MICROMINT INC.
917 Midway
Woodmere, N.Y. 11598

Z8 Basic Computer/Controller

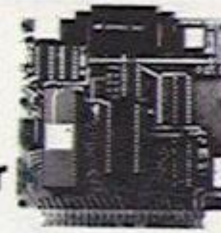


TABLE OF CONTENTS

The Z8 BASIC COMPUTER/CONTROLLER was designed by Steve Ciarcia and presented in "Ciarcia's Circuit Cellar" in BYTE Magazine. The two part article entitled "Build a Z8-Based Control Computer with BASIC", appeared in the July and August 1981 issues.

THE MICROMINT INC. Z8 BASIC COMPUTER/CONTROLLER BOARD

Owners Manual and Assembly Instructions

The MICROMINT INC Z8 BASIC COMPUTER/CONTROLLER board is a milestone in microcomputer price-performance. It is cheap enough to dedicate to a specific application, intelligent enough to be programmed directly in a high level language, and efficient enough to be battery operated if required. Depending upon your needs, it can be utilized as an inexpensive tiny BASIC computer for general interest or dedicated to specialized tasks such as security control, energy management, solar system monitoring, or intelligent peripheral control.

The entire computer is 4" by 4 1/2" and includes a tiny BASIC interpreter, 4K bytes of program memory, one RS-232 serial port and two parallel ports, plus a variety of other features (A condensed functional specification is outlined in figure 1). Using a Z8 microcomputer chip and Z6132 4K X 8 RAM, the Z8 BASIC COMPUTER/CONTROLLER board is completely self contained and optimized for use as a dedicated controller.

For example, to program it for a dedicated application you would merely attach a terminal to the RS-232 connector, power up the system, and type in a BASIC program using familiar commands such as GOTO, IF, GOSUB, LET, etc. Executing the program is simply a matter of typing RUN. For higher speed applications or just to experiment with the Z8 instruction set, a USR command facilitates calls to machine language subroutines.

Once a program has been written and tested with the aid of the terminal, the finished program can be transferred to an EPROM via a memory dump program and the terminal removed. Next, the 28 pin Z6132 RAM is removed from its socket and either a 2716 (2K X 8) or 2732 (4K X 8) EPROM (depending upon the length of the program) plugged into the lower 24 pins. THE EPROM AND Z6132 RAM ARE PIN COMPATIBLE. PERMANENT PROGRAM STORAGE IS SIMPLY A MATTER OF PLUGGING AN EPROM INTO THE Z6132 RAM SOCKET. When the Z8 board is powered up, the stored program is immediately executed. More on all these features later.

Z 8 BASIC MICROCOMPUTER BOARD (figure 1)

Technical Specifications:

Processor.............	Zilog Z8 8-bit microcomputer with RAM, ROM and I/O in a single package. Z8671 includes a 2K tiny BASIC/DEBUG resident interpreter in ROM, 144 bytes RAM, and 32 I/O lines. System runs at 7.3728 MHz clock rate. 1.5 usec instruction time. 6 interrupts.
Memory	Onboard 4K byte RAM memory. Uses Z6132 4K quasi-static RAM (pin compatible with 2716 and 2732 EPROMs); 2K BASIC/DEBUG ROM based interpreter; Memory externally expandable to 62K bytes of program memory and 62K bytes of data memory.
Input/Output	Serial Port --- RS-232C compatible and switch selectable to 110, 150, 300, 1200 2400, 4800, 9600 bps. Parallel I/O --- 2 parallel ports: one dedicated input and one bit-programmable as input or output. Programmable interrupt and handshaking lines; All signals are LSTTL compatible. External I/O --- 16 bit address and 8 bit bidirectional data bus brought out to expansion connector.
BASIC Commands	GOTO, GO@, (USR), GOSUB, IF THEN, INPUT, LET, LIST, NEW, REM, RETURN, RUN , STOP, IN, PRINT, PRINT HEX. Integer arithemetic: +,-,/,*,AND BASIC can call machine language sub-routines for increased execution speed; Allows complete memory and register interrogation and modification.
Power Supply .... Requirements	+5 Volts +/- 5% @ 250 mA. +12 Volts +/-10% @ 30 mA. -12 Volts +/-10% @ 30 mA. (The 12 volt supplies are only required for RS-232C operation.)
Dimensions and ...... Connections	4" by 4 1/2" board, dual 22 pin (.156") edge connector. 25 pin RS-232C female D-subminiature connector,4 pole DIP switch data rate selector.
Operating .... Conditions	Temperature: 0-50C (32-122F) Humidity: 10-90% Relative humidity (non condensing)

MICROPROCESSORS VERSUS SINGLE CHIP MICROCOMPUTERS

The central component in this miniature computer is the ZILOG Z8671 (while we have been simply calling it the Z8, this is a generic term referring to a family of devices. The Z8671 is just one of them). Rather than a microprocessor like the Z80, the Z8 is a single chip microcomputer. It contains RAM, ROM, and I/O as well as other standard CPU functions. Microprocessors such as the Z80 or 8080 require support circuitry to build a functional computer. A single chip micro-computer, on the other hand, can function solely on its own with no other chips.

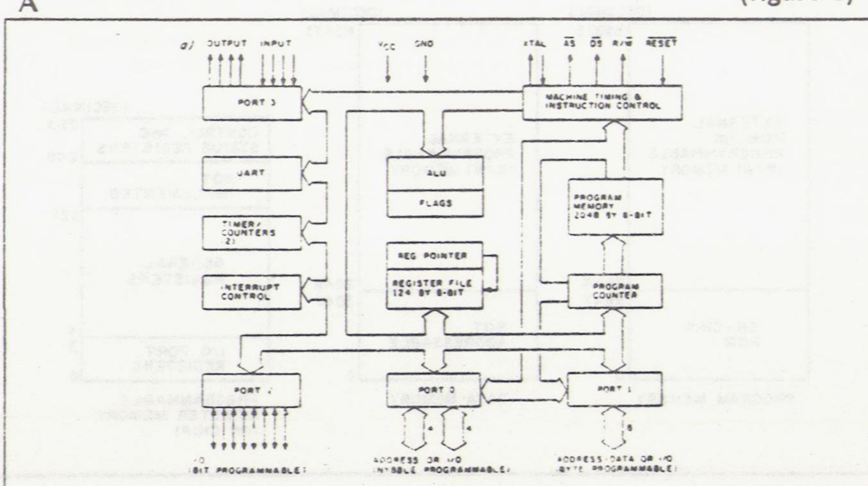
The concept is not new. Single chip microcomputers have been around for quite a while and millions of them are used in electronic games. The Z8, however, raises the capabilities of single chip computers to new heights and provides many powerful features usually found only in general application micro-processors.

Traditionally, single chip microcomputers have been designed for microcontroller applications and optimized for I/O manipulation. On a 40 pin package, often as many as 32 of the pins can be I/O related. A ROM programmed single chip microcomputer used in an electronic chess game might offer a thousand variations of game strategy but it could not be reprogrammed to function as a word processor. The ability to reorient processing functions and reallocate memory has generally been the province of memory intensive microprocessors.

