

---

# Chapter 16

---

## PC-Compatible Bus Connectors

The left rear corner of the SK68K board holds up to six 62-pin card-edge connectors like those of an IBM PC/XT or clone computer. These connectors can hold most PC- or XT-compatible plug-in I/O cards (although cards intended for AT-type slots will not work.)

### 16-1. Discussion

SK68K software supports the monochrome or CGA color video boards, and the WDXT-GEN (and similar) hard disk controller, but it is fairly straightforward to write software for other cards as well. It does not support some of the other popular PC-type cards such as floppy disk controllers, serial and parallel I/O cards, clock/calendar boards, or multi-function boards which combine several of the above, for the simple reason that all of these options are already contained on the main SK68K board itself and so there is no need for these extra boards.

Although there are many hard disk controllers available for PCs and their clones, SK\*DOS - the SK68K disk operating system - currently supports only the Western Digital WDXT-GEN controller, and its older versions such as the WD1002-WX1 version controller. This is because PC-compatible hard disk controllers contain a ROM which contains the 8088 code to operate them. The ROM is customized to fit the particular hardware configuration of the controller. On a true PC or clone, MS-DOS or PC-DOS simply calls the ROM to do the work without having to concern itself with the actual hardware.

But that ROM is in 8088 code; the 68000 in the SK68K can *read* that ROM, but it cannot *understand* it - 8088 machine language has no meaning to the 68000. Thus the disk software in SK\*DOS has to do all the work of handling the actual controller hardware, and has to be customized to work with the

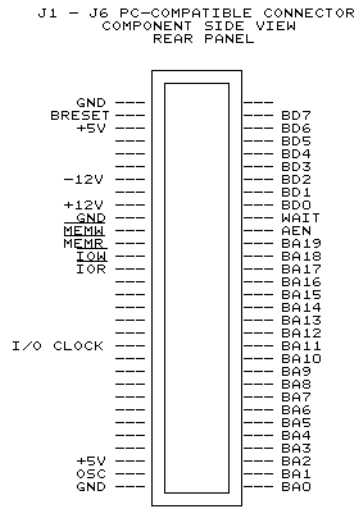


Fig. 16-1. PC-compatible expansion connector.

particular circuitry on a given controller. We therefore chose an inexpensive, widely available hard disk controller and standardized on it.

Fig. 16-1 shows the pinout of each of the six PC-compatible connectors. (This is a top view of the connector, as seen from the front of the board). All of the labelled pins are used; the unlabelled pins are not needed in a 68000 system. Some of the pins, such as ground or power, are obvious; the other connections are shown in the next few diagrams.

Most of the pins on the right hand side of the connector are simply buffered address or data lines. Fig. 16-2 shows the buffering of the address lines, and the data buffers are shown on Fig. 16-4. As you can see in Fig. 16-4, the eight-bit data bus (BD0 through BD7) on the expansion connectors comes from the lower eight bits of the 68000's data bus, so each address on the expansion bus becomes an odd address for the 68000. Moreover, we see that the 68000's A1 line becomes BA0 on the expansion bus, A2 becomes BA1, and so on, up to A20 which becomes BA19. This is necessary because the expansion connectors need a BA0 line, but the 68000 does not have an A0 output; hence everything is shifted over one bit.

To understand how expansion slot addresses equate to 68000 addresses, we have to understand the circuit in Fig. 16-3. In a 68000 (or, in fact, most Motorola processors) there is only one set of addresses, which are used for both memory and I/O. The 8080 (and most Intel processors) have two sets of addresses - one for memory, and another just for I/O. Memory addresses

