brainchip

# AKD-1500
# Neuromorphic Processor
# System Hardware
# Data Sheet

Version 1.2

June 2025

**Table of Contents:**

**Figures:**



**Tables:**

**Revision History:**

| Version | Note |
|---|---|
| | |
| 1.2 | Add timing specifications.<br>Add theta Ja Jb, Correct Tj min max.<br>Add parameters for the clock input and xtal.<br>Add layout guidelines.<br>Fix document cross-referencing.<br>Added target application.<br>Add clocking modes. |
| 1.1 | Add ball composition and humidity. |
| 1.0 | Add specs for VPH33; Add 0.8V current spec.<br>Add block diagrams and supported modes for HRC. |

**References**:

**Acronyms:**

| | |
|---|---|
| NN | Neural Network, a network of neural processing cores |
| MM | Mesh Network Packet, containing event data transferred between neural processing cores |
| SS | Sub-system |
| NPU | Neural Processing Unit |
| NPE | Neural Processing Engine |
| EE | Execution Engine |
| HPC/HRC | High Performance/Resolution Convolution – Image to Events converter |
| | |

# 1 FEATURES

- 32-Core low-power Neural Processor
- Based on Brainchip's Akida1.0 IP.
- 8-bit x 8-bit x 3-ch front end convolution
- 4/2/1-bit activations on inner layers
- Supports multi-modal models
- Automatic model sequencing (Partial Reconfiguration)
- Edge Learning
- Easy to use, Brainchip's MetaTF development environment
- Interrupt on completion
- Up to 24 GPIO pins
- Host drivers available:
  - PCIe x2 Gen2
  - SPI
- Voltage: 0.8V (core), 1.8V (io), 3.3V (PCIe)
- Also available as IP
- Applications
  - Image Classification
  - Object Detection
  - Regression
  - Face Recognition
  - Keyword Spotting
  - Anomaly Detection
  - Point Cloud Classification

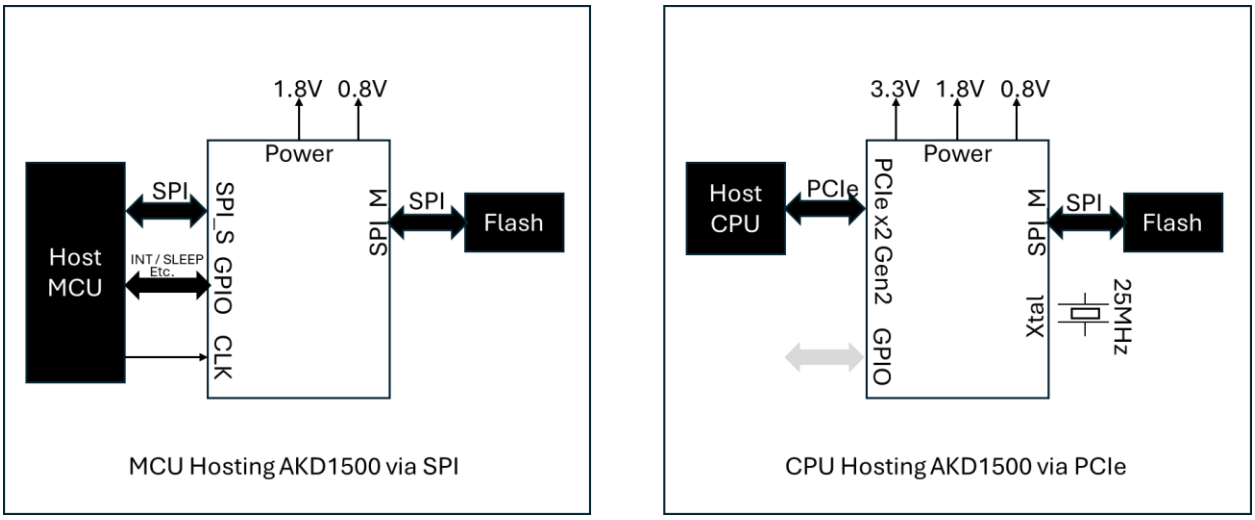



*Figure 1: Typical Configurations*

# 2 INTRODUCTION

This document contains the AKD-1500 Neuromorphic Processor data sheet and hardware design information, showing how to use the AKD1500 in a system. It is meant for system hardware engineers who are designing a board that includes the AKD1500 chip and for system software engineers who are developing products that incorporate the AKD1500.