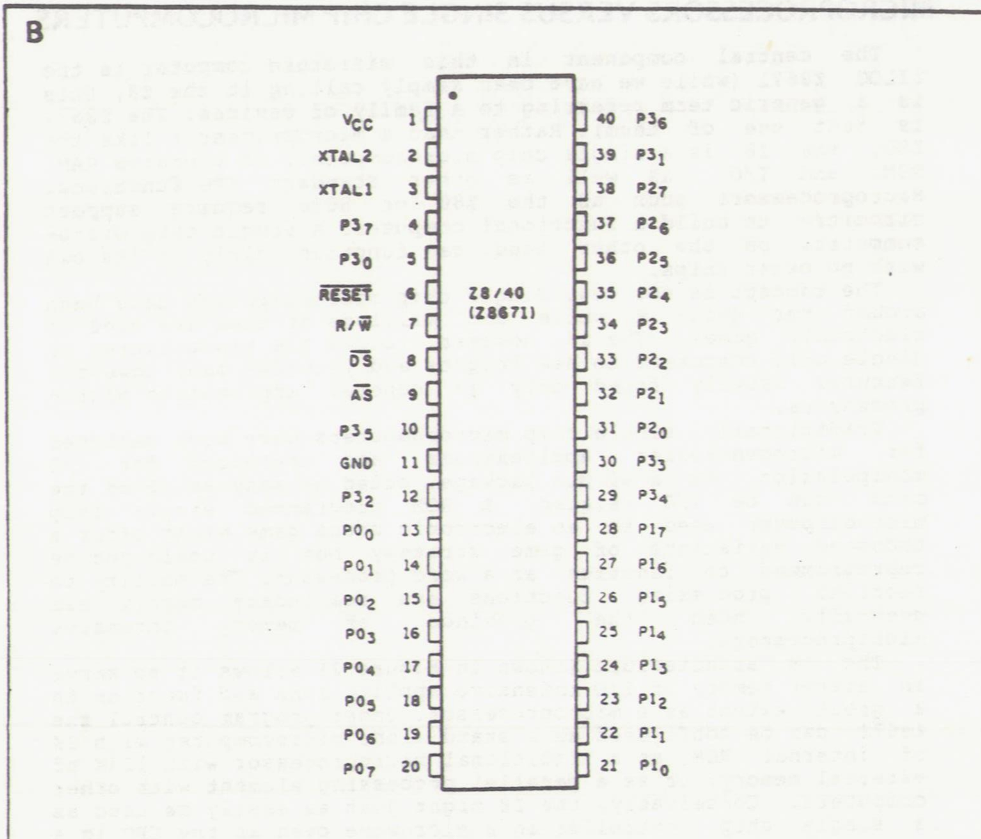
The Z8 architecture (shown in figure 2) allows it to serve in either memory or I/O intensive applications and function to a great extent as a microprocessor. Under program control the Z8671 can be configured as a stand alone microcomputer with 2K of internal ROM, as a traditional microprocessor with 124K of external memory, or as a parallel processing element with other computers. Conceivably, the Z8 might just as easily be used as a single chip controller in a microwave oven as the CPU in a point-of-sale terminal complete with floppy disks.

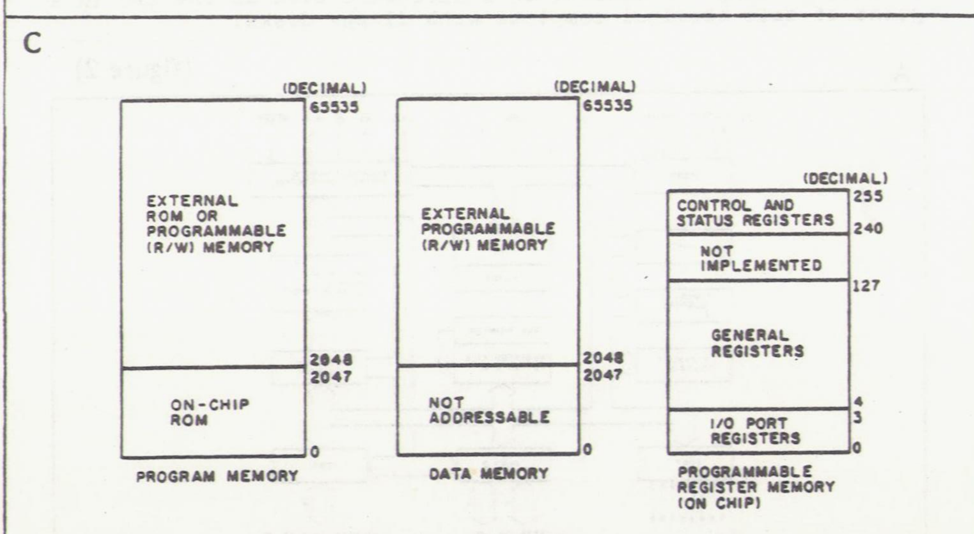
A

(figure 2)



(figure 2)


B
V_{CC} 1
XTAL2 2
XTAL1 3
$P3_7$ 4
$P3_0$ 5
RESET 6
R/W 7
DS 8
AS 9
$P3_5$ 10
GND 11
$P3_2$ 12
$P0_0$ 13
$P0_1$ 14
$P0_2$ 15
$P0_3$ 16
$P0_4$ 17
$P0_5$ 18
$P0_6$ 19
$P0_7$ 20
Z8/40
(Z8671)
40 $P3_6$
39 $P3_1$
38 $P2_7$
37 $P2_6$
36 $P2_5$
35 $P2_4$
34 $P2_3$
33 $P2_2$
32 $P2_1$
31 $P2_0$
30 $P3_3$
29 $P3_4$
28 $P1_7$
27 $P1_6$
26 $P1_5$
25 $P1_4$
24 $P1_3$
23 $P1_2$
22 $P1_1$
21 $P1_0$


C
(DECIMAL)
65535
EXTERNAL ROM OR PROGRAMMABLE (R/W) MEMORY
2048
2047
ON-CHIP ROM
0
PROGRAM MEMORY
(DECIMAL)
65535
EXTERNAL PROGRAMMABLE (R/W) MEMORY
2048
2047
NOT ADDRESSABLE
0
DATA MEMORY
(DECIMAL)
255
CONTROL AND STATUS REGISTERS
240
NOT IMPLEMENTED
127
GENERAL REGISTERS
4
3
I/O PORT REGISTERS
0
PROGRAMMABLE REGISTER MEMORY (ON CHIP)

THE ZILOG Z8671 MICROCOMPUTER

The Z8671 contains 2K bytes of ROM (preprogrammed with a tiny BASIC interpreter), 32 I/O lines and 144 bytes of random access memory. The internal RAM is actually a register file (illustrated in figure 3) composed of 124 general purpose registers (R4-R127), 16 status control registers (R240-R255), and 4 I/O port registers (R0-R3). Any of the general purpose registers can be an accumulator, address pointer, index register, or part of the internal stack (the significance of these registers is explained in detail in the Z8 Technical and BASIC/DEBUG manuals).

The 32 I/O lines are grouped into 4 ports and treated internally as 4 registers. They are software configureable for either input or output and are LSTTL compatible. Port 1 and Port 0 can additionally serve as a multiplexed address/data bus for connection of external memory and peripherals. In traditional nomenclature, Port 1 supplies the D0-D7 data bus and A0-A7 address bus signals. Port 0 supplies the remaining A8-A15 address lines for a total of 16 bits. This allows 62K bytes (plus 2K of ROM) of program memory to be directly addressed. If more memory is required, one bit on Port 3 facilitates the addition of another 62K bytes which is referred to as data memory. In the MICROMINT Z8 BASIC COMPUTER/CONTROLLER, separate data memory is not presently implemented and program and data memory are considered to be the same.

