The Intel 8080A

Without a doubt the most popular and widely used of all current microprocessors is the Intel 8080A, the first of the "second generation" devices and one which is now supported by a very broad range of hardware and software. This article looks at the 8080A and its companion chips and then reviews the Intel SDK-80 evaluation and system development kit.

by JAMIESON ROWE

The Intel Corporation was the first to develop and market a microprocessor, back in 1971. In fact Intel's founder and Chairman Dr. Robert Noyce is acknowledged as the inventor of the microprocessor, and apparently still holds key patents.

The first Intel microprocessor was a 4-bit P-channel device, the 4004, with 46 instructions. This was followed by an enhanced 4-bit device with a repertoire of 60 instructions, the 4040. Then in 1973 came the first P-channel 8-bit processor, the 8008. This had a repertoire of 48 instructions, and an instruction cycle time of 20us (now 12.5us).

But undoubtedly the most popular Intel microprocessor to date has been the 8080, an enhancement of the 8008 which came on the market in early 1974. An N-channel device, the 8080 is about 10 times faster than its predecessor. It also has a larger instruction repertoire of 78 instructions.

The 8080 has been the most popular microcessor ever produced, and even though it may lack some of the features offered by later devices, it seems likely to remain popular for many years.

Intel themselves have announced a number of newer microprocessors, including a more self-contained device called the 8085 and a one-chip microcomputer with internal ROM and RAM called the 8048. But they are at pains to point out that these newer chips are not regarded as superseding the 8080.

One of the prime attractions of the 8080 from the commercial point of view is that it is now supported by a tremendous amount of software and applications information—much of it user generated. This inevitably tends to reduce development costs of an 8080 system compared with a system based on one of the newer chips, if other considerations are equal.

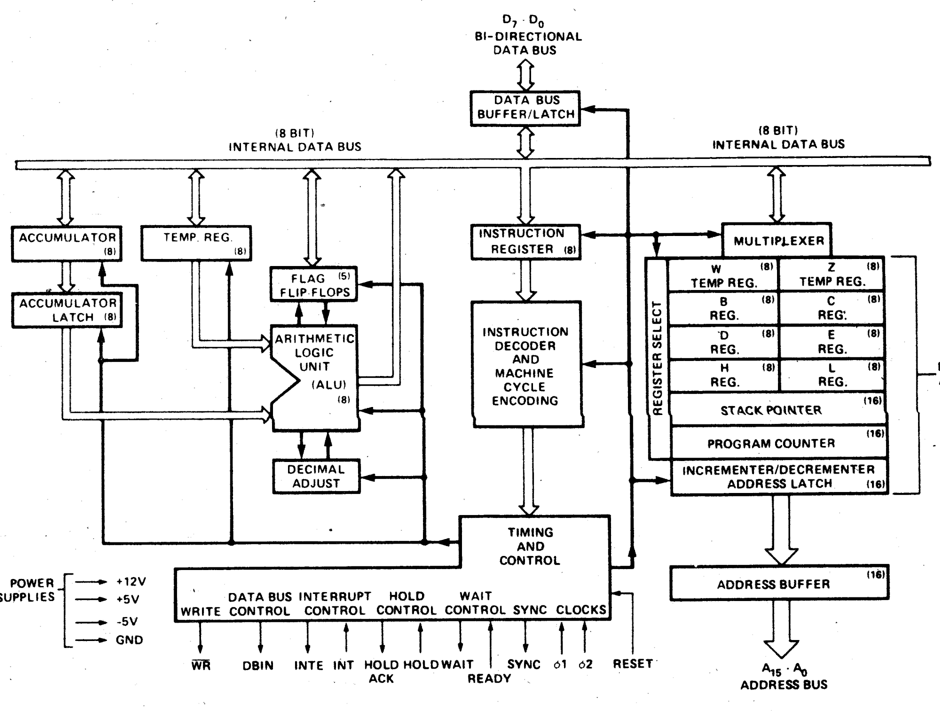
The block diagram for the 8080 CPU chip is shown below. Its architecture is not unlike many minicomputers. There are 10 internal registers accessible to the programmer, two of which are 16 bits long: the program counter and a stack pointer. A third register called the Program Status Word (PSW) provides the five status flags.

