Introduction

The purpose of this document is to explain the differences between the API of CORE Driver Version 3.4.1 and CORE Driver Version 3.6.0, thus making it easier for the user to port the Intermediate Application Layer (IAL) based on CORE Driver Version 3.4.1 to the IAL based on Version 3.6.0.

The following sections list the modifications. If a change is backward compatible this is noted at the end of the section. For backward compatible changes there is no need to modify the IAL. Otherwise the IAL must be modified.

1.   Added defines for the ID's of the new devices supported

`MV_SATA_DEVICE_ID_6042`   Defines the PCI device ID for the MVSX6042.

`MV_SATA_DEVICE_ID_7042`   Defines the PCI device ID for the MVSX7042.

Note that this change is backward compatible.

2.   Added defines for the EDMA 128 entries mode

The EDMA of the 6042/7042 devices supports the 128 entries mode, in addition to the legacy 32 entries mode.

To support this feature the IAL must define the macro `MV_SATA_SUPPORT_GEN2E_128_QUEUE_LEN`. When it has been defined, the CORE Driver allocates up to 127 commands for each channel.

To override this default allocation length, the IAL must define `MV_SATA_OVERRIDE_GEN2E_SW_QUEUE_SIZE,` then it must define the macro `MV_SATA_REQUESTED_GEN2E_SW_QUEUE_SIZE` with the requested length.

Note that this change is backward compatible.

3.   Added enumerator for the SATA interface power management states

The enumerator `MV_SATA_IF_POWER_STATE` was added. Its values are:

`MV_SATA_IF_POWER_PHY_READY`   Interface in active state

`MV_SATA_IF_POWER_PARTIAL`   Interface in partial state

`MV_SATA_IF_POWER_SLUMBER`   Interface in slumber state

Note that this change is backward compatible.

4. Added support for the single data region feature

The 6042/7042 devices support this new single data region feature. To support this feature in the driver, the IAL must define the macro `MV_SATA_SUPPORT_EDMA_SINGLE_DATA_REGION`.

In addition, the following two fields were added to the `MV_UDMA_COMMAND_PARAMS` structure:

singleDataRegion	When set to `MV_TRUE`, the single data region is used for this command.
byteCount	Holds the byte count of the data buffer.

Note that this change is backward compatible.

5. Added support for allocating the command info structure on the IAL stack

Older versions of the CORE Driver used static allocation for the commands info data structure (`MV_QUEUE_COMMAND_INFO`). In Version 3.6.0 this allocation may be removed. Instead the CORE Driver uses memory allocated by the IAL.

To use this mode the IAL must add the define `MV_SATA_STORE_COMMANDS_INFO_ON_IAL_STACK`, and must make sure that the memory pointed by `pCommandInfo` (part of the `MV_QUEUED_COMMAND_ENTRY` structure) is owned by the CORE Driver as long as the command is not completed.

Note that this change is backward compatible.

6. Added new function `mvSataSetFBSMode`

PROTOTYPE

> `MV_U8 mvSataSetFBSMode (MV_SATA_ADAPTER *pAdapter, MV_U8 channelIndex,MV_BOOLEAN enableFBS, MV_BOOLEAN useQueueLen128)`

DESCRIPTION

> Enable FIS-based switching mode (FSB).

INPUTs

pAdapter	Pointer to the adapter data structure
channelIndex	Index of the required channel
enableFBS	If true then enable FBS mode
useQueueLen128	If true then enable the 128 EDMA queue length

RETURN

> MV_FALSE if
> The FBS feature is set to Enable for an adapter that doesn't support this feature.
> or
> The function is called while EDMA is enabled.
> or
> `useQueueLen128` is true and `enableFBS` is false.

COMMENTS

> This function only sets the software configuration. The hardware is configured when the `mvSataConfigEdmaMode()` function is called.

CONFIDENTIAL

Document Classification: Proprietary Information   June 20, 2005, Preliminary

Note that this change is backward compatible.

7. Added new function `mvSataSetInterfacePowerState`

PROTOTYPE

```
MV_U8 mvSataSetInterfacePowerState (MV_SATA_ADAPTER *pAdapter, MV_U8 channelIndex,
MV_SATA_IF_POWER_STATE ifPowerState)
```

DESCRIPTION

Sets the SATA interface power state of a specific SATA channel.

INPUT

pAdapter	Pointer to the adapter data structure
channelIndex	Index of the required channel
ifPowerState	SATA interface power state.

RETURN:

MV_FALSE if
The FBS feature is set to Enable for an adapter that doesn't support this feature.
or
The function is called while EDMA is enabled.
or
`useQueueLen128` is true and `enableFBS` is false.

COMMENTS

This function only sets the software configuration. The hardware is configured by `mvSataConfigEdmaMode()`.

Note that this change is backward compatible.

8. Added new function `mvSataGetInterfacePowerState`

PROTOTYPE

```
MV_BOOLEAN mvSataGetInterfacePowerState (MV_SATA_ADAPTER *pAdapter, MV_U8 channelIndex,
MV_SATA_IF_POWER_STATE *ifPowerState)
```

DESCRIPTION:

Gets the SATA interface power state of a specific SATA channel.

INPUT:

pAdapter	Pointer to the adapter data structure
channelIndex	Index of the required channel
ifPowerState	SATA interface power state.

RETURN

ifPowerState	SATA interface power state.

Note that this change is backward compatible.

9. Added new function `mvSata60X1B2CheckDevError`

PROTOTYPE

```
MV_BOOLEAN mvSata60X1B2CheckDevError(MV_SATA_ADAPTER *pAdapter, MV_U8 channelIndex)
```

DESCRIPTION

This function checks if the drive reported device errors. The 60X1 B2 may not issue an error interrupt when the device error is reported after transferring part of the data. This function must be called at regular intervals (e.g., 0.5 seconds).

INPUT

pAdapter   Pointer to the adapter data structure

channelIndex   Index of the required channel

RETURN

`MV_TRUE` if a device error is reported, and `MV_FALSE` otherwise.

Note that this change is *not* backward compatible, unless the 60x1 B2 device is not used. In that case the IAL must call this function, as described above.

Doc. No. MV-S103293-00 Rev. A

CONFIDENTIAL

Copyright © 2005 Marvell

Page 4

Document Classification: Proprietary Information

June 20, 2005, Preliminary