

INTEL CORPORATION

Simple Firmware Interface

Draft 0.8

Len Brown

len.brown@intel.com

Intel Software and Solutions Group – Open Source Technology Center

Kerry Vander Kamp

kerry.vander.kamp@intel.com

Intel Ultra Mobile Group – Low Power IA

03/02/2010

Copyright © 2010, Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others

Intel® Dynamic Acceleration Technology is a trademark of Intel Corporation in the U.S. and other countries.

Introduction

This document defines the Simple Firmware Interface (SFI). Platform firmware prepares SFI tables upon system initialization for the benefit of the Operating System (OS). The OS consults the SFI tables to augment platform knowledge that it gleans from native hardware interfaces, such as CPUID and PCI.

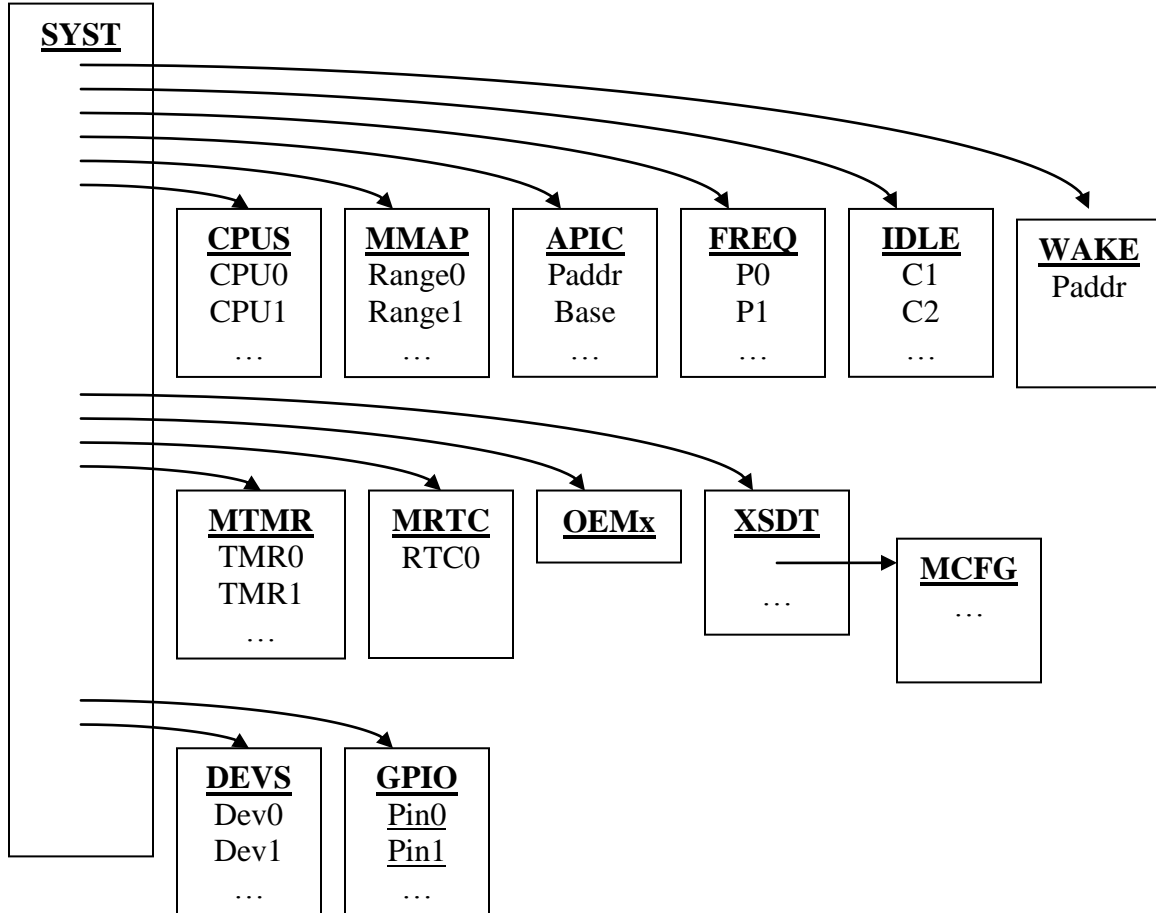


Figure 1 SFI Table Structure

The remainder of this document specifies the format of each of the SFI tables shown in figure 1.

SFI Common Table Header Format

All SFI tables have the common header (DESCRIPTION_HEADER), shown here. This format is a sub-set of the ACPI static table header format¹. Several SFI table fields contain character strings. Only strings that are shorter than the field need be NULL terminated. The OS will not use the characters after that NULL in string comparisons. However, the entire field, irrespective of character values, must always be included in table checksum calculations.