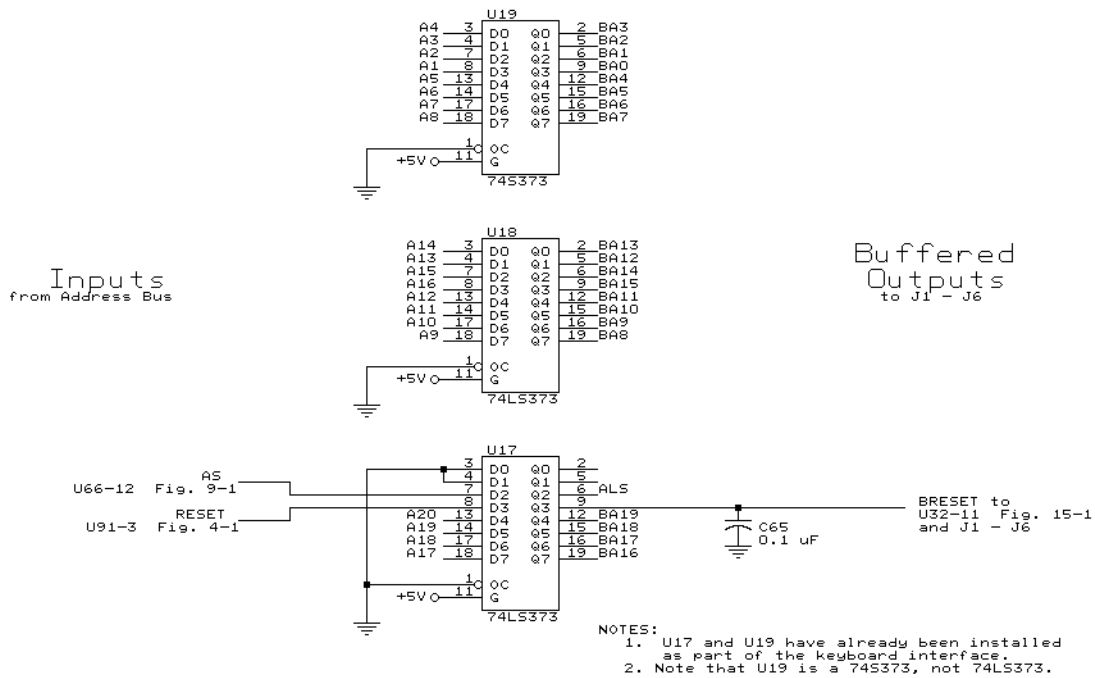


Fig. 16-2. Address bus buffers.

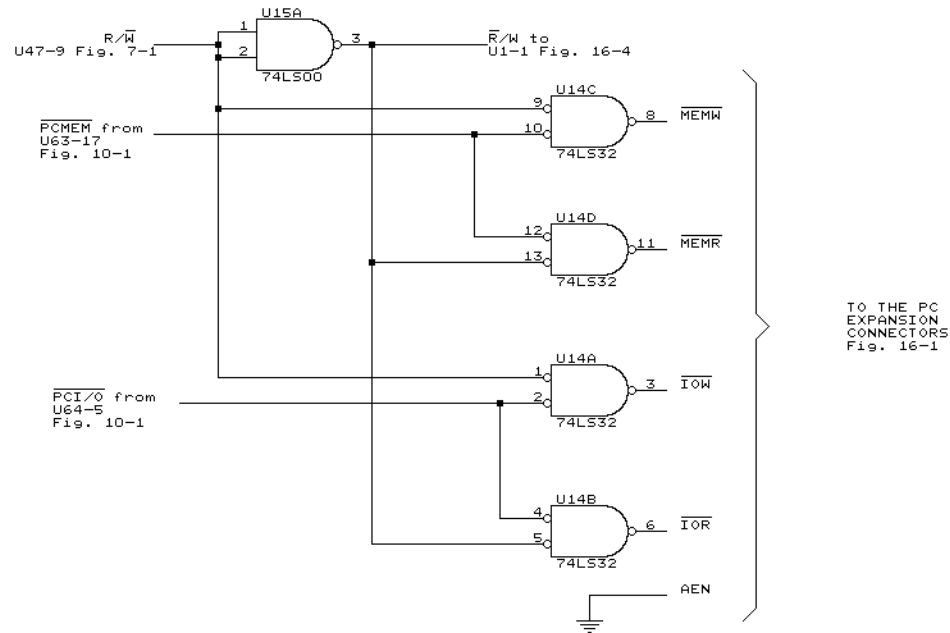


Fig. 16-3. Memory and I/O select logic.

