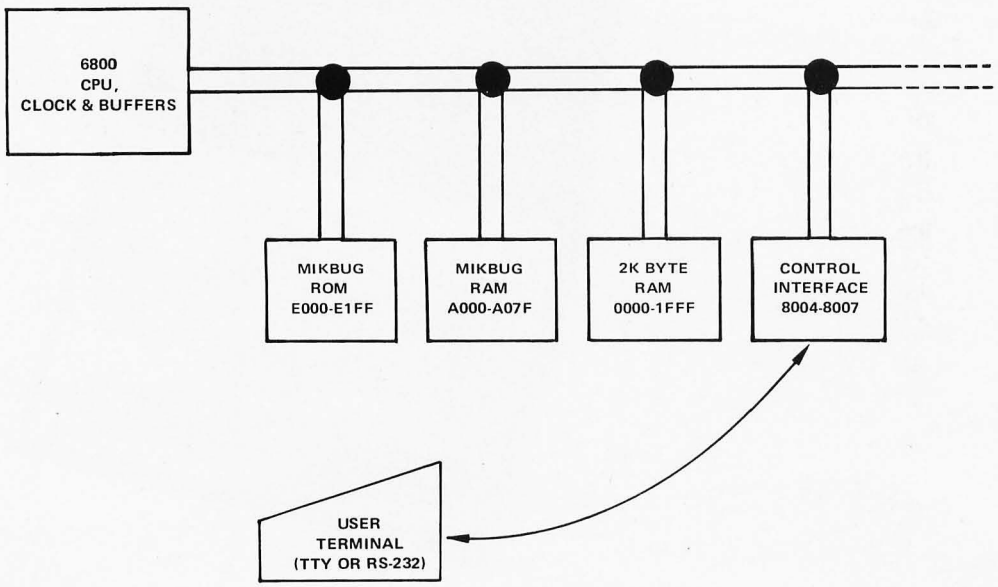
Build A 6800 System With This Kit

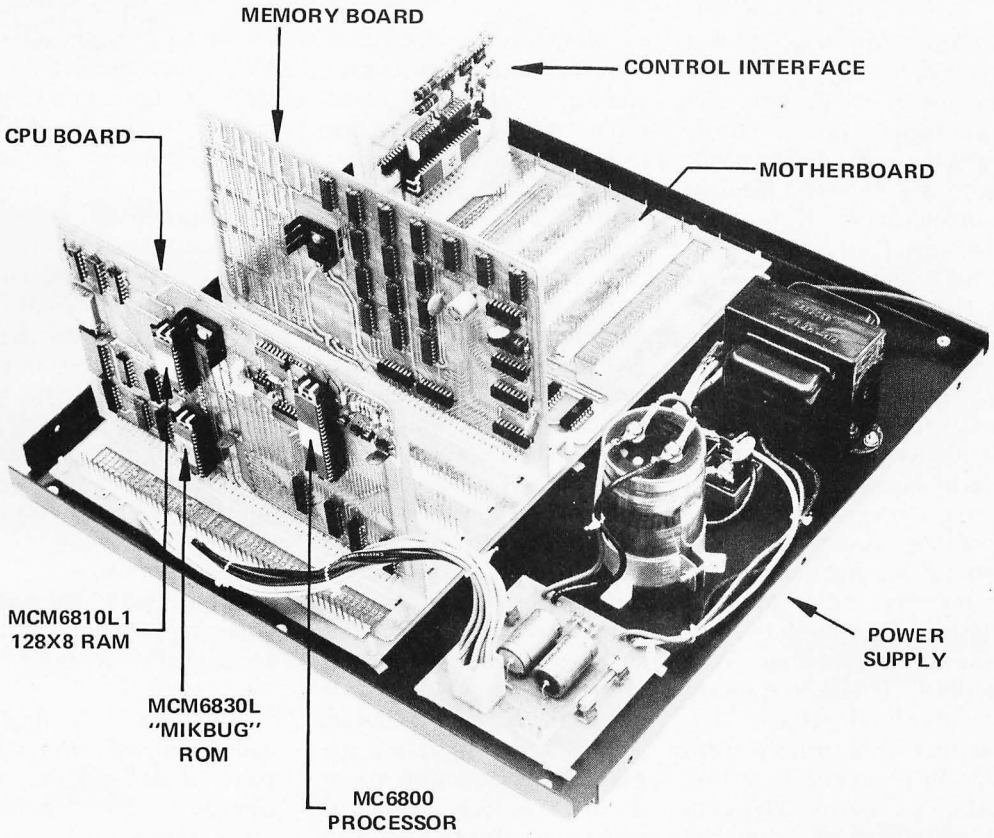
by
Gary Kay
Southwest Technical Products Corp.
219 W. Rhapsody
San Antonio TX 78216

