chapter 9

KEYBOARDING/POINTING SUBSYSTEM

The internal keyboard consists of a keyboard matrix with keyboard controller on the system board, an integrated pointing stick (EasyPont IV) and select buttons. The computer and optional Convenience Bases have a connector for an external full-sized keyboard and mouse.

Internal Keyboard

The internal keyboard provides all the functionality of the full-sized keyboard with 12 function keys. The keyboard membrane is automatically identified by the system through the keyboard controller KSO (KSO0 - KSO3) and GPIO18 (KSIN15) scan lines.

The keyboard assembly has four additional keys located above the standard keys that are used as Windows and application logo keys, and as user programmable keys to enable programs and custom or frequently used functions to be activated with a single keystroke. Located on the left side of the computer (if it is facing you) is the audio volume control buttons. Also located above the standard keys is the Suspend mode button, and the power on/off slide switch.

Five LEDs located above the Suspend mode button indicate hard drive access, diskette drive access, Scroll Lock, number lock, and Caps Lock status. Two other indicator LEDs located below the touch pad indicate Power/Suspend and battery charger status. The two pick buttons for the pointing device are positioned below the keyboard.

System status information is provided through a popup screen display when any of the hotkeys (**Esc**, **F1**, **F4** ... **F10**, and **T**) are depressed along with the **Fn** key. The different character sets are selected through Setup.