The remaining 7 registers are all 8 bits long. One is the single primary accumulator A, while the other six are secondary accumulator/data counters. Two of these, designated the H and L registers, are generally used together as a 16-bit data counter. The other four registers can also be grouped together in this way for some instructions.

Data and address information pass between the 8080 and the rest of a system via two separate buses, an 8-bit bidirectional data bus and a 16-bit address bus. The latter gives the 8080 the ability to directly address 65,536 or "65k" bytes of memory. The output buffers on both the data and address bus lines are 3-state, and may be disabled for external control of the system bus. This facilitates DMA operation, for example.

The 8080 implements its stack in external RAM, and as the stack pointer register is 16 bits long this means that the stack may be virtually anywhere in the 65k of memory space.

The 8080 status register or PSW has five flag bits. These signify zero result, result sign, carry, even parity, and 4 bit carry (for BCD arithmetic). None of the status



A functional block diagram for the 8080A microprocessor itself. There are 10 internal registers accessible to the programmer, two of which are 16 bits long: the program counter and the stack pointer. Apart from the 8-bit primary accumulator A there are six secondary accumulator/data counter registers and a 5-bit status register.

flags is accessible directly via external device pins, only internally via certain instructions.

Although I/O devices may be interfaced so that they appear in 8080 memory space, and are accessed via memory address instructions, the 8080 also has facilities for separate I/O addressing. 8 bits are allocated for peripheral addresses, so that there is potential for addressing up to 256 input ports and 256 output ports quite separately from normal memory space.

The 8080 chip uses N-channel silicon gate MOS technology. It requires three operating voltages: +5V, −5V and +12V.

There is no clock oscillator on the 8080 chip itself, which requires high level non-overlapping two phase clock signals. These are normally provided by the 8224, a companion device designed specifically as a clock generator and driver. It also provides synchronisation for certain system control signals.

Some of the system control signals generated by the 8080 are multiplexed out on the data bus lines, during one of the clock phases. To provide demultiplexing of these signals most 8080 systems use another companion device, the 8228 System Controller.

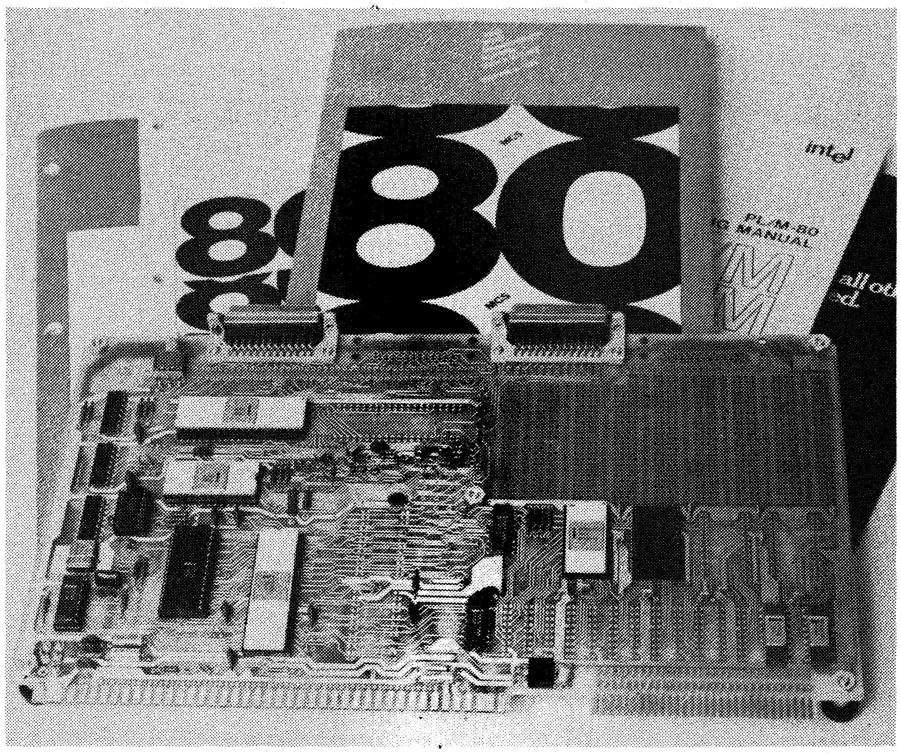
Actually the 8228 is not only a control signal demultiplexer, but a transceiver/driver for the data bus as well. And it also provides automatic control of 8080 interrupt vectoring—providing a hardware generated vector itself when none is provided by the interrupt source.