If you are one of the many people getting ready to purchase one of the reasonably priced microprocessor system kits on the market today, you might ask yourself whether or not you will be able to start entering programs once you get it all put together. Of course you can always load programs and data through the front panel programmer's console, but most individuals aware of the front panel's slow speed and difficult readability prefer to use a Teletype or low cost video terminal such as the TV Typewriter II (February 1975, *Radio Electronics*) for data and program input/output. This is all well and good except that in order to attach a terminal, you'll have to purchase an interface for your computer if it is not supplied with the basic system. In fact you will generally need a separate interface for each I/O (input/output) device connected to your computer. This can run your system investment up considerably since such interfaces typically cost between $75 and $150 each, and there are more surprises yet to come.

So now you've got your computer, with interface, attached to your terminal; you're ready to sit down, power up and start typing in your program, right? Well, not quite. You see, in order to be able to use the terminal for either entering programs or getting data in and out of the computer you must have a program resident or loaded into memory telling the processor how and what to do. Without this software (program), you can pound on the keyboard all you want and the computer won't do anything. Computers are no smarter than their programming lets them be and without programming they're not very smart at all. How do you get this software into memory? Well, you could load it in from paper or cassette tape, that is if you have a paper tape reader or cassette tape interface (another sizable investment) or you could enter it directly from the programmer's console. The problem here is two fold. Software to give the terminal reasonable system control will probably be around 500 words in length. This is far too long to enter from the programmer's console especially when you consider it has to be re-entered every time the system is powered up or after a wayward program overwrites any of its allocated area of memory. The second problem is that few if any of the manufacturers supply a listing, paper tape or cassette tape of such a program to begin with. Their terminal control routines are contained within editor/assembler and higher

Fig. 1. Block diagram of the SWTPC 6800 system. The address allocations of the elements of the system are noted inside the blocks.





Details of the SWTPC 6800 System. This photo illustrates what you see when you remove the cover of a typical SWTPC computer system. This is an assembly of the parts which come in the MP-68 kit.

level language packages which not only must be loaded from some kind of tape reader, but require from 4,096 to 8,192 words of memory to operate. And you thought the interfaces were expensive, just check the prices on 8,192 words of memory. Many of the systems now on the market are supplied with an amount of memory with the basic unit which is considerably less than what might actually be needed for useful programming.

So what's the alternative? Well, the system presented in this article has been designed to eliminate the aforementioned problems and allow the user to have a powerful and fully functional system at a minimum cost (see Fig. 1). The entire system is built around the Motorola MC6800 microprocessor and its family of support devices. The computer itself is being made available in kit form including the chassis, cover, power supply and all circuit boards, parts and hardware necessary to build a Motorola 6800 based microprocessor including a 1,024 word ROM (read only memory) stored operating system with 128-word scratch pad memory, serial interface baud rate generator, serial interface, and 2,048 words of memory for $450. This article gives a description of the microprocessor and mother board. A future article will describe the power supply, memory and interface boards.