Field	Byte Length	Byte Offset	Description
Signature	4	0	The ASCII string representation of the table identifier. The OS ignores tables with signatures that it does not recognize even though the values in the Length and Checksum fields are correct.
Length	4	4	The length of the table, in bytes, including the header, starting from offset 0. This field records the size of the entire table.
Revision	1	8	The revision of the structure corresponding to the signature field for this table. Larger revision numbers are backward compatible to lower revision numbers with the same signature.
Checksum	1	9	The entire table, including the checksum field, must add to zero to be considered valid.
OEMID	6	10	An OEM-supplied string that identifies the OEM.
OEM Table ID	8	16	An OEM-supplied string that the OEM uses to identify the particular data table.

SFI System Table

The SFI System table consists of an SFI header followed by an array of entries. Each entry is simply a 64-bit physical address. Each address points to a table with an SFI common header.

Field	Byte Length	Byte Offset	Description
Signature	4	0	'SYST' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string

¹ SFI deleted 3 fields from the end of the ACPI common header – OEM Revision, Creator ID, and Creator Revision.

Field	Byte Length	Byte Offset	Description
OEM Table ID	8	16	OEM Table ID string
Entry	8*n	24	An array of 64-bit physical addresses that point to other DESCRIPTION_HEADERS. OSPM assumes at least the DESCRIPTION_HEADER is addressable, and then can further address the table based upon its Length field.

The OS finds the System Table by searching 16-byte boundaries between physical address 0x000E0000 and 0x000FFFFF. The SYST shall not cross a 4KB boundary. The OS shall search this region starting at the low address and shall stop searching when the 1st valid SFI System Table is found.

SFI CPU Table

The optional SFI CPU table enumerates the local-APIC ID's of all the processors present and enabled in the system.

Field	Byte Length	Byte Offset	Description
Signature	4	0	'CPUS' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Entry for Processor 0	4	24	Processor0 local APIC ID.
			...
Entry for Processor n			ProcessorN local APIC ID.

SFI APIC Table

The optional SFI APIC table enumerates the IO-APICs present in the system.

Field	Byte Length	Byte Offset	Description
Signature	4	0	'APIC' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1

Field	Byte Length	Byte Offset	Description
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Entry for IO-APIC 0	8	24	Physical Address IO-APIC 0. ²
			...
Entry for IO-APIC n	8		Physical Address of IO-APIC n.

SFI Memory Map Table

The optional SFI Memory Map table describes the RAM present in the system. It contains memory descriptors as defined in GetMemoryMap() API of the Unified Extensible Firmware Interface v2.1 (UEFI).³

Field	Byte Length	Byte Offset	Description
Signature	4	0	'MMAP' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Memory Descriptor	36*n	24	An array of EFI_MEMORY_DESCRIPTOR entries. The memory descriptor format is defined in the UEFI specification.

SFI Idle Table

The optional SFI Idle table describes the power saving CPU idle states (C-states) available to the OS. These states are accessed via native hardware support for MWAIT hint extensions. The IDLE table also enumerates the worst-case latency in microseconds to enter and exit each C-state. The C-states are enumerated in order of increasing latency and power savings. There are no latency restrictions.

The Idle table applies to every logical processor in the system. If there are topology dependencies between the logical processors that affect how the OS enters idle, the OS must discover those dependencies via native hardware interfaces.

² The SFI IO-APIC entry specifies only the base address of the IO-APIC. It omits two unneeded fields that are included in the ACPI MADT IO-APIC entry, the APIC-id and the global system interrupt (GSI) base.

³ <http://www.uefi.org>

Field	Byte Length	Byte Offset	Description
Signature	4	0	'IDLE' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Entry for C1	4	20	C1 MWAIT hint
	4	24	C1 latency in microseconds ...
Entry for Cn	4		Cn MWAIT hint
	4		Cn latency in microseconds

SFI Frequency Table

The optional SFI Frequency Table enumerates the processor performance states (P-states) available to the OS. PERF_CTL is the native MSR used to affect P-state transitions. The FREQ table applies to every logical processor in the system. If there are topology dependencies between the logical processors that affect how the OS enters P-states, the OS must discover them via native hardware interfaces.

Field	Byte Length	Byte Offset	Description
Signature	4	0	'FREQ' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Entry for P0	4	24	P0 Frequency in MHz
	4	28	P0 transition latency in microseconds
	4	32	P0 Control, value to write to PERF_CTL to enter P0 ...
Entry for Pn	4		Pn Frequency in MHz
	4		Pn transition latency in microseconds
	4		Pn Control, value to write to PERF_CTL to enter Pn

The SFI Frequency Table has no knowledge of Intel Dynamic Acceleration Technology (IDA). As with ACPI systems, the firmware writer may choose to enable IDA only for P0 and not for P1 and deeper. Or the OS may know how to control IDA via native hardware interfaces.