are used for normal reads and writes, while I/O addresses are used only by special IN and OUT instructions. A typical Intel processor then manipulates these addresses with four control lines:

$\overline{\text{MEMW}}$  is asserted (low) to write to memory

$\overline{\text{MEMR}}$  is asserted to read from memory

$\overline{\text{IOW}}$  is asserted to write to an I/O device

$\overline{\text{IOR}}$  is asserted to read from an I/O device.

In a typical IBM PC or clone, these four signals are brought to the bus connectors, and all four are often used by I/O cards. For example, on a monochrome or color video board,  $\overline{\text{IOW}}$  and  $\overline{\text{IOR}}$  are used to control the card, but  $\overline{\text{MEMW}}$  and  $\overline{\text{MEMR}}$  are used to access the video RAM which stores the data to be displayed. On a hard disk controller,  $\overline{\text{IOR}}$  and  $\overline{\text{IOW}}$  again control the hardware, but  $\overline{\text{MEMR}}$  is needed to read the ROM on the board.

Although most PCs or clones have a maximum of 640K of memory, they can actually address 1 megabyte of memory and 64K of I/O addresses, using all 20 address bits for memory and 16 bits for I/O addresses. The megabyte of memory addresses has room for 640K of plain RAM, plus 128K of video memory, 64K of hard disk ROM, and up to 192K of other ROM such as the BIOS and ROM Basic. All of these addresses - both memory and I/O addresses - have to be squeezed into the 68000's single memory address space.

As we saw in Fig 10-1, the address decoder generates a  $\overline{\text{PCMEM}}$  signal for 68000 addresses \$C00000 through \$DFFFFFF, and a  $\overline{\text{PCI/O}}$  signal for addresses \$FA0000 through \$FBFFFF. These two enable signals are used by Fig. 16-3 as follows:

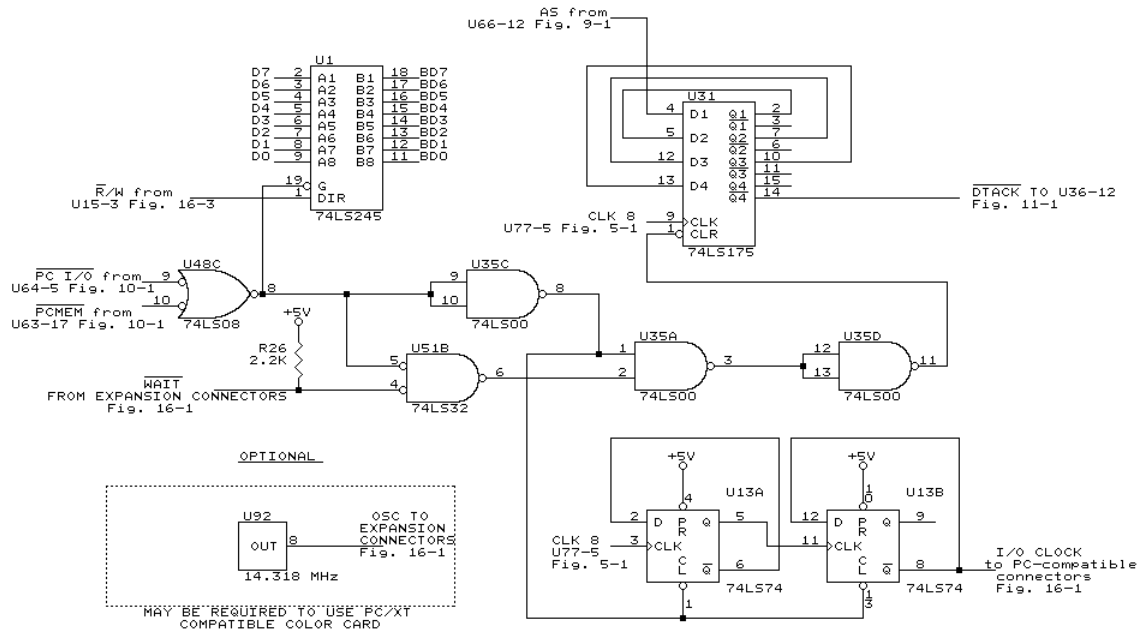


Fig. 16-4. Miscellaneous expansion circuitry.