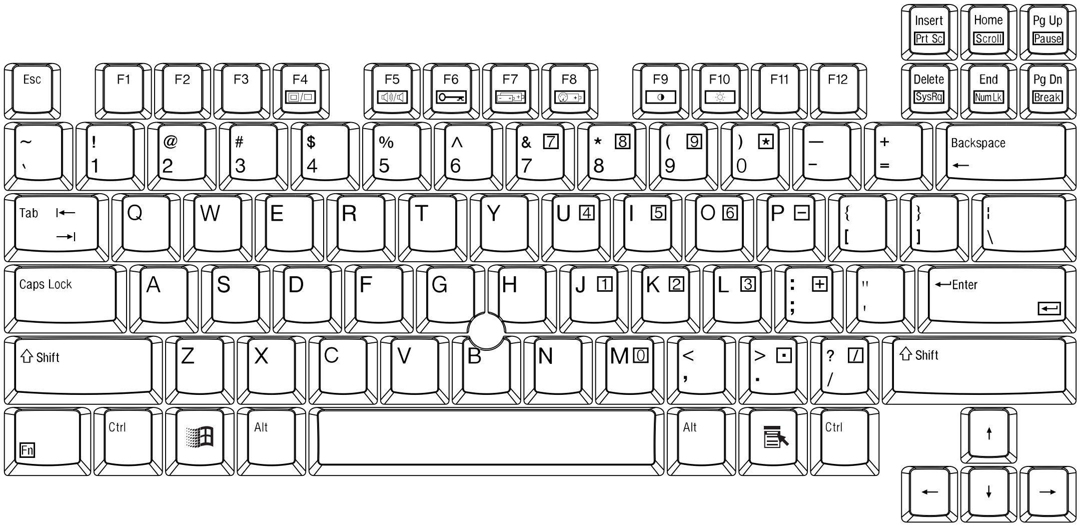

Figure 9-1. U.S. Enhanced Internal Keyboard Layout

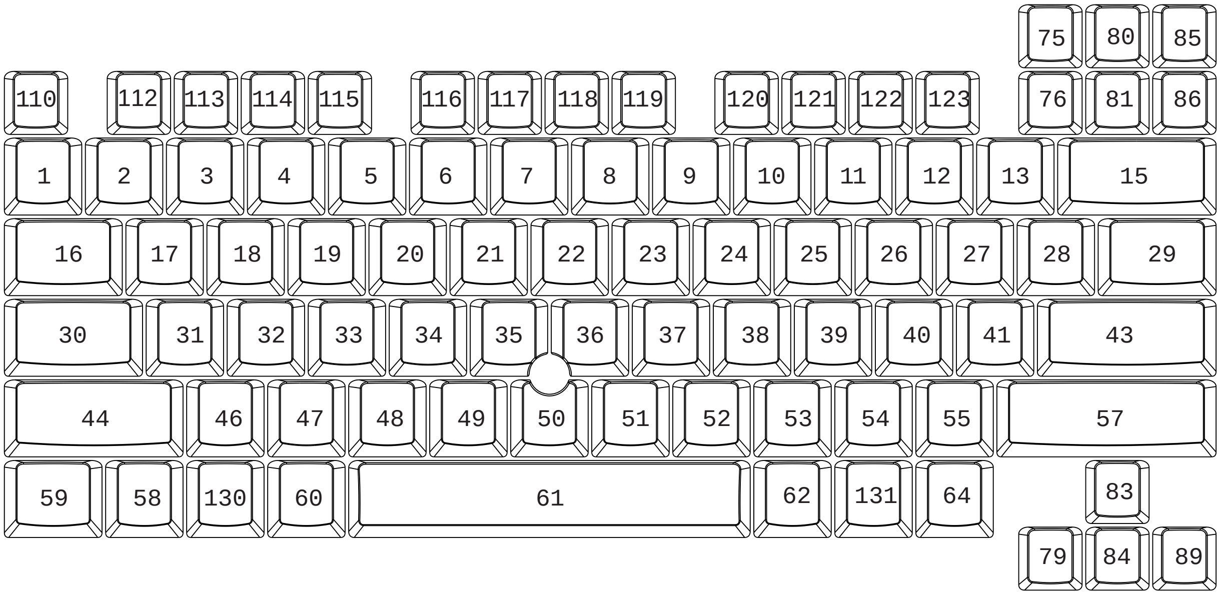

Figure 9-2. Internal Keyboard Assigned Key Location Numbers.

Keyboard Control

Keyboard control is provided by the 8051 keyboard controller in the SMC FDC37C95x FR Ultra I/O (MSIO) on the system board. Keyboard scanning is accomplished by the keyboard controller scan logic which provides 14 scan out (KSO) lines and 8 scan input (KSI) lines forming a 96-key (12 × 8) scan matrix. When all KSO lines are driven active, the scan logic can then generate an interrupt when any KSI line is activated. This type of scanning allows the keyboard controller to wait for a key depression rather than repeatedly polling the entire matrix.

Modifier keys (**Shift**, **Ctrl**, **Alt**, and **Fn**) are positioned in the matrix by diodes to allow three-key combinations and to prevent aliasing.

Functional Description

After power-on occurs, the default (initial state) conditions of the keyboard are:

- Embedded keyboard is enabled if the optional external keyboard is not present.
- Scroll Lock and Caps Lock LEDs are off. The NumLk LED is defined in the system setup.
- Typematic rate = 10 characters/sec (+/- 20%)
- Typematic delay = 500 milliseconds (+/- 20%)

The keyboard controller has both a first-in, first-out (FIFO) buffer and a repeating key function. Both make and break codes are generated when keys are used. Make codes are transmitted when a key is pressed, break codes when it is released. This combination of codes is referred to collectively as the "scan codes" of a key.

Scan codes for up 8 characters (24 bytes) are stored in the FIFO buffer if the system cannot immediately accept them when they are generated,. If two or more keys are pressed simultaneously, the keyboard processes the first scan code detected and stores the others in the buffer in the order in which they are detected. If a key is pressed when the buffer is full, no scan code is generated. An overrun code is stored in the last buffer location, which is reserved for overrun conditions.

Modes of Operation

The default mode at power-up is the Normal mode. Enhanced operation is available in the Normal mode. QWERTY, numeric pad, and separate cursor key functions are available.

Normal Mode

The Normal mode is compatible with a standard 11-bit serial keyboard interface. Each key has a unique make and break code. The make code is transmitted when the key is pressed, and the break code is transmitted after the key is released. The non-standard keyboard scan codes for this mode are shown in Table 9-1 and Table 9-2.

NumLk Key

The **NumLk** key provides access to the embedded numeric keys. The operation of the embedded keypad depends on the state of the **NumLk** key.

Function (Fn) Key

The operation of the **Fn** (function) key is unique; its only purpose is to flag the system keyboard controller.