In responding to an interrupt request, the 8080 looks for an externally-supplied vectoring instruction. In most cases this is a one-byte jump to subroutine or RST instruction, capable of specifying one of eight subroutine starting addresses in the first 63 bytes of memory space. However for systems required to service more than eight interrupt sources, it is possible to use a 3-byte CALL instruction to routines anywhere in memory space.

Apart from the 8224 and 8228, Intel has provided the 8080 with a number of support devices to facilitate system design. Among these devices are the 8212 8-bit I/O port, the 8255 Programmable Peripheral Interface—which provides 24 bits of bidirectional parallel interfacing—and the 8251 Programmable Communication Interface which is basically a USART with programmable formatting and control.

There is also a large and still growing family of memory devices, with ROMs and PROMs as well as RAMs. These include a 2k byte mask-programmed ROM and a 1k byte UV-erasable PROM, with larger devices coming.

Also coming are more specialised interfacing devices, such as a programmable DMA controller, a programmable interrupt controller and a floppy disk controller and formatter.



The smallest 8080A-based system available from Intel themselves is the SDK-80 assemble-it-yourself kit, shown here assembled and with its accompanying manuals. A review of the kit is given in this article.

The instruction set of the 8080 includes 1, 2 and 3-byte instructions. While it is nominally said to comprise 78 different instructions, in fact almost all of the 256 possible op codes are used for distinct instruction variants.

Thus the data move or MOV instruction actually has 63 different variants, with different sources and destinations specified—including 14 which involve implied memory addressing. Similarly each of the 8 accumulator operate instructions has 8 variants, while the increment and decrement instructions have 12 variants each.

Included in the 8080 repertoire are 10 jump instructions, 9 CALL and 8 RST instructions for subroutine calling, 9 subroutine return instructions, 10 instructions for operating on the stack, and 20 immediate-addressing data instructions—four of which handle 16-bit data.

Apart from immediate addressing, the 8080 provides only two memory addressing modes. One is direct absolute addressing, used for 3-byte load, store, jump and call instructions. The other mode is implied addressing, used for all other memory reference instructions. These are all 1-byte instructions, and generally use the H and L registers as a 16-bit data counter.

An interesting aspect of the 3-byte instructions using direct absolute addressing is that the second instruction byte carries the eight LEAST significant address bits, with the third byte having the eight MOST significant bits. This is the opposite of the scheme used by virtually all other microprocessors, and slightly confusing at first. However, it doesn't take long to make the mental adjustment.

The variety of conditional branching instructions provided by the 8080 is quite impressive. Each of the three types of conditional branch instruction (jump, call and return) has eight condition tests available: not zero, zero, no carry, carry, parity odd, parity even, plus, and minus.

This feature makes it possible to write very compact and efficient programs for the 8080, particularly if liberal use can be made of subroutines and nesting.

In short, although it was one of the first 8-bit microprocessors developed, the 8080 has a powerful instruction set and compares well in this respect with many of the later chips. This together with the large amount of support available on the software side still makes the 8080 a very popular device.

Intel itself has produced a number of microcomputer systems based on the 8080, ranging from quite pretentious development systems with floppy discs and in-circuit emulation facilities down to single-board systems for OEM applications. Generally these are supplied as completely wired and tested systems.

Quite apart from these is the SDK-80

System Design Kit, a small 8080 system sold as a kit and designed as a low cost evaluation and prototyping tool.

A sample of the SDK-80 kit was made available to us for evaluation by the new Australian distributors for Intel, Warburton Franki Pty Ltd. We were thus able to see how the kit goes together, and to use it to try our hand at programming an 8080-based system.