The Microprocessor/System Board (MP-A)

The Microprocessor/System Board (coded MP-A) is the primary logic board for the system. It is a 5 1/2" x 9" double sided plated-through hole circuit board containing the 6800 microprocessor chip, the 6830 ROM which stores the mini-operating system and the 6810, 128-word scratch pad random access memory (RAM) needed by the ROM. There is a crystal controlled processor clock driver and baud rate generator providing serial interface baud rates of 110, 150, 300, 600 and 1200 baud for all but the terminal control interface which is operable at 110 or 300 baud. Also provided is a power up/manual reset circuit which restarts the ROM stored mini-operating system whenever activated. Full I/O buffering is provided for the 16 address lines and eight bidirectional data lines with these and other connections made to the rest of the system through the mother board via a 50-pin connector. Power for the board is derived from a +5 volt regulator fed from the system's unregulated 7 volt, 10 Amp power supply. Average current consumption for the board is 0.8 Amps.

The mini-operating system stored in the 6830 ROM on this board has got to be one of the most outstanding features of this system. It is through this Motorola written software package called "MIKBUG" that the user can 1) enter program or data into memory from either the terminal's keyboard or tape (where applicable), 2) jump to and execute a program loaded in memory, 3) list programs or data stored in memory, on the terminal or tape (where applicable), 4) examine and/or change the contents of the internal CPU registers, 5) examine and/or change the contents of specified memory locations. These operations are performed using a 20 mA current loop Teletype or an RS-232C compatible serial ASCII terminal.

This ROM mini-operating system does not have to be loaded from tape and it cannot be overwritten. It is always there at your fingertips – just pressing the RESET button or simply powering the system up automatically restarts this firmware (ROM stored software). When activated, this system responds with a

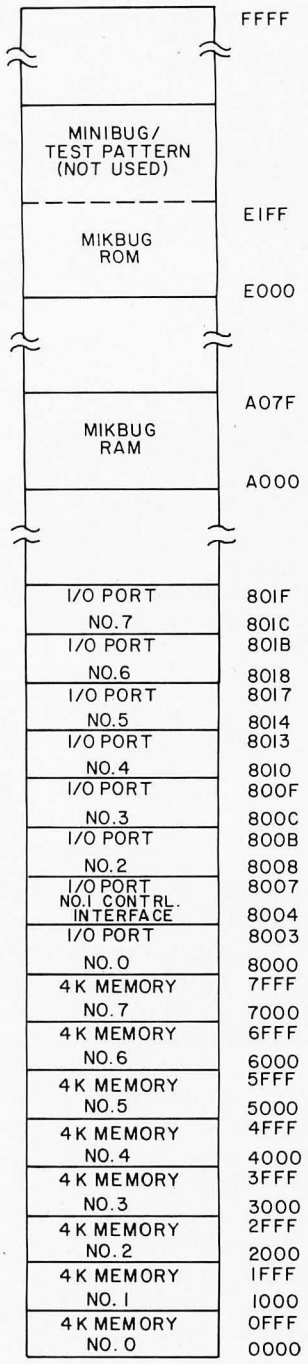


Fig. 2. SWTPC 6800 Microprocessor System memory map. The 64K address space of a 6800 CPU is divided up into the segments shown here. The first 32K locations are available for user read-write memory. The second 32K is devoted to I/O port assignments and the requirements of the MIKBUG program supplied by Motorola.

carriage return, line feed and then prints a * on the terminal at which time you may enter various single character control commands such as M for memory examine/change, L for load from tape, P for punch or list, R for examine registers or G for go to and execute a loaded program. A program debug routine can also be implemented by using the software interrupt (SWI) instruction as a "breakpoint" which forces a jump from your program to the operating system to allow you to examine the contents of memory and/or the CPU registers. All data entered or displayed through the terminal is in convenient hexadecimal (base 16) notation. This means you can type in a command to load address location $A000_{16}$ with $9E_{16}$ instead of setting 24 console switches to an address of 1010 0000 0000 0000 with data of 1001 1110 as must be done with the conventional programmer's console. Since the operating system is stored in ROM, it consumes no user RAM memory, in fact, it actually gives the user a little extra. There is a 128-word scratch pad memory utilized by the operating system for storing various addresses and data, but there are more than 54 locations within this 6810 RAM memory which are totally unassigned plus a 46-word deep push-down stack. All of this memory is in addition to the 2,048 words (expandable to 4,096 words) contained on the standard memory board.

