

Microcomputer News & Products

paper-tape form, as an alternative to buying a set of PROMs. A tape costs \$15, and may be ordered direct from NS Electronics, Cnr Stud Road and Mountain Highway, Bayswater, Victoria 3153.

Faster SC/MP chip, too

National Semiconductor has also announced a new N-channel version of the SC/MP chip itself. It offers three main features over the existing P-channel chip: Twice the speed, one quarter the power, and only a single +5V supply.

Designated SC/MP-II, the new chip will be available in Australia shortly, from NS Electronics and their various distributors in each state.

SC/MP Tiny BASIC

Back in the December 1976 issue, we announced that National Semiconductor was coming out with a Tiny-BASIC interpreter for their SC/MP, called NIBL. At that stage only a 3k bytes preliminary version was available, with an improved 4k version still to come.

Well, the 4k version of NIBL has now arrived, and it's even better than was predicted. It is now very much an extended Tiny-BASIC, with many powerful features which should make it of great interest and value to professional and hobby computer users alike.

As predicted, it now offers an RND function to generate 16-bit random numbers, and a LINK statement to allow calling machine language subroutines. The hoped-for DO . . . UNTIL statements are also provided, too.

In addition, there is now the ability to perform FOR . . . NEXT loops as in many full BASICs. There is also the ability to handle character strings, and to handle hexadecimal constants. There is also a REM statement for remarks, a MOD function for absolute values, a STAT function to return the current value of SC/MP's status register (allowing the program to manipulate flag and sense lines), and paging functions.

NIBL's formal grammar is now somewhat more flexible, too, allowing greater programming efficiency. Multiple statements per line are now allowed, while LET is no longer mandatory in assignment statements.

And you can now buy NIBL in punched

New MC6800 kit

Motorola Semiconductor Products has released a new evaluation kit for their well-known MC6800 microprocessor. Called the MEK6800D2, the new kit has its own 24-key keyboard and 6-digit hex display, obviating the need for an expensive data terminal. It also features an inbuilt audio cassette interface, for convenient program storage at low cost.

The kit comes with 256 bytes of RAM for user programs, a 1k byte ROM with a debug/monitor called JBUG, and 16 lines of parallel I/O via a PIA. Provision is made for easy expansion.

By the time you read this, the new kits should be available from local Motorola distributors for around \$240.

By courtesy of Motorola, EA has one of the new kits and we are currently putting it through its paces. We hope to publish the results next month, all going well.



MOTOROLA M6800 EVALUATION KIT



MEK6800D2 provides a useful and expandable tool for those who wish to develop systems with the M6800 Microprocessor without investing in expensive terminals. All parts needed to complete the system and get up and running are provided in the kit with the exception of the power supply. In addition to the expansion available on the basic microcomputer module, additional RAM, ROM and I/O parts can be accommodated at a later date to implement more complex systems. Machine language programs can be entered through the system keyboard or via a built-in audio cassette interface system. Hexadecimal LED displays are provided for monitoring data and address information. A crystal-controlled clock generator is used to eliminate timing adjustments.

- JBUG Monitor. Trace One Instruction. Set up to Five Breakpoints. Examine and Change Memory and Registers.
- Parallel and Serial Interface Capability.
- 16 I/O Lines, 4 Control Lines.

FIRMWARE FEATURES

The monitor firmware included with this system sets a new standard of performance for evaluation kits. The function of the JBUG monitor is to allow the user to communicate with and hence control the M6800 microcomputer by using the hexadecimal keyboard and display module. The intelligence and diagnostic capability of JBUG is provided in line 1K x 8 ROM.

MEK6800 D2

HARDWARE FEATURES

Three I/O devices are provided in the kit. One Peripheral Interface Adapter (PIA) is dedicated to the hexadecimal keyboard and display module. A second PIA is made available exclusively for the user. An Asynchronous Communications Interface Adapter (ACIA) is also included to interface with your audio cassette tape recorder.

TOTAL ELECTRONICS

MELBOURNE
SYDNEY
BRISBANE
ADELAIDE
PERTH (ATHOL HILL P/L)
NEWCASTLE (DIGITRONICS)
CANBERRA (ELECTRONIC COMP)

96 2891;
439 6722;
229 1633;
268 2922;
25 7811;
69 2040;
95 6811;

WANT TO HEAR MORE?
CALL MOTOROLA SYDNEY
-438 1955, 43 4299
MELBOURNE-561 3555
MOTOROLA DISTRIBUTORS

CEMA

MELBOURNE
SYDNEY
BRISBANE (ELECTRONIC COMP)
ADELAIDE (WOOLLARD & CRABBE)
CANBERRA (CUSTOM SCIENTIFIC)
WOLLONGONG (MACELEC)
PERTH (RESERVE ELECTRONICS)

877 5311;
439 4655;
371 5677;
51 4713;
47 6179;
29 1276;
87 1026;



MOTOROLA Microsystems

—making it happen in microcomputers

The Motorola 6800

One of the most established microprocessors, the Motorola MC6800 is supported by a continuously expanding "family" of memory and specialised interfacing chips—and also by a great deal of proven software and applications experience. In this article we look at the MC6800, its main support chips, and also the recently released MEK6800D2 "Mark II" evaluation kit.

by JAMIESON ROWE

Motorola Semiconductor Products were only the second major US manufacturer to enter the microprocessor field, in 1974. The MC6800 was their initial entry, and the fact that it is still one of the market leaders—and showing no signs of giving ground—testifies to the soundness of the basic design concept. Needless to say it is now supported by a large amount of development and applications software, much of it generated by users. This inevitably increases its appeal for potential new users, compared with newer entries to the field.

It is strongly supported in another sense, too: along with the basic microprocessor chip there are a number of matching memory and interfacing chips, all designed to simplify system design. These include a 1024 x 8-bit ROM (MCM6830), a 128 x 8-bit RAM (MCM6810), a programmable 16-bit bidirectional Peripheral Interface Adap-

tor or "PIA" (MC6820), a programmable Asynchronous Communications Interface Adaptor or "ACIA" (MC6850), a programmable Synchronous Serial Data Adaptor or "SSDA" (MC6852), and a number of more specialised devices including some which are still in development. There are also various system housekeeping devices, including a family of hybrid clock oscillators.

As the block diagram suggests, the basic architecture of the MC6800 microprocessor chip itself is fairly straightforward. This is perhaps part of the secret of its success, although there are a number of subtle strengths which also emerge upon closer inspection.

There are only six internal working registers, one of which is a condition code or processor status register. Two of the remaining registers are 8-bit accumulators, both full primary accumulators of almost equal status. The other three registers are the program counter,

an index register and a stack pointer, all three of which are 16 bits long. The MC6800 implements its stack in external RAM.

The status register has six active bits, one of which is the flag for enabling and disabling the master interrupt input. The remaining status bits are condition code flags, representing arithmetic carry and overflow, sign, zero and a bit-3 carry used for BCD arithmetic. None of the status register bits is accessible directly via external device pins.

Data and address information pass between the MC6800 and the rest of a system via two separate buses. One is an 8-bit bidirectional data bus; the other is a 16-bit address bus which gives the MC6800 the capability of directly addressing 65,536 or "65k" bytes of memory space.

The output buffers on both the data and address lines are 3-state, and may be disabled for DMA operation.

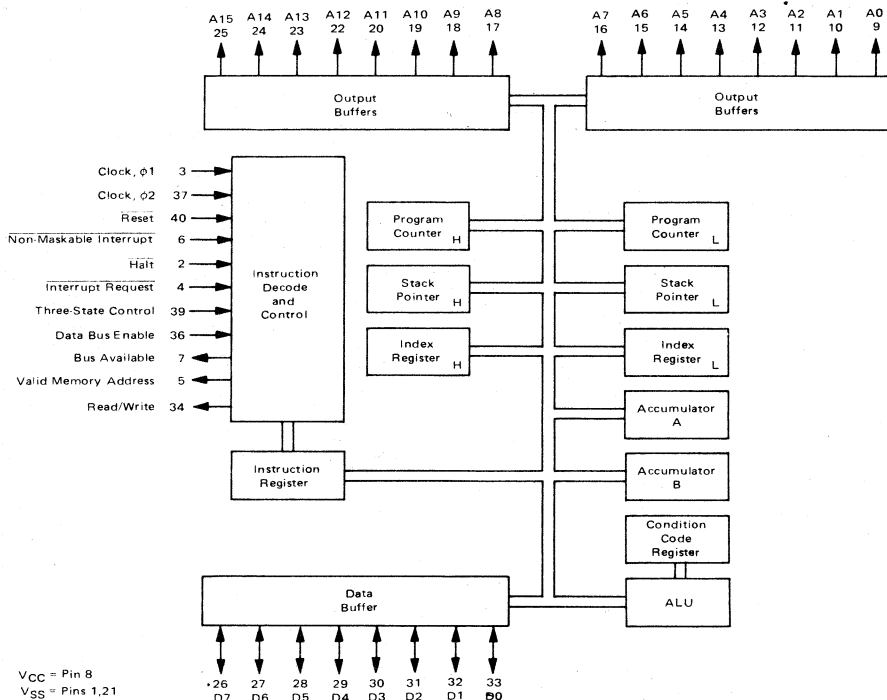
There is no clock oscillator on the MC6800 chip itself. Instead there are two clock input pins, which must be fed with non-overlapping two-phase clock signals. Maximum clock frequency is 1MHz, and a machine cycle corresponds directly to a clock period. Fetching and execution of an instruction ranges from 2 to 12 machine cycles depending upon the instruction, or from 2 to 12µs at the maximum clock rate.

Incidentally the MC6800 is an N-channel depletion mode MOS device, made using silicon gate technology. As a result it operates from a single +5V supply.

Apart from the normal reset and master interrupt inputs the MC6800 is also provided with a non-maskable interrupt input. There is also a software-interrupt instruction, for program abortion. Vectoring is provided only for the different interrupt mechanisms; within each mechanism an interrupt source must be identified by polling.

