ADVANCED MICRO DEVICES

# AMD Opteron™ A1100 Series Processor ACPI Porting Guide

## Version 1.00

1/20/2015

# AMD Opteron™ 1100 Series Processor ACPI Porting Guide

## Document History

| Date | Version | Change |
|---|---|---|
| Jan 20, 2015 | 1.00 | Initial release for Seattle RevA0 |

# Contents

# Introduction

The ACPI (Advanced Configuration and Power Interface) Specification defines a common set of firmware interfaces between the OS and the hardware platform that enables configuration and power management of the computer system and its devices. ACPI was co-developed by Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba with the initial specification published in 1996. Since October 2013, the UEFI Forum has taken the responsibility of developing and maintaining ACPI.

ACPI is a prevalent abstraction method used today for PC, Workstation, and Server platforms. Hardware manufacturers have products and processes built around ACPI descriptions and are requesting the same technology between x86 and ARM product offerings.

The intent of this document is to provide guidance on the ACPI interfaces required to support compliant operating systems running on a platform using an AMD Opteron™ A1100 Series Processor. It represents definitions AMD has developed based on published specifications and documents (see References section) in collaboration with firmware and operating systems partners.

Platform firmware must comply with the following requirements:

- UEFI must support the EFI_ACPI_20_TABLE_GUID (or later) configuration table.
- The pointer to the RSDP is passed via UEFI to the OSPM as described in UEFI 2.4 or later specification.
- The structure of all tables is consistent with the ACPI 5.1 specification.
- Use the 64-bit address option for address fields in ACPI tables.
- Implement the ACPI Hardware-reduced model.
- Support GPIO-signaled events for runtime notifications to the OSPM.

The Operating System directed Power Management (OSPM) on the platform receives a pointer to the Root System Description Pointer (RSDP) from the boot loader. OSPM then uses the information in the RSDP to determine the addresses of all other ACPI tables.

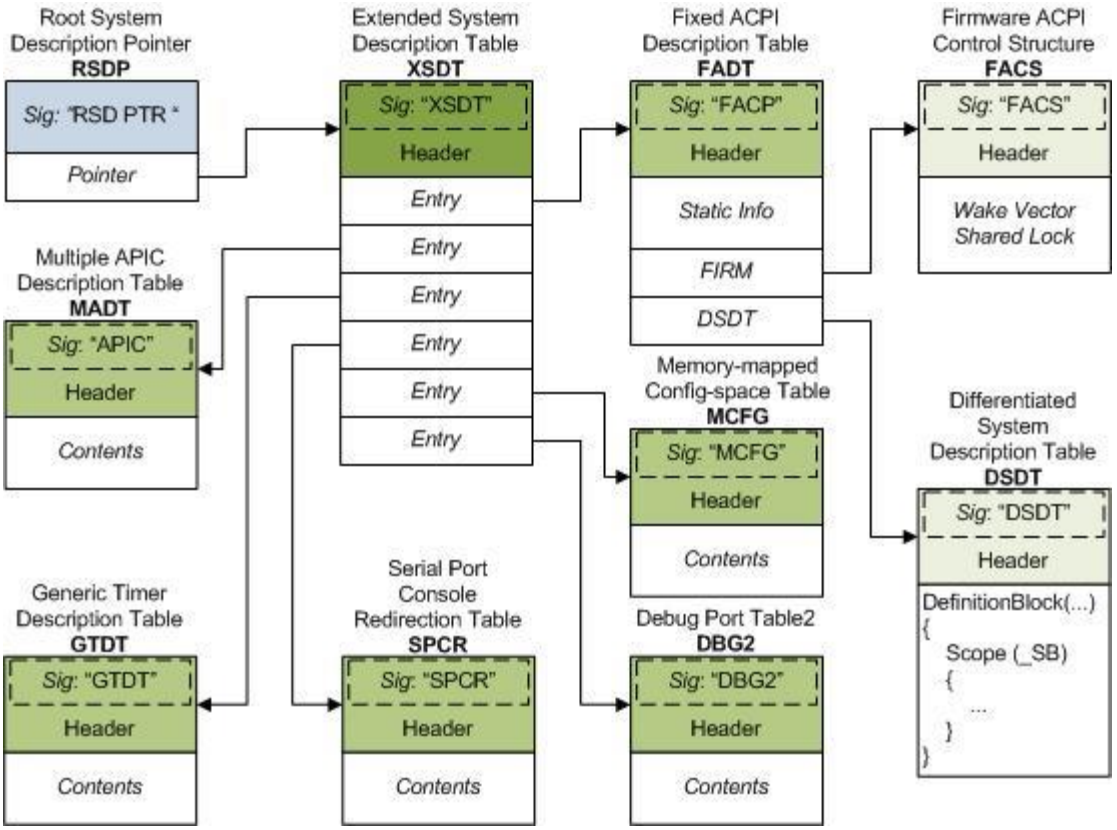


**Figure 1 – ACPI Description Tables**

# References

**[1] Advanced Configuration and Power Interface Specification, Revision 5.1 (ACPI 5.1 Spec)**
http://www.uefi.org/ACPIv5.1

**[2] ARM Server Base System Architecture (SBSA)**
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.den0029/index.html

**[3] ARM Server Base Boot Requirements (SBBR)**
http://infocenter.arm.com/help/topic/com.arm.doc.den00044a/index.html

**[4] Processor Programming Reference (PPR) for AMD Family 20h (PID# 53790)**
Access to this document currently requires and NDA with AMD

**[5] GIC-400 Generic Interrupt Controller Technical Reference Manual (GIC-400 TRM)**
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0471b/index.html

**[6] Multi-processor Startup for ARM Platforms**
https://acpica.org/sites/acpica/files/MP%20Startup%20for%20ARM%20platforms.doc

**[7] Windows ACPI Design Guide for SoC Platforms - Dev Center**
http://msdn.microsoft.com/en-us/library/windows/hardware/dn495676(v=vs.85).aspx

# TO-DO: What's next in this document?

- APEI Tables: Pending definition of RAS support in Seattle
- I/O Remapping Table: Pending definition of spec for SMMU (IORT) table

# Required ACPI Tables

The following tables comprise the minimal set required for compliant systems.

1. **RSDP - Root System Description Pointer**
   - The **RsdtAddress** field must be zero, and the **XsdtAddress** field must be a non-zero 64-bit value.

2. **XSDT - Extended System Description Table**
   - Must contain an array of 64-bit addresses that point to all other description tables.