Since the terminal and mini-operating system provide the user with complete system control, there is no need for the conventional programmer's console. Take note also that once system control is turned over to your program, the control terminal is totally available for your program input/output. In fact, since the character input/output subroutines are already stored within the operating system ROM, they can be used by your programs simply by loading or storing the characters to be handled in the proper register and executing a jump to subroutine (JSR).

The Motorola MC6800 microprocessor chip is the element around which this entire system is built. It is an 8-bit parallel processor with eight bidirectional data lines and 16 address lines giving it an addressing capability of up to 65,536 words. There is no distinction between memory and I/O addressing on this system, therefore, all input/output data transfers are handled just as are the memory transfers. This means the I/O interfaces must have their own allocated memory addresses where neither ROM or RAM memory may be located. This may at first seem to be a disadvantage until you realize that all memory handling instructions are usable for the interface data handling as well, thus eliminating the need for special data I/O instructions. The memory assignments for this system have to be made as shown in Fig. 2. User RAM may be located anywhere in the lower 32K (0000_{16} to 8000_{16}) addresses with the upper 32K addresses reserved for the operating system ROM, RAM and interface boards.

There are six registers internal to the MC6800 microprocessor element which consist of the program counter, stack pointer, index register, accumulator A, accumulator B and condition code register. The stack pointer is a 16-bit register used to store the address of the push-down stack which is located in RAM memory external to the MC6800 microprocessor element. The push-down stack itself is used to store the program counter and/or processor data during branch to subroutine (BSR), jump to subroutine (JSR), push (PHS) or interrupt routines. The index register is a 16-bit register generally used as an address pointer for many processor instructions.

There are 72 basic instructions for the 6800 microprocessor system (Fig. 3) with most of the 72 utilizing several of the seven possible addressing modes: Accumulator, implied, relative, direct, immediate, extended and indexed.

- *Accumulator* — In accumulator addressing, either accumulator A or accumulator B must be specified.
- *Implied* — In implied addressing the instruction code itself specifies the operand (stack pointer, index register, etc.).
- *Relative* — Relative addressing is used for the branch instructions and indicates the value contained in the word of memory immediately following the instruction code added to the program counter +2 with the result then loaded back into the program counter. Positive data (bit 7 = 0) generates forward jumps up to 129 words from the branch instruction while negative data (bit 7 = 1) generates backward jumps up to 125 words from the branch instruction.
- *Direct* — In direct addressing, the value contained in the word of memory immediately following instruction code is an actual memory address within the first 256 words of memory (0000_{16} to 00FF) which contains the operand of the instruction. This mode typically saves one CPU cycle of execution when compared to *extended* addressing.
- *Immediate* — In immediate addressing, the

value contained in the word, or in some cases two words of memory, immediately following the instruction code is the operand of the instruction.

• *Extended* – In extended addressing, the two words of memory immediately following the instruction code contain the address of the memory location which contains the operand of the instruction.

• *Indexed* – In indexed addressing, the value contained in the word of memory, immediately following the instruction code, is temporarily added to the contents of the index register generating a new address where the operand of the instruction is located. The jump is positive only, going from 0 to 255 words and the actual contents of the index register are not changed.

Also provided on the main processor board is an MC14411 baud rate generator which uses an external 1.8432 MHz crystal and internal oscillator and divide chain to generate serial interface clocks for baud rates of 110, 150, 300, 600 and 1200 baud. Also derived from this circuit is the 921.6 kHz clock used by the MC6800 microprocessor element. It is first, however, fed into a gating circuit generating two non-overlapping, 50% duty cycle, complementary clock signals ϕ_1 and ϕ_2.

Mother Board (MP-B)

The Mother Board (coded MP-B) is a 9" x 14" double sided, plated-through hole circuit board onto which all of the various processor boards are plugged. Provisions have been made for one Microprocessor/System Board, up to four 4,096 word random access memory boards plus two unused slots. This allows the system to be expanded to 16,384 words of memory. For those demanding even more memory, the 50-line system information bus may be paralleled onto another mother board with separate power supply expanding the system to a maximum of 32,768 words of random access memory.