What the MC6800 does do upon being interrupted is automatically save the contents of all its registers on the stack, in external RAM. This takes place in only 10 machine cycles, far less than would be required if the interrupt routine had to perform the saving by explicit instructions (required by most other micros).

Left: Basic architecture of the MC6800 microprocessor chip. Six registers are accessible to the programmer.



The instruction set of the MC6800 comprises some 72 different instructions, ranging from one to three bytes in length. There are 29 instructions involving the accumulators and/or memory; 11 involving the pointer register and the stack pointer; 23 are jump, branch and special operation instructions; and the remainder are for status register manipulation.

The relatively large number of memory reference instructions helps to compensate for the relatively small number of working registers within the CPU chip itself. An interesting aspect of this is that many of the arithmetic and logic operations can be performed not only on the contents of the accumulators, as in other microprocessors, but on the contents of memory and interfacing chip registers as well. Thus the MC6800 lets you clear, form either the 1's or 2's complement, decrement, increment, rotate left or right, shift left or right, and test the contents of memory address and interface chip registers—as well as those of the accumulators—all with single instructions.

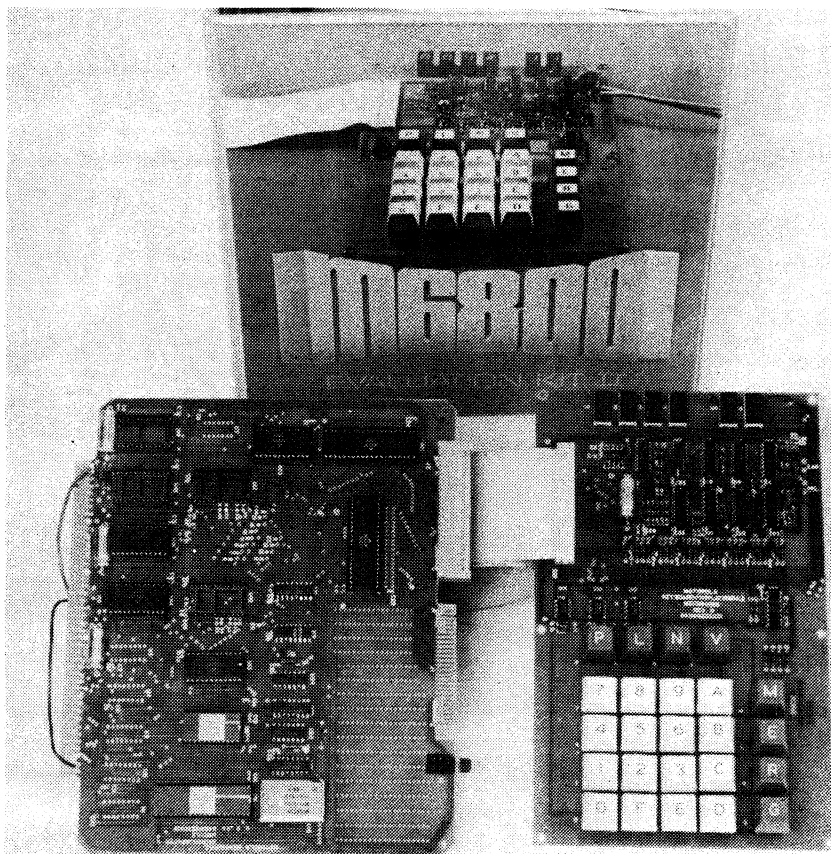
Apart from the inherent addressing used for instructions involving only accumulators and/or other internal registers, the MC6800 provides five different memory addressing modes. These are immediate addressing, direct or 1-byte absolute addressing, extended or 2-byte absolute addressing, relative addressing and indexed addressing.

Not all of these addressing modes are available for all instructions, however. In fact relative addressing is available only on branch instructions, and these instructions have no other mode available. Similarly the Jump and Jump-to-Subroutine instructions and many of the shift and rotate instructions have only indexed and extended addressing options.

With immediate addressing, the second byte of the instruction itself is interpreted as the operand data. With direct addressing the second byte is interpreted as an unsigned 8-bit absolute address, allowing addressing of the first 256 locations in memory space (00-FF hex inclusive). Extended addressing is similar to the latter except that the instruction has three bytes, and the second and third bytes are interpreted as an unsigned 16-bit absolute address, allowing addressing of any location in the 65k memory space.

In the relative addressing mode the processor interprets the second instruction byte as a signed 8-bit number, which is added to the current program counter contents to give the effective address. This allows addressing in the range from -128 to +127 bytes away from the location immediately after the second instruction byte.

In the indexed addressing mode the contents of the index register are used



Here is the assembled MEK6800D2 evaluation kit. The two PCBs are shown in front of the binder containing hardware data and programming manuals.

in generating the effective address, as with other microprocessors. However, in this case the second byte of the instruction is interpreted as an unsigned 8-bit number to be added to the index register contents, not a signed number as with the relative addressing mode. This means that the indexed address range only extends forward from the location specified by the index register, not backward; however, there is still a full 256-address range.

The large number of branch instructions provided by the MC6800 allow for considerable programming economy, particularly when data is being manipulated. Conditional branching conditions include carry clear, carry set, zero, greater or equal to zero, greater than zero, higher negative or zero, negative greater than zero, lower or same, minus, not equal to zero, overflow clear, overflow set, and plus.

Incidentally the MC6800 has no separate IOT instructions; all I/O devices are accessed as locations within the 65k memory space. As the MC6800 provides some fancy memory reference instructions, this can simplify programming for complex data communication applications.

Returning to the hardware side, one of the essential devices in any system using

the MC6800 is a clock oscillator. As the microprocessor requires fairly critical non-overlapping two phase clock signals, Motorola provide a family of hybrid crystal clock modules in modified 24-pin DIL compatible packages. These are the MC6870 series, some of which provide just the basic two phase signals for the CPU together with a TTL signal for memory timing, while others provide a number of other signals as well.

There are quite a few memory devices provided by Motorola to support the MC6800, including both static and dynamic RAMs, mask-programmed ROMs, and an EPROM. Most of the devices are byte-organised and provided with multiple chip-select inputs to simplify system design, while both the MCM6810 128-byte RAM and the MCM6830 ROM operate from a single 5V supply like the MC6800 itself.

Of the specialised peripheral interfacing devices in the 6800 family, the MC6820 PIA is used for parallel interfacing. It provides 16 I/O pins, grouped in two sets of 8 although all pins may be individually programmed as either inputs or outputs. Associated with each set of 8 I/O pins within the PIA are three separate 8-bit registers, making six in all.

(continued overleaf)

One in each group is a data buffer, another a latch whose bits specify whether the device pins are used as inputs or outputs; the third is a control register used to define interfacing protocol and status. All six PIA registers are addressable in MC6800 memory space, although in a slightly confusing manner: the control registers are addressed directly, while the data and direction registers share common addresses and must be distinguished by setting a control register bit.

The other peripheral interfacing device most likely to be found in smaller 6800 systems is the MC6850 ACIA, used for asynchronous serial interfacing. The ACIA is rather like a UART, having separate sections for asynchronous transmission and reception. However unlike a UART these share a common 8-bit parallel bidirectional interface to the 6800 system data bus. In addition, there is an 8-bit control register addressable separately in 6800 memory space, which allows program control of serial data format, a choice of three communication rate clock division ratios, and the handshaking protocol. There is also a status register, sharing the same address as the control register, whose bits may be read to determine ACIA status.

Having looked briefly at the MC6800 microprocessor, its instruction set and some of its support chips, let us now turn to the new MEK6800D2 evaluation kit. This has been produced by Motorola to provide a complete low-cost 6800 system, for both evaluation and basic program development.

The MEK6800D2 is an assemble-it yourself kit, which goes together to produce two PCB assemblies. One is the microcomputer itself, on a PCB measuring 248 x 210mm overall; the other is low-cost terminal unit on a PCB measuring 254 x 159mm.

The assembled microcomputer board has the MC6800 itself, a crystal clock module (614.4kHz), a 1k byte ROM with resident "JBUG" monitor program, three 128-byte RAMs (one of which is allocated to the monitor, leaving 256 bytes for user programs), two PIA devices and an ACIA.

The PCB also has decoding and sockets for easy expansion using a further two 128-byte RAMs, and two 1k byte EPROMs (MCM68708). It also has space for data bus and address bus buffers, if the user wishes to expand further into a multi-board system.

The assembled "terminal" PCB has a 24-key keyboard and a display using six 7-segment LEDs. Together these can be used with the JBUG monitor for entering programs, examining memory and registers, single stepping through a program, setting and removing breakpoints (five

are permitted), and transferring control to the user program.

In addition, the terminal PCB contains a full audio tape interface, to allow dumping and loading of programs using a normal cassette or reel-to-reel tape recorder. All that is required apart from the kit (and power supply) are a couple of shielded leads with suitable audio connectors.

The loading and dumping operations are controlled by further JBUG routines. Transfer takes place at a rate of 300 bauds, and the format conforms to the "Kansas City Standard" with 2400/1200Hz tones. The interfacing circuitry requires no "tweaking", using a stable counter-type decoder.

The terminal PCB connects to the main microcomputer PCB by means of a 50-way flat ribbon cable and edge connector. The keyboard and LED display interface via one of the two PIAs, while the tape interface uses the ACIA. The remaining PIA on the main PCB is available for user interfacing, with the terminal connected. If the user later decides to use the kit for a dedicated application, without the terminal PCB, all three interfacing chips can be used for interfacing.

It is also possible to convert the kit over for operation with a teleprinter or other serial asynchronous terminal. The main change required is to replace the JBUG ROM with another containing the terminal-orientated monitor "MINIbug III".

The complete MEK6800D2 kit operates from a single +5V power supply, drawing about 1 amp.