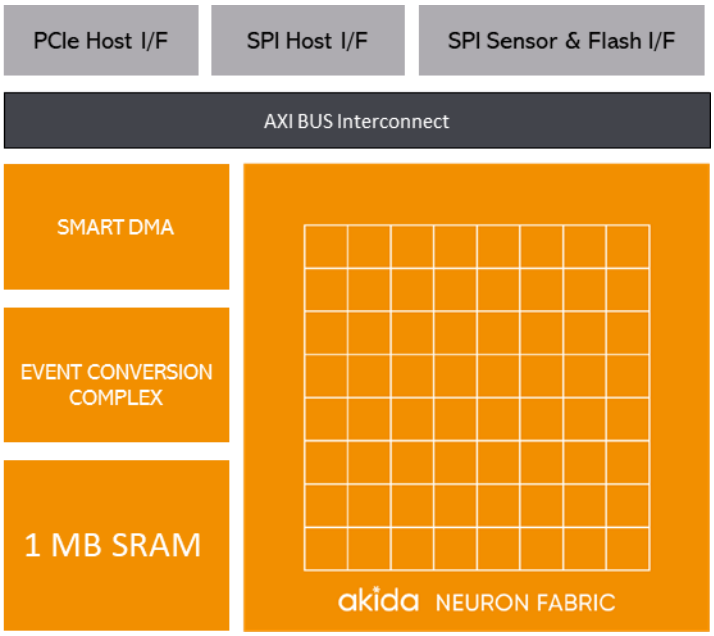



*Figure 2: Neural fabric, image to spike converter, and peripheral I/O channels*

The AKD1500 Neuromorphic Processor is targeted at the fast, self-contained execution of neural networks. To that means, the chip contains a neural fabric of 32 parallel Neural Processing Units (NPUs), arranged in 8 Physical Nodes that are interconnected through an on-chip mesh network.

Neural networks run entirely within the neural fabric, whereby each individual NPU is addressing its own memory, thereby avoiding memory contention.

The integrated PCIe or SPI interfaces are used to initialize the neural fabric with weight values and configuration details that can be stored in external flash memory.

The chip contains a 24-bit data to events converter for generating image events for the AKIDA neural fabric. Details of the internal interconnections of various blocks are shown in the following block diagram:

