17 The High Kernel jumpblock.

Separate from the main firmware jumpblock is a small jumpblock for Kernel routines associated with ROM state and ROM selection. The routines accessed through this jumpblock are all RAM resident, to avoid confusion while the ROM state and ROM select are changed! The RAM area is copied out of ROM during the power-up initialization. The jumpblock should not be altered by the user.

The entry KL POLL SYNCHRONOUS is the ‘odd man out’ amongst the routines in this jumpblock. Unlike the other synchronous event handling routines, which are in the lower ROM, this routine is RAM resident. This minimises the overhead involved in polling for synchronous events.

A brief listing of the entries in this jumpblock can be found in section 14.3. A discussion of ROMs and the memory map can be found in section 2, further discussion of ROMs can be found in section 10 and a discussion of events can be found in section 12.

HI: KL U ROM ENABLE #B900

Enable the upper ROM.

Action:

Enables the currently selected upper ROM. Reading from addresses #C000 and up will now return the contents of the ROM.

Entry conditions:

No conditions.

Exit conditions:

A contains the previous ROM state.

Flags corrupt.
All other registers preserved.

Notes:

The mechanisms provided for calling subroutines in the upper ROM and for selecting upper ROMs automatically enable the ROM as required. This routine is used by the firmware but is otherwise of little use.

The previous ROM state may be passed to KL ROM RESTORE to reset the state to what it was before this routine was called.

This routine enables interrupts.

Related entries:

KL L ROM ENABLE
KL ROM RESTORE
KL ROM SELECT
KL U ROM DISABLE

HI: KL U ROM DISABLE #B903

Disable the upper ROM.

Action:

Disables the upper ROM. Reading from addresses #C000 and up will now return the contents of the RAM.

Entry conditions:

No conditions.

Exit conditions:

A contains the previous ROM state.

Flags corrupt.
All other registers preserved.

Notes:

Disabling the upper ROM gives read access to the top 16K of RAM, which is usually used as the screen memory. Note that the mapping of a location in screen memory to pixels on the screen depends on the mode and on the screen offset. It is inadvisable to disable the upper ROM while executing instructions in it! The previous ROM state may be passed to KL ROM RESTORE to reset the state to what it was before this routine was called. This routine enables interrupts.

Related entries:

KL L ROM DISABLE
KL ROM RESTORE
KL U ROM ENABLE

HI: KL L ROM ENABLE #B906

Enable the lower ROM.

Action:

Enables the lower ROM. Reading from addresses below #4000 will now return the contents of the ROM.

Entry conditions:

No conditions.

Exit conditions:

A contains the previous ROM state.

Flags corrupt.
All other registers preserved.

Notes:

In general the lower ROM is disabled except when a firmware routine is called. The firmware jumpblock arranges to enable the lower ROM automatically and to disable it again when the routine returns. This routine is used by the firmware but is otherwise of little use.

The previous ROM state may be passed to KL ROM RESTORE to reset the state to what it was before this routine was called.

This routine enables interrupts.

Related entries:

KL L ROM DISABLE
KL ROM RESTORE
KL U ROM ENABLE

HI: KL L ROM DISABLE #B909

Disable the lower ROM.

Action:

Disables the lower ROM. Reading from addresses below #4000 will now return the contents of the RAM.

Entry conditions:

No conditions.

Exit conditions:

A contains the previous ROM state. Flags corrupt. All other registers preserved.

Notes:

In general the lower ROM is disabled except when a firmware routine is called. The firmware jumpblock arranges to enable the lower ROM automatically and to disable it again when the routine returns.

The previous ROM state may be passed to KL ROM RESTORE to reset the state to what it was before this routine was called.

This routine enables interrupts.

Related entries:

KL LROM ENABLE
KL ROM RESTORE
KL U ROM DISABLE

HI: KL ROM RESTORE #B90C

Restore the previous ROM state.

Action:

The ROM state change routines all return a value giving the previous ROM state. Given that value KL ROM RESTORE will reset the state to what it was before the change.

Entry conditions:

A contains the previous ROM state.

Exit conditions:

AF corrupt. All other registers preserved.

Notes:

The previous ROM state is the value returned by one of:

KL U ROM ENABLE
KL U ROM DISABLE
KL L ROM ENABLE
KL L ROM DISABLE
KL ROM SELECT

It is possible to use KL U ROM DISABLE to reverse the effect of a call of KL U ROM ENABLE (amongst various other combinations). However, calling KL ROM RESTORE is the prefered method since it restores the state to what it was, which might have been enabled anyway.

This routine enables interrupts.

Related entries:

KL L ROM DISABLE
KL L ROM ENABLE
KL ROM SELECT
KL U ROM DISABLE
KL U ROM ENABLE

HI: KL ROM SELECT #B90F

Select a particular upper ROM.

Action:

Select a given upper ROM and enable the upper ROM.

Entry conditions:

C contains the ROM select address of the required ROM.

Exit conditions:

C contains the ROM select address of the previously selected ROM.
B contains the previous ROM state.

AF corrupt. All other registers preserved.

Notes:

The previous state can be passed to KL ROM RESTORE to reset the ROM enable to what it was. Both the previous state and the previous selection can be passed to KL ROM DESELECT to restore the state to what it was and to select the previously selected ROM again.

The mechanisms provided for calling routines in expansion ROMs automatically perform ROM selection as required (see section 2).

It is inadvisable to select another upper ROM whilst executing instructions in the upper ROM.

This routine enables interrupts.

Related entries:

KL CURR SELECTION
KL PROBE ROM
KL ROM DESELECT
KL ROM RESTORE

HI: KL CURR SELECTION #B912

Ask which upper ROM is currently selected.

Action:

Returns the ROM select address of the currently selected upper ROM.

Entry conditions:

No conditions.

Exit conditions:

A contains the ROM select address of the currently selected ROM.

All other registers and flags preserved.

Notes:

It is not possible to predict the ROM select address at which any particular expansion ROM will be fitted. The 'far address' used to reference subroutines in expansion ROMs includes a ROM select byte which must be set up at run time. This routine returns the ROM select address of the current ROM so that it can set up suitable ‘far addresses'.

Related entries:

KL PROBE ROM
KL ROM SELECT

HI: KL PROBE ROM #B915

Ask class and version of a ROM.

Action:

The first few bytes of all upper ROMs contain information in a standard form about the ROM. This routine extracts the class, mark number and version number bytes from the ROM at the given ROM select address.

Entry conditions:

C contains the ROM select address of the ROM to probe.

Exit conditions:

A contains the ROM's class.
L contains the ROM's mark number.
H contains the ROM's version number.

B and flags corrupt.

All other registers preserved.

Notes:

The ROM class returned may take any of the following values:

0:	Foreground ROM.
1:	Background ROM.
2:	Extension foreground ROM.
#80:	On board ROM (the built in BASIC foreground program).

Selecting a ROM address where no ROM is fitted implicitly selects the on-board ROM and so it will return #80 as its class.

The meaning of the mark and version numbers depends on the ROM.

See section 10 for a description of expansion ROMs.

This routine enables interrupts.

Related entries:

KL ROM SELECT
KL CURR SELECTION

HI: KL ROM DESELECT #B918

Restore previous upper ROM selection.

Action:

Set the ROM state and upper ROM selection to what they were before KL ROM SELECT was called.

Entry conditions:

C contains the ROM select address of the previously selected ROM.
B contains the previous ROM state.

Exit conditions:

C contains the ROM select address of the currently selected ROM.

B corrupt.
All other registers and flags preserved

Notes:

The previous ROM selection and state are the values returned by KL ROM SELECT. The currently selected ROM returned by this routine is the ROM that was selected by calling KL ROM SELECT (unless further selections have been made).

The mechanisms provided for calling subroutines in expansion ROMs automatically perform ROM selection as required.

It is inadvisable to select another upper ROM whilst executing instructions in the upper ROM.

This routine enables interrupts.

Related entries:

KL CURR SELECTION
KL ROM RESTORE
KL ROM SELECT

HI: KL LDIR #B91B

Move store (LDIR) with ROMs turned off.

Action:

Performs an LDIR instruction (LoaD Increment and Repeat) with both upper and lower ROMs disabled.

Entry conditions:

BC, DE, HL as required by the LDIR instruction.

Exit conditions:

F, BC, DE, HL as set by the LDIR instruction.
All other registers preserved.

Notes:

This routine may be used to move areas of RAM irrespective of the ROM state. This routine enables interrupts.

Related entries:

KL LDDR
RAM LAM (RST4)

HI: KL LDIR #B91E

Move store (LDDR) with ROMs turned off.

Action:

Performs an LDDR instruction (LoaD Decrement and Repeat) with both upper and lower ROMs disabled.

Entry conditions:

BC, DE, HL as required by LDDR instruction.

Exit conditions:

F, BC, DE, HL as set by LDDR instruction.
All other registers preserved.

Notes:

This routine may be used to move areas of RAM irrespective of the ROM state. This routine enables interrupts.

Related entries:

KL LDIR
RAM LAM (RST4)

HI: KL POLL SYNCHRONOUS #B921

Check if an event with higher priority than the current event is pending.

Action:

If the synchronous event queue is not empty then the priority of the highest priority pending event is compared with the current event's priority (if any).

Entry conditions:

No conditions.

Exit conditions:

If there is a higher priority event pending:

Carry true.

If there is no higher priority event pending:

Carry false.

Always:

A and other flags corrupt.
All other registers preserved.

Notes:

This routine is in the high jumpblock to minimise the overhead of polling for synchronous events. If the synchronous event queue is empty then the routine takes only a few instructions.

While a synchronous event is being processed the Kernel remembers its priority. The synchronous event routine may itself poll the synchronous event queue, but only events of a higher priority than itself are notified to it.

This routine may enable interrupts.

Related entries:

KL EVENT
KL DONE SYNC
KL DO SYNC
KL NEXT SYNC

HI: KL POLL SYNCHRONOUS #B92A

Ensure keyboard is scanned at next opportunity.

Action:

Force the Key Manager to scan the keyboard when the next ticker interrupt occurs. This may be used to reduce the probability of key pressings being missed while interrupts are disabled.

Entry conditions:

No conditions.

Exit conditions:

AF and HL corrupt.
All other registers preserved.

Notes:

The keyboard is normally scanned on every sixth ticker interrupt (every fiftieth of a second). While interrupts are disabled the ticks are lost and the keyboard will not be scanned. If interrupts are disabled for a significant period (more than three ticks) then this routine should be called just before interrupts are re-enabled. If interrupts are disabled for a long time (more than twelve ticks) then the user might consider calling this routine and re-enabling interrupts for a short time every fiftieth of a second.

Related entries:

KM SCAN KEYS