), however at 5v they will be hardly seen at all.

I suggest a 330R resistor.

Your second point is correct however we have opted for a resistor which will cover the full range of output voltage without damaging the leds.

If you decide on a pair of 5v regulators, you can replace the resistors with 330R.

Thirdly, I note that the maximum current considered for each regulator is 600mA. I note that 700mA is the maximum which can be safely drawn from the power transformer.

However there is another limiting factor. The diodes are rated at 1 amp maximum forward current. When mounting them together in a bridge arrangement on a PC board with little copper tracks, the maximum current should be limited to 500mA. This point should be mentioned in the text for those who wish to modifiy the current rating considerably.

This point is also a valid suggestion. The diodes certainly form part of the current limit consideration. However, as mentioned in the article, the regulators were the first to heat up. The diodes remained relatively cool and it would seem that they were within their heat dissipation.

Fourthy, I draw your attention to your statement: "The 100n capacitor prevevents high frequency oscillations from occuring." This is a real problem in practice and without warning, the regulators start to oscillate at about 1MHz or 10MHz and go into thermal breakdown. The permanent cure is to install 2 tantalum capacitors on the actual leads of the regulators. Mount a 0.1ufd between input and common, and a 1ufd between output and common.

Fifthly, I find it hard to believe that there will be zero volts between two 2155 transformers when connected incorrectly. This is because they are not perfect inductors.

When we tested two 2155's we got an absolute zero reading. This is because both transformers are identical and have an exact 180 degree phase shift in the output.

Finally, the holes in the PC board are not plate-through. This should be mentioned in your article.

D. Brownsey,
Fortitude Valley, 4006.

How many readers would expect the holes to be plate-through? Don't forget we are only an experimenter's magazine.

This type of letter greatly appreciated as it shows we are being read by more than just the beginner.

It can also bring our attention to any faults which arise in the projects.

THE TV MAN TELLS. . .

Three stories from his wonderful bag of tricks.

Remember, many issues ago, I said I had so many stories to relate, that I couldn't fit them all into the one article. Here they are now. . .

As technology improves, and new models emerge from the manufacturers production line, most of the old faults are being cleared up, only to be replaced with newer and more complex ones. I would prefer the old faults as I have spent many hours locating and solving them, and they owe me a considerable amount, both in time and money.

It can take up to 10 faults to recoup the costs which have been incurred in locating the first fault and, even today, many faults have not repaid their debt!

This is a point little understood by customers. They think a repair should be charged according to the amount of time spent on the job. Unfortunately this is not the case, and never will be.

If you were to apportion costs strictly according to the fault, you would find some repairs costing more than a set in a second-hand shop!

That's why repairs have to be levelled to a basic figure; so that you win on some and lose on others.

The second, and most important reason for arriving at a standardised figure, is to avoid embarrassment.

How can this be?

Take the simple case of a repair performed in the home. Only once you have actually fixed the fault or or are nearing completion, are you in a position to advise the customer of the cost. It is absolutely impossible to guess a price, even though you may have carried out similar repairs in the past.

Your patter goes something like this: "I think I have found the fault and it will need one of these, and one of these and two of these etc. etc." "I can repair it for you today and it will cost about $55."

Most customers will give you the go-ahead immediately and everything will continue smoothly. The most important point to remember when quoting is to give a figure which you will be able to remember!

Between the time the customer accepts the quote and the final completion of the job will possibly be only a few minutes, but in that time you can be so pre-occupied that you forget the amount of the quote! Yes, I actually forgot once! Never again, I now have a set of standard prices which are low enough not to shock the customer yet high enough to cover the cost of the job.

I have found it better to charge an average, over-all, fee in preference to creating a special price for each job.

And customers prefer this too. Many new customers are recommendations and already have a idea of the charges from Mrs Brown and Mr Jones. They expect costs to be in the same bracket for their repair and quite often have the exact amount of cash ready on the TV!

Unlike car repairs, where the customer can differentiate between a clutch fault and a brake fault, they are unable to distinguish between a picture fault and a sound fault. ANY fault is the TV man's responsibility and if the TV fails again, within the guarantee period, he is expected to fix it free of charge. If you try to charge, you will lose a customer!

To cover this guarantee work, you must allow between $4 and $6 on every repair.

So, now to the faults.

This time they mainly apply to new sets. Sets which are supposed to be trouble-free.

Almost without exception, the cost of replacement parts for a new set is less than for an earlier model. This is because the fault is mostly a single component such as a transistor, resistor, capacitor, diode or a dry joint.

Since most new sets are of Japanese origin, we cannot say reliability has improved because Japanese sets have always been very reliable.

The only improvement has been in the layout of the parts. All the first-generation sets were a night-mare to work on. The PC boards were crammed with components and many parts were mounted behind larger items so that their removal was almost impossible.

Fortunately much of the dogs-breakfast has been cleared up but the major disappointment with the new sets is the absence of modular construction. The only aspect which made the old sets fixable was the fact that a whole module could be taken out and a change-over one fitted for testing purposes. Once the faulty module was located, it could be either repaired or a new one fitted.

Modular servicing has its advantages and disadvantages and I honestly consider it is the best method of producing a piece of equipment for the consumer market.

However manufacturers have thought otherwise and opted for a single mother-board layout.

This has literally made many of the new sets unrepairable for reasons you will see later in the article and also on economic grounds. Many technicians are not capable of analysing a whole TV and prefer the speedier approach of module changing. When the faulty module is found, a decision can be made for its repair or replacement. But when the whole board has to be worked-on, the frustrations are increased considerably.

Fortunately many of the newer sets have been simplified and this has resulted in fewer components and a more open layout. Some of the components have been incorporated into IC's while others have been omitted altogether.

This has not resulted in any appreciable loss of picture quality, just a rationalising of componentry.

One major disadvantage with single board construction becomes apparent when water is spilt down the back of the set. This occurs in a sufficient number of cases to be a real worry.

When you are told of this happening, you must make a decision as to whether you will attend to the job *poste haste* or wait for the water to dry out. A bad decision can render the set unrepairable, while on the other hand, you may need to make two or more trips to complete the repair. The first call will enable you to see the extent of the damage and determine how much water remains on the board. The second and subsequent calls will be needed to complete the job.

The amount of damage will be different with each set and will largely depend on the type of water (dirty plant water, coffee, tea, or clean water) and the quantity spilt. It is also important to know which section of the board has been wet.

It is only after attending to a number of these accidents, that you can make a decision over the phone. If it is your first experience with water, you should attend the job as soon as possible - even if it means a night call. Plant-water contains salts which readily attack fine copper wire in the transformers. After a few days the copper looks like **copper sulphate!**

If you attend to the set in time, you can displace the water with a pressure-can of propellant such as 'CO CONTACT CLEANER'. This has the effect of dispelling the water and evaporates very quickly, leaving no residue. It is dynamite to use sprays which contain a lubricating agent such as oil as this will be left behind when all the solvent has evaporated. 'CO CLEANER' leaves the board quite dry and although it is expensive, you will save hours of work in the long run.

Such was the case with the first repair. Water had been spilt down the back of the set some months prior and nothing had been said or done about the accident.

At first it did not seem to affect the performance of the set, but finally it lost colour. It wasn't complete loss, of colour, just extremely difficult to lock into position via the fine tune control.

The set in question was a fairly new HMV model, which is really a well-known Japanese make, overstamped with HMV.

On taking the back off the set, it became obvious that the water had dribbled onto the PC board some months ago. All traces of moisture had vanished, leaving only a dirty, muddy stain.

The rest of the set was thick with dust and we can safely assume that all the board was once covered with an equally thick layer.

The water had consolidated the dust around the chrominance section, leaving most of the transistors with traces of electrolysis and carbonising between the leads. This would have a dramatic effect on the performance of the amplifying circuits and the first task was to carefully clean away all the carbon and scum.

After this initial operation the set was turned on and was found to show the same fault. The colour would not lock into position and some of the channels remained black-and-white. It was now clear that some of the water had penetrated under the covers of the pot-core transformers and fixing the set was going to be a protracted task.

To remove each of the transformers, test their continuity, and replace them, would be a big job, but it had to be done. Otherwise the whole set would have to be scrapped.

Imagine the advantage of a modular system!

As the whole circuit was contained on a single board, there was no chance of getting a replacement board. If it were possible, the cost of the board would be half the cost of the set and additional time would be needed to connect all the input and output wires.

So, the only solution would be to remove each transformer and test it. This is not an easy job at the best of times. Desoldering tools do not help very much. They are not capable of fitting over the lugs of the transformer case, so the only way is to use a soldering iron, desolder braid and SPEED.

A hot soldering iron is applied to three pads at once and this will allow one side of the transformer to lift off the board slightly. With a see-sawing motion, one side and then the next is heated until finally, the hot transformer comes away.

By carefully checking each winding with a mulitmeter set to low ohms, the primary and secondary can be tested for continuity.

My luck came on the third transformer! One of the fine wires had been completely eaten away by the acid effect of the water. It was necessary to add an equally fine length of wire, obtained from a piece of super-flex hook-up wire, to make the repair. Trying to tin the solderite wire with a soldering iron is a difficult task. Theoretically, the enamelling on the wire will burn off if the soldering iron is held against the wire long enough. I patiently kept the soldering iron applied and the wire eventually took solder.

It is impossible to scrape the enamel off with a file or knife without running the risk of breaking the wire. If it were to break next to the main winding, you have no chance of repairing it. So you have to use heat.

Eventually I managed to solder the two pieces of wire together and then solder it to the terminating pin.

When I refitted the transformer and switched the set on, it worked perfectly, much to my relief.

This type of repair must contain a high fee for possible recalls as additional faults could show up at a later date. And you will be expected to repair them FREE!

Fortunately I have been lucky with this set. It has been over 6 months since the repair was carried out, and no further faults have come to light.

The second fault also involved a new-style set. It was a Rank 63cm. When you take the back off these sets, you get quite a shock. In comparison with the first generation sets, they have only a fraction of the cabinet taken up by componentry. Ninety per cent of the inside of the cabinet in empty space!

Situated below the picture tube is a single PC board containing all the components.

The fault with the Rank in this story was a black screen.

When I turned the set on, I heard the power supply start up normally, indicating that a short-circuit or heavy load was not pulling the power supply down. But, the EHT section was not producing a high voltage. This I ascertained by placing a screwdriver against the bobbin of the EHT transformer and checked for a discharge.

I then tested the case of the horizontal output transistor (the collector) and found the full supply voltage was present. The fault had to be somewhere in the horizontal oscillator section such that the horizontal output transistor was being turned off.

Now, since these sets are so reliable in design, I decided to go for the weakest link. This is the solder connections on or around the leads of the output transistors.

There I found it. The leads had created their own dry joints due to heating and cooling of the transistor.

Careful soldering with the application of additional solder is all that is necessary to produce a perfect joint.

Again, I have not heard from the customer. And possibly the set will never break down again. That's the extent of the reliability of Japanese sets.

Lastly, the third of our stories. It was an AWA set with very poor focus. These are Mitsubishi sets and once again they are designed on a single PC board.

In general, poor focussing can be attributed to focus voltage. If this voltage is not correct, the spot will not be fine when striking the face of the tube and the picture will appear to be blurred. Normally, a slight adjustment of the focus control will make the picture come clear. But not with this set. No amount of turning would improve the picture.

A focus control consists of a very high value of resistance (10M to 100M) and has 'stopper' resistors at each end. These can change value over a period of time or even burn out. And the focus control itself can wear out, break, or become 'spotty'.

In this case the wiping arm had broken and the control had to be replaced.

Unfortunately the focus control was of special design. It was mounted on a piece of insulating board and the full focus voltage from the tripler was fed into one terminal.

The two stopper resistors were contained inside the control and withot these, the voltage to the tube could be increased to a point where flash-over occurs.

Since I did not have one of these focus pots in my kit, I had to make do with the closest type. This is a rotary 10M focus pot. It does not have good insulation resistance between shaft and wiper and should not be mounted on an earth point.

Being guided by the mounting of the previous control, I mounted it on the insulating board and connected stopper resistors to the two END terminals.

When the set was switched on, I detected over heating in one of the resistors. Even though only a few milliamps flow through this network, the voltage across each resistor was sufficient to create a wattage problem. I changed one of the resistors for two separate resistors and the problem was solved.

Always check any new parts or modifications for over heating by running the set for a minute or so then turning the set off and feeling each component. Don't try doing this with the set on or you'll get bitten!

This is only a couple of faults but should give you an idea of the new breed of troubles. In fact they arn't really new, just more sophisticated.

The main difficulty arises when you have to fight off the notion that new sets are trouble-free and get down to the job of locating the fault.

It may take time for faults to develop but most new sets have a range of faults to take the place of the old. As yet, I cannot see the role of a TV technician dwindling. He will still be needed for a few years yet.

Hopefully I will still be around and tell you more of my exploits.

Even NEW sets are starting to break down. But the economics of repairing them is a whole new ball-game.

While on the subject of new sets, I thought I would add this one:

Some time ago I mentioned the importance of working in a dark room.

Abiding by this rule has helped me once again. It turned a 3 hour repair into a 5 minute repair.

Recently I was called to repair a new Pye 56cm colour set. As soon as you hear the word PYE, you immediately assume it to be either a Japanese set or a Philips set.

Pye no longer make their own models and this is a great relief, after the disasterous attempt at making an Australian set fizzed out about three years ago.

To my surprise, the set turned out to be Philips. And the model was new to me. It was one of the new models, especially made for Pye. And, according to the custom of Pye, the circuit diagram was not included with the set.

This is an enormous failing of a set manufacturer. Not only does it show lack of faith in the product, but very soon reflects back in their sales.

It works like this:

A TV serviceman is a very powerful representative. If he finds it difficult to repair a set (through lack of circuit diagrams or the cost of spare parts etc), he will condemn the make of set. This upsets the customer who avoids the particular make when buying another set and tells two or three friends in the process.

We have a lot of power. This is borne out by the enormous growth of Philips' sets in their second generation.

This has been mainly due to the serviceman. In general they have found Philips sets to be the most easily repaired and the best set for picture quality, even after the third and fourth year of service.

Most other sets, including the suposedly superbly designed German sets, have failed miserably in Australia. After two or three years the quality of the picture on these sets is little more than a coloured comic. The overall brightness level of the tube is also lamentable.

Back to the fault.