3. **FADT - Fixed ACPI Description Table**
   - For legacy reasons the signature field is "**FACP**", and does not match the table's name.
   - The **DSDT** field must be zero, and the **X_ DSDT** field must be a non-zero 64-bit value.
   - If an **FACS** table is provided, the **FIRMWARE_CTRL** field must be zero, and the **X_FIRMWARE_CTRL** field must be a non-zero 64-bit value.
   - The **Preferred_PM_Profile** field should be set to a server profile (see section 5.2.9.1, ACPI 5.1 Spec [1]).
   - All SMI and SCI related fields must be set to zero.
   - All legacy Hardware Register Block fields must be set to zero.
   - The **Flags** field must indicate support for **HW_REDUCED_ACPI** and **LOW_POWER_S0_IDLE_CAPABLE** (Seattle does not support S3/S4 states).

4. **DSDT - Differentiated System Description Table**
   - This table provides definition blocks (AML code) for devices and system resources.

5. **MADT - Multiple APIC Description Table**
   - This table describes the ARM Generic Interrupt Controller (GIC).

6. **GTDT - Generic Timer Description Table**
   - This table describes the ARM Generic Timer in the Cortex-A57 cores.

7. **DBG2 - Microsoft Debug Port Table 2**
   http://msdn.microsoft.com/en-us/library/windows/hardware/dn639132(v=vs.85).aspx

8. **SPCR - Serial Port Console Redirection Table**
   http://msdn.microsoft.com/en-us/library/windows/hardware/dn639132(v=vs.85).aspx

# Appendix A – Sample Description Tables

## Root System Description Pointer Structure (RSD PTR)

**RSD PTR Structure (36 bytes):**

| <<RSD PTR >> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Signature** | 8 | 0 | 'RSD PTR ' |
| **Checksum** | 1 | 8 | <checksum> |
| **OEMID** | 6 | 9 | 'AMDINC' |
| **Revision** | 1 | 15 | 2 |
| **RsdtAddress** | 4 | 16 | 0 |
| **Length** | 4 | 20 | 36 |
| **XsdtAddress** | 8 | 24 | <XSDT Address> |
| **Extended Checksum** | 1 | 32 | <ext checksum> |
| ***Reserved*** | 3 | 33 | 0 |

## Extended System Description Table (XSDT)

**XSDT Table Format**

| <<XSDT>> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Signature** | 4 | 0 | 'XSDT' |
| **Length** | 4 | 4 | <36+8*n descriptors> |
| **Revision** | 1 | 8 | 1 |
| **Checksum** | 1 | 9 | <checksum> |
| **OEMID** | 6 | 10 | 'AMDINC' |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' |
| **OEM Revision** | 4 | 24 | 0 |
| **Creator ID** | 4 | 28 | <'AMD '> |
| **Creator Revision** | 4 | 32 | < 0 > |
| **Description Header[0]** | 8 | 36 | <FADT Address> |
| **Description Header[1]** | 8 | 44 | <MADT Address> |
| **Description Header[2]** | 8 | 52 | <GTDT Address> |
| **Description Header[3]** | 8 | 60 | <DBG2 Address> |
| **Description Header[4]** | 8 | 68 | <SPCR Address> |
| **Description Header[5]** | 8 | 76 | <MCFG Address> |

## Fixed ACPI Description Table (FADT)

**FADT Table Format**

| <<FACP>> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Signature** | 4 | 0 | 'FACP' |
| **Length** | 4 | 4 | 268 |
| **Revision (Major Version)** | 1 | 8 | 5 |
| **Checksum** | 1 | 9 | <checksum> |
| **OEMID** | 6 | 10 | 'AMDINC' |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' |
| **OEM Revision** | 4 | 24 | 0 |
| **Creator ID** | 4 | 28 | <'AMD '> |
| **Creator Revision** | 4 | 32 | < 0 > |
| **FIRMWARE_CTRL** | 4 | 36 | 0 |
| **DSDT** | 4 | 40 | 0 |
| ***Reserved*** | 1 | 44 | 0 |
| **Preferred_PM_Profile** | 1 | 45 | 4 <Enterprise Server> |
| **SCI_INT** | 2 | 46 | 0 |
| **SCI_CMD** | 4 | 48 | 0 |
| **ACPI_ENABLE** | 1 | 52 | 0 |
| **ACPI_DISABLE** | 1 | 53 | 0 |
| **S4BIOS_REQ** | 1 | 54 | 0 |
| **PSTATE_CNT** | 1 | 55 | 0 |
| **PM1a_EVT_BLK** | 4 | 56 | 0 |
| **PM1b_EVT_BLK** | 4 | 60 | 0 |
| **PM1a_CNT_BLK** | 4 | 64 | 0 |
| **PM1b_CNT_BLK** | 4 | 68 | 0 |
| **PM2_CNT_BLK** | 4 | 72 | 0 |
| **PM_TMR_BLK** | 4 | 76 | 0 |
| **GPE0_BLK** | 4 | 80 | 0 |
| **GPE1_BLK** | 4 | 84 | 0 |
| **PM1_EVT_LEN** | 1 | 88 | 0 |
| **PM1_CNT_LEN** | 1 | 89 | 0 |
| **PM2_CNT_LEN** | 1 | 90 | 0 |
| **PM_TMR_LEN** | 1 | 91 | 0 |
| **GPE0_BLK_LEN** | 1 | 92 | 0 |
| **GPE1_BLK_LEN** | 1 | 93 | 0 |
| **GPE1_BASE** | 1 | 94 | 0 |
| **CST_CNT** | 1 | 95 | 0 |
| **P_LVL2_LAT** | 2 | 96 | 0 |

| Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **P_LVL3_LAT** | 2 | 98 | 0 |
| **FLUSH_SIZE** | 2 | 100 | 0 |
| **FLUSH_STRIDE** | 2 | 102 | 0 |
| **DUTY_OFFSET** | 1 | 104 | 0 |
| **DUTY_WIDTH** | 1 | 105 | 0 |
| **DUTY_ALRM** | 1 | 106 | 0 |
| **MON_ALRM** | 1 | 107 | 0 |
| **CENTURY** | 1 | 108 | 0 |
| **IAPC_BOOT_ARCH** | 2 | 109 | 0 |
| ***Reserved*** | 1 | 111 | 0 |
| **FADT Flags** | 4 | 112 | <<FADT Flags>> |
| **RESET_REG** | 12 | 116 | <<NULL GAS>> |
| **RESET_VALUE** | 1 | 128 | 0 |
| **ARM_BOOT_ARCH** | 2 | 129 | <<ARM_Arch Flags>> |
| **FADT Minor Version** | 1 | 131 | 1 |
| **X_FIRMWARE_CTRL** | 8 | 132 | <FACS Address> |
| **X_DSDT** | 8 | 140 | <DSDT Address> |
| **X_PM1a_EVT_BLK** | 12 | 148 | <<NULL GAS>> |
| **X_PM1b_EVT_BLK** | 12 | 160 | <<NULL GAS>> |
| **X_PM1a_CNT_BLK** | 12 | 172 | <<NULL GAS>> |
| **X_PM1b_CNT_BLK** | 12 | 184 | <<NULL GAS>> |
| **X_PM2_CNT_BLK** | 12 | 196 | <<NULL GAS>> |
| **X_PM_TMR_BLK** | 12 | 208 | <<NULL GAS>> |
| **X_GPE0_BLK** | 12 | 220 | <<NULL GAS>> |
| **X_GPE1_BLK** | 12 | 232 | <<NULL GAS>> |
| **SLEEP_CONTROL_REG** | 12 | 244 | <<NULL GAS>> |
| **SLEEP_STATUS_REG** | 12 | 256 | <<NULL GAS>> |