The SDK-80 provides all parts required to build a complete single-board 8080 system. The kit comes together with a set of literature which includes an 8080 user's manual, an assembly language programming manual, kit instructions and a user's guide for the SDK-80 itself, a programming reference card and other material.

When assembled the SDK-80 provides an 8080 system with 256 bytes of RAM and a 1k-byte ROM containing a monitor-debug program. A 1k-byte EPROM is also provided for user program storage. In addition, the PC board provides decoding and DIL locations for simple expansion of the system to one having 1k bytes of RAM and 4k bytes of ROM/PROM, merely by wiring in the additional memory chips.

The kit also provides an 8255 programmable peripheral interface (PPI), giving 24 bits of parallel I/O interfacing. A second 8255 is provided for on the PCB pattern, to provide a further 24 bits if desired. In addition the kit includes an 8251 USART for communication with a teleprinter, video terminal or another serial terminal. This is supported by a baud-rate generator deriving 7 standard data rates from the 8080 system clock: 75, 110, 300, 600, 1200, 2400 and 4800 baud.

The serial interfacing may be set by jumpers for either 20mA current loop, RS232-type voltage levels or TTL logic levels. All other interfaces to the SDK-80 are TTL compatible, including the 8255 parallel interface.

Two 25-pin RS232C sockets are provided on the PCB for interfacing, and matching plugs are provided. There is also provision on the PCB for address bus buffering, so that the SDK-80 may be expanded beyond the PCB via the main edge connector.

The monitor program which comes with the SDK-80, in the ROM, provides six basic commands. These are listed below, together with the command letter:

Display memory contents.............	D
Move blocks of memory	M
Substitute memory locations	S
Insert hex code................................	I
Examine registers	X
Go to user program........................	G

Normally the I command is used for feeding in a user program from the terminal keyboard, and the D command for checking that it has been entered correctly. The S command can be used to correct or otherwise change instructions or data in memory, and the X command to set the 8080 registers before a program is run, by then using the G command.

The M command is a rather powerful one, making it possible to move a block of instructions or data along in memory. This can save a lot of tedious re-entering, for example, if you dicover you have left out an essential instruction near the start of a program!

To my knowledge no other small microprocessor evaluation kit has a monitor program providing such a "move" command, which is generally found on more pretentious systems. So the SDK-80 is rather unique in this respect.

On the other hand most other evaluation kits have monitors which provide commands to allow a program in memory to be dumped onto paper tape or cassette, and then reloaded again next time. Strangely enough the SDK-80 monitor doesn't seem to provide commands for this purpose: while the D command could be used for dumping, the dumping format is not compatible with that used by the I command.

This seems a disappointing oversight on the part of the SDK-80 designers. I suspect most users would have been prepared to forego the luxury of the M command, if this had been necessary in order to provide for convenient dumping and loading of programs.

It took about 7 hours to assemble the SDK-80 from the kit, taking care with the soldering as the PCB has narrow conductors and closely spaced pads. I found the kit assembly instructions quite clear, although one has to be careful when it comes to fitting the various option links. The designers have provided an almost bewildering array of options, in an effort to make the kit as flexible as possible.

The finished unit requires three power supplies: +5V at 1.3A, +12V at 350mA and –10V or –12V at 200mA.

When the sample SDK-80 was completed, I connected it to the EA Video Data Terminal and turned on the power. It responded with the encouraging message "MCS-80 KIT", and gave its prompt sign (a full-stop), to show that the monitor program was awaiting a command.

It was when I tried to "talk back" that a minor problem became apparent: nothing happened!

After a little troubleshooting, the reason for the lack of communication became apparent. In their wisdom, the SDK-80 designers have set the "mark" current level for the teleprinter keyboard input at around 40mA, double the nominal 20mA figure.