The Sydney office of Motorola Semiconductor sent us one of the MEK6800D2 kits, so that we would be able to assemble it and report to readers on our findings.

The kit comes in a single box, which opens to reveal one of the large spring-clip binders. Inside are two blister packs containing most of the parts for the two PCB modules, together with a plastic bag containing the rest of the parts. There are also a number of handbooks, including a kit manual, programming reference manual and M6800 system design book.

I found it fairly easy to put the kit together, although the instructions are rather cursory and assume that the builder has a fair amount of experience. Assembly took me about 7 hours, but I wasn't trying to break any records.

The kit worked perfectly when power was applied, and I was then able to run through the introductory program load-run-debug example which Motorola have thoughtfully provided in the kit manual. This is well done, and should give a newcomer to microcomputer systems a good idea of the basic concepts of program manipulation via a monitor.

When it came to writing our own

SIMPLE DISPLAY PROGRAM FOR MOTOROLA MEK6800D2 KIT WRITTEN BY J. ROWE, ELECTRONICS AUSTRALIA 9.3.1977

0000 CE 00 24	START, LDX DISBUF	SET X AS BUFF PTR
0003 DF 22	STX XBUF	& SAVE
0005 86 20	LDA A \$20	SET PIA FOR DISPLAY UI
0007 B7 80 22	STA A DISREG	
000A A6 00	LOOP, LDA A 0, X	FETCH CHAR VIA X
000C B7 80 20	STA A SEGREG	& DISPLAY
000F CE 00 4D	LDX \$4D	SET UP X FOR 1MS DELAY
0012 BD E0 E0	JSR DLY1	CALL JBUG DELAY S-R
0015 7C 00 23	INC XBUF+1	INCREMENT SAVED BUFF PTR
0018 DE 22	LDX XBUF	& RESTORE TO X
001A 0C	CLC	CLEAR CARRY
001B 74 80 22	LSR DIGREG	UPDATE DIGIT PTR IN PIA
001E 24 EA	BCC LOOP	CONTINUE UNTIL 6 DONE
0020 20 DE	BRA START	BACK TO BEGIN AGAIN
0022 00 00	XBUF,	X IS STORED HERE
0024 02		START OF MESSAGE BUFFER
0025 00		
0026 40		
0027 40		
0028 21		
0029 24		

A simple novelty program for the MEK6800D2 kit. It displays encoded characters stored in locations 0024-0029 on the kit's 6-digit LED display.

programs, the going wasn't quite as easy. The programming reference manual doesn't seem to me to be particularly well written, at least as far as the introductory material is concerned. For example the material describing the various 6800 addressing modes complicates the issue by talking quite a lot about assembly language syntax, so that a beginner could get very confused.

As I was not too familiar with the 6800, it took a while to sort out chip operation from assembler operation. One thing which helped was a look through the listing for the kit's J-BUG monitor program, which Motorola have thoughtfully given in the manual.

Once the addressing modes were sorted out, I was able to begin writing a few short programs and try them out. A sample program is reproduced here, as readers may find it interesting. It was written as a little exercise to see how one can use the LED display under user program control.

The kit manual doesn't help a great deal in telling you how to display data, so I had to deduce the way of doing this from the terminal circuit diagram and the J-BUG listing. It turns out that the routines in J-BUG itself are not capable of being called by user programs, as they are not self-contained subroutines. However, in any case it is fairly easy to provide a routine in one's own program, as you can see. This is largely because the PIA does most of the work.

The comments in the right-hand column of the listing should give you a fair idea how the program works. Note that the display digit multiplexing is per-

formed by loading a 1 into one data register of the PIA, labelled "DIGREG", and then shifting the 1 along using the LSR instruction (address 001B). Similarly the actual digits are fed to the display segments by loading them into the other PIA data register, labelled "SEGREG".

The only part of the J-BUG monitor made use of by this little program is the subroutine DLY1, used to obtain the 1ms delay between displayed digits. This is used by first loading the index register with hex 4D, to specify a 1ms delay time, and then calling the subroutine at address E0E0. The two instructions involved are those with their first bytes stored in addresses 000F and 0012.

Note that the simple 6-character message displayed by this program is stored in locations 0024-0029. They are not in ASCII code, but in a code whose first 7 bits correspond to the seven display segments, in complement form. Any characters capable of being displayed on 7-segment LEDs can be shown, by working out the appropriate codes.

To summarise, the MEK6800D2 kit seems a well-designed one, and should enable those with reasonable experience at electronic kit building to build up a low-cost 6800 system. When assembled it becomes a small but businesslike development system, adequate for learning 6800 programming and working up quite respectable programs. And with plenty of provision for expansion, you can make it grow into a more elaborate system when this is needed.

In short, good value for money at the quoted price of around \$240. The kit should be available from Motorola distributors, in every state.

Using the Motorola D2 kit with a terminal

Computer hobbyists and schools who have the Motorola MEK6800D2 evaluation kit should find this article interesting. It explains how the kit may be converted easily and with minimum outlay to interface with a video terminal or teleprinter, using standard 20mA current loops. It also gives details of useful subroutines available in the terminal-orientated MINIBUG-III ROM.

by JAMIESON ROWE

Since we wrote about the Motorola MEK6800D2 "Mark 2" evaluation kit in the May 1977 issue, quite a few of the kits have apparently been sold. It would seem that many people agreed with our basic finding, that the kit provided an attractive way of building up a 6800-based system suitable for learning 6800 programming and working on program development.

Probably one of the features that appealed to people was the kit's expandability. For example although it comes with only 256 bytes of user-available RAM, the PCB provides decoding and sockets for the addition of a further 256 bytes merely by plugging in a further pair of MCM6810 RAM chips. There are also two further sockets, which may be arranged quite easily to accept either a pair of 1024-byte EPROMs (2708 or similar), or a pair of 2048-byte mask-programmed ROMs (MCM68316 or similar).

To allow for still further expansion the PCB is provided with spaces for the addition of data and address bus buffers, so that external memory boards and further peripheral interfaces may be connected. As the kit already offers two MC6820 PIA (peripheral interface adapter) devices, each providing two 8-bit bidirectional peripheral interface ports, and an MC6850 ACIA (asynchronous communications interface adapter) which provides a serial input and output, this makes it very suitable for expansion into a full-scale minicomputer or development system.

Motorola itself makes available a number of compatible memory boards, including two dynamic RAM boards offering either 4096 bytes or 16,384 bytes of memory (MM68100, MMS68104). However these are

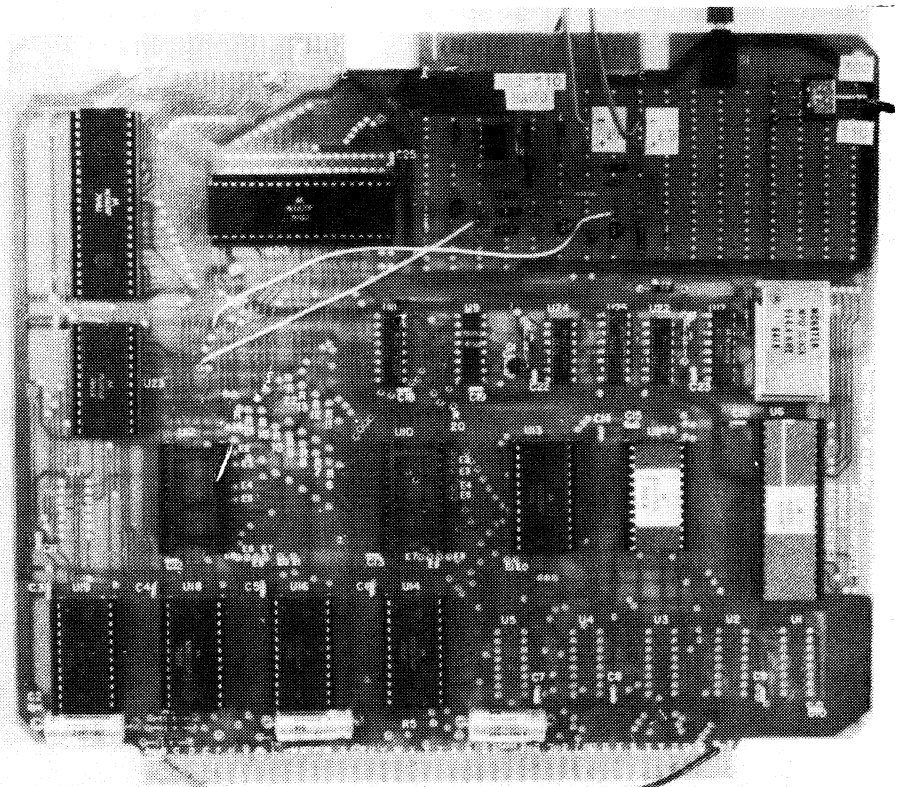
probably a little elaborate for most hobbyists, being more suitable for the professional user. A more attractive approach for the hobbyist would probably be to use one of the low cost memory modules now available using static RAM chips.

Apart from memory expansion, perhaps the most likely thing that both hobbyist and professional D2 kit users will want to do to their system is modify it for use with a video terminal or teleprinter. Although the original 24-key keyboard and LED-display "terminal" supplied with the kit is quite sufficient for one's initial foray into 6800 system operation and development of simple programs, it is not really suitable for serious work. Programs may be stored on cassette tape, but cannot

be entered efficiently in the first place or listed conveniently for examination. Word processing and games are not really feasible with the simple terminal, either.

Anticipating that many users would want to modify their D2 kit for use with a standard terminal, Motorola has made available a suitable terminal-orientated debug/monitor program. The debug/monitor is called "MINIBUG-III", and it comes in a mask-programmed ROM which is pin-compatible with the original JBUG monitor ROM supplied as part of the D2 kit.

The firm has also made available an application note (AN-771), giving details of the ways that the D2 kit may be expanded, and including informa-



Pictured at right is our D2 kit, modified for 110/300 baud terminal operation as described in this article.