**FADT Flags (4 bytes):**

| <<FADT Flags>> Field | Bit Length | Bit Offset | Value |
|---|---|---|---|
| ***Various Flags*** | 12 | 0 | 0 |
| **HEADLESS** | 1 | 12 | 1 |
| ***Various Flags*** | 7 | 13 | 0 |
| **HW_REDUCED_ACPI** | 1 | 20 | 1 |
| **LOW_POWER_S0_IDLE_CAPABLE** | 1 | 21 | 1 |
| ***Reserved*** | 10 | 22 | 0 |

**ARM_Arch Flags (2 bytes):**

| <<ARM_Arch Flags>> Field | Bit Length | Bit Offset | Value |
|---|---|---|---|
| **PSCI_COMPLIANT** | 1 | 0 | 1 if PSCI is implemented, else 0 |
| **PSCI_USE_HVC** | 1 | 1 | 1 if HVC should be preferred to SMC as the PSCI conduit, else 0 |
| ***Reserved*** | 14 | 2 | 0 |

**NULL Generic Address Structure (12 bytes):**

| <<NULL GAS>> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Address Space ID** | 1 | 0 | 0 |
| **Register Bit Width** | 1 | 1 | 0 |
| **Register Bit Offset** | 1 | 2 | 0 |
| **Access Size** | 1 | 3 | 0 |
| **Address** | 8 | 4 | 0 |

## Firmware ACPI Control Structure (FACS)

**FACS Structure Format**

| <<FACS>> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Signature** | 4 | 0 | 'FACS' |
| **Length** | 4 | 4 | 64 |
| **Hardware Signature** | 4 | 8 | 0 |
| **Firmware Waking Vector** | 4 | 12 | 0 |
| **Global Lock** | 4 | 16 | 0 |
| **Flags** | 4 | 20 | 0 |
| **X_Firmware Waking Vector** | 8 | 24 | 0 |
| **Version** | 1 | 32 | 2 |
| ***Reserved*** | 3 | 33 | 0 |
| **OSPM Flags** | 4 | 36 | 0 |
| ***Reserved*** | 24 | 40 | 0 |

# Multiple APIC Description Table (MADT)

**MADT Table Format**

| <<APIC>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Signature** | 4 | 0 | 'APIC' | |
| **Length** | 4 | 4 | 548 | |
| **Revision** | 1 | 8 | 3 | |
| **Checksum** | 1 | 9 | <checksum> | |
| **OEMID** | 6 | 10 | 'AMDINC' | |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' | |
| **OEM Revision** | 4 | 24 | 0 | |
| **Creator ID** | 4 | 28 | <'AMD '> | |
| **Creator Revision** | 4 | 32 | < 0 > | |
| **Controller Address** | 4 | 36 | 0xE112_F000 | PPR [4] |
| **APIC Flags** | 4 | 40 | 0 | Non PC-AT-compatible |
| **Local GIC[0]** | 76 | 44 | <<GICC Structure[0]>> | |
| **Local GIC[1]** | 76 | 120 | <<GICC Structure[1]>> | |
| **Local GIC[2]** | 76 | 196 | <<GICC Structure[2]>> | |
| **Local GIC[3]** | 76 | 272 | <<GICC Structure[3]>> | |
| **Local GIC[4]** | 76 | 348 | <<GICC Structure[4]>> | |
| **Local GIC[5]** | 76 | 424 | <<GICC Structure[5]>> | |
| **GIC Distributor** | 24 | 500 | <<GICD Structure>> | |
| **GIC MSI Frame** | 24 | 524 | <<GIC MSI Frame>> | |

**GIC Distributor (GICD) Structure (24 bytes):**

| <<GICD Structure>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xC | 0xC=GICD (Distributor) |
| **Length** | 1 | 1 | 24 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **GICD ID** | 4 | 4 | 0 | |
| **Physical Base Address** | 8 | 8 | 0xE111_0000 | PPR [4] |
| **System Vector Base** | 4 | 16 | 0 | |
| **GIC Version** | 1 | 20 | 0x2 | 0x2=GIC2vm |
| ***Reserved*** | 3 | 21 | 0 | |

### GIC CPU Interface (GICC) Structure[0]:

| <<GICC Structure[0]>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xB | 0xB=GICC (CPU Interface) |
| **Length** | 1 | 1 | 76 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **CPU Interface Number** | 4 | 4 | 0 | Bit index in distributor's GICD_ITARGETSR register |
| **ACPI Processor UID** | 4 | 8 | 0x0 | Same as MPIDR value |
| **GICC Flags** | 4 | 12 | <<GICC Flags>> | 1=Enabled |
| **Parking Protocol Version** | 4 | 16 | 1 | Set to 1 for valid Parked Address |
| **Performance Interrupt GSIV** | 4 | 20 | 39 | ID39, PPR [4] |
| **Parked Address** | 8 | 24 | <set by UEFI> | SBBR [3] |
| **Physical Base Address** | 8 | 32 | 0xE112_F000 | PPR [4] |
| **GICV Base Address** | 8 | 40 | 0xE116_F000 | PPR [4] |
| **GICH Base Address** | 8 | 48 | 0xE114_0000 | PPR [4] |
| **VGIC Maintenance Interrupt** | 4 | 56 | 25 | ID25, GIC-400 TRM [5] |
| **GICVR Base Address** | 8 | 60 | 0 | Not used in GICv2m |
| **MPIDR** | 8 | 68 | 0x0 | (ClusterId << 8) + CoreId; ClusterId=0..3, CoreId=0..1 |

### GIC CPU Interface (GICC) Structure[1]

| <<GICC Structure[1]>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xB | 0xB=GICC (CPU Interface) |
| **Length** | 1 | 1 | 76 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **CPU Interface Number** | 4 | 4 | 1 | Bit index in distributor's GICD_ITARGETSR register |
| **ACPI Processor UID** | 4 | 8 | 0x1 | Same as MPIDR value |
| **GICC Flags** | 4 | 12 | <<GICC Flags>> | 1=Enabled |
| **Parking Protocol Version** | 4 | 16 | 1 | Set to 1 for valid Parked Address |
| **Performance Interrupt GSIV** | 4 | 20 | 40 | ID40, PPR [4] |
| **Parked Address** | 8 | 24 | <set by UEFI> | SBBR [3] |
| **Physical Base Address** | 8 | 32 | 0xE112_F000 | PPR [4] |
| **GICV Base Address** | 8 | 40 | 0xE116_F000 | PPR [4] |
| **GICH Base Address** | 8 | 48 | 0xE114_0000 | PPR [4] |
| **VGIC Maintenance Interrupt** | 4 | 56 | 25 | ID25, GIC-400 TRM [5] |
| **GICVR Base Address** | 8 | 60 | 0 | Not used in GICv2m |
| **MPIDR** | 8 | 68 | 0x1 | (ClusterId << 8) + CoreId; ClusterId=0..3, CoreId=0..1 |