The following tables and paragraphs describe the combined effects of the keypad, **Fn** key, and **NumLk** key on the keyboard controls.

Embedded Numeric Keypad

The embedded numeric keypad (Figure 9-3) is a set of 16 keys and is enabled and disabled by the **NumLk** key. If **NumLk** is on, the embedded numeric keypad is enabled; if **NumLk** is off, the keypad is disabled. When enabled, the embedded keypad transmits the scan codes of the enhanced numeric keypad. In this mode, the **Shift** key enables the cursor-control functions instead of the numeric functions as it would on an enhanced keyboard.

If **NumLk** is off, the embedded keypad may also be enabled if the **Fn** key is pressed. When the **Fn** key is released, the keyboard returns to the QWERTY mode. If the **Fn** key and **Shift** keys are pressed simultaneously, the embedded keypad functions as cursor-control keys.

If **NumLk** is on, regular (QWERTY) keyboard activity can be enabled temporarily by holding down the **Fn** key. When the **Fn** key is released, the embedded keypad is enabled.

The embedded keypad is disabled if an external keyboard or keypad is present. The keyboard remains enabled.

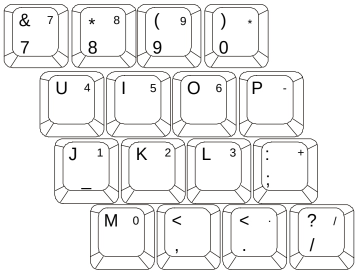

Figure 9-3. *Internal Keyboard Embedded Numeric Keypad Layout*

Cursor- and Screen-Control Functions

The embedded numeric keypad also provides cursor- and screen-control functions. During regular (QWERTY) keyboard activity, these functions can be accessed by pressing the **Fn** and **Shift** keys. Releasing the **Fn** key returns the keypad to normal keyboard operations.

If **NumLk** is on (embedded numeric keypad enabled), pressing the **Shift** key enables these functions. Releasing the **Shift** key returns the keypad to numeric keypad operations.

When cursor- and screen-control functions are active, the *, +, -, and / arithmetic functions are also active. Some applications require the use of the cursor- and screen-control functions in the embedded numeric keypad rather than in the cursor control cluster.

Typematic Function

When a key is held down, it will have a repeating rate of 10 (+/- 1) Hz after a delay of 500 (+/- 100) milliseconds. If multiple keys are depressed, the last depressed key is valid for typematic action. Auto-repeating will stop when the last depressed key is released.

Break codes of keys released are sent during the typematic transmission. The typematic action consists of multiple transmissions of the make code.

Typematic transmission of multibyte make codes require a one millisecond delay between each make code byte. This delay is measured from the falling edge of the stop bit of the first make code to the falling edge of the start bit of the next make code. After a one millisecond delay, the keyboard will check the status of the KBDCLK line and respond with the proper protocol procedure (i.e., RTS). If the computer system has clamped the KBDCLK low, the keyboard will restart the one millisecond delay again. This operation will repeat until the next make code byte is sent.

Keyboard Scan Modes

Normal mode with Scan Set 2 non-standard scan codes for the keyboard subsystem are given in Table 9-1 and Table 9-2. Scan codes not listed are industry standard scan codes. The keyboard translates the Scan Set 2 scan code to produce the Scan Set 1 system codes.

Normal Mode Scan Codes (Scan Set 2)

Normal mode is the default mode of the enhanced keyboard. In this mode, the keyboard controller translates the make codes generated by the keyboard and converts them to the AT 101/102 keyboard Scan Set 1 system codes required by the system BIOS.

In the Normal mode, the keyboard generates the break code, a 2-byte sequence that consists of a make code immediately preceded by F0h.

Table 9-1 lists the non-standard scan codes for the keyboard operating in the Normal mode. Refer to the keyboard key assignments in Figure 9-1 and Figure 9-2 for keyboard key locations.