The address and data buses are brought out to a 44 pin expansion connector. These signals are fully LSTTL compatible as mentioned. However, if more than 2 additional LSTTL gates are connected to the address and data strobe lines, additional buffering may be necessary.

The Z8 has 47 instructions, 9 addressing modes and 6 maskable interrupts. Using an 8 MHz crystal most instructions average 1.5 to 2.5 microseconds. In the Z8671 the BASIC/DEBUG interpreter allows machine language programs to be written and executed through subroutine calls in BASIC. This feature greatly enhances the capabilities of a tiny computer and allows connection to high speed peripherals.

The final area of concern is communication. The Z8 contains a full duplex Universal Asychronous Receiver/Transmitter (UART) and two counter/timers with prescalers. One of the counters divides a 7.3728 MHz clock to one of 7 standard data rates. With the Z8671 these rates range between 110 and 9600 bps and are switch or software selectable.

Serial data is received through bit 0 of Port 3 and transmitted from bit 7 of Port 3. While the Z8 can be set for either even or odd parity, the Z8671 is preset for 8 data bits, no parity and 2 stop bits. Received data must always have 8 data bits, at least 1 stop bit, and no parity.

(figure 3)

LOCATION		IDENTIFIERS
255	STACK POINTER (BITS 7-0)	SPL
254	STACK POINTER (BITS 15-8)	SPH
253	REGISTER POINTER	RP
252	PROGRAM CONTROL FLAGS	FLAGS
251	INTERRUPT MASK REGISTER	IMR
250	INTERRUPT REQUEST REGISTER	IRQ
249	INTERRUPT PRIORITY REGISTER	IPR
248	PORTS 0-1 MODE	P01M
247	PORT 3 MODE	P3M
246	PORT 2 MODE	P2M
245	TO PRESCALER	PRE0
244	TIMER/COUNTER 0	T0
243	T1 PRESCALER	PRE1
242	TIMER/COUNTER 1	T1
241	TIMER MODE	TMR
240	SERIAL I/O	SIO
	NOT IMPLEMENTED	
127	GENERAL PURPOSE REGISTERS	
4		
3	PORT 3	P3
2	PORT 2	P2
1	PORT 1	P1
0	PORT 0	P0

THE Z6132 QUASI-STATIC RANDOM ACCESS MEMORY

The Z6132 outlined in Figure 4 is a 5 volt 32K bit (4K X 8) dynamic RAM. It uses single transistor dynamic storage cells but performs and controls its own refresh. This eliminates the need for external refresh circuitry. The result is a combination of the convenience of static RAM and the low power consumption of dynamic memory. The refresh operation is completely transparent to the user. The entire 4K bytes of memory is supplied in a single 28 pin chip which typically draws about 30 milliamps.

An additional benefit in using this RAM is that it is pin compatible with standard 2716 (2K X 8) and 2732 (4K X 8) EPROMs. This feature is extremely beneficial when configuring this Z8 board for use as a dedicated controller. The Z8671 is programmed to add an extra clock cycle when addressing extended memory. This allows use of 450 nsec EPROMs (the Z6132 supplied with the Z8 BASIC COMPUTER/CONTROLLER has a 420 nsec access time).

As previously described, the Z6132 can be removed and an EPROM inserted in the lower 24 pins of the same socket. Thus any program written and operating in RAM can be placed in non-volatile EPROM (there are some limitations on the number of subroutine calls and variables allowed since these items must be stored in the Z8 register area instead of external RAM. The limitations are explained in the BASIC/DEBUG manual).

THE Z8 BASIC COMPUTER/CONTROLLER BOARD

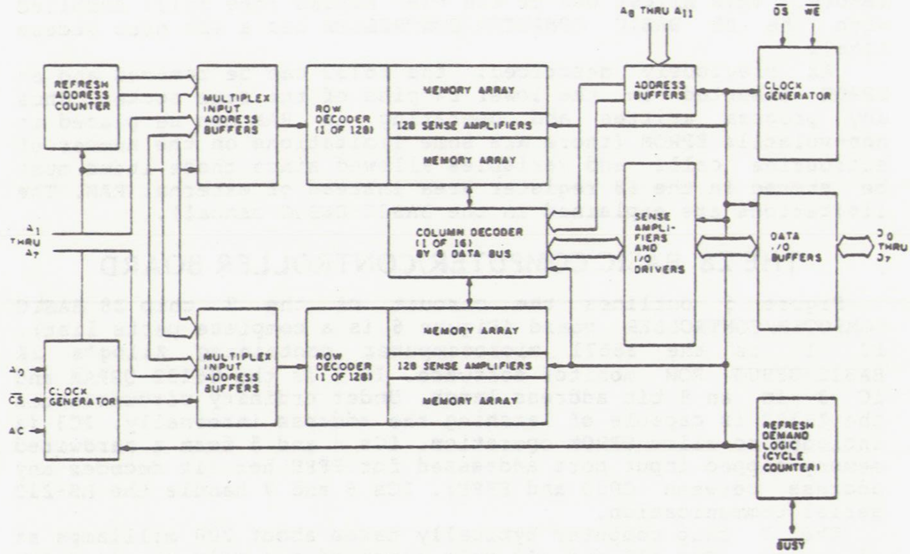
Figure 5 outlines the circuit of the 7 chip Z8 BASIC COMPUTER/CONTROLLER board (Figure 6 is a complete parts list). IC 1 is the Z8671 microcomputer containing Zilog's 2K BASIC/DEBUG ROM monitor software. IC 2 is the Z6132 QSRAM and IC 3 is an 8 bit address latch. Under ordinary circumstances the Z6132 is capable of latching the address internally. IC3 is included to allow EPROM operation. ICs 4 and 5 form a hardwired memory mapped input port addressed for FFFD hex (it decodes any address between C000 and FFFF). ICs 6 and 7 handle the RS-232 serial communication.

The 7 chip computer typically takes about 200 milliamps at +5 volts. The +12 and -12 volt supplies are only required for RS-232 operation. Power required is typically 25 milliamps on each.