### GIC CPU Interface (GICC) Structure[2]:

| <<GICC Structure[2]>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xB | 0xB=GICC (CPU Interface) |
| **Length** | 1 | 1 | 76 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **CPU Interface Number** | 4 | 4 | 2 | Bit index in distributor's GICD_ITARGETSR register |
| **ACPI Processor UID** | 4 | 8 | 0x100 | Same as MPIDR value |
| **GICC Flags** | 4 | 12 | <<GICC Flags>> | 1=Enabled |
| **Parking Protocol Version** | 4 | 16 | 1 | Set to 1 for valid Parked Address |
| **Performance Interrupt GSIV** | 4 | 20 | 41 | ID41, PPR [4] |
| **Parked Address** | 8 | 24 | <set by UEFI> | SBBR [3] |
| **Physical Base Address** | 8 | 32 | 0xE112_F000 | PPR [4] |
| **GICV Base Address** | 8 | 40 | 0xE116_F000 | PPR [4] |
| **GICH Base Address** | 8 | 48 | 0xE114_0000 | PPR [4] |
| **VGIC Maintenance Interrupt** | 4 | 56 | 25 | ID25, GIC-400 TRM [5] |
| **GICVR Base Address** | 8 | 60 | 0 | Not used in GICv2m |
| **MPIDR** | 8 | 68 | 0x100 | (ClusterId << 8) + CoreId; ClusterId=0..3, CoreId=0..1 |

### GIC CPU Interface (GICC) Structure[3]

| <<GICC Structure[3]>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xB | 0xB=GICC (CPU Interface) |
| **Length** | 1 | 1 | 76 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **CPU Interface Number** | 4 | 4 | 3 | Bit index in distributor's GICD_ITARGETSR register |
| **ACPI Processor UID** | 4 | 8 | 0x101 | Same as MPIDR value |
| **GICC Flags** | 4 | 12 | <<GICC Flags>> | 1=Enabled |
| **Parking Protocol Version** | 4 | 16 | 1 | Set to 1 for valid Parked Address |
| **Performance Interrupt GSIV** | 4 | 20 | 42 | ID42, PPR [4] |
| **Parked Address** | 8 | 24 | <set by UEFI> | SBBR [3] |
| **Physical Base Address** | 8 | 32 | 0xE112_F000 | PPR [4] |
| **GICV Base Address** | 8 | 40 | 0xE116_F000 | PPR [4] |
| **GICH Base Address** | 8 | 48 | 0xE114_0000 | PPR [4] |
| **VGIC Maintenance Interrupt** | 4 | 56 | 25 | ID25, GIC-400 TRM [5] |
| **GICVR Base Address** | 8 | 60 | 0 | Not used in GICv2m |
| **MPIDR** | 8 | 68 | 0x101 | (ClusterId << 8) + CoreId; ClusterId=0..3, CoreId=0..1 |

### GIC CPU Interface (GICC) Structure[4]:

| <<GICC Structure[4]>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xB | 0xB=GICC (CPU Interface) |
| **Length** | 1 | 1 | 76 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **CPU Interface Number** | 4 | 4 | 4 | Bit index in distributor's GICD_ITARGETSR register |
| **ACPI Processor UID** | 4 | 8 | 0x200 | Same as MPIDR value |
| **GICC Flags** | 4 | 12 | <<GICC Flags>> | 1=Enabled |
| **Parking Protocol Version** | 4 | 16 | 1 | Set to 1 for valid Parked Address |
| **Performance Interrupt GSIV** | 4 | 20 | 43 | ID43, PPR [4] |
| **Parked Address** | 8 | 24 | <set by UEFI> | SBBR [3] |
| **Physical Base Address** | 8 | 32 | 0xE112_F000 | PPR [4] |
| **GICV Base Address** | 8 | 40 | 0xE116_F000 | PPR [4] |
| **GICH Base Address** | 8 | 48 | 0xE114_0000 | PPR [4] |
| **VGIC Maintenance Interrupt** | 4 | 56 | 25 | ID25, GIC-400 TRM [5] |
| **GICVR Base Address** | 8 | 60 | 0 | Not used in GICv2m |
| **MPIDR** | 8 | 68 | 0x200 | (ClusterId << 8) + CoreId; ClusterId=0..3, CoreId=0..1 |

### GIC CPU Interface (GICC) Structure[5]

| <<GICC Structure[5]>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xB | 0xB=GICC (CPU Interface) |
| **Length** | 1 | 1 | 76 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **CPU Interface Number** | 4 | 4 | 5 | Bit index in distributor's GICD_ITARGETSR register |
| **ACPI Processor UID** | 4 | 8 | 0x201 | Same as MPIDR value |
| **GICC Flags** | 4 | 12 | <<GICC Flags>> | 1=Enabled |
| **Parking Protocol Version** | 4 | 16 | 1 | Set to 1 for valid Parked Address |
| **Performance Interrupt GSIV** | 4 | 20 | 44 | ID44, PPR [4] |
| **Parked Address** | 8 | 24 | <set by UEFI> | SBBR [3] |
| **Physical Base Address** | 8 | 32 | 0xE112_F000 | PPR [4] |
| **GICV Base Address** | 8 | 40 | 0xE116_F000 | PPR [4] |
| **GICH Base Address** | 8 | 48 | 0xE114_0000 | PPR [4] |
| **VGIC Maintenance Interrupt** | 4 | 56 | 25 | ID25, GIC-400 TRM [5] |
| **GICVR Base Address** | 8 | 60 | 0 | Not used in GICv2m |
| **MPIDR** | 8 | 68 | 0x201 | (ClusterId << 8) + CoreId; ClusterId=0..3, CoreId=0..1 |

**GICC Flags (4 bytes):**

| <<GICC Flags>> Field | Bit Length | Bit Offset | Value | Comments |
|---|---|---|---|---|
| **Processor Enabled** | 1 | 0 | 1 | |
| **Performance Interrupt Mode** | 1 | 1 | 0 | 0=Level-Trigger, 1=Edge-Trigger |
| **VGIC Maintenance Interrupt Mode** | 1 | 2 | 0 | 0=Level-Trigger, 1=Edge-Trigger |
| ***Reserved*** | 29 | 3 | 0 | |