Table 9-1
Non-Standard Scan Codes (Hex) for the Normal Mode

Key Location	Scan Set 2 Make Code	Scan Set 2 Break Code	Scan Set 1 System Code
14 ****	6A	F0 6A	7D
29 *	5D	F0 5D	2B
42 **	5D	F0 5D	2B
45 ***	61	F0 61	56
56 ****	51	F0 51	—
70 ****	67	F0 67	—
71 ****	64	F0 64	—
72 ****	13	F0 13	—
P1^	10	F0 10	65
P2^	20	F0 20	67
P3^	30	F0 30	69
P4^	40	F0 40	6B

NOTE: * U.S. keyboard only
** International and Japanese keyboard
*** International keyboard only
**** Japanese keyboard only
^ User programmable keys

Table 9-2
Function Key Scan Code for the Normal Mode

Fn key (59) + Key Number	Make Code	Break Code
Fn+58	E0 14	E0 F0 14
Fn+75*	E0 12 E0 7C	E0 F0 7C E0 F0 12
Fn+76*	84	F0 84
Fn+80*	7E	F0 7E
Fn+81*	77	F0 77
Fn+85*	E1 14 77	E1 F0 14 F0 77
Fn+86*	14 E0 7E	E0 F0 7E F0 14
Fn+112*	78	F0 78
Fn+113*	07	F0 07

NOTE: *U.S. keyboard only

Hotkeys

Hotkeys and buttons are used to provide the user with easy access to various functions by using the **Fn** key or pushbuttons. The hotkeys and system responses are listed in Table 9-3 along with the indices to locations in memory for these and other system operations.

Table 9-3
Hotkey and SMI Switch/Key Parameters

Key Combination or Switch	Function	System Response
Fn + Esc	BIOS version	BIOS version information
Fn + F1	Popup Icon	Adjust location of popup icon
Fn + F3	Unused	
Fn + F4	CRT/LCD switch	Cycles between CRT and /LCD and both
Fn + F5	Warning Beep Control	on/off
Fn + F6	QuickLock/QuickBlank	Locks keyboard and turns off display
Fn + F7	Power Mgt.	Allows Power Drain level to be set
Fn + F8	Fuel Gauge	Displays battery charge condition
Fn + F9	Unused	
Fn + F10	Brightness	Display Brightness setting
Fn + F11	Reset ESCD	See NOTE
Fn + F12	Unused	
Fn + T	Stretch Video	Toggles the stretch video functions
Arrow Keys	Cursor Movement	Left, Right, Down, Up cursor (respectively)
Esc	Escape key	Escape functions
Lock Icon	Request Lock Icon	Keyboard/mouse activity while locked
Unlock Icon	Request Unlock Icon	Valid password entered while locked
Suspend Button	Suspend SMI	Enters/Exits Suspend Mode
Fn + Suspend Button	Hibernate SMI	Enters/Exits Hibernation Mode
Off Button	Power Off SMI	System power off

NOTE: Fn + **F11** clears the ESCD configuration information. If the **Fn** + **F11** sequence is pressed very early after powering the machine on (after you see the keyboard LEDs blink, but before the video is initialized), CMOS memory will be invalidated. The ESCD is cleared, the machine is reset and boots with the "162 - System Options Not Set" message. This is a way to clear out configuration information, such as Windows 95's knowledge about a docking station. It may help clear up problems if the configuration information had been corrupted. Timing of this keystroke sequence is critical, as there is a **very** narrow window during which the keys will be recognized.

Pointing Device Interface

The EasyPoint IV pointing device is a PS/2 compatible internal pointing device. If an external PS/2 pointing device is connected, the internal pointing device will work in parallel.[1]

---

[1] An external USB keyboard and mouse are also supported in BIOS firmware. Refer to Chapter 6 for details.

Data and clock outputs from the pointing device controller(s) are routed to the keyboard controller data port and onto the controller internal data bus for processing. The external keyboard and pointing devices from the expansion/convenience base are connected through separate keyboard and mouse PS/2 connectors. The computer has a single PS/2 connector that will accept an external mouse and/or keyboard through a Y-connector. As many as three PS/2 devices are able to communicate with the system at the same time as shown in Figure 9-4.

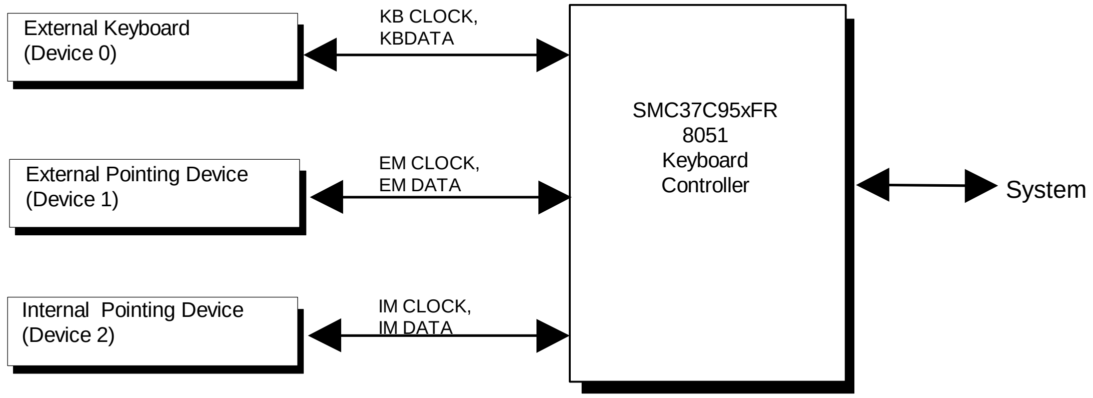

Figure 9-4. *Pointing Device Interface, Block Diagram*

Pointing Device Initialization

When the PS/2 mouse is powered on, it sends 0xAAh, then 0x00h.

Data Format

The data transmitted by the pointing device to the system consists of the X/Y coordinates of the pointer and status of the buttons. This data is transmitted in a three-byte serial PS/2 data packet that is compatible with the Microsoft PS/2 mouse.

Serial PS2 data, illustrated in Figure 9-5, is transmitted to the system using the following parameters:

Data Bits	8 bits
Start Bits	1 bit
Stop Bits	1 bit
Parity	1 bit, odd

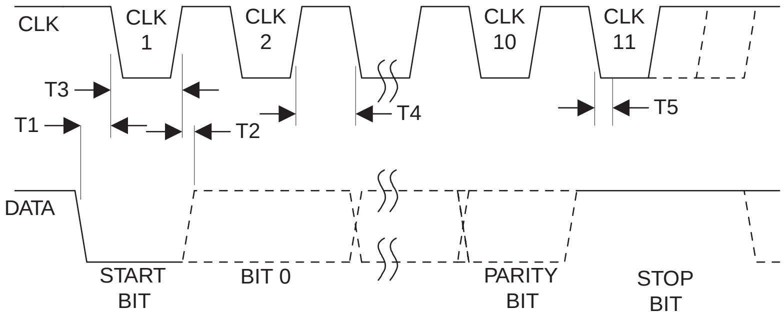

Figure 9-5. *PS/2 Data Transmission Sequence*

External Keyboard/Mouse Communications

Communication between the computer system and the external keyboard or mouse is bi-directional. The keyboard controller of the computer system controls both KBDDATA and KBDCLK for status signaling. When the keyboard is idle or between scan code transmissions, both KBDCLK and KBDDATA are set high.

Keyboard/Mouse-to-System Transmissions

PS/2 keyboards and PS/2 mice use the same data protocol. The following discussion applies to both devices. Prior to the beginning of a transmission, the keyboard must check the status of KBDDATA and KBDCLK. A low KBDCLK indicates keyboard inhibition and all keystrokes are loaded into the keyboard buffer. A low KBDDATA indicates the computer system is issuing a request-to-send (RTS) command. When the computer system issues an RTS, the keyboard will load all keystrokes into the buffer and prepare to receive data.

The keyboard will begin transmission when both KBDDATA and KBDCLK are high. Data transmissions consist of a start bit, eight data bits, an odd parity bit, and a stop bit. The keyboard will first set the correct level on KBDDATA, and then pulse KBDCLK low. After the transmission, the system will make KBDCLK low until the transmitted data is processed.

During a keyboard transmission, the computer system can request a transmission interrupt by lowering KBDCLK. The keyboard will check the state of KBDCLK every 100 microseconds during a transmission. If the line is detected low and the parity bit has been clocked out, the keyboard will finish the transmission. If KBDCLK is detected low before the rising edge of the parity bit is clocked out, the keyboard will abort the transmission. All data aborted during a transmission will be sent later.

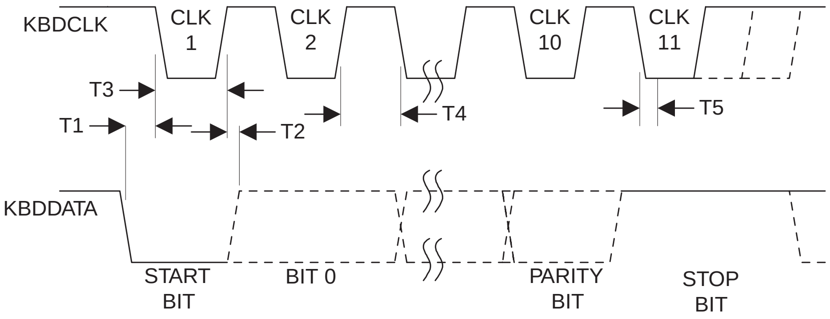

Figure 9-6. *Timing Diagram for Keyboard-to-System Transmissions*

System-to-Keyboard/Mouse Transmissions

The computer system has several commands that it may issue to the keyboard or mouse at any time. When the keyboard or mouse is transmitting to the computer system, the system first clamps the KBDCLK signal line to request a keyboard or mouse transmission halt. To ensure that the keyboard or mouse recognizes the interface request, the KBDCLK line must remain low (0) for at least 100 µs. If the keyboard or mouse transmission is past the rising edge of the parity bit clock pulse, the keyboard or mouse completes its transmission; if not, the keyboard or mouse will load the data into the character buffer and receive the data from the computer system.

Figure 9-7 shows the timing diagram for system-to-keyboard or mouse transmissions.

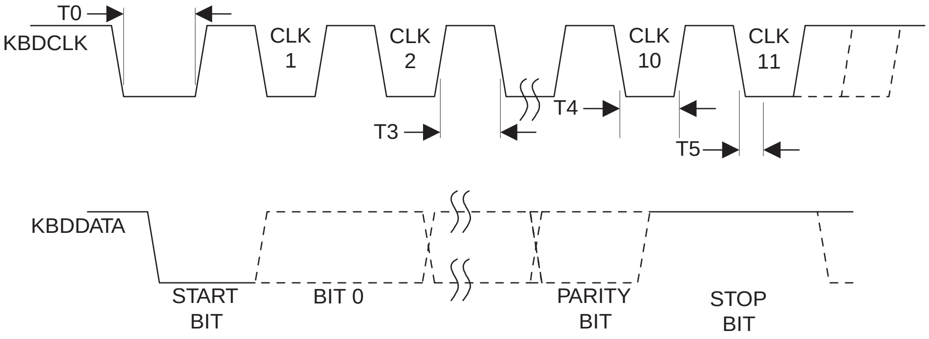

Figure 9-7. *Timing Diagram for System-to-Keyboard or Mouse Transmissions*

When the system is ready to transmit a command to the keyboard or mouse, it sets the DATA line low (0). This action serves as both a request-to-send and a start bit. Upon detecting the KBDDATA line low, the keyboard or mouse sets the KBDCLK line low, causing the start bit to be clocked out of the system. The interface then places the least-significant bit (LSB, data bit <0>) on the KBDDATA line, and the keyboard or mouse clocks out the LSB on the next negative going clock pulse. This process continues until all eight data bits are clocked out of the interface.

After all data bits are clocked out of the interface, the system places an odd parity bit on the KBDDATA line. The keyboard or mouse repeats its clocking of the parity bit as before. The keyboard or mouse then sets the KBDDATA line low and clocks this line to the system for a stop bit. When the keyboard or mouse receives the stop bit, the interface sets the KBDCLK line low, to inhibit the keyboard or mouse, while it is processing the received data.

After the external device receives a controller command, the external device returns an ACK code (0xFAh) to the controller. If a parity error or timeout occurs, a resend command (0xFEh) is sent to the keyboard controller.