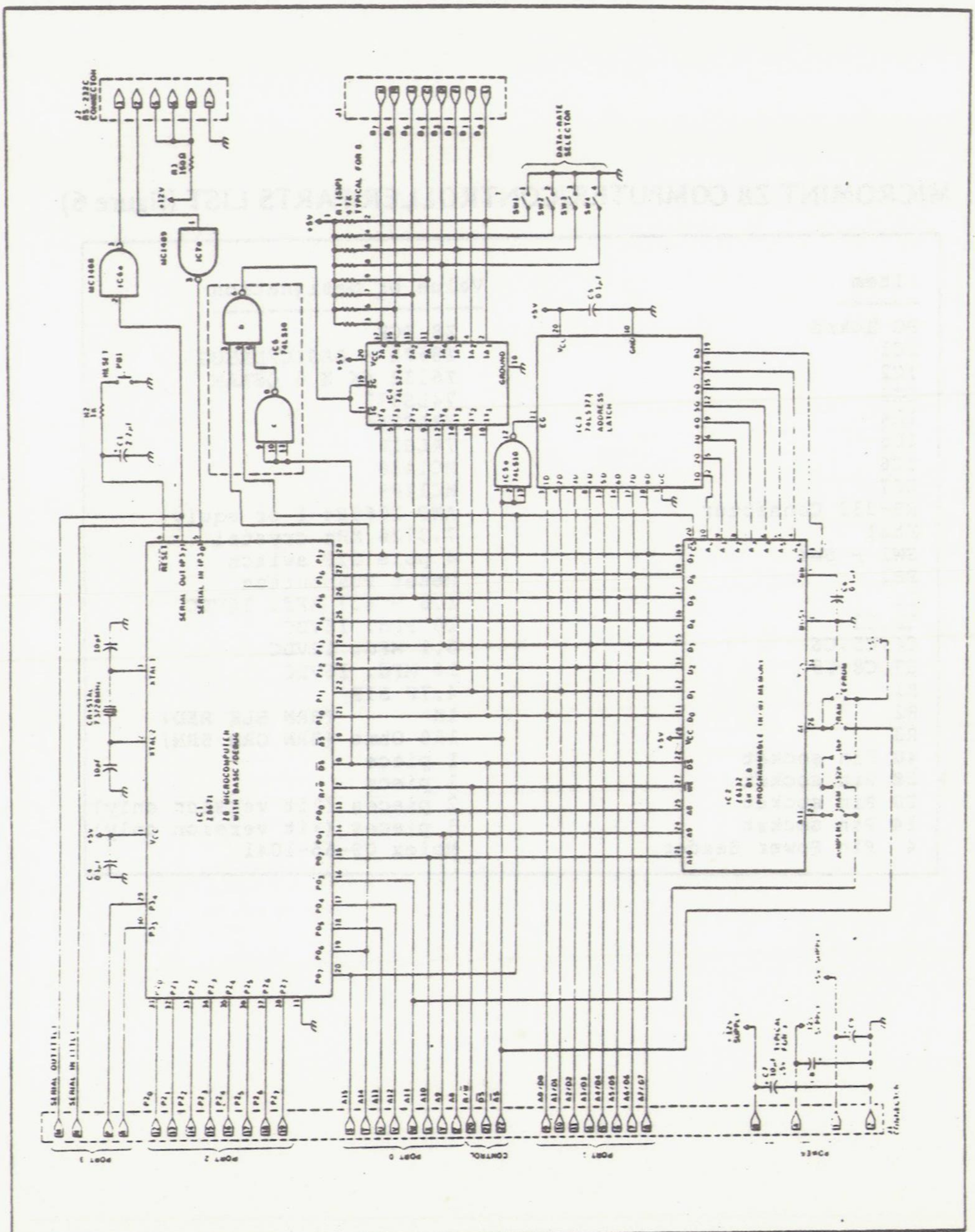
The easiest way to check out the Z8 computer is to attach a terminal to the RS-232 connector and set the data rate selector switch to some convenient rate (8 bit data, no parity and 1 stop bit). For example, to select 1200 baud, SW2 should be on and SW1, SW3 and SW4 are off. After applying power, simply press the RESET pushbutton.

Pressing "RESET" starts the Z8 initialization procedure. One of the things it does is read memory location FFFD hex where the data rate selector switches are located. When it has acquired this information, it sets the appropriate data rate and transmits a colon to the terminal. At this point the Z8 board is completely operational and programs can be typed in BASIC.

(figure 4)


A8 THRU A11
DS
WE
REFRESH ADDRESS COUNTER
MULTIPLEX INPUT ADDRESS BUFFERS
ROW DECODER (1 OF 128)
MEMORY ARRAY
128 SENSE AMPLIFIERS
MEMORY ARRAY
ADDRESS BUFFERS
CLOCK GENERATOR
A1 THRU A7
COLUMN DECODER (1 OF 16) BY 8 DATA BUS
SENSE AMPLI-FIERS AND I/O DRIVERS
DATA I/O BUFFERS
D0 THRU D7
A0
CS
AC
CLOCK GENERATOR
MULTIPLEX INPUT ADDRESS BUFFERS
ROW DECODER (1 OF 128)
MEMORY ARRAY
128 SENSE AMPLIFIERS
MEMORY ARRAY
REFRESH DEMAND LOGIC (CYCLE COUNTER)
BUSY

(figure 5)



MICROMINT Z8 COMPUTER/CONTROLLER PARTS LIST (figure 6)

Item	Value or Designation
PC Board	Z8 PCB
IC1	Z8671 - BASIC/DEBUG
IC2	Z6132 4K X 8 QSRAM
IC3	74LS373
IC4	74LS244
IC5	74LS10
IC6	MC1488
IC7	MC1489
RS-232 Connector	AMP 206584-1 or equiv.
Xtal	7.3728 MHz crystal
SW1 - SW4	4 pole DIP switch
PB1	Reset Pushbutton
C1	1.0 - 4.7 MFd. 16VDC
C2,C3	10 PFd. 16VDC
C4,C5,C6	0.1 MFd. 12VDC
C7,C8,C9	10 MFd. 16VDC
R1	4.7K SIP
R2	1K (BRN BLK RED)
R3	150 Ohms (BRN GRN BRN)
40 Pin socket	1 piece
28 Pin socket	1 piece
20 Pin socket	2 pieces (kit version only)
14 Pin socket	3 pieces (kit version only)
4 Pin Power Header	Molex 09-65-1041

BASIC/DEBUG is a Process Control Oriented BASIC Monitor

Essentially, an integer math subset of Dartmouth BASIC, the Zilog BASIC/DEBUG monitor is specifically designed for process control. BASIC/DEBUG recognizes 15 command words. They include GOTO, GO@, USR, GOSUB, IF THEN, INPUT, IN, LET, LIST, NEW, REM, RUN, RETURN, STOP, PRINT, and PRINT HEX. Standard syntax and mathematical operators are used.

Twenty six variables (A-Z) are supported. Each letter is a variable name. Variables can be used to designate program line numbers. For example, GOSUB B*100 and GOTO A*B*C are valid expressions. It allows examination and modification of any memory location, I/O port, or register. The interpreter processes data in both decimal and hexadecimal notation and accesses machine language code as either a subroutine or a user defined function.

BASIC/DEBUG can directly address the Z8 internal registers and all external memory. Byte references, which use the "@" character with an address, may be used to modify a single register in the CPU, control I/O device, or memory location. For example, @4096 specifies memory location 4096 and @%F6 specifies the PORT 2 mode control register at decimal location 246. To set 45 in memory location 4096 the command is simply, @4096=45 (or @%1000=%2D).

Command abbreviations are standard with most tiny BASICs but this interpreter allows some extremes if you want to limit program space. For example:

```
IF 1>X THEN GOTO 1000      can be abbr.   IF 1>X 1000

PRINT"THE VALUE IS ";S  can be abbr.  "THE VALUE IS ";S

IF X=Y THEN IF Y=Z THEN PRINT "X=Z"
                              can be abbr. IF X=Y IF Y=Z "X=Z"
```

One important difference between standard Dartmouth BASIC and BASIC/DEBUG is that the latter allows variables to define statement numbers and variable storage is not cleared before a program is run. Commands such as GOSUB X or GOTO A*E-Z are all valid. It is also possible to pass values from one program to another. Generally speaking, the BASIC is standard in its approach but features such as those mentioned above serve to extend its capabilities.