The Mother Board also provides the line buffering and address decoding for up to eight interface boards. Although one of the eight must be the serial terminal, control interface, the other seven may be any combination of parallel or serial interfaces the user may choose to have. For those demanding even more interfacing capability, the 50-line system information bus may be paralleled onto another mother board with separate power supply expanding the interfacing capability to one terminal, control interface plus any combination of up to 15 serial or parallel interfaces.

The following is a brief description of each of the 50 lines on the system information bus:

The A0 – A15 lines carry address bits 0 through 15 respectively, forming a 16-bit address which is used to define either a memory location or interface address.

The BUS AVAILABLE line goes high acknowledging a processor halt, meaning the processor has stopped and that the system information bus is available for external control.

The $\overline{D0}$ – $\overline{D7}$ lines carry inverted data bits 0 through 7 respectively, forming 8-bit data words which are exchanged between the various boards within the system.

The GND line is the system's common power supply ground point.

Fig. 3. The 6800 microprocessor's instruction set. This is a list of the mnemonics available. A more complete explanation of the basic operations of the processor is found in Motorola's programming manual for the 6800 which is part of the SWTPC documentation package.

ABA	ADD ACCUMULATORS
ADC	ADD WITH CARRY
ADD	ADD
AND	LOGICAL AND
ASL	ARITHMETIC SHIFT LEFT
ASR	ARITHMETIC SHIFT RIGHT
BCC	BRANCH IF CARRY CLEAR
BCS	BRANCH IF CARRY SET
BEQ	BRANCH IF EQUAL TO ZERO
BGE	BRANCH IF GREATER OR EQUAL ZERO
BGT	BRANCH IF GREATER THAN ZERO
BHI	BRANCH IF HIGHER
BIT	BIT TEST
BLE	BRANCH IF LESS OR EQUAL
BLS	BRANCH IF LOWER OR SAME
BLT	BRANCH IF LESS THAN ZERO
BMI	BRANCH IF MINUS
BNE	BRANCH IF NOT EQUAL TO ZERO
BPL	BRANCH IF PLUS
BRA	BRANCH ALWAYS
BSR	BRANCH TO SUBROUTINE
BVC	BRANCH IF OVERFLOW CLEAR
BVS	BRANCH IF OVERFLOW SET
CBA	COMPARE ACCUMULATORS
CLC	CLEAR CARRY
CLI	CLEAR INTERRUPT MASK
CLR	CLEAR
CLV	CLEAR OVERFLOW
CMP	COMPARE
COM	COMPLEMENT
CPX	COMPARE INDEX REGISTER
DAA	DECIMAL ADJUST
DEC	DECREMENT
DES	DECREMENT STACK POINTER
DEX	DECREMENT INDEX REGISTER
EOR	EXCLUSIVE OR
INC	INCREMENT
INS	INCREMENT STACK POINTER
INX	INCREMENT INDEX REGISTER
JMP	JUMP
JSR	JUMP TO SUBROUTINE
LDA	LOAD ACCUMULATOR
LDS	LOAD STACK POINTER
LDX	LOAD INDEX REGISTER
LSR	LOGICAL SHIFT RIGHT
NEG	NEGATE
NOP	NO OPERATION
ORA	INCLUSIVE OR ACCUMULATOR
PSH	PUSH DATA
PUL	PULL DATA
ROL	ROTATE LEFT
ROR	ROTATE RIGHT
RTI	RETURN FROM INTERRUPT
RTS	RETURN FROM SUBROUTINE
SBA	SUBTRACT ACCUMULATORS
SBC	SUBTRACT WITH CARRY
SEC	SET CARRY
SEI	SET INTERRUPT MASK
SEV	SET OVERFLOW
STA	STORE ACCUMULATOR
STS	STORE STACK REGISTER
STX	STORE INDEX REGISTER
SUB	SUBTRACT
SWI	SOFTWARE INTERRUPT
TAB	TRANSFER ACCUMULATORS
TAP	TRANSFER ACCUMULATORS TO CONDITION CODE REG.
TBA	TRANSFER ACCUMULATORS
TPA	TRANSFER CONDITION CODE REG. TO ACCUMULATOR
TST	TEST
TSX	TRANSFER STACK POINTER TO INDEX REGISTER
TXS	TRANSFER INDEX REGISTER TO STACK POINTER
WAI	WAIT FOR INTERRUPT

The normally high $\overline{HALT}$ line when brought low halts the processor and frees the system information bus for external control.