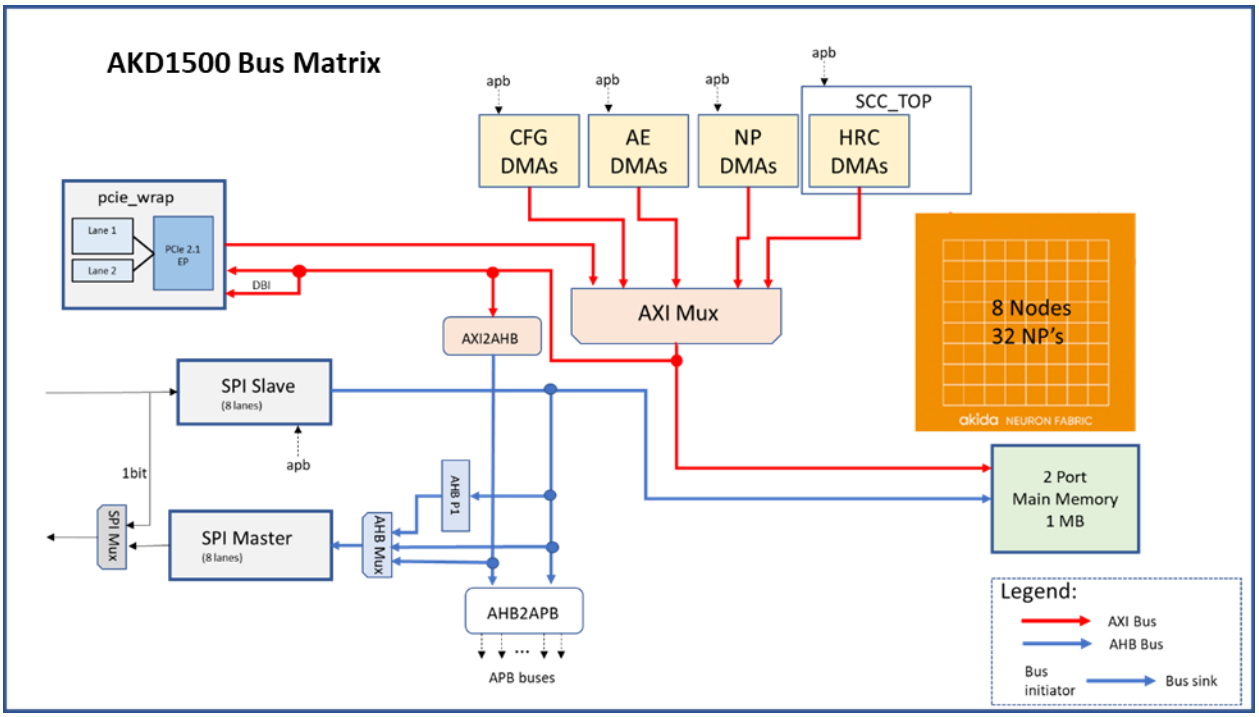



*Figure 3: AKD1500 Bus Matrix*

## 2.1 TARGET APPLICATIONS

The Akida AKD1500 from BrainChip is an ultra-low-power, event-based neuromorphic AI processor designed for developing edge AI solutions. It mimics brain-like processing by analyzing only essential sensor data, enabling highly efficient, on-chip learning that reduces power consumption, latency, and reliance on the cloud.

This technology is ideal for industries such as **consumer electronics** (smart homes, wearables), **Industrial IoT** (predictive maintenance, automation), **security** (intelligent surveillance), **AIoT**, and **healthcare** (vital sign monitoring).

| Industry | Applications/Use Cases |
|---|---|
| Consumer Electronics | Smart home devices (e.g., automatic doorbells, appliance voice interfaces), wearables (e.g., fitness trackers, smart glasses, presence detection), personal electronics. |
| Industrial IoT (IIoT) | Predictive maintenance, industrial automation, robotics (including autonomous quadruped robotics), intelligent sensors, vibration detection, anomaly detection. |
| Security | Intelligent surveillance systems, biometric recognition, post-quantum cryptographic security, cybersecurity acceleration. |

| | |
|---|---|
| Healthcare | Vital sign monitoring, seizure prediction (in wearables), medical device control, audio enhancement for hearing assistance. |
| Aerospace & Defense | Low Size, Weight, and Power (SWaP) machine learning applications in spacecraft and robotics, defense and intelligence applications. |
| Robotics | Autonomous navigation, object detection and classification, gesture recognition, AI for industrial and service robots. |
| Other | Low power applications |

# 3 AKD1500 BLOCK DESCRIPTIONS

## 3.1 GENERAL

## 3.2 CLOCKING MODES

### 3.2.1 PLL Setup

In Normal Mode (OP_MODE=0), the default setup of the PLL is such that the AKD1500 internal logic is clocked at 400MHz, based on a 25MHz input reference clock. After reset, once the PLL locks, the clocking logic automatically switches from the slow reference clock to the fast PLL's output clock.

### 3.2.2 Host Select

The input pin PCIE_HOST_SEL determines which host interface is used in the AKD1500. When set to 1, the PCIe Interface is used, when set to 0, the SPI Slave is used. The unused interface will have its clocks turned off automatically.