The main feature which separates this BASIC from others is the level of documentation supplied with the Z8671. While the 2K source code for BASIC/DEBUG is not available (since it is in ROM it can't be changed anyway), the location of all variables, pointers, stacks, etc., are fixed and their locations are defined and described in detail in the BASIC/DEBUG manual.

The 2K interpreter is extremely powerful. Because it operates so easily on register and memory locations, arrays and blocks of data can be easily manipulated. The Z8 BASIC COMPUTER/CONTROLLER with it's interpretive language, virtually eliminates the need for costly development systems, memory consuming editors, assemblers, and debug programs.

MEMORY ALLOCATION

The Z8 has three kinds of memory: registers, internal ROM, and external ROM or RAM. BASIC/DEBUG assigns address 0 to 255 to the register file (no registers are implemented between 128-239 inclusive). The 144 registers include 4 I/O port registers, 124 general purpose registers, and 16 status and control registers. The 2K byte ROM on the Z8 chip contains the BASIC/DEBUG interpreter. It extends between address 0 through 2047. User memory starts at address 2048 (800 hex). A memory map of the Z8 BASIC COMPUTER/CONTROLLER board is shown in figure 7.

On power up, BASIC/DEBUG sizes RAM, initializes memory pointers, and checks for an auto start-up program. In a RAM system, the top page of memory is then used for the line buffer, variable storage and GOSUB stack. Program execution begins at location 800 hex.

When BASIC/DEBUG tests memory and finds no RAM, the internal registers are used to store the variables, line buffer, and GOSUB stack. There are limits on the depth of the stack and the number of variables which can be used simultaneously but it is a relatively significant amount. In a system with no RAM, automatic program execution begins at 1020 hex. In a system using a 2K EPROM however, there is wrap-around addressing because of the way it is decoded on the address bus. A 2716 inserted in the Z6132 socket will read the same value at address 800 and 1000 hex. Similarly, address 820 and 1020 hex will be the same. Therefore, when using a 2K 2716 EPROM, even though the auto-start is normally at 1020 hex it will automatically execute any program beginning at 820 hex as well. For purposes of this discussion, all EPROM programs are on 2716s and references to 1020 hex also refer to 820 hex.

The 32 bytes between 800 and 820 hex are reserved for jump vectors to user supplied interrupt routines. When none are implemented, FF hex should be inserted in this memory area.

Z8 BASIC COMPUTER/CONTROLLER MEMORY MAP (figure 7)

```
FFFF
FFFD -- Data Rate Switch
        UNDEFINED
C000
-------------------------
BFFF
   USER MEMORY AND I/O
     EXPANSION AREA
8000
-------------------------
7FFF
        UNDEFINED
2000
-------------------------
17FF
    Onboard 4K RAM
       or EPROM
800
-------------------------
7FF
      Z8 BASIC ROM
100
-------------------------
FF
      Z8 Registers
00
```

PROGRAM FORMAT AND STORAGE

Whether or not the program resides in EPROM or RAM the format is the same. Each BASIC statement begins with a line number and ends with a delimiter. If you were to connect a terminal to the RS-232 serial port and type in the following line:

```
100 PRINT "TEST"
```

it would be stored in memory beginning at 800 hex as follows:

```
           100       P    R    I    N    T         "    T
800     00  64  50   52   49   4E   54   20   22   54

         E    S    T    "
80A     45   53   54   22   00   FF   FF
```

The first 2 bytes of any statement are the binary equivalent of the line number (100 decimal equals 64 hex). Next is the ASCII statement followed by a delimiter (00) which indicates the end of the line. If it is also the last statement in the program (in this case the only one) the next 2 bytes will be FF FF which designate line number 65535.

A multi line program further illustrates this format:

```
100 A=5
200 B=6
3005 "A*B=";A*B
```

```
           100       A    =    5         200       B    =
800     00  64  41   3D   35   00   00   C8   42   3D

         6        3005       "    A    *    B    =    "
80A     36   00   0B   BD   22   41   2A   42   3D   22

         ;    A    *    B
814     3B   41   2A   42   00   FF   FF
```

One final example of this is illustrated in Figure 8. Here is a program written to examine itself. Essentially, it is a memory dump routine which lists the contents of memory in hex. As shown, the 15 line program takes 355 bytes and occupies hex 800 to 963.

There is a reason for explaining the internal program format. One of the major features of this computer is its ability to function with programs residing solely in EPROM. Getting the programs into the EPROM can be quite another matter however.

The easiest way to write an application program into EPROM is to serially transmit the contents of the RAM to an EPROM programmer. (Some of you may only have a manual programmer or one with no communication facility. If you are willing to spend the time you can see how easy it is to dump the RAM contents and manually enter the values into the EPROM programmer).

Normally, the list function or terminal dump routine cannot be used because the listing is filled with extraneous spaces and carriage returns. It is possible however to write a program that transmits the contents of memory with no alterations. The data received by the EPROM programmer would be the exact program to load into the EPROM.

The Z8 has the capability of executing machine language programs directly through the USR and GO@ commands. The serial input and serial output subroutines in the BASIC/DEBUG ROM can be executed independently using these commands. The serial input driver is at location 54 hex and the serial output driver is at location 61 hex. Transmitting a single character is simply: GO@ %61,C where C is the value to be transmitted. Receiving a serial character is simply: C=USR (%54) where C equals the received data.

To dump the entire RAM contents to the programmer the following statements should be included at the end of your program (use a GOTO rather than RUN statement to start program execution):

```
1000 X=%800 :REM BEGINNING OF RAM
1010 GO@ %61,@X :REM TRANSMIT CONTENTS OF LOCATION X
1020 X=X+1 :IF X=%1801 THEN STOP
1030 GOTO 1010
```

Execution begins with a GOTO 1000 and ends when the 4K RAM has been dumped. The transmission rate (110 to 9600 bps) is whatever is selected on the data rate selector switches. Conceivably, this technique can also be used to create a cassette storage capability for the Z8. In theory, a 3 or 4 line BASIC program can be entered in high memory (set the memory pointer (R8 and R9) to put the program there) which reads in serial data and loads it in lower memory. Changing the pointers back to 800 hex allows the newly loaded program to be executed. Since the Z8 already has a serial port, any conventional or specialized FSK modem and tape recorder can be used as the cassette interface.

(figure 8)

```
100 PRINT"ENTER START ADDRESS FOR HEX DUMP ";:INPUT X
102 PRINT"THE LIST IS HOW MANY BYTES LONG ";:INPUT C
103 PRINT:PRINT
105 B=X+8 :A=X+C
107 PRINT"ADDRESS                    DATA":PRINT
110 PRINT HEX (X);"      ";
120 GOSUB 300
130 X=X+1
140 IF X=B THEN GOTO 180
150 GOTO 120
180 IF X>=A THEN 250
200 PRINT:PRINT:B=X+8:GOTO 110
250 PRINT:STOP
300 PRINT HEN (@X); :PRINT"    ";
310 RETURN

:RUN
ENTER START ADDRESS FOR HEX DUMP ? 2048
THE LIST IS HOW MANY BYTES LONG ? 5000

ADDRESS                    DATA

800      0     64    50    52    49    4E    54    22

808     45     4E    54    ...........
.
.
.
958      0      1    36    52    45    54    55    52

960     4E      0    FF    FF     0     0     0     0
.
.
```

PARALLEL I/O EXPANSION

The Z8 BASIC COMPUTER/CONTROLLER has two parallel ports in addition to the serial port previously described. One port is a memory mapped input port used to read the baud rate switches (only SW1 thru SW3 are used) and is addressed at FFFD. It is brought out to the expansion connector through pins C, D, E, F, H, J, K, L. Since the baud rate switches are only read once at reset and not needed at all for programs without serial communication, this port is available for general input data. However, because of the way its address is decoded by IC 5, any memory address between C000 and FFFF will access this port. For example, a PRINT @65000 will print the port value. If it is necessary to use the memory space between C000 and FFFF, the onboard baud rate selection circuitry should be disabled and FFFD decoded through an external decoder.