When the set was switched on, the EHT failed to come up and thus the set appeared dead. But a very short 'plop' was heard each time the power was applied and it seemed that some thing was failing under load.

Fortunately I was working in very low level light and as I turned the set on, I saw a flash come from one of the diodes near the EHT transformer. I could actually see the short-circuit occuring inside the body of the diode.

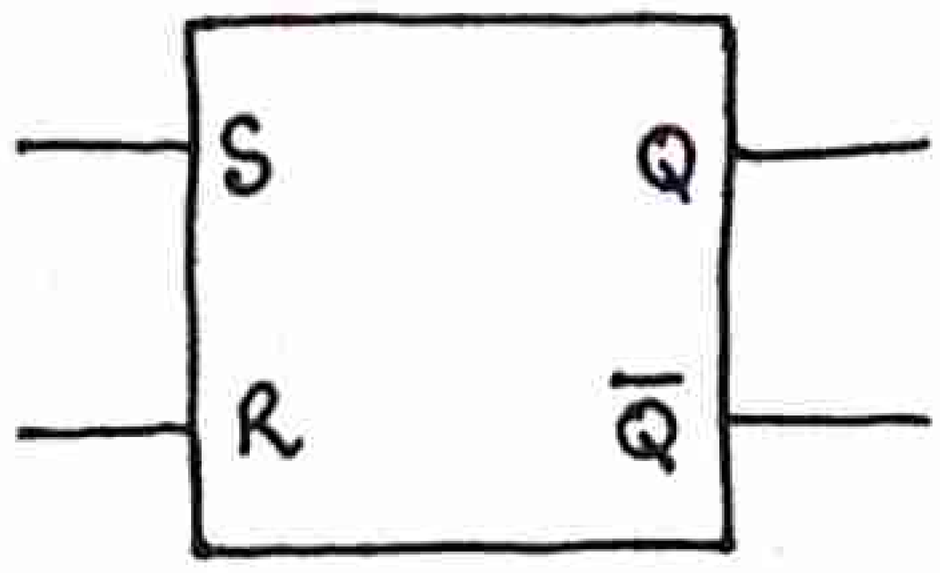
This was quite a stroke of luck as the diode was only breaking down under load and the normal test procedure would show the diode to be functional.

When the diode was replaced, the set worked perfectly.

Since then I have fixed two more sets with the same symptoms but was not fortunate enough to see the flash inside the diode.

Many readers have used the facts contained in these stories to fix their own set. In a future issue we will relate more stories and include a transistor substitution list.

SUMMARY OF THE R-S FLIP FLOP (LATCH)



R-S FLIP FLOP — (ACTIVE HIGH TYPE)

BASICALLY THE R-S FLIP FLOP IS VERY LIMITED IN OPERATION. IT IS ONLY CAPABLE OF CHANGING BACK AND FORTH BETWEEN ITS TWO STATES: SET & RESET EACH TIME THE S & R LINES ARE REVERSED. IT CANNOT BE "FROZEN" OR CLEARED OR PROVIDE ANY OTHER FEATURE SUCH AS FREQUENCY DIVISION. — THESE ARE LEFT FOR THE MORE ADVANCED VERSIONS.

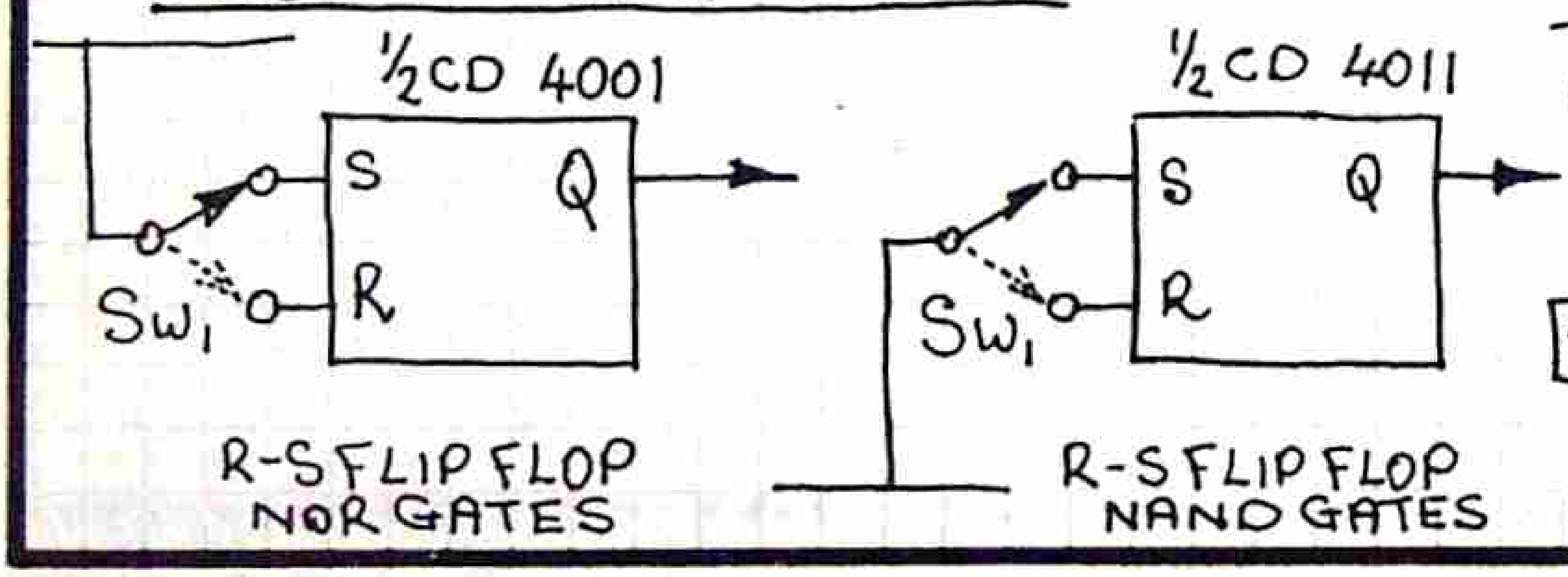
WHEN A FLIP FLOP IS "SET" THE NORMAL OUTPUT Q IS HIGH. THE BLOCK DIAGRAM ABOVE SHOWS AN ACTIVE-HIGH FLIP FLOP IN WHICH THE SET & RESET INPUTS CHANGE THE STATE OF THE FLIP FLOP DURING THE LOW-TO-HIGH PORTION OF THE PULSE. IN OTHER WORDS IT IS MADE FROM NOR GATES.

BUT TROUBLE ARISES WITH ANY TYPE OF R-S FLIP FLOP WHEN THE INPUT LINES ARE TAKEN HIGH OR LOW TOGETHER (DEPENDING ON THE GATES USED.)
THE OUTPUTS FAIL TO CO-ORDINATE AND FALSE READINGS CAN BE PRODUCED. — THIS CAN BE PREVENTED BY CAREFUL CIRCUIT DESIGN. BUT THE MAIN REASON FOR THE LIMITED USE OF THE R-S FLIP FLOP IS ITS SINGLE MODE OF OPERATION.

THE R-S FLIP FLOP IS A SIMPLE MEMORY CELL. IT CAN BE USED IN CONJUNCTION WITH A TOGGLE SWITCH TO SHOW THIS FEATURE.

REFER TO THE DIAGRAM ON THE PREVIOUS PAGE OR THE TWO BLOCK DIAGRAMS BELOW. THE SET PULSE CAN BE APPLIED FOR ANY LENGTH OF TIME & MUST THEN BE REMOVED. THE FLIP FLOP WILL THEN REMAIN SET. THE RESET PULSE WILL THEN RESET THE [illegible] FLOP & THE PULSE MUST THEN BE TAKEN AWAY. THESE PULSES DO NOT HAVE TO BE APPLIED IMMEDIATELY AFTER ONE ANOTHER — THE ONLY STIPULATION IS THAT THEY ARE NOT APPLIED AT THE SAME TIME. THE PULL-UP OR PULL-DOWN RESISTORS SHOWN IN THE DEBOUNCE CIRCUITS PREVENT THE FLIP FLOP FROM SEEING THIS UNWANTED CONDITION.

DEBOUNCING A SWITCH



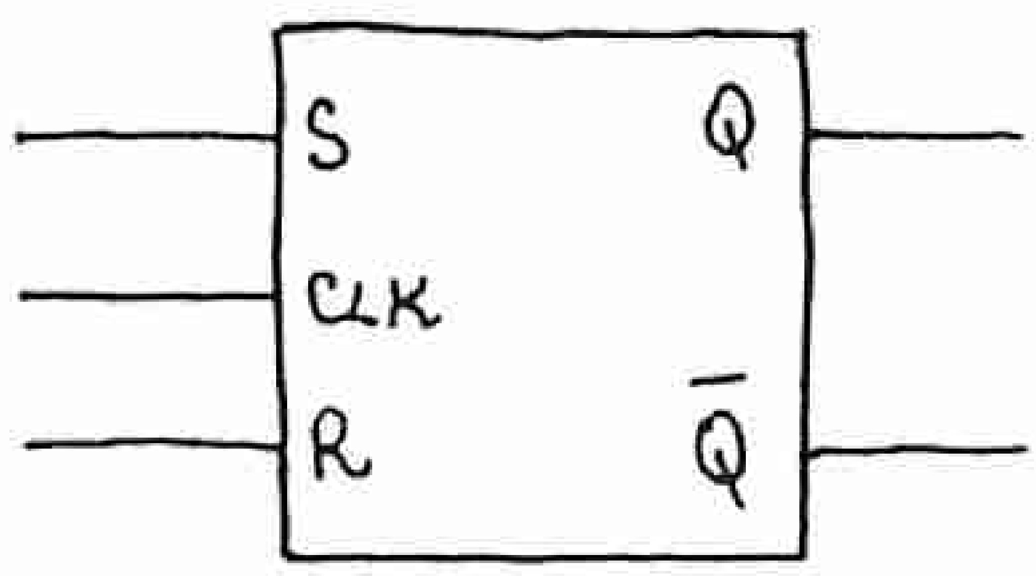
BOTH CIRCUITS PRODUCE EXACTLY THE SAME RESULT. THE ONLY DIFFERENCE IS THE CONNECTION OF THE SWITCH TO EITHER HIGH OR LOW RAIL.

[THE INPUT RESISTORS HAVE BEEN OMITTED FOR CLARITY. BUT ARE NEEDED]

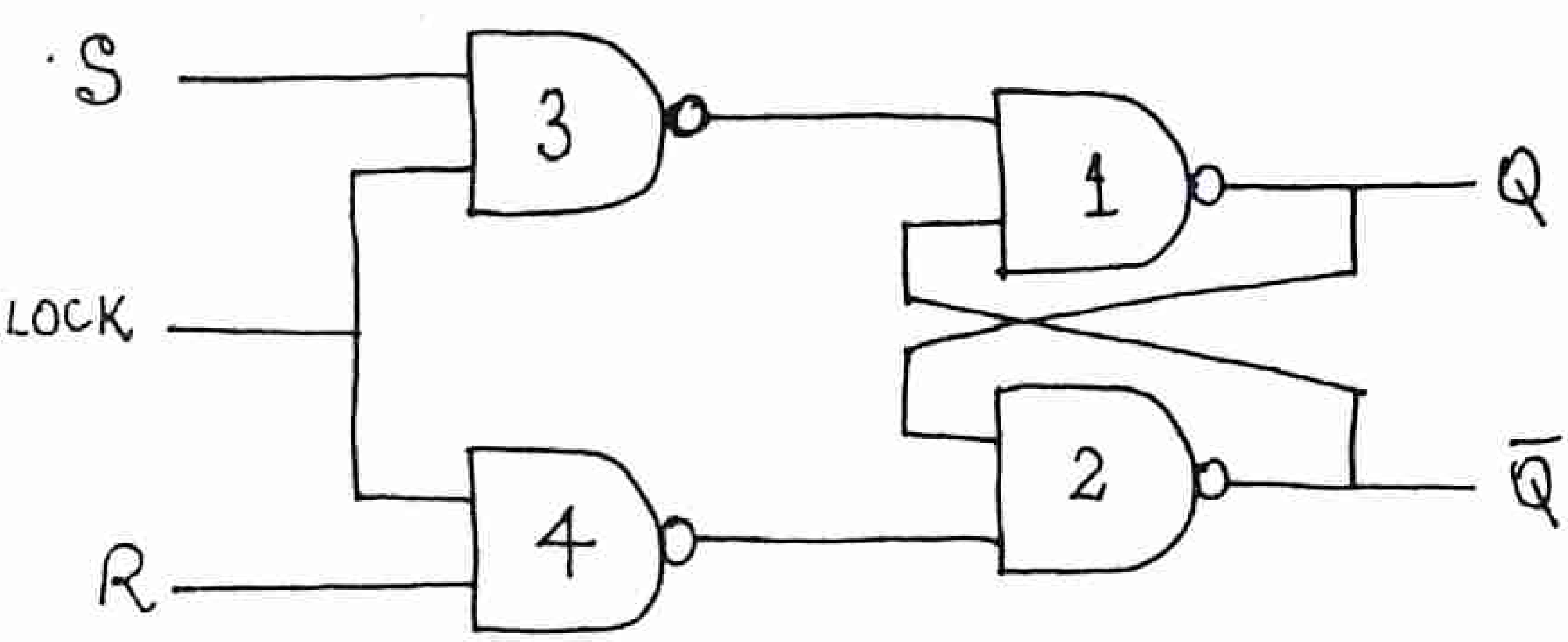
CLOCKED R-S FLIP FLOP

THE BASIC R-S FLIP FLOP IS AN ASYNCHRONOUS DEVICE. IT DOES NOT OPERATE IN STEP WITH A CLOCK. WHEN ONE OF ITS INPUTS IS ACTIVATED, THE NORMAL OUTPUT IMMEDIATELY RESPONDS. THIS IS A LIMITATION. LARGE CIRCUITS QUITE OFTEN NEED TO BE TIMED THROUGHOUT SO THAT A SET OF EVENTS CAN FOLLOW A PRESCRIBED SEQUENCE. THIS MAY INVOLVE HOLDING DATA FOR A PERIOD OF TIME UNTIL REQUIRED. THIS IS THE CAPABILITY OF THE CLOCKED R-S FLIP FLOP.

THE CLOCKED R-S FLIP FLOP IS A SYNCHRONOUS DEVICE. A HIGH OR LOW LEVEL ON THE R OR S INPUTS DOES NOT CAUSE AN IMMEDIATE CHANGE IN THE STATE OF THE FLIP FLOP. THE ARRIVAL OF THE CLOCK PULSE IS NEEDED TO COMPLETE THE OPERATION.



CLOCKED R-S FLIP FLOP.....

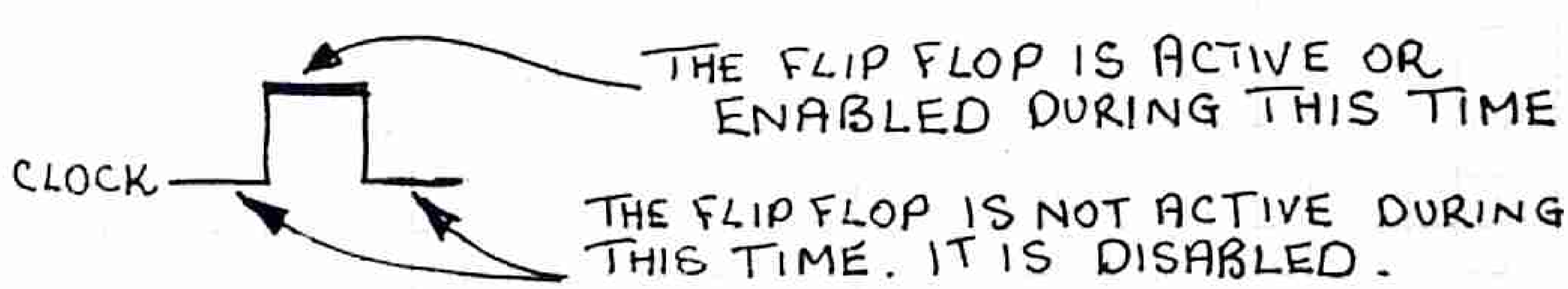


.....USING NAND GATES.

GATES 1&2 MAKE UP THE R·S LATCH, AND FROM A PREVIOUS DISCUSSION THIS IS AN ACTIVE LOW ARRANGEMENT.

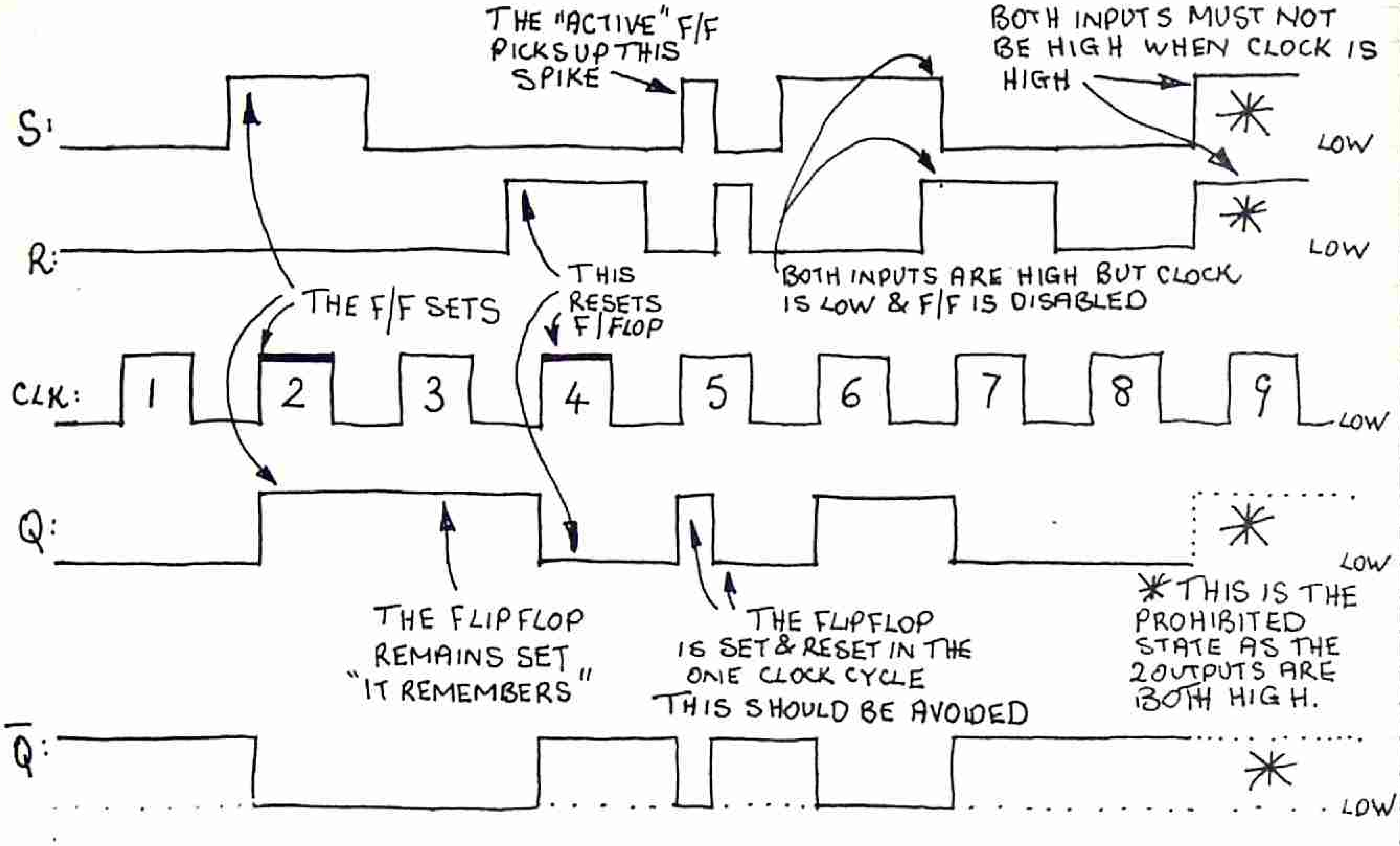
GATES 3&4 GATE THE CLOCK PULSE TO THE R-S LATCH AND BECAUSE OF THE INVERTING EFFECT OF THESE GATES, THE R&S INPUTS ARE NOW ACTIVE HIGH.

THE CLOCK LINE TRIGGERS THE FLIP FLOP WHEN THE CLOCK PULSES ARE HIGH & THIS IS A LEVEL-TRIGGERED DEVICE.



THE NORMAL ARRANGEMENT IS TO SUPPLY DATA TO THE R - S LINES BEFORE THE ACTIVE PORTION OF THE CLOCK PULSE. THIS MEANS THAT AS SOON AS THE CLOCK PULSE ARRIVES, THE FLIP FLOP WILL PROCESS THE INFORMATION.

A TIMING DIAGRAM FOR THE FLIP FLOP SHOWS THE EXACT STATE OF THE OUTPUTS FOR THE VARIOUS COMBINATIONS ON THE INPUT LINES.



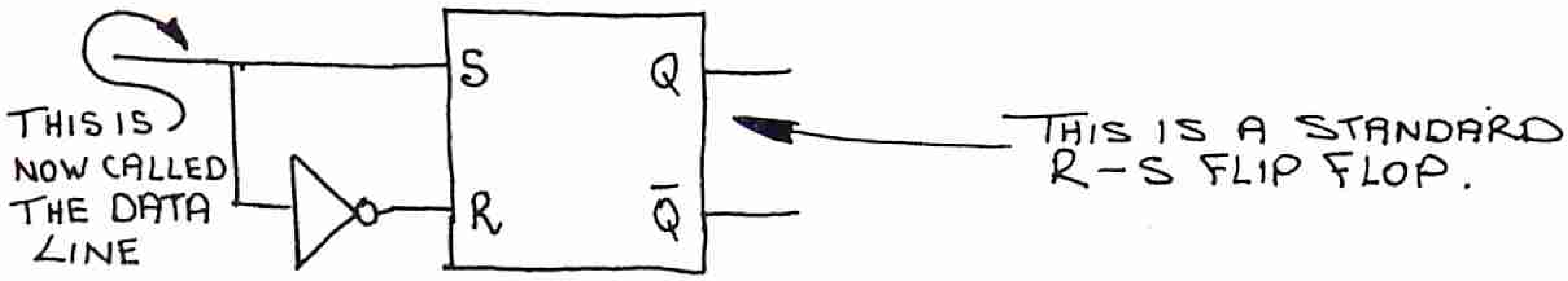
THE SEQUENCE OF EVENTS IS AS FOLLOWS:

THE "SET" INPUT ARRIVES BEFORE THE OPENING EFFECT OF THE 2ND CLOCK PULSE & THE FLIP FLOP WAITS FOR THE CLOCK PULSE TO ARRIVE. THIS PULSE "SETS" THE Q OUTPUT (& THE $\overline{Q}$ OUTPUT IS RESET). CLOCK PULSE 3 SEES THE R & S INPUTS IN THE HOLD MODE AND THEREFORE THE OUTPUTS DO NOT CHANGE. — THE FLIP FLOP ACTS AS A MEMORY CELL.

CLOCK PULSE 4 "SEES" THE RESET LINE HIGH & THIS CAUSES THE FLIP FLOP TO RESET OR CLEAR THE Q OUTPUT. CLOCK PULSE 5 SEES A HIGH FOR HALF THE CLOCK CYCLE AND A LOW FOR THE OTHER HALF. THE FLIP FLOP RESPONDS AS SHOWN. THIS EMPHASISES THE FACT THAT THE INPUT LINES SHOULD BE HELD STABLE TO PRODUCE A PROPERLY GATED OUTPUT. BETWEEN CLOCK PULSES 6 & 7 BOTH INPUT LINES ARE HIGH FOR A SHORT PERIOD OF TIME. THIS DOES NOT AFFECT THE FLIP FLOP AS THE CLOCK LINE IS LOW & THE FLIP FLOP IS DISABLED. — IT DOES NOT SEE THE PULSES DURING THIS INTERVAL OF TIME. CONDITIONS ARE CORRECT FOR THE OPENING OF THE FLIP-FLOP AT PULSE 7.
INPUTS ARE NOT "ACCEPTABLE", HOWEVER, ON THE ARRIVAL OF CLOCK PULSE 9. BOTH LINES ARE HIGH AND THIS CAUSES BOTH OUTPUTS TO GO HIGH. THIS IS A DISALLOWED STATE AND MUST NOT BE ALLOWED TO OCCUR.

THE D FLIP FLOP

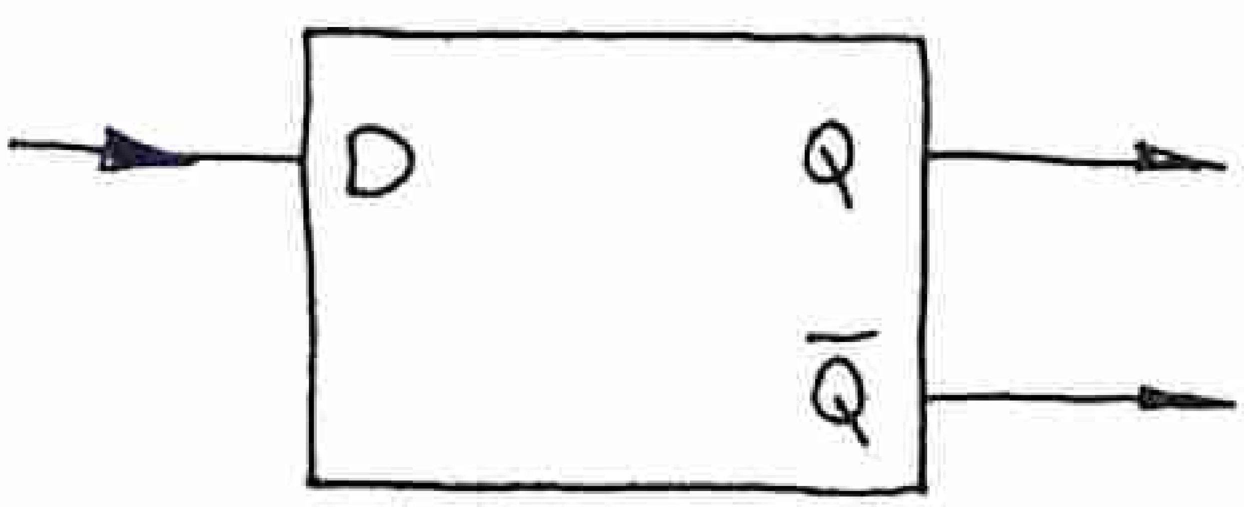
TO PREVENT THE "LIMBO" OR UNDESIRABLE CONDITION FROM OCCURING IN THE R-S FLIP FLOP, IT CAN BE MODIFIED BY ADDING AN INVERTER TO ONE OF THE INPUT LINES & JOINING THEM TOGETHER TO PRODUCE A SINGLE INPUT LINE. WITH THIS ARRANGEMENT WE HAVE PRODUCED A D-FLIP FLOP.



BASIC D FLIP FLOP

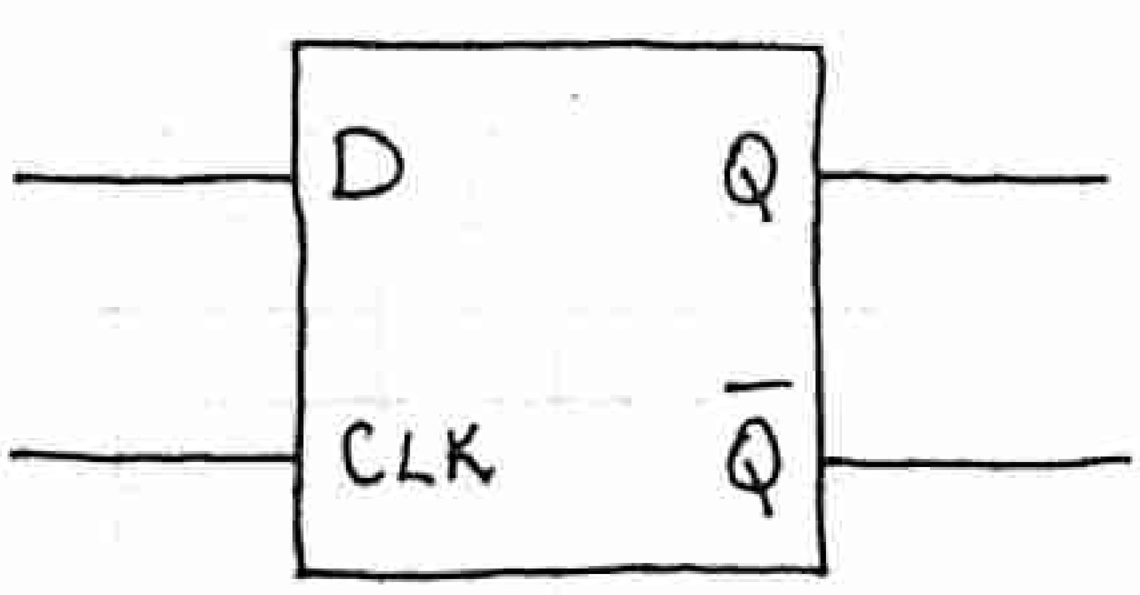
THE INVERTER AT THE INPUT MAKES IT IMPOSSIBLE FOR THE R & S INPUTS TO HAVE THE SAME STATE. THUS THE OUTPUT CONDITIONS FOR THE FLIP FLOP ARE PREDICTABLE.