Using the Motorola D2 kit with a terminal . . .

According to the Motorola manual, the ACIA is used with an audio FSK (frequency shift keying) interface on the matching "terminal" PCB, for dumping on and loading programs from cassette tapes. The data rate used is 300 baud, and the corresponding 4800Hz clock signal required by the ACIA transmitter (16 times 300) is derived from U17. This is an MC14040 multi-stage binary divider IC, driven in turn by a 1.2288MHz signal from the MC6871 crystal clock generator (U15).

As originally wired the 4800Hz signal connects to ACIA pin 4, the transmitter clock input (TXC). The receiver clock input RXC (pin 3) is fed with a recovered 4800Hz clock signal, derived by the FSK interface from the tape recording itself.

To use the ACIA with a standard video terminal or teleprinter, it is necessary to connect the TXC and RXC inputs of the ACIA together so that they operate at the same clock rate. This is most easily done by joining the two together with a wire link near the J2 edge connector. The two clock inputs run to connector pads 17 and 19, and they can be joined easily near the pads.

As the TXC input is still at this stage connected to U17 via the PCB pattern, the link between the TXC and RXC inputs of the ACIA will enable the system to communicate with a terminal at a fixed, crystal-locked 300 baud rate.

The MINIBUG-III monitor ROM can communicate at this rate, as can many video terminals and some teleprinters. However with MINIBUG-III it is actually necessary to use 110-baud communication at least upon initial switch-on, in order to tell the monitor to change to 300-baud operation. For this reason and to allow the system to work with 110-baud terminals, it is necessary to modify the clock feed circuit.

The photographs below should give you a good idea how we added the additional circuitry to the PCB of our D2 kit.

The first step is to cut the existing PCB track from U17 to pin 4 of the ACIA (U23), to allow a choice of baud rates. Do this by carefully cutting the PCB track on the underside of the board, at a point about 30mm from the ACIA. This is just beyond a small plated-through hole which carries the track from pin 4 to the top of the board and hence to the J2 connector. The cut therefore severs only the underside track leading to U17, and leaves the track to J2 intact.

Having done this the next step is to provide a source of ACIA clock pulses for 110-baud operation. As there is no suitable source of pulses in the existing D2 kit circuit, it is necessary to add a simple pulse generator using a 555 timer. This is shown in the circuit, and as you can see it involves very few components apart from the IC itself. The

22k trimpot allows the 555 output frequency to be set to 1760Hz, the correct frequency for the ACIA to operate at 110 baud.

The remaining thing is to add a small switch, to allow the ACIA to be fed with either the 4800Hz or 1760Hz, corresponding to 300 or 110 bauds respectively. The 4800Hz signal fed to the switch is taken from pin 13 of U17, as before.

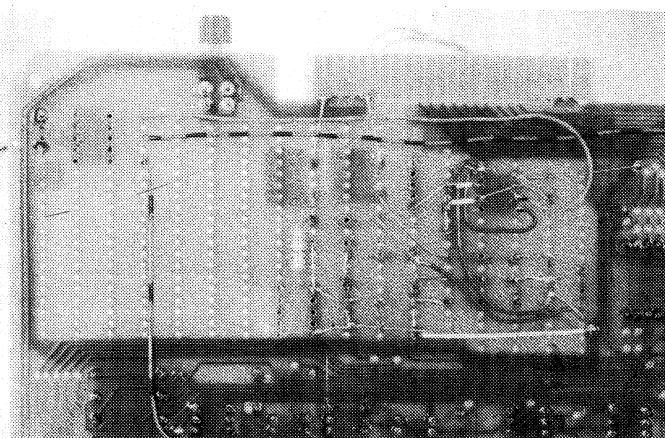
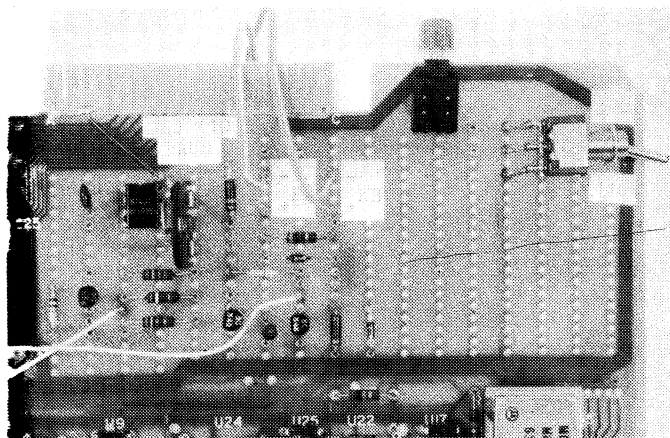
Incidentally, it should be noted that U17 actually provides a number of other crystal-derived signals, capable of being used to run the ACIA at other baud rates. Although the MINIBUG-III monitor program is only designed to communicate at 110 or 300 baud, you may well want to use higher rates once you have written a few programs of your own — assuming your terminal is capable of running at higher rates also.

TABLE 1 : MINIBUG-III FUNCTIONS

MONITOR FUNCTION	MINIBUG III COMMAND	NOTES
Display Internal Registers	R	1
Load RAM from Tape	L	
Dump RAM to Tape (Punch)	P	2
Memory Examine/Change	M	3
Go to Entered Address and Execute	G	4
Set Terminal Baud Rate	S	5
Trace One Instruction	N	
Set a Breakpoint	V	6
Reset a Breakpoint	U	
Continue Execute from Breakpoint	C	
Delete All Breakpoints	D	6
Print Addresses of All Breakpoints	B	
Trace N Instructions	T	

NOTES

1. Order of Display: (PC,SP,CC,B,A,X).
2. Before executing, load beginning and ending address of range in locations A002 to A005.
3. Enter M followed by address. Contents are displayed after typing last address character.
4. Type G, followed by address. Execution begins after type of last character.
5. For 110 Baud: Type S1. For 300 Baud: Type S3.
6. Type address where breakpoint is desired, followed by V. A total of 8 may be entered. Removal of all breakpoints executed by typing V not preceded by address.
7. IRQ vector must be stored at A000/A001, NMI must be stored at A006/A007.



Using the Motorola D2 kit with a terminal . . .

The full list of frequencies available from U17 is listed below, together with the corresponding baud rates:

Pin No.	Frequency	Baud Rate
14	1200	75
12	2400	150
13	4800	300
4	9600	600
2	19200	1200
3	38400	2400
5	76800	4800
6	153600	9600

You may care to fit a multi-position switch in place of the two-position switch shown in the circuit, to allow further baud rates to be selected. However note that the keyboard input filter circuitry connected to Q3 will have to be modified for very high baud rates.

As you can see from the photographs, I wired the additional circuitry in the area on the D2 kit PCB provided for additions. The wiring is not critical, and there is plenty of room. The 110/300 baud rate switch is mounted at the side of the board, near the reset button. It is held in place by the three connection wires, soldered to both the board holes and the switch lugs, and also by a small loop of tinned copper wire pulled tight around the threaded ferrule and soldered to the board.

The terminal connections are made available at four PCB pins, two for the display/printer and two for the keyboard. On the hardware side, that is all there is to it.

As far as software is concerned, the monitor and debug functions performed by MINIBUG-III are shown in Table 1 together with some brief notes on their operation. As you can see, MINIBUG-III allows programs to be loaded, examined, dumped and executed with breakpoints for debugging. This allows programs to be developed very quickly and conveniently.

Apart from using MINIBUG-III in this way, it contains a number of useful subroutines which may be called by your own programs. There are 13 such subroutines, according to the Motorola literature; all of them concerned with input and output of information via the terminal. Table 2 lists them for your information, together with their address in the ROM and details of their entry and exit requirements in terms of register contents, etc.

At the time of writing, no listing of MINIBUG-III is available from Motorola, so this information is all that is available. However if you are really interested in finding out more, you could try writing a disassembler program and turning it loose on the ROM, for a start.

TABLE 2 : Accessible subroutines in MINIBUG-III

(Hexadecimal starting address of each is shown in brackets)

BADDR (E0F8) — Built a 16-bit hexadecimal address from four digits entered from the keyboard. Entry requirements: none. Exit: X-register contains the 16-bit address. A & B registers destroyed.	PDATA1 (E14B) — Print at terminal the ASCII data string pointed to by X-register. Data string must contain an ASCII EOT (04) as a terminator. Entry requirements: X-register contains the address of the 1st byte of the data string. The data string is terminated with a 04 character. Exit: A-register is destroyed. X-register contains address of 04 character.
BYTE (E106) — Input two hex characters from the keyboard and form a 1-byte number. Entry requirements: none. Exit: A-register contains the 8-bit number. B-register is destroyed.	OUT2H (E18D) — Output two hex characters, pointed to by X-register contains the address of the characters to be output. Exit: A-register is destroyed. X-register is incremented.
OUTH (E118) — Output left digit of hex number to console. Entry requirements: A-register contains hex number. Exit: A-register is destroyed.	OUT2HA (E10F) — Output two hex character in A-register to the terminal. Entry requirements: A-register contains the characters to output. Exit: A-register is destroyed. X-register is incremented.
OUTHR (E11C) — Output right digit of hex number to console. Entry requirements: A-register contains hex number. Exit: A-register is destroyed.	OUT4HS (E196) — Output four hex characters (2 bytes) plus a space to the terminal. Entry requirements: X-register contains address of first byte. Exit: A-register is destroyed. X-register contains address of second byte.
OUTCH (E126) — Output one ASCII character to terminal. Entry requirements: A-register contains ASCII character output. Exit: No change.	OUT2HS (E198) — Output two hex characters (1 byte) and a space to the terminal. Entry requirements: X-register contains address of byte to output. Exit: A-register is destroyed. X-register is incremented.
INCHP (E133) — Input one character, with parity, from terminal to A-register. Entry requirements: None. Exit: A-register contains character input.	
OUTS (E19A) — Output a space. Entry requirements: none. Exit: A-register destroyed.	