1. If  $\overline{\text{PCMEM}}$  is low and  $\overline{\text{R/W}}$  is low, then U14c generates a low  $\overline{\text{MEMW}}$  signal.
2. If  $\overline{\text{PCMEM}}$  is low and  $\overline{\text{R/W}}$  is high, then U14d generates a low  $\overline{\text{MEMR}}$  signal.
3. If  $\overline{\text{PCI/O}}$  is low and  $\overline{\text{R/W}}$  is low, then U14a generates a low  $\overline{\text{IOW}}$  signal.
4. If  $\overline{\text{PCI/O}}$  is low and  $\overline{\text{R/W}}$  is high, then U14b generates a low  $\overline{\text{IOR}}$  signal.

Thus when the 68000 reads or writes into memory addresses \$C00000 through \$DFFFFFF, cards plugged into the expansion slots get a memory read or memory write signal; when the 68000 writes into memory addresses \$FA0000 through \$FBFFFF, the cards get an I/O read or I/O write signal.

The result is that cards in the expansion connectors can be written to or read, but the address they get is slightly different from the address the 68000 is accessing. For PC memory addresses, the relationship is this:

PC memory address	68000 memory address
\$00000	\$C00001
\$00001	\$C00003
\$00002	\$C00005
\$00003	\$C00007
:	:
\$FFFFFF	\$DFFFFFF

We can use the formula

$$68000 \text{ memory address} = \$C00001 + 2 \times (\text{PC memory address})$$

to convert one to the other. For example, the top left corner of a monochrome video board is at address \$B0000, which translates to address \$D60001 in the SK68K.

For PC I/O addresses, the relationship is this:

PC I/O address	68000 memory address
\$0000	\$FA0001
\$0001	\$FA0003
\$0002	\$FA0005
\$0003	\$FA0007
:	:
\$FFFF	\$FBFFFF

We can use the formula

$$68000 \text{ memory address} = \$FA0001 + 2 \times (\text{PC I/O address})$$

to convert one to the other. For example, the control port of a monochrome video board is at I/O address \$03B8, which translates to address \$FA0771 in the SK68K.

The final result is that the 1 megabyte of PC RAM translates to 2 megabytes of 68000 addresses, from \$C00000 through \$DFFFFFF, while the 64K of PC I/O addresses translate to 128K of 68000 addresses, from \$FA0000 through \$FBFFFF, but only the odd addresses are used in the SK68K because only the lower half of the 16-bit address bus is connected to the expansion connectors. This means that it is not practical to connect a PC-type memory board to the SK68K since the board could only store odd addresses.

The rest of the circuitry is shown in Fig. 16-4. Whenever either PCI/O or PCMEM goes low, indicating that the 68000 is trying to access the expansion connectors, U48c outputs a low PCEN signal which enables U1, the bidirectional transceiver which buffers the BD0 through BD7 data lines to the expansion connectors; the direction of data flow is determined by the R/W signal (which is just R/W inverted; this signal is therefore low when reading and high when writing.)

PCEN also goes to U35c, which inverts it to a high PCEN signal; this permits U13a and U13b to divide the 8 MHz CLK8 signal by 4 and send the 2 MHz I/O CLOCK to the connectors. This signal is used by some video boards as a clock for an MC6845 video controller chip.

PCEN also goes through U35a and U35d to the clear input of U31. This IC is wired as a shift register to produce a time delay. In normal operation, U31 is held cleared and does nothing. But when PCEN goes high, U31's clear input also goes high and it starts to shift a high (from AS) through the register, one flip-flop for every pulse of CLK8. After four clock pulses, or about 500 nanoseconds, the 4Q output goes low and sends DTACK to U36.

This gives PC-compatible cards 500 nsec to work, but some cards need additional time and send back a low PC/XT WAIT signal. This signal goes through U51b and prevents U31 from timing out until the WAIT signal goes

high. U31 then gives these cards an extra 500 nsec or so after the  $\overline{\text{WAIT}}$  signal returned to high.

Finally, Fig. 16-4 also shows the 14.31818 MHz oscillator; it is needed by some color video boards. (This signal is four times 3.579545 MHz, which is the color burst frequency). Some color boards have their own oscillator; others need one on the motherboard; either way, it is convenient to supply this signal in all cases.

## 16-2. Construction

Now install the following components:

J1 through J6	62-pin card edge connectors (if you install fewer than six, then space them apart)
U18	74LS373 and its socket
U15	74LS00 and its socket
U14	74LS32 and its socket
U1	74LS245 and its socket
U48	74LS08 and its socket
U35	74LS00 and its socket
U51	74LS32 and its socket
U31	74LS175 and its socket
U13	74LS74 and its socket
U92	14.31818 MHz oscillator (soldered directly to the board, and with the pointed corner identifying pin 1 closest to J4)
R26	2.2K 1/4-watt resistor
C1	0.1 $\mu\text{F}$ disk ceramic capacitor

U17 and U19 have already been installed.

Finally, place a short wire jumper from U15 pin 12 to pin 7. Now that we have installed U15, part of that IC is generating a false  $\overline{\text{DTACK}}$  which is upsetting everything else. For now, this wire jumper disables this circuit; we will remove the jumper as soon as we install U16 in a future step.

## 16-3. Testing

If you have a PC-compatible video board and matching monitor, plug it in at this time and turn on the power. If all goes well, on the screen you should now see the message "Please press enter".

Some users have reported problems with a CGA (color graphics) board being unreliable. This is often due to noise on the RESET line at the PC slots, which resets the board when it shouldn't. If you encounter this, place a 0.1  $\mu\text{F}$  disk capacitor right on the CGA board, from pin B2 (left side of connector, second pin from rear - see sheet 4 of your diagram) to ground (which

is on pin B1, about 1/4 inch away). Don't solder to the plated pin - find a solder pad a few tenths of an inch away.

In case of difficulty, follow the general procedures in section 13-4 of this manual. To make the process more understandable, let's first discuss what should happen when all is working correctly, and how the computer decides which I/O devices to use.

When you turn on the power (or short the reset pins at J23), the HUMBUG monitor program in the computer's ROM tries to initialize the input and output ports, makes a list of what options you have installed, and sounds the beep-boop from the speaker. If it detects that a video board is plugged into one of the interface connectors, it will then display a "Please press Enter" in the top left corner of its monitor; it does not, however, display that message on a serial terminal because it doesn't yet know what baud rate to use.

HUMBUG is now monitoring both the serial input and the keyboard connector, waiting for you to press the ENTER key (also called RETURN or CR), so it can determine (a) which keyboard you will be using, and (b) what baud rate you are using if you choose the serial keyboard. Both keyboards can thus be connected, but the first one to get an ENTER will be chosen as the input device. (In order to make sure the correct baud rate is chosen, you may have to press ENTER several times on a serial keyboard.)

After the ENTER is received, HUMBUG displays its sign-on message, the prompt (\*), and a cursor (an underline in this example):

```
HUMBUG (R) Copyright (C) 1986-1991 by Peter A. Stark  
*_
```

Whichever keyboard you use, this message will go to the video board(s), if any. If you use a serial keyboard, then it will also go out the serial port to the terminal (at the same baud rate as the keyboard). Once you get the \* prompt and cursor, you may type in any of 32 HUMBUG commands; we will get to those later. For now, try typing in the command HE to get a Help screen which shows the HUMBUG commands.