SFI M-Timer Table

The optional SFI M-Timer Table enumerates these system timers if present. The OS programming model for the M-Timer is described in the [Moorestown OS Writer's Guide].

Field	Byte Length	Byte Offset	Description
Signature	4	0	'MTMR' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	2
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Entry for M-Timer 0	8 4 4	24 32 36	Physical Address of M-Timer 0. Frequency of M-Timer 0 [Hz]. IRQ of M-Timer 0 ...
Entry for M-Timer n	8 4 4		Physical Address of M-Timer n. Frequency of M-Timer n [Hz]. IRQ of M-Timer n

SFI M-RTC Table

The optional SFI M-Real-Time-Clock Table enumerates these devices, if they are present. The OS programming model for the M-RTC is described in the [Moorestown OS Writer's Guide].

Field	Byte Length	Byte Offset	Description
Signature	4	0	'MRTC' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1

Field	Byte Length	Byte Offset	Description
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Entry for M-RTC 0	8	24	Physical Address of M-RTC 0.
	4	32	IRQ of M-RTC 0
			...
Entry for M-RTC n	8		Physical Address of M-RTC n.
	4		IRQ of M-RTC n.

SFI OEMx Table

The OEMx table id allows OEMs to define vendor-specific SFI tables while avoiding name-space collision with other platform vendors.

When the OS discovers an OEMx SFI table (OEM0, OEM1, etc.) it shall qualify that table signature with the 6-byte “OEM-id” and the 8-byte “OEM Table id”. If the OS can not identify the table using these three fields, then it shall ignore the table

SFI Wake Vector Table

The optional SFI Wake Vector table contains a 64-bit entry holding the address of the location where the OS writes its resume vector.

Field	Byte Length	Byte Offset	Description
Signature	4	0	‘WAKE’ table signature.
Length	4	4	32, length, in bytes, of the entire table.
Revision	1	8	2
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
Wake Vector Address	8	24	Physical Address of wake routine vector

SFI Platform Devices Table

The optional SFI platform device table contains a list of devices attached to the system which cannot be detected and enumerated via standard native methods, such as PCI.

Field	Byte Length	Byte Offset	Description
Signature	4	0	'DEVS' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
First Device Entry	1	24	Host Type. Defined in the following table.
	1	25	Host Number. Index of the host controller.
	2	26	Device Address. The definition is specific to the host type. For SPI, this is a chip select. For I2C, it is a slave address.
	1	28	Device IRQ. Use 0xFF for a device with no interrupt or a device with interrupt(s) attached via GPIO(s).
	4	29	Device Maximum Frequency in Hz.
	16	33	Device Name. Unique ASCII string used to identify device. ...
Last Device Entry	1		Host Type
	1		Host Number
	2		Device Address
	1		Device IRQ
	4		Device Maximum Frequency in Hz
	16		Device Name

Field	Value	Description
Host Type	0	Device is attached to a SPI host controller
	1	Device is attached to a I2C host controller
	2	Device is attached to a UART
	3	Device is attached to an HSI host controller
	4 - 255	Reserved for future host controller types

GPIO Pin Table

The optional SFI GPIO pin table contains a list of platform GPIOs used by the system for buttons, wakeup signals, keys, IRQ lines, etc.

Field	Byte Length	Byte Offset	Description
Signature	4	0	'GPIO' table signature.
Length	4	4	Length, in bytes, of the entire table. The length implies the number of Entry fields (<i>n</i>) at the end of the table.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID string
OEM Table ID	8	16	OEM Table ID string
First GPIO Entry	16	24	GPIO Controller Name. Unique ASCII string to identify the controller.
	2	40	GPIO Pin Number. This value is unique across the platform.
	16	42	GPIO Pin Name. Unique ASCII string to identify the pin.
			...
Last GPIO Entry	16		GPIO Controller Name
	2		GPIO Pin Number
	16		GPIO Pin Name

Appendix A. Change Log

0.1	2/20/2008	Initial draft
0.5	10/21/2008	Initial public draft
0.6	6/11/2009	WAKE table (rev 2) contains wake vector's address, not wake vector's value.
0.7	7/31/2009	SYST must be on 1 page, added I2CB and SPIB tables.
0.8	2/26/2010	Consolidated I2CB and SPIB tables into a new DEVS table. Added GPIO table.