In the standard configuration Port 2 is available as an 8 bit, bit-programmable I/O port. Depending upon the setting in the Port 2 mode register (R246), it can be all input, all output, or any combination. In addition, certain bits of Port 3 can function as hand shaking lines for Port 2 (see the Z8 technical manual mode control register section). These bits are selected through the Port 3 mode control register (R247).

For example, if you wanted to configure Port 2 for output with a data ready strobe, you would set 0 in the Port 2 mode control register and 113 in the Port 3 mode control register. The data ready output and data accepted input lines are on Port 3, bits 1 and 6. These two lines should be jumpered together (pins 39 and 40) to produce the strobe. It can be wired over to pin 3 (spare) on the edge connector for external use.

Transmitting a value through Port 2 would be done using the "@" command. Doing this in BASIC is simply:

```
100 @246=0:@247=113 :REM SET PORT 2 TO BE OUTPUT
110 @2=X :REM X EQUALS THE DATA TO BE TRANSMITTED
```

BAUD RATE SWITCHES

The serial communication rate on the Z8 COMPUTER/CONTROLLER board is selected by reading memory location FFFD on reset. As configured, this memory location is decoded as an input port with 3 data rate selection switches (on input bits 0, 1, and 2) attached. The switch position versus data rate is as follows:

Baud Rate	SW1	SW2	SW3	SW4
150	ON	ON	ON	N/A
not used	OFF	ON	ON	N/A
9600	ON	OFF	ON	N/A
4800	OFF	OFF	ON	N/A
2400	ON	ON	OFF	N/A
1200	OFF	ON	OFF	N/A
110	ON	OFF	OFF	N/A
300	OFF	OFF	OFF	N/A

OFF = Open circuit / logic 1
ON = Closed circuit / logic 0
N/A = Don't care

EPROM and RAM Jumpers

There are 5 jumpers positions silkscreened on the Z8 BASIC COMPUTER/CONTROLLER board to select between 2716/2732 EPROMs and Z6132 RAM use. They are defined as follows (from top to bottom):

RAM -- Connects the A11 address signal to the Z6132 RAM
EPROM -- Connects +5V to pin 24 on EPROMs
16K -- Connects +5V to pin 21 on 2716 type EPROMs
32K -- Connects the A11 address signal to pin 21 on 2732 EPROMs
RAM -- Connects the processor address strobe (AS) to the Z6132

When using the Z6132 4K byte onboard RAM, the two "RAM" jumpers are inserted (these are installed at the factory on production boards). No other jumpers should be installed.

When using a 2716 EPROM (single supply type only!), the "EPROM" and "16K" jumpers should be the only ones connected. With a 2732 type EPROM, the "EPROM" and "32K" jumpers would be installed.

If you are going to be changing back and forth frequently it is a good idea to put switches into these jumper positions or just clip the jumper in the center in such a way that it can be tack soldered together again. Continued soldering of the same spot on the board will lift the printed circuit pads.

NOTE: The two "RAM" jumpers refer only to the use of 28 pin Z6132 4K by 8 RAM memory devices. If you use a 24 pin 2716 compatible RAM such as the Hitachi HM6116 or NEC UPD446 (2K X 8) then the jumpers should be set as if you are using an EPROM. The selection of either the "16K" or "32K" jumper will depend upon the capacity of the particular RAM being used.

ASSEMBLING THE Z8 BASIC COMPUTER/CONTROLLER BOARD

Step 1 --- Insert and solder the IC sockets. Note that there is an indentation or dot designating pin 1. The sockets should line up with the silk screen on the board.

Step 2 --- Insert and solder all resistors and capacitors. Note the polarity of the electrolytic capacitors (C1, C7, C8,and C9) and orient the positive side of the capacitor with the "+" marked on the board. Generally, the negative side of the capacitors are marked with an arrow and a "-" (minus) sign. The opposite side of the capacitor is the positive side.

Step 3 --- Insert and solder R1, R2, R3 and the disc capacitors. R1 is a 10 pin single-in-line (SIP) package. Pin 1 is designated with a dot on the side of the SIP. It should be inserted such that it lines up with the dot on the silk screen. When inserting C2 and C3 carefully bend the leads to fit the 1/4" hole spacing without fracturing the capacitor material.

Step 4 --- Insert and solder the push button, dip switch and crystal (the crystal should lie flat - a dab of glue helps secure it). Note that the numbers on the dip switch should line up with positions on the silk screen.

Step 5 --- Insert and solder the 25 pin D subminiature connector and the 4 pin power plug header (if supplied). With all 25 pins soldered, screws and nuts are not necessary to hold the RS-232 connector. When installing the power plug header, orient the locking side of the plug (the 1/2" plastic lip) to be on the outside edge of the board. This will keep your Z8 power connection compatible with production units should you use the MICROMINT UNIVERSAL POWER SUPPLY for the Z8 at any time.

Step 6 --- Install Jumpers. Insert two pieces of wire in the "RAM" jumper positions. This will enable use of the Z8 with the Z6132 RAM memory.

NOTE: Unless otherwise specified, jumper positions not designated for connection should be left open. These other jumpers are provided to allow various EPROM connection options to the Z8. Refer to the appropriate manual section for this information.

Step 7 --- Carefully insert ICs 3, 4, 5, 6, and 7 into the sockets. Note that the pin 1 position is designated with either an indentation or dot on the chip.

Step 8 --- Attach the power supply and temporarily apply power to the Z8 computer board and measure the three supply voltages. If any of the three voltages (+5, +12, and -12V) is not within operating limits (5% on the 5V, 10% on the + and - 12V), the Z8 board should be inspected for a possible short or improperly inserted IC. Remove the power.

Step 9 -- Insert IC1 and IC2 into their sockets. Be careful with these devices and treat them as you would a CMOS device (avoid static). Note that the chips have an indentation on one edge. This indentation should line up with the silk screen.

Step 10 --- The easiest way to check out the Z8 computer is to attach a terminal to the RS-232 connector and set the data rate selector switch to some convenient rate (8 bit data, no parity and 1 stop bit). For example, to select 1200 baud, SW2 should be on and SW1, SW3, and SW4 are off. After applying power, simply press the RESET pushbutton.

Pressing "RESET" starts the Z8 initialization procedure. One of the things it does is read memory location FFFD hex where the data rate selector switches are located. When it has acquired this information, it sets the appropriate data rate and transmits a colon to the terminal. At this point the Z8 board is completely operational and programs can be entered directly from the terminal in BASIC.

You can check the performance of the Z6132 RAM by entering a multiple line program. The program will not be stored and cannot be listed if the Z6132 isn't working.