With the mechanical contacts of a normal teleprinter this current level would cause no problems, but the keyboard

```
NOVELTY ANSWER-BACK PROGRAM FOR INTEL SDK-8Ø KIT
WRITTEN BY J.ROWE, ELECTRONICS AUSTRALIA APRIL 1977


13ØØ CD DØ Ø1 LOOP:CALL CI      ;FETCH CHAR FROM TERMNL
13Ø3 5F                MOV E,A   ;COPY INTO E
13Ø4 4F                MOV C,A   ;AND INTO C
13Ø5 CD E3 Ø1          CALL CO   ;NOW ECHO
13Ø8 7B                MOV A,E   ;RESTORE IN A
13Ø9 E6 7F             ANI 7FH   ;STRIP OFF PARITY
13ØB EE ØD             XRI ØDH   ;WAS IT A CR?
13ØD C2 ØØ 13          JNZ LOOP  ;NO -- CONTINUE
131Ø ØE ØA             MVI C,ØAH ;YES -- SUPPLY LF
1312 CD E3 Ø1          CALL CO
1315 21 26 13          LXI H,2613 ;SET UP H&L AS ANSWER POINTER
1318 4E           ANSR:MOV C,M   ;LOAD ANSWER CHAR INTO C
1319 79                MOV A,C   ;COPY CHAR INTO A
131A FE ØØ             CPI ØØH   ;CHECK IF ZERO
131C CA ØØ 13          JZ LOOP   ;YES -- END OF ANSWER SO RETURN
131F CD E3 Ø1          CALL CO   ;NO -- SEND TO TERMINAL
1322 23                INX H     ;INCREMENT POINTER
1323 C3 18 13          JMP ANSR  ;AND CONTINUE
1326              ; ANSWER BUFFER BEGINS HERE
1326 47 4F 2Ø
1329 41 57 41
132C 59 2C 2Ø
132F 49 27 4D
1332 2Ø 42 55
1335 53 59 21
1338 ØD ØA ØØ                    ;ANSWER MUST END WITH A ZERO BYTE
```

Here is a short novelty program which the author wrote to run on the SDK-80 kit. Note that in 3-byte memory reference instructions, the 8080A expects the LESS significant address byte first.

output of the EA video terminal is an opto-coupler circuit designed to switch the nominal 20mA current. While it can cope with a moderately higher level, a current of 40mA causes its voltage drop to rise quite significantly, making it seem like the keyboard is continuously in the "space" condition.

As it happens the trouble is easily fixed. I simply changed the value of a resistor in the SDK-80 interfacing circuit (R19) from 430 ohms/1W to 1k/1W. This reduced the loop current to 20mA, and all was well.

Those with video terminals having opto-coupler interfacing like the EA design may also need to make this modification to the SDK-80.

Once this modification was done, I was able to use the SDK-80 to try out some simple 8080 programs and get them going. This proved to be quite straightforward using the various monitor commands, which are very helpful apart from the lack of dump and load facilities.

Among other things, I tried writing some programs which call the utility subroutines in the monitor ROM. There are a number of useful routines available, although the kit manual suggests that the user can only use the two terminal drivers. It also suggests a rather strange indirect way of calling these, but I found that it was possible to call both the drivers and a number of other subroutines directly.

To illustrate this, I am reproducing here a listing of a simple novelty program which echoes input from the terminal, and replies with a curt "GO AWAY, I'M BUSY!" when the user terminates a line with a carriage return. This program uses the two terminal driver routines CI and CO, calling them directly via their addresses in ROM (01D0 and 01E3).

It would probably be possible to make the program shorter by using the monitor subroutine ECHO instead of CO. However, even as it stands it should give you an idea of the power and flexibility of the 8080A instruction set, at least for this type of application. You could try writing a shorter version yourself, as an exercise.

Summing up, I found the SDK-80 kit fairly easy to assemble, and easy to get going apart from the minor complication caused by the high keyboard loop current. The resident monitor program provides a powerful set of commands, including a rarely-found block move function, although there seems to be no provision for dumping to and loading from tape.

All in all, though, SDK-80 seems a businesslike little system, and one which would make a good introduction to the 8080 microprocessor. It seems quite good value for money at the quoted price of $315 plus tax.

SDK-80 kits and other Intel 8080 systems are available from Warburton Franki Pty Ltd, who have offices in each state.