### 3.2.3 Sleep Mode

Sleep Mode can be entered by pulling high the AKD1500 Input pin "SLEEP". All internal clocks will be shutdown, except the ones required to keep the PCIe Link alive (if applicable). PLL and clock dividers are kept active for quick recovery. The SPI slave and master could delay the sleep request until no more transactions are pending.

### 3.2.4 Safe Mode

In Safe Mode (OP_MODE1 = 1), automatic clock shutdown based on PCIE_HOST_SEL is disabled, and both SPI and PCIe interfaces are active. The automatic switching to PLL clock is disabled in this mode, and PLL manual override is enabled. In safe mode, after Reset, the Host needs to monitor PLL lock and switch the system from reference clock to the fast PLL clock once lock is detected.

## 3.3 GPIO

- 4-bits of general purpose Input/Output ports.
- Up to 20 of the unused SPI master and SPI slave interface signals can be programmed as GPIO ports.
- Maximum of 24 GPIO pins are available if unused SPI interfaces are utilized for GPIO's.

## 3.4 SRAM

AKD1500 has 1MB of dual-port memory for expediting data transfers.

## 3.5 SPI_M (SPI Master) Interface Access

AKD1500 SPI_M provides two ways to access SPI: eXecute In Place (XIP) access and TRX FIFO access. XIP access is the preferred method because it requires much less programming effort while converting the 32-bit data access straight into SPI traffic. TRX FIFO access method is only used when single-byte SPI commands need to be issued to the external SPI memory device. SPI_M supports DDR (Double Data Rate) SPI accesses. Table 1 shows the supported SPI protocols.

*Table 1: SPI protocols supported by SPI_M*

| SPI Protocol | XIP support | TRX FIFO support |
|---|---|---|
| Single | No | Yes |
| Dual | Yes | Yes |
| Dual DDR | Yes | Yes |
| Quad | Yes | Yes |
| Quad DDR | Yes | Yes |
| Octal | Yes | Yes |
| Octal DDR | Yes | Yes |
| Hyperbus | Yes | Yes |

### 3.5.1 XIP Access

In order to enable XIP access, the user needs to configure the following registers in SPI_M. Please refer to Section 5.1 and 5.3 of **DWC_SSI databook** for details. During mission mode, the SPI master is automatically placed in XIP mode.

#### 3.5.1.1 *Baud Rate Select Register*

Offset: 0x14

Unmentioned register bits shall be set to 0.

[15:1] – SCKDV: SSI Clock Divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation:

Fsclk_out = Fssi_clk/BAUDR

where BAUDR is any even value between 2 and 65534 (BAUDR = {SCKDV*2}). For example: for Fssi_clk = 3.6864MHz and BAUDR =2 Fsclk_out = 3.6864/2 = 1.8432MHz

#### 3.5.1.2 *Control Register 0*

Offset: 0x00

Unmentioned register bits shall be set to 0.

[31] – SSI_IS_MST: shall always be 1.

[24] – SPI_HYPERBUS_EN: set to 1 to activate Hyperbus mode.

[23:22] – SPI_FRF: select SPI rate: 2'b00 = single; 2'b01 = double; 2'b10 = quad; 2'b11 = octal

[4:0] – DFS: data frame size. Set to 5'h1f for 32-bit serial data transfer.

#### 3.5.1.3 *Slave Enable Register*

Offset: 0x10

Unmentioned register bits shall be set to 0.

[1:0] – SER: Each bit in this register corresponds to a slave select line (SPI_S_2MCS0_N and SPI_S_2MCS1_N) from SPI_M. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins.

#### 3.5.1.4 *SPI Control Register*

Offset: 0xf4

Unmentioned register bits shall be set to 0.

[27:26] – XIP_MBL: XIP Mode bit length: 2'b00 = 2 bits; 2'b01 = 4 bits; 2'b10 = 8 bits; 2'b11 = 16 bits. Please configure according to the external SPI memory device's requirements.

[25] – SPI_RXDS_SIG_EN: set to 1 to support *rxds* signaling by Hyperbus slave devices during Command-Address (CA) phase.

