elektor

72
April
U.K. 60 p
U.S.A. $2

up-to-date electronics for lab and leisure

special supplement

16-bit microprocessors

transistor tester

select matched pairs the easy way

chess computer

using a 16-bit µP

Make it for a Song!

The New Maplin Matinée

Amazing Value For Only **£299.95** + £99.50 for cabinet if required.

Easy to build. Latest technology — means less cost, less components and 80% less wiring. Comparable with organs selling for up to £1,000.00. Two 49-note manuals. 13-note pedalboard. All organ voices on drawbars. Preset voices: Banjo, Accordion, Harpsichord, Piano, Percussion. Piano sustain Sustain on both manuals, and pedalboard. Electronic rotor, fast and slow. Vibrato and Delayed vibrato. Reverb. Manual and Auto-Wah. Glide (Hawaiian Guitar Sound). Single finger chording plus memory. 30 Rhythms! 8-instrument voicing. Major, Minor and Seventh chords. Unique walking bass lines with each rhythm. Unique countermelody line with each rhythm. Truly amazing value for money.
Full construction details in Electronics & Music Maker magazine.

The complete buyers' guide to electronic components. With over 300 pages, it's a comprehensive guide to electronic components with thousands of photographs and illustrations and page after page of invaluable data. Get a copy now — it's the one catalogue you can't afford to be without.

Post this coupon now for your copy of our 1981 catalogue price £1.
Please send me a copy of your 320 page catalogue. I enclose £1 (Plus 25p p&p). If I am not completely satisfied I may return the catalogue to you and have my money refunded. If you live outside the UK send £1.68 or 12 International Reply Coupons.
I enclose £1.25. E4/81

Name ______________________

Address ______________________

____

MAPLIN

Maplin Electronic Supplies Ltd.
All mail to: P.O. Box 3, Rayleigh, Essex SS6 8LR
Telephone: Southend (0702) 554155. Sales (0702) 552911.
Shops:
159-161 King Street, Hammersmith, London W6. Telephone: (01) 748 0926.
284 London Road, Westcliff-on-Sea, Essex. Telephone: Southend (0702) 554000
Both shops closed Mondays.

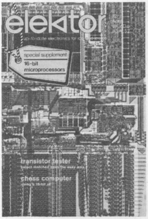
Catalogue now on sale in all branches of WHSMITH Price £1.00

contents

EDITOR:
W. van der Horst

UK EDITORIAL STAFF
T. Day E. Rogans
P. Williams

TECHNICAL EDITORIAL STAFF
J. Barendrecht
G.H.K. Dam
P. Holmes
E. Krempelsauer
G. Nachbar
A. Nachtmann
K.S.M. Walraven

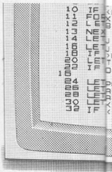


SPECIAL SUPPLEMENT:

16-bit microprocessors

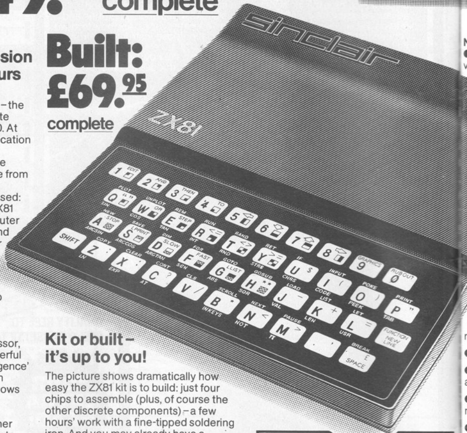
The pace of present-day developments in computer science is amazing. Now even 16-bit 'micro' processor systems are integrated on a single chip and already equal, if not surpass, modern 'minicomputers'. This means that a full-fledged personal computer is within any enthusiast's reach. The only problem is: which one? This article gives a survey of the available types and discusses the pros and cons of each particular system.

New! Sinclair ZX81 Personal Computer. Kit: £49.95 complete



```
10 IF
11 FOR
12 LET
13 NEX
14 LET
16 LET
18 IF
20 LET
22 IF
16
24 LET
26 LET
28 LET
30 LET
32 IF
```

Built: £69.95 complete



Reach advanced computer comprehension in a few absorbing hours

1980 saw a genuine breakthrough – the Sinclair ZX80, world's first complete personal computer for under £100. At £99.95, the ZX80 offered a specification unchallenged at the price.

Over 50,000 were sold, and the ZX80 won virtually universal praise from computer professionals.

Now the Sinclair lead is increased: for just £69.95, the new Sinclair ZX81 offers even more advanced computer facilities at an even lower price. And the ZX81 kit means an even bigger saving. At £49.95 it costs almost 40% less than the ZX80 kit!

Lower price: higher capability
With the ZX81, it's just as simple to teach yourself computing, but the ZX81 packs even greater working capability than the ZX80.

It uses the same micro-processor, but incorporates a new, more powerful 8K BASIC ROM – the 'trained intelligence' of the computer. This chip works in decimals, handles logs and trig, allows you to plot graphs, and builds up animated displays.

And the ZX81 incorporates other operation refinements – the facility to load and save named programs on cassette, for example, or to select a program off a cassette through the keyboard.

Higher specification, lower price – how's it done?
Quite simply, by design. The ZX80 reduced the chips in a working computer from 40 or so, to 21. The ZX81 reduces the 21 to 4!

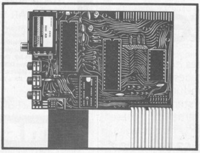
The secret lies in a totally new master chip. Designed by Sinclair and custom-built in Britain, this unique chip replaces 18 chips from the ZX80!

Proven micro-processor, new 8K BASIC ROM, RAM – and unique new master chip.

Kit or built – it's up to you!

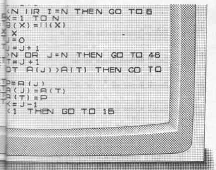
The picture shows dramatically how easy the ZX81 kit is to build: just four chips to assemble (plus, of course the other discrete components) – a few hours' work with a fine-tipped soldering iron. And you may already have a suitable mains adaptor – 600 mA at 9 V DC nominal unregulated (supplied with built version).

Kit and built versions come complete with all leads to connect to your TV (colour or black and white) and cassette recorder.



New Sinclair teach-yourself BASIC manual

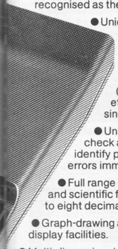
Every ZX81 comes with a comprehensive, specially-written manual – a complete course in BASIC programming, from first principles to complex programs. You need no prior knowledge – children from 12 upwards soon become familiar with computer operation.



```
<N IIR I=N THEN GO TO 6
5X=1 TO N
B(X)=II(X)
X
J=0
J=J+1
>N OR J=N THEN GO TO 46
T=J+1
DT A(J)>A(T) THEN GO TO

P=A(J)
A(J)=A(T)
A(T)=P
K=J-1
<1 THEN GO TO 16
```

New, improved specification



- ● Z80A micro-processor – new faster version of the famous Z80 chip, widely recognised as the best ever made.
- ● Unique 'one-touch' key word entry: the ZX81 eliminates a great deal of tiresome typing. Key words (RUN, LIST, PRINT, etc.) have their own single-key entry.
- ● Unique syntax-check and report codes identify programming errors immediately.
- ● Full range of mathematical and scientific functions accurate to eight decimal places.
- ● Graph-drawing and animated-display facilities.
- ● Multi-dimensional string and numerical arrays.
- ● Up to 26 FOR/NEXT loops.
- ● Randomise function – useful for games as well as serious applications.
- ● Cassette LOAD and SAVE with named programs.
- ● 1K-byte RAM expandable to 16K bytes with Sinclair RAM pack.
- ● Able to drive the new Sinclair printer (not available yet – but coming soon!)
- ● Advanced 4-chip design: micro-processor, ROM, RAM, plus master chip – unique, custom-built chip replacing 18 ZX80 chips.

sinclair ZX81

Sinclair Research Ltd,
6 Kings Parade, Cambridge, Cambs.,
CB2 1SN. Tel: 0276 66104.
Reg. no: 214 4630 00

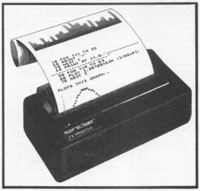
If you own a Sinclair ZX80

The new 8K BASIC ROM used in the Sinclair ZX81 is available to ZX80 owners as a drop-in replacement chip. (Complete with new keyboard template and operating manual.)

With the exception of animated graphics, all the advanced features of the ZX81 are now available on your ZX80 – including the ability to drive the Sinclair ZX Printer.

Coming soon-the ZX Printer

Designed exclusively for use with the ZX81 (and ZX80 with 8K BASIC ROM), the printer offers full alphanumerics across 32 columns, *and* highly sophisticated graphics. Special features include COPY, which prints out exactly what is on the whole TV screen without the need for further instructions. The ZX Printer will be available in Summer 1981, at around £50 – watch this space!



```
10 FOR I=1 TO 22
20 PRINT "/"
30 NEXT I
40 PRINT AT 11,0;"--------------"
------------------------------
50 FOR I=0 TO 63
60 PLOT I,22+20*SIN (I/32*PI)
70 NEXT I
PLOTS THIS GRAPH:-
```

16K-BYTE RAM pack for massive add-on memory.

Designed as a complete module to fit your Sinclair ZX80 or ZX81, the RAM pack simply plugs into the existing expansion port at the rear of the computer to multiply your data/program storage by 16!

Use it for long and complex programs or as a personal database. Yet it costs as little as half the price of competitive additional memory.

How to order your ZX81

BY PHONE – Access or Barclaycard holders can call 01-200 0200 for personal attention 24 hours a day, every day.

BY FREEPOST – use the no-stamp-needed coupon below. You can pay by cheque, postal order, Access or Barclaycard.

EITHER WAY – please allow up to 28 days for delivery. And there's a 14-day money-back option, of course. We want you to be satisfied beyond doubt – and we have no doubt that you will be.

To: Sinclair Research Ltd, FREEPOST 7, Cambridge, CB2 1YY.

Remember all prices shown include VAT, postage and packing. No hidden extras. Please send me:

Qty	Item	Code	Item price £	Total £
	Sinclair ZX81 Personal Computer kit(s). Price includes ZX81 BASIC manual, excludes mains adaptor.	12	**49.95**	
	Ready-assembled Sinclair ZX81 Personal Computer(s). Price includes ZX81 BASIC manual and mains adaptor.	11	**69.95**	
	Mains Adaptor(s) (600 mA at 9 V DC nominal unregulated).	10	**8.95**	
	16K-BYTE RAM pack(s).	18	**49.95**	
	8K BASIC ROM to fit ZX80.	17	**19.95**	

TOTAL: £

Please tick if you require a VAT receipt □

*I enclose a cheque/postal order payable to Sinclair Research Ltd, for £________

*Please charge my Access/Barclaycard account no.

*Please delete/complete as applicable. Please print.

Name: Mr/Mrs/Miss________

Address________

FREEPOST – no stamp needed. ELK04

selektor

Private TV by public telephone

Conventional television signals are very expensive to transmit over landlines, because the signal occupies such a large part of the electromagnetic spectrum that costly circuits and cables have to be used. British Telecom, the telecommunications part of the British Post Office, has developed a cheaper, digital slow-scan television system which uses the two-wire domestic telephone line and is now being tried out in many different applications. For example, TV conferences can be set up by dialling-out and distant radar displays are being made available at a port control office, using existing lines.

If the cost of sending a television signal over considerable distances by landline were small, there would be many more commercial applications in which it could play an important part. Obvious examples are remote surveillance of widely dispersed sites and obviating a lot of travel by holding 'video' conferences. But conventional television is exceedingly expensive to transmit by line because of the very large bandwidth of the basic signal; that is, the range of frequencies that the signal takes up in the electromagnetic spectrum is uncomfortably big. A 625-line television signal occupies a band 5½ MHz wide, more than five times the width of the entire medium-wave broadcast band.

This means that to send a TV signal over distances greater than a few hundred metres by line requires special cables to be laid, and for trunk connections a satellite or a chain of microwave radio links may have to be used. The capital and operating costs of such systems are usually so high that only rarely can an economic case be made for television when alternative arrangements will do. For some years, British Telecom (the telecommunications part of the British Post Office) has been studying ways of reducing the cost of television transmission in an effort to make more applications economically attractive. All the techniques put forward inevitably compromise the quality of the display in some way (that is, they send less information than the basic TV signal is capable of carrying) and therefore call into question the acceptability of such displays for various purposes; in each case the suitability can be judged only by trial under realistic circumstances. Industrial development has been started on a range of novel equipment, including slow-scan TV converters, narrow-band (1-MHz bandwidth) TV equipment, video-conference terminals and real-time picture-compression converters, and over the next two to four years will be conducting trials within private and public-sector companies which have suitable applications.

The first of these developments to go into production and on practical trials is a slow-scan TV system which operates over the public telephone network or any data circuit.

Reduction

Because conventional television signals occupy a bandwidth equivalent to almost 2000 telephone circuits, a great deal of compression of that frequency range is obviously needed. There are three distinct ways of achieving that aim. In increasing order of complexity, and therefore cost, it can be done by reducing picture clarity; by reducing the speed of transmission of each image (sending fewer picture 'frames', or complete scans, in a given time); or by reducing the amount of redundant information in the picture. The first two means are fairly obvious, the third is less so.

In the present system the most significant reduction is in speed: for many purposes it simply is not necessary to transmit the usual 25 frames every second, so an immediate reduction of 100 or 1000 times is realized by taking four or 40 seconds respectively to send one image. A further factor of five is achieved by reducing the clarity, through sending only one complete set of scanning lines, or field, instead of two interlaced sets per frame as in broadcast TV, and limiting the horizontal resolution (along the scanning line) to about 210 visible points per line.

In conventional TV cameras and monitors, the signal representing one field is generated and displayed in 20 milliseconds. It follows that to make a discrete field available for transmission and viewing over a much longer period, a picture store must be provided at both the transmitting and receiving terminals. At the start of a sequence, or upon an alarm or other command, a TV field is captured in the transmitting-end store. From there it is sent, at a rate determined by the available bandwidth of the transmission network, to the identical receiving-end store, the contents of which are continually displayed on the monitor. The observer therefore sees each new picture gradually over-writing the previous one from left to right, which is why we call the technique slow-scan TV.

Digital System

Slow-scan TV is not new. Analogue systems, in which the video-signal wave is carried over voice-band telephone lines, have been on the market for some years. They suffer from accumulation of noise, especially over long distances where slow-scan TV offers the greatest savings. Distortions also show up directly. Digital transmission systems, in which the analogue waveform is encoded as a series of digital pulses of equal amplitude, do not suffer these defects; provided digit errors in transmission are infrequent, the quality of picture is independent of the distance travelled. Moreover, with the advent of silicon-chip TV digitizers and cheap digital memories, the terminals match much better to digital transmission than they do to analogue. There is also scope for the third category of picture reduction, that is, removing some redundant information from pictures which have already been translated into digital code.

In our equipment, only relatively simple encoding is used. The picture is expensive to provide over any great distance, their use tends to be in the

1

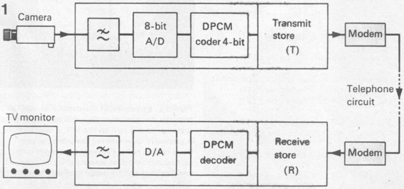


Figure 1. In television broadcasting, one complete picture field is generated and displayed in 20 milliseconds. To make a field available for transmission and viewing over a much longer period, picture stores must be used at both terminals, as in the above block diagram of a typical system. Listed below are various applications selected for system trials. Applications 1 to 9 can be classed as remote surveillance systems, with transmission in one direction only and the transmitter usually unmanned. The rest use two-way communication, though not necessarily involving the transmission of pictures in both directions. They work on voice-band telephone lines with a bandwidth restricted to about three kilohertz.

range of only a few kilometres where transmission can be over ordinary captured as 290 scan lines, each containing 210 visible picture elements (pels). When speech is encoded for telephone communication, 8-digit 'words' are used to convey information about the original speech waveform. This is known as 8-bit pulse-code modulation (PCM). Here, each picture element is encoded as a 4-bit word which represents the *difference* between that pel and the one immediately above it in the picture. In this way both picture stores are only half the size that they would be for 8-bit PCM. Transmission time is halved, too; picture quality, of course (at least to the practised eye), is slightly reduced. In the receiver terminal, the 4-bit differential words are decoded to 8-bit PCM before display, and field-repeating is necessary to feed the conventional 625-line TV monitor.

By further extending the coding scheme we can reduce the 4-bit words to an average of two to three bits, depending on the content of the picture. However, there is a good prospect that in later generations of slow-scan systems more complex processing will eventually reduce the average information to, perhaps, between half a bit and one bit per sample, with a corresponding reduction in time before the next frame can be displayed to keep the picture up to date.

The use of entirely digital apparatus has the additional advantage that it can be matched to any digital circuit. So, while a good quality voice-band circuit can support a data rate of 9·6 kbit/s, giving a picture time of 15 to 25 seconds, an international circuit accepting subscriber dialling will be slower than that. On the other hand, a private wire or a 'packet-switched' system, in which data is stored and then transmitted rapidly when the user's turn in the traffic queue comes round, may be a good deal faster. So digital slow-scan is considerably versatile in the way it fits into existing telecommunications networks.

Applications

Because of the cost of terminal equipment, slow-scan TV is not appropriate to short distances, for it is then probably cheaper either to install a special cable or to use 313-line TV signals over telephone lines with repeating amplifiers at intervals of, say, 1½ km or less. In fact such a system is included in plans for other visual-service trials. But for applications where service must be provided at short notice, or only temporarily, slow-scan TV may be attractive if the resolution and up-date time are acceptable. The same is true of connections longer than a few kilometres, where conventional TV becomes too costly or impractical.

To study how well it works, various applications have been selected for trials. Systems in operation or already planned are listed in the table on page 13. The upper group can be classed as remote surveillance; the systems are uni-directional, with the transmitting end usually unmanned. Some are permanently in operation, but those using dial-up circuits, in which communication over the link is set up by dialling over the telephone, either in a private network or in the public system, require an automatic-answering or alarm-triggered automatic-dialling device. In a minority of applications the transmission rate is 48 kbits/s, highly desirable from an operational view-point

Table 1

	Application	Distance	Transmission
1	Protecting radio station after terrorist threat	5 km	4-8 kb/s dial-up on PSTN
2	Detecting illegal dumping; rapid installation, temporary use	2 - 3 km	4-8 kb/s dial-up on PSTN
3	Protecting bullion vault	5 km	48 kb/s on metallic pair
4	Security monitoring of premises at night from two cameras on one system	120 km	4-8 kb/s dial-up on PABX
5	Monitoring road traffic for control of signals	25 km	48 kb/s on repeatered pair
6	Extension of radar displays to port control office	10 km	48 kb/s on metallic pair
7	Underwater surveillance from submersible craft (via ultrasonic transmission link)	—	Data via acoustic transmission
8	Security monitoring of premises at night, from one place	400 km	4-8 kb/s dial-up on PSTN
9	Occasional access to remote camera for traffic control	20 km	4-8 kb/s dial-up on PSTN
10	Sending X-ray pictures from hospital to consultant	100 km	4-8 kb/s dial-up on PSTN
11	Editorial submission to upper management	80 km	9-6 kb/s on private voice circuit
12	Liaison between processing plants	300 km	4-8 kb/s dial-up on PSTN
13	Aid to project collaboration between laboratory and contractors	150 km	4-8 kb/s dial-up on PSTN
14	Conferences between people in various places	250 km	4-8 kb/s dial-up on PSTN
15	Conferences between people in various places	40 km	48 kb/s on private groupband circuit
16	Conferences between people in various places	200 km	4-8 kb/s on private telephony system
17	Editorial conferences between newspaper offices	150 km	4-8 kb/s on private circuit

2

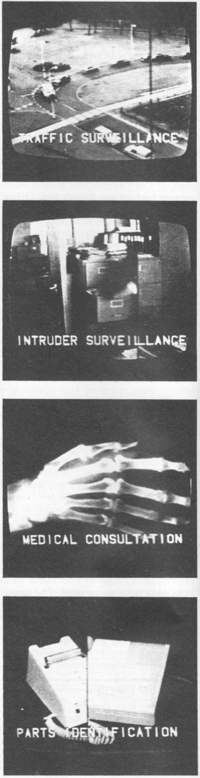


Figure 2. Examples of the quality of various pictures received over digital slow-scan TV systems. In the two to the left, a new picture is gradually over-writing the previous one from left to right. Prolonged tests are being carried out to assess acceptability in practice and findings will be made available late in 1981.

because the up-date time is only five seconds; but because such circuits are telephone wire pairs. If necessary, for greater distances, amplifiers are used at intervals of about 10 km to maintain signal strength. The only likely use over greater distances is where a so-called 'groupband' circuit, with a bandwidth of 48 kHz, is already available for other purposes but can be taken over when it is idle.

All the other systems are on voice-band circuits, that is, with bandwidths of about three kilohertz. Although this means the picture speeds have to be slower, such circuits are by far the most readily available, especially if dial-up access to the telephone network is acceptable. Not surprisingly, many security-surveillance systems are not needed during working hours, but their continuous use of lines and switch outlets at night, when not needed for other traffic, makes for economical use of those resources.

The lower group in the table involves two-way communication, though not necessarily sending pictures both ways (see applications 10 and 11). The main use is as an aid to working discussions between people in different places but closely involved with the same project, product or service. It is in these 'conferencing' applications that there is greatest room for doubt about the quality and speed of the picture. The restricted resolution of 210 x 290 visible pels is thought to be good enough for sketches, diagrams, many (but not all) X-ray pictures, newspaper lay-outs and views of most solid objects such as printed circuit boards, but it does not reproduce 200-mm lines of typescript well enough for comfortable viewing; it remains to be seen how often this becomes a disadvantage in practice. It is, of course, technically feasibly to make a slow-scan system of, say, 420 x 580 pels, but the transmission time becomes four times longer, which is perhaps an even greater detriment to conference applications.

Few people who are likely to use slow-scan TV in Telecom trials or in the first years of a public service are familiar with the form of picture presentation. Many react quite favourable at first contact, tending not to notice the lower resolution nor to condemn the slow speed or lack of colour. But it is now becoming apparent that such spontaneous opinions are a poor guide to the true worth of the system. Consider the frustration of the security man who feels sure something is in need of attention but must wait 50 seconds to be absolutely certain, or of the energetic designer who wishes to display several modifications to his sketch in quick succession.

Feed-back

Only a prolonged test in the service for which it is intended can give a reliable indication of how acceptable a system is for a particular job. The economics of the system, too, can be assessed only against alternative ways of performing the same essential task, such as having a security guard at the site to be protected. This means that feed-back from people trying the system out, usually after they have been using it for at least six months, is vital to Telecom's visual-service trials.

In addition, experience of the engineering requirements is a valuable pointer to the directions that further system developments should take. It has already been found, for example, that means for camera selection and other forms of control, probably by momentarily reserving the data flow, would greatly improve the value of a system for surveillance work, as would an ability to detect movement.

The findings of our slow-scan television trial will be summarized in a report late in 1981, a digest of which will be released to interested organizations. The outcome promises to be a profitable new public service.

Dr N.D. Kenyon,
British Telecom Research Laboratories,
Martlesham Heath

(660 S)

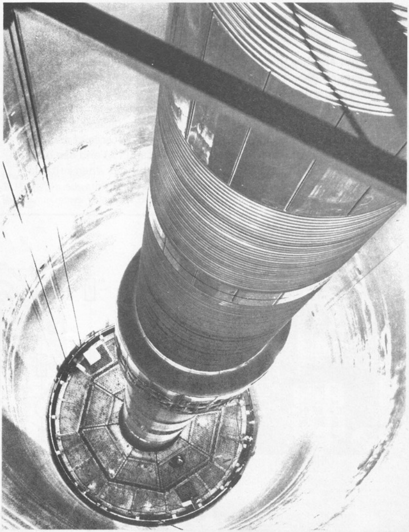
selektor

Rocket to inner space

The 30-megavolt vertical tandem Van de Graaff accelerator due to come into use early this year at the UK Science Research Council's Daresbury Laboratory, near Liverpool, is the largest in the world and will provide scientists with a novel means of studying nuclear matter. Known as the Nuclear Structure Facility (NSF), it will be used mainly to accelerate heavy ions. Collisions of heavy ions are predicted to be capable of generating shock waves in which abnormally high-density matter may be created, such as exists only in the cores of neutron stars. This view is of the 41-metre high-voltage column, looking down past the centre high-voltage terminal to the base; the picture was taken before the intershield was fitted.

(Spectrum 172)

(661 S)



'Transistor match-maker'

A transistor tester for finding matched pairs.

'Oh no, not another transistor tester!" may well be several readers' initial reaction. Don't worry, this article is designed to save you and your eyes hours of strain and boredom. The device is capable of picking matched transistor pairs from a whole pile of 'possibles', and all within seconds. Two transistors will be 'matched' if their base/emitter voltage and their current amplification are the same. The degree of accuracy may range from 'roughly the same' to 'identical' (1%) and can be adjusted, as required. It really is an indispensable aid when suitable matched transistors are needed for differential amplifiers, or temperature sensors.

Finding matching transistors is a highly unpopular and tedious occupation. Nevertheless, it is one of those jobs that just have to be done from time to time, as such transistor pairs are often used in differential amplifiers and in particular, of course, when they act as temperature sensors. Usually, it means improvising a test circuit, getting hold of a universal meter and spending the entire evening testing a whole pile of transistors while jotting down the results. Better make sure there's nothing on the box that night!

Elektor has now come up with a short-cut in the form of a transistor tester. It makes life a lot easier, as it actually compares two transistors. LEDs light to indicate whether their U_{BE} and H_{FE} correspond or not. The circuit does all the work — you just plug in transistors and watch the LEDs. There are three LEDs altogether: one to indicate that sample no. 1 is 'better' than no. 2, one to indicate the opposite and another to show the pair is a perfect match.

Operation

All this may seem rather complicated, but in actual fact the tester is based on a fairly straightforward principle. Figure 1 shows a simplified version of the circuit to make matters clear. A triangular wave-shape is applied to the transistors under test (TUTs). Any differences between their collector voltages are detected with the aid of two comparators and will be indicated by the LEDs. That, in a nutshell, is the theory.

Now to put it into practice. As shown in figure 1, the two TUTs are driven by exactly the same control voltage, but their collector resistors are marginally different. $R2_a$ and $R2_b$ together are slightly greater in value than R1, whereas $R2_a$ alone is a little smaller than R1. And that is the whole trick of the tester circuit.

Let us suppose the two TUTs are identical as far as their U_{BE} and H_{FE} are concerned. The rising slope of the input voltage will then switch them both 'on' at the same time and the voltage at their collectors will drop. If we were to freeze the action at any point, we would see that TUT2's collector voltage is a tiny bit lower than that of TUT1, due to its total collector resistance being slightly greater. Since, on the other hand, $R2_a$ is a little smaller in value than R1, the voltage at the $R2_a/R2_b$ junction will be slightly higher than that at the collector of TUT1.

As a result of this, the '+' input of comparator 1 will be positive with respect to its '—' input. This means that the output of K1 will be high and LED

1

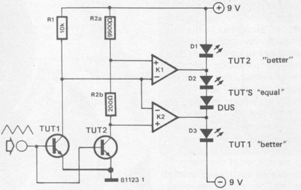


Figure 1. The simplified circuit diagram of the transistor tester. Comparators check the two transistors for differences in voltage and the result is indicated by the LEDs.

2

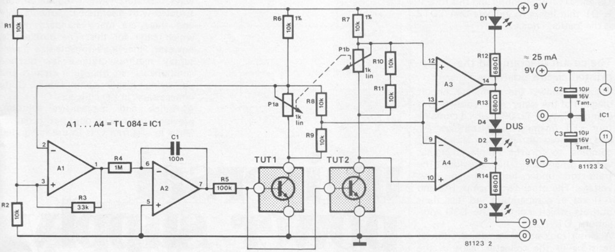


Figure 2. The final version of the circuit diagram. It is built around a set of four opamps. Two (A1 and A2) constitute the triangular wave generator and the other two act as comparators.

D1 will not light. At the same time, the '+' input of K2 is negative with respect to its '—' input and so its output will be low and LED D3 will not light either. In this situation, where K1's output is high and K2's is low, D2 will light up as an indication that the two transistors are in fact identical.

Now let us see what happens when TUT1 has a lower U_{BE} and/or a higher H_{FE} than TUT2. During the positive edge of the triangular signal, the voltage at the collector of TUT1 will drop sooner and/or faster than that of TUT2. Comparator K1 will react to this in the same manner as before, in that the '+' input will again be positive with respect to the '—' input and its output will therefore be high. Since TUT1's low collector voltage is also connected to the '—' input of K2, that particular '—' input will now be lower than the '+' input connected to TUT2's collector. This will cause the output of K2 to rise. Since the two comparator outputs are high, D1 will not light; D2, like D1, will be connected between two high levels and thus unable to light either, and now there is nothing to stop D3 from lighting. D3's LED will therefore indicate that TUT1 is the 'better man' of the two transistors.

If TUT2 turns out to be 'better', this will of course cause its collector voltage to drop at a faster rate. As a result, both the voltage at the collector itself and that at the $R2_a/R2_b$ junction will be lower than the collector voltage of TUT1. This means that the '+' inputs of the comparators will both become low

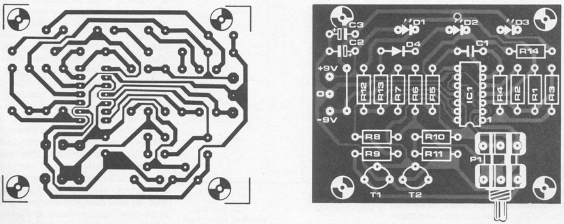
Parts List

Resistors:
R1,R2,R8 . . . R11 = 10 k
R3 = 33 k
R4 = 1 M
R5 = 100 k
R6,R7 = 10 k 1%
R12,R13,R14 = 680 Ω
P1 = 1 k tandem potentiometers, linear

Capacitors:
C1 = 100 n
C2,C3 = 10 µ/16 V

Semiconductors:
IC1 = A1 . . . A4 = TL 084
D1 . . . D3 = LED
D4 = DUS

3



The component overlay and track pattern of the tester printed circuit board. Reliable sockets should be used for the two TUTs.

with respect to the '—' input, so that the two outputs will be low. This prevents D2 and D3 from lighting and this time it is D1 that lights to indicate that TUT2 is the 'better' choice.

The circuit diagram and the printed circuit board

Figure 2 shows the complete circuit diagram of the tester. All it consists of is a single IC, type TL 084, which contains four FET opamps. Schmitt trigger A1 and the integrator built up around A2 combine to form a simple triangular wave generator. This provides the transistors under test with an input voltage. The other two opamps (A3 and A4) act as comparators and it is their outputs which control the LED indications, D1 . . . D3.

A closer look at the conglomeration of resistors in the collector leads of the two TUTs will explain why we used a simplified version of the circuit to clarify the principle. The final circuit looks much more complicated, as a tandem pot (P1) has been added to preset the range within which the transistors may be considered to be identical. If P1 is turned left as far as it will go and the middle LED (D3) lights, the two TUTs will be identical within about 1%. The 'matched pair' criterion is relaxed to about 10% tolerance when the pot is turned fully clockwise.

The maximum possible accuracy is limited by the tolerance in R6 and R7, by the offset voltage of the TL 084 and by the tracking accuracy of P1a and P1b. In addition, the transistors under test will react to changes in their temperature, which is something to watch out for. If, for example, a transistor is held in the hand and then inserted in the tester, the results of the test will be affected, and so it is better to wait until it cools off again before jumping to conclusions.

The tester requires a symmetrical power supply. The level of the supply voltage is not critical and the circuit will not only work well at the indicated + and —9 V, but also at + and —7 V or even at + and —12 V. The circuit can easily be powered from two 9 V batteries, as its current consumption is only 25 mA and, in any case, the tester is hardly likely to be switched on for hours on end. Being battery fed, the circuit can be constructed into a neat, compact device and is therefore easy to work with.

Figure 3 shows the tester's printed circuit board. It is difficult to see how anything could go wrong (touch wood!), considering the small amount of components required. All that it needs are a single IC, two transistor sockets for the TUT's, a few resistors and three LEDs. Make sure resistors R6 and R7 are 1% types.

universal power supply

As regular readers will agree, ample space and attention has been devoted to power supplies on Elektor's pages in recent years. They have become one of the designer cuisine's specialities, so to speak. Our September '80 issue, for instance, featured the precision power unit, a very neat, accurate device that can also act as a reference voltage source.

This time Elektor wishes to cater for more universal tastes and has therefore produced a less exclusive, but highly popular power unit: a cheap, straightforward, multi-purpose experimental supply that can be adjusted between 0 and 20 V.

• output voltage:	0 . . . 20 V coarse and fine adjustment
• output current:	2 A maximum
• short-circuit current:	about 2.3 A
• ripple < 1 mV (at full-load)	

An adjustable power supply can, of course, be designed in a number of ways. To start with, it could be constructed with discrete components only and there are many standard recipes which cater for this. The problem is, however, that a reasonable-size power supply requires quite a few discrete components, so that such circuits end up being highly complicated. Quite unnecessary, in the chip age.

A quick and inexpensive solution, provided the supply only has to deliver fairly low currents, is to use integrated voltage regulators. As soon as higher currents are involved, however, the price of integrated regulators also tends to go up . . . which brings us back to square one.

That is why a compromise must be sought: a reasonable quality power supply without breaking the bank. This particular power supply is a step in the right direction. It combines a few cheap, integrated low-power regulators and several series transistor 'heavies'. The ICs stabilise and control the voltage, without complicating matters, and the transistors provide the required number of amps.

Since the supply voltage of most circuits rarely exceeds 18 . . . 20 V, the upper voltage threshold has been chosen at 20 V. If a higher voltage is required (to test amplifiers, for instance) two power supplies may be connected in series. We'll come back to how that is done later. Besides, a double power supply has considerable advantages, as more and more circuits nowadays need both a positive and a negative voltage. A double version is therefore to be recommended for experimental purposes.

One of this circuit's greatest attributes is that its lower voltage threshold is really and truly 0 V — a commodity which very few other circuits can boast. Because of the large voltage range, provision has been made for both coarse and fine adjustment of the output voltage. Any experimental power supply will, of course, have to be short-circuit proof and this is certainly the case here. However, no arrangement was made for a presettable current limitation, as this would only serve to complicate matters and, in any case, experience has shown that this 'luxury item' is hardly ever used.

The supply can deliver up to 2 A — plenty for most applications. Furthermore, the power supply features a very low ripple voltage due to the IC's high

1

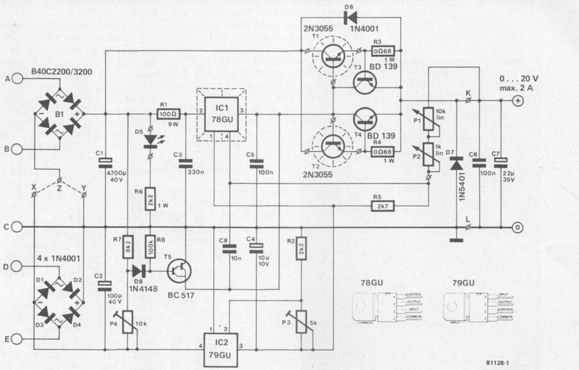


Figure 1. The circuit diagram of the power supply. IC1 is the main voltage regulator and IC2 provides a negative bias voltage enabling the output to be adjusted down to 0 V.

'ripple rejection' (70 dB). It will not exceed 1 mV over the full range of output current and voltage.

The circuit diagram

As figure 1 shows, the power supply circuit really isn't that complicated. In all fairness, this is not the complete version. We have omitted the 'front-end', because the transformer(s) and bridge rectifier B1 and D1 . . . D4 may be connected in various ways, according to the supply's purpose. This aspect will be dealt with further on in the article.

The rectified and smoothed transformer voltages appear across C1 and C2. The higher of the two goes, via R1, to the heart of the circuit: IC1, a four-pin 78GU voltage controller in a 'power watt' case. Normally speaking, the common input of the IC should be grounded, giving a minimum output voltage of 5 V. As it would be nice to have the lower voltage threshold at 0 V, however, the common input is connected to a –5 V negative voltage in this circuit. This negative bias voltage is obtained from the second voltage regulator (IC2) and is adjusted with P3.

The output of IC1 is buffered by two emitter followers (T1 and T2) that are connected in parallel to provide a maximum output current of 2 A. The output voltage can be coarsely adjusted with P1 between 0 and 20 V; P2 provides the fine adjustment.

T3 and T4 ensure short-circuit protection. As soon as the output current exceeds 2 A, the voltage across R3 and

2

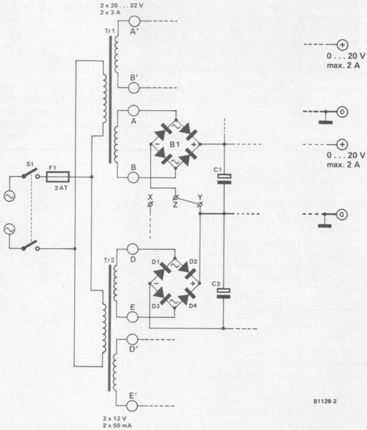


Figure 2. The transformer and bridge rectifier section when a double power supply is built up on two boards.

3

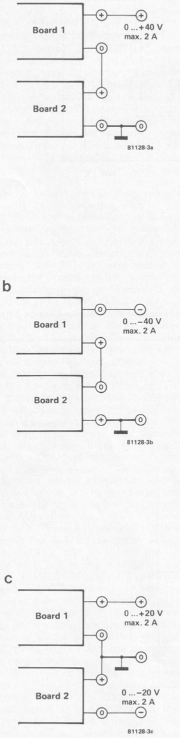


Figure 3. A double power supply is clearly much more universal for experimental supply purposes.

R4 reaches the point where T3 and T4 will start to conduct. This causes the power supply load to be directly connected to the output of IC1. IC1 would now like to deliver its maximum short-circuit current (about 1 A), but is prevented from doing so by R1. R1 usually only has to pass the base drive current to T1 and T2. When there is a short, the voltage across it drops to such an extent that the output current of IC1 is reduced to around 250 mA. In addition, the resistor prevents the thermal overload protection in IC1 from cutting in, as this would produce a square wave at the output.

Obviously, things could go wrong if the negative bias voltage collapses while the main positive supply is still present (immediately after switching off, for instance). For this reason, T5 is added: it shorts the output of IC1 if the negative supply fails.

C6 and C7 serve to 'kill' high frequency components and they also improve the transient response. That is why these capacitors are *not* mounted on the board but directly across the output terminals. D6 and D7 are protection diodes. D6 will bypass IC1, should any irregularity in the load cause current to pass in the wrong direction. D7 prevents IC1 from being blown up, if by chance the output of the power supply is connected to a voltage with the wrong polarity.

LED D5 has the 'cushiest' job of all the diodes: it merely acts as an on/off indicator.

Two versions

As we mentioned before, the transformers and bridge rectifiers have various possibilities, as the circuit can be built as either a single or a double power supply.

Double power supply:
2 x 0 . . . 20 V/2 A

If two individual presettable voltages are to be available, the 'raw' supply section will have to be constructed as shown in figure 2. Transformer Tr1 provides the main supply and a (small) transformer Tr2 provides the supply for IC2. Obviously, the double version will require two printed circuit boards, which are both constructed in the same manner and both include the Z-Y link. The points marked with an accent (A', etc.) belong to the second board.

Two separate supply voltages naturally allow for all sorts of combinations. This is illustrated in figure 3.

Single power supply 0 . . . 20 V/2 A

As figure 4 shows, a single power supply involves fewer components and less work. Only one transformer is now required (although it must have a double winding) and D1 . . . D4 may be omitted. In this case the Z-X link needs to be made.

Connecting B1 in the manner drawn in figure 4 provides IC1 with a positive voltage and IC2 with a negative voltage. In spite of the fact that the voltages are not equally loaded, the supply transformer will be under a symmetrical load.

Construction and setting-up

Figure 5 shows the printed circuit board for the power supply. A number of components are not mounted on the board: the supply transformer(s), the power transistors T1 and T2, the two potentiometers and C6 and C7.

T1 and T2 are mounted together on a heat sink with mica insulation. The heat sink must have 1.7°C/W thermal resistance, or less. These are available with pre-drilled holes (for 2 x TO3). The wiring from the transistors to the circuit

4

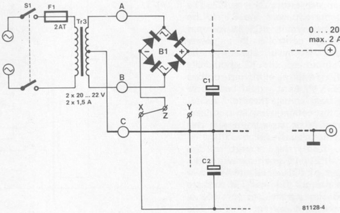


Figure 4. An ordinary single power supply only requires one transformer.

5

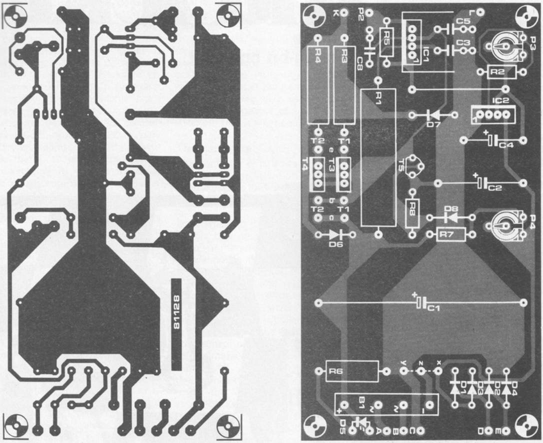


Figure 5. The printed circuit board and component overlay of the universal power supply.

Parts list

Resistors:

R1 = 100 Ω/9 W
R2 = 2k2
R3,R4 = 0.68 Ω/1 W
R5 = 2k7
R6 = 2k2/1 W
R7 = 8k2
R8 = 100 k

Capacitors:

C1 = 4700 μ/40 V
C2 = 100 μ/40 V
C3 = 330 n
C4 = 10 μ/10 V
C5,C6 = 100 n
C7 = 22 μ/35 V
C8 = 10 n

Semiconductors:

IC1 = 78 GU
IC2 = 79 GU
T1,T2 = 2N3055
T3,T4 = BD 139
T5 = BC 517
D1 . . . D4,D6 = 1N4001
D5 = LED
D7 = 1N5401
D8 = 1N4148

Miscellaneous:

P1 = 10 k, linear
P2 = 1 k, linear
P3 = 5 k preset
P4 = 10 k preset
S1 = mains switch
F1 = 2 A fuse, slo blo
B1 = B40 C2200/3200 (40 V/2 A bridge rectifier)
Tr1 = 2 x 20 . . . 22 V/2 x 3 A transformer (figure 2)
Tr2 = 2 x 12 V/2 x 50 mA transformer (figure 2)
Tr3 = 2 x 20 . . . 22 V/2 x 1.5 A transformer (figure 4)

board should be as short as possible and preferably of equal length. The base and collector connections to T1 and T2 all require their own leads to the board.

IC1 must also be provided with a heat sink, albeit a very small one. Please note: the power supply case may *not* be connected to the circuit's '0'. It should only be connected to mains 'earth'.

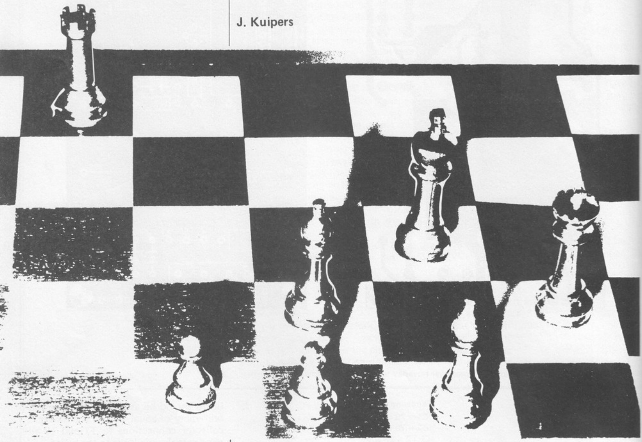
All that remains now is to calibrate the circuit. This can be done quite easily with a good quality multimeter. Let's deal with it step by step:

- Turn P3 and P4 to 0 Ω (fully anti-clockwise).
- Switch on (mains switch S1) and set P1 and P2 to minimum resistance.
- Turn up P4 until 0 V is measured at the R7-P4-D8 junction.
- Now turn P3 until exactly 0 V is measured at the output of the power supply.
- P1 and P2 can now be used to adjust the output voltage between 0 and 20 V.

intelekt

a sixteen-bit chess set

J. Kuipers



Do you play chess? Are you looking for an opponent who is always available . . . never gets impatient . . . plays a reasonably strong game . . . and even allows you to cheat a little, if you really want to? If so, it's time you met Intelekt!
So much for the advertising blurb. Actually, the chess computer described in this article does play a good game. It is designed around Intel's new 16-bit microprocessor, the 8088, which makes for reasonable speed and reasonable intelligence. Even at its 'stupidest' level of play (25 seconds per move) it will make a worthy opponent for many chess enthusiasts. At level three (out of eight), it thinks for five minutes or so per move – and provides what we considered a challenging game. Obviously, this evaluation is based to a large extent on our own chess skills. You can judge for yourself: some examples of actual games are included, with comments. If you feel that we played a stupid game, there are still five more intelligence levels to go; on the other hand, if the games look complicated, Intelekt would love to challenge you!

Chess computers are no longer a novelty. This is surprising, when you think of it: a few years ago, it seemed unlikely that even big commercial computers could be taught to play a reasonable game! By now, however, you can buy domestic versions for anywhere between £ 20 and £ 500. By and large, the 'good' machines cost anything from £ 200 up; unfortunately, however, 'intelligence' is not always proportionate to price.

To really determine the best value for money, you would have to play several games against all the available chess computers. We have yet to find somebody who has done this! As 'second-best evaluation', you can either play the machines against each other (with the risk that both play stupid moves without realising it) or else try them out on chess problems. The latter course, in particular, seems very popular for 'comparative reviews' in magazines. In our opinion, this is a very second-rate approach: the fun and challenge in chess is not in solving 'mate in three' (when 'mate in four' is easy); the idea is to manoeuvre your opponent into a position where you can 'mate' him! In other words, the fun is in playing the game — not in ending it.

What is all this leading up to? Quite simple: if you want to know how 'good' Intelekt is in comparison with other chess computers . . . we don't know! (Hurriedly:) But we get the impression that it's pretty good. We tried it out on chess problems that have been used in reviews. Where there was one obvious 'correct' move, Intelekt found it — often even at level 1. Where there was an obvious move that led to mate in four or five and an unexpected one that gave mate in three, it invariably selected the 'obvious' move. However, we played games against a few commercial machines that scored highly in reviews, and found them rather unexciting; we played against Intelekt and it was good fun.

To sum up its strong points in a few nutshells:

- it is easy to set up any position (even halfway through a game);
- illegal moves are not accepted;
- it knows all the rules of the game; castling, for instance, is obviously taken into account as a 'possible move';
- it can play either black or white (or even both sides!);
- it knows the value of sacrificing a piece to gain positional advantage: not only will it ignore this kind of 'sacrifice', it will even propose them where this seems worth trying;
- it plays a good game. This, in our opinion, is what counts.

Who or what is Intelekt? He (or it) is an electronic circuit containing a microprocessor and a chess program (in ROM), with an input/output that must be connected to a 'terminal' — the 'Elekterminal' (Elektor, November/December 1978), for instance. You make your moves by entering them on the keyboard of the terminal; Intelekt answers by displaying the board, his moves and comments (!) on a TV screen, via the same terminal. In other words, Intelekt is a brain; to speak to him and receive his replies you also need a 'terminal'.

In this article, we will give a brief description of the 'hardware' that is involved (circuit and printed circuit board) but no indication of the 'software' (the actual program). Instead, we will attempt to give as clear an impression as possible of his chess skills. After all, that is what counts!

The hardware

The complete circuit is shown in figure 1. It is not our intention to discuss it in minute detail, but we will attempt to paint a sufficiently clear overall picture.

To start with the '16-bit brain' (the 8088): this microprocessor can run in either 'minimum' or 'maximum' mode, depending on the logic level at pin 33. As the words indicate, maximum mode is intended for large systems and minimum mode for little ones. Intelekt belongs in the latter category. As explained in the supplement on 16-bit microprocessors, the 8088 produces the bus control signals itself when it is set to minimum mode; in maximum mode, a further IC would be needed to control the (more extensive) bus.

Inside the CPU itself, data is handled as 16-bit 'words'. However, the data bus that connects it to the outside world is only 8 bits wide. This means that each 16-bit word must be cut into two 8-bit bytes before it can be put on the data bus. For obvious reasons, these two chunks of data are transmitted one after the other — not simultaneously . . . In other words, they are 'time multiplexed'.

In actual fact, things are even more complicated. When Intel introduced the 8085 (a 'normal' 8-bit microprocessor), they used a single set of pins for a multiplexed address/data bus. Now, in the 8088, they've used the same system: the lowest eight address bits also appear on what we have so far called the data bus. This saves pins, making for a smaller and cheaper IC package, and the information is still available at the exact moment that it is required. First the address bits, obviously (the ALE pin indicates that a valid address is being output); then the data, in two 8-bit chunks.

1

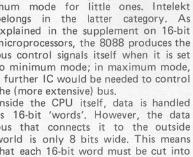


Photo 1. Some of the main control signals, as they appear on the screen of a normal oscilloscope.

Having saved seven pins (the eight multiplexed pins are saved, but ALE must be added), any normal designer immediately starts wondering what he can do with them. Apparently, Intel designers are no different. On the 8085, they used the pins for interrupt signalling; now, in the 8088 we find the address range has been extended to 1 Mbyte (one *million* bytes of memory!).

If the special Intel memory ICs are used, seven tracks can also be saved on the printed circuit board. However, we decided against this; instead, the data and address buses are separated by means of an octal latch (IC3), so that the address information is always available and normal memory ICs can be used.

If all this seems complicated, take a look at photo 1. This shows a group of signals, as they would appear on a 'normal' oscilloscope (not 'cleaned up' by a logic analyser). The upper line is the clock, ticking over at 5 MHz (!); all further timing is derived from this. At point ①, the processor has transmitted the new address information. One of the address/data outputs (ADO) is shown as the second trace. This is followed immediately by the ALE pin (third trace) going high, indicating that a valid address is now present at the output of the CPU. The corresponding output from the address latch (IC3) is shown as the fourth line from the top; as can be seen, each time ALE goes high this output assumes the same level as that on the ADO line, and holds it until the next ALE pulse appears.

If the processor now intends to 'read' data, it sets pin 32 ($\overline{RD}$) to a low logic level as can be seen in the fifth trace on the photo. When the data is to be read from EPROM, the correct chip has already been selected by the preceding address cycle. The $\overline{RD}$ line is connected to the $\overline{OE}$ pin (output enable) of both EPROMs, so the selected memory chip will now put the desired data on the bus (at ③ on the second trace). The processor 'reads' this data and immediately returns the $\overline{RD}$ pin to a high logic level. It can now put the next address on the bus, after which the whole cycle is repeated.

When reading from RAM, the basic principle is the same. However, this type of memory does not include an 'output enable' pin, so the read (or write) signal is included in the 'chip select' logic (CS).

Writing into RAM is similar to reading. As before, the first step is to select the address. Then, immediately after the negative-going edge of the ALE

1

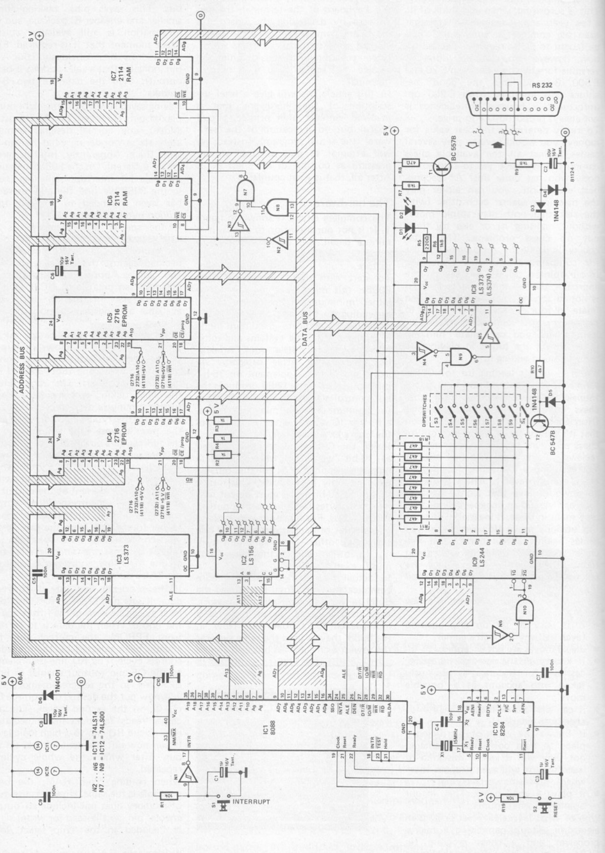


Figure 1. The complete circuit of Intelekt. The terminal is connected as shown at the lower right-hand corner in the circuit. Note that an interrupt key is included, although this is not required for operating the chess computer.

pulse, the processor puts the data onto the AD lines (③). It then sets $\overline{WR}$ at logic 0 (the sixth trace on the photo) to indicate that the data is valid. This 'write' signal is combined with the address information; the correct address is selected and the data is stored in RAM. The data on the bus remains valid for the complete duration of the write pulse.

So far, so good — but how is the RAM to know whether it is to transmit or receive data? This is where the DT/R signal comes in ('data transmit/receive'; the lower trace in the photo). As the address information goes out, this pin is set to logic 1 for a write cycle, or to logic 0 for read. It is passed through an inverter to drive the $\overline{WE}$ (write enable) inputs to the RAMs.

One address in a million

Although the 8088 can handle over one million addresses, Intelekt only needs a good 16,000. Obviously, things would tend to get confusing if several 'chips' started to 'talk' at once. At any given moment, the CPU should only be in contact with one memory IC, and this is where the address decoder (IC2) comes in. This IC monitors three of the address lines (A11 . . . A13) and converts them into eight chip-select signals. Depending on the address range indicated, one of these chip-select signals goes to logic 0 and the corresponding memory IC can communicate with the CPU via the data bus. If the processor wants to talk to an input/output IC, it sets the 10/$\overline{M}$ line to logic 1, deactivating the address decoder.

The address decoder has open-collector outputs. This simplifies matters if several outputs are to be combined — for instance when using larger EPROMs in some future updated version. Each output defines a 2 k address block, so two of these blocks would have to be combined (by connecting the two corresponding pins of the address decoder together) if a 4 k EPROM, type 2732, is to be used.

Although the RAM chip used (the 2114) is only a 1 K type, there is no reason why it should not be allocated its own 2 K block of addresses. (Note that the two 2114s each take care of four data bits; together, they form the 1 K x 8 memory). As shown in figure 2, the RAM is located at the lowest memory addresses — from 00000 to 003FF. Since it is enabled during the complete 2 K block, a duplicate RAM area appears from 00400 to 007FF. In other words, two different addresses define each RAM memory cell.

The input/output (I/O) chips IC8 and IC9 don't need an address decoder. All I/O write instructions enable IC8, via the combined 10/$\overline{M}$ and $\overline{WR}$ signals; similarly, all I/O read operations refer to IC9.

Allocating the addresses

One might assume that EPROM and RAM could theoretically be located anywhere in memory. In practice, this is not quite true. When the processor is reset, it starts to run a program from address FFFF0 on. To make sure that there is a program there, it is advisable to locate an EPROM in this final address block.

2

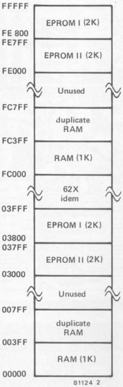


Figure 2. The 'memory map'. Although the theoretical address range is 1 Mbyte, it actually consists of 64 identical 16 Kbyte sections.

Furthermore, after an interrupt, the processor goes looking for an 'interrupt vector' (this is the address where the corresponding interrupt routine is located) at one of the lower memory addresses. Since it is useful to be able to change these addresses, RAM must be located in the lowest address block. Even though Intelekt doesn't actually make use of the interrupt facility, it was decided to locate the memory at the 'normal' addresses. This leads to the situation shown in figure 2: RAM (with its duplicate) in low memory, and two blocks of EPROM (4 k in all) at the top.

All this may seem quite reasonable, until you start thinking it over. EPROM is at the top and RAM at the bottom of a one *mega*-byte address range — but the address decoder is only defining eight 2 K blocks of memory! How can 16 K be equal to 1 M?

Since we are now tossing out K's and M's at the rate of one or two in each sentence, it is perhaps a good idea to digress briefly and explain what they signify. Using a single address line, you could distinguish between two addresses. With two lines, you get an 'address range' of four addresses; three lines define eight addresses, and so on. By the time you get up to ten lines, you find that you can distinguish between 1024 adresses. This is referred to as a '1 K block'. Since it is slightly more than one thousand, we use a capital K. It's rather like the difference between Imperial and US gallons: they're both gallons, but one is slightly more than the other. Similarly, the 1 Mbyte address range of the 8088 is slightly more than one million addresses: 20 address lines define 1,048,576 addresses.

Back to our 'problem': how can 16 K be equal to 1 M? Fourteen address lines define a 16 K block; of these lines, the highest three (A11 . . . A13) go to the address decoder. All higher address lines are simply ignored! This means that the address decoder can't see any difference between addresses 0000, 04000, 08000 and so on. This can be seen in table 1, where the actual bits on the various address lines are shown for these addresses. Reading from left to right, these bits are used as follows:

- A19 . . . A14 are ignored. They can have any value, without making any difference to the actual memory location that is selected.
- A13 . . . A11 go to the address decoder. They define eight 2 K blocks; the highest two enable the EPROMs, and the lowest 2 K block is for the RAM.
- A10 . . . A0 define the 2048 addresses in each 2 K block. In the lowest (RAM) block, A10 is also ignored; this means that the same RAM is addressed in both the first and second 1 K block (these are referred to as 'RAM' and 'RAM duplicate', respectively).

The answer to the 'problem' should now be clear: the basic 16 K address range is simply duplicated 64 times in the total 1 Mbyte range, as shown in figure 2. After reset, the processor looks at address FFFF0. The address decoder looks at lines A11 to A13, finds them all at logic 1, and enables the first EPROM. Exactly the same result would be obtained if the processor tried to address 'location 03FF0'.

Interface

Communication with the outside world runs over a simple RS-232 interface (T1 and T2). The 'receiver' is a single transistor that converts the input signal levels to TTL logic levels:

−12 . . . −5 V = logic 1 → +5 V;
+5 . . . +12 V = logic 0 → 0 V.

Diode D5 protects the transistor when the input signal swings negative.

The 'transmitter' end is also a single transistor. This one operates as a voltage-to-current converter, which

3

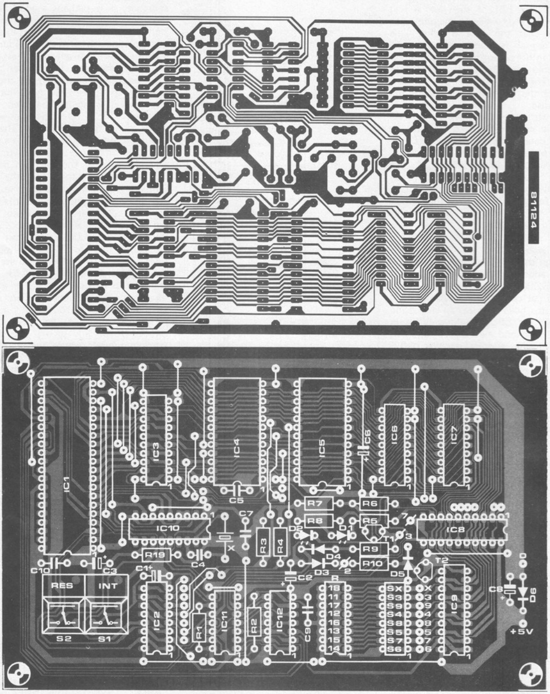


Figure 3. The printed circuit board. Particular care should be taken with the wire links: 43 are required, at least. If the interrupt key is omitted, a further wire link is required to connect what would otherwise have been the centre contact pins, as shown. The same applies for the reset key, if this is mounted off the board or replaced by a type that does not contain this internal connection. Where several wire link options are possible, for other memory ICs, only the correct link is shown.

Parts list

Resistors:
R1,R19 = 10 k
R2,R3,R4 = 1 k
R5 = 220 Ω
R6,R7,R9 = 1k8
R8 = 47 Ω
R10 = 4k7
R11 . . . R18 = 8 x 4k7 (or 16-pin DIP resistor network)

Capacitors:
C1,C3 = 1 μ/16 V Tantalum
C2,C6,C8 = 10 μ/16 V Tantalum
C4 = 10 p
C5,C7,C9,C10 = 100 n

Semiconductors:
D1 = LED
D2 = LED (red)
D3 . . . D5 = 1N4148
D6 = 1N4001 or surge diode (TVS 505, for instance)
T1 = BC 557B
T2 = BC 547B
IC1 = 8088
IC2 = 74LS156
IC3,IC8 = 74LS373
IC4,IC5 = 2716 EPROM 450 ns
IC6,IC7 = 2114 RAM 450 ns
IC9 = 74LS244
IC10 = 8284
IC11 = 74LS14
IC12 = 74LS00

Miscellaneous:
S1 = digitast switch or wire link (see text)
S2 = digitast switch
S3 . . . S9, Sx = 14- or 16-pin DIP switch (or wire links, see text)
X1 = 15 MHz crystal small size HC-18/U

Table 1.

Address	Address (binary)																			
(hexadecimal)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 3 F F F	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0 4 0 0 0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 8 0 0 0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F C 0 0 0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F F F F 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Table 1. Of the 20 address lines, only the lower 14 are actually used. A11 . . . A13 are passed to the address decoder, to determine which memory chip must be enabled. The logic levels of the higher address bits (A14 . . . A19) are irrelevant.

Table 2.

Baud rate	S_6	S_5	S_4
9600	0	0	0
4800	0	0	1
2400	0	1	0
1200	0	1	1
600	1	0	0
300	1	0	1
150	1	1	0
110	1	1	1

Table 2. The baud rate (transmission speed in bits per second) is determined by the setting of three of the switches in the DIP switch block. If a fixed rate is sufficient, wire links can be used instead.

4

```
8  RR  NN  BB  QQ  KK  BB  NN  RR

7  PP  PP  PP  PP  PP  PP  PP  PP

6  :::     :::     :::     :::

5      :::     :::     :::     :::

4  :::     :::     :::     :::

3      :::     :::     :::     :::

2  P   P   P   P   P   P   P   P

1  R   N   B   Q   K   B   N   R

   A   B   C   D   E   F   G   H
                              81124 4
```

Figure 4. The initial board position, as it will appear on the screen. The single letters (R, N, B, . . .) are white pieces; the letter pairs are black; the dots (:::) indicate a vacant white square.

automatically makes it short-circuit proof. LED D2 is used to set the base voltage — it will barely light, since the current through it is only 2 mA. To meet the RS-232 standard, a negative output voltage is also required. Since Intelekt only uses a positive supply, a little trick is used. The *input* signal, coming from the terminal, printer or whatever, swings between positive and negative levels. This signal is rectified (by D3, D4 and C2) to provide the negative 'supply' for the output.

A second output from IC8 drives LED D1. This LED flashes on and off when the chess program is running. Regular and fairly rapid flashes indicate that he is waiting for you to enter data; slower flashes (corresponding to the depth of the 'search') will appear when he is thinking.

Of the eight inputs to IC9, one is used for the RS-232 input. The others can be connected to a DIP switch; this is a unit that contains seven or eight miniature switches, and can be plugged into a normal IC socket. Note that, if an eight-switch version is used, the lower switch should not be closed — otherwise it would short the RS-232 input to ground!. Three of the switches set the baud rate, as listed in table 2. Obviously, for a fixed baud rate (=transmission speed to and from the terminal) wire links can be used instead of the switches.

Construction

The printed circuit board is shown in figure 3. To keep the cost down to a reasonable level, it was decided to use a single-sided board. This does lead to a larger number of wire links. There are 43 in all, and it's worth counting them before switching on for the first time!

The possibility of future extensions was also considered, and some points were brought out even though Intelekt doesn't use them. However, this does not mean that the board can be used as the basis for an extensive system: the bus is not buffered, and the addresses are not fully decoded. The only possible extensions we have in mind are the use of other EPROMs (or ROMs) with a 4 K range, and extension of the RAM area by substituting a 4118, say, for one of the EPROMs. In general, the flexibility that the board offers is only intended to facilitate its use in other small-system applications.

The main wire links to watch in this connection are:

- those at each EPROM socket: they determine whether a 2716, 2732 or 4118 can be used — for Intelekt, the '2716' link is used.
- the chip enable (CE) inputs to the EPROMs and RAM are connected to the address decoder as required; for Intelekt, EPROM 1 is driven from output 7, EPROM 2 from output 6 and RAM from output 0.
- the DIP switch (or wire links) set the baud rate; it is set according to table 2.

Two digitast switches are also mounted on the board. Note that, on this type of switch, the centre contact is brought out onto two pins. *This fact is used on the board to connect 'supply common' to a whole section of board. If other types of switches are used, or if the switches are mounted off-board, two further wire links will be required at this point!*

Communicating with Intelekt

As explained earlier, Intelekt is controlled via the keyboard of a terminal and he 'talks back' by means of the associated display (TV screen or printer). To get some idea of how this works in practice, assume that Intelekt is connected to the Elekterminal.

After switching on, first operate the Reset key on the chess computer itself — note that this is the only command that is not entered via the Elekterminal keyboard. Intelekt will respond by displaying the following message:

TINY CHESS VI.O
LEVEL IS 1 CHANGE TO _

You can now enter a '1', followed by Carriage Return — the 'why' of this will be explained later on. The chess board will now appear on the screen, in the initial position as shown in figure 4. The single letters (R, N, B, . . .) stand for the white pieces, the letter pairs (RR, NN, etc) are black pieces and the dots (:::) are empty white squares on the board. Intelekt then asks for your first move:

01W:

To enter your move, key in:

- the square containing the piece that is to be moved;
- a space;
- the square to which the piece is to be moved;
- Carriage Return.

Intelekt will now check whether or not you have entered a legal move (if not, he will request a new entry) and then proceed to calculate his response. Initially he will invariably find a response in his 'book of standard openings' and reply immediately. Later on, when he has to start thinking out his moves, the response time can vary from 25 seconds (lowest level of skill) up to several hours (highest level). Having worked out his move, Intelekt will print it on the screen, and immediately display the new position on the board. The result so far could look like this (bold type: your moves):

TINY CHESS VI.O
LEVEL IS 1 CHANGE TO **1**
(initial board situation)

01W: **e2 e5 (CR)** — an illegal move, so:
01W: **e2 e4 (CR)**
01B : c7 c5
(new board situation)
02W: - waiting for your next move.

If you notice a typing error before entering Carriage Return, it is possible to correct this by operating 'Backspace'. However, if it was a legal move and you have already typed Carriage Return, there is no easy way to correct it. If you really want to cheat, it is possible to enter an illegal move, provided you terminate it with 'Line Feed' instead of 'Carriage Return'.

When it is your move, you can also enter one of the following instructions:
Control X: change players. You now play black; he responds by printing 'my move'. After he has made his move, you can change back to playing white by again entering Control X.
Control A: autoplay. He plays both sides!
Control N: to set the 'number' of the level that you want to play. He responds with 'Level is 1 change to _' (assuming that you were at level 1); you can then enter any number between 1 and 8, followed by Carriage Return. Level 1 is the easiest and level 8 the most difficult. We found level 3 to be a good compromise between response time (about 5 minutes, on average) and skill.
Control C: change board mode. Intelekt responds with a dash prompt: '–'; you can now enter one of several commands:

- erase the board (remove all pieces) by entering Control E;
- change any square, as follows:
  - enter the number of the square (b5, say); Intelekt responds by printing what is on that square;
  - if desired, update the square by entering either a colon (:) to empty the square, or a single letter (K, Q, R, B, N or P) for the corresponding White piece, or two letters (KK, QQ etc.) for a Black piece;
  - enter a Space to step to the next square (whether or not you have updated the preceding one);
  - enter Carriage Return when a sequence of squares has been updated. Intelekt again responds with a dash prompt, waiting for you to enter a new square.
- after editing the board, return to normal mode by entering Carriage Return. It may be worth noting that Intelekt will refuse to play unless there are a Black and White king on the board . . .

All the commands listed above can only be entered when it is your turn. So what do you do when Intelekt is thinking? You can 'interrupt' him by entering *Break* or *several spaces.* This has the same effect as the Control N instruction described above: you can change the level of play. By entering a lower level (level 1, say) you can ensure that it will be your turn within half a minute; at that point you can of course enter any command.

Reset: This resets the board and program for a new game.

Special moves

Entering 'normal' moves was explained above. For those who are not so familiar with the numbering of the squares (A to H left to right, and 1 to 8 from bottom to top), each board print-out includes these letters and numbers. There are also a few special moves: castling, en passant taking of a pawn, check and pawn promotion. All of these possibilies are known to Intelekt. They are dealt with as follows:
Castling: only enter the move for the king. Intelekt will interpret this correctly, check whether or not it is permissable and then move both king and rook accordingly.
En-passant: this move is not as well known as it ought to be. To put it in a nutshell: when a pawn is initially moved up two squares, passing a square that is attacked by a pawn, it can be taken at the next move by that pawn. As an example, assume that Black has a pawn on b4. If white moves a pawn a2-a4, Black can take it immediately by moving b4-a3. To execute this move, you would simply enter 'b4-a3'; Intelekt will know what is meant.
Check: Intelekt will print a warning when it places you in Check (or Checkmate); it then rejects any move that doesn't remove your king from check. Stalemate is also recognised.
Pawn promotion: It is presumed that when you promote a pawn, you want a queen; and Intelekt calculates its moves on this basis (a minor 'blind spot'). If you want anything else, this can be obtained via the 'change board' mode.

A few games

Three complete games are given in tables 3 . . . 5. In the first, fairly straightforward game, Intelekt played black; in the second, he played white. In the third game, Intelekt again played black; furthermore, in this game white made a deliberate effort to 'draw out' the machine as far as possible before striking back — too late, as things turned out . . .

Obviously, it would take up too much space to examine each game in great detail. However, if you are interested in playing out each game according to the moves listed in the corresponding table, we will attempt to pick out the interesting highlights.

Game 1

The first two moves were according to his opening 'book': Black's response was immediate. White's third move *(g2-g3)* put a stop to this; from now on, Intelekt must start thinking for himself . . .

After some manoeuvering and minor skirmishes, White's move *13. f2-f4* was a deliberate attempt to make things complicated. If, on the next move, White takes one of Black's pawns (*f4xe5* or *f4xg5*) the rook on f1 would attack Black's queen and things would start to happen . . . Black can't take the pawn by playing *e5xf4*, and *g5xf4* opens a lot of interesting possibilities.

In fact, things developed nicely. Then, at the 17th move, White was faced with the choice: *Nd5-f4* or attempt to break up Black's central pawn formation? He chose the latter option, but it didn't quite work out as planned . . . Not yet, anyway.

Moves 20 and following may seem rather strange at first sight. *20 Ra1-d1* is safe enough: Black can't play *Bh5xd1*, since this would be followed by *Qf2xf7 mate*! To remove this threat, Black tried *f7-f5*. This led to the loss of a pawn, and White even got the opportunity to continue the

Table 3.

1.	e2-e4	c7-c5
2.	c2-c4	e7-e5
3.	g2-g3	d7-d6
4.	Bf1-g2	a7-a6
5.	Nb1-c3	Nb8-c6
6.	d2-d3	Nc6-d4
7.	Ng1-e2	Bc8-d7
8.	O - O	Bd7-g4
9.	b2-b3	Nd4xe2†
10.	Nc3xe2	Qd8-f6
11.	Bc1-b2	g7-g5
12.	Qd1-d2	Bg4-e6
13.	f2-f4	g5xf4
14.	Ne2xf4	Be6-g4
15.	Nf4-d5	Qf6-d8
16.	Qd2-f2	Bg4-h5
17.	b3-b4	c5xb4
18.	d3-d4	b4-b3?
19.	a2xb3	Bf8-g7
20.	Ra1-d1	f7-f5
21.	Qf2xf5	Bh5-g6
22.	Qf5-f2	Bg6-h5
23.	g3-g4	Bh5-g6
24.	h2-h4	Ng8-h6
25.	g4-g5	Nh6-g8?
26.	d4xe5	Bg6-h5
27.	e5xd6	Bg7xb2
28.	Qf2xb2	Bh5xd1
29.	Qb2xh8	Bd1xb3
30.	Qh8xg8†?	Ke8-d7
31.	Qg8xh7†	Kd7xd6
32.	Nd5-b6?	Qd8xb6†
33.	Kg1-h1	Bd3xc4?
34.	Rf1-f6†	Bc4-e6
35.	Rf6xe6†!	Kd6xe6
36.	Qh7-h6†	Ke6-f7
37.	Qh6xb6	. . . . .
	"I knew that"	
	. . . . .	Ra8-h8
38.	Qb6-f6†	Kf7-g8
39.	g5-g6	Rh8xh4†
40.	Qf6xh4	Kg8-g7
41.	Qh4-h7†	. . . . .
	"GRRR"	
	. . . . .	Kg7-f6
42.	g6-g7	b7-b5
43.	g7-g8 (Q)	. . : . .
	"I knew that"	
	. . . . .	"I give up"

Table 3. The first game, with Intelekt playing black.

'mopping up' action in the centre (moves 26 and following). During a momentary lull in the battle, Black decided it would like to take a pawn (*29 . . . Bd1xb3*), leaving White with so many options that he didn't know which to choose! *Nd5-c7*†, followed by *Qh8xg8*†, might well win a rook. On the other hand, *Qh8xg8*† seems quite promising already. Or *Qh8xh7*? Or *Rf1-b1*? Or *e4-e5*? The game had already lasted three hours, so White decided to pick one alternative at random . . .

32 Nd5-b6 was a mistake, pure and simple. The idea was to pin things in that corner *(Qd8xb6* was to be followed by *Rf1-f6*, and if Black tried to save this rook, White could follow up with *Rf1-d1*), but White forgot that Qb6 gives check! Amazingly, Intelekt offered this option one move later — apparently assuming that *Bc4-e6* would be a sufficient answer. It wasn't, as moves 35 . . . 37 show.

What followed was just a fairly brutal end-game, punctuated by various comments from Intelekt.

Game 2

By coincidence, this game (with Intelekt playing White) developed along the same initial lines as the previous one. As before, Black's third move (*g7-g6*) put an end to Intelekt's use of his 'opening book'. From here on, the game progressed in a fairly conventional manner, until things started to happen around the tenth move. After the dust had cleared (at the fourteenth move), Black was two pawns up and had a considerable positional advantage.

As things seemed to be developing in Black's favour, White obviously decided that it would be wise to exchange queens (moves 18 and 19). There didn't seem to be much point in avoiding this exchange . . . From here on, things again developed slowly but surely. For the fun of it, Black tried a little trap at move 31 (*e5-e4*†): *32 Kd3xe4* would have been followed by *Nc4-d2*†, winning a rook. As expected, White saw this trap and avoided it.

Some further manoeuvring (up to and including move 39) led to a position where most of the remaining pieces were tied up in one corner, leaving the game essentially as a standard two-pawn-against-one-pawn end-game. Surprisingly, White gave the impression that it was going to chase away Black's rook with its king, so Black decided to simply promote the pawn without taking further precautions (move 43). However, the White king came back — prolonging the agony. By move 49, the end result was clear (even to Intelekt); in sheer desperation he tried sacrificing his bishop. To no avail.

If you think it looks easy to 'beat the monster', the following may prove interesting. At move 19 in this game, the situation was as shown in figure 5. At this point, Black seriously considered playing *19 . . . Qa5-c3*. For various reasons (mainly that it didn't 'feel' right), he chose the alternative given earlier (*e4xd3*). However, since it is very easy to set up any position when playing against Intelekt, the second-best choice was also tried later on. The results are shown in the continuation of table 4 . . .

The key move was *23 Qb5-e8*†, leading to mate in four moves — or so it seems. However, when it came to move 25 he played *Qe7xh7*† instead of *Rh1xh7* followed by mate (*. . . Qc1xc4*†*, 26 Nd2xc4, Ng4-f6, Qe7-g7 mate*). Prolonging the agony? Or was this 'beyond his horizon'? Or did he see greater danger in *. . . d4-d3*†, which opens new possibilities for harassment by Black? However this may be, Black now has a small reprieve; he even gets a chance to set a small trap: *29 . . . Qc1-a3*. It seemed conceivable that White might play *Qd4-c3*, which could be followed by *Ng4xf2* — winning a rook). However, no such luck. Even though it took some time, Intelekt succeeded in ending the game. To be quite honest: his end-game could do with some improvement! He gets there in the long run, but it could often be a lot shorter . . .

Table 4.

1.	e2-e4	e7-e5
2.	c2-c4	c7-c5
3.	d2-d3	g7-g6
4.	h2-h3	Bf8-g7
5.	h3-h4	Ng8-f6
6.	g2-g3	d7-d6
7.	Qd1-a4†	Nb8-c6
8.	Bc1-g5	O - O
9.	h4-h5	Qd8-b6
10.	h5-h6	Bg7xh6!
11.	Bg5xh6	Qb6xb2
12.	Bh6xf8	Qb2xa1
13.	Bf8xd6?	Qa1xb1†
14.	Qa4-d1	Qb1xa2
15.	Ng1-f3	Nf6-g4
16.	Nf3-d2	f7-f5
17.	Bd6xc5	f5xe4
18.	Qd1-b1	Qa2-a5
19.	Qb1-b5	e4xd3
20.	Qb5xa5	Nc6xa5
	"I knew that"	
21.	Bf1xd3	Bc8-e6
22.	Bc5-b4	Na5-c6
23.	Bb4-c5	Ra8-c8
24.	Nd2-f3	Nc6-a5
25.	Bc5xa7	Na5xc4
26.	Nf3-g5	Nc4-b2
27.	Ke1-d2	Be6-c4
28.	Bd3xc4†	Nb2xc4†
29.	Kd2-d3	h7-h5
30.	Ng5-e6	b7-b6
31.	Rh1-f1	e5-e4†
32.	Kd3-d4	e4-e3
33.	f2xe3	Ng4xe3
34.	Rf1-f3	Rc8-e8
35.	Ne6-c7	Re8-e7
36.	Nc7-b5	Re7-d7†
37.	Kd4-e4	Nc4-d2†!
38.	Ke4xe3	Nd2xf3
39.	Ke3xf3	Rd7-b7
40.	g3-g4	Kg8-h7
41.	g4xh5	g6xh5
42.	Kf3-f4	Kh7-g6
43.	Kf4-e5?	h5-h4
44.	Ke5-f4!	Kg6-h5
45.	Kf4-f3	Kh5-g5
46.	Kf3-g2	Kg5-g4
47.	Kg2-h2	h4-h3
48.	Kh2-h1	Kg4-g3
49.	Ba7xb6!	Rb7xb6
50.	Nb5-c3	Rb6-e6
	"I give up"	

Mistakes are not permitted. This is illustrated by the following alternative play from move 19:

19.	Qb1-b5	Qa5-c3?
20.	d3xe4	Qc3-c1†
21.	Ke1-e2	Nc6-d4†??
22.	Bc5xd4	e5xd4
23.	Qb5-e8†!!	Kg8-g7
24.	Qe8-e7†	Kg7-g8
25.	Qe7xh7†	Kg8-f8
26.	Qh7-h8†	Kf8-f7
27.	Qh8xd4	Bc8-f5
28.	e4xf5	Ra8-e8†
29.	Ke2-d3	Qc1-a3†
30.	Kd3-c2	Qa3-a4†
31.	Nd2-b3	Re8-c8
32.	Rh1-h7†	Kf7-g8
33.	Qd4-g7†	Mate.

Table 4. In this game, Intelekt played white. From the nineteenth move on, two variations were tried.

5

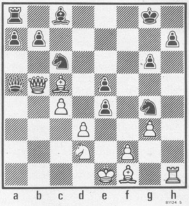


Figure 5. In the second game, this position was reached after White's 19th move. An alternative play was tried from this point.

Game 3

This game proved quite interesting. Intelekt played Black, and White deliberately attempted to 'draw him out'. As it proved, giving Intelekt an advantage is fatal — even at level 3!

The surprises started at move 7: *Nd4-f3*†! Suicide? Not at all, as the continuation proves. To compound the misery, White's thirteenth move was an out-and-out mistake, and Black's response was immediate. Things start to get hectic at this point, culminating in Black's clincher at the twentieth move: *Bf4-h2*. He didn't take the rook — he was after greater glory. Greeting White's only response (*Rg1xg2*) with the comment 'Dummy!' was adding insult to injury. White now decided to get vicious, but it didn't help.

What follows is relatively uninspired. White tried a little trap at moves 42 and 43: the idea was to follow up with *44 Bg5-e3*, winning the pawn on h5. It didn't work. By the fiftieth move, White didn't feel like doing any hard thinking. *51 c6-c7* might have been better than *Kd6-e7*; and *52 Ke7-d7* might be better than *Ke7-d8*. However: the result seems a foregone conclusion, no matter what. Moves 57 . . . 60 offer a clear opportunity for a draw, on the basis of repeated moves; however, White would like to see how Intelekt wins this game. It's a disappointment. Even with one rook up, he can't work out a clear strategy. After move 73, White had had enough — and switched to autoplay. After move 79 it seemed a good idea to go to bed, and have a look in the morning to see what had developed. Nothing! He was still playing ring around the roses. So I pulled the plug.

The moral of the story: Intelekt plays a good game, but he doesn't always know how to win in an end-game. This is a fairly common failing with chess computers. Fortunately, it doesn't detract much from their charm: the fun is in fighting the game until there is a clear win for one of the two sides. At that point, it would be normal to give up. Playing on against a human opponent would lead to a fairly quick death, so you don't do it; playing on against Intelekt may lead to an endless game, so you don't do that either.

In conclusion

In earlier articles ('How I beat the monster' and 'Computers and Chess', Elektor January 1979), we examined the operating principles and common failings of chess-playing computers. Basically, Intelekt uses the brute force of a fast 16-bit microprocessor to overcome the shortcomings of a straightforward 'mini-max' procedure. Surprisingly enough, this approach works! For most 'average' human chess players, he presents a good challenge at reasonably short response times. In fact, for most of us poor mortals it is fortunate that his 'creator' didn't add sophisticated short-cuts in the program. It would be just too humiliating to be wiped up by a handful of electronics that only took a few seconds to work out a move!

However, there is room for improvement. We have already discussed the possibilities of improving the end-game and including a survey of pawn promotion to other pieces than a queen. Both would involve additional memory and slower response — as things stand at present. But Intelekt's skills are stored in EPROM — and this can be exchanged, at a later date, for a more sophisticated program! Who knows?

Intelekt is intended as a chess opponent. He doesn't like solving chess problems on his own. So what? Playing chess is fun, but getting someone (or something) else to solve chess problems for you is as pointless as looking for a Scrabble opponent who will solve crossword puzzles. We got to like Intelekt — he has a charm of his own. If you like chess, you should make his aquaintance.

Table 5.

1.	e2-e4	c7-c5
2.	g2-g3	e7-e5
3.	Bf1-g2	d7-d6
4.	b2-b3	Nb8-c6
5.	Bc1-b2	Nc6-d4
6.	Ng1-e2	Bc8-g4
7.	Nb1-c3	Nd4-f3†!
8.	Bg2xf3	Bg4xf3
9.	Rh1-f1	Ng8-f6
10.	d2-d3	h7-h5
11.	h2-h4	Qd8-a5
12.	Qd1-d2	g7-g6
13.	O-O-O	Bf8-h6!
14.	Ne2-f4	e5xf4
15.	Nc3-e2	f4xg3
16.	Ne2-f4	Qa5xd2†
17.	Rd1xd2	. . . . .
	"I knew that"	
	. . . . .	g3-g2
18.	Rf1-g1	Bh6xf4
19.	Bb2xf6	O - O!
20.	Bf6-g5	Bf4-h2!
21.	Rg1xg2	. . . . .
	"Dummy!"	
	. . . . .	Bf3xg2
22.	f2-f4	f7-f6
23.	Bg5-h6	Bh2xf4
24.	Bh6xf4	Bg2-h3
25.	Bf4xd6	Rf8-c8
26.	Rd2-f2	f6-f5
27.	e4xf5	Bh3xf5
28.	Rf2-g2	Rc8-e8?
29.	Bd6xc5	Re8-c8
30.	b3-b4	Kg8-h7
31.	Kc1-d2	Bf5-e6
32.	c2-c4	a7-a5
33.	a2-a3	Be6-h3
34.	Rg2-e2	a5xb4
35.	Re2-e7†	Kh7-h6
36.	Bc5-e3†	g6-g5
37.	Be3xg5†	Kh6-g6
38.	a3xb4	Bh3-g2
39.	Kd2-c3	Rc8-f8
40.	Re7-e6†	Kg6-g7
41.	c4-c5	Rf8-f2
42.	Re6-e5	Bg2-c6
43.	Kc3-d4	Rf2-f1
44.	b4-b5	Bc6xb5
45.	Re5-e7†	Kg7-g6?
46.	Re7xb7	Bb5-c6
47.	Rb7-b6	Ra8-a4†
48.	Kd4-e5	Rf1-e1†
49.	Ke5-d6	Bc6-f3
50.	c5-c6	Re1-d1
51.	Kd6-e7?	Rd1-e1†
52.	Ke7-d8?	Ra4-d4†
53.	Kd8-c8	Re1-e8†
54.	Kc8-b7	Bf3-g2?
55.	Kb7-a7	Rd4xd3
56.	c6-c7†!	Kg6-f5
57.	Rb6-b8!	Rd3-a3†
58.	Ka7-b6	Ra3-b3†
59.	Kb6-a7	Rb3-a3†
60.	Ka7-b6	Ra3-b3†
61.	Kb6-c5	Re8xb8
62.	c7xb8	Rb3xb8
63.	Kc5-d4	Bg2-c6?
64.	Kd4-c5	Bc6-a4
65.	Kc5-d6	Kf5-e4
66.	Kd6-e6	Ba4-c6
67.	Ke6-f6	Ke4-d3
68.	Kf6-g7	Kd3-e2
69.	Kg7-h6	Bc6-f3
70.	Bg5-f4	Rb8-b6†
71.	Kh6-g5	Ke2-d3
72.	Bf4-c7	Rb6-b5†
73.	Kg5-f4	Kd3-e2
Autoplay:		
74.	Bc7-e5	Bf3-d5
75.	Be5-c3	Rb5-b1
76.	Kf4-f5	Rb1-g1
77.	Bc3-d4	Rg1-c1
78.	Kf5-g6	Ke2-d3
79.	Bd4-e5	Rc1-f1
	. . . . .	. . . . .
	. . . .	. . . . .
	. . . . .	. . . . .
148.	Kg6-g5	. . . . .
	giving the board situation:	

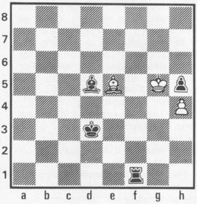
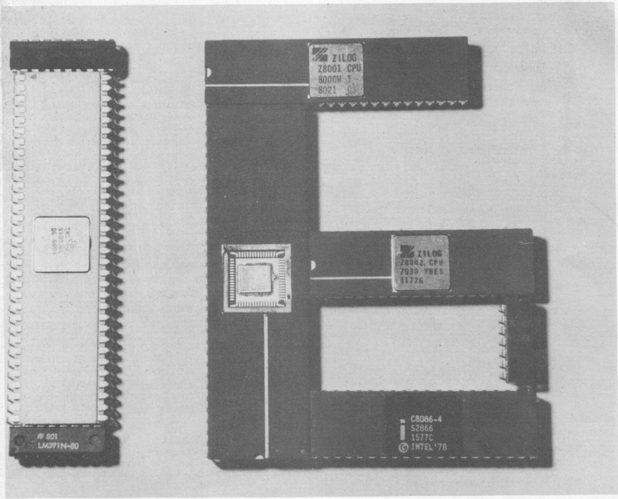


Table 5. The computer plays black, and white tries a deliberately 'passive' game to provoke Intelekt into attacking. This proved fatal . . .

David Levy: 'How I beat the monster'
Elektor, January 1979, p. 140
'Computers and Chess'
Elektor, January 1979, p. 134
'16-bit microprocessors'
Special supplement in this issue.



16-bit microprocessors

**"16 bits, and what do you get?
Another day older, and deeper in debt!"**

It seems only a few years ago that we were getting used to the idea of a complete computer on a single chip. At the time, only very simple 'computers' were involved: microprocessors made sophisticated systems controllers, but they were vastly inferior to 'true' minicomputers.
Now, things are changing. Rapidly. The new generation of 16-bit 'micro'processor systems can equal or even better the performance of present-day 'minis'. This means that a fully-fledged personal computer is now within reach of any enthusiast. The only question is: which system do you choose?
This article is intended as a brief survey of the field. As was apparent in the past, and will become even more apparent in future articles, our personal conclusion is: you can make almost any processor do almost any job. So you just pick the one that happens to come to hand, or that appeals to you most for personal reasons.

From valve to transistor . . .
From transistor to (TTL) IC . . .
From TTL IC to CMOS . . .
From CMOS to microprocessor . . .
And now: 16-bit processors!

All this in the space of some thirty years. No wonder a lot of electronics enthusiasts have lost track of developments! However, they are still interested: we often get letters with comments like 'I wish I understood what it's all about!'. In this article, we will attempt the impossible: to give a general impression of what these super-microprocessors are, and at the same time compare their capabilities in greater depth for the benefit of microprocessor enthusiasts.

What is a 16-bit microprocessor?

Funnily enough, it is not at all easy to decide whether a particular type belongs in this catagory or not — for reasons that will be explained. However, we can make some general assumptions that broadly define the group.
A 16-bit digital 'word' defines more than 64,000 numbers — from −32,000 to +32,000, for instance. This is considerably more precise than the 256 numbers defined by the older 8-bit microprocessors (or 'μPs'). Given this large numerical range, it becomes worthwhile to make the processor 'do sums'. The new 16-bit μPs can not only add and subtract (as the 8-bitters could); they can also be instructed to multiply and divide.
Broadly speaking, any computer system can be subdivided into a few distinct sections: inputs and outputs (keyboard, display, control lines etc.); memory (containing both the program that must be run and the data involved); and the 'central processing unit'. This 'CPU' moves data to and fro as required and performs the necessary operations (add, subtract, multiply, divide, AND, OR, EXOR, etc.) as well as ensuring that all these data movements and operations are carried out in the correct sequence, as specified by the program. Quite a job, you would think; but in fact these 16-bit microprocessor chips do all this and, very often, more. Any comparison of 16-bit processors must therefore take several things into account:

- what 'operations' can they do (arithmetic, logic, etc.)?
- how large a memory can they handle comfortably, and what possibilities do they offer for moving data to and from this memory?
- what options do they offer to the programmer (jumps, loops, subroutines, etc.)?

A further general point regarding computer systems is that they tend to grow. As more and more memory and 'peripheral' (input/output) devices are added, some further points become important when comparing microprocessors:

- how easily can they cater for external devices that want to break into the program ('interrupt' it) at awkward moments?
- how willingly can they co-operate with other microprocessors, sharing the same resources (memory, peripheral devices, etc.)? Bearing in mind that these 'resources' often constitute the bulk of the cost in a computer system, using several processors in one system ('multi-processor operation') often makes very good sense!
- how fast are they? As systems grow, programs may well become more complicated. Doing a complete division in 40 μseconds may seem fast, but when you have a program that requires umpteen thousand calculations and data movements, time does tend to run out . . . Think of the chess computers that may take several hours to work out one complicated move!

Back now to the question: what is a 16-bit microprocessor? As a first generalisation, it seems 'logical' to say: any processor that provides the main features outlined above and works with 16-bit data. But there's the rub. Several processors work with 16-bit data *inside* the processor chip itself, but only move data in 8-bit 'bytes' — two consecutive bytes are required for one 16-bit word. Is this still a 16-bit processor? We are inclined to say: yes, in a sort of a way. After all, it does the same job — even if it takes twice as long to move the data around. But in that case, do you also include processors that work with 32-bit data inside the CPU and move data in 16-bit chunks? Or are they 32-bitters? The Motorola MC 68000, for instance, was defined by somebody as 'a 32-bit microprocessor, masquerading as a 16-bit CPU'.
For the purposes of this article, we have simply drawn up a list of processors that 'seem to belong in the category', at first sight. This gives us the eleven main types listed in table 1. For various reasons (price, intended application) this list was 'pruned', leaving us with the short list given in table 2: the μPs that are of primary interest to (amateur) enthusiasts. Variations on these five main types are listed in table 3, together with some comparative information.

First impressions

There are two distinct tendencies in 16-bit μP design: on the one hand, upgrading from 8-bit processors; on the other, 'downgrading' from minicomputers. To some extent, both of these tendencies can influence the design of the same microprocessor. Motorola and Zilog, for instance, have both based their instruction set on an analysis of 'most frequently used instructions'. Depending on how the balance is struck between these two

Table 1

16-bit microprocessors, main types

type	originator	process	application
MN 601	Data General	NMOS	minicomputer OEMs
9440	Fairchild	I^2L	minicomputer OEMs
F100L	Ferranti	Bipolar	military
CP 1600	General Instruments	NMOS	electronic games
8086	Intel	HMOS	general-purpose μP
MC 68000	Motorola	NMOS	general-purpose μP
NS 16032	National Semiconductor	XMOS	general-purpose μP
MN 1610	Panafacom	NMOS	?
TMS 9900	Texas Instruments	NMOS	general-purpose μP
WD 16	Western Digital	NMOS	minicomputer OEMs
Z 8001	Zilog	NMOS	general-purpose μP

Table 2

16-bit microprocessors, short list

type	manufacturers
8086	**Intel**, Mitsubishi, Mostek, Siemens
68000	**Motorola**, Hitachi, Rockwell, Thomson
16032	**National Semiconductor**, Fairchild
9900	**Texas Instruments**, AMI, ITT
8001	**Zilog**, AMD, SGS-Ates

Table 3a

main type	derived types*	data length in CPU/bus	address range bus/memory/with support	data/address bus multiplexed	derivation
8086		16/16 bits	20 bit/1 Mbyte/1 Mbyte	yes	upgrade from 8080 (+ downgrade from mini's)
	8080	16/8 bits			
68000		32/16 bits	23 bit/16 Mbyte/64 Mbyte	no	upgrade from 6800; downgrade from mini's
16032		32/16 bits	24 bit/16 Mbyte/	yes	upgrade from 8080; downgrade from mini's
	16016	16/16 bits	16 bit/64 Kbyte/		
	16008	16/8 bits	16 bit/64 Kbyte/		
9900		16/16 bits	15 bit/64 Kbyte/	no	downgrade from minicomputers
	9940	no external data/address bus; 2 Kbyte RAM/ROM on chip			
	9980/9981	16/8 bits	14 bit/16 Kbyte		
	9995	16/8 bits	15 bit/64 Kbyte/		
8001		16/16 bits	23 bit/8 Mbyte/48 Mbyte	yes	upgrade from Z80; downgrade from mini's
	8002		16 bit/64 Kbyte/384 Kbyte		
	8003		as 8001		
	8004		as 8002		

* for derived types, only differences with respect to main type are listed.

Table 3b

main type	derived types	registers		data stored in memory*	clock frequency	shortest** instruction	longest** instruction
		general purpose	dedicated + control				
8086		–	14 (16-bit)	low-high	8/5/4 MHz	0.25 µs	20 µs (①)
	8088				5 MHz	0.4 µs	32 µs (①)
68000			18 (32-bit), 1 (16-bit)	high-low	8/6/4 MHz	0.5 µs	20 µs (②)
16032		8 (32-bit)	6 (24-bit), 2 (16-bit)	low-high	10 MHz	0.3 µs	8 µs (②)
	16016	8 (16-bit)	8 (16-bit)				
	16008	8 (16-bit)	8 (16-bit)				
9900		16 (16-bit)***	3 (16-bit)	high-low	3.3/4 MHz	2 µs	31 µs (①)
	9980/ 9981				2.5 MHz	2.6 µs	41 µs (①)
	9995				6 MHz	1.1 µs	17 µs (①)
8001		16 (16-bit)	7 (16-bit)	high-low	6/4 MHz	0.5 µs	140 µs (③)
	8002	16 (16-bit)	4 (16-bit)		6/4 MHz	0.5 µs	(19 µs (②))
	8003				10 MHz	0.3 µs	80 µs (③)
	8004				10 MHz	0.3 µs	(11 µs (②))

* 'low-high': least significant byte at lower address; 'high-low': most significant byte first.

** at highest permissible clock frequency

*** these registers are located in RAM, not in the CPU

① unsigned divide, (32-bit) ÷ (16-bit) = 16-bit result + 16-bit remainder

② signed divide, (32-bit) ÷ (16-bit) = 16-bit result + 16-bit remainder

③ signed divide, (64-bit) ÷ (32-bit) = 32-bit result + 32-bit remainder

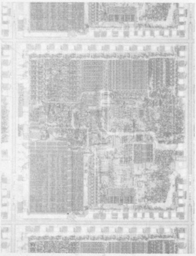
Table 3c

main type	derived types	interrupt types				I/O area	instruction queue	ABORT for virtual memory
		NMI	traps	non-vect.	vectored			
8086		1	4	–	251	64 Kbyte	6 byte	no
	8088						4 byte	
68000		–	27	–	227	*	no	no
16032		1	9	1	240	*	8 byte	yes
	16016							
	16008							
9900		2	16	–	15	4 Kbit	no	no
	9980/ 9981							
	9995							
8001		1	4	1	128	64 Kbyte	?	no
	8002				255			no
	8003				128			yes
	8004				255			yes

* memory-mapped only

opposing tendencies, the results differ:

- Intel (8086, 8088) have aimed primarily at upgrading the 8080 family; in fact, the 8080 registers are a subset of those in the 8086, so that existing programs can be run with only minor modifications. This has the drawback that the registers are often dedicated to specific instructions; although this often makes for more compact machine-language instructions, it also tends to limit the programming options.
- Motorola (MC 68000) have aimed for the future: 32-bit registers and a powerful instruction set (based, in part, on minicomputer practice). At the same time they have maintained compatibility with the existing 6800-family, so that existing support chips can often be used (usually in pairs).
- National Semiconductor (NS 16032, 16016, 16008) have also aimed for the future, without forgetting the past. This has led to an intriguing combination of old and (very) new ideas: on the one hand '8080-features' like low-high data storage (more on this later); on the other hand 32-bit registers, 16 M byte address range, slave processor concept and provision for virtual memory systems (which will also be discussed further on).
- Texas Instruments (TMS 9900 family) have aimed, quite simply, at putting a minicomputer on a chip. The result is distinctly slower than all other processors; its addressing range is much smaller; it doesn's cater for anything like as many interrupts; its instruction set is much more limited. Why? It's older! At that time, memory and peripherals were much more expensive, so you didn't use so many in one system. This is a great pity: it offers the unique feature of locating a complete general-purpose register set in RAM, which simplifies interrupt processing and subroutine branches enormously, as will be explained.
- Zilog (Z8001, 8002, 8003, 8004) have aimed at providing a powerful,



1a

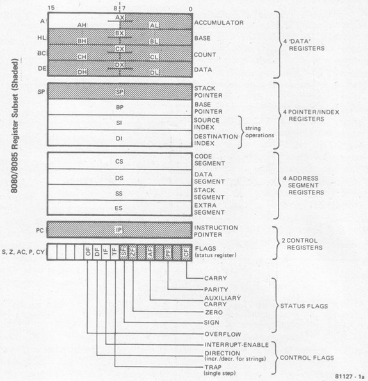


Figure 1a. The 8086 register set. This can be considered an extension of the 8080/8085 registers, as indicated by the shaded areas.

1b

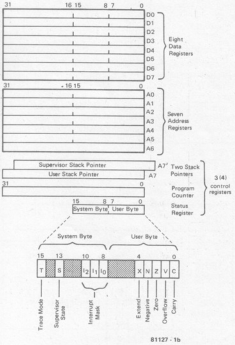


Figure 1b. The registers in the 68000 are 32-bit 'wide'. Is this still a 16-bit processor?

1c

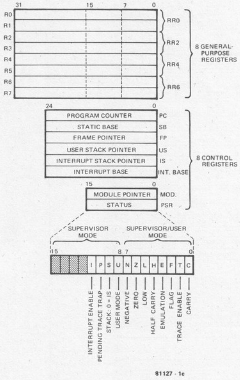


Figure 1c. The 16000 also uses 32-bit general-purpose registers.

1d

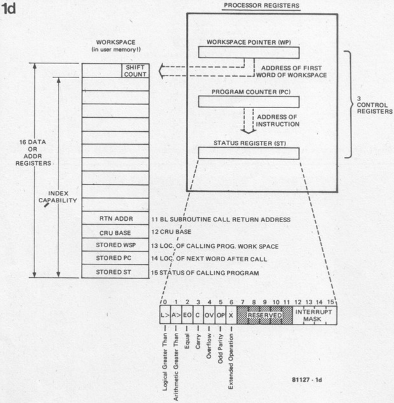


Figure 1d. The 9900 registers are located in RAM. This can be extremely useful in many applications.

general-purpose microprocessor. They seem to have done a good job of combining the best of existing microprocessors with minicomputer practice.

Registers

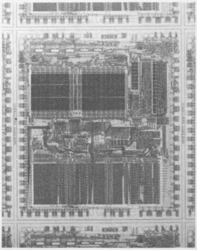
In any microprocessor, registers are used for three distinct applications:

- data is loaded into a register, prior to performing some 'operation' on it (add, subtract, shift or whatever);
- specific memory addresses are contained in registers (the first address of a group of data, stack or program section);
- processor control functions are stored in control registers (the program counter that points to the next instruction to be executed; 'status flags'; etc.).

There are two distinct approaches to the use of registers. In many older 8-bit processors, each register is 'dedicated' to a specific job. There is one 'accumulator' for data operations; a 'stack pointer' for the first address of a stack; and so on. A more flexible system is used in some μPs: 'general-purpose' registers are implemented to perform any data operation or addressing function that the programmer cares to specify. While being more flexible, this approach does have the slight disadvantage that the instructions may be longer. You can't just write 'add 1 to the data'; instead, you must specify 'add 1 to the data *in register 2*', for instance.

In 16-bit processors, there is a clear tendency towards the latter system. Figure 1 shows the register set available to the various processors.

The 8086 (figure 1a) has 14 16-bit registers in all. In principle, these are dedicated as shown. However, Intel goes to great pains to point out that the first eight are 'general' registers: "The data registers can be used without constraint in most arithmetic and logic operations. The pointer and index registers can also participate in most



arithmetic and logic operations. In fact, all eight general registers fit the definition of 'accumulator' as used in first and second generation microprocessors."

Something similar applies to the 68000 (figure 1b). In this case, the first eight 32-bit (!) registers are intended for data manipulation and the second group of eight for 'stack' and 'base' addressing. All sixteen registers can be used for indexing.

The 16000 (figure 1c) has eight (32-bit!) general-purpose registers, as well as an extensive group of control registers.

A rather different approach is used in the 9900 (figure 1d). The processor itself contains the two normal control registers (program counter and status register) plus a 'workspace pointer'. The latter points to the address *in RAM* of the first 'register'; in all, sixteen 'general-purpose' registers are specified in this way. If a new group of 16 registers is required for a subroutine or interrupt, you only have to change the address in the 'workspace pointer'!

Finally, the 8000 family (figure 1e) contains 16 general-purpose registers, one or two of which are actually doubled for 'system' or 'normal' mode.

A general point worth mentioning is the use of registers for other data lengths than 16 bits. This is illustrated by dotted lines in the diagrams:

- 8086: the first four registers can be addressed as two individual 8-bit sections each. Effectively, this means that they can be used as four 16-bit or eight 8-bit registers, or any combination.
- 68000: 8-bit and 16-bit sections of the first eight 32-bit registers can be used individually, as shown; the other registers only offer the 16-bit option.
- 16000: for 8-bit or 16-bit data lengths, the lower part of a register is used. It is also possible to combine two registers and use this pair as a single 64-bit register.
- Z8000: the first eight registers can be split up into 8-bit halves. Furthermore, pairs of 16-bit registers can be used as 32-bit registers; it is even possible to group them as 'quadruples', making 64-bit registers!

Addressing modes

Obviously, when writing a program you must not only tell the processor what to do with data — you must also tell it where to find the data in the first place! In memory and, if so, where? In a register? Or is the data part of the instruction ('add *one* to . . .')?

As any programmer will know, a large number of different ways to indicate where the data is can be a great help. Most processors offer the following options:

- 'register': the register specified in the

1e

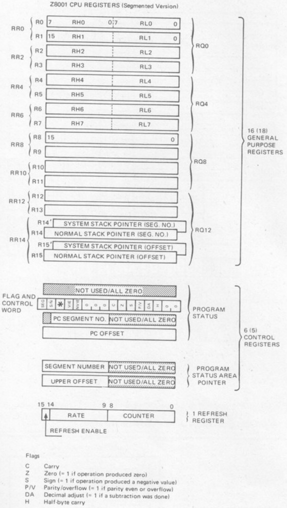
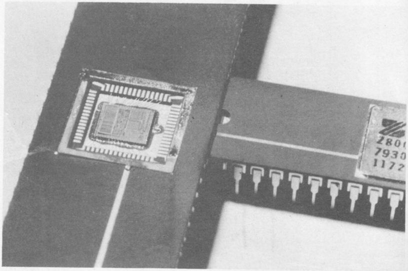
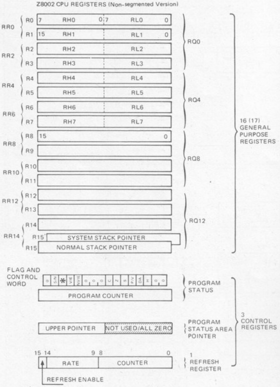


Figure 1e. The registers in the 8001 and 8002 can be combined into 32-bit 'pairs' or even 64-bit





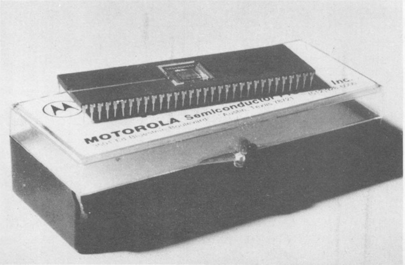
Controls

VIE	Vectored interrupt enable
NVIE	Non-vectored interrupt enable
SEG	Segmentation enable
S/N̄	System/normal mode

• = Future slave processor enable

81127 - 1e

'register quadruples'.



instruction contains the data.

- 'immediate': the data is included in the instruction.
- 'direct': the instruction contains the memory address where the data is to be found.
- 'indirect': the register or memory location specified in the instruction contains the address where the data is to be found.
- 'relative': the data is contained at an address that is a specified number of steps above or below the address pointed to by the program counter.
- 'indexed': the data is at an address that is found by adding a fixed address to a value in an 'index' register. This is useful when retrieving data from a table, for instance. An instruction can read 'retrieve the *fifth* data value (*five* in the index register) from the table starting at address 1000'. Adding one to the value in the index register converts this instruction to 'retrieve the *sixth* data value . . .'

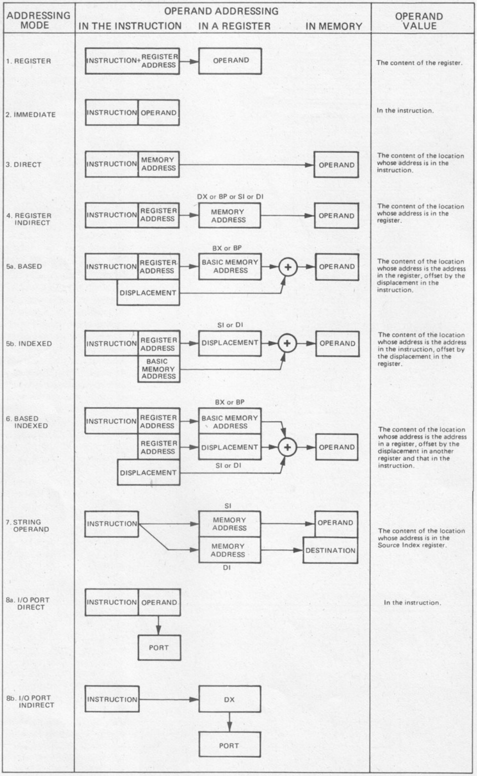
In addition to these basic options, each processor adds its own variations, as illustrated in figure 2. It should be noted that manufacturers don't seem to agree on what name should be given to any particular variation, which can become rather confusing. Most manufacturers, for instance, use the phrase 'direct' addressing when the instruction contains the memory address where the data is located. Motorola, however, call this 'absolute' addressing; '(register) direct' refers to the situation where the data is contained in a specified register.

There are a few other points that are worthy of note. What, for instance, is the difference between 'based' and 'indexed' addressing in the 8086? At first sight, very little. However, there is a difference between them in intent. Assume, for instance, that all kinds of data for all employees in a company are contained in data tables in memory. If you want to print out all data concerning one employee, you can use indexed addressing: specify the first address of the employee's data table and step through it by updating the index register. On the other hand, if you want to total the salary of all the employees, you specify which entry in the tables to look at (the fifth, say) and step through the complete memory by updating the 'base address' register.

Obviously, this sort of thing often involves updating an 'index' register in equal steps at regular intervals. Some processors include this as an extension to indexed addressing instructions ('increment' and/or 'decrement'); others have separate increment/decrement by 1, 2, 4 or even 'n' (Z8000) instructions.

While we're on the subject of 'memory', one further point should be discussed. Existing memory systems are designed for 8-bit processors, so how do you store 16-bit data? In two 8-bit blocks,

2a

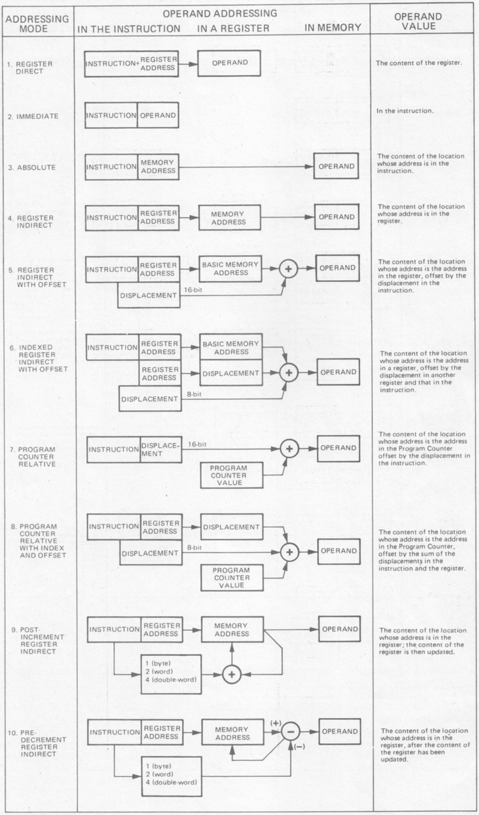


ADDRESSING MODE	OPERAND VALUE
1. REGISTER	The content of the register.
2. IMMEDIATE	In the instruction.
3. DIRECT	The content of the location whose address is in the instruction.
4. REGISTER INDIRECT	The content of the location whose address is in the register.
5a. BASED	The content of the location whose address is the address in the register, offset by the displacement in the instruction.
5b. INDEXED	The content of the location whose address is the address in the instruction, offset by the displacement in the register.
6. BASED INDEXED	The content of the location whose address is the address in a register, offset by the displacement in another register and that in the instruction.
7. STRING OPERAND	The content of the location whose address is in the Source Index register.
8a. I/O PORT DIRECT	In the instruction.
8b. I/O PORT INDIRECT	

81127 - 2a

Figure 2a. Addressing modes in the 8086 (and 8088). Note that it is often necessary to use a particular register for the job: for indexed addressing, say, either the SI or DI register *must* be used.

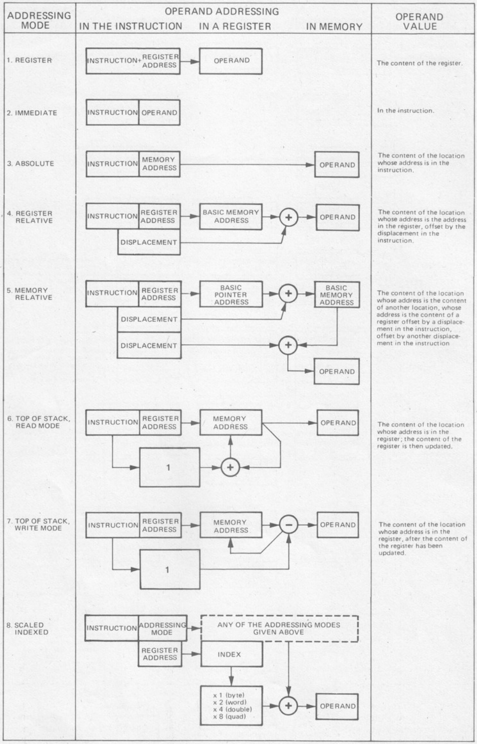
2b



ADDRESSING MODE	OPERAND VALUE
1. REGISTER DIRECT	The content of the register.
2. IMMEDIATE	In the instruction.
3. ABSOLUTE	The content of the location whose address is in the instruction.
4. REGISTER INDIRECT	The content of the location whose address is in the register.
5. REGISTER INDIRECT WITH OFFSET	The content of the location whose address is the address in the register, offset by the displacement in the instruction.
6. INDEXED REGISTER INDIRECT WITH OFFSET	The content of the location whose address is the address in a register, offset by the displacement in another register and that in the instruction.
7. PROGRAM COUNTER RELATIVE	The content of the location whose address is the address in the Program Counter offset by the displacement in the instruction.
8. PROGRAM COUNTER RELATIVE WITH INDEX AND OFFSET	The content of the location whose address is the address in the Program Counter, offset by the sum of the displacements in the instruction and the register.
9. POST-INCREMENT REGISTER INDIRECT	The content of the location whose address is in the register; the content of the register is then updated.
10. PRE-DECREMENT REGISTER INDIRECT	The content of the location whose address is in the register, after the content of the register has been updated.

Figure 2b. The 68000 also offers 'post-increment' and 'pre-decrement' addressing modes. These are extremely useful when manipulating large blocks of data.

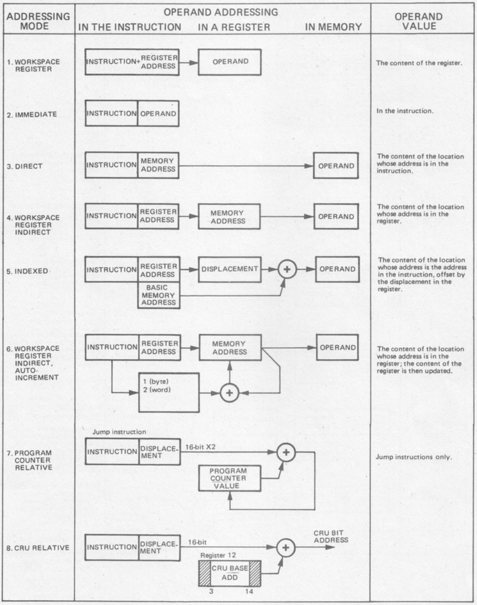
2c



ADDRESSING MODE	OPERAND VALUE
1. REGISTER	The content of the register.
2. IMMEDIATE	In the instruction.
3. ABSOLUTE	The content of the location whose address is in the instruction.
4. REGISTER RELATIVE	The content of the location whose address is the address in the register, offset by the displacement in the instruction.
5. MEMORY RELATIVE	The content of the location whose address is the content of another location, whose address is the content of a register offset by a displacement in the instruction, offset by another displacement in the instruction
6. TOP OF STACK, READ MODE	The content of the location whose address is in the register; the content of the register is then updated.
7. TOP OF STACK, WRITE MODE	The content of the location whose address is in the register, after the content of the register has been updated.
8. SCALED INDEXED	

Figure 2c. The 16000 has some unique addressing modes: 'memory relative' and 'scaled indexed', in particular. Since 'scaled indexed' can be used with any of the other modes that address an operand in memory, it becomes an extremely powerful tool.

2d

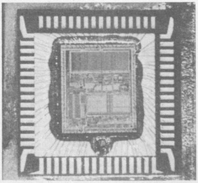


ADDRESSING MODE	OPERAND VALUE
1. WORKSPACE REGISTER	The content of the register.
2. IMMEDIATE	In the instruction.
3. DIRECT	The content of the location whose address is in the instruction.
4. WORKSPACE REGISTER INDIRECT	The content of the location whose address is in the register.
5. INDEXED	The content of the location whose address is the address in the instruction, offset by the displacement in the register.
6. WORKSPACE REGISTER INDIRECT, AUTO-INCREMENT	The content of the location whose address is in the register; the content of the register is then updated.
7. PROGRAM COUNTER RELATIVE	Jump instructions only.
8. CRU RELATIVE	

81127 - 2d

Figure 2d. The 9900 has all the usual basic addressing modes, plus a few that relate specifically to this particular processor's 'architecture'.

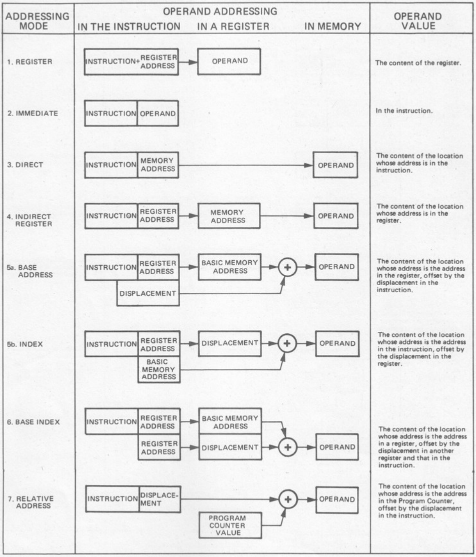
obviously! However, this means that each 16-bit word takes up two memory addresses, and each manufacturer has drawn different conclusions from this. In the first place, Intel and National have decided to store the least significant byte at the lower memory address — like writing '8119' when you mean '1981'. The other manufacturers do it the other way around. Furthermore, in most cases the data must always be 'aligned': the first address for each 16-bit word must be an even-numbered address. This saves an address line and gives a greater range for 'relative' addressing. However, it also means that data and instructions cannot always be tightly 'packed' in memory, and for this reason Intel (8086/8088) cater for both aligned and non-aligned data — the former being faster.



Instruction sets

The more instructions the merrier, you would think. However, this is not strictly true: it all depends on how powerful your instructions are in the first place. To give an example: for

2e



ADDRESSING MODE	OPERAND VALUE
1. REGISTER	The content of the register.
2. IMMEDIATE	In the instruction.
3. DIRECT	The content of the location whose address is in the instruction.
4. INDIRECT REGISTER	The content of the location whose address is in the register.
5a. BASE ADDRESS	The content of the location whose address is the address in the register, offset by the displacement in the instruction.
5b. INDEX	The content of the location whose address is the address in the instruction, offset by the displacement in the register.
6. BASE INDEX	The content of the location whose address is the address in a register, offset by the displacement in another register and that in the instruction.
7. RELATIVE ADDRESS	The content of the location whose address is the address in the Program Counter, offset by the displacement in the instruction.

81127 - 2e

Figure 2e. At first sight, the 8001 and 8002 appear to offer less addressing modes than the other processors. However, some 'addressing possibilities' (increment and decrement, for instance) are included in the instruction set.

block transfer, the 8086 offers the instructions 'repeat', 'compare' and 'decrement'. The Z8000 set includes a single instruction 'compare, decrement and repeat'. Each processor has its own strong and weak points — the 8086, for instance, is the only one to include 'ASCII adjust for add and subtract'.

Table 4 gives an approximate comparison of the various instruction sets, but to get the full picture the manufacturer's literature will have to be studied. Some processors have 'machine language' instructions that are relatively easy to memorise — a boon to amateur programmers! Some assemblers are more powerful than others, which should appeal to professional users. Some instruction sets are more suited

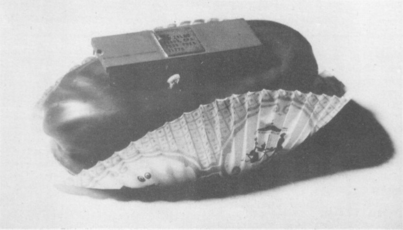


Table 4

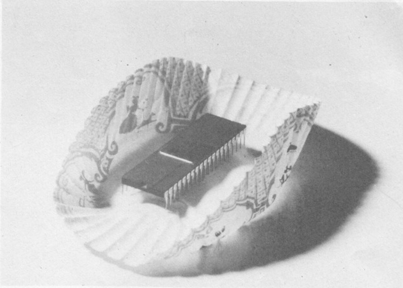
	INTEL 8086	MOTOROLA 68000	NATIONAL 16032	TEXAS 9900	ZILOG 8001
DATA TRANSFER					
• **move:** general purpose	X	X	X	XX	}13 (B/W/DW)
immediate to register	X }X	}X	}X	X	
immediate to memory					XX
to/from dedicated registers (accu, addr. reg, program status etc.)	XX (accu) XX (segm. reg.) XX (EA/pointers) XX (flags)	X (CCR) XX (SR) X (USP) XX (An) XX (SP/An)		(XX CRU = I/O) XXX (workspace pointer) X (SR) X (int. mask)	
move multiple registers		X			
• **stack:** push	XXXX				XX
pop	XXXX				XX
save registers			X		XX
restore registers			X		
• **other:** exchange data	XX	X			
clear		X			XX
swap bytes		X		X	XX
load address	X		X	X	
translate byte	X				XX
BLOCK TRANSFER AND STRING MANIPULATION					
• repeat	X				
• move	X		XX		(n.a.)
load	X				XXXX
store	X				
move and repeat			(X)		
• compare	X		XX		XXXX
compare and repeat			(X)		8
scan	X				8
translate	X				
translate and repeat					XX
translate, test					XX
translate, test and repeat					XX
• skip string			X		XX
INPUT/OUTPUT					
• **input**	XX	m	m	X (CRU)	XX
input and incr./decr.	see 8089	e	e		XXXX
input, incr./decr. and repeat		m	m		XXXX
special input		o	o		XX
special input and incr./decr.		r	r		XXXX
special input/incr./decr. and repeat		y	y		XXXX
• **output**	XX	m	m	X (CRU)	XX
output and incr./decr.	see 8089	a	a		XXXX
output, incr./decr. and repeat		p	p		XXXX
special output		p	p		XX
special output and incr./decr.		e	e		XXXX
special out, incr./decr. and repeat		d	d		XXXX
• **more** peripheral data (8-bit)		X			
• **communication register:**					
test CRU bit				X	
set CRU bit				X	
clear CRU bit				X	
ARITHMETIC					
• **add**	XXX	XXXX	XX	XXX	XXX
add with carry	XXX		X	(X: 9940)	XX
add decimal		X	X		
decimal adjust for add	X	(n.a.)			X
ASCII adjust for add	X				
increment by one	XX			X	
increment by two				X	
increment by 'n'					XX
add address		X	X		
• **subtract**	XXX	XXXX	X	XX	XXX
subtract with borrow	XXX		X		XX
subtract decimal		X	X	(X:9940)	
decimal adjust for sub.	X	(n.a.)			(X)
ASCII adjust for sub.	X				
decrement by one	XX			X	
decrement by two				X	
decrement by 'n'					XX
change sign	X	X	X	X	XX
change sign, decimal		X			
subtract address		X			
• **multiply**, unsigned	X	X	X	X	
multiply, signed	X	X	X		XX
ASCII adjust for mult.	X				
• **divide**, unsigned	X	X	X	X	
divide, signed	X	X	X		XX
ASCII adjust for divide	X				
extend sign	XX	X	XX		XXX
evaluate periodic function			X		
modulus of periodic function			X		
remainder			X		
• **compare**	XXX	XXX	XX	XXX	XXXXX
check R against bounds		X	X		
compare address		X	X		
• **absolute value**			X	X	

Table 4, continued

	INTEL 8086	MOTOROLA 68000	NATIONAL 16032	TEXAS 9900	ZILOG 8001
LOGIC					
• **AND**	XXX	XX	X	X	XX
• **OR**	XXX	XX	X	X	XX
• **EXOR**	XXX	XX	X	X	XX
• **NOT**	X	X	X	X	XX
• **test** flag(s)/CC	XXX	(n.a.)		(X CRU = I/O)	XX
test operand		X			XXX
test and set		X			
ROTATE AND SHIFT					
• **shift** logical left	}X	X	X	}X	XXX
shift arithmetic left		X	X		XXX
shift logical right	X	(X) ≈ SLL	(X) ≈ SLL	X	XXX
shift arithmetic right	X	(X) ≈ SAL	(X) ≈ SAL	X	XXX
shift dynamic logical			(X) ≈ SLL		XXX
shift dynamic arithmetic			(X) ≈ SAL		XXX
• **rotate** right	X	X	X	X	XX
rotate right through carry/extend	X	X			XX
rotate left	X	X	(X) ≈ RR		XX
rotate left through carry/extend	X	X			XX
rotate digit left					X
rotate digit right					X
BIT MANIPULATION					
• **bit test**		X	X		XXXX
bit test and change		X			
bit test and clear		X			
bit test and set		X			XX
compare ones corresponding				X	
compare zeroes corresponding				X	
find first set bit			X		
• **set** ones				X	
set bits corresponding			X	XX	XXXX
set CRU bit				(X = I/O)	
set bit			XX		
• **reset** bits corresponding			X	XX	XXXX
reset CRU bit				(X = I/O)	
reset bit			XX		
• **invert** bit			X		
• extract **bit field**			XX		
insert bit field			XX		
convert bit field pointer			X		
PROGRAM CONTROL					
• **call** subroutine	XXXX	XX	XX	XX	XX
return from call	XXXX	X	X		X
extended operation (user-def.)			XX	X	
execute (variable instruction)				X	
system call			X		X
• **interrupt** call	XXX				
return from interrupt	X	XX	XX		X
• **jump/branch**, unconditional	XXXXX	XX	XX	XX	
jump/branch, conditional	16	X	14	12	4
multiway branch			X		
• **loop**, conditional	XXX	X	X		
jump from loop	X				
PROCESSOR CONTROL					
• **control bits**, clear	XXX		X		XX
control bits, set	XXX	X	XX		XX
control bits, invert	X				X
control bits, move			XX		XXXXX
multi-micro request					X
multi-micro set					X
multi-micro reset					X
multi-micro test					X
• halt, wait	XX	X	X	X	X
NOP		X	X		X
reset (external devices)		X		X	
escape (to external device)	X				
restart				X	
clock bus	X				
segment override	X				
trap		X	X		
trap on overflow		X			
clock off				X	
clock on				X	
breakpoint			X		

Table 4. A comparison of the five instruction sets. The number of crosses indicates how many variations of a particular instruction type exist; where this would tend to get out of hand, a number is used instead. This survey is intended only to give a general impression; for the exact details, the manufacturer's instruction set must be studied. It should also be noted that many instructions go under several different names. For instance, what we have listed as 'extend sign' may actually be shown in the official instruction set as 'extend MSB', 'convert byte to word', etc.

to higher programming languages (Pascal, for instance) than others. To go into all this in sufficient detail is beyond the scope of this article.



Interrupts

The basic idea behind 'interrupts' in computer systems is that when a program is running it may have to be 'interrupted' at any time, so that the computer can do some other (more urgent) job first. When that job is finished, the computer can return to the original program and carry on where it left off. For instance, some chess computers 'think' in the opponent's time. When he makes his move, the computer's calculation must be interrupted and the new position of the piece is entered; only then can it continue its calculations to work out its own move.

Obviously, different interrupt sources will require different interrupt routines; and the sooner the computer knows which routine to run, the better. For this reason, all the 16-bit processors offer a 'vectored interrupt' facility: the interrupt source points to a position in an address table, that contains the initial address of the required interrupt routine.

This address table must be located somewhere in memory. As can be seen in figure 3, most processors reserve a (large) section from address 00000 on, and some also require a (small) section at the last memory addresses. The Z8000 is an exception: its pointer table ('program status area') can be located anywhere in memory; the NS 16000 also provides for a freely locatable 'interrupt and trap vector table'.

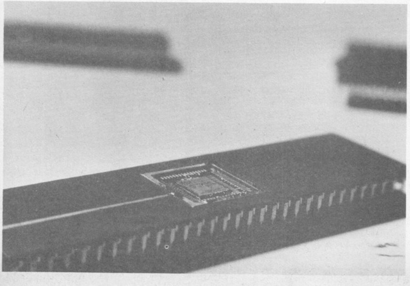
It is also extremely useful to know how 'urgent' a particular interrupt request is, in relationship to the program that is actually running at the time. This leads to the distinction between:

- *non-maskable interrupts:* when this occurs, the corresponding routine must be executed without delay. A prime example would be a 'power-failure interrupt': emergency procedures must be carried out without delay!
- *priority-coded interrupts:* an interrupt request includes a code that indicates its urgency. If this proves to be more important than what the computer is actually doing, the request is acknowledged; otherwise it is ignored.

Generally speaking, all interrupts except the non-maskable one are priority-coded.

A further distinction is made between:

- *(normal) interrupts*: these are generated by some external device, as explained above.
- *(software) traps*: these occur when 'something funny' happens during normal execution of a program — overflow, for instance. In some processors, they can also be deliberately 'called' by a normal program instruction; in fact, the 8086 offers the possibility of initiating *all* interrupt routines (even the hardware type) by giving a suitable instruction.



System extensions

As we stated at the outset: computer systems tend to grow. Figure 4 gives some idea of what this means in pracpractice... Long as it is, the list is by no means complete: new support chips are being announced daily, and even in existing literature some manufacturers tend to list more than others. Furthermore, some of the 'extensions' shown will not be necessary (or even desirable) in many applications. For instance, we have consistently 'demultiplexed' the data- and address buses, even though this will often be unnecessary.

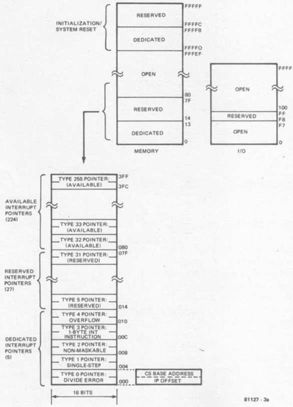
It should be noted that some of the support chips are (slave) microprocessors in their own right. In figure 4a, for instance, the 8089 input/output processor is derived from the 8080 family.

The same applies to the 'memory management unit' shown in figures 4b, 4c and 4e. Here, a new concept is introduced at the same time: 'virtual memory' as opposed to real memory. Obviously, when processors can operate with 48 or even 64 *mega*-bytes of memory, it is hardly feasible to have all this as RAM. For this reason, it is common practice to have a much smaller RAM area and store data that is not in use at that time on a floppy disk or some similar 'low-cost' memory. As required, sections of program or data are retrieved from the disk and located in RAM, where the processor can reach them.

To avoid having to burden the processor (and the programmer!) with this job, a separate 'memory management unit' is used. This looks at the 'logical address' that the processor is putting out and

3a

8086



81127 - 3a

3b

68000

ADDRESS HEX	VECTOR NUMBER(S)	ASSIGNMENT
000	0	RESET: INITIAL SSP
004	–	RESET: INITIAL PC
008	2	BUS ERROR
00C	3	ADDRESS ERROR
010	4	ILLEGAL INSTRUCTION
014	5	ZERO DIVIDE
018	6	CHK INSTRUCTION
01C	7	TRAPV INSTRUCTION
020	8	PRIVILEGE VIOLATION
024	9	TRACE
028	10	LINE 1010 EMULATOR
02C	11	LINE 1111 EMULATOR
030	12*	(UNASSIGNED, RESERVED)
034	13*	(UNASSIGNED, RESERVED)
038	14*	(UNASSIGNED, RESERVED)
03C	15	UNINITIALIZED INTERRUPT VECTOR
040	16-23*	(UNASSIGNED, RESERVED)
≈	≈	⋮
05F	23	•
060	24	SPURIOUS INTERRUPT
064	25	LEVEL 1 INTERRUPT AUTOVECTOR
068	26	LEVEL 2 INTERRUPT AUTOVECTOR
06C	27	LEVEL 3 INTERRUPT AUTOVECTOR
070	28	LEVEL 4 INTERRUPT AUTOVECTOR
074	29	LEVEL 5 INTERRUPT AUTOVECTOR
078	30	LEVEL 6 INTERRUPT AUTOVECTOR
07C	31	LEVEL 7 INTERRUPT AUTOVECTOR
080	32-47	TRAP INSTRUCTION VECTORS
≈	≈	⋮
0BF	47	•
0C0	48-63*	(UNASSIGNED, RESERVED)
≈	≈	⋮
0FF	63	•
100	64-255	USER INTERRUPT VECTORS
≈	≈	⋮

81127 - 3b

Figure 3. All processors require some dedicated RAM area. Among other things, they expect to find the start address of interrupt routines there (the 'interrupt vectors'). When using the 16000 (figure 3c) and the 8001 (figure 3e), this RAM area can be freely located anywhere in memory; all other processors require some dedicated RAM in extreme low and/or extreme high memory.

checks whether that particular section is actually in RAM. If so, well and good: the MMU simply puts the correct RAM address (the 'physical address') on the bus. Otherwise, it gives a warning to the processor ('hold everything!'); makes space in RAM by storing some or all of its contents on the disk; loads the necessary section from disk into RAM; and finally tells the main processor that it can continue the program. For this sort of thing to work properly, the main processor must be stopped in time without losing or modifying the data that it is currently working on. This is where the 'Abort' facility comes in. As Zilog put it when they introduced the Z8003 and Z8004: 'The Abort capability allows for the interruption of instructions or access to data that do not reside in main memory. More generally, when the Z8003/4 tries to access non-existent memory the attempted access is aborted gracefully."

Other features can be important when extending a system: Direct Memory Access (DMA), multi-processor operation and so on. However, since all the processors discussed here provide all these options in one way or another, there is little point in going into greater detail. The same applies to 'software support': for all these processors there is 'an adequate abundancy' of literature, assembler routines, software and so on.

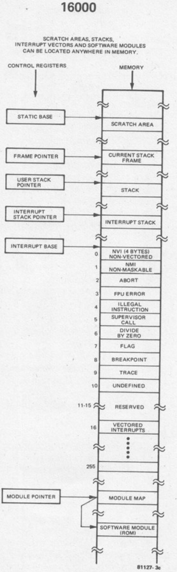
In conclusion

Each of the five processors has its own strong and weak points, but each of them will do almost any job. As someone put it: 'Even if there was a "best" micro, other factors (such as your own skills and attitude, the available software and so on) will soon reduce any advantage of this "best" micro to zilch. If you don't happen to like the "best" micro, just wait a month or two and it will get shot out of the saddle by something much more promising.' There's a lot of truth in this!

However, if you want to pick a 16-bit processor and get started now, the choice may well depend on factors that have not been discussed above:

- Price and availability? These can change within weeks, so it is advisable to contact the various manufacturers (a list of addresses is included at the end of this article). The National Semiconductor NS 16000, for instance, is so new that the data we needed had to be flown in from America! The first samples should be available later this year.

3c



3d

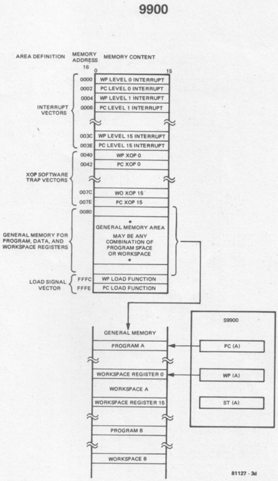


Figure 3e on page 18 →

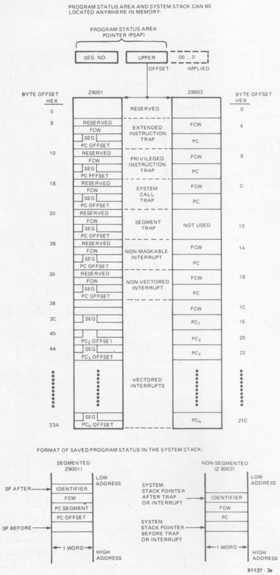
- A clear and straightforward 'machine language' instruction set? This is more important to many amateurs than a 'powerful assembler'! But what do you mean by 'clear and straightforward'? To some extent, that depends on what you're used to. Also, a point to watch is the difference between what manufacturers claim and what they do. Motorola, for instance, stress the fact that they have a set of 'powerful, general-purpose instructions' so that the programmer 'has less to remember when writing software'. This is true, up to a point. Zilog, on the other hand, claim one of the most comprehensive instruction sets. Also true, up to a point. However, if you cut through the 'mnemonics' and look at the actual bits in the instructions, the results can be surprising. To give an example: Shift instructions. Motorola list four for the 68000 (Arithmetic Shift Left or Right, and Logical Shift Left or Right), whereas Zilog give six for the Z8000 (Shift Dynamic Arithmetic or Logical, Shift Left Arithmetic or Logical and Shift Right Arithmetic or Logical). Motorola point out that they use the same instruction for either 'dynamic' or 'static' shifts – "fewer being better"! ('Dynamic' means that the number of positions that the data is to be shifted is contained in a register; 'static' means that the number is part of the instruction). What is the truth of the matter? Both processors use a single basic instruction for all shift operations! Two bits make the distinction between Byte, Word or Double-word data; one bit determines whether an arithmetic or logic shift is required. The 68000 uses one bit to distinguish between shifting left or right; the Z8000 makes this distinction by using a positive or negative number to specify the shift (for left or right, respectively) – limiting its dynamic shift range to 32 positions, as opposed to Motorola's 64 positions. On the other hand, the Z8000 uses one bit to distinguish between Static and Dynamic shift, and this leads to a greater range for the static shift (up to 32 positions, as opposed to Motorola's 8). So which processor is 'better'?

Furthermore, some manufacturers distinguish instructions that others consider to be the same instruction with different addressing modes. To give one example: in the Z8000 addressing modes summary (figure 2e), 'Indirect register with increment or decrement' is conspicuously absent. On the other hand, the instruction set includes: 'Load', 'Load and Decrement', 'Load, Decrement and Repeat', and so on.

To sum it all up: when you cut down to the bare bones, you find that all

3e

Z8000



8086/8088:
- *The 8086 Family User's Manual (Intel 1979)*
- *16-bit microprocessor benchmark report (Intel 1980)*

MC 68000:
- *Microcomputer forum (Motorola 1980)*
- *MC 68000 reference summary (Motorola)*
- *User information/preliminary descriptions available from Motorola (MC 68000), Rockwell (R68000), Thomson (EF 68000), Hitachi (HD 68000).*
- *MC 68000 article reprints (Motorola 1980)*

NS 16000:
- *NS 16000 family overview (National Semiconductor 1980)*
- *NS 16000 technical marketing brief (Nat. Sem. 1980)*

TMS 9900:
- *9900 family systems design (Texas Instruments 1978)*
- *data sheets/product information available from Texas Instruments (TMS 9900), AMI (S9900), ITT (ITT 9900)*
- *16-bit μP Technical articles (AMI 1979)*

Z 8000:
- *Z 8001 and Z 8002 programming manual (SGS/Ates 1980)*
- *AM Z 8000 family data book (AMD 1980)*
- *Z 8000 Technical Manual (Zilog 1980)*
- *Z 8000 Assembly language programming manual (Zilog 1980)*
- *data sheets/Application notes available from Zilog (Z 8000), AMD (AM Z 8000)*
- *Programming the Z 8000 (Sybex 1980)*

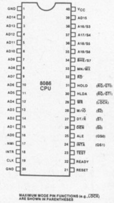
these processors are very similar in most respects. they are all much more powerful than 8-bit processors – it is not just a question of going from narrow gauge to standard gauge. Any choice between them must be based to a large extent on personal taste, and to a lesser extent on the intended application. The 9900, for instance, has its own particular charm – but it could do with some general-purpose registers on the CPU chip as well (to speed things up) and a more extensive instruction set. The 68000, 16000 and Z8000 are all very close in capabilities and general structure; it is very difficult to name a 'winner', even on (decimal) points. The 8086, on the other hand, is closer in some ways to 8-bit practice. This can be an advantage or a disadvantage, depending on how you look at it.

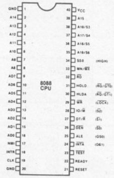
Future Developments?

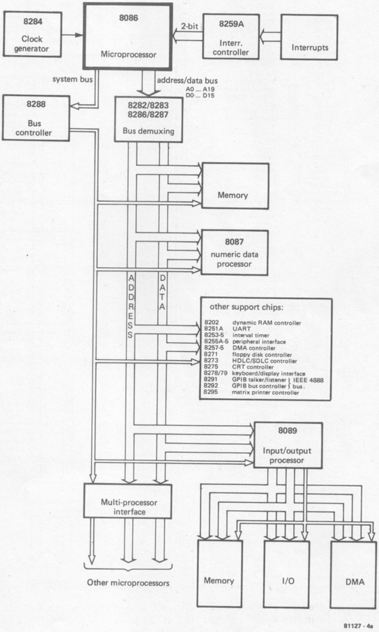
Each processor is likely to be improved in the future. Motorola, for instance, states specifically: 'The present version of the 68000 does not offer string operations, but they will be available on the next version, along with floating point operations'. Texas Instruments are working hard 'behind the scenes' – on what?

It will be interesting to see how things develop. In future issues, we will be including articles on each processor family – with as much detail and 'expectations for the future' as we can obtain! Meanwhile, we will continue our policy of using the processor that comes to hand for any given job.

4a


GND 1
AD14 2
AD13 3
AD12 4
AD11 5
AD10 6
AD9 7
AD8 8
AD7 9
AD6 10
AD5 11
AD4 12
AD3 13
AD2 14
AD1 15
AD0 16
NMI 17
INTR 18
CLK 19
GND 20
8086 CPU
40 Vcc
39 AD15
38 A16/S3
37 A17/S4
36 A18/S5
35 A19/S6
34 $\overline{BHE}$/S7
33 MN/$\overline{MX}$
32 $\overline{RD}$
31 HOLD ($\overline{RQ}$/$\overline{GT0}$)
30 HLDA ($\overline{RQ}$/$\overline{GT1}$)
29 $\overline{WR}$ ($\overline{LOCK}$)
28 M/$\overline{IO}$ ($\overline{S2}$)
27 DT/$\overline{R}$ ($\overline{S1}$)
26 $\overline{DEN}$ ($\overline{S0}$)
25 ALE (QS0)
24 $\overline{INTA}$ (QS1)
23 $\overline{TEST}$
22 READY
21 RESET
MAXIMUM MODE PIN FUNCTIONS (e.g., $\overline{LOCK}$) ARE SHOWN IN PARENTHESES





Common Signals

8086	Function	8088
AD15-AD0	Address/Data Bus	AD7-AD0
—	Address Bus	A15-A8
A19/S6- A16/S3	Address/Status	A19/S6- A16/S3
$\overline{BHE}$/S7	Bus High Enable/ Status	—
MN/$\overline{MX}$	Minimum/Maximum Mode Control	MN/$\overline{MX}$
$\overline{RD}$	Read Control	$\overline{RD}$
$\overline{TEST}$	Wait On Test Control	$\overline{TEST}$
READY	Wait State Control	READY
RESET	System Reset	RESET
NMI	Non-Maskable Interrupt Request	NMI
INTR	Interrupt Request	INTR
CLK	System Clock	CLK
V_{CC}	+5 V	V_{CC}
GND	Ground	GND

Minimum Mode Signals (MN/MX = V_{CC})

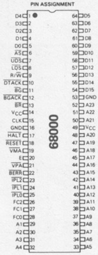
8086	Function	8088
HOLD	Hold Request	HOLD
HLDA	Hold Acknowledge	HLDA
$\overline{WR}$	Write Control	$\overline{WR}$
M/$\overline{IO}$	Memory I/O Control	IO/$\overline{M}$
DT/$\overline{R}$	Data Transmit/ Receive	DT/$\overline{R}$
$\overline{DEN}$	Data Enable	$\overline{DEN}$
ALE	Address Latch Enable	ALE
$\overline{INTA}$	Interrupt Acknowledge	$\overline{INTA}$
—	S0 Status	SS0

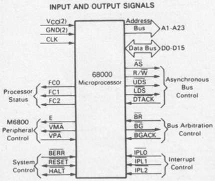
Maximum Mode Signals (MN/MX = GND)

8086	Function	8088
$\overline{RQ}$/$\overline{GT1}$, 0	Request/Grant Bus Access Control	$\overline{RQ}$/$\overline{GT1}$, 0
$\overline{LOCK}$	Bus Priority Lock Control	$\overline{LOCK}$
$\overline{S2}$-$\overline{S0}$	Bus Cycle Status	$\overline{S2}$-$\overline{S0}$
QS1, QS0	Instruction Queue Status	QS1, QS0

Figure 4a. The 8086 and 8088 microprocessors are members of what Intel calls the iAPX-86 family. This family includes several 'slave' processors: microprocessor-based support chips, that perform jobs that the processor itself cannot — or certainly not as easily. Clear examples are the 'Numeric Data Processor' and 'Input/Output processor', with further 'slaves' to be announced. This processor can be operated in either 'minimum' or 'maximum' mode. In minimum mode, the processor controls the bus itself; in maximum mode, a 'bus controller' is added as shown above. The maximum mode is intended for large systems, where a more extensive control bus is required.

4b





Pin	Function
A1-A23	Address Bus
D0-D15	Data Bus
$\overline{AS}$	Address Strobe
R/$\overline{W}$	Read/Write
$\overline{UDS}$, $\overline{LDS}$	Upper and Lower Data Strobes
$\overline{DTACK}$	Data Transfer Acknowledge
$\overline{BR}$	Bus Request
$\overline{BG}$	Bus Grant
$\overline{BGACK}$	Bus Grant Acknowledge
$\overline{IPL0 \ldots 2}$	Interrupt Priority Level
$\overline{BERR}$	Bus Error
$\overline{RESET}$	Reset
$\overline{HALT}$	Halt
E	Enable
$\overline{VMA}$	Valid Memory Address
$\overline{VPA}$	Valid Peripheral Address
FC0,FC1,FC2	Function Code Output
CLK	Clock
VCC	Power Input
GND	Ground

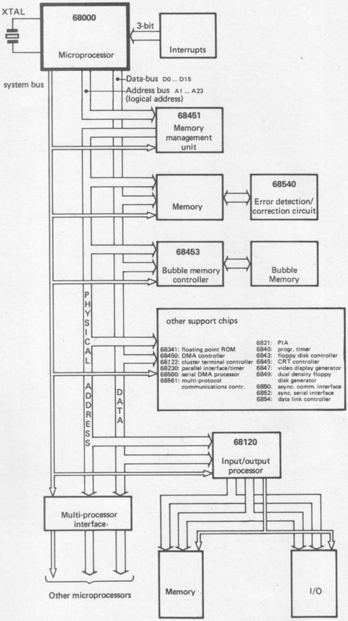
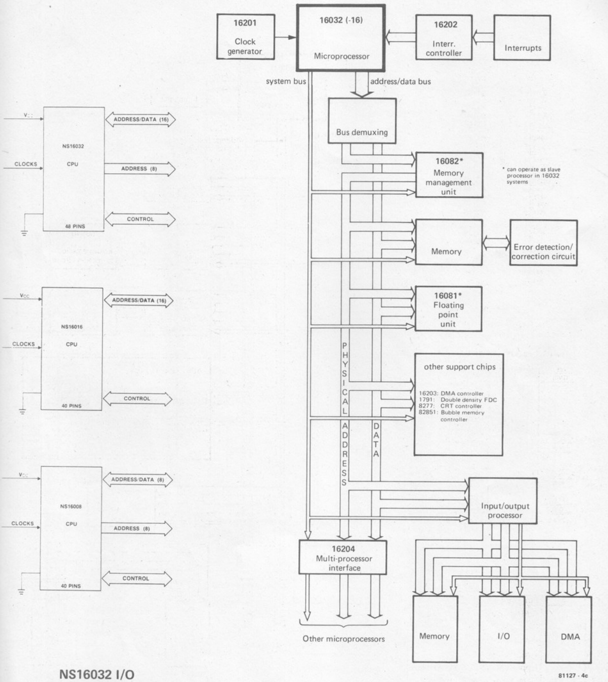

XTAL
68000
Microprocessor
3-bit
Interrupts
system bus
Data-bus D0 ... D15
Address bus A1 ... A23
(logical address)
68451
Memory management unit
Memory
68540
Error detection/ correction circuit
68453
Bubble memory controller
Bubble Memory
PHYSICAL ADDRESS
DATA
other support chips
68341: floating point ROM
68450: DMA controller
68122: cluster terminal controller
68230: parallel interface/timer
68560: serial DMA processor
68561: multi-protocol communications contr.
6821: PIA
6840: progr. timer
6843: floppy disk controller
6845: CRT controller
6847: video display generator
6849: dual density floppy disk generator
6850: async. comm. interface
6852: sync. serial interface
6854: data link controller
68120
Input/output processor
Multi-processor interface
Other microprocessors
Memory
I/O
81127 - 4b

Figure 4b. The 68000 also belongs to a large family. Some of the other members of the family are also 'intelligent': microprocessor-based support chips like the 'Memory Management Unit' and 'Input/Output controller'. There are also a large number of 'normal' support chips. Furthermore, as an added bonus to many potential users, Motorola have ensured that existing 6800 peripherals and support chips can be used in 68000 systems — usually in pairs.
It is worth particular note that the 68000 is one of the few 16-bit processors with separate data and address buses.

4c



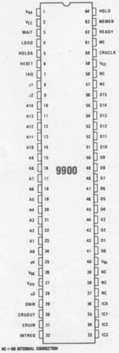
NS16032 I/O

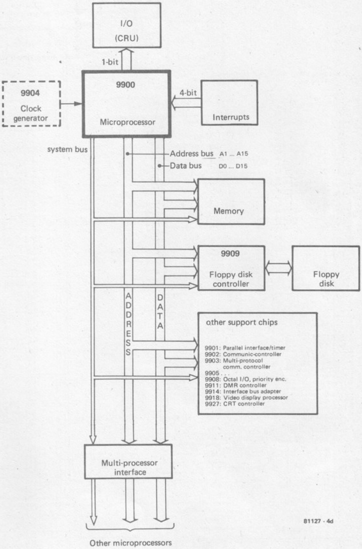
- 16-Bit Address/Data (MUX)
- 8-Bit Address
- 4 Bits Status
- $\overline{\text{ADS}}$
- $\overline{\text{DDIN}}$
- $\overline{\text{HBE}}$
- RDY
- $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, $\overline{\text{ILO}}$
- $\overline{\text{NMI}}$, $\overline{\text{INT}}$
- $\overline{\text{ABT}}$
- $\overline{\text{FLT}}$, U/$\overline{\text{S}}$, $\overline{\text{PFS}}$
- PH1, PH2
- $\overline{\text{RST}}$
- $\overline{\text{SPC}}$
- 2 GNDS And V_{CC}

Figure 4c. The 16000 is so new that the 'pinning' is not yet known! However, we have received sufficient 'preliminary' information to give a fairly complete picture of its possibilities. As with many of the other 16-bit processors, the 16000 'family' includes several intelligent support chips. In fact, National Semiconductor even take this principle one step further in their literature: when discussing the register set, they also count the registers in the 'Floating point unit' and the 'Memory management unit'! Although there is something to be said for this, it didn't seem quite fair to the other manufacturers — and so figure 1c only lists the registers in the CPU itself.

A strong point of the 16000 family is not readily apparent from the block diagram given above: the ease with which 'software modules' (program sections and subroutines in ROM) can be located anywhere in memory. The instruction set and addressing modes were designed with this possibility in mind, and National state that they intend to provide an extensive 'software library'. That would certainly simplify things: 'Don't re-invent the wheel, use the existing plans'!

4d

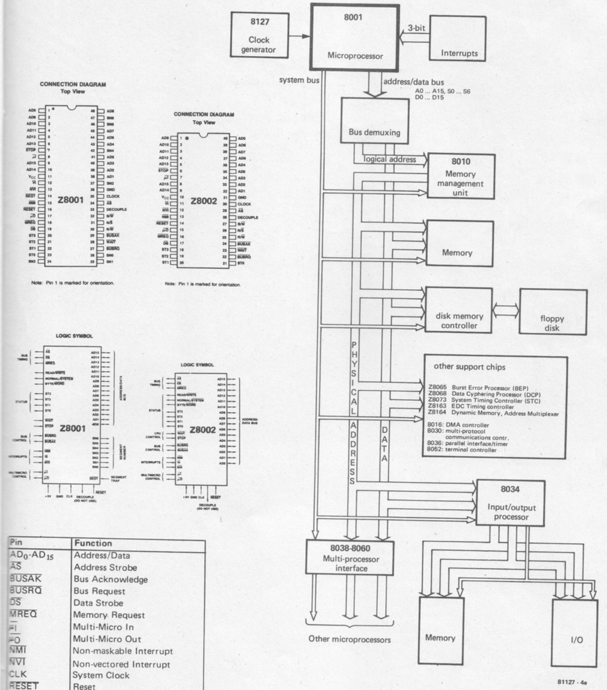




Pin	Function
A0-A14	Address bus
D0-D15	Data bus
Ø1-Ø4	clock
V_{BB}	–5 V
V_{CC}	+ 5 V
V_{DD}	+12 V
V_{SS}	GND
$\overline{\text{INTREQ}}$	Interrupt request
IC0-IC3	Interrupt codes
CRUIN	CRU data in
CRUOUT	CRU data out
CRUCLK	CRU clock
DBIN	Data bus in
$\overline{\text{MEMEN}}$	Memory enable
$\overline{\text{WE}}$	Write enable
READY	Memory ready
$\overline{\text{HOLD}}$	Hold request
HOLDA	Hold acknowledge
WAIT	Wait indication
$\overline{\text{RESET}}$	Reset
IAQ	Instruction acquisition
$\overline{\text{LOAD}}$	Load WP and PC

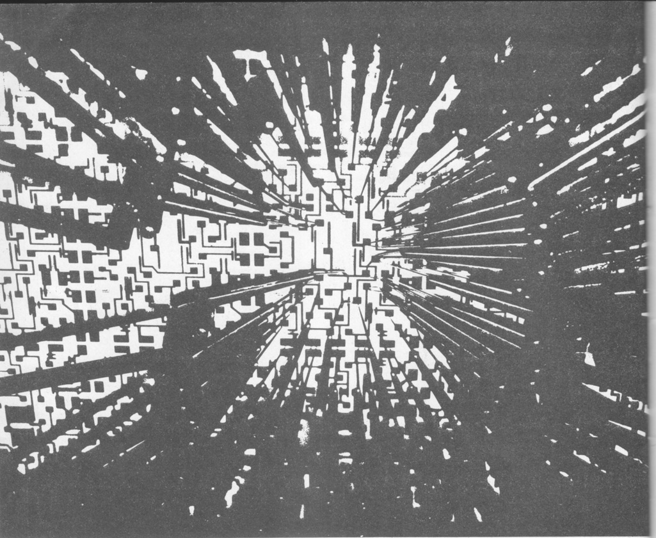
Figure 4d. This type of block diagram can't give a complete picture of the 9900 family. Texas Instruments not only supply a wide range of 'support chips'; there is also a whole series of microprocessor chips that are derived from the basic 9900. With or without RAM and/or ROM on the chip; with various types of data in- and output; for different applications. As Texas Instruments put it: 'The 9900 family is a compatible group of microprocessors, microcomputers, microcomputer modules and minicomputers'. Furthermore, in the not-too-distant future a new member of the family is expected!

4e


CONNECTION DIAGRAM
Top View
Z8001
AD0 1, AD9 2, AD10 3, AD11 4, AD12 5, AD13 6, STOP 7, μI 8, AD15 9, AD14 10, Vcc 11, VI 12, NVI 13, SEGT 14, NMI 15, RESET 16, μO 17, MREQ 18, DS 19, ST3 20, ST2 21, ST1 22, ST0 23, SN3 24
48 AD8, 47 SN6, 46 SN5, 45 AD7, 44 AD6, 43 AD4, 42 SN4, 41 AD5, 40 AD3, 39 AD2, 38 AD1, 37 SN2, 36 GND, 35 CLOCK, 34 AS, 33 DECOUPLE, 32 B/W, 31 N/S, 30 R/W, 29 BUSAK, 28 WAIT, 27 BUSRQ, 26 SN0, 25 SN1
Note: Pin 1 is marked for orientation.
CONNECTION DIAGRAM
Top View
Z8002
AD9 1, AD10 2, AD11 3, AD12 4, AD13 5, STOP 6, μI 7, AD15 8, AD14 9, Vcc 10, VI 11, NVI 12, NMI 13, RESET 14, μO 15, MREQ 16, DS 17, ST3 18, ST2 19, ST1 20
40 AD0, 39 AD8, 38 AD7, 37 AD6, 36 AD4, 35 AD5, 34 AD3, 33 AD2, 32 AD1, 31 GND, 30 CLOCK, 29 AS, 28 DECOUPLE, 27 B/W, 26 N/S, 25 R/W, 24 BUSAK, 23 WAIT, 22 BUSRQ, 21 ST0
Note: Pin 1 is marked for orientation.
LOGIC SYMBOL
Z8001
BUS TIMING
READ/WRITE
NORMAL/SYSTEM
BYTE/WORD
STATUS
BUS CONTROL
INTERRUPTS
MULTI-MICRO CONTROL
ADDRESS/DATA BUS
SEGMENT NUMBER
SEGMENT TRAP
+5V GND CLK DECOUPLE (DO NOT USE) RESET
LOGIC SYMBOL
Z8002
BUS TIMING
STATUS
CPU CONTROL
BUS CONTROL
INTERRUPTS
MULTI-MICRO CONTROL
ADDRESS/DATA BUS
+5V GND CLK DECOUPLE (DO NOT USE) RESET
8127
Clock generator
8001
Microprocessor
3-bit
Interrupts
system bus
address/data bus
A0 ... A15, S0 ... S6
D0 ... D15
Bus demuxing
logical address
8010
Memory management unit
Memory
disk memory controller
floppy disk
other support chips
Z8065 Burst Error Processor (BEP)
Z8068 Data Cyphering Processor (DCP)
Z8073 System Timing Controller (STC)
Z8163 EDC Timing controller
Z8164 Dynamic Memory, Address Multiplexer
8016: DMA controller
8030: multi-protocol communications contr.
8036: parallel interface/timer
8052: terminal controller
PHYSICAL ADDRESS
DATA
8034
Input/output processor
8038-8060
Multi-processor interface
Other microprocessors
Memory
I/O
81127 - 4e

Pin	Function
AD_0-AD_{15}	Address/Data
$\overline{AS}$	Address Strobe
$\overline{BUSAK}$	Bus Acknowledge
$\overline{BUSRQ}$	Bus Request
$\overline{DS}$	Data Strobe
$\overline{MREQ}$	Memory Request
$\overline{\mu I}$	Multi-Micro In
$\overline{\mu O}$	Multi-Micro Out
$\overline{NMI}$	Non-maskable Interrupt
$\overline{NVI}$	Non-vectored Interrupt
CLK	System Clock
$\overline{RESET}$	Reset
$R/\overline{W}$	Read/Write
SN_0-SN_6	Segment Number
$\overline{SEGT}$	Segmentation Trap
ST_0-ST_3	Status
$\overline{STOP}$	Stop
$\overline{VI}$	Vector Interrupt
$\overline{WAIT}$	Wait
$B/\overline{W}$	Byte/Word reference
$N/\overline{S}$	Normal/System Mode
Decouple	Output from on-chip negative substrate-bias generator. Presently not connected.

Figure 4e. The Z8001 also has several brothers, sisters and cousins. The 8002, 8003 and 8004 are alternative versions of the CPU itself; then, as usual, there is an intelligent 'memory management unit', 'input/output processor', and so on. It is clear from several little details that Zilog put great value on 'Multi-processor systems', where several 8001's are used in the same system — sharing memory and input/output resources. Not only does the instruction set cater for this sort of thing: it is also apparent from the 'multi/micro control' pins!



Processor type	Manufacturer	Importer	Telephone number
AMZ 8001/2	AMD	Advanced Micro Devices (UK) Ltd, AMD House, Goldsworth Road, Woking, Surrey.	(04862) 22121
S 9900	AMI	AMI Microsystems Ltd, Princes House, Princes Street, Swindon, Wilts. SN1 2HU	(0793) 37852
HD 68000	Hitachi	Hitachi (UK) Ltd, Pie Building, 2, Rubastic Road, Southall, Middlesex UB2 5LF.	(01) 574 0732/38
8086/8088	Intel	Intel Corporation (UK) Ltd, Dorkan House, Eldene Drive, Swindon, Wilts. SN3 3TU.	(0793) 26101
M5L 8086	Mitsubishi	Mitsubishi (UK) Ltd, Otterspool Way, Watford, Herts.	(0923) 40566
MK 8086	Mostek	Mostek UK Ltd, Masons House, 1, Valley Drive, Kingsbury Road, London NW9.	(01) 204 9322
MC 68000	Motorola	Motorola Ltd, York House, Empire Way, Wembley, Middlesex.	(01) 902 8836
NS 16000	National Semiconductor	National Semiconductor (UK) Ltd, 301, Harpur Centre, Horne Lane, Bedford MK40 1TR	(0234) 47147
NS 16000	Fairchild	Fairchild Camera and Instrument (UK) Ltd, 230, High Street, Potters Bar, Herts.	(0707) 51111
R 68000	Rockwell	Pelco (Electronics) Ltd, Regency Square House, 26-27 Regency Square, Brighton BN1 2F8.	(0273) 722155
Z 8001/2	SGS-ATES	SGS-ATES (UK) Ltd, Planar House, Walton Street, Aylsbury, Bucks. HP21 7QJ	(0296) 5977
SAD 8086	Siemens	Siemens Ltd, Siemens House, Windmill Road, Sunbury-on-Thames.	(09327) 85691
TMS 9900	Texas Instruments	Texas Instruments Ltd, Manton House, Bedford MK41 7FA.	(0234) 67466
EF 68000	Thomson	Thomson CSF Components and Materials Ltd, Ringway House, Bell Road, Danes Hill, Basingstoke, Hants.	(0256) 29155 ext. 232
Z 8001/2/3/4	Zilog	Zilog (UK) Ltd, Babbage House, King Street, Maidenhead, Berks.	(0623) 36131

humidity sensor

Somewhat surprisingly perhaps, detecting humidity by electronic means involves a great deal more than meets the eye. In fact, until recently, the few reliable devices that were available were too complex and therefore too expensive for widespread use. The German company, Valvo, recently released details of a capacitive humidity sensor that, inspite of what might be expected from unsophisticated circuitry and low cost, has many advantages. It can be incorporated directly into an electrical measuring circuit, will serve a variety of purposes and is easier to operate, maintain and calibrate than its mechanical counterparts. Not only will it detect humidity in the home, but also in greenhouses or tumble dryers.

Before dealing with the circuit itself, it will be as well to define humidity as such. What is known as the *absolute* humidity is the number of grammes of water per cubic metre of air at a certain temperature. The absolute or maximum relative humidity is exceeded when the atmosphere absorbs greater quantities of water, thereby becoming saturated or 'damp'. How much water is absorbed depends on the ambient temperature of the atmosphere. To give an example, living room windows tend to 'steam up' in winter time, as contact with the outside air makes them much colder in comparison with the room temperature. The amount of moisture in the air is expressed in terms of *relative* humidity. This is calculated by dividing the actual amount of water in the air by the maximum quantity at the same temperature and then multiplying the result by 100%. The relative humidity must be between 40 and 70% for plants, pets and persons to breathe comfortably and so it is important to maintain it at an optimum level. Excessive humidity will cause metals to rust and wood to rot.

For the above reasons, the sensor is designed to respond to changes in the ambient relative humidity. Figure 1 shows that the system consists of a perforated plastic case containing a stretched membrane of non-conducting foil, coated on both sides with gold. The membrane and coating form the dielectric and electrodes respectively of a parallel plate capacitor. As illustrated in the graph in figure 2 the capacitance C_s is determined by the degree of ambient relative humidity H_{rel}. This is because the layer of gold is thin enough to allow moisture to penetrate through to the dielectric, in other words, an increase in humidity will cause the capacitance to rise.

The sensor is reliable within a 10% . . . 90% humidity range. Outside these limits the sensor will have a nominal accuracy of only 5%. However, such levels should only occur in extreme cases.

The measuring circuit

Before dealing with the circuit diagram, the principle behind the operation must be considered. This is shown in figure 3. As can be seen, operation is based on the measurement of pulse width variations. The block diagram shows two synchronized multivibrators M1 and M2, which are connected to a trimmer capacitor C_T and to the

1

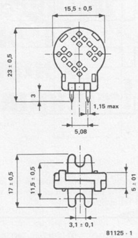


Figure 1. The capacitive humidity sensor as designed by Valvo and its dimensions in mm.

2

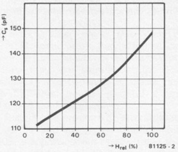


Figure 2. The relationship between relative humidity H_{rel} and sensor capacitance C_s.

3

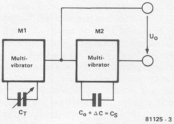


Figure 3. The block diagram of the measuring circuit.

4

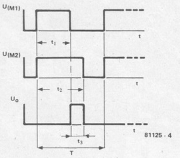


Figure 4. How pulses are formed in the circuit of figure 3. The pulse width determines the degree of humidity H_{rel}.

humidity sensor of capacitance C_s, respectively. The latter comprises a constant contribution C_o and a contribution ΔC dependent on H_{rel}, in other words

$C_s = C_o + \Delta C$.

M1 and M2 produces pulses, t1 and t2 in length, which are proportional to C_T and C_s respectively (see figure 4). What happens is that M1 synchronizes M2, so that the pulse width difference t3 is equal to t2-t1. The length of the pulse width t3 therefore determines the degree of ambient humidity H_{rel}. Thus, if t3 is fairly short, the atmosphere will only be slightly humid, whereas a lengthy t3 would mean a high degree of humidity (as in a botanical garden, for instance).

If M1 and M2 have equal proportional constants and C_T is equal to C_o, t3 will be proportional to ΔC. If the pulse frequency is set at 1/T, where $T = 2t_1$ (figure 4) and all pulses have equal amplitude U_B, then the average output voltage will be:

$\overline{U}_o = (t3/T)\ U_B = (\Delta C/2\ C_o)\ U_B$.

The term t3/T is called the relative pulse width. Its temperature and voltage dependence are very small, provided:

- the characteristics of both multivibrators are identical (constructed for example from a single 4001);
- C_s and C_T have equal temperature coefficents.

Output voltage $\overline{U}_o$ is directly related to the supply voltage which should therefore be stabilized to obtain the best results.

Practical circuit

A design based upon two 4001 IC's is shown in figure 5. The circuit may be either battery or mains powered, depending upon its application.

Multivibrators M1 and M2 are each formed by a pair of NOR gates in the first 4001. 10 kHz pulses produced by M1 and M2 are fed to the second 4001. This generates a pulsed-output voltage with an average value $\overline{U}_o$ proportional to the pulse width difference. The four NOR gates of this IC are connected in parallel to provide low output impedance. Any parasitic oscillations in the circuit will be suppressed by an RC network in the supply line (C5, C6, R3).

Linearizing network

Since the relation between C_s and H_{rel} is non-linear, the pulsed output signal $\overline{U}_o$ is fed to a linearizing network. For clarity's sake this is shown separately in figure 6. Voltage pulses charge capacitor C7 by way of diode D1 and resistor P1. At the same time a discharge current in proportion to the voltage across the capacitor flows through resistors R4 and R5, and an additional current flows from the supply line via resistor R6. Thus, the output voltage U'_o is a nonlinear function U_o and with suitable choice of C7, P1 and R4, R5, this function can be profiled to allow the relationship between H_{rel} and U'_o to be substantially linear.

With respect to the circuit in figure 5, the output voltage can vary between 80 mV and 1 V. This can be used either to indicate or to control relative humidity (H_{rel}).

Tumble drye: control

As we mentioned before, the humidity

5

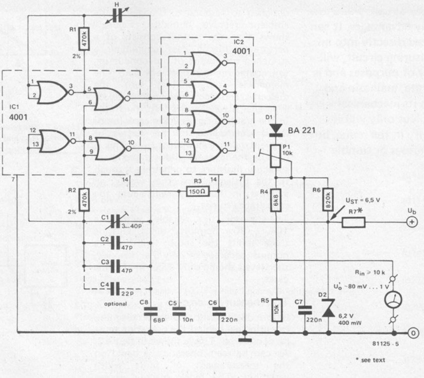


Figure 5. The measuring circuit with linearized output. The circuit can be connected to an external power supply. R7 is chosen so that $R7 \approx (U_B - U_{ST})/2\ mA\ \Omega$.

6

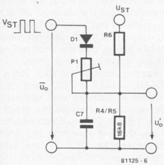


Figure 6. The linearizing network used in figure 5.

indicator may serve a variety of purposes. Let us therefore see what happens in the case of a tumble dryer.

A tumble dryer operates by heating a damp load whilst tumbling it slowly in a rotating drum. The H_{rel} at the air outlet provides a reasonable indication of how damp the load is. The measuring circuit described above can be made to control the dryer, switching it off as soon as the load has reached a certain (preset) level of dryness.

The circuit operates by comparing U'_o with a constant voltage, in this instance a preset voltage that corresponds to the required level of H_{rel} (in other words, the level indicating a dry load).

The humidity sensor is situated in the air outlet of the dryer and an NTC thermistor is located in the drum. The thermistor is used to control the air temperature within the drum. When this exceeds 60°C the heater will be switched off and when it drops below 50°C the heater will be switched back on again. The on/off switch is door-operated so that drying starts the moment the door is closed and stops whenever it is opened.

7

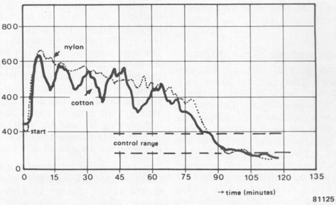


Figure 7. What happens during the tumble dryer control operation. The relative humidity in the tumble dryer falls as the load dries. The curves here relate to a fully loaded standard dryer as used in the home.

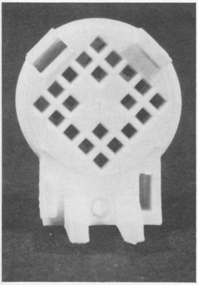


Photo 1. The humidity sensor as manufactured by Philips.

The relative humidity will obviously rise when a damp load is inserted. It must, however, be prevented from rising to such a extent that the dryer is activated. For this reason a delay circuit is included to hold U'_o above the preset voltage for about 2 minutes after the door is closed. The dryer can then start and run for sufficient time for the humidity at the outlet to rise above the preset value, after which the humidity sensor will control the operation.

Figure 7 shows how the relative humidity at the air outlet varies with time. It increases as soon as the motor starts, then gradually falls until the load reaches the required level of dryness and the dryer switches off.

Source: Technical information 063, Valvo Ltd. ◼

we haven't forgotten the TV games computer!

In the 'Junior Cookbook' article, elsewhere in this issue, we mention the fact that our desks are being snowed under with letters requesting further extensions to the Junior Computer. The same applies to the TV Games Computer — admittedly, to a slightly lesser extent.

'When will you publish a memory extension board?', 'Please put some more games on tape!', 'Any hardware extensions will be welcome!', 'How do you get score displays on the screen?' . . . and so on.

Rest assured, we haven't forgotten you! A memory extension board is in the final development stage, and we hope to publish it in June. An extensive 'sound effects generator' will be included on the same board .

More programs? We've got loads of them! Unfortunately, they often require more or less elaborate modifications, before they are suited for the ESS service. In this connection, please bear the following points in mind if you intend to send us programs:

- Record the program at several levels, up to and even over the nominal 100% mark. We often receive tapes that are recorded at −10 dB or even −20 dB, and it can take hours of fiddling before we can load them successfully.
- All joysticks are different. If they are used in a program, a joystick calibration routine (like that described in an earlier article) is virtually a 'must'.
- A full listing is important. If the program is perfect as it stands, well and good; but when modifications are required, it may go to the bottom of the pile until we have time to 'disassemble' the relevant sections.
- Bear in mind that Elektor is an international magazine. Texts will often enhance a program, but they may need translation — although English, fortunately, is a very international language. Explanation where the text routines are and how to modify them saves us a lot of time.

More information? We often receive requests for fuller explanation of score routines, collision detection, the monitor software, etc. To cater for this, we intend to publish a TV Games Computer book. Hopefully, we should be able to 'assemble' it within the next few months. ◼

logic analyser II

Last month, the basic principles of the logic analyser were explained with the aid of block diagrams. Now the moment has arrived to see what the actual circuit diagrams look like. Again, the unit has been split up into two sections: the logic analyser itself and the cursor circuit. This makes it easier to 'place' the various parts previously shown in the block diagrams.

The logic analyser block diagram was by no means straightforward and so it is not surprising that it leads to a rather elaborate circuit diagram. For this reason, it is advisable to look up the previous article and have the block diagrams at hand, so as to see which components represent which block. Otherwise, the circuit diagram is bound to cause confusion.

To start with, a few general remarks. The logic analyser consists of three main sections: the analyser, the cursor and of course the power supply. Wherever possible, LS TTL ICs have been used to limit current consumption. The printed circuit boards and the constructional details will be dealt with next month. The circuits are more than enough for now.

The logic analyser circuit diagram

For a change, we'll start in the middle of figure 1 (that is, the lower left-hand corner, to be precise) — at the heart of the logic analyser. This consists of the clock oscillator and the time base switch. With the given capacitor values, the voltage controlled oscillator IC9 produces a frequency of 4 MHz. The oscillator's stability can be improved considerably by replacing capacitors C7 and C8 by a 4 MHz crystal. With the aid of divide-by-two and divide-by-five stages (IC12 . . . 14a) different sampling rates are obtained from the oscillator frequency. The desired rate can be selected with switch S1. S2 enables the division ratios of the dividers to be changed, thereby extending the number of rates. The final position of S1 (K) is connected to gate N12. An external clock can be connected to this. The position of S3 will then determine whether the circuit reacts to the positive or to the negative edge of the external signal.

Table 1 lists the sampling rates for the different positions of S1 and S2. Gates N20, N22 and N23 are used to switch from the selected sampling frequency to the fixed scan frequency, and vice versa. For this purpose one input of N20 is connected to the Q output of FF2 and one of N22's inputs is connected to the $\overline{Q}$ output. Thus, the state of FF2 will determine which signal is passed.

What about the inputs? The eight data inputs are connected to the latch (IC1). This transfers the input data to the outputs, at the sampling rate determined by S1 and S2. The delay time that elapses between the sampling pulse and the data transfer can be preset with the monostable multivibrator MMV1. The delay switch S19 allows for two alternatives. When it is in position a, the delay will be fixed at 50 ns; when it is in position b, the delay can be adjusted between 150 and 500 ns by means of P1. The A input of MMV1 is connected to the output of N22 which produces the sampling frequency during data entry. In the display mode, N22 is blocked so that the latch will not receive any sampling pulses either.

The memory consists of two 256 x 4 bit RAMs, 2101A-2 (IC2 and IC3) which have to perform their utmost in this design. This is because the shortest sampling time is 250 ns which happens to be the shortest time that the RAMs can process. The data at the latch outputs is passed to the memory ICs, four lines leading to IC2 and another four to IC3. The addresses are provided by IC4 and IC5. Together they constitute the 8 bit counter A shown in the block diagram. This counter continually scans all the addresses in the memory, as its clock inputs receive the sampling or scan frequency from N23.

Back to the trigger section. After passing the latch, the data is also applied to the word recognizer. This consists of gates N1 . . . N10 and switches S5 . . . S14. The open collector outputs of the gates are all interconnected and are linked to the positive supply through R11. This means FF1 will only be provided with a trigger pulse when all gate outputs are logic 1. Two external trigger inputs are included via N1 and N2. One input of each of the remaining gates (N3 . . . N10) is connected to one of IC1's outputs.

The other input of every EXNOR is connected to +5 V via a 5k6 resistor and to the centre contact of a three-way switch. When the switch is on a, the input connected to it will be logic zero; logic one corresponds to the centre position, and in position b the two inputs of the gate will be linked. The three switch positions are labelled 'L', 'H' and 'x' (both in the circuit and on the front panel). In position L, the output of the corresponding gate will only become high if that of the latch is low. Similarly, 'H' indicates that the output from the latch must be high; finally, in the 'x' (don't care) position the EXNOR output will always be logic one, regardless of the latch's state. The switches can therefore be used to preset an eight-bit pattern, or 'word'. As soon as this word appears at the outputs of the latch, FF1 will be triggered by the word recognizer. That is, assuming the two switches for the external trigger inputs are on 'x'.

When FF1 is triggered, either by the word recognizer or by the 'manual trigger' key (S15), the analyser is switched to the 'display' mode. The reset key S16 resets the flipflop and thus the analyser. LED D1, driven from the $\overline{Q}$ output, will light as soon as the analyser is triggered.

The data outputs of the RAMs are connected to the inputs of the 8-to-1 multiplexer IC6. Counter C (IC14b) determines which input of the multiplexer is connected through to the

Table 1

Sampling rates

S1	S2	
	a	b
a	250 ns	250 ns
b	500 ns	500 ns
c	500 ns	1 µs
d	2.5 µs	5 µs
e	5 µs	10 µs
f	25 µs	50 µs
g	50 µs	100 µs
h	250 µs	500 µ
i	500 µ	1 ms
j	2.5 ms	5 ms
k	EXT	EXT

1

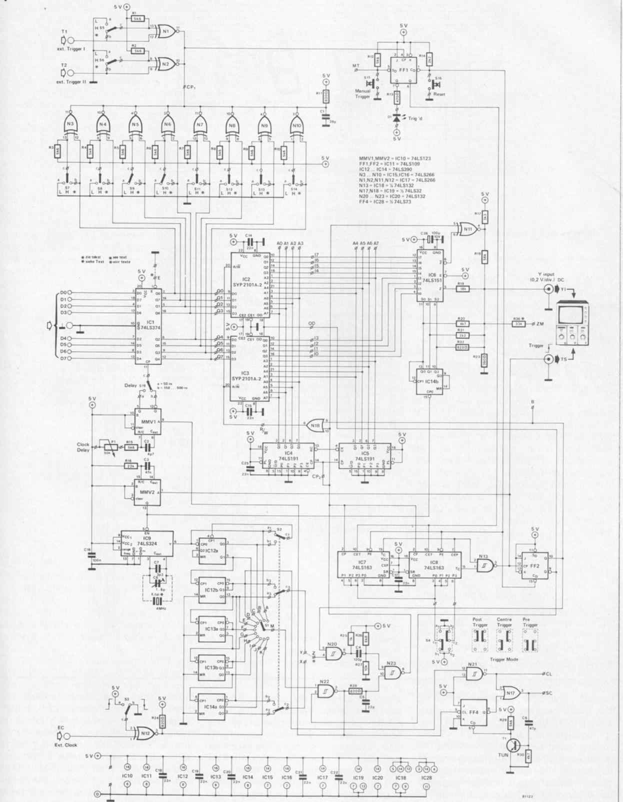


Figure 1. The logic analyser circuit diagram does seem rather elaborate, but then it offers quite a few interesting facilities.

2

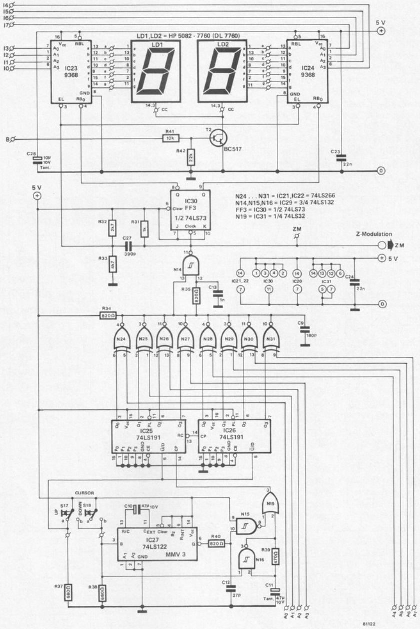


Figure 2. The cursor circuit, a great help when reading data on the screen.

output. The D/A converter shown in the block diagram is not immediately apparent in the circuit. In actual fact, it consists of R19 . . . R23. These resistors sum the data output from the multiplexer and the outputs of counter C, all in the correct proportion to obtain the analogue voltage required for the y input of the 'scope'. A further signal is also mixed in at this point (via N11 and R18) to produce a 'dotted line' between the logic 0 and logic 1 levels, as can be seen in the photos of the display.

Counter B in the block diagram appears as IC7 and IC8 in the circuit. From the trigger signal and the preset of S4 it ascertains when the data entry into RAM is to stop. Furthermore, the counter ensures that the memory is read out again in the same (correct) manner. The 'trigger mode' switch S4 sets the 'preset' inputs of the counter to 0, 126 or 254, as explained last month. When counter B generates a carry pulse, FF2 is triggered; in turn, this flipflop switches the system from 'sampling' to 'scan' mode.

To sum it up

Having located all the sections, running briefly through a complete load and display cycle may help to clarify things. Initially, the desired sampling rate is set by S1; a 'trigger word' is programmed into S7 . . . S14 (S5 and S6 are set to 'x'); S4 is set, say, to 'centre trigger'.

Operating the reset key (S16) clears FF1 and FF2, and causes the preset (126 in this example) to be loaded into counter B (IC7 and IC8). The data now starts to appear 'byte-by-byte' at the memory inputs, via IC1; the memory ICs (IC2 and IC3) are set to 'write' mode and counter A (IC4, IC5) cycles continuously through the full address range, so that each incoming 'sample' is stored at the next higher address.

If and when the incoming data is identical to the combination set up on S7 . . . S14, the word recognizer will trigger FF1. LED D1 now lights, and counter B (IC7, IC8) is enabled. This counter starts to count sampling pulses, starting from the preset number (126), until it reaches 255. Depending on the preset a further 1, 129 or 255 samples are required. It then gives a 'carry', toggling FF2.

Simultaneously, MMV2 is triggered; this stops the clock oscillator (IC9) for a short period. The circuit is now in 'display' mode: as soon as the clock is started again, the memory will be 'read' at the fixed scan frequency (200 kHz).

During the first scan, one of the data lines is selected by the multiplexer and is displayed on the screen. At the end of this scan (after 256 bits, in other words), counter B again produces a carry pulse. As before, this stops the 'clock', counter C (IC14b) is incremented to select the next data line, the clock is re-started and the 'scope receives the next trigger pulse.

From the above it will be clear that the eight traces are not displayed simultaneously on the screen — how could they, on a single-channel 'scope'? However, they are 'multiplexed' at such a high rate (less than 10 milliseconds for a complete 8-channel display) that they all 'appear' at the same time.

So much for figure 1. The handful of components in the lower right-hand corner (N17, N21, FF4, etc.) will not be dealt with here. They are part of an extension circuit that converts the logic analyser into a 'storage oscilloscope front-end'. Hold your horses! We'll get to that in two or three months.

3

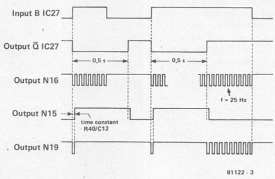


Figure 3. This pulse diagram gives a good idea of the way in which the pulse generator in the cursor works.

The cursor

The cursor circuit is shown in figure 2. The wire links that connect it to the main circuit in figure 1 are labelled A0 . . . A7, I0 . . . I7 and B. In the block diagram these correspond to the connections to RAM, counter A and FF2.

The two displays LD1 and LD2 are controlled by IC23 and IC24. These binary-to-seven segment converters convert 8-bit data into two hexadecimal numbers. Each converter is connected to four data lines in memory. The common cathodes of the two displays are switched by T2. The base of this transistor is connected to the Q output of FF2. As a result, the displays only light if data is being written onto the screen.

The idea behind the cursor is that an address can be selected; the data at this address must appear in the displays, with some position indication on the screen.

The circuit that recognizes the preset address is very similar to the word recognizer circuit. Gates N24 . . . N31 compare the contents of counter A (IC4 and IC5) to the contents of counter D (IC25 and IC26). When they are identical, the output of the comparator circuit will become logic one. Via N14, this causes the display decoders to read the data. FF3 switches at every pulse generated by the comparator circuit: once for each complete memory scan. This multiplexes the displays.

The output of N14 can be connected to the Z modulation input of the oscilloscope. As a result, eight brighter dots appear on the screen in a vertical line, one on each scan. These dots indicate the position of the data being displayed as two hexadecimal digits.

The contents of counter D, which determine the position of the dots on the screen and the data on the displays, can be preset with pushbuttons S17 and S18. They operate the 'cursor control' which produces the clock pulses and the up/down signal for counters IC25 and IC26. When S17 (up) is depressed the up/down signal becomes logic zero; the counter will now count the pulses that appear at its clock input. If, on the other hand, S18 is operated (down), the up/down signal becomes logic one and the counter will count down.

The up/down pulse generator may look a little complicated but this does provide some interesting facilities. If either S17 or S18 is depressed for less than 0.5 seconds, only one pulse will be sent to the counter and the cursor will only shift one position. If, however, one of the two switches is held down for longer, a 25 Hz frequency will start to appear at the output of N19 and the cursor then moves left or right across the screen at a much higher speed. This is achieved as follows. When a key is operated, the output of the oscillator around N16 immediately becomes logic zero. At the same time, MMV3 is activated to make its $\overline{Q}$ output '0' as well. The R40/C12 combination briefly delays N15 from reacting to the '0'. Consequently, N19 passes N16's logic zero to the counter. After this short interval, the output of N15 becomes logic one and the oscillator signal is therefore blocked by N19. When the MMV time (0.5 seconds) has elapsed, the output of N15 becomes low again and the oscillator frequency of 25 Hz is passed to the counters. Figure 3 should help to clarify this.

Z modulation can take place in various ways. If the oscilloscope has a suitable connection, the corresponding output can be connected to it. In some cases this may require an inverter, depending on the type of oscilloscope used. If, however, the oscilloscope does not have a Z modulation input, the cursor can be made visible by including resistor R36 in the circuit. The cursor will then be represented on the screen as a slight dent on every line.

The power supply

The Junior Computer's supply (Elektor, May 1980, p. 5-11) turned out to suit the circuit perfectly, so there was no need to design a new version. The logic analyser only requires the +5 V section, but the +12 V and −5 V supplies will be used for the 'storage scope' extension board we mentioned earlier.

In the next episode . . .

Obviously, not everything in the logic analyser circuit diagram could be discussed in full detail. Nevertheless, we hope the circuit's operation will now be clearly understood. In the next issue, the final episode of the logic analyser saga will include constructional details, printed circuit boards and a front panel design.

After that, we will come to the 'storage oscilloscope'. ◼

crystal-controlled stroboscope

Gramophone records are supposed to be played at exactly 33 ⅓, 45 or 78 RPM, as the case may be. Nowadays it is usual for record player manufacturers to leave the final calibration of their product to the user, by providing a 'fine speed control'. However, this means that the user needs some clear indication of the turntable speed and common practice is to include a stroboscope with a speed calibration disc. This is very cheap and extremely accurate — provided the stroboscope is running at the correct frequency! Normally the mains frequency is used, but this is not as reliable as one might expect. A crystal-controlled stroboscope is a far more accurate solution.

Why it is so essential for a turnable to have the correct speed? Well, for the simple reason that the slightest deviation will affect all the frequencies and tempi on the record. In other words, the pitch may change. This can, of course, lead to various interesting 'special' effects, but it is hardly 'high-fidelity'!

The above can be avoided by using a crystal-controlled stroboscope. This can be used to calibrate the turntable speed if some means of motor speed adjustment is provided. This type of record player is often equipped with a separate stroboscope disc (see figure 1) that can be placed on the turntable. When this is illuminated by a (mains driven) light bulb, a correct speed adjustment will produce a stationary image. Alternatively, the stroboscope may be situated on the rim of the turntable (figure 2). It is then lit by a small built-in lamp which is connected to the mains.

Unfortunately, mains powered stroboscopes suffer from a couple of disadvantages. First, the mains frequency is not very constant over a short period of time (only long-term accuracy is required to keep clocks and the like running on time). In the second place, the image which appears on the stroboscope disc is often rather blurred. This is because the stroboscope lamp is supplied with a sine wave derived from the mains, which causes a fairly slow transition to take place from light to dark, and vice versa. This effect is aggravated by the length of time it takes the light bulb to light up and then fade. It means the brightness is fairly evenly distributed throughout the period that the bulb is lit, so that no peak intensity will be reached. As a result, the image on the disc is bound to become 'fuzzy'. Better results can be obtained with a neon bulb, although the mains frequency will of course still be inaccurate. Even better is to use a crystal-controlled stroboscope. Having a crystal act as a reference source enables the speed to be adjusted with maximum precision.

In this circuit the disc is illuminated by three red LEDs. These have an ad-

1

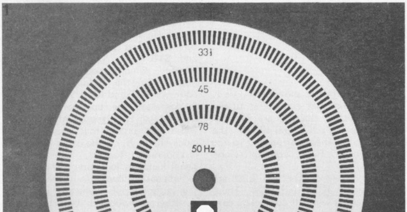


Figure 1. A full-size stroboscope disc used to adjust 33 ⅓, 45 and 78 RPM on record players. The 50 Hz indicated refers to the mains frequency for which the disc was designed.

2

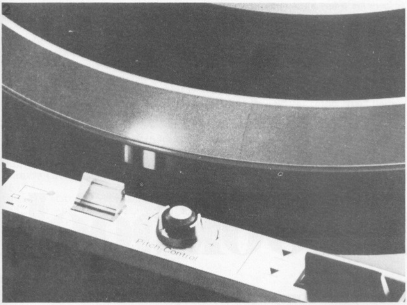


Figure 2. A stroboscope that is printed on the rim of the turntable. This particular record player has its own light source.

vantage over normal light bulbs, in that they light up and extinguish very quickly, creating a clearly outlined image. The effect is heightened by powering the LEDs from a square wave voltage with an amplitude in the 9 . . . 12 volt range. The crystal stroboscope will then produce a symmetrical 'square wave' light output, in other words, it has a clearly defined light-to-dark ratio.

Stroboscope discs are normally designed for an illumination frequency of 100 Hz. This may sound surprising, as the frequency of the mains voltage is 50 Hz. Nevertheless, a lamp (or neon lamp) lights up every half period, so that the illumination frequency will be double the mains frequency (100 Hz).

The circuit diagram

The crystal stroboscope (see figure 3) is fairly straightforward, as can be seen from the circuit diagram. IC1 contains an oscillator and a 2^{14} divider. Provided the oscillator loop is correctly calibrated with C1, the output (Q_{14}) will produce a 200 Hz square wave (3.2768 MHz ÷ 2^{14} = 200 Hz). The square wave voltage is divided by 2 by IC2 and the 100 Hz frequency, required to switch the LEDs on and off, will appear at the base of T1.

Resistor R3 has a low value to allow plenty of current to pass through the LEDs and so provide sufficient light.

Since the device only consumes about 25 mA, it can be battery-powered.

Calibration

A precise frequency meter — perhaps you can borrow one? — is an absolute must where calibrating the stroboscope is concerned, as it needs a 6-digit display at least. The frequency meter is connected to testing point TP (pin 7 of IC1). Trimmer C1 is then used to adjust the frequency to exactly 204,800 Hz. If a frequency meter is not available, C1 can either be placed in the middle position or be replaced by a fixed 12 pF capacitor. The frequency deviation will then be not more than 0.01%.

Construction

Once the circuit has been built (on Veroboard, for instance) and calibrated, it can be inserted into an (old) torch. There will often be enough room for a 9 V 'power-pack' battery as well. The switch on the outside of the torch can then act as S1. The three LEDs are mounted very close together in the torch bulb's place. If your record player already has a stroboscope (either a bulb or a neon lamp), this can be replaced by the crystal-controlled version.

It should be noted that the speed must be adjusted while a record is playing. Place the record on the turntable first and the disc on top of it. The disc's diameter may not exceed that of the record label, as otherwise the running-out groove will be covered. ◀

3

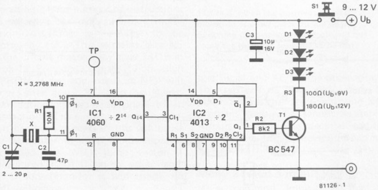


Figure 3. This crystal-controlled stroboscope makes it possible to adjust the speed of a record player with great precision. The LEDs, supplied with a square wave voltage, produce a better, clearer image than any light bulb or neon lamp would; furthermore, the mains frequency is not as accurate.

junior cookbook

a few healthy recipes to keep your computer in shape

With thanks to H.P. Diehl and H.D. De Mülder

Our Elektor staff desks are covered in piles of letters, the telephone never stops ringing and even telexes are streaming in by the dozen. If this carries on, the offices will have to be evacuated . . . And all because of the Junior Computer! Book One whetted people's appetites to such an extent that they just cannot wait for Book Two; they are craving to try out more Junior recipes . . . NOW!

Far be it from us to let anyone starve, so here are a few hints, alternatives and other useful 'brain food' to keep you and your computer going until the main course, Book Two, arrives.

Menu 1: Decimal arithmetic

As you will remember from Book 1, subtractions and additions may be carried out both in binary and in decimal. Whenever instruction SED (F8) for decimal appears in the program, problems will arise if the computer jumps back to the monitor (1CØØ) via a BRK at the end of the program or when operating in single-step mode. What happens is that if you attempt to return to the monitor when the D flag is 1, keys F, +, AD, DA, PC and GO will have become ineffective; the key *functions* A . . . F will be lost.

The key function AD will be taken over by key A, B has become DA, C is now the + function, D is GO and E will assume PC's function. In other words, the numeric key functions A . . . F are no longer available. Addresses containing A . . . F cannot be entered directly, but only by means of a slight detour. The solution is to type in the nearest lower address below the one required that uses decimals Ø . . . 9 only, and increment it as often as necessary by depressing the + key function (key C!). . Like going from London to Liverpool via Newcastle.

But how can this happen? During the monitor subroutine GETKEY the key value is determined by adding Ø7 nought times, once or twice to a basic value. Everything works out fine provided this takes place in binary, but when the computer tries to do it in decimal the key codes get mixed up.

One solution to the problem is to make sure that the SAVE routine of the monitor contains the instruction CLD (opcode D8), like the RESET routine. This will do no harm to the program, as the D=1 situation is kept in the saved P register (ØØF1) and it will be restored upon the computer's return from the monitor (section GOEXEC). This will mean changing the EPROM as follows:

```
1C1A 4C 32 1C  JMP-START
1C31 78        SEI
1C32 D8        CLD D=Ø START (tempo-
               rary) binary arithmetic
```

As a result, the central START section of the monitor will now begin with CLD. After SAVE the machine will 'convert' back to binary and all keys will retain their normal functions. By the way,

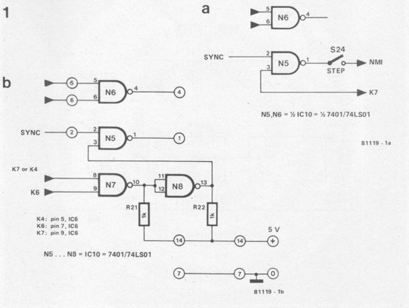


Figure 1. The hardware necessary to hold up the step-by-step procedure whenever the computer is either in the monitor (1a) or inside one of the two memory ranges activated by the two chip select signals, K4 or K7/K6 respectively (1b).

there are no plans to start supplying modified EPROMS, as there are several other alternative solutions to the problem.

It might be useful at this stage to look at an example of decimal arithmetic. For this we will use the adapted addition program given on page 68 in Book 1:

```
0100  18     CLC
0101  A9 13  LDA # 13
0103  F8     SED       D=1 decimal arith-
                       metic
0104  69 08  ADC 08
0106  D8     CLD       D=0 binary arithmetic
0107  00     BRK
1A7E  00     IRQ vector points to monitor
1A7F  1C
```

Once the start address has been entered and GO has been depressed, the program is run; it leads to a jump to the monitor. The keys may now be used as normal, as after the addition the computer went back to binary. Typing in address 00F3 will give the result of the addition (21).

Alternatively:

```
0100  18          CLC
0101  A9 13       LDA # 13
0103  F8          SED           D=1 decimal arithmetic
0104  69 08       ADC 08
0106  00          BRK
1A7E  00          IRQ vector points to 1A00
1A7F  1A
1A00  D8          CLD           D=0 binary arithmetic
1A01  4C 00 1C    JMP-SAVE jump to monitor
```

At the end of the program the BRK will bring the computer to 1A00 by way of the IRQ jump vector. After going back to binary it will jump to the monitor. If the program is to be run in one go, the method we have just described is not so suitable; if it is stepped through, however, from 0103, this is in fact the only feasible method.

With reference to the latter (step mode), a few aspects have to be considered. The hardware will have to be modified, for instance. This is because it is prohibited to step through the monitor. The monitor's task involves executing a large number of instructions in a continuous cycle (display multiplexing, waiting for a key to be depressed, etc.). This explains gate N5 in the circuit diagram on page 14/15 in Book 1. Provided signal K7 is high (EPROM is not addressed) an NMI will occur after every SYNC pulse (generated during the op-code phase of an instruction). Once the current instruction has been processed, this will cause the computer to jump to the monitor (provided the NMI jump vector is pointing to 1C00). If, on the other hand, K7 is low (monitor is addressed), no NMI will occur.

The forthcoming expansion possibilities include a printer monitor comprising the address range 1000 . . . 13FF. This is selected by K4. The printer monitor may not be stepped through either; in other words, the circuit around N5 will have to be expanded. The details are given in figure 1b. Now there are two ways in which to block an NMI via SYNC. Either K4 or K7 will be used for one signal and K6 for the other. (Figure 1b itself will be dealt with later). The point is here that by connecting K6 a program in page 1A cannot be stepped through either. The decimal addition program mentioned previously can now be stepped through.

2

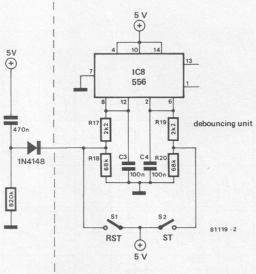


Figure 2. The 'automatic alarm clock', formed by the addition of three components, to wake up the Junior Computer when it is to be started automatically (RESET) after being switched on.

3

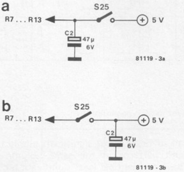


Figure 3. When the positive end of C2 is moved to another place on the board, switching the display on and off will not cause the memory contents to be lost.

Menu 2: Add a minus key

It is common knowledge that the address to be displayed during the monitor routine can be incremented by one by depressing the plus key. This can be done as often as is required. When checking a program, the user may well like to know what the previous address contained.

On such occasions it would be an advantage to have a *minus key* function. This can be achieved by using the STOP/NMI key. The NMI jump vector (1A7A and 1A7B) is now pointed at address 1A00, where the following program is located:

```
MIN 1A00 48       PHA           save accumulator on stack
    1A01 A5  FA   LDAZ-POINTL   fetch low order address byte
    1A03 D0  02   BNE ADL       on to ADL if A≠00000000
    1A05 C6  FB   DECZ-POINTH   decrement POINTH by 1
ADL 1A07 C6  FA   DECZ-POINTL   decrement POINTL by 1
    1A09 68       PLA           restore A to original value
    1A0A 40       RTI           return to monitor
```

Thus, depressing the STOP/NMI key during the monitor allows the address in the display to be decremented by one. It is now also possible to enter data via the DA key in the reverse order, that is to say, per decremented address, as the minus key, like the plus key, operates irrespective of whether the machine is in the data or in the address mode.

Menu 3: Automatic start

Depress the RST key as soon as the Junior Computer is switched on and the monitor will be ready for use. Using the circuit designed by Mr. H. Diehl which is shown in figure 2, the RST does not need to be depressed, because the computer is reset automatically. It does mean adding another three components to the main board, but these should be able to be squeezed in near the RST key.