**GIC MSI Frame Structure (24 bytes):**

| <<GIC MSI Frame>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Type** | 1 | 0 | 0xD | 0xD (MSI Frame) |
| **Length** | 1 | 1 | 24 | |
| ***Reserved*** | 2 | 2 | 0 | |
| **GIC MSI Frame ID** | 4 | 4 | 0 | |
| **Physical Base Address** | 8 | 8 | 0xE118_0000 | PPR [4] |
| **Flags** | 4 | 16 | <<MSI Frame Flags>> | 1=Use SPI Count/Base fields |
| **SPI Count** | 2 | 20 | 256 | |
| **SPI Base** | 2 | 22 | 64 | |

**MSI Frame Flags (4 bytes):**

| <<MSI Frame Flags>> Field | Bit Length | Bit Offset | Value | Comments |
|---|---|---|---|---|
| **SPI Count/Base Select** | 1 | 0 | 1 | 0=Ignore SPI Count/Base fields, and use MSI_TYPER register. 1=Ignore MSI_TYPER register, and use SPI Count/Base fields. |
| ***Reserved*** | 31 | 1 | 0 | |

## Generic Timer Description Table (GTDT)

**GTDT Table Format:**

| <<GTDT>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Signature** | 4 | 0 | 'GTDT' | |
| **Length** | 4 | 4 | 96 | |
| **Revision** | 1 | 8 | 2 | |
| **Checksum** | 1 | 9 | <checksum> | |
| **OEMID** | 6 | 10 | 'AMDINC' | |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' | |
| **OEM Revision** | 4 | 24 | 0 | |
| **Creator ID** | 4 | 28 | <'AMD '> | |
| **Creator Revision** | 4 | 32 | < 0 > | |
| **CntControlBase Address** | 8 | 36 | 0xFFFF_FFFF_FFFF_FFFF | All 0xF's (Optional) |
| ***Reserved*** | 4 | 44 | 0 | |
| **Secure EL1 Timer GSIV** | 4 | 48 | 29 | ID29, GIC-400 TRM [5] |
| **Secure EL1 Timer Flags** | 4 | 52 | <<Timer Flags>> | <Timer Flags> |
| **Non-Secure EL1 Timer GSIV** | 4 | 56 | 30 | ID30, GIC-400 TRM [5] |
| **Non-Secure EL1 Timer Flags** | 4 | 60 | <<Timer Flags>> | <Timer Flags> |
| **Virtual Timer GSIV** | 4 | 64 | 27 | ID27, GIC-400 TRM [5] |
| **Virtual Timer Flags** | 4 | 68 | <<Timer Flags>> | <Timer Flags> |
| **Non-Secure EL2 Timer GSIV** | 4 | 72 | 26 | ID26, GIC-400 TRM [5] |
| **Non-Secure EL2 Timer Flags** | 4 | 76 | <<Timer Flags>> | <Timer Flags> |
| **CntReadBase Address** | 8 | 80 | 0 | No CntReadBase |
| **Platform Timer Count** | 4 | 88 | 0 | No SBSA Timers |
| **Platform Timer Offset** | 4 | 92 | 0 | Table offset |

**Timer Flags: Secure & Non-Secure EL1, Virtual & Non-Secure EL2 (4 bytes):**

| <<Timer Flags>> Field | Bit Length | Bit Offset | Value | Comments |
|---|---|---|---|---|
| **Timer Interrupt Mode** | 1 | 0 | 0 | 0=Level-Trigger, 1=Edge-Trigger |
| **Timer Interrupt Polarity** | 1 | 1 | 0 | 0=Active-High, 1=Active-Low |
| **Always-On Capability** | 1 | 2 | 0 | 0=May lose context, 1=Context saved |
| ***Reserved*** | 29 | 3 | 0 | |

# Debug Port Table 2 (DBG2)

**DBG2 Table Format**

| <<DBG2>> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Signature** | 4 | 0 | 'DBG2' |
| **Length** | 4 | 4 | 90 |
| **Revision** | 1 | 8 | 0 |
| **Checksum** | 1 | 9 | <checksum> |
| **OEMID** | 6 | 10 | 'AMDINC' |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' |
| **OEM Revision** | 4 | 24 | 0 |
| **Creator ID** | 4 | 28 | <'AMD '> |
| **Creator Revision** | 4 | 32 | < 0 > |
| **OffsetDbgDeviceInfo** | 4 | 36 | 44 |
| **NumberDbgDeviceInfo** | 4 | 40 | 1 |
| **DbgDeviceInfo** | 46 | 44 | <<DbgDeviceInfo>> |

**DBG Device Information Structure (46 bytes):**

| <<DbgDeviceInfo>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Revision** | 1 | 0 | 0 | |
| **Length** | 2 | 1 | 46 | |
| **NumberofGenericAddressRegisters** | 1 | 3 | 1 | |
| **NameSpaceStringLength** | 2 | 4 | 8 | |
| **NameSpaceStringOffset** | 2 | 6 | 38 | |
| **OemDataLength** | 2 | 8 | 0 | |
| **OemDataOffset** | 2 | 10 | 0 | |
| **Port Type** | 2 | 12 | 0x8000 | DBG2 Spec: Serial |
| **Port Subtype** | 2 | 14 | 0x0003 | DBG2 Spec: ARM PL011 UART |
| ***Reserved*** | 2 | 16 | 0 | |
| **BaseAddressRegisterOffset** | 2 | 18 | 22 | |
| **AddressSizeOffset** | 2 | 20 | 34 | |
| **BaseAddressRegister[0]** | 12 | 22 | <<DBG GAS>> | |
| **AddressSize[]** | 4 | 34 | 4096 | |
| **NamespaceString[]** | 8 | 38 | "COM1" | ASCIIZ String |

**DBG Generic Address Structure (12 bytes):**

| <<DBG GAS>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Address Space ID** | 1 | 0 | 0 | System Memory |
| **Register Bit Width** | 1 | 1 | 32 | 32-bit |
| **Register Bit Offset** | 1 | 2 | 0 | |
| **Access Size** | 1 | 3 | 3 | Dword (32-bit) |
| **Address** | 8 | 4 | 0xE101_0000 | PPR [4] |

