Appendix B Remote Lockout

This appendix provides supporting information for requirement BIOS-0014.5, “System BIOS provides remote lockout capability.”

Initial Conditions

The initial state of the RLI setting after the power is turned on, cold boot, or warm boot is all events are enabled. Management software uses this interface to lock out events. Management software should re-enable events when sensitive operations have been completed.

Invocation and Parameter Passing

The RLI is only available in real mode and is invoked using INT 15H. All parameters are passed to and from the RLI functions using processor registers. The AH register is set to 25H for the RLI, and the AL register indicates the desired RLI function.

The AX, BX CX, DX, SI, and DI registers may be altered by the RLI. Management software should save these registers before calling RLI functions and restore them on return (after retrieving any parameters returned by the RLI). All other processor registers are preserved.

If the function is successful, the RLI function returns with the carry flag (CF) reset and the AH register set to zero.

If an error is encountered, the RLI function returns with CF set and AH set to:

```
86hERR_FUNCTION_NOT_SUPPORTED
```

The RLI supports three functions:

- Inquire Lockout Capabilities
- Get Remote Lockouts
- Set Remote Lockouts

All of the RLI functions use a bitmap to describe lockouts. Inquire Lockout Capabilities and Get Remote Lockouts return this bitmap. Set Remote Lockout uses this bitmap to select events to lock out. The bits are numbered from zero to 15 with the right-most bit being the least significant and bit 0. At a minimum, bits

1, 2, 4, and 5 in the Lockout Bitmap must be supported. The Lockout Bitmap is shown in the following table.

Lockout Bitmap

Bit	Definition
0	Permanently reserved.
1	Soft on/off switch. Controlled via ACPI power-button interfaces.
2	Reset button.
3	Mouse.
4	CTRL+ALT+DEL key combination.
5	All keyboard activity except CTRL+ALT+DEL.
6–15	Reserved for future use. Reset to zero for backward and forward compatibility.

Inquire Lockout Capabilities (00H)

This function returns the lockout capabilities of the system. The bit-mapped value returned in the CX register indicates the lockouts managed by the RLI. All return values are static. This function always returns the same values. The current lockout setting does not affect the value returned by this function.

Input

[AX] = 2500H

Output

[CF] = status

Set = error

Reset = success

[AH] = Return code, if [CF] set, then error code

86h ERR_FUNCTION_NOT_SUPPORTED

If [CF] reset,

[AH] = Zero (00)

[BX] = Revision (BCD-encoded with an implied decimal point between the bytes; for example, 0110H is 1.10

[CX] = Lockout capabilities (see Lockout Bitmap table)

If bit set, event lockout supported

If bit reset, event lockout not supported

Example

```
MOV     AX, 2500H           ; Inquire Lockout Capability
MOV     BX, 0               ; Clear output registers
MOV     CX, 0
INT     15H                 ; Able to get capabilities?
JC      ERROR               ; No, error out
MOV     wRevision, BX       ; Yes, save interface revision
MOV     wCapabilities, CX   ; and system lockout capabilities
```

Get Remote Lockouts (01H)

This function returns the current lockout setting. The bit-mapped value returned in the CX register indicates which events are enabled and which events are locked out.

Input

[AX] = 2501H

Output

[CF] = status

Set = error

Reset = success

[AH] = Return code, if [CF] set, then error code

86h ERR_FUNCTION_NOT_SUPPORTED

If [CF] reset,

[AH] = Zero (00)

[CX] = Current lockout setting for all supported events (see Lockout Bitmap table)

If bit set, event locked out

If bit reset, event enabled (or unsupported)

Example

```
MOV     AX, 2501H           ; Get Remote Lockouts
MOV     CX, 0               ; Clear output registers
INT     15H                 ; Able to get remote lockouts?
JC      ERROR               ; No, error out
```

```
MOV     wInitialSetting, CX     ; Save initial setting
TEST    CX, 4                   ; Is Reset event locked out?
JZ      RESET_ENABLED           ; No, continue
                                ; Yes
```

Comments: Only supported events are reported. Attempts to lock out unsupported events are not reflected in the Lockout Bitmap returned by this function. Only bits for supported events are ever set in the Lockout Bitmap.

Set Remote Lockouts (02H)

This function locks out or enables the specified events. The bit-mapped value passed in the CX register indicates which events to allow and which events to lock out.

Input

[AX] = 2502H

[CX] = Desired lockout setting (see Lockout Bitmap table)

If bit set, requests that the event be locked out

If bit reset, requests that the event be enabled

Output

[CF] = status

Set = error

Reset = success

[AH] = Return code, if [CF] set, then error code

86h ERR_FUNCTION_NOT_SUPPORTED

If [CF] reset,

[AH] = Zero (00)

Example

```
MOV     AX, 2502H               ;Set Remote Lockouts
MOV     CX, 2                   ;Disable Soft On/Off events, enable all others
INT     15H                     ;Able to set remote lockouts
JC      ERROR                   ;No, error out
```

Comments: If supported, this request always succeeds. Requesting lockout of an unsupported event is not an error. The interface simply ignores the request for that event and sets supported events as requested (locked out or enabled).