Menu 4: Display switch

The display switch (S25 in the main circuit diagram on pages 14/15 in Book 1) enables the display to be switched off. If, for instance, you've been having a 'computer session' until past midnight and you'd like to call it a day, but wish to continue on the following evening without having to type in all the data again, the Junior Computer may be left on, but the display must be switched off. Leaving

4

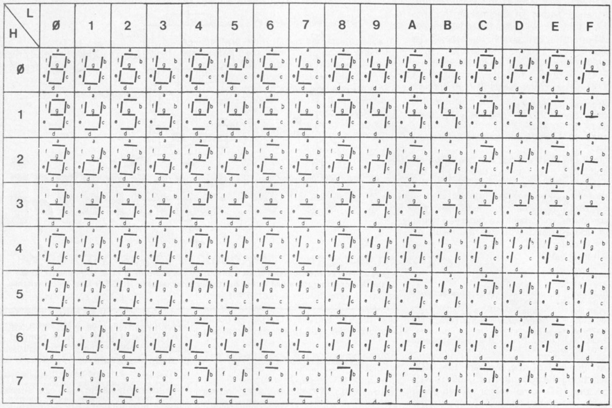

L
H
Ø
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
Ø
1
2
3
4
5
6
7
a
b
c
d
g
81119 - 4

Figure 4. A useful table showing the full 128 segment pattern possibilities.

the display 'on' might have a detrimental effect on its lifespan. When ready for another session, the display is switched on again. (Obviously, this will all be superfluous, once the cassette interface can be connected and all the data can be stored on tape).

It rarely happens, but something could go wrong when the display is switched on again. Capacitor C2 (figure 3a) must then be recharged to top supply level. Closing S25 can in some cases lead to a considerable spike on the +5 V supply lines. As a result, the RAM memory contents, so carefully preserved until then, may be lost. Then everything will depend on *your* memory whether the damage can be undone!

This headache can be remedied by connecting the positive end of C2 to the other side of S25. C2 is shown on the board next to the S25 connections.

Menu 5: Another Junior text dish

In response to the 'text display on the Junior Computer' article published last month in Elektor, Mr. H. De Mülder has kindly sent us a useful table (see figure 4). This presents the full 128 different possible combinations of segment patterns. Patterns in the same row have an identical most significant nibble (H) and patterns in the same column all have an identical least significant nibble (L). What it comes down to is that the bit corresponding to a segment that is lit must be *zero*

Menu 6: A few constructive recipes

To start with, let's look at the pin assignments of the ICs belonging to the main board (top view) and to the seven segment displays, as shown in figure 5. The pin assignments of the expansion connector and the port connector were already dealt with in Appendix 4 of Book 1. They are useful aids to have when servicing the computer or when tracking down errors.

Several thousand Junior Computers have been built by now, both in the UK and abroad, and so far remarkably few problems have come to light. This does not mean, however, that the Junior Computer is fool-proof, as the following description plus photographs illustrates.

The mind boggles . . .

Does anything in these photographs strike you as odd? Is it a U.F.O. perhaps? Take another swig, readjust your glasses and look again. Surely not . . . ???!! Yes, right second time. It's a slightly unusual Junior Computer power supply, seen from a 'different' angle. It was sent to our editorial staff for 'repair' by a reader who invested a fair amount of artistic ingenuity and good faith into its construction. The work of art which is currently on show at the Museum of Modern Art, New York, will not work for some reason. However, it is full of surprises:

- The unconventially placed heat sink is a real treat (or threat?) to the eye. Being rather ebullient in nature, it was deemed necessary to 'pin' it down lest it should escape — heat tending to rise. The nails chosen for the purpose are highly effective, as they are 10 cm long and ½ cm in diameter. This however did not satisfy the artist who thought 'better safe than sorry' and soldered them to the board. As a result, the heat sink now maternally screens the voltage stabiliser IC, LM 309, from all outside influence.
- IC1 (79L12) unhappily tried to take his (78L12) brother's rightful place on the board. For this heinous offence he was executed on the spot.

1

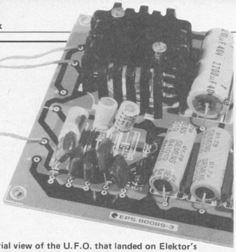


Figure 6. An aerial view of the U.F.O. that landed on Elektor's doorstep, seen from the east at a safe distance.

2

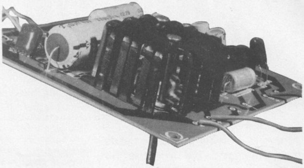


Figure 7. The same mysterious object viewed from the south-west.

5

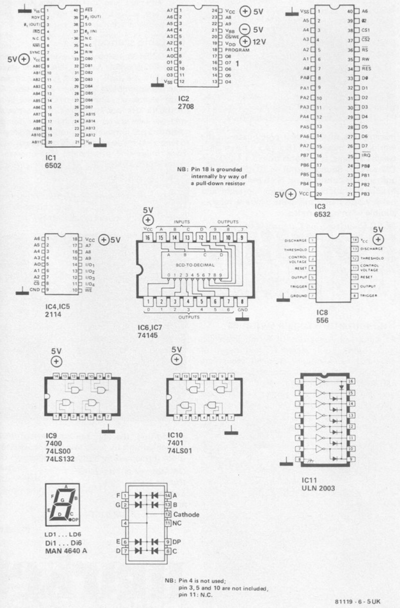


Figure 5. The pinning (top view) of all the ICs used on the Junior Computer's main and display boards.

- As can be seen, the dark capacitor family is very close and has such high aspirations that it now hovers some 10 mm above the board.

Seriously though, we all make mistakes at some time or another . . . Regrettably, refusing to ask for advice means components have to die an untimely (and expensive!) death. This can be avoided by ringing up 'Technical Queries'. Why muddle through and let yourselves in for unpleasant and often explosive shocks?

If in doubt, give a shout!

Our editorial staff will be pleased to assist you with your projects at all times. ◼

Earlier articles:

1. *The Junior Computer; May 1980, p. 5-08. Introducing the computer.*
2. *8K RAM + 4, 8 or 16K EPROM on a single card; Elektor September 1980, p. 9-04. Universal memory expansion.*
3. *Elektor µP's; Elektor May 1980, p. 5-04. The various microcomputer systems published in Elektor.*
4. *The Junior Computer memory card; Elektor, October 1980, p. 10-22. How to connect the card to the Junior Computer.*
5. *The Junior Computer Book 1, the practical introduction to a powerful system; Elektor Publishers Ltd. ISBN 0 905705 05x. Copyright 1980.*
6. *Junior's growing up; Elektor, February 1981, p. 2-16. The short and long-term outlook.*
7. *Text display on the Junior Computer; Elektor March 1981, p. 3-36. A seven-segment alphabet enables written texts to be displayed.*

coming soon...

Computer extension boards

Junior is growing up, so he needs more memory. He will also be provided with a cassette and a printer interface, plus the associated software.

The TV games computer could also do with more RAM, and this will be taken care of in the near future.

☆

Junior Computer Book 2

It's on the way — details will be announced shortly. Editing, assembling and the monitor routines are all dealt with in detail.

☆

Speech synthesis

In theory and practice. It has always been a challenge to make a computer talk — and now it is becoming feasible, at a reasonable cost.

☆

And more . . .

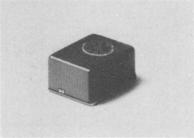
- camping c(l)ock
- square/sinewave generator
- storage 'scope
- scrambler
- measure miles on the map

☆

market

Mini audible alarm

A sub-miniature IC Electronic Buzzer type MMB-01 is now available from STAR.
High efficiency is obtained in this small-sized buzzer through the use of a custom IC. As all circuitry is solid state, the MMB's simplicity of design provides high reliability not found in conventional 'make and break' type alarms. There are no mechanical points to arc or which require maintenance, and R.F. noise is nonexistent.



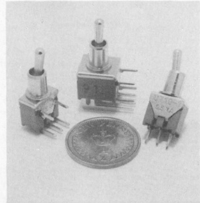
Possibly the smallest electronic buzzer in the world, it measures 18 x 14 x 11 mm and weighs 5 grams. It operates from a 1.1-2.0 volt DC supply. Sound output is 70 dB at 20 cm and current consumption is 15 mA maximum making it suitable for use on portable and battery-operated equipment where a reliable audible warning is needed.

STAR Manufacturing Company Ltd.,
Walmar House, Room 313,
296 Regent Street,
London W1.
Telephone: 01 - 637 0555

(1922 M)

Sub-miniature toggle switches

P. Caro and Associates of Birmingham can now offer from stock a new range of advanced sub-miniature switches which are designed for direct PC mounting. Called the UT series, the switches are all three position and can be provided in single or double-pole double-throw versions with straight or right angled terminals.



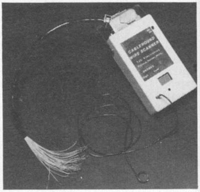
All terminals have standardised 2.54 mm centres and the switches have integral stand-off spacers for maximum stability and versatility for cleansing etc. Contacts have a minimum of 1.3 microns gold plating over 3 microns nickel. Switches have a total projection above the PC mounting base of only 20.5 mm with body sizes of 5.5 mm x 8 mm and 9.5 mm x 8 mm for the single pole and double version respectively. This degree of miniaturisation facilitates a much closer stacking of PC boards as well as permitting maximum density mounting of the switches themselves.

P. Caro & Associates Ltd,
2347 Coventry Road,
Sheldon,
Birmingham B26 3LS,
Telephone: 021-742 1328.

(1931 M)

Portable scanner

The CWS100 battery powered wire scanner from OK Machine & Tool (UK) Ltd. greatly simplifies the task of identifying wire ends when assembling cable harnesses and eliminates errors. One end of the cable, wired in the normal way, is plugged into the scanner and the free ends are then ready for identification. This is achieved by a high impedance connection made through the operator's body by a 'finger ring' attachment. The operator is then able to search for the required wire by touching each end.
An automatic sequential step allows the user to identify all wires in a bunch until the correct one is found, and numbers are indicated on a liquid crystal display.



In addition, the unit can be used to test an assembled unit by using low impedance which is not influenced by body contact.

OK Machine & Tool (UK) Ltd,
Dutton Lane,
Eastleigh,
Hants SO5 4AA,
Telephone: 0703 610944

(1933 M)

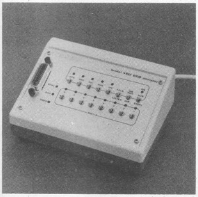
market

GPIB Analyser

Model 4881 GPIB Analyser announced by WASEC is an invaluable accessory for all personal computers which use the IEEE 488 bus for peripheral interconnection. With Talker, Listener and Controller capabilities the model 4881 facilitates development, investigation and troubleshooting of systems incorporating GPIB compatible processors such as the ABC80, Apple, Commodore PET, Hewlett Packard HP85 and Research Machines 380Z.

16 LEDs monitor the GPIB signals with individual switch control over each line. Using the Single Step Listener facility the model 4881 can display system activity allowing the user to check through each Bus transaction one at a time. Alternatively, as a Talker, the Analyser can output switch selected data bytes to a Listener either in a continuous or a Single Step mode.

As a Controller model 4881 can send Bus Commands and control the Bus management signals ATN, EOI, SRQ, REN and IFC. This permits complete tests to be made on Bus Systems at a simple level prior to attempting to run complex Bus programs.



Model 4881 is fully portable, self contained and is powered from any 240 V. 50 Hz supply. A top-of-panel GPIB connector facilitates easy CRO monitoring of Bus signals and a simple adapter affords connection to the IEC 625.1 Instrumentation Bus.

Sales Department,
WASEC,
45 Hurstcourt Road,
Sutton,
Surrey SM1 3JF,
Telephone: 01 669 2423

(1925 M)

Low cost elapsed time indicators

Frepar Electronics have been appointed U.K. distributors for a range of elapsed time indicators, manufactured by one of the foremost Swiss Instrument companies.

Designated the CH-6, their small size and low cost make them ideal for use in applications where it is necessary to know the total running hours of equipment (Bailers, Machinery etc), and will also provide accurate measurement of intervals between servicing.

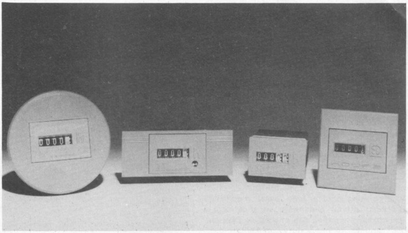
Voltage ranges are 6-440 V a.c., and 6-487 V d.c., with a choice of mounting plates if required. A spring retaining clip eliminates the need for fixing screws.

Military style and plug-in versions are available, the CH-6 is suitable for 35 mm standard DIN rail fixing.

Available from stock at Witham; cost, one off £ 8.55 attractive discounts for quantity.

Roger Hasler,
Frepar Electronics Ltd,
119, Newland Street,
Witham,
Essex CM8 1BE

(1932 M)



New servo amplifier IC

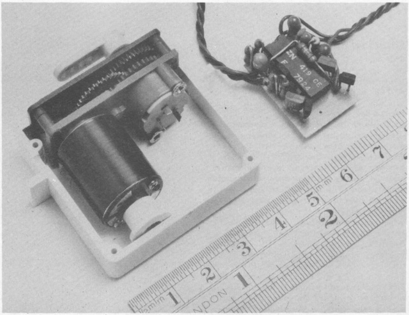
To meet the strong market demand for a low cost servo-amplifier suitable for use in general industrial applications, Ferranti Electronics Limited has developed and is now offering the ZN409CE, a servo-amplifier device in a 19.3 mm (standard length) dual in-line package.

This use of standard packaging has enabled the Company to market the new product at a pricing level which is some twenty five per cent below that of the popular and well established ZN419CE servo-amplifier, which in its shorter length package (17.78 mm) was specifically designed for use in areas where available space was limited.

Both the ZN409CE and ZN419CE have the same specifications, either being ideal for inclusion in a variety of pulsewidth position control applications ranging from model control to industrial equipment control. The two devices are also well suited for use in motor speed control circuits.

Ferranti Electronics Ltd.,
Fields New Road,
Chadderton, Oldham,
Lancs OL9 8NP.
Telephone: 061-624 0515

(1924 M)



market

Colourful cabinets

A new lightweight series of 19" instrument cabinets, called the E2000 range, is being launched by Optima Enclosures Ltd.

Built almost entirely from aluminium, the E2000 is available in 2, 3, 4 and 5U heights and 430 and 522 mm panel-to-panel depths, incorporating square hole panel mounting brackets. Interior dimensions are thus identical to corresponding size of existing Optima 19" cabinets for heavy duty applications, but the E2000 offers considerable savings in weight and price to make it particularly interesting for mass produced instruments, modems, etc.

One added attraction of the E2000 is that apart from its bold modem styling, it can be supplied in any single or two-colour variation from the Optima range of 12 standard colours.

Duncan Smith, Marketing Manager,
Optima Enclosures Ltd.,
Macmerry, Tranent,
East Lothian EH33 1EX.
Telephone: 0875 610747

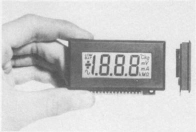
(1923 M)

Ultra low power DPM

This new LCD meter is claimed to be the first of a new generation of DPMs, giving at least ten times the battery life of any existing type. A PP3 battery will power the meter for typically two years, if operated for eight hours a day, seven days a week.

LCD watch manufacturing techniques are used to reduce the depth to a minimum, the meter being fitted into a DIN bezel of 72 mm x 36 mm. The 0.6" (15 mm) digits can be read at distances of up to ten metres, and the display contains many other useful annunciators. The totally flush face makes it easy to screen print any other logo or engineering unit onto the display.

Other standard features include a Digital Hold facility, Auto-zero, Auto-polarity, External bandgap reference for maximum stability, single-rail supply of 5 - 15 V d.c. drawing 200 μA, programmable decimal points and a 200 mV full scale deflection. The DPM can be used in single ended, differential or ratiometric modes, and may also be used in applications such as temperature measure-



ment or weighting systems, where a variable offset or tare is required. Display backlighting is a customer option. The meter is supplied with brackets for front or rear panel mounting.

Lascar Electronics Limited,
Unit 1, Thomasin Road,
Burnt Mills, Basildon,
Essex SS13 1LH,
Telephone: Basildon (0268) 727383

(1934 M)

A Supermum for Nascom 1

A low cost multi-purpose motherboard that turns the basic Nascom-1 into a sophisticated and extensively supported microcomputer system, is available from Gemini Microcomputers of Amersham, Bucks.

Appropriately named 'Supermum' the composite board contains a five-slot motherboard, a 5 A power supply and a buffer board which interfaces the Nascom-1 to the five-card bus. The buffer section also includes a reset jump facility — a feature not normally found on expanded Nascom-1 systems.

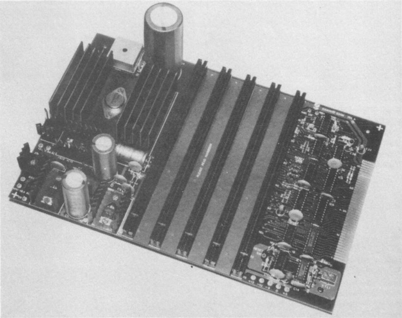
The power supply, fed from a separate transformer, is more than sufficient for the Nascom-1, the Supermum itself and up to five Nasbus expansion boards. It provides:
+5 V at 5 A
+12 V at 1 A
−5 V at 1 A
−12 V at 100 mA

As a 12" x 8" piggy back board, Supermum has fixing holes in line with those on a Nascom-1. By using spacers the board can be mounted directly above the Nascom.

Supermum is supplied in kit form complete with edge connectors for £ 85 plus VAT. (post and packing £ 2.50 plus VAT extra).

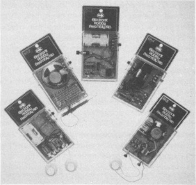
Gemini Microcomputers Ltd,
Oakfield Corner,
Sycamore Road,
Amersham, Bucks.
Telephone: 02403 22307.

(1926 M)



Hobby kits

OK Machine & Tool (UK) Ltd's new Hobby Products Division is now offering a range of silicon chip based hobby kits, on a mail order or credit card 'phone-in basis for between £ 3.99 and £ 8.60. Anyone aged 12 and over



can assemble one of these kits by following the very comprehensive instructions. The five initial kits are quick reaction, electronic dice, digital roulette, morse code and electronic organ, all of which can be assembled and repacked in their original plastic packs.

OK Machine & Tool (UK) Ltd,
Dutton Lane,
Eastleigh,
Hants SO5 4AA,
Telephone: 0703 610944

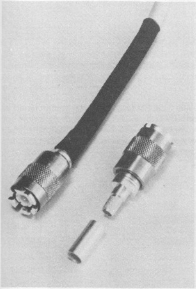
(1928 M)

market

Collet lock BNC connector

Greenpar Connectors Limited have introduced a completely new coaxial connector to mate with standard BNC sockets which provides a significant improvement in screening over the conventional BNC plug.

Known as the Collet Lock BNC series, the new connector has been developed by Greenpar in response to a suggestion by the Telecommunications Division of The Post Office which required a connector of superior performance when testing multi-coaxial installations for cross talk. Applications which could benefit from the new connector include any installation requiring a consistent and high level of screening performance, and include both test laboratory and field service applications. In both areas its compatibility with standard BNC sockets enables system performances to be improved without changing fixed connectors.



The attainment of high screening levels has been achieved through the use of an external collet which locks over the mating socket body (hence the designation Collet Lock BNC). This collet replaces the normal BNC bayonet coupling ring. Tightening the coupling sleeve locks the collet, and achieves a high force, low resistance contact which functions independently of the normal BNC screen contact.

Variants of the design are available both for normal double-screened coaxial cables and also for superscreened cables of the type developed by the United Kingdom Atomic Energy Autority.

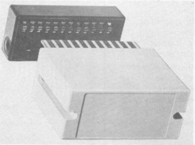
Pat Jones,
Greenpar Connectors Limited,
P.O. Box 15,
Station Works,
Harlow,
Essex CM20 2ER,
Telephone: 0279 27192

(1927 M)

Electronic module boxes

Frepar Electronics of Witham have been appointed sole U.K. agents and distributors for a range of module boxes manufactured in Germany.

These boxes (size 12 x 52, height 106 mm) have built-in edge connectors allowing up to 12 connection terminals.



They are simple to use — the base unit is screwed onto equipment or walls with the wires inserted into the springs. The front panel with all the electronic metres switches etc is then simply plugged into the base and held by two screws ideal for all those tight corners.

Very inexpensive — the case (complete assembly) with 12 terminals — gold plated contact springs — is only £ 8.50. The cover is moulded from impact resistant polystyrol or, at slightly extra cost, moulded from ABS, which gives a much higher temperature rating. The boxes can be supplied complete with printed circuit board ready to screen and is ideal for prototype use. Labels for the front panel can be supplied manufactured from either PVC or Paper.

Frepar Electronics Ltd,
119, Newland Road,
Witham Essex CM8 1BE

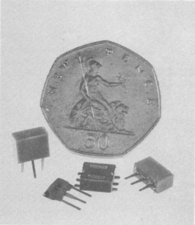
(1929 M)

Micro-sized mechanical filters

The present trends of miniaturisation in broadcast radio receiver equipment have led TOKO to develop the world's smallest series of IF filters for broadcast and communications receivers.

Clockwise from the right, the photograph shows the CFL ceramic AM IF filter — available for the range 450-470 kHz with a nominal 8 kHz bandwidth at −6 dB.

Next is the CMFC — designed for an ultra low profile in the increasing use of thick film techniques and offering an electrical performance equivalent to the CFMS (far left of picture). Both the CFMC and the CFMS offer selectivity equivalent to three conventional single tuned IF transformers in the frequency range 450-530 kHz. (Including the proposed 526.5 kHz for the CARFAX traffic information system).



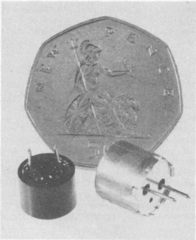
The remaining filter is the CFSKM FM IF filter, which is a derivative of the standard size CFSH series of ultra low temperature coefficient filters for use in wideband FM IF systems. The available bandwidths are: 280 kHz (CFSKM1) and 230 kHz (CFSMK2). Spurious responses are kept below −40 dB in the range 8-12 MHz, enabling the CFSKM to be used as a roofing filter in dual conversion receivers.

Ambit International,
200 North Service Road,
Brentwood,
Essex, CM14 4SG.
Telephone: 0277 230909

(1921 M)

Miniature high value inductors

The 10RBM is a high 'Q', high stability LF inductor with ferrite bobbin and shroud core for applications from 10 kHz to 200 kHz. Despite its very compact dimensions, the 10RBM spans the range 560 uH to 15 mH in E12 series with Q values as high as 100.



The 10RSB is similar to the 10RBM but with a shield case to prevent interaction from stray electrostatic fields. The same basic electrical parameters apply to the 10RBS as for the 10RBM, except that 'Q' is slightly degraded.

Ambit International,
200 North Service Road,
Brentwood,
Essex, CM14 4SG.
Telephone: 0277 230909

(1930 M)

market

Snap-in Fibre Optic Link

Available for under £ 35, the new HFBR-0500 Snap-in Fibre Optic Link is a TTL compatible fibre optic link. Hewlett-Packard supplies, in kit form and as discrete units, all of the elements of the link including transmitter, LSTTL/TTL compatible receiver, one millimetre core diameter plastic fibre in bulk or terminated lengths, connectors that are quick and easy to attach, and a polishing kit. The connectors snap in to the dual-in-line transmitter and receiver modules which are colour coded for easy field installation and repair.

The HFBR-0500 series Fibre Optic Link can be used for low cost, short length inter- or intra-system data links to solve commode or high voltage isolation problems. Because the receiver has an internal shield, it is resistant to Electromagnetic Interference and may be used in high EMI applications. HP's Fibre Optic Link may also be used to meet Electromagnetic Compatibility (EMC) specifications of the FCC and VDE and safety requirements of the VDE.

The grey plastic HFBR-1500/1 transmitter incorporates a 665 nm LED and can be easily interfaced to logic families with an open collector TTL buffer gate. The blue plastic HFBR-2500 Receiver incorporates an integrated photo detector a shielded wide bandwidth d.c. amplifier, and open collector output circuit. It is compatible with most +5 volt logic families and has a d.c. to 10 MBaud data rate.

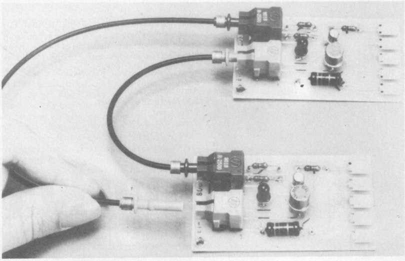
The HFBR-3500, 1 mm core, fibre optic cable is terminated in colour-coded snap-in plastic connectors. Terminated lengths of cable are available in 0.1 metre increments and may be purchased separately from the transmitter and receiver modules. Bulk lengths of unconnected cable, HFBR-3590/1, can easily be terminated with the HFBR-4510 (grey) and HFBR-4511 (blue) connectors, then finished with the HFBR-4595 Polishing Kit. The plastic connectors are designed for quick installation with a minimum of tools and no adhesives.

The cable's 1 mm core diameter matches the active areas of the transmitter and receiver for maximum light coupling. Access is available to both the anode and cathode pins of the transmitter LED to allow the system designer to optimise system layout and performance by implementing the proper drive configuration. Detailed information is included in the kit in Application Note 1009, 'Designing with the HFBR-0500 series Snap-in Fibre Optic Link'.

In quantities of one to 99, the HFBR-0500 Snap-in Fibre Optic Link kit is £ 32.67 and it is stocked at Hewlett-Packard authorised distributors.

Hewlett-Packard Ltd,
King Street Lane,
Winnersh,
Wokingham,
Berks RG11 5AR,
Telephone: (0734) 784774

(1869 M)



Bench-top frequency standard

A new bench-top frequency standard from CSC, the Model 4401, provides a source of discrete, selectable precision frequencies for use as either a time or frequency standard or a highly accurate signal source. A 10 MHz precision crystal over oscillator gives an accuracy within ± 0.5 parts in 10^6 from 0°C to 40°C, and the provision of only two controls — a frequency-select pushbutton and a frequency-multiplier switch — makes the instrument very easy to use.

The CSC Model 4401 has two d.c.-coupled outputs available via front-panel BNC connectors. One output always provides a 10 MHz square-wave signal, while the other 'select' output provides any one of 24 discrete, selectable outputs from 0.1 Hz to 5 MHz. The frequency-select pushbutton covers a range of 0.1 Hz to 10 MHz in nine decade steps, while the multiplier control gives a selection of 1 X, 2 X or 5 X multiplication factors.

Both outputs have a 50 Ω impedance, are compatible with TTL logic, and are short-circuit protected. Square-wave rise and fall times are 20 ns. Eight front-panel light-emitting diodes indicate the selected frequency decade, and an additional 'oven-ready' indicator is provided.

The 10 MHz crystal oscillator is oven-controlled at 55°C, and is factory-calibrated to the National Bureau of Standards. Ageing is less than one part in 10^6 per year. The crystal oven normally takes between three and five minutes before it reaches operating temperature and the unit locks on to the correct frequency.

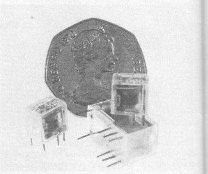
The 4401 is ideally suited to applications such as the calibration of oscilloscopes, timers and frequency counters, as a precision clock source for microprocessors, or as a precision time reference in the laboratory, field-service, educational or industrial environments. It measures 76 x 254 x 178 mm, and weighs 0.9 kg.

Continental Specialties Corporation,
Shire Hill Industrial Estate,
Saffron Walden,
Essex,
CB11 3AQ,
Telephone: (0799) 21682

(1874 M)

Miniature low cost relays

The RBU series relay is a flux resistant PC mounting relay with two pole changeover contacts, each capable of handling 2 A at 24 VDC. Stock types are provided with 10 - 12 VDC 320 ohm coils, although any value from 3 V to 24 V can be accomodated to order. Life expectancy is a minimum of 10 million mechanical cycles, with 100,000 cycles for the contacts when run at maximum capacity.



The RCU relay is one of the smallest changeover relays available (excluding TO5 types). It is a single pole unit, capable of switching 2 A at 24 VDC (or 2 A at 100 VAC) The 'stock' coil is a 10 - 12 VDC 320 ohm winding although 3 - 24 V is available to order. Life expectancy is the same as for the RBU relay.

Applications of both these relays include antenna switchover, vending machinery, remote control systems etc. The RBU costs 99 p in 100 off, and the RCU costs 84 p in 100 off. One off pricing is £ 1.85 and £ 1.65 respectively.

Ambit International,
200 North Service Road,
Brentwood,
Essex CM14 4 SG,
Telephone: (0277) 230909.

(1873 M)