# Serial Port Console Redirection Table (SPCR)

### SPCR Table Format

| <<SPCR>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Signature** | 4 | 0 | 'SPCR' | |
| **Length** | 4 | 4 | 80 | |
| **Revision** | 1 | 8 | 2 | |
| **Checksum** | 1 | 9 | <checksum> | |
| **OEMID** | 6 | 10 | 'AMDINC' | |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' | |
| **OEM Revision** | 4 | 24 | 0 | |
| **Creator ID** | 4 | 28 | <'AMD '> | |
| **Creator Revision** | 4 | 32 | < 0 > | |
| **Interface Type** | 1 | 36 | 0x03 | DBG2 Spec: ARM PL011 UART |
| **Reserved** | 3 | 37 | 0 | Must be 0 |
| **Base Address** | 12 | 40 | <<GAS>> | PPR [4] |
| **Interrupt Type** | 1 | 52 | 0x08 | Bit[3] ARMH GIC (GSIV) |
| **IRQ** | 1 | 53 | 0 | |
| **Global System Interrupt** | 4 | 54 | 360 | (328 + 32)<br>PPR [4] (IRQS_328) |
| **Baud Rate** | 1 | 58 | 7 | SPCR Spec: 7 = 115200 |
| **Parity** | 1 | 59 | 0 | SPCR Spec: 0 = No Parity |
| **Stop Bits** | 1 | 60 | 1 | SPCR Spec: 1 = 1 Stop-bit |
| **Flow Control** | 1 | 61 | 0 | SPCR Spec: 0 = None |
| **Terminal Type** | 1 | 62 | 3 | SPCR Spec: 3 = ANSI |
| **Reserved** | 1 | 63 | 0 | Must be 0 |
| **PCI Device ID** | 2 | 64 | 0xFFFF | Not a PCI device |
| **PCI Vendor ID** | 2 | 66 | 0xFFFF | Not a PCI device |
| **PCI Bus Number** | 1 | 68 | 0 | Not a PCI device |
| **PCI Device Number** | 1 | 69 | 0 | Not a PCI device |
| **PCI Function Number** | 1 | 70 | 0 | Not a PCI device |
| **PCI Flags** | 4 | 71 | 0 | |
| **PCI Segment** | 1 | 75 | 0 | |
| **Reserved** | 4 | 76 | 0 | Must be 0 |

### Generic Address Structure (12 bytes):

| <<GAS>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Address Space ID** | 1 | 0 | 0 | System Memory |
| **Register Bit Width** | 1 | 1 | 32 | 32-bit |
| **Register Bit Offset** | 1 | 2 | 0 | |
| **Access Size** | 1 | 3 | 3 | Dword (32-bit) |
| **Address** | 8 | 4 | 0xE101_0000 | PPR [4] |

# Memory-mapped Configuration-space Table (MCFG)

### MCFG Table Format

| <<MCFG>> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Signature** | 4 | 0 | 'MCFG' |
| **Length** | 4 | 4 | 60 |
| **Revision** | 1 | 8 | 1 |
| **Checksum** | 1 | 9 | <checksum> |
| **OEMID** | 6 | 10 | 'AMDINC' |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' |
| **OEM Revision** | 4 | 24 | 0 |
| **Creator ID** | 4 | 28 | <'AMD '> |
| **Creator Revision** | 4 | 32 | < 0 > |
| ***Reserved*** | 8 | 36 | 0 |
| **ConfigSpace Structure[]** | 24 | 44 | <<CfgSpace>> |

### Configuration-space Structure Format

| <<CfgSpace>> Field | Byte Length | Byte Offset | Value | Comments |
|---|---|---|---|---|
| **Base Address** | 8 | 0 | 0xF000_0000 | PPR [4] |
| **PCI Segment Group Number** | 2 | 8 | 0 | |
| **Start Bus Number** | 1 | 10 | 0 | PPR [4] |
| **Start End Number** | 1 | 11 | 15 | PPR [4] |
| ***Reserved*** | 4 | 12 | 0 | |

## Differentiated System Description Table (DSDT)

**DSDT Table Format**

| <<DSDT>> Field | Byte Length | Byte Offset | Value |
|---|---|---|---|
| **Signature** | 4 | 0 | 'DSDT' |
| **Length** | 4 | 4 | <36 + n bytes of AML code> |
| **Revision** | 1 | 8 | 1 |
| **Checksum** | 1 | 9 | <checksum> |
| **OEMID** | 6 | 10 | 'AMDINC' |
| **OEM Table ID** | 8 | 16 | 'SEATTLE ' |
| **OEM Revision** | 4 | 24 | 0 |
| **Creator ID** | 4 | 28 | <ASL Compiler ID> |
| **Creator Revision** | 4 | 32 | <ASL Compiler Revision> |
| **Definition Block[0]** | n | 36 | <n bytes of AML Code> |

# Appendix B - Sample DSDT (CPU Cores)

```
DefinitionBlock ("dsdt.aml", "DSDT", 1, "AMDINC", "SEATTLE ", 3)
{
  Scope (_SB)
  {
    Device (CPU0)
    {
      Name (_HID, "ACPI0007")    // Hardware ID
      Name (_UID, 0x000)         // Unique ID
    }

    Device (CPU1)
    {
      Name (_HID, "ACPI0007")    // Hardware ID
      Name (_UID, 0x001)         // Unique ID
    }

    Device (CPU2)
    {
      Name (_HID, "ACPI0007")    // Hardware ID
      Name (_UID, 0x100)         // Unique ID
    }

    Device (CPU3)
    {
      Name (_HID, "ACPI0007")    // Hardware ID
      Name (_UID, 0x101)         // Unique ID
    }

    Device (CPU4)
    {
      Name (_HID, "ACPI0007")    // Hardware ID
      Name (_UID, 0x200)         // Unique ID
    }

    Device (CPU5)
    {
      Name (_HID, "ACPI0007")    // Hardware ID
      Name (_UID, 0x201)         // Unique ID
    }
```

## Sample DSDT (AHCI and Ethernet)

```
Device (AHC0)   // AHCI Controller
{
  Name(_HID, "AMDI0600")     // Seattle AHCI/SATA
  Name (_CCA, 1)             // Cache-coherent controller
  Name (_CLS, Package (3)
  {
    0x01,       // Base Class: Mass Storage
    0x06,       // Sub-Class: Serial ATA
    0x01,       // Interface: AHCI
  })
  Name (_CRS, ResourceTemplate ()
  {
    Memory32Fixed (ReadWrite, 0xE0300000, 0x00010000)
    Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 387 } // GSIV
  }) // ResourceTemplate()
} // Device()

Device (ETH0)   // Ethernet
{
  Name(_HID, "AMDI8000")     // Seattle XGMAC
  Name(_UID, 0)
  Name (_CCA, 1)             // Cache-coherent controller
  Name (_CRS, ResourceTemplate ()
  {
    Memory32Fixed (ReadWrite, 0xE0700000, 0x00010000)  // XGMAC
    Memory32Fixed (ReadWrite, 0xE0780000, 0x00080000)  // XPCS
    Memory32Fixed (ReadWrite, 0xE1240800, 0x00000400)  // SERDES_RxTx
    Memory32Fixed (ReadWrite, 0xE1240000, 0x00000800)  // SERDES_CMU
    Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 357 } //XGMAC
    Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 355 } // XPCS
  }) // ResourceTemplate()

  Name (_DSD, Package () {
    ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
    Package () {
      Package (2) {"mac-address", Package (6) { 0, 0, 0, 0, 0, 0 }},
      Package (2) {"phy-mode", "xgmii"},
      Package (2) {"amd,serdes-channel", 0},
      Package (2) {"amd,speed-set", 0},
      Package (2) {"amd,dma-freq", 250000000},
      Package (2) {"amd,ptp-freq", 250000000}
    }
  }) // _DSD()
} // Device()
```

## Sample DSDT (UART, and SPI)

```
Device(COM1)    // Generic UART
{
  Name(_HID, "AMDI0511”)
  Name(_CID, "ARMH0011")
  Name (_ADR, 0xE1010000)
  Name(_CRS, ResourceTemplate() {
    Memory32Fixed (ReadWrite, 0xE1010000, 0x1000)
    Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 360 } // GSIV
  })
}

Device (SPI0)   // SPI controller
{
  Name(_HID, "AMDI0500")
  Name(_UID, 0)
  Name(_CRS, ResourceTemplate() {
    Memory32Fixed(ReadWrite, 0xE1020000, 0x1000) // GPIO Base Address
    Interrupt(ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 362 } //GSIV
  })
} // Device()

Device (SPI1)   // SPI controller
{
  Name(_HID, "AMDI0500")
  Name(_UID, 1)
  Name(_CRS, ResourceTemplate() {
    Memory32Fixed(ReadWrite, 0xE1030000, 0x1000) // GPIO Base Address
    Interrupt(ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 361 } //GSIV
  })

  Device(SDC0)
  {
    Name(_HID, "AMDI0501")   // SD Card/MMC slot
    Name(_CRS, ResourceTemplate() {
      SPISerialBus(1,                     // DeviceSelection
                   PolarityLow,           // DeviceSelectionPolarity
                   FourWireMode,          // WireMode
                   8,                     // DataBitLength
                   ControllerInitiated,   // SlaveMode
                   20000000,              // ConnectionSpeed
                   ClockPolarityLow,      // ClockPolarity
                   ClockPhaseFirst,       // ClockPhase
                   “\\SB.SPI1”,           // ResourceSource
                   0,                     // ResourceSourceIndex
                   ResourceConsumer,      // ResourceUsage
      ) // SPISerialBus()
      // SD Card “Presence” signal
      GpioInt(Edge, ActiveBoth, ExclusiveAndWake, PullDown, , "\\_SB.GIO0") {7}
    }) // ResourceTemplate()
  } // Device()
} // Device()
```

## Sample DSDT (GPIO, I2C and CCP)

```
Device (GIO0)       // GPIO controller
    {
      Name(_HID, "AMDI0400")
      Name(_UID, 0)
      Name(_CRS, ResourceTemplate() {
        Memory32Fixed(ReadWrite, 0xE1040000, 0x1000) // GPIO Base Address
        Interrupt(ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 391 } //GSIV
      })
    } // Device()

    Device (GIO1)   // GPIO controller
    {
      Name(_HID, "AMDI0400")
      Name(_UID, 1)
      Name(_CRS, ResourceTemplate() {
        Memory32Fixed(ReadWrite, 0xE1050000, 0x1000) // GPIO Base Address
        Interrupt(ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 390 } //GSIV
      })
    } // Device()

    Device(I2C0)    // I2C controller
    {
      Name(_HID, "AMDI0510")
      Name(_CRS, ResourceTemplate() {
        Memory32Fixed (ReadWrite, 0xE1000000, 0x1000)
        Interrupt (ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 389 } // GSIV
      })
    }

    Device (CCP0)   // Crypto Co-Processor
    {
      Name(_HID, "AMDI0C00")
      Name(_CRS, ResourceTemplate() {
        Memory32Fixed(ReadWrite, 0xE0100000, 0x10000) // GPIO Base Address
        Interrupt(ResourceConsumer, Level, ActiveHigh, Exclusive,,,) { 35 } //GSIV
      })
    } // Device()
```

## Sample DSDT (PCIe Root bus)

```
//
// PCIe Root bus
//
Device (PCI0)
{
  Name (_HID, "PNP0A08")     // PCI Express Root Bridge
  Name (_CID, "PNP0A03")     // Compatible PCI Root Bridge
  Name(_SEG, 0)              // Segment of this Root complex
  Name(_BBN, 0)              // Base Bus Number
  Name (_CCA, 1)             // Cache-coherent bus

  Name(_PRT, Package(4) {    // PCI Routing Table
    Package(4) {0x000000FFFF, 0, 0, 320}, // A
    Package(4) {0x000000FFFF, 1, 0, 321}, // B
    Package(4) {0x000000FFFF, 2, 0, 322}, // C
    Package(4) {0x000000FFFF, 3, 0, 323}  // D
  })

  Method (_CRS, 0, Serialized) { // Root complex resources
    Name (RBUF, ResourceTemplate () {
      WordBusNumber ( // Bus numbers assigned to this root
        ResourceProducer, MinFixed, MaxFixed, PosDecode,
        0,          // AddressGranularity
        0,          // AddressMinimum - Minimum Bus Number
        15,         // AddressMaximum - Maximum Bus Number
        0,          // AddressTranslation - Set to 0
        16          // RangeLength - Number of Busses
      )

      DWordMemory ( // 32-bit BAR Windows (1-1.5G)
        ResourceProducer, PosDecode,
        MinFixed, MaxFixed,
        Cacheable, ReadWrite,
        0x00000000, // Granularity
        0x40000000, // Min Base Address
        0x5FFFFFFF, // Max Base Address
        0x00000000, // Translate
        0x20000000  // Length
      )

      DWordMemory ( // 32-bit BAR Windows (1.5-2G)
        ResourceProducer, PosDecode,
        MinFixed, MaxFixed,
        Cacheable, ReadWrite,
        0x00000000, // Granularity
        0x60000000, // Min Base Address
        0x7FFFFFFF, // Max Base Address
        0x00000000, // Translate
        0x20000000  // Length
      )
```

```
DWordMemory ( // 32-bit BAR Windows (2-2.5G)
   ResourceProducer, PosDecode,
   MinFixed, MaxFixed,
   Cacheable, ReadWrite,
   0x00000000, // Granularity
   0x80000000, // Min Base Address
   0x9FFFFFFF, // Max Base Address
   0x00000000, // Translate
   0x20000000  // Length
 )

 DWordMemory ( // 32-bit BAR Windows (2.5G-3G)
   ResourceProducer, PosDecode,
   MinFixed, MaxFixed,
   Cacheable, ReadWrite,
   0x00000000, // Granularity
   0xA0000000, // Min Base Address
   0xBFFFFFFF, // Max Base Address
   0x00000000, // Translate
   0x20000000  // Length
 )

 QWordMemory ( // 64-bit BAR Windows (4-8G)
   ResourceProducer, PosDecode,
   MinFixed, MaxFixed,
   Cacheable, ReadWrite,
   0x0000000000000000, // Granularity
   0x0000000100000000, // Min Base Address
   0x00000001FFFFFFFF, // Max Base Address
   0x0000000000000000, // Translate
   0x0000000100000000  // Length
 )

 QWordMemory ( // 64-bit BAR Windows (8-16G)
   ResourceProducer, PosDecode,
   MinFixed, MaxFixed,
   Cacheable, ReadWrite,
   0x0000000000000000, // Granularity
   0x0000000200000000, // Min Base Address
   0x00000003FFFFFFFF, // Max Base Address
   0x0000000000000000, // Translate
   0x0000000200000000  // Length
 )

 QWordMemory ( // 64-bit BAR Windows (16-32G)
   ResourceProducer, PosDecode,
   MinFixed, MaxFixed,
   Cacheable, ReadWrite,
   0x0000000000000000, // Granularity
   0x0000000400000000, // Min Base Address
   0x00000007FFFFFFFF, // Max Base Address
   0x0000000000000000, // Translate
   0x0000000400000000 // Length
 )
```

```
      QWordMemory ( // 64-bit BAR Windows (32-64G)
        ResourceProducer, PosDecode,
        MinFixed, MaxFixed,
        Cacheable, ReadWrite,
        0x0000000000000000, // Granularity
        0x0000000800000000, // Min Base Address
        0x0000000FFFFFFFFF, // Max Base Address
        0x0000000000000000, // Translate
        0x0000000800000000  // Length
      )
      QWordMemory ( // 64-bit BAR Windows (64-128G)
        ResourceProducer, PosDecode,
        MinFixed, MaxFixed,
        Cacheable, ReadWrite,
        0x0000000000000000, // Granularity
        0x0000001000000000, // Min Base Address
        0x0000001FFFFFFFFF, // Max Base Address
        0x0000000000000000, // Translate
        0x0000001000000000  // Length
      )
      QWordMemory ( // 64-bit BAR Windows (128-256G)
        ResourceProducer, PosDecode,
        MinFixed, MaxFixed,
        Cacheable, ReadWrite,
        0x0000000000000000, // Granularity
        0x0000002000000000, // Min Base Address
        0x0000003FFFFFFFFF, // Max Base Address
        0x0000000000000000, // Translate
        0x0000002000000000  // Length
      )
      QWordMemory ( // 64-bit BAR Windows (256-512G)
        ResourceProducer, PosDecode,
        MinFixed, MaxFixed,
        Cacheable, ReadWrite,
        0x0000000000000000, // Granularity
        0x0000004000000000, // Min Base Address
        0x0000007FFFFFFFFF, // Max Base Address
        0x0000000000000000, // Translate
        0x0000004000000000  // Length
      )
    }) // Name(RBUF)

    Return (RBUF)
  } // Method(_CRS)
```

```
//
// OS Control Handoff
//
Name(SUPP,0) // PCI _OSC Support Field value
Name(CTRL,0) // PCI _OSC Control Field value
Method(_OSC, 4) {
  CreateDWordField(Arg3, 0, CDW1)
  If (LEqual(Arg0, ToUUID("33DB4D5B-1FF7-401C-9657-7441C03DD766"))) {
    CreateDWordField(Arg3,4,CDW2)
    CreateDWordField(Arg3,8,CDW3)
    Store(CDW2,SUPP)
    Store(CDW3,CTRL)
    If (LNotEqual(And(SUPP, 0x16), 0x16)) {
      And(CTRL,0x1E,CTRL)
    }
    And(CTRL,0x1D,CTRL)
    If (LNotEqual(Arg1,One)) {
      Or(CDW1,0x08,CDW1)
    }
    If (LNotEqual(CDW3,CTRL)) {
      Or(CDW1,0x10,CDW1)
    }
    Store(CTRL,CDW3)
    Return (Arg3)
  } Else {
    Or(CDW1,4,CDW1)
    Return (Arg3)
  }
} // Method(_OSC)

//
// Device-Specific Methods
//
Method(_DSM, 0x4, NotSerialized) {
  If (LEqual(Arg0, ToUUID("E5C937D0-3553-4d7a-9117-EA4D19C3434D"))) {
    switch (ToInteger(Arg2)) {
      //
      // Function 0: Return supported functions
      case(0) {
         Return (Buffer() {0xFF})
      }

      //
      // Function 1: Return PCIe Slot Information
      //
      case(1) {
        Return (Package(2) {
          One, // Success
          Package(3) {
            0x1,  // x1 PCIe link
            0x1,  // PCI express card slot
            0x1   // WAKE# signal supported
          }
        })
      }
```

```
//
// Function 2: Return PCIe Slot Number.
//
case(2) {
  Return (Package(1) {
    Package(4) {
      2,  // Source ID
      4,  // Token ID: ID refers to a slot
      0,  // Start bit of the field to use.
      7   // End bit of the field to use.
    }
  })
}

//
// Function 3: Return Vendor-specific Token ID Strings.
//
case(3) {
  Return (Package(0) {}) // TO-DO: Fill this in
}

//
// Function 4: Return PCI Bus Capabilities
//
case(4) {
  Return (Package(2) {
    One, // Success
    Buffer() {
      1,0,            // Version
      0,0,            // Status, 0:Success
      24,0,0,0,       // Length
      1,0,            // PCI
      16,0,           // Length
      0,              // Attributes
      0x0D,           // Current Speed/Mode
      0x3F,0,         // Supported Speeds/Modes
      0,              // Voltage
      0,0,0,0,0,0,0   // Reserved
    }
  })
}

//
// Function 5: Return Ignore PCI Boot Configuration
//
case(5) {
  Return (Package(1) {1})
}
```

```
            //
            // Function 6: Return LTR Maximum Latency
            //
            case(6) {
              Return (Package(4) {
                // TO-DO: Fill these in
                Package(1){0},  // Maximum Snoop Latency Scale
                Package(1){0},  // Maximum Snoop Latency Value
                Package(1){0},  // Maximum No-Snoop Latency Scale
                Package(1){0}   // Maximum No-Snoop Latency Value
              })
            }

            //
            // Function 7: Return PCI Express Naming
            //
            case(7) {
              Return (Package(2) {
                Package(1) {0},
                Package(1) {Unicode("PCI0")}
              })
            }

            //
            // Not supported
            //
            default {
            }
          }
        }
        Return (Buffer(){0})
    } // Method(_DSM)

    //
    // Root Complex 0
    //
    Device (RP0) {
      Name(_ADR, 0xF0000000)    // Dev 0, Func 0
    }
  } // Device(PCI0)

 } // Scope()
} // DefinitionBlock()
```

**DISCLAIMER**

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of non-infringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

---

**Trademarks**

AMD, the AMD Arrow logo, AMD Catalyst, AMD Opteron and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective owners.

ARM is a registered trademark of ARM Limited.

PCIe is a registered trademark of PCI-SIG Corporation.

---