[20] – XIP_INST_EN: set to 1 to enable the SPI instruction field for XIP transfer.

[18] – SPI_RXDS_EN: set to 1 when using the data strobe input pin SPI_M_DQS to capture read data in DTR mode.

[17] – INST_DDR_EN: set to 1 when the SPI instruction field is in DDR mode.

[16] – SPI_DDR_EN: set to 1 when the SPI is in DDR mode (not the instruction).

[15:11] – WAIT_CYCLES: the wait time between control frames transmit and data reception in SPI clock cycles.

[9:8] – INST_L: SPI instruction length in bits. Set to 2 for 8-bit length.

[7] – XIP_MD_BIT_EN: set to 1 to enable Mode bits in XIP mode, which are always 8 bits and sent after the address field.

[5:2] – ADDR_L: defines the length of the address to be transmitted. The length is ADDR_L x 4 bits.

[1:0] – TRANS_TYPE: address and instruction transfer format. Set to 2 to have both specified by SPI_FRF.

#### 3.5.1.5 *Transmit Drive Edge Register*

Offset: 0xf8

This register is used to control the driving edge of TXD register in DDR mode. Unmentioned register bits shall be set to 0.

[7:0] – TDE: TXD Drive edge register which decided the driving edge of transmit data. The maximum value of this register is = (BAUDR/2) -1.

#### 3.5.1.6 *XIP Mode Bits Register*

Offset: 0xfc

Unmentioned register bits shall be set to 0.

[7:0] – XIP_MD_BITS: XIP mode bits to be sent after address phase of XIP transfer.

#### 3.5.1.7 *XIP INCR Transfer Opcode Register*

Offset: 0x100

Unmentioned register bits shall be set to 0.

[15:0] – INCR_INST: XIP INCR transfer opcode, the instruction field content sent by SPI_M.

#### 3.5.1.8 *SSI Enable Register*

Offset: 0x8

Unmentioned register bits shall be set to 0.

[0] – SSIC_EN: SSI Enable. Enables and disables all DWC_ssi operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the DWC_ssi control registers when enabled, so the user needs to clear this bit before programming the registers aforementioned.

#### 3.5.1.9 *Example: Micron MT25QL128ABA Serial NOR Flash Memory*

The example below shows the configuration settings for SPI_M to access the Micron flash in quad DDR SPI mode.

```
w fcf20014 00000002 // SPI master AHB; BAUDR

w fcf20000 8080001f // SPI master AHB; CTRLR0

w fcf20010 00000001 // SPI master AHB; SER

w fcf200f4 0811421a // SPI master AHB; SPI_CTRLR0

w fcf200f8 00000001 // SPI master AHB; DDR_DRIVE_EDGE

w fcf200fc 000000cc // SPI master AHB; XIP_MODE_BITS

w fcf20100 0000006d // SPI master AHB; XIP_INCR_INST; 0x6D =
DDR Quad I/O Read

w fcf20008 00000001 // SPI master AHB; SSIENR
```

*Figure 4: Micron flash SPI_M configuration example*

#### 3.5.1.10 *Example: Winbond W959D8NFYA HyperRAM*

The Hyperbus feature (the protocol based on 8-bit DDR SPI) of SPI_M needs to be enabled in order to access HyperRAM. The example below shows the configuration settings for SPI_M to access the Winbond HyperRAM in Hyperbus mode.

```
w fcf20014 00000004 // SPI master AHB; BAUDR

w fcf20000 81c0001f // SPI master AHB; CTRLR0

w fcf20010 00000001 // SPI master AHB; SER

w fcf200f4 02153832 // SPI master AHB; SPI_CTRLR0

w fcf200f8 00000001 // SPI master AHB; DDR_DRIVE_EDGE

w fcf20008 00000001 // SPI master AHB; SSIENR
```

*Figure 5: Winbond HyperRAM SPI_M configuration example*

### 3.5.2 TRX FIFO Access

Because of its low efficiency, the user should only use TRX FIFO access when issuing single-byte command packet on SPI_M. The user needs to program the registers in Section 3.5.1 accordingly before programming DR0 at offset 0x60, the register accessing the top of TRX FIFO.

By programming DR0, SPI_M will transmit the data in DR0 on the SPI bus. The example below shows the programming sequence of sending a command byte 0x06 on the SPI bus.

```
w fcf20014 00000002 // SPI master AHB; BAUDR
w fcf20000 80800407 // SPI master AHB; CTRLR0
w fcf200f4 00000202 // SPI master AHB; SPI_CTRLR0
w fcf20010 00000001 // SPI master AHB; SER
w fcf20008 00000001 // SPI master AHB; SSIENR
w fcf20060 00000006 // SPI master AHB; DR0
```

*Figure 6: Single byte transmission command on the SPI bus configuration example*

## 3.6 PCIe Host interface

The Akida chip can be configured to operate with a PCIe Root Complex or an SPI Master Host. The host port is chosen with the PCIE_HOST_SEL pin.

*Table 2: PCIe ID*

| | Decimal | Hexadecimal |
|---|---|---|
| Brainchip PCIe Vendor ID | 7804 | 1E7C |
| Device ID | 42240 | A500 |
| Class Code | 11 | 0B |

### 3.6.1 PCIe Endpoint Interface features

- Compliant with PCI Express Gen2 Base specification
- Supports 5.0 GT/s and 2.5GT/s serial data transmission rate per Lane
- Dual Lane
- 100MHz reference clock
- Supports all non-optional features of PCI Express Base Specification Revision 4.0
- Single Function
- Embedded DMA with 2 Read Channels, 2 Write Channels, and support for Linked List mode of operation.
- AKD1500 Endpoint can be PCIe master and initiate transaction to the Root complex
- MSI with Per-Vector-Masking
- 1KB Max Payload Size
- Three BAR of type Mem 64b, with 4MB Size, map internal local address space into external PCIe memory space.

- Internal Address Translation Unit provides programmable internal local address offset, with support for BAR matched or address matched translations.
- Automatic Lane Reversal
- Support for all Link Power Management States
- Akida compatible driver available on GIT hub.

For more information, please refer to PCIe 2.1 controller documentation from Synopsys Designware cores PCI express DM controller

https://www.synopsys.com/dw/ipdir.php?ds=dwc_pci_express_controllers

### 3.6.2 Driver notes

Note that the supported Linux kernel versions are from 5.4 through 6.8. There are no plans for updates. 6.8 is current Ubuntu 24.04.1 LTS kernel version, later releases will require a driver update.

### 3.6.3 When PCIe is not used

When PCIe is not used, it is not necessary to supply power to the PCIe PHY. You can save power by using one of these methods.

- Method 1: Tie VP08, VPH33, VPTX008, VPTX108 to ground.
- Method 2: Float VP08, VPH33, VPTX008, VPTX108.

## 3.7 SPI_S (SPI Slave) Host Interface

Synopsys SPI_S IP requires no configuration register programming effort before operating: the user can access AKD1500 SPI_S interface right away after the chip is out of the reset state. But there are some aspects of the SPI protocol that the user needs to be aware of and pay attention to.

### 3.7.1 External SPI Host Serial Clock Frequency

The SPI host serial clock frequency must be **no higher** than ¼ of that of the internal SPI_S core clock. Thus, the user needs to adjust the serial clock frequency accordingly: after AKD1500 is powered up, the user needs to use slow serial clock since the SPI_S core clock is using the bypassed PLL clock output. After the PLL is enabled and drives the SPI_S core clock, the user can speed up the serial clock. Please refer to application notes for AKD1500 clock programming details.

### 3.7.2 SPI_S Bus Width Selection

AKD1500 SPI_S supports single (standard), dual, quad and octal SPI frame rates. It does NOT support Double Data Rate (DTR). The AKD1500 input pins SPI_S_MODE0 and SPI_S_MