RISC-V Functional Fixed Hardware Specification

RISC-V Platform Specification Task Group

Version v1.0.1, 2024-10-10: Ratified

Table of Contents

Preamble . . . 1
Copyright and license information . . . 2
Contributors . . . 3
Change Log . . . 4
Version 1.0.0 . . . 4
Version 1.0.1 . . . 4
1. Introduction . . . 5
1.1. Terms and Abbreviations . . . 5
1.2. References . . . 5
2. RISC-V FFH Resource Descriptor Encoding . . . 6
3. Use Cases . . . 7
3.1. Lower Power Idle States . . . 7
3.1.1. Entry Method . . . 7
3.1.2. Arch. Context Lost Flags . . . 7
3.2. Collaborative Processor Performance Control . . . 8
3.2.1. _CPC Object Resource Descriptor . . . 8
Appendix A: _LPI Object Examples . . . 9
Appendix B: _CPC Object Examples . . . 11

Preamble

This document is in the Ratified state

=== No changes are allowed. Any desired or needed changes can be the subject of a follow-on new extension. Ratified extensions are never revised.

Copyright and license information

Contributors

This RISC-V specification has been contributed to directly or indirectly by:

- Andrew Jones ajones@ventanamicro.com
- Atish Patra atishp@rivosinc.com
- Rafael Sene rafael@riscv.org
- Vasudevan Srinivasan vasu@rivosinc.com

Change Log

Version 1.0.0

- Initial version

Version 1.0.1

- Only print the specification state on the title page, not the entire text.

Chapter 1. Introduction

RISC-V systems which use Advanced Configuration and Power Interface (ACPI) require additional specifications for some ACPI object fields, typically those of type “Resource Descriptor”. A Functional Fixed Hardware (FFH) specification provides those additional specifications. The use cases addressed by this FFH are as follows:

- Lower Power Idle States (LPI), [ACPI 6.5] section 8.4.3
- Collaborative Processor Performance Control (CPPC), [ACPI 6.5] section 8.4.6

1.1. Terms and Abbreviations

This specification uses the following terms and abbreviations:

Term	Meaning
ACPI	Advanced Configuration and Power Interface Specification
ASL	ACPI Source Language
CPC	Continuous Performance Control
CPPC	Collaborative Processor Performance Control
FFH	Functional Fixed Hardware
HSM	Hart State Management
LPI	Low Power Idle
OSPM	Operating System-directed configuration and Power Management
SBI	Supervisor Binary Interface

1.2. References

Reference	Description
[ACPI 6.5]	Advanced Configuration and Power Interface Specification version 6.5 uefi.org/specs/ACPI/6.5/
[SBI]	RISC-V Supervisor Binary Interface Specification version 2.0 github.com/riscv-non-isa/riscv-sbi-doc/

Chapter 2. RISC-V FFH Resource Descriptor Encoding

Resource descriptors, which are formatted per the “Generic Register Descriptor” definition ([ACPI 6.5] section 6.4.3.7, also see ASL Register in section 19.6.114), are used by ACPI tables to provide the OSPM read and/or write access to platform-specific “registers”. A resource descriptor may be a fixed value, hardware address, function or function parameter identifier, or any other encoding of its bits. While encodings are specific to their applications, RISC-V uses a common top-level encoding whenever possible. That encoding is:

- Register bit width is 64
- The uppermost 4 bits are used to specify the type of identifier represented in the remaining bits. The possible types are described in table Table 1

Table 1. RISC-V FFH Resource Descriptor Identifier Type

Type	Description
0x0	None / Other
0x1	Bits[31:0] represent an SBI identifier
0x2	Bits[11:0] represent a CSR identifier

NOTE: While it is possible for identical RISC-V FFH Resource Descriptor addresses to appear across ACPI tables, the addresses may not share the same semantics. Each address must be interpreted per its respective ACPI table type.

Chapter 3. Use Cases

3.1. Lower Power Idle States

It is desirable for RISC-V system harts to transition to lower power states when they go idle. ACPI provides Lower Power Idle State (_LPI) objects which support an operating system's implementation of transitioning to lower power states. The following subsections specify some fields of the _LPI object for RISC-V systems. Refer to the ACPI specification for all remaining fields.

3.1.1. Entry Method

The _LPI object uses a "Resource Descriptor", which is formatted per Chapter 2, to specify the entry method for a lower power state. RISC-V systems set the resource descriptor as specified in Table 2:

Table 2. LPI Entry Method

Field	Value
Address Space	0x7F (FFixedHW)
Register Bit Width	64
Register Bit Offset	0
Access Size	4 (QWord)
Register Address	As specified in the Bits[63:60], Bits[59:32], and Bits[31:0] columns of Table 3

Table 3. LPI Entry Method Address

Bits[63:60] (Type)	Bits[59:32]	Bits[31:0]	Description
0x0	0x000_0000	0x0000_0000	WFI
0x1	0x000_0000	SBI HSM hart suspend type	Suspend the hart using the SBI HSM extension

All other encodings for LPI entry methods with *Address Space* set to *FFixedHW* (0x7f) are reserved for future use.

3.1.2. Arch. Context Lost Flags

_LPI objects also provide an *Arch. Context Lost Flags* field, which is a 32-bit integer, and may be used to indicate what processor context is lost. RISC-V _LPI objects must have appropriate flags set in this field as specified in Table 4:

Table 4. LPI Arch. Context Lost Flags

Bit offset	Bit width	Description
0	1	Set when the hart timer context is lost.

Bit offset	Bit width	Description
1	31	Reserved. Must be zero.

Appendix A provides examples for both a WFI entry method and SBI HSM hart suspend entry methods.

3.2. Collaborative Processor Performance Control

ACPI describes the Collaborative Processor Performance Control (CPPC) mechanism, which is an abstract and flexible mechanism for the operating system to collaborate with an entity in the platform to manage the performance of the harts. The platform entity may be the hart itself, the platform chipset, or a separate controller.

The ACPI _CPC object provides a way for the operating system to transition the hart into a performance state selected from an abstract, continuous range of values. Fields in the _CPC object may be static integers or “Resource Descriptors”. The following subsection specifies a RISC-V system “Resource Descriptor” for the _CPC object.

3.2.1. _CPC Object Resource Descriptor

The _CPC object may use a “Resource Descriptor”, which is formatted per Chapter 2, for many of its fields. When using an FFH Resource Descriptor for a _CPC field, it must be formatted as specified in Table 5:

Table 5. _CPC Resource Descriptor

Field	Value
Address Space	0x7F (FFixedHW)
Register Bit Width	64
Register Bit Offset	0
Access Size	4 (QWord)
Register Address	As specified in the Bits[63:60], Bits[59:32], Bits[31:12] and Bits[11:0] columns of Table 6

Table 6. _CPC Register Address

Bits[63:60] (Type)	Bits[59:32]	Bits[31:12]	Bits[11:0]	Description
0x1	0x000_0000	SBI CPPC Register ID		SBI CPPC access
0x2	0x000_0000	0x00000	CSR number	CSR access

All other encodings for _CPC Resource Descriptors with *Address Space* set to *FFixedHW* (0x7f) are reserved for future use.

Appendix B provides examples for both a CSR access and an SBI CPPC access.

Appendix A: _LPI Object Examples

```
Device (C000) {              // HART0
    Name (_HID, "ACPI0007")
    Name (_LPI,
        Package () {
            0,               // Revision
            0,               // LevelID
            3,               // Count

            // LPI1
            Package () {
                1,           // Min Residency (us)
                1,           // Worst case wakeup latency (us)
                1,           // Flags
                0,           // Arch. Context Lost Flags
                100,         // Residency Counter Frequency
                0,           // Enabled Parent State
                ResourceTemplate () {
                             // Entry Method
                    Register(FFixedHW, 64, 0,
                             0x0000_0000_0000_0000,
                             QWord)
                },
                ResourceTemplate () {
                             // Residency Counter Register
                    Register(SystemMemory, 0, 0, 0, 0)  // NULL
                },
                ResourceTemplate () {
                             // Usage Counter Register
                    Register(SystemMemory, 0, 0, 0, 0)  // NULL
                },
                             // State Name
                "RISC-V WFI"
            },

            // LPI2
            Package () {
                10,          // Min Residency (us)
                10,          // Worst case wakeup latency (us)
                1,           // Flags
                0,           // Arch. Context Lost Flags
                100,         // Residency Counter Frequency
                1,           // Enabled Parent State
                ResourceTemplate () {
                             // Entry Method
                    Register(FFixedHW, 64, 0,
                             0x1000_0000_0000_0000,
                             QWord)
```

```
                },
                ResourceTemplate () {
                            // Residency Counter Register
                    Register(SystemMemory, 0, 0, 0, 0)  // NULL
                },
                ResourceTemplate () {
                            // Usage Counter Register
                    Register(SystemMemory, 0, 0, 0, 0)  // NULL
                },
                            // State Name
                "RISC-V RET_DEFAULT"
            },

            // LPI3
            Package () {
                3500,       // Min Residency (us)
                100,        // Worst case wakeup latency (us)
                1,          // Flags
                0,          // Arch. Context Lost Flags
                100,        // Residency Counter Frequency
                1,          // Enabled Parent State
                ResourceTemplate () {
                            // Entry Method
                    Register(FFixedHW, 64, 0,
                             0x1000_0000_8000_0000,
                             QWord)
                },
                ResourceTemplate () {
                            // Residency Counter Register
                    Register(SystemMemory, 0, 0, 0, 0)  // NULL
                },
                ResourceTemplate () {
                            // Usage Counter Register
                    Register(SystemMemory, 0, 0, 0, 0)  // NULL
                },
                            // State Name
                "RISC-V NONRET_DEFAULT"
            }
        }
    )
}
```

Appendix B: _CPC Object Examples

```
Device (C000) {              // HART0
    Name (_HID, "ACPI0007")
    Name (_CPC,
        Package () {
            23,              // NumEntries
            3,               // Revision
            120,             // Highest Performance
            100,             // Nominal Performance
            40,              // Lowest Nonlinear Performance
            20,              // Lowest Performance
            ResourceTemplate () {
                             // Guaranteed Performance Register
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                             // Desired Performance Register
                Register(FFixedHW, 64, 0,
                         0x1000_0000_0000_0005,
                         QWord)
            },
            ResourceTemplate () {
                             // Minimum Performance Register
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                             // Maximum Performance Register
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                             // Performance Reduction Tolerance Register
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                             // Time Window Register
                Register(FFixedHW, 64, 0,
                         0x1000_0000_0000_0009,
                         QWord)
            },
            ResourceTemplate () {
                             // Counter Wraparound Time
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                             // Reference Performance Counter Register
                Register(FFixedHW, 64, 0,
                         0x2000_0000_0000_0C01,
                         QWord)
```

```
            },
            ResourceTemplate () {
                        // Delivered Performance Counter Register
                Register(FFixedHW, 64, 0,
                         0x1000_0000_0000_000C,
                         QWord)
            },
            ResourceTemplate () {
                        // Performance Limited Register
                Register(FFixedHW, 64, 0,
                         0x1000_0000_0000_000D,
                         QWord)
            },
            ResourceTemplate () {
                        // CPPC EnableRegister
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                        // Autonomous Selection Enable
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                        // AutonomousActivityWindowRegister
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            ResourceTemplate () {
                        // EnergyPerformancePreferenceRegister
                Register(SystemMemory, 0, 0, 0, 0)  // NULL
            },
            1,              // Reference Performance
            20,             // Lowest Frequency
            100,            // Nominal Frequency
        }
    )
}
```