The INDEX line is an unused one and is provided so the pin on each of the male connectors may be cut with the corresponding female connector pins plugged, preventing the circuit boards from being plugged on incorrrectly.

The $\overline{IRQ}$ is the maskable, single level interrupt request line feeding the processor board. If not inhibited by software it will when momentarily given a TTL zero level signal, force the processor into a push-down stack store routine followed by a program jump to a user selected address stored in the operating system RAM.

The M. RESET line, when momentarily grounded manually, indirectly resets the registers internal to the processor and interfaces, and loads the ROM stored mini-operating system. This line is normally grounded by depressing the RESET button on the system's front panel.

The $\overline{NMI}$ is the non-maskable, single level interrupt line feeding the processor board. When momentarily given a TTL zero level it forces the processor into a push-down stack store routine, followed by a program jump to a user selected address stored in the operating system RAM. The $\overline{NMI}$ is not maskable thus cannot be inhibited by the programmer through software.

$\overline{\phi_2}$ is one of the two complementary system clock outputs and is used to signal that valid data is on the data lines $\overline{D0}$ – $\overline{D7}$ when low.

ϕ_1 is the non-overlapping clock complement of $\overline{\phi_2}$.

The $\overline{RESET}$ line when low resets the registers internal to the processor and interfaces, and loads the ROM stored mini-operating system. This line is activated by one shot on the Microprocessor/System board when the system is first powered up or when M. RESET line is momentarily grounded.

The R/$\overline{W}$ line establishes the direction of data flow on the eight data lines, $\overline{D0}$ – $\overline{D7}$. It is high for a read from memory or interface and is low for a write to memory or interface.

$\overline{VMA}$ is the valid memory address line which goes low to confirm that valid memory address data is being presented on the 16 address lines, A0 – A15.

The UD1 and UD2 are user defined lines and have not been assigned a function.

The –12 and +12 points are lines to which an isolated ground –12 @ 200 mA and +12 @ 200 mA power supply should be connected. The voltages are necessary for generating the currents required by 20 mA current loop Teletype equipment on the serial interfaces.

The 7 – 8 VDC UNREG point is the line to which a +7 to 8 volt dc @ 10 Amp unregulated power supply should be attached. This voltage is then regulated down to +5 V dc by independent regulators on the various boards within the system.

The five 110b, 150b, 300b, 600b, 1200b lines carry 1758.8, 2400, 4800, 9600 and 19200 Hz clocks required by the serial interfaces for 110, 150, 300, 600 and 1200 baud communication.

Also attached to the 50-line system information bus are the interface decode and driver circuits. A considerable cost savings is made here by providing the address decoding and information bus buffering for all of the interfaces right on the mother board instead of providing it on each of the interface boards individually. Since each of the parallel interfaces require four address locations and the serial two, four addresses are provided for each of the interface positions. They are assigned as shown in the memory map, Fig. 2. Interface position 1 (8004 – 8007) is reserved for the terminal control interface. The signals carried on the interface information bus are almost identical to those on the system bus. UD3 and UD4 are here again User Defined data lines and RS0 and RS1 are Register select lines which are identical to address lines A0 and A1 respectively. Power for the address decode and buffer circuits on the mother board is provided by a separate on board regulator with a current consumption of typically 0.4 Amp. ■

(More SWTPC 6800 data is coming in BYTE.)

Once you've assembled and checked out the operation of your MP-68 kit, the result will be a product which looks like this. Note the complete absence of most of the usual control panel functions you might expect. This is achieved by using a serial communications device such as a Teletype or an RS-232C compatible terminal as the "front panel."