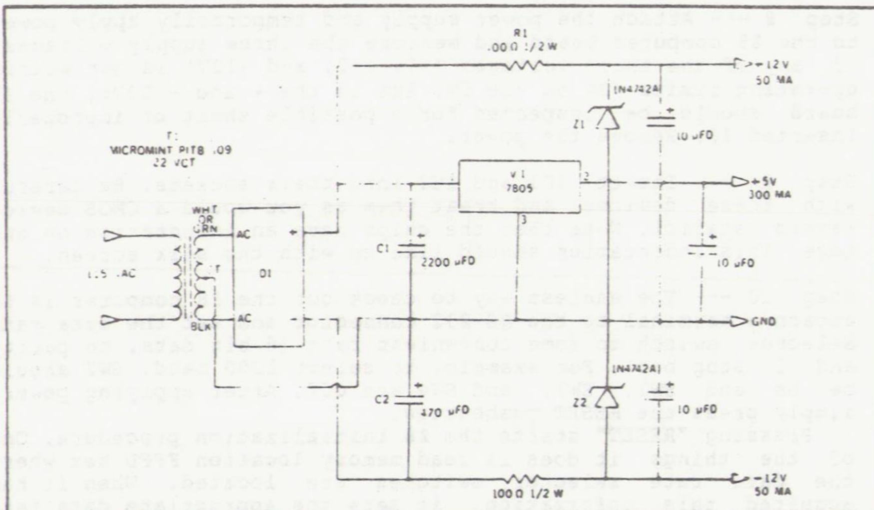
Z8 UNIVERSAL POWER SUPPLY BOARD

The Z8 BASIC COMPUTER/CONTROLLER typically requires 200 milliamps at +5 volts and 25 milliamps each at +12 and -12 volts (the +12V and -12V supplies are only required for RS-232 operation). This power can be supplied either through the 4 pin power header mounted on the lower right corner of the board or through the 44 pin edge connector.

The MICROMINT Inc. manufactures an optionally available triple voltage power supply (the schematic is shown in Figure 9 and the parts are listed in Figure 10). Designated as the Z8 Universal Power Supply, it is designed specifically for the Z8 board. It is 2.1" X 4.5" and has a UL listed power plug type transformer. It connects to the Z8 board through the power plug header and can be mounted directly to the Z8 board on two 1/2" spacers (the mounting holes are located next to C2 and IC4).

The UNIVERSAL POWER SUPPLY can provide 300 milliamps at +5V (+/- 5%) and 50 milliamps each at +12V and -12V (+/- 10%).

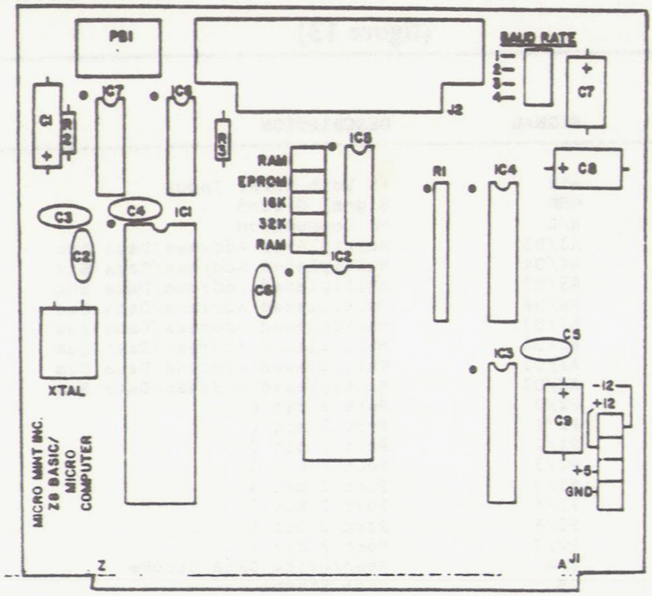
(figure 9)



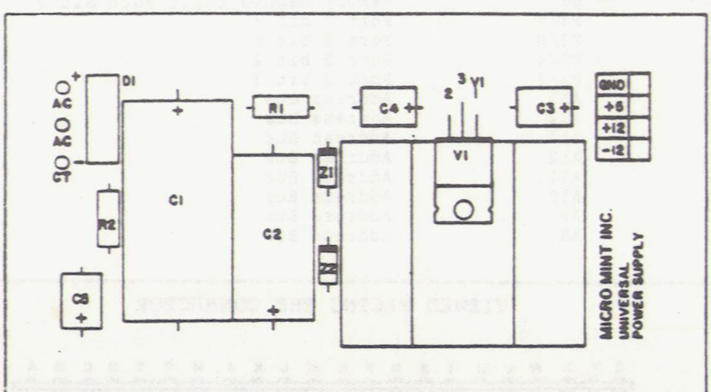
Z 8 UNIVERSAL POWER SUPPLY PARTS LIST (figure 10)

ITEM	PART # or DESCRIPTION
PCB	U/PWR-PCB Circuit Board
T1	Transformer PITB-109 (22VCT @ 300 Ma.)
V1	MCM7805CT or equiv.
R1,R2	100 Ohm 1/2 Watt (BRN BLK BRN)
Z1,Z2	1N4742A Zener Diode
D1	KBP02 Diode Bridge or equiv.
C1	2200 MFd. 25VDC
C2	470 MFd. 25VDC
C3,C4,C5	10 MFd. 25VDC
Heatsink	THM6071B and THM6072B
Hardware	Screw and nut (3/8" 6-32)
Header	4 pin Molex Connector

(figure 11)



(figure 12)



Z8 BASIC COMPUTER/CONTROLLER EXPANSION CONNECTOR J 1

(figure 13)

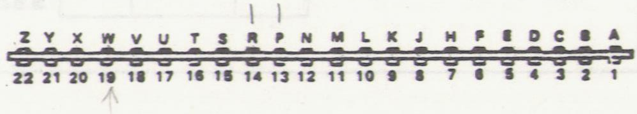
PIN	SIGNAL	DESCRIPTION
1	+5V	+5 Volt Power Input
2	GND	Signal Ground
3	N/C	No Connection
4	A3/D3	Multiplexed Address/Data Bus
5	A4/D4	Multiplexed Address/Data Bus
6	A5/D5	Multiplexed Address/Data Bus
7	A6/D6	Multiplexed Address/Data Bus
8	A7/D7	Multiplexed Address/Data Bus
9	A0/D0	Multiplexed Address/Data Bus
10	A1/D1	Multiplexed Address/Data Bus
11	A2/D2	Multiplexed Address/Data Bus
12	P2/0	Port 2 bit 0
13	P2/1	Port 2 bit 1
14	P2/2	Port 2 bit 2
15	P2/3	Port 2 bit 3
16	P2/4	Port 2 bit 4
17	P2/5	Port 2 bit 5
18	P2/6	Port 2 bit 6
19	P2/7	Port 2 bit 7
20	R/W	Read/Write Data Strobe
21	DS	Data Strobe
22	AS	Address Strobe
A	-12V	-12 Volt Power Input
B	+12V	+12 Volt Power Input
C	B4	Memory Mapped Input Port Bit 4
D	B3	Memory Mapped Input Port Bit 3
E	B5	Memory Mapped Input Port Bit 5
F	B2	Memory Mapped Input Port Bit 2
H	B6	Memory Mapped Input Port Bit 6
J	B1	Memory Mapped Input Port Bit 1
K	B7	Memory Mapped Input Port Bit 7
L	B0	Memory Mapped Input Port Bit 0
M	P3/7	Port 3 bit 7
N	P3/0	Port 3 bit 0
P	P3/4	Port 3 bit 4
R	P3/3	Port 3 bit 3
S	A15	Address Bus
T	A14	Address Bus
U	A13	Address Bus
V	A12	Address Bus
W	A11	Address Bus
X	A10	Address Bus
Y	A9	Address Bus
Z	A8	Address Bus

12,13 USED BY EXPANSION BD FOR CASSETTE (OUT 12, IN 13)

RS232 (OUT, IN)

GIVE UP BIT 3 P3/3 FOR CON. PIN.

VIEWED FACING THE CONNECTOR



ADDENDUM TO THE Z8601 PRODUCT SPECIFICATION FOR THE Z8671 (MAY 1981)

GENERAL DESCRIPTION

The Z8671 is a preprogrammed version of the Z8601, a 2K ROM version of Zilogs single-chip microcomputer family. For a general description of the Z8601 refer to the Z8601 Product Specification (Zilog document number #00-2037-A). The Z8601 is configured by the Zilog Basic/Debug interpreter in the following way;

I/O Port Configuration

- o Port 0 - provides the upper eight bits of address for accessing external memory
- o Port 1 - provides the multiplexed address/data lines for accessing external memory
- o Port 2 - not used by Zilog Basic/Debug; eight pins available to the user
- o Port 3 - pins 0 and 7 are used for serial I/O; six pins available to the user

RAM Search

- o the Z8671 sizes external RAM automatically upon power-up/reset

Counter/Timers

- o T0 - used for baud rate control
- o T1 - available to the user

Interrupts

- o interrupts are enabled for serial communication
- o three external sources for interrupts available to the user
- o one internal source for interrupt available to the user
- o all interrupts are vectored to external memory starting at location 1000 hex

Memory Requirements

- o No external RAM is needed for the Z8671
- o more than 90 registers available to the user in a system with external RAM. Variables are stored externally in the upper page of RAM. See Basic/Debug Manual, Table 6-4 (Memory Map)
- o more than 10 registers available to the user in a system without external RAM. Variables are stored in the internal registers. See Basic/Debug Manual, Table 6-3 (Register Map

for No-RAM System)

- o the Z8671 uses extended memory cycle operation which extends the required memory access time to 570 nsec. maximum (8MHz. XTAL)
- o external memory expandable to 124K bytes

Automatic Execution of ROM Programs

- o a program will execute automatically upon power-up/reset if:
  1) program is stored in ROM
  2) program begins at 1020 hex
  3) the line number at 1020 hex is between 1 and 254 inclusive
- o autoexecution is important for stand-alone control applications

Terminal Interface

- o the Z8671 provides for serial communication
- o only an RS232 driver/receiver pair is needed to use a terminal
- o eight standard baud rates from 110 to 19200 are selectable with external switches
- o 7.3728 MHz crystal is used to drive the on-chip oscillator

Assembly Language Program Execution

- o Zilog Basic/Debug can call assembly language subroutines with a GO@ <address> instruction and will return to the Basic program after a RET instruction in the assembly language program
- o Zilog Basic/Debug can call assembly language functions and return a value with a VAR=USR(<address>, ARGUMENT1, ARGUMENT2) instruction. The return value is returned in the VAR expression.
- o Z8 Family assembly language has 47 powerful instructions combined with eleven addressing modes

Basic Command Set

- o list
- o new
- o run

Basic Statement Set

- o go@
- o goto
- o gosub
- o if/then
- o in
- o input

o let
o print
o rem
o return
o stop

Basic Function Set

o and
o usr

Summary

The Z8671 is an example of "software on silicon" that Zilog has designed to minimize the design turn around time of Z8 Family based applications. A more complete description of the Zilog Basic/Debug/Debug program may be found in the Z8 Basic/Debug Software Reference Manual (Zilog document #03-3149A).

Addendum to the Z8 Basic/Debug Manual
For General Use With The Z8671
(May 1981)

Introduction

The Z8 Basic/Debug Software Reference Manual (Zilog document #03-3149A) was originally intended as a companion to Zilogs Z8 SBC board and as such contains several statements that do not apply to the Z8671 in general This addendum lists the references and gives the general case for the Z8671.

General Case Modifications

The Preface refers to the Z8 CPU board in all three paragraphs; these can be ignored. The Z8 CPU Board Manual that is referenced is not in print. The terms Z8 CPU Board and Z8 SBC Board will be used interchangeably in this addeneum.

Section 6: The Memory Environment, is the next section that needs some modification. The first paragraph states that the Z8 system has three kinds of memory: registers, internal ROM, and external RAM or ROM. The Z8671 can munipulate data in three spaces: registers, program memory (both internal, the interpreter resides internally, and external, Basic statements are stored externally) and external data memory (accessable with assembly langauge instructions). Both external RAM and ROM can be used and Basic will automatically determine the upper and lower limits of RAM.

Paragraph three in Section 6.1 (Memory Structure): The manual states that the Z8 CPU board will not recognize the DM- (Data Memory) signal. The Z8671, however, can be configured to provide this signal so that 62K bytes of external data memory can be accessed with assembly language statements (these statements can be called by Basic as a subroutine).

Section 6.2 (Initalization and Automatic Start-Up): The manual informs us that socket U8 on the CPU board is configured to start at memory location 1000 hex. Actually, the Z8671 automatically sizes external RAM, and external memory can reside anywhere from 800 hex to FFFF hex. The Z8671 will automatically execute a program stored in ROM that starts at location 1020 hex. If autoexecution is used, the best place to start ROM is at 1000 hex.

Table 6-4 (Memory Map for RAM System): The Manual gives the address for socket U7, the address for socket U8, and shows where to put ROM in the Z8 CPU board for autoexecution. The address 1020 hex is an absolute location that the Z8 goes to for autoexecution, however, direct references to the Z8 CPU board can be ignored.

Section 7.2 (Interrupts): The Manual shows how interrutps are routed by the Z8671. The first paragraph contains a direct reference to the Z8 CPU board that can be ignored.

Baud Rate Switch Settings: The baud rate settings listed in Appendix B of the manual assume that the baud rate switches go through inverters before the Z8671 reads them. The values that the Z8671 actually expects to read and decode for baud rate selection are:

baud rate	value read
110	110
150	000
300	111
1200	101
2400	100
4800	011
9600	010
19200	001

Conclusion

The Z8 Basic/Debug Software Reference Manual is useful as a reference guide for using the Z8671. It makes references to the Z8 CPU board which either do not apply or are unnecessary restrictions when considering the ZS671 standing alone. Most references can be safely ignored. The statments which sould be modified for the general case have been noted here.

Factory manufactured and tested Z8 BASIC MICROCOMPUTER boards carry a 60 day warrantee including parts and labor. No credit will be given for boards which show damage through neglect or user modification. Any unit returned for repair after the warantee period must be shipped prepaid and insured. There is a minimum inspection fee for boards not under warantee. Under no circumstances is any hardware to be returned to the MICROMINT without prior authorization. The MICROMINT will assume no liabilities for unauthorized returns.

The MICROMINT makes no warantee on user assembled Z8 systems, power supplies, or blank boards.

The MICROMINT reserves the right to change any specification at any time.

MICROMINT INC., 917 Midway, Woodmere, N.Y. 11598,