Hard copy for your Motorola D2 kit

by MICHAEL MOTE

Those who have the Motorola MEK6800D2 evaluation kit should find this article of great interest. Written by an applications engineer at Daneva Control Pty Ltd, it tells how that company's "Duoprint" miniature printer may be used with the D2 kit to provide hard copy facilities. Details are also given for a hex dumping routine to suit the Duoprint.

There's nothing quite like hard copy. You can touch it, hold it, store it, scribble changes on it and even tear it to shreds if things are getting you down. If you've worked on a computer system with hard copy facilities, no doubt you wonder how you could get along without it. And if you are currently struggling along without it, hopefully you aren't fully aware of what you're missing.

Admittedly the main thing which has prevented many microcomputer users from adding hard copy facilities to their systems has been cost. The price of printers has fallen over the last few years, but the cost of microcomputers themselves has plummeted. It is only in the last year or so that small printers have appeared at a price which makes

them attractive enough for use with microcomputers which themselves cost only a few hundred dollars.

A miniature printer designated the Duoprint has been developed and is being marketed by a Melbourne company, Daneva Control Pty Ltd. The Duoprint is a combination of an overseas mechanical unit with a locally-designed electronics module, and has a price tag of \$275 plus tax where applicable. It uses 60mm-wide metallised paper, and is capable of printing the full 64 characters of the 6-bit ASCII subset. Either 20 or 40 characters per line may be printed, selected by logic, at a speed of two lines per second.

Duoprint is capable of interfacing to a microcomputer or microcontroller system via either an asynchronous serial

line or a parallel port with "handshaking". However, the easiest way to connect it to the Motorola MEK6800D2 evaluation kit is to use parallel interfacing, via the user-defined PIA device already provided on the D2 kit PC board. The interfacing required is quite simple, as shown in Fig. 1.

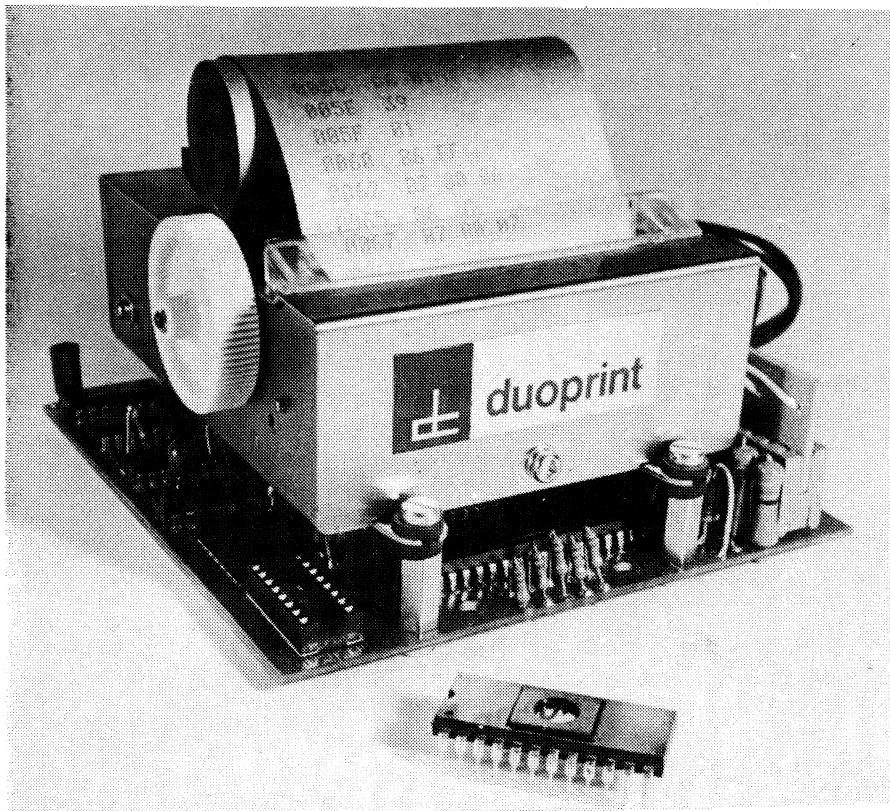
Power requirements for the Duoprint are +5V at 210mA and -24V at 1A. The -24V supply is for the printer head, and should be provided with a 1000uF bypass capacitor to ensure a high peak current capability. An internal regulator on the Duoprint PCB makes available a -12V supply capable of supplying up to 25mA, which may be used by external logic.

Daneva Control offers a firmware package designed to facilitate use of the Duoprint/D2 kit combination. Called PRINTBUG, the firmware normally sells for \$29.95, and comes in a 2708 EPROM which plugs into the IC socket "U10" on the PC board of the D2 kit. This is shown in Fig. 1, together with the links required on the PCB to suit the 2708.

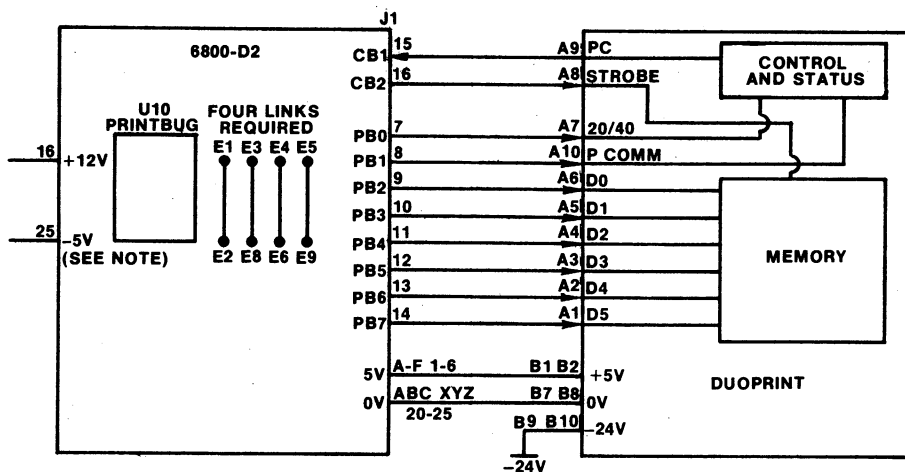
As the 2708 requires both +12V and -5V supplies in addition to the +5V provided on the D2 kit, these additional supplies must be provided. Luckily the -5V may be derived from the -12V output on the Duoprint, using a series resistor and 5V zener diode. These can be added to the D2 PCB in locations provided for this purpose — see the D2 kit manual.

PRINTBUG allows Duoprint to print any data accessible to the 6800 processor, whether the data is in RAM, ROM or the registers of interface adapters. The data can be printed in block format, with "N" bytes per line (where N is programmable). Alternatively when a program is being dumped, PRINTBUG may be arranged for partial disassembly. This causes each instruction to be printed on a separate line, as a group of one, two or three bytes (in hexadecimal) as appropriate.

The program for PRINTBUG was developed jointly by the author and Mr G. Foley, Lecturer at Box Hill Technical College. Mr Foley acted both as a software consultant and as an experienced microprocessor user who



At left is Daneva Control's Duoprint module, with an EPROM containing the PRINTBUG routine in front.



NOTE: THE -5V SUPPLY MAY BE DELETED BY USING THE -12V SUPPLY IN DUOPRINT AND ADDING THE ZENER/RESISTOR IN THE LOCATIONS PROVIDED ON BOARD

Fig 1: Here is the interfacing required to connect the Duoprint to the Motorola MEK6800D2 evaluation kit. The links shown in the D2 kit PCB are for the 2708 PRINTBUG EPROM.

helped set the PRINTBUG goals. His help is greatly appreciated.

For those who would like to either run PRINTBUG in RAM or blow it into their own EPROM or fusible-link PROMs, a complete hex listing is shown in Fig. 2. As you can see it occupies only 261 bytes, being currently located in the hex address range C000-C105.

PRINTBUG uses a few spare locations in the JBUG-dedicated RAM of the D2 kit, for its own scratchpad and control parameter storage. There are seven control parameter storage locations, from A050 to A056 inclusive, storing five control parameters as shown in Table 1. Each of the five parameters shown must be specified before PRINTBUG is called.

The first two locations store the starting address of the block to be printed, with the more significant byte in A050 and the less significant byte in A051, ie. to start listing at location 03F7, store 03 in A050 and F7 in A051.

The next two locations store the end address of the block to be printed, with the more significant byte in A052 and the less significant in A053. But note that the contents of the end address given are not actually printed, so that the address given should be of the location one higher than the last location to be printed.

The format control parameter stored in location A054 determines whether PRINTBUG prints in data block mode, or in program mode. To produce printing in program mode, where the bytes

for each instruction are printed on a separate line, load hex "00" into A054. Note that any data encountered by PRINTBUG in this mode will be interpreted as instruction codes, and "disassembled" accordingly.

To print out in data block mode, the control parameter stored in location A054 should be "0N", where N is the number of bytes to be printed per line. Needless to say the maximum number of bytes per line depends upon the character size; with 20 characters per line you can only print 5 bytes per line, while with 40 characters per line you can print up to 11 bytes per line.

For convenience I suggest you specify 4 bytes per line (code 04) if you are using 20 characters per line, and 8 bytes per line (code 08) if you are using 40 characters per line. This fits in neatly with the hexadecimal addresses.

The characters per line control parameter is stored in location A055. To make PRINTBUG print 20 characters per line, store "81" in this location; to make it print 40 characters per line, store "80".

The final control parameter is the exit control code, stored in location A056. This must be set to suit the way in which PRINTBUG is used. If it is called as a subroutine, the stored exit code must be "39", corresponding to an RTS instruction. On the other hand if you are running PRINTBUG as a free-standing program, use the code "3F", corresponding to an SWI instruction. Or if PRINTBUG is going to be accessed as

PRINTBUG listing:

```

C000 49 25 09 46 C008 8D D1 B6 A0
C004 46 46 84 3C C00C 58 48 48 48
C008 8B C2 26 12 C090 48 8D C8 86
C00C 2B 07 49 2B C094 83 87 88 86
C010 03 46 20 EF C098 87 88 06 F6
C014 46 46 8B 70 C09C 80 54 26 02
C018 46 46 84 3C C0A0 8D 88 A6 00
C01C 8B 02 B7 80 C0A4 8D B5 A6 00
C020 86 39 A6 00 C0A8 48 48 48 48
C024 81 8C 27 2D C0AC 8D A0 86 83
C028 81 8F 27 29 C0B0 87 88 06 88
C02C 81 CE 27 25 C0B4 8C A0 52 27
C030 84 F0 81 78 C0B8 16 5A 26 E6
C034 27 1F 81 80 C0BC 86 A0 55 87
C038 27 1B 81 F0 C0C0 80 06 FF A0
C03C 27 17 81 00 C0C4 57 B6 88 07
C040 27 16 81 10 C0C8 2A F6 F6 80
C044 27 12 81 30 C0CC 06 20 A0 B6
C048 27 0F 81 40 C0D0 A0 55 B7 80
C04C 27 0A 81 50 C0D4 06 B6 88 07
C050 27 06 C6 02 C0D8 2A FB F6 80
C054 39 C6 03 39 C0DC 06 CE C0 F1
C058 C6 01 39 20 C0E0 A6 00 B7 80
C05C A3 7F 88 07 C0E4 86 08 8C C1
C060 FE A0 50 FF C0E8 05 26 F5 F7
C064 A0 57 01 86 C0EC 80 06 7E A0
C068 FF B7 88 06 C0F0 56 83 83 12
C06C 86 2C B7 80 C0F4 56 3E 42 4A
C070 87 B6 88 06 C0F8 26 3A 51 83
C074 FE A0 57 86 C0FC 0E 3E 36 42
C078 A0 57 8D 84 C100 32 16 51 16
C07C 86 A0 57 48 C104 83 83
C080 48 48 48 5D DUOPRINT COMPLETE
C084 D6 B6 A0 58

```

A full hex listing of PRINTBUG, printed out by the program itself driving a Duoprint. This is an example of the data block mode, with the program set for 4 bytes per line. (The listing has been cut in two, with the halves pasted side by side).

the result of an interrupt, you should use the code "3B", corresponding to an RTI instruction. One of these three codes should be stored in location A056.

Once the five control parameters have been stored, PRINTBUG may be called for execution. Its starting address is C05D.

Note that the PCB jumpers shown in Fig. 1 are for the 2708 version of PRINTBUG and may not be appropriate for other PROMs, including the single-supply equivalents to the 2708. The links shown are suitable for the MC68708, but for the MCM7641 you need only three links, joining E3-E9, E4-E7 and E5-E7. Similarly for the MCM68317 you again need only three links, joining in this case E0-E4, E1-E2 and E5-E9.

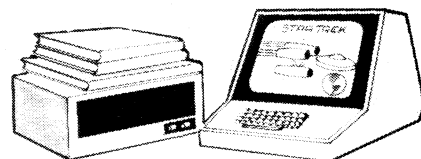
You can get some idea of the results possible with the Duoprint and D2 kit combination from the PRINTBUG listing, which was itself produced by such a combination working in the data block mode. As you can see the characters are well formed and easy to read.

Well, there it is: a simple way of adding hard copy facilities to a low cost D2 evaluation kit. Once you've tried it, you'll wonder how you got along without it!

TABLE 1: PRINTBUG CONTROL PARAMETERS

Parameter	Location in scratchpad (Hex.)
Start Address	A050/A051 (high & low bytes respectively)
Stop Address	A052/A053 (high & low bytes respectively)
Format Control	A054 (00 for program print, 0N for data)
Character Control	A055 (80 for 40 chars, 81 for 20 chars)
Exit Control	A056 (39, 3F or 3B)

Microcomputer News & Products



New Motorola MEK6802D3 uses 6800 hardware and software and is greatly expandable

Choose a single board computer, add a suitable power supply and, perhaps, some additional external memory; put these into an enclosure and you have a complete microcomputer — ready to go to work.

Motorola have released their MOKEP range of low-cost, assembled and tested microcomputer products to enable people to do just that.

The MEK6802D3 microcomputer module is a single board with eight LED digits and hex keypad. Based on the MC6802 MPU/RAM/CLOCK chip and the MC6846 ROM/IO/TIMER chip, this module allows a user to easily evaluate and use Motorola's M6800 hardware and software. The module features a 2k byte Monitor (D3BUG), 256 bytes of user RAM and 128 bytes of stack, and is fully expandable with the other products in the MOKEP range via a 60-pin bus.

Memory requirements for EPROM/ROM or RAM can be satisfied through the inclusion of the three memory modules offered — the MEK68MM16 16k byte dynamic memory, the MEK68MM32 32k byte dynamic memory or the MEK68RR ROM and RAM board. The latter accepts up to 8k bytes of static RAM

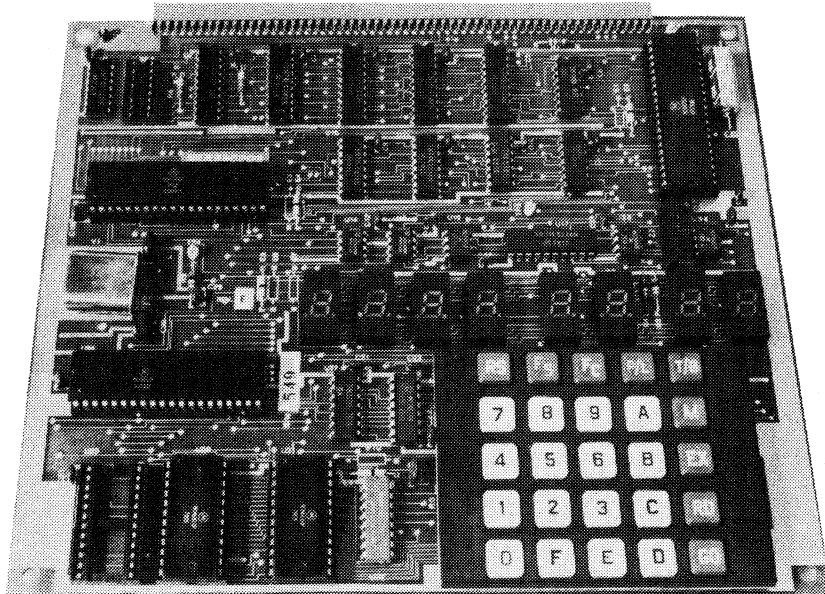
and up to 32k bytes of EPROM or 64k bytes of ROM. The MEK6802D3 has memory paging circuitry giving an expansion capability to 160k bytes of ROM and 256k bytes of RAM.

Owners of the ever popular MEK6800D2 kits have not been forgotten. Each of the above-

mentioned boards has a counter-part compatible for use with the D2.

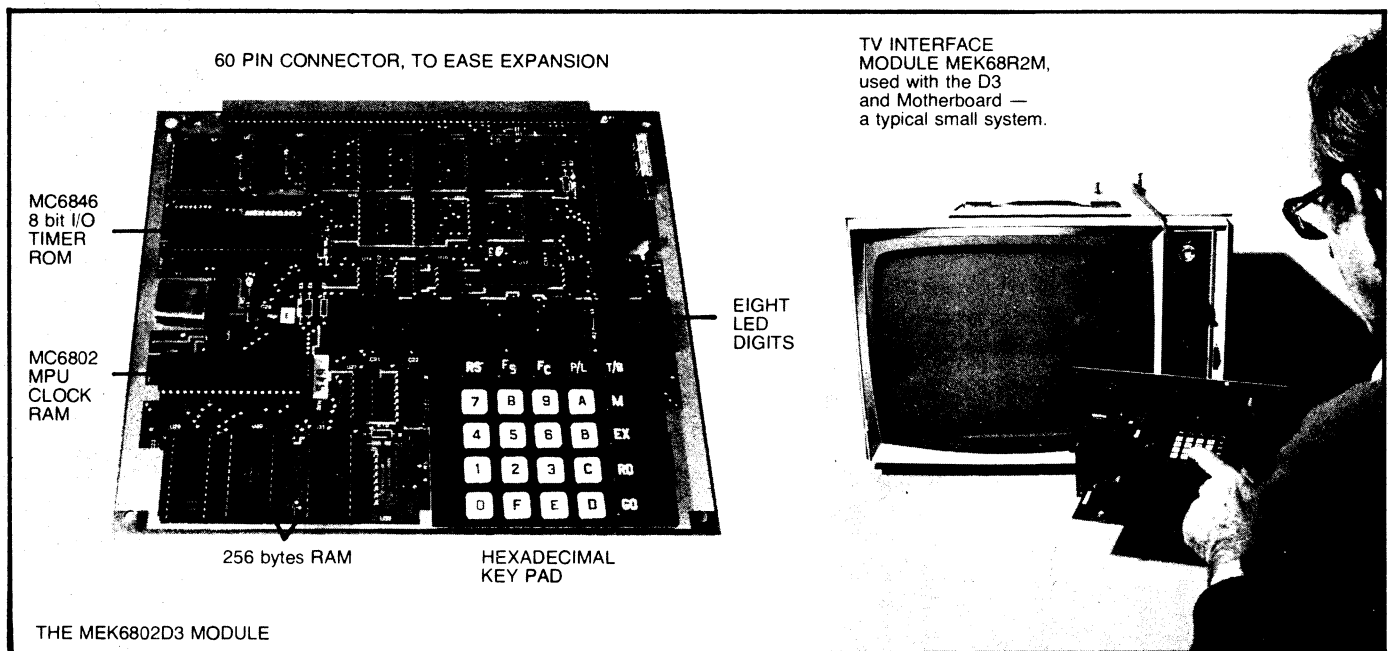
Further information about these products is available from Total Electronics: Melbourne, (03) 596 2891; Sydney (02) 428 3500; Brisbane (07) 52 4357.

Micronews Continued ➔



D3, THE NEW LOW COST POWER BASE TO BUILD YOUR SYSTEM ON!

Motorola's Microcomputer Expansion Systems are built around the completely new D3 module, and that means you now get all the versatility you want on one printed circuit board. This new module is the power base on which you can build your own powerful micro system, and it's big capacity allows you to construct simple or complex systems — as you want them.



D3 features a multilevel memory paging technique, memory expandable to 256 K bytes RAM, and 160 K bytes ROM, a 2 K byte monitor — D3 BUG, for control of the systems microprocessor through the D3 hexadecimal keypad and LED display.

And there's the 16 K byte dynamic memory module, a television interface board, the combination card cage

and motherboard, wirewrap board, and a RAM/ROM board.

No assembly or soldering because Motorola have designed each module to simply plug into the motherboard and the D3 module does the rest!

If you have one of the ever popular D2 kits — don't panic. There is a new range of expansion products to suit these kits too!

D3 IS NOW AVAILABLE FROM:

Stewart Electronic Components Pty. Ltd.
44 Stafford Street
Huntingdale, Vic. 3166. Ph. (03) 543 3733

Ellistronics
289 Latrobe Street
Melbourne, Vic. 3000. Ph. (03) 602 3836

Electronic Distributors
Shop 2-4, 7-10 Joyce Street
Post Office Arcade
Pondie Hill, N.S.W. 2145. Ph. (02) 636 6052

Martin De Launay Pty. Ltd.
Cnr. King & Darby Streets
Newcastle, N.S.W. 2300. Ph. (049) 2 4741

Radio Dispatch Pty. Ltd.
869 George Street
Sydney, N.S.W. 2000. Ph. (02) 211 0191
Electronic Components (Qld)
16/205 Moggill Road
Taringa, Qld. 4068. Ph. (07) 371 5677

George Harvey Electric Pty. Ltd.
162 Argyle Street
Hobart, Tas. 7000. Ph. (002) 34 2233
76 York Street
Launceston, Tas. 7250. Ph. (002) 31 6533

Motorola 6802D3 evaluation kit

Features a hexadecimal keyboard and LED readout

As recently announced, Motorola has just released a new microprocessor evaluation kit based on the MC6802. Called the MEK6802D3, the new system is supplied fully assembled and supersedes the well known MEK6800D2 two-board evaluation kit. The MEK6802D3 is accommodated on a single PCB which has a hexadecimal keyboard and an eight-digit LED readout.

by JOHN CLARKE

Many of our readers will probably be sorry to see the end of the Motorola MEK6800D2 but they may rest assured that most of its features are carried on in the newly released MEK6802D3 system.

The single board D3 system employs the MC6802 microprocessor chip which has all the registers and accumulators of the MC6800 plus an internal clock oscillator and driver. In addition, the 6802 has 128 bytes of RAM and a "Vcc standby" feature which allows the first 32 bytes of RAM memory to be retained in a low power mode.

In addition to the MC6802 chip there are two MC6810 static RAMs, one MC6821 Peripheral Interface Adaptor (PIA) and MC6846 ROM-I/O-Timer. A 3.579545MHz crystal reference is provided for the clock. This frequency is divided by four within the MC6802 chip itself which consequently operates on 894.88kHz.

The MC6821 PIA is provided for interfacing the keyboard and LED displays to the processor. The only user input/output device is the relatively new chip, the MC6846. The MC6846 provides 2048 8-bit bytes of ROM, an 8-bit bidirectional data port for parallel interface plus two control lines and programmable interval timer-counter functions. The ROM accommodates the D3BUG monitor program. The data port differs from a standard PIA in that all the control registers can be accessed so that, for example, the choice of positive or negative edge triggering and/or latching can be easily implemented.

The timer within the MC6846 can be operated in various modes such as pulse width comparison, frequency comparison, normal single-shot con-

tinuous and cascaded single-shot operations. Comprehensive information is given on all these modes in the data sheets.

The total volatile memory (RAM) available on the D3 computer board is 256 bytes.

The D3 evaluation kit is supplied fully assembled and tested. All that is needed to have it running is a 5V power supply capable of delivering 900 milliamps. Dimensions of the double-sided PCB are 210 x 180mm.

Comprehensive literature is provided in loose-leaf form, suitable for insertion into a standard 3-ring binder. The literature includes a manual on the D3 system itself, with chapters on general description and operation, hardware, software and expansion. Data sheets on all the major chips in the D3 system are provided plus schematics and an appendix on number systems and 6800 addressing modes. Also supplied is a programming reference manual on the M6800.

The D3BUG monitor is quite an advance on the JBUG monitor program of the D2 kit. With the D2, memory could only be advanced when in JBUG, however, with the D3BUG memory can be also stepped backwards. Breakpoints with the D3 computer can be removed separately, whereas the D2 monitor only removed breakpoints all at once. Up to eight breakpoints can be inserted compared to the D2's five.

Using breakpoints is easy with the D3BUG since the registers and their contents are displayed. The D3BUG will also calculate offsets for branch instructions. Just enter the start address, type the "FS" key and enter the final address. Type "GO" and the hexadecimal offset is displayed. The D2 kit had a

separate program to calculate offsets.

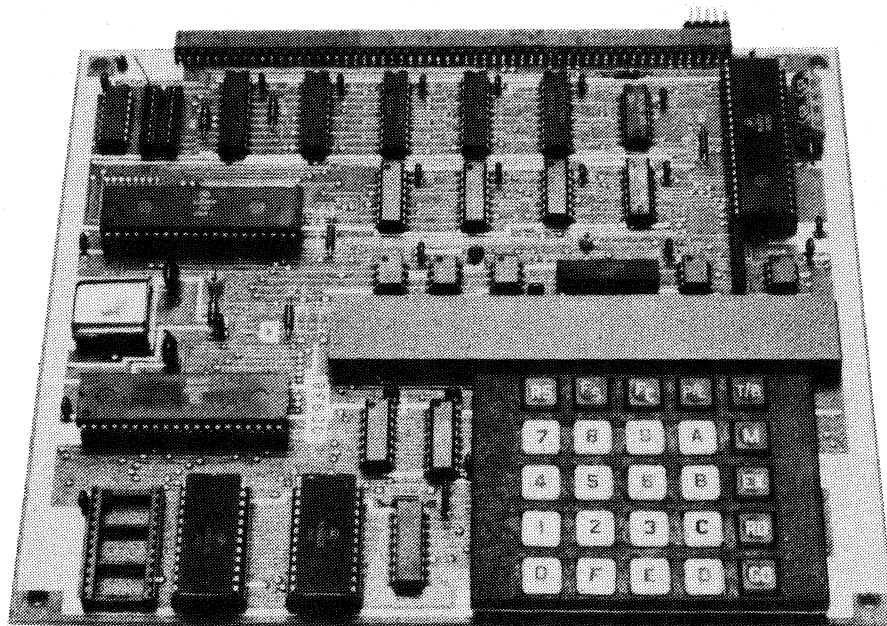
The D3 computer as it stands is not capable of transferring RAM contents onto an audio tape, although there is routine in the D3BUG monitor program. An extra board is required to be connected to the D3 system bus to enable use of the dump and load routine.

A chapter in the manual is devoted to expansion of the D3 computer. This chapter is well written and describes the necessary modifications required for the expansion boards. Extra memory boards and a colour TV monitor interface board are examples of the variety of expansion boards available.

Working through the introductory program load-run-debug example which Motorola have thoughtfully provided in the kit manual, I quickly learned to operate the system in D3BUG. The example is well done and should be equally effective in introducing a newcomer to microprocessing systems.

Programming the system was more difficult until I was able to sort out the addressing modes. Once this was done, another problem arose; how to access the I/O peripheral port? The manual refers to the PIA timer address being located at 8080 to 8087 but the order in which the registers are located is not clear. Reference to the data sheets showed a truth table for the register selection.

After conversion of the addresses 8080 to 8087, it became clear that the registers are located in memory in the order in which they occur in the above truth table. With that sorted out I was able to write a short program which



The Motorola 6802D3 evaluation kit is supplied fully assembled and tested.

MOTOROLA PROGRAM

0000 8E 00 7F	LDS STACK	DEFINE STACK IN USER AREA
0003 86 FF	LDAA \$FF	
0005 B7 80 82	STAA DDR	SET DATA LINES AS OUTPUTS
0008 86 01	LDAA \$01	SET DATA LINE PO HIGH
000A B7 80 83	AGAIN STAA PDR	ONE LINE HIGH
000D BD 00 14	JSR DELAY	KEEP PDR LINE HIGH WITH DELAY
0010 49	ROLA	SET NEXT LINE HIGH
0011 7E 00 0A	JMP AGAIN	
0014 CE FF 00	DELAY LDX \$FF00	SET DELAY TIME
0017 09 DELI	DEX	DECREASE INX REGISTER
0018 26 FD	BNE DELI	RETURN IF NOT EQU ZERO
001A 39	RTS	DELAY ENDED

The author's sample program. It sequentially sets each line of the user I/O port (the SK1 socket) to a high state, and keeps it there for a brief period.

demonstrates that the I/O port is working.

What the program does it to set the PIA of the MC6846 to a "write" operation. Each data line P0 to P7 is in turn set to a high state. A delay subroutine keeps each data line high for a short time. If a meter is connected from the negative supply to one of the outputs on the SK1 socket, while the program is running, the operation can be watched.

To summarise, the MEK6802D3 is a

small, well thought out microprocessor development system, adequate for learning 6800 programming. It has plenty of provision for expansion so that it can be made into a more elaborate system if desired. In short, it is good value for money.

The MEK6802D3 computer is distributed by Total Electronics and our computer for review came from Radio Despatch Service, 869 George Street, Sydney. Price of the computer is \$175 plus 15% sales tax where applicable.

ANNOUNCING 68000 DEVELOPMENT SYSTEMS

The EXORmacs Development System for 68000 has just been announced. The system includes 15 slot chassis and power supply, MPU module, memory management module, deBUG module, 128K byte dynamic memory module, and an intelligent floppy disk controller module.

Software includes Structured Macro Assembler/Linkage Editor, screen based editor, symbolic deBUG and a PASCAL compiler.

Peripherals include intelligent CRT console, a 1 Mbyte, two drive double sided floppy disk, and a model 703 (180 CPS 132 column) line printer.

A large number of support modules are also available.

6800/6809 EXORciser Development Systems

The popular EXORciser II Development System is now available in both 6800 and 6809 versions. For those people already owning a 6800 EXORciser or EXORterm Development System, a 6809 upgrade kit is available. Both EXORcisers may be expanded to allow development of the

MC6802/MC6808

MC6801/MC6803/MC68701 MC6805

A wide range of support modules are available, including support modules for PIA, ACIA, SSDA, ADLC, GPIA, CRT controller and universal support module.

High level languages are available for —

6800: BASIC, FORTRAN, COBOL, MPL.

6809: FORTRAN, MPL, PASCAL.

Bulk storage is available on both single and double sided floppy disks. Up to 4 drives are supported.

A 10 Mbyte Hard Disk System is also available with expansion to 40 Mbyte. The Hard Disk System may co-reside with the floppy system to allow easy transfer of old files.

68000 Courses

Rank Electronics has much pleasure in announcing a visit from two leading Motorola lecturers to give courses on the MC68000.

Subjects will include both hardware and software aspects.

Course costs: \$250.00 per head for each course.

Dates: Sydney — May 19, 20, 21 and 22.
Brisbane — May 23, 26 and 27.
Adelaide — May 28, 29 and 30.
Melbourne — June 2, 3, 4 and 5.
Hours: Sydney and Melbourne — 9.00 a.m. to 5.00 p.m.
Brisbane and Adelaide — 8.30 a.m. to 7.30 p.m.

For further details and application form please contact:



**RANK
ELECTRONICS**
PTY LIMITED
INDUSTRIAL SYSTEMS

12 Barcoo Street, East Roseville, Sydney 2069.
Phone 406 5666

60 Rosebank Ave, Clayton South, Vic 3169
Telephone 541 8444

299 Montague Road, West End, Brisbane 4101.
Telephone 44 2851

101-105 Mooringe Avenue, Camden Park, SA 5038.
Telephone 294 6555

430 Newcastle Street, Perth 6000.
Phone 328 3933

120 Parry Street, Newcastle 2300.
Phone 26 2466

MAS5561

MC68000

Break away from
the past.



Motorola's MC68000 sets new MPU system performance standards for the '80s. Now.

Motorola announces a new microprocessor so advanced in concept it offers the designer virtually unlimited freedom of system design. Advanced technologies provide it with a speed/power product four times that of standard NMOS. Break away from the past and step into the new era of microprocessors.

Everything you need for system design is available now—a Design Module for evaluating the MC68000, development hardware and software, full documentation and training. In addition, the MC68000 interfaces directly and easily with existing M6800 Family peripherals, MPUs, and MCUs.

The MC68000 is, by design, perfect for an easy-to-handle, block-structured, high-level language like PASCAL. It simplifies modern programming techniques like modular programming.

Complete M68000 design support available now. Concurrent with introduction of the MC68000, the design tools to help you break away from the past are now available.

A major commitment to development of timely and effective M68000 hardware and software has been part of the program from the beginning. The results, part innovation and part evolution, give you a choice among Motorola's EXORciser®, the

IBM 370 and the PDP*-11 for system hardware and software development.

Motorola offers the total range of microcomponents for your advanced systems.

Motorola's original M6800 Family pioneered the concept of the fully-compatible, matched set microprocessor plus I/O, peripheral controller and memory family, complete with support hardware, software and training.

The MC68000 springs from a rich tradition in both microcomponents and MOS technology and is designed to develop in the same total family way as the M6800 Family.

Pace-setting products like the MCM6664 64K dynamic RAM, the MC6809 super 8-bit microprocessor, and the definitive MC6801 and M6805 Family one-chip MCUs have firmly established Motorola's capability.

For further details and application form please contact:



**RANK
ELECTRONICS**
PTY LIMITED
INDUSTRIAL SYSTEMS

12 Barcoo Street, East Roseville NSW 2069. Phone: 406 5666
60 Rosebank Avenue, Clayton South Vic 3169. Phone: 541 8444
299 Montague Road, West End, Brisbane 4101. Phone 44 2851
101-105 Mooring Avenue, Camden Park SA 5038. Phone 294 6555
430 Newcastle Street, Perth WA 6000. Phone: 328 3933
120 Parry Street, Newcastle NSW 2300. Phone: 26 2466

*Trademark of Digital Equipment Corporation.

MAS5694



Motorola evaluation board for the MC68000

Motorola Inc has released an evaluation board for the MC68000 16-bit microprocessor. For those interested in exploring the capabilities of this chip the MEX68KECB (Educational Computer Board) provides a (relatively) low cost 68000 system, needing only a power supply and a serial terminal.

Intended primarily for training and educational use, the Educational Computer Board serves as a comprehensive introduction to systems based on the 68000 microprocessor. The board measures just 264 x 168mm yet provides everything necessary to begin writing and testing machine language programs for the MC68000 16-bit microprocessor.

On the board are a MC68000 processor running at 4MHz, 32K bytes of

selected by jumper blocks and may be any standard rate between 110 and 9600 baud.

So that the ECB can be used immediately, extensive software is included in ROM on the board. Called "Tutor", this software provides the user with a monitor and debugging package, assembler and disassembler and I/O control functions.

The MC68000 microprocessor is often

range of 16 megabytes (16,777,216 bytes). A CHK (Check register against bounds) instruction allows the programmer to create a variety of memory protection/management routines.

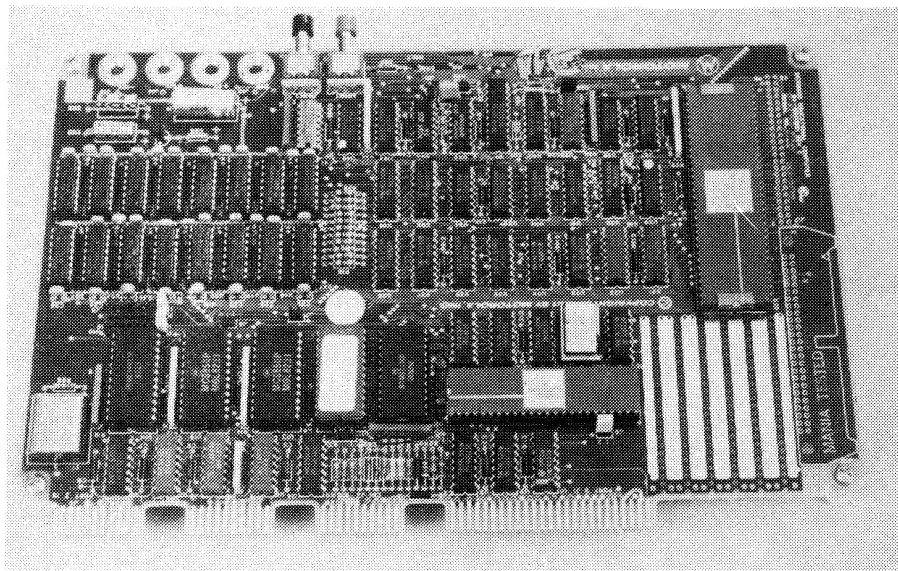
Other features of the processor are the ease of single step debugging provided by a trace mode called up by the instruction "TRAP". The processor also makes a distinction between two modes of operation, the "user" and "supervisor" states. The user mode prohibits access to certain of the memory instructions, allowing programs for multi-user systems to be written in such a way so that they will not interfere with each other even in the event of an error.

The 16K "Tutor" software provided with the Educational Computer Board is basically a system monitor which controls communication with the terminal and allows the creation and testing of assembly language programs. The line-by-line assembler does not save the source program but translates each instruction into the correct machine code for storage in memory. Facilities are limited to the specification of operation and operands — line numbers, labels and comments are not allowed.

The disassembler also operates on a line-by-line basis, disassembling the machine code in memory and displaying the instruction mnemonic and operands. No editing facilities are provided other than re-entering the source statement. For other facilities a macro assembler or cross assembler can be run on a host computer, with data communicated to the 68000 board through the second serial port.

In addition to the assembler/disassembler, Tutor provides commands to display and modify memory and registers, commands for executing programs, either continuously or with breakpoints, or in a Trace mode. Other commands allow blocks of memory to be filled, moved or searched for a specified character and there are also commands for controlling the serial communications ports and parallel printer interface and for dumping and loading programs from a cassette recorder.

Documentation provided with the board consists of a 200-page loose leaf manual which covers the set up and



The MEX68KECB board provides 32K of RAM, parallel and serial ports and software.

dynamic RAM arranged as 16K x 16 bit words, two RS232C serial communications ports, a 16-bit parallel input/output port (which can be configured as a Centronics compatible printer port), a cassette tape interface for program storage, a 24-bit programmable timer and Reset and Abort pushbutton switches.

For those wishing to add their own interface circuits a small prototyping area is also provided.

The user must add power supplies of $\pm 1V$ and $\pm 12V$, and a serial terminal. Communications rates for the board are

referred to as a 16/32 bit microprocessor, as it has a 16-bit external data bus but internally is organised around 32-bit registers. In addition to a 32-bit program counter and 16-bit status register, the processor provides 17 32-bit registers. Of these, the first eight may be used for byte, word or "long word" (32-bit) storage while the second group of eight and the 32-bit stack pointer can be used as address registers, for word and long word manipulation and as index registers.

The external address bus is 24-bits wide, providing a direct addressing