TO SIMPLIFY THE DIAGRAM WE CAN INCORPORATE THE INVERTER INTO THE FLIP-FLOP PACKAGE THUS:



THE R INPUT IS NO LONGER AVAILABLE AS IT IS AUTOMATICALLY THE COMPLEMENT OF THE DATA LINE.

THE NEXT IMPROVEMENT INTRODUCES AN INPUT LINE WHICH ALLOWS THE TIMING OF THE FLIP FLOP TO BE CONTROLLED. THE LINE IS CALLED A CLOCK LINE. IT CONTROLS THE MOMENT WHEN THE FLIP FLOP CHANGES STATE.

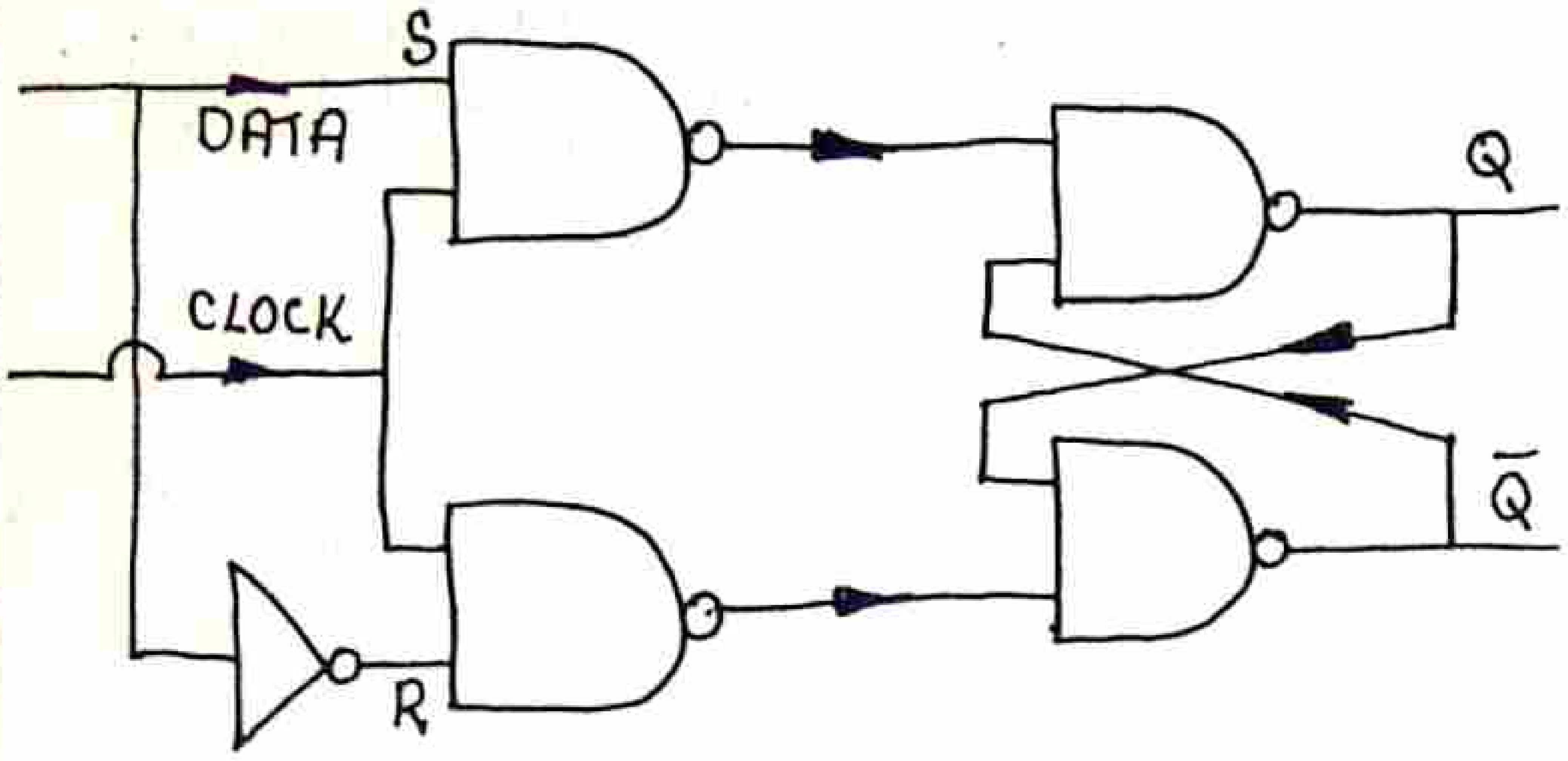


D-TYPE FLIP FLOP.

THE INFORMATION IS PRESENTED AT THE D INPUT. THE CLOCK LINE TELLS THE FLIPFLOP WHEN TO CHANGE STATE. IT WORKS LIKE THIS:

FOR THIS DISCUSSION THE DATA CAN BE HIGH OR LOW. WHEN THE CLOCK LINE IS LOW, THE DATA SITS AT THE INPUT AWAITING ACCEPTANCE. WHEN THE CLOCK LINE GOES HIGH THEN LOW, THE INFORMATION ON THE DATA LINE IS LOCKED INTO THE FLIP FLOP. IF THE DATA IS HIGH THE FLIP FLOP IS "SET" IF THE DATA IS LOW THE FLIP FLOP IS "RESET". THIS RESULT IS AVAILABLE AT THE Q OUTPUT AND REMAINS IN THE FLIP FLOP UNTIL CHANGED BY A DIFFERENT DATA STATE & ONE CLOCK CYCLE.

D FLIP FLOP WITH NAND GATES

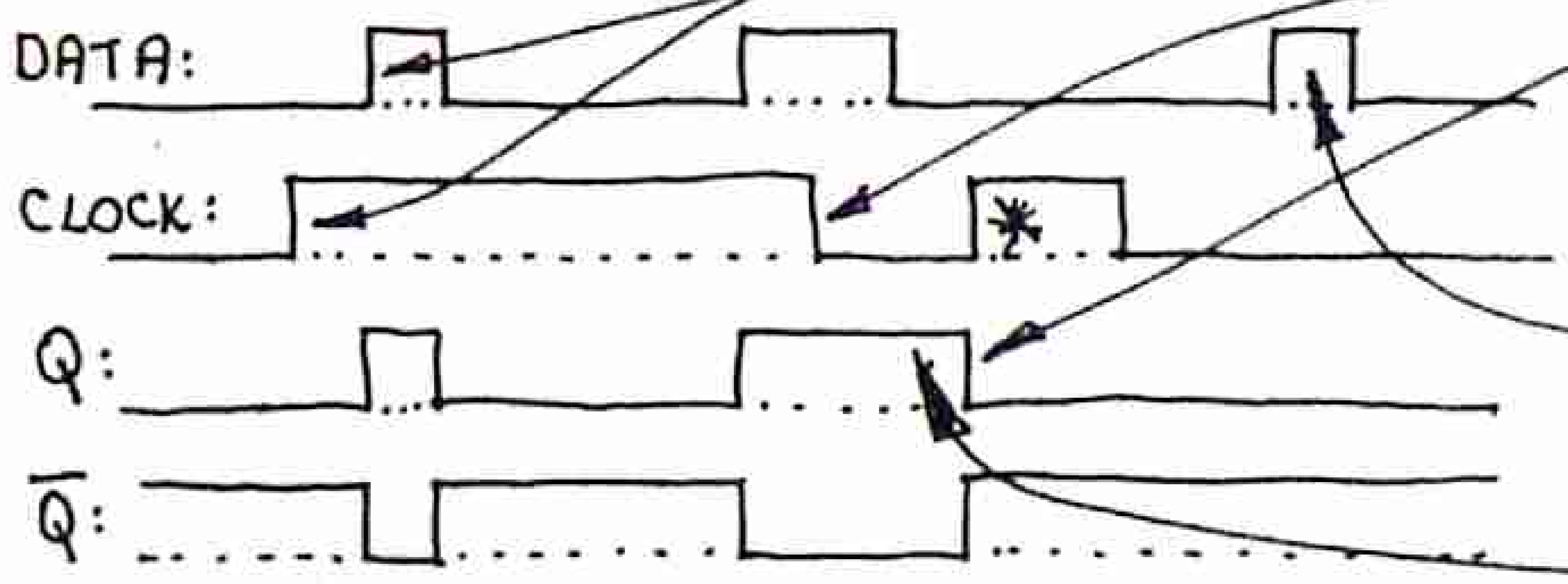


INPUTS		OUTPUTS	
DATA	CLOCK	Q	$\overline{Q}$
0	0	X	$\overline{X}$
1	0	X	$\overline{X}$
0	1	0	1
1	1	1	0

NAND D FLIP-FLOP TRUTH TABLE.

DATA IS ACCEPTED WHEN THE CLOCK LINE IS HIGH.

WAVE-FORMS:



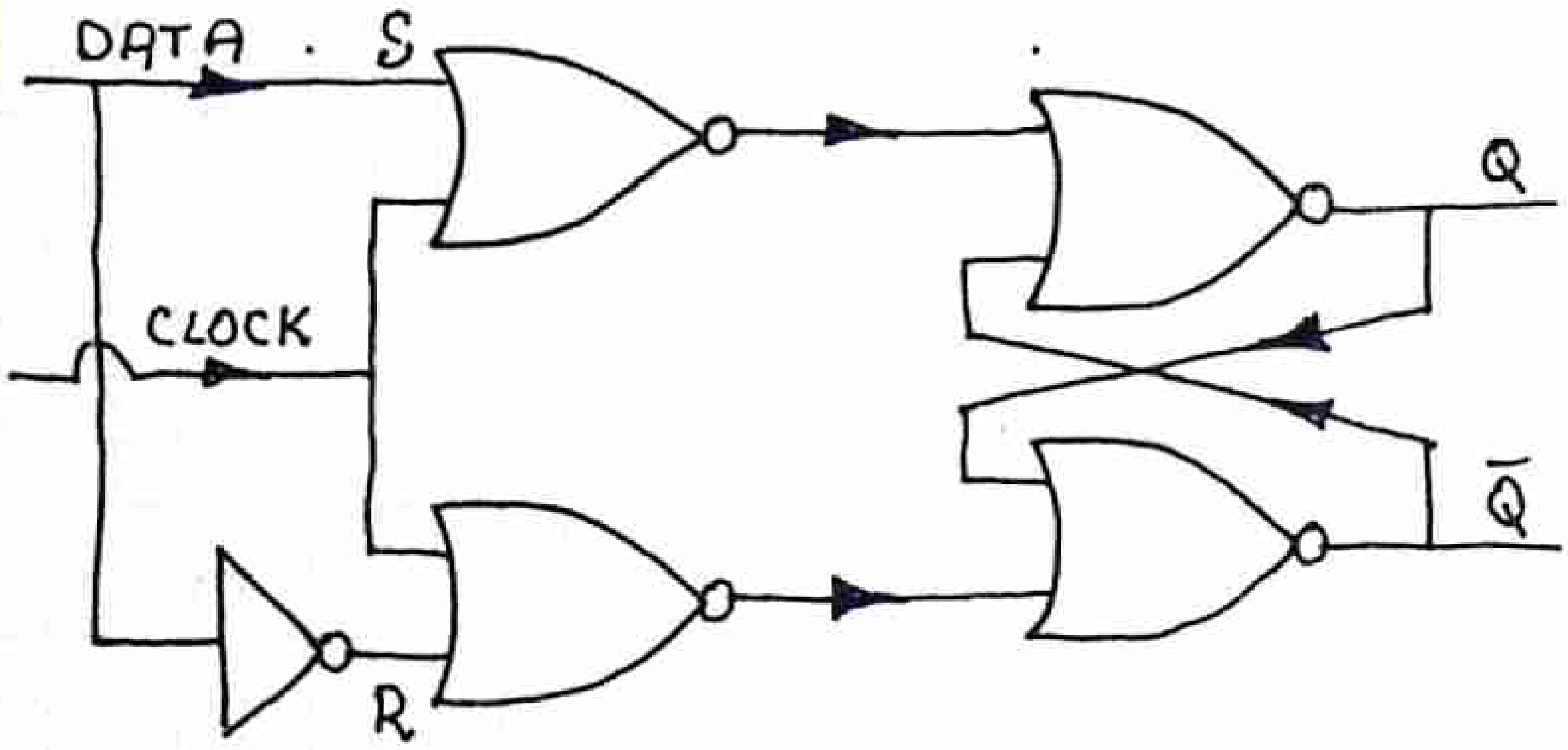
THE CLOCK LINE HAS OPENED THE FLIP-FLOP FOR THE DATA PULSE.

THE FLIP FLOP IS STILL OPEN FOR THIS DATA PULSE & LOCKS THE DATA HIGH UNTIL THE NEXT CLOCK PULSE MARKED* OPENS THE F-F & RETURNS Q LOW.

THIS PULSE IS NOT RECOGNISED AS THE CLOCK LINE IS LOW.

THIS IS WHERE MEMORY OCCURED

D FLIP FLOP WITH NOR GATES

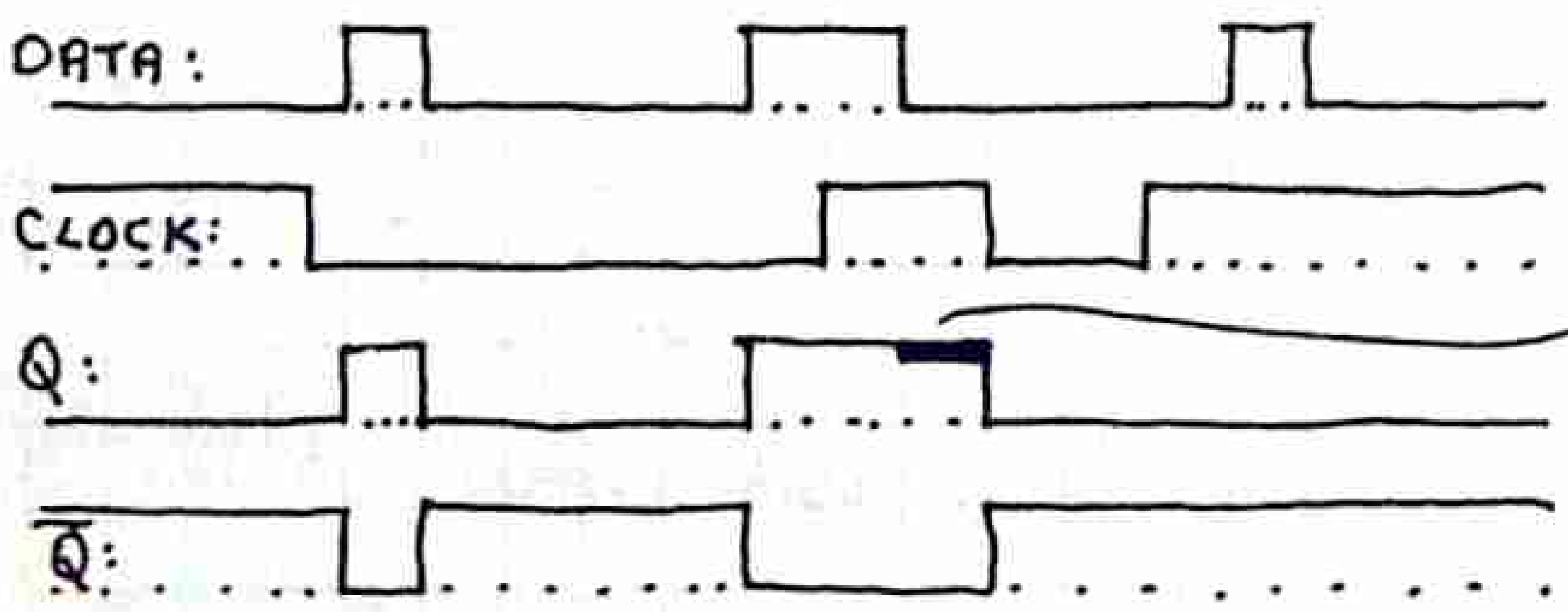


INPUTS		OUTPUTS	
DATA	CLOCK	Q	$\overline{Q}$
0	0	0	1
1	0	1	0
0	1	X	$\overline{X}$
1	1	X	$\overline{X}$

TRUTH TABLE NOR D-FLIP FLOP

DATA IS ACCEPTED WHEN THE CLOCK LINE IS LOW

WAVEFORMS:

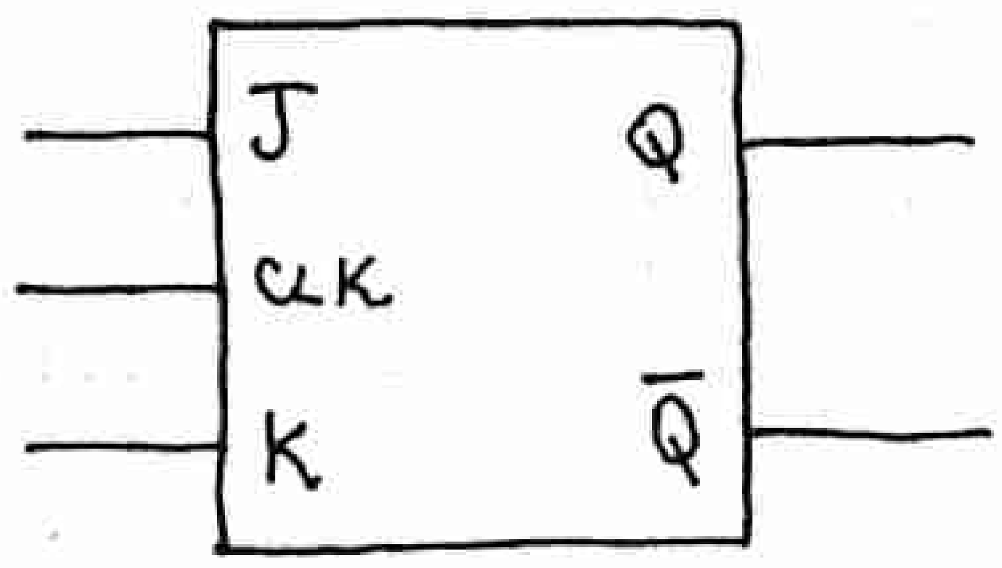


THIS IS THE TIME INTERVAL WHERE (WHEN) THE LATCH STORED THE DATA "HIGH". THE OUTPUT IS OF LONGER DURATION THAN THE INPUT PULSE, INDICATING MEMORY.

THE J-K FLIP FLOP

THE NEXT IMPROVEMENT IN THE FLIP FLOP FAMILY IS TO HAVE MORE CONTROL OVER THE OPERATION OF THE FLIP FLOP ITSELF. ONE IMPORTANT REQUIREMENT IN SOME CIRCUITS IS TO BE ABLE TO PRESET A FLIP FLOP OR "LOAD" IT WITH A VALUE, BEFORE THE COMMENCEMENT OF A CYCLE. ON THE OTHER HAND IT MAY BE NECESSARY TO CLEAR THE FLIP FLOP BEFORE STARTING AN OPERATION.

THESE FEATURES REQUIRE MORE CONTROL LINES & A MORE COMPLEX LATCH IS REQUIRED. FOR THIS A J-K FLIP FLOP HAS BEEN PRODUCED. (THE LETTERS J & K ARE MERELY THE FIRST AVAILABLE LETTERS IN THE ALPHABET.) IT IS ACTUALLY 2 FLIP FLOPS IN ONE. THE FIRST FLIP FLOP IS CALLED THE MASTER & THE OTHER IS CALLED THE SLAVE. THE MASTER CONTROLS THE SLAVE & WE READ THE VALUE OF THE FLIP FLOP FROM THE SLAVE.



J-K FLIP FLOP

THIS SIMPLE BLOCK HIDES A LOT OF COMPLEX CIRCUITRY.—THATS THE ADVANTAGE OF BLOCK DIAGRAMS — THEY MAKE DIGITAL ELECTRONICS LOOK SO SIMPLE.

THE OPERATION OF THE JK FLIP FLOP IS QUITE COMPLEX, AS IT IS CAPABLE OF PERFORMING A NUMBER OF DIFFERENT OPERATIONS, DEPENDING ON THE STATE OF THE INPUT LINES

THE MOST IMPORTANT FEATURE OF THE JK FLIP FLOP IS THE PREDICTABILITY OF THE OUTPUTS. THEY WILL ALWAYS BE COMPLEMENTARY.

THE CLOCK LINE CONTROLS WHEN THE FLIP FLOP WILL PROCESS THE INCOMING SIGNALS & THE J&K LINES PRODUCE 4 DIFFERENT EFFECTS AT THE OUTPUT.
THIS IS A SUMMARY OF THE EFFECTS:

1. IF THE J&K INPUTS ARE LOW — THE FLIP FLOP WILL FREEZE.
2. IF J IS LOW & K IS HIGH —— THE FLIP FLOP WILL RESET
3. IF J IS HIGH & K IS LOW —— THE FLIP FLOP WILL SET
4. IF BOTH HIGH. —— THE FLIP FLOP WILL TOGGLE.

ASSOCIATED WITH 4 IS A VERY INTERESTING FEATURE

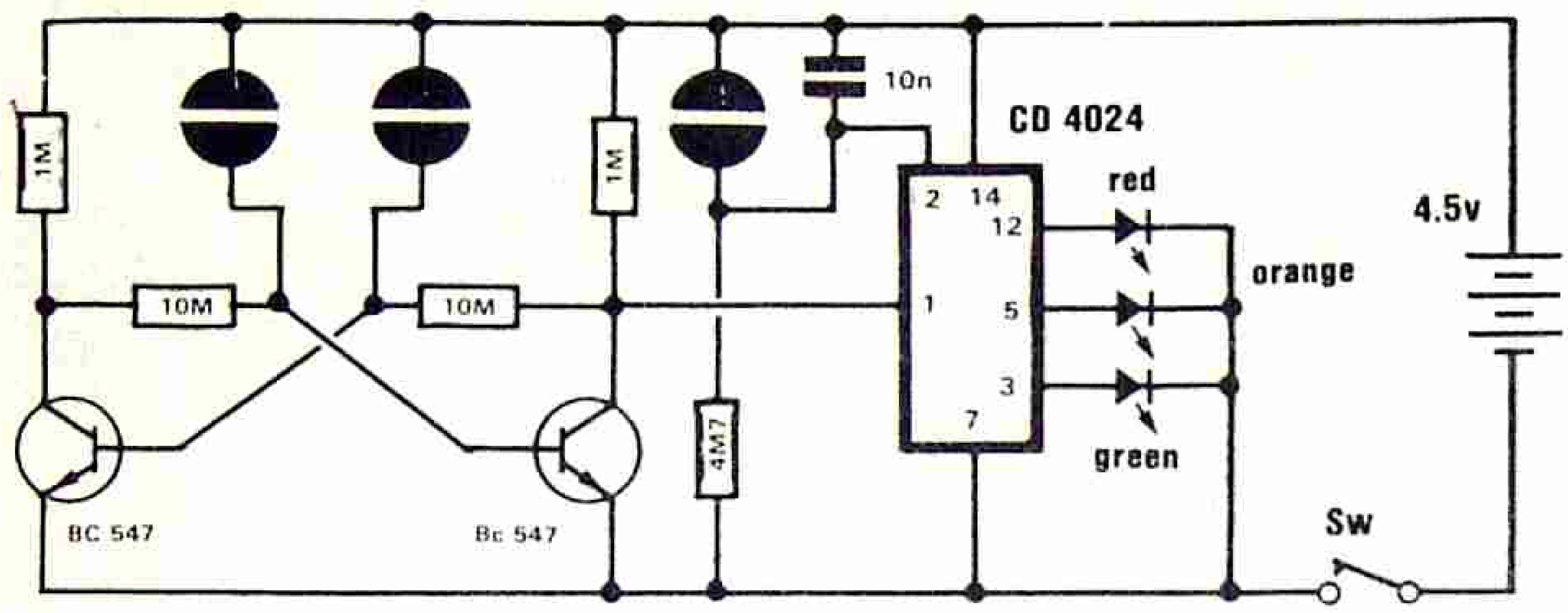
...... SO WE WILL START AT THE BEGINNING.

. . . continued next issue. You can read the complete course in **DIGITAL ELECTRONICS REVEALED.**

TOUCH PUZZLE

'..an ideal learning project....'

PC board: $2.10
Parts: $4.15
Complete: $6.25



COMPLETE TOUCH PUZZLE CIRCUIT

This project looks simple yet covers two very important aspects of digital electronics.

The TOUCH PUZZLE is basically composed of two sections. A transistor bistable switch, and a 4024 binary counter. These combine to form a simple counting circuit in which the two-transistor switch is designed to amplify the effect of your finger touching a set of pads and create a fast waveshape with sufficient amplitude to clock a CMOS chip.

We will investigate 4 features of the circuit and expand its capability to create a 'key-less' lock or a puzzle using 3 or more switches.

The project introduces the following:
1. Counting,
2. High-speed switching,
3. Auto reset.

Binary counters can be wired in two ways. 1. **Continuous counting** in which the chip cycles through all its counting stages, and: **2. Count and reset,** in which the chip counts to a specified number then resets.

The most important aspect of any arrangement is the value assigned to each output.

Most often they are labelled Q1, Q2, Q3, Q4 etc however they could equally be called Q0, Q1, Q2, Q3, depending on the manufacturer. The actual division value or counting value will depend on the mode of operation as discussed above.

This project will show how the output value will depend on the circuit arrangement.

ASSEMBLY

For this project the PC tracks run on the top of the board and the components solder directly to them without the need for any holes.

An IC socket is used for the 4024 so that it can be placed in another circuit if needed.

The remainder of the parts are low-cost items and are not worth saving. However any of them can be de-soldered and reused if needed.

The drawings and overlays should be sufficient for you to construct the PUZZLE. If a problem arises, refer to the circuit diagram above.

Keep the parts off the board and yet not so high that they get in the way when the project is being used.

Two bridges are required to jump over the tracks and these are made from resistors leads.

The three coloured LEDs are mounted so that the red LED is near the 10n capacitor, the yellow LED is in the middle and then the green LED.

The leads on some of the LEDs must be parted slightly to fit onto the lands. Do this very carefully to avoid bending the leads inside the LED.

This project will be a good test of soldering. LEDs can be easily damaged with excess heat so use your fingers to prevent them getting too hot.

Carefully check the difference between the 1M and 10M resistors and mount them as shown in the diagram.

PARTS LIST

2 - 1M
1 - 4M7
2 - 10M

1 - 10n greencap
2 - 47n for experimenting

1 - 4024 IC
1 - 14 pin IC socket

1 - 3mm red LED
1 - 3mm orange LED
1 - 3mm green LED

1 - slide switch

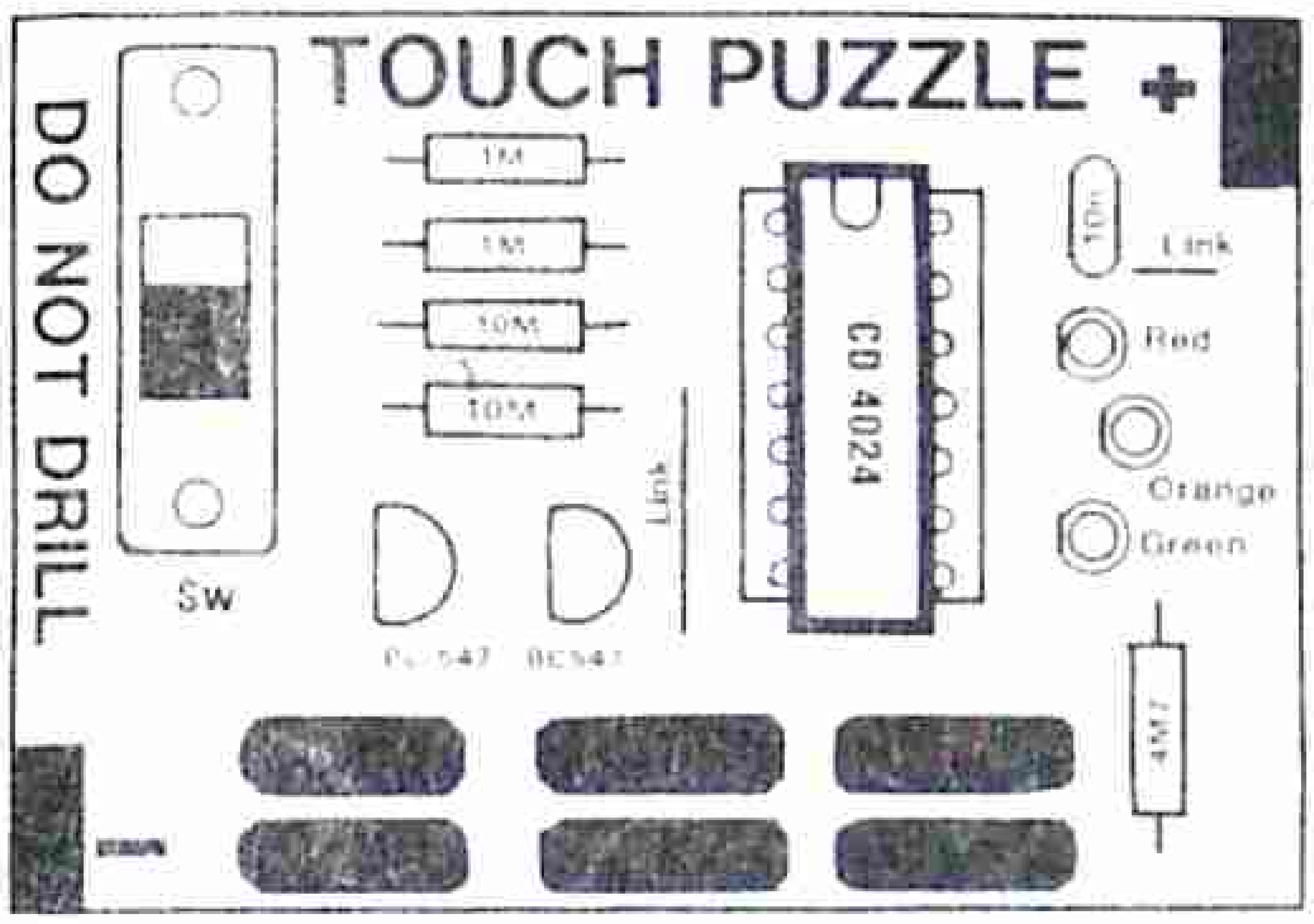
3 - AAA cells

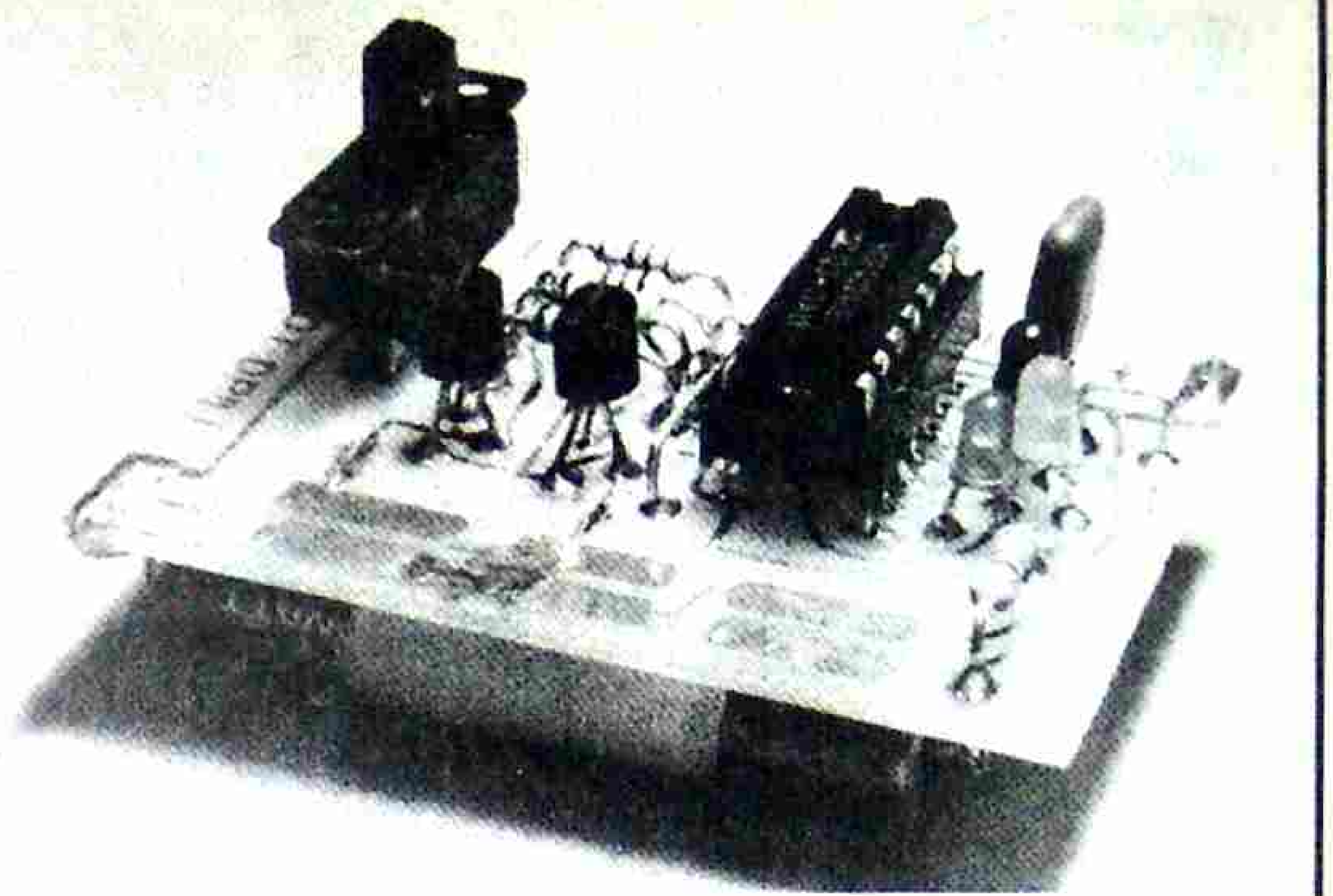
TOUCH PUZZLE PC BOARD

Solder the IC socket so that pin 1 is near the letters 'Puzzle'.

The balance of the components are neatly soldered to the board and this will emphasize the need for a fine tipped soldering iron.

The three cells are soldered together with tinned copper wire and taped to form a battery pack. Sand-paper the terminals first to help them take solder. The leads are then soldered directly to the PC board, fit the IC and the project is complete.





The PC board has all the components soldered on the copper side. This allows all the circuit to be seen at the one time.

OPERATION

The operation of the TOUCH PUZZLE can be tested with the TE LOGIC PROBE. This project has already been covered and you should have already constructed it by now.

All the sections of the Puzzle are HIGH IMPEDANCE and this means they cannot be successfully tested with an ordinary multimeter.

The reason for this is a multimeter 'loads' any circuit it is testing and requires about 30 microamp to swing the needle. Our project consumes only about 5 microamp and this is not sufficient to move the needle.

To test the circuit we need a high impedance tester such as a logic probe. And the TE Logic Probe is ideal.

Clip the leads of the probe to the 4.5v battery (to power the probe) and turn the Puzzle on.

Probe the positive and negative terminals of the battery and you will be able to see the HIGH and LOW on the probe.

Next probe the 6 TOUCH PADS. You will detect a HIGH on the three pads linked to the positive rail - this is obvious.

The other 'A' pad presents a LOW as does the second 'B' pad. This is because they are connected to the base of the transistors and they cannot go higher than .6v - effectively a LOW.

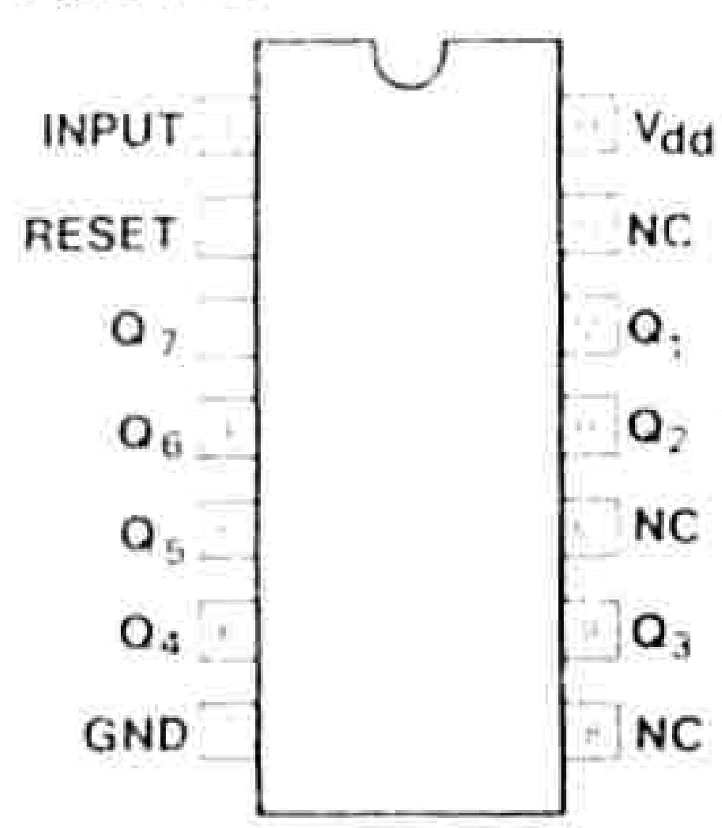
As you probe back and forth between active pads of A and B, you will be able to clock the Puzzle. This is due to the sensitivity of the transistors detecting the logic probe and causing the circuit to switch.

The outputs of the IC can be probed for HIGHs and LOWs while the chip is being clocked.

This will identify which pins are output pins and which are **RAIL** and **RESET.**

Once you have established the output pins, compare your results with the pinout diagram below.

CD4024 B



7-BIT BINARY COUNTER

The reset line is rather interesting. It is being kept LOW by a 4M7 and is effectively in a very high impedance state. It may show a definite low on the probe or be just above earth, depending on the leakage within the IC or the effect of the probe. Clock the chip a number of times and record the effect on the probe. This will vary with different chips and probes as it is only a leakage or 'loading' effect.

AUTO RESET

Every time the Puzzle is turned on, an AUTO RESET circuit resets the flip flops inside the 4024. This is necessary to make sure the chip starts at zero, otherwise some LEDs may come on when the power is applied and create a false count.

The two components responsible for the **AUTO RESET** are the 10n capacitor and 4M7 resistor.

The 10n capacitor is initially in an uncharged condition and when the power is applied, it pulls the reset line to the positive rail.

This resets all the flip flops. The capacitor gradually charges via the 4M7 and within a short time the reset line is LOW and counting can commence.

The 4M7 keeps the capacitor charged while the power is applied.

When the circuit is switched off, the capacitor discharges via the 4M7 and the other components in the circuit, ready for the next switch-on.

THE OUTPUTS

Each of the outputs of a BINARY COUNTER represent a value of division.

The lowest value output represents a division of 2, the next a division of 4, then a division of 8 etc etc.

We can see this division on the TOUCH PUZZLE. The red LED comes on only every second time we touch pad B. This means it is dividing the pulses by two.

Place your finger back and forth from Pad A to B, to A, to B and the red LED will come on ONCE.

Each successive output will halve the repetition rate and IC's containing 12, 14 and even 24 stages are available. Our chip contains 7 stages and makes it easy to count a full cycle without having to wait for hours.

EXPERIMENTS

1. The first modification for the TOUCH PUZZLE involves turning the bistable switching circuit into an astable arrangement.

Solder a 47n capacitor across each of the two 10M resistors.

By placing your finger carefully so that it covers all 4 pads A and B, the circuit will clock slowly and this can be observed on the LEDs.

You have produced a simple astable multivibrator with your finger providing the turn-on voltage for the two transistors.

2. Construct a self-oscillating astable multivibrator by removing the two 10M resistors and placing them across pads A and B as shown in the diagram below:

Switch the puzzle on and the LEDs will flash automatically.

3. Next count the number of flashes for one complete cycle of the IC. The slow oscillator will make this easy to do.

You would think you would get the value 128, but in fact you will get 64.

The reason for this is the red LED represents the value 2 every time it is lit. This can be proven by connecting the logic probe and probing the input pin. You will see the LED on the probe flashing at a rapid rate. But this also put a load on the circuit and the 4024 chip will not be able to count. To overcome this, probe the other 1M resistor and compare the flash rate of the green LED on the probe with the red LED on the puzzle.

Repeat the counting sequence by saying "2, 4, 6, 8, 10 etc. for each flash of the red LED.

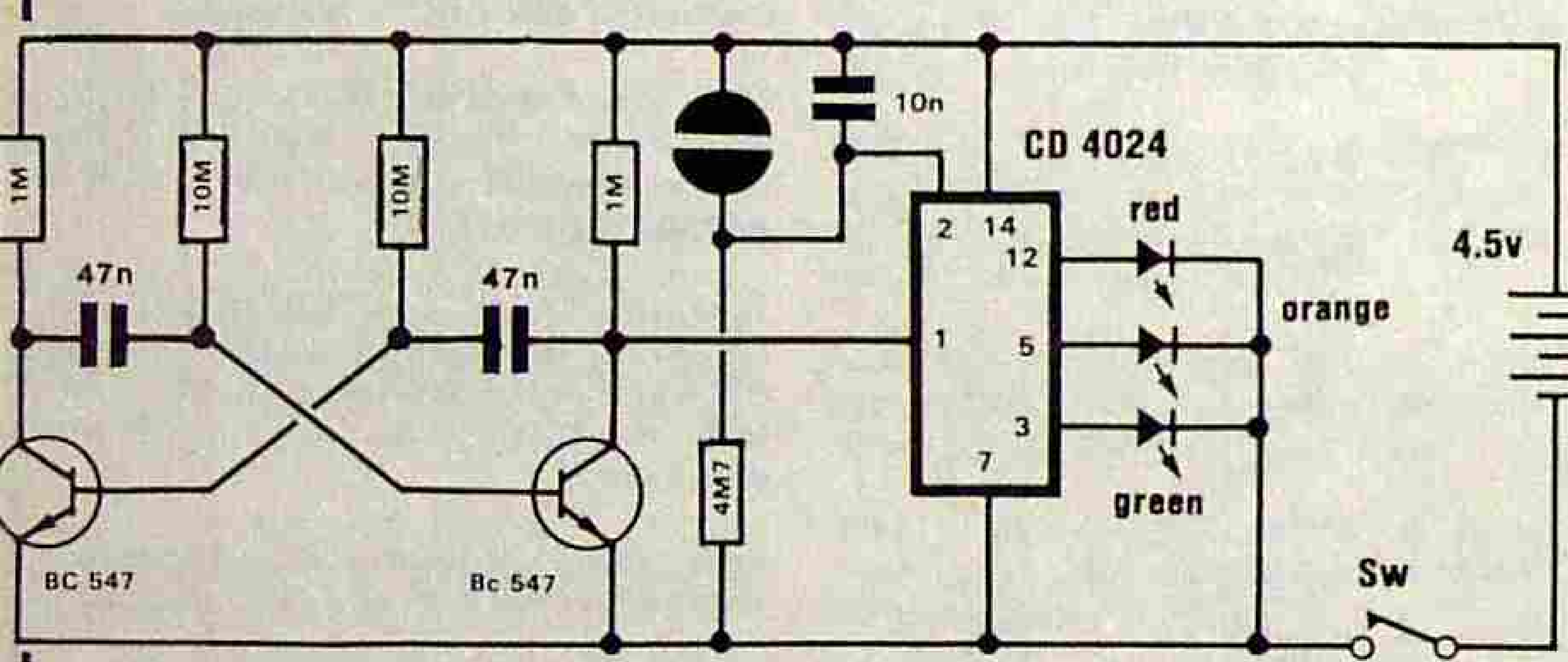
HIGH-SPEED SWITCHING

All digital circuits require fast wave-forms on their input line for correct clocking.

Normal mechanical switches are not suitable for this purpose due to the noise they introduce as the contacts open and close.

The same applies to TOUCH SWITCHES. But if a touch switch is connected via a transistor switch, the effect of touching the plate will be speeded up by an action called REGENERATION. This is covered in Notebook 2, P38 and 39.

Briefly it states that the transistors are speeding up the slow effect of your finger touching the PC pad and converts this into a fast rise-time.



The transistor section converted into an astable multivibrator to show the output pattern of the counter. The clock rate is slow enough for you to be able to count the full cycle of the 4024 and answer questions in the text.

RISING AND FALLING EDGE

Chips can be designed to clock on the rising edge of a waveform or the falling edge.

We are going to find out the characteristic of the 4024.

With the puzzle in its AUTO COUNT MODE, determine at what instant the orange LED turns on.

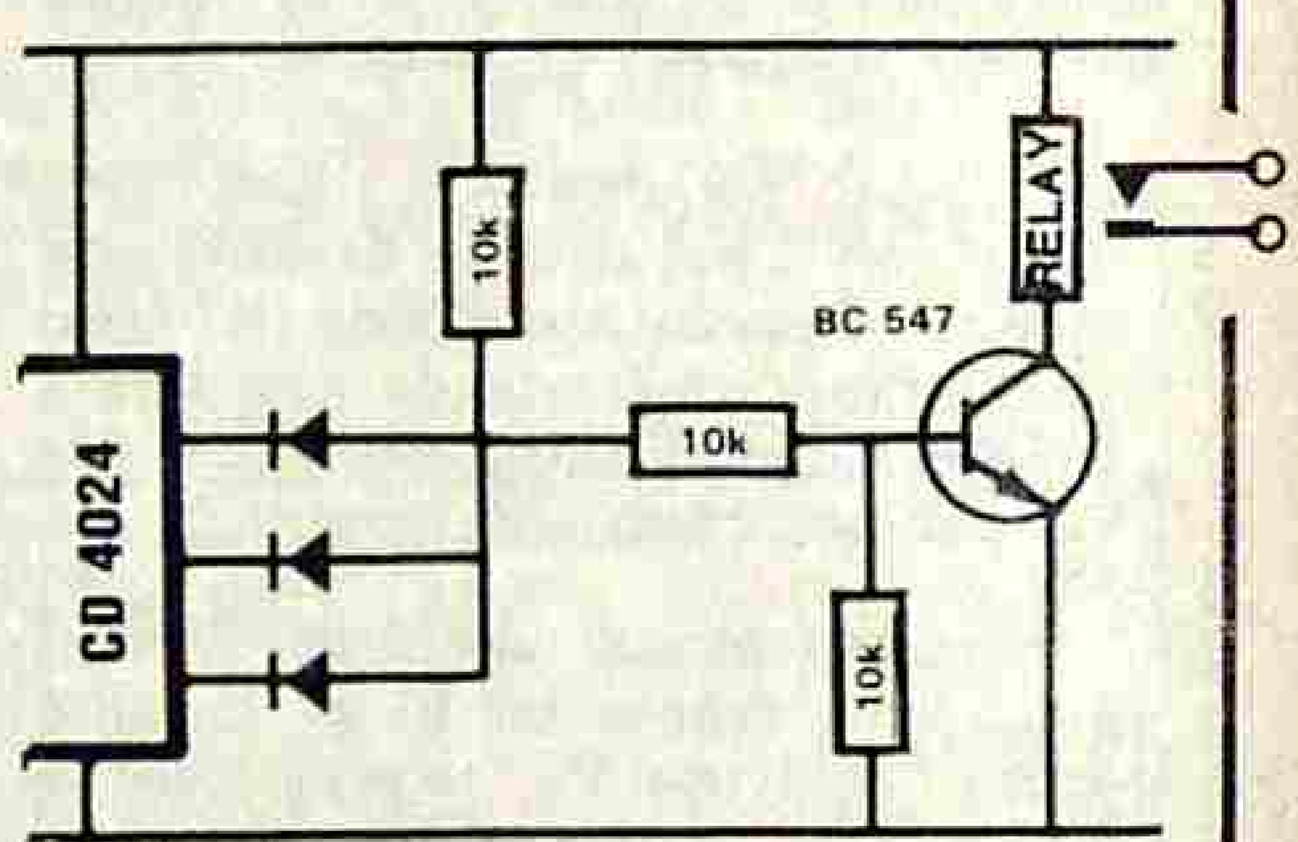
Connect the probe to the first transistor in flip-flop. Watch the green LED on the probe as it will represent a HIGH to the input of the 4024. You should see it switch OFF when the red LED on the puzzle turns ON. This indicates the chip is advancing during the LOW transition of the clock pulse.

FURTHER IDEAS

To turn the TOUCH PUZZLE into a challenge, remove all but the highest priority LED and ask your partner to 'TURN ON THE LED'.

The output could also be connected to a transistor (via a 10k resistor) and wired to operate a relay. It could then be used as a delay, lock, puzzle, game of speed or any type of project requiring counting.

The outputs can be decoded (see DIGITAL ELECTRONICS REVEALED P. 39.) to operate at any value between 2 and 64. This is done by forming an AND gate and coupling the output to a buffer as shown in the diagram:



This concludes our puzzle project however here are some questions which you should be able to answer:

QUESTIONS

1. What is the voltage on the project?
1(a) Why have we chosen this voltage?
2. What is the value of idle current?
2(a) How can this be measured without removing any leads?
3. Why is the idle current so low?
4. Why don't we need an electrolytic across the battery?
5. What is the continuous counting division for a 4024?
6. Is the 4024 RISING-EDGE or FALLING-EDGE triggered?

Z80 Machine Codes FOR DISASSEMBLY

SHEET 12

This is a Z-80 MACHINE CODE disassembly table. Use it in conjunction with the Z-80 Machine Codes presented previously, for the creation of your own programs.

These lists make programming and disassembly easy. Fit them into a plastic sleeve and keep them handy.

00	NOP
01	LD BC,dddd
02	LD (BC),A
03	INC BC
04	INC B
05	DEC B
06	LD B,dd
07	RLCA
08	EX AF,A'F'
09	ADD HL,BC
0A	LD A,(BC)
0B	DEC BC
0C	INC C
0D	DEC C
0E	LD C,dd
0F	RRCA
10	DJNZ dis
11	LD DE,dddd
12	LD (DE),A
13	INC DE
14	INC D
15	DEC D
16	LD D,dd
17	RLA
18	JR dis
19	ADD HL,DE
1A	LD A,(DE)
1B	DEC DE
1C	INC E
1D	DEC E
1E	LD E,dd
1F	RRA
20	JR NZ dis
21	LD HL,dddd
22	LD (ADDR),HL
23	INC HL
24	INC H
25	DEC H
26	LD H,dd
27	DAA
28	JR Z,dis
29	ADD HL,HL
2A	LD HL,(ADDR)
2B	DEC HL
2C	INC L
2D	DEC L
2E	LD L,dd
2F	CPL
30	JR NC,dis
31	LD SP,dddd
32	LD (ADDR),A
33	INC SP
34	INC (HL)
35	DEC (HL)
36	LD (HL),dd
37	SCF
38	JR C,dis
39	ADD HL,SP
3A	LD A,(ADDR)
3B	DEC SP
3C	INC A
3D	DEC A
3E	LD A,dd
3F	CCF
40	LD B,B
41	LD B,C
42	LD B,D
43	LD B,E
44	LD B,H
45	LD B,L
46	LD B,(HL)
47	LD, B,A
48	LD C,B
49	LD C,C
4A	LD C,D
4B	LD C,E
4C	LD C,H
4D	LD C,L
4E	LD C,(HL)
4F	LD C,A
50	LD D,B
51	LD D,C
52	LD D,D
53	LD D,E
54	LD D,H
55	LD D,L
56	LD D,(HL)
57	LD D,A
58	LD E,B
59	LD E,C
5A	LD E,D
5B	LD E,E
5C	LD E,H
5D	LD E,L
5E	LD E,(HL)
5F	LD E,A
60	LD H,B
61	LD H,C
62	LD H,D
63	LD H,E
64	LD H,H
65	LD H,L
66	LD H,(HL)
67	LD H,A
68	LD L,B
69	LD L,C
6A	LD L,D
6B	LD L,E
6C	LD L,H
6D	LD L,L
6E	LD L,(HL)
6F	LD L,A
70	LD (HL),B
71	LD (HL),C
72	LD (HL),D
73	LD (HL),E
74	LD (HL),H
75	LD (HL),L
76	HALT
77	LD (HL),A
78	LD A,B
79	LD A,C
7A	LD A,D
7B	LD A,E
7C	LD A,H
7D	LD A,L
7E	LD A,(HL)
7F	LD A,A
80	ADD A,B
81	ADD A,C
82	ADD A,D
83	ADD A,E
84	ADD A,H
85	ADD A,L
86	ADD A,(HL)
87	ADD A,A
88	ADC A,B
89	ADC A,C
8A	ADC A,D
8B	ADC A,E
8C	ADC A,H
8D	ADC A,L
8E	ADC A,(HL)
8F	ADC A,A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
96	SUB (HL)
97	SUB A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
9E	SBC A,(HL)
9F	SBC A,A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L
A6	AND (HL)
A7	AND A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
AE	XOR (HL)
AF	XOR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
B6	OR (HL)
B7	OR A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H
BD	CP L
BE	CP (HL)
BF	CP A
C0	RET NZ
C1	POP BC
C2	JP NZ ADDR
C3	JP ADDR
C4	CALL NZ ADDR
C5	PUSH BC
C6	ADD A,dd
C7	RST 00
C8	RET Z
C9	RET
CA	JP Z ADDR
CB	★
CC	CALL Z,ADDR
CD	CALL ADDR
CE	ADC A,dd
CF	RST 08
D0	RET NC
D1	POP DE
D2	JP NC ADDR
D3	OUT port,A
D4	CALL NC ADDR
D5	PUSH DE
D6	SUB dd
D7	RST 10
D8	RET C
D9	EXX
DA	JP C ADDR
DB	IN A,port
DC	CALL C ADDR
DD	★
DE	SBC A,dd
DF	RST 18
E0	RET PO
E1	POP HL
E2	JP PO ADDR
E3	EX (SP),HL
E4	CALL PO ADDR
E5	PUSH HL
E6	AND dd
E7	RST 20
E8	RET PE
E9	JP (HL)
EA	JP PE ADDR
EB	EX DE,HL
EC	CALL PE ADDR
ED	★
EE	XOR dd
EF	RST 28
F0	RET P
F1	POP AF
F2	JP P ADDR
F3	DI
F4	CALL P ADDR
F5	PUSH AF
F6	OR dd
F7	RST 30
F8	RET M
F9	LD SP,HL
FA	JP M ADDR
FB	EI
FC	CALL M ADDR
FD	★
FE	CP dd
FF	RST 38
CB 00	RLC B
CB 01	RLC C
CB 02	RLC D
CB 03	RLC E
CB 04	RLC H
CB 05	RLC L
CB 06	RLC (HL)
CB 07	RLC A
CB 08	RRC B
CB 09	RRC C
CB 0A	RRC D
CB 0B	RRC E
CB 0C	RRC H
CB 0D	RRC L
CB 0E	RRC (HL)
CB 0F	RRC A
CB 10	RL B
CB 11	RL C
CB 12	RL D
CB 13	RL E
CB 14	RL H
CB 15	RL L
CB 16	RL (HL)
CB 17	RL A
CB 18	RR B
CB 19	RR C
CB 1A	RR D
CB 1B	RR E
CB 1C	RR H
CB 1D	RR L
CB 1E	RR (HL)
CB 1F	RR A
CB 20	SLA B
CB 21	SLA C
CB 22	SLA D
CB 23	SLA E
CB 24	SLA H
CB 25	SLA L
CB 26	SLA (HL)
CB 27	SLA A
CB 28	SRA B
CB 29	SRA C
CB 2A	SRA D
CB 2B	SRA E
CB 2C	SRA H
CB 2D	SRA L
CB 2E	SRA (HL)
CB 2F	SRA A
CB 38	SRL B
CB 39	SRL C
CB 3A	SRL D
CB 3B	SRL E
CB 3C	SRL H
CB 3D	SRL L
CB 3E	SRL (HL)
CB 3F	SRL A
CB 40	BIT 0,B
CB 41	BIT 0,C
CB 42	BIT 0,D
CB 43	BIT 0,E
CB 44	BIT 0,H
CB 45	BIT 0,L
CB 46	BIT 0,(HL)
CB 47	BIT 0 A
CB 48	BIT 1,B
CB 49	BIT 1,C
CB 4A	BIT 1,D
CB 4B	BIT 1,E
CB 4C	BIT 1,H
CB 4D	BIT 1,L
CB 4E	BIT 1,(HL)
CB 4F	BIT 1,A
CB 50	BIT 2,B
CB 51	BIT 2,C
CB 52	BIT 2,D
CB 53	BIT 2,E
CB 54	BIT 2,H
CB 55	BIT 2,L
CB 56	BIT 2,(HL)
CB 57	BIT 2,A
CB 58	BIT 3,B
CB 59	BIT 3,C
CB 5A	BIT 3,D
CB 5B	BIT 3,E
CB 5C	BIT 3,H
CB 5D	BIT 3,L
CB 5E	BIT 3,(HL)
CB 5F	BIT 3,A
CB 60	BIT 4,B
CB 61	BIT 4,C
CB 62	BIT 4,D
CB 63	BIT 4,E
CB 64	BIT 4,H
CB 65	BIT 4,L
CB 66	BIT 4,(HL)
CB 67	BIT 4,A
CB 68	BIT 5,B
CB 69	BIT 5,C
CB 6A	BIT 5,D
CB 6B	BIT 5,E
CB 6C	BIT 5,H
CB 6D	BIT 5,L
CB 6E	BIT 5,(HL)
CB 6F	BIT 5,A
CB 70	BIT 6,B
CB 71	BIT 6,C
CB 72	BIT 6,D
CB 73	BIT 6,E
CB 74	BIT 6,H
CB 75	BIT 6,L
CB 76	BIT 6,(HL)
CB 77	BIT 6,A
CB 78	BIT 7,B
CB 79	BIT 7,C
CB 7A	BIT 7,D
CB 7B	BIT 7,E
CB 7C	BIT 7,H
CB 7D	BIT 7,L
CB 7E	BIT 7,(HL)
CB 7F	BIT 7,A
CB 80	RES 0,B
CB 81	RES 0,C
CB 82	RES 0,D
CB 83	RES 0,E
CB 84	RES 0,H
CB 85	RES 0,L
CB 86	RES 0,(HL)
CB 87	RES 0,A
CB 88	RES 1,B
CB 89	RES 1,C
CB 8A	RES 1,D
CB 8B	RES 1,E
CB 8C	RES 1,H
CB 8D	RES 1,L
CB 8E	RES 1,(HL)
CB 8F	RES 1,A
CB 90	RES 2,B
CB 91	RES 2,C
CB 92	RES 2,D
CB 93	RES 2,E
CB 94	RES 2,H
CB 95	RES 2,L
CB 96	RES 2,(HL)
CB 97	RES 2,A
CB 98	RES 3,B
CB 99	RES 3,C
CB 9A	RES 3,D
CB 9B	RES 3,E
CB 9C	RES 3,H
CB 9D	RES 3,L
CB 9E	RES 3,(HL)
CB 9F	RES 3,A
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 4,L
CB A6	RES 4,(HL)
CB A7	RES 4,A
CB A8	RES 5,B
CB A9	RES 5,C
CB AA	RES 5,D
CB AB	RES 5,E
CB AC	RES 5,H
CB AD	RES 5,L
CB AE	RES 5,(HL)
CB AF	RES 5,A
CB B0	RES 6,B
CB B1	RES 6,C
CB B2	RES 6,D
CB B3	RES 6,E
CB B4	RES 6,H
CB B5	RES 6,L
CB B6	RES 6,(HL)
CB B7	RES 6,A
CB B8	RES 7,B
CB B9	RES 7,C
CB BA	RES 7,D
CB BB	RES 7,E
CB BC	RES 7,H
CB BD	RES 7,L
CB BE	RES 7,(HL)
CB BF	RES 7,A
CB C0	SET 0,B
CB C1	SET 0,C
CB C2	SET 0,D
CB C3	SET 0,E
CB C4	SET 0,H
CB C5	SET 0,L
CB C6	SET 0,(HL)
CB C7	SET 0,A
CB C8	SET 1,B
CB C9	SET 1,C
CB CA	SET 1,D
CB CB	SET 1,E
CB CC	SET 1,H
CB CD	SET 1,L
CB CE	SET 1,(HL)
CB CF	SET 1,A
CB D0	SET 2,B
CB D1	SET 2,C
CB D2	SET 2,D
CB D3	SET 2,E
CB D4	SET 2,H
CB D5	SET 2,L
CB D6	SET 2,(HL)
CB D7	SET 2,A
CB D8	SET 3,B
CB D9	SET 3,C
CB DA	SET 3,D
CB DB	SET 3,E
CB DC	SET 3,H
CB DD	SET 3,L
CB DE	SET 3,(HL)
CB DF	SET 3,A
CB E0	SET 4,B
CB E1	SET 4,C
CB E2	SET 4,D
CB E3	SET 4,E
CB E4	SET 4,H
CB E5	SET 4,L
CB E6	SET 4,(HL)
CB E7	SET 4,A
CB E8	SET 5,B
CB E9	SET 5,C
CB EA	SET 5,D
CB EB	SET 5,E
CB EC	SET 5,H
CB ED	SET 5,L
CB EE	SET 5,(HL)
CB EF	SET 5,A
CB F0	SET 6,B
CB F1	SET 6,C
CB F2	SET 6,D
CB F3	SET 6,E
CB F4	SET 6,H
CB F5	SET 6,L
CB F6	SET 6,(HL)
CB F7	SET 6,A
CB F8	SET 7,B
CB F9	SET 7,C
CB FA	SET 7,D
CB FB	SET 7,E
CB FC	SET 7,H
CB FD	SET 7,L
CB FE	SET 7,(HL)
CB FF	SET 7,A
DD 09	ADD IX,BC
DD 19	ADD IX,DE
DD 21	LD IX,dddd
DD 22	LD (ADDR),IX
DD 23	INC IX
DD 29	ADD IX,IX
DD 2A	LD IX,(ADDR)
DD 2B	DEC IX
DD 34	INC,(IX + dis)
DD 35	DEC,(IX + dis)
DD 36	LD (IX + dis)dd
DD 39	ADD IX,SP
DD 46	LD B,(IX + dis)
DD 4F	LD C,(IX + dis)
DD 56	LD D,(IX + dis)
DD 5E	LD E,(IX + dis)
DD 66	LD H,(IX + dis)
DD 6E	LD L,(IX + dis)
DD 70	LD (IX + dis),B
DD 71	LD (IX + dis),C
DD 72	LD (IX + dis),D
DD 73	LD (IX + dis),E
DD 74	LD (IX + dis),H
DD 75	LD (IX + dis),L
DD 77	LD (IX + dis),A
DD 7E	LD A,(IX + dis)
DD 86	ADD A,(IX + dis)
DD 8E	ADC A,(IX + dis)
DD 96	SUB (IX + dis)
DD 9E	SBC A,(IX + dis)
DD A6	AND (IX + dis)
DD AE	XOR (IX + dis)
DD B6	OR (IX + dis)
DD BE	CP (IX + dis)
DD CB XX 06	RLC (IX + dis)
DD CB XX 0E	RRC (IX + dis)
DD CB XX 16	RL (IX + dis)
DD CB XX 1E	RR (IX + dis)
DD CB XX 26	SLA (IX + dis)
DD CB XX 2E	SRA (IX + dis)
DD CB XX 3E	SRL (IX + dis)
DD CB XX 46	BIT 0,(IX + dis)
DD CB XX 4E	BIT 1,(IX + dis)
DD CB XX 56	BIT 2,(IX + dis)
DD CB XX 5E	BIT 3,(IX + dis)
DD CB XX 66	BIT 4,(IX + dis)
DD CB XX 6E	BIT 5,(IX + dis)
DD CB XX 76	BIT 6,(IX + dis)
DD CB XX 7E	BIT 7,(IX + dis)
DD CB XX 86	RES 0,(IX + dis)
DD CB XX 8E	RES 1,(IX + dis)
DD CB XX 96	RES 2,(IX + dis)
DD CB XX 9E	RES 3,(IX + dis)
DD CB XX A6	RES 4,(IX + dis)
DD CB XX AE	RES 5,(IX + dis)
DD CB XX B6	RES 6,(IX + dis)
DD CB XX BE	RES 7,(IX + dis)
DD CB XX C6	SET 0,(IX + dis)
DD CB XX CE	SET 1,(IX + dis)
DD CB XX D6	SET 2,(IX + dis)
DD CB XX DE	SET 3,(IX + dis)
DD CB XX E6	SET 4,(IX + dis)
DD CB XX EE	SET 5,(IX + dis)
DD CB XX F6	SET 6,(IX + dis)
DD CB XX FE	SET 7,(IX + dis)
DD E1	POP IX
DD E3	EX (SP),IX
DD E5	PUSH IX
DD E9	JP (IX)
DD F9	LD SP,IX
ED 40	IN B,(C)
ED 41	OUT (C),B
ED 42	SBC HL,BC
ED 43	LD (ADDR),BC
ED 44	NEG
ED 45	RETN
ED 46	IM 0
ED 47	LD I,A
ED 48	IN C,(C)
ED 49	OUT (C),C
ED 4A	ADC HL,BC
ED 4B	LD BC,(ADDR)
ED 4D	RETI
ED 4F	LD R,A
ED 50	IN D,(C)
ED 51	OUT (C),D
ED 52	SBC HL,DE
ED 53	LD (ADDR),DE
ED 56	IM 1
ED 57	LD A,I
ED 58	IN E,(C)
ED 59	OUT (C),E
ED 5A	ADC HL,DE
ED 5B	LD DE,(ADDR)
ED 5E	IM 2
ED 5F	LD A,R
ED 60	IN H,(C)
ED 61	OUT (C),H
ED 62	SBC HL,HL
ED 63	LD (ADDR),HL
ED 67	RRD
ED 68	IN L,(C)
ED 69	OUT (C),L
ED 6A	ADC HL,HL
ED 6B	LD HL,(ADDR)
ED 6F	RLD
ED 72	SBC HL,SP
ED 73	LD (ADDR),SP
ED 78	IN A,(C)
ED 79	OUT (C),A
ED 7A	ADC HL,SP
ED 7B	LD SP,(ADDR)
ED A0	LDI
ED A1	CPI
ED A2	INI
ED A3	OUTI
ED A8	LDD
ED A9	CPD
ED AA	IND
ED AB	OUTD
ED B0	LDIR
ED B1	CPIR
ED B2	INIR
ED B3	OTIR
ED B8	LDDR
ED B9	CPDR
ED BA	INDR
ED BB	OTDR
FD 09	ADD IY,BC
FD 19	ADD IY,DE
FD 21	LD IY,dddd
FD 22	LD (ADDR),IY
FD 23	INC IY
FD 29	ADD IY,IY
FD 2A	LD IY,(ADDR)
FD 2B	DEC IY
FD 34	INC,(IY + dis)
FD 35	DEC,(IY + dis)
FD 36	LD (IY + dis)dd
FD 39	ADD IY,SP
FD 46	LD B,(IY + dis)
FD 4E	LD C,(IY + dis)
FD 56	LD D,(IY + dis)
FD 5E	LD E,(IY + dis)
FD 66	LD H,(IY + dis)
FD 6E	LD L,(IY + dis)
FD 70	LD (IY + dis),B
FD 71	LD (IY + dis),C
FD 72	LD (IY + dis),D
FD 73	LD (IY + dis),E
FD 74	LD (IY + dis),H
FD 75	LD (IY + dis),L
FD 77	LD (IY + dis),A
FD 7E	LD A,(IY + dis)
FD 86	ADD A,(IY + dis)
FD 8E	ADC A,(IY + dis)
FD 96	SUB (IY + dis)
FD 9E	SBC A,(IY + dis)
FD A6	AND (IY + dis)
FD AE	XOR (IY + dis)
FD B6	OR (IY + dis)
FD BE	CP (IY + dis)
FD CB XX 06	RLC (IY + dis)
FD CB XX 0E	RRC (IY + dis)
FD CB XX 16	RL (IY + dis)
FD CB XX 1E	RR (IY + dis)
FD CB XX 26	SLA (IY + dis)
FD CB XX 2E	SRA (IY + dis)
FD CB XX 3E	SRL (IY + dis)
FD CB XX 46	BIT 0,(IY + dis)
FD CB XX 4E	BIT 1,(IY + dis)
FD CB XX 56	BIT 2,(IY + dis)
FD CB XX 5E	BIT 3,(IY + dis)
FD CB XX 66	BIT 4,(IY + dis)
FD CB XX 6E	BIT 5,(IY + dis)
FD CB XX 76	BIT 6,(IY + dis)
FD CB XX 7E	BIT 7,(IY + dis)
FD CB XX 86	RES 0,(IY + dis)
FD CB XX 8E	RES 1,(IY + dis)
FD CB XX 96	RES 2,(IY + dis)
FD CB XX 9E	RES 3,(IY + dis)
FD CB XX A6	RES 4,(IY + dis)
FD CB XX AE	RES 5,(IY + dis)
FD CB XX B6	RES 6,(IY + dis)
FD CB XX BE	RES 7,(IY + dis)
FD CB XX C6	SET 0,(IY + dis)
FD CB XX CE	SET 1,(IY + dis)
FD CB XX D6	SET 2,(IY + dis)
FD CB XX DE	SET 3,(IY + dis)
FD CB XX E6	SET 4,(IY + dis)
FD CB XX EE	SET 5,(IY + dis)
FD CB XX F6	SET 6,(IY + dis)
FD CB XX FE	SET 7,(IY + dis)
FD E1	POP IY
FD E3	EX (SP),IY
FD E5	PUSH IY
FD E9	JP (IY)
FD F9	LD SP,IY

DESIGNED and PUBLISHED by COLIN MITCHELL at 35 Rosewarne Ave., CHELTENHAM, VICTORIA. 3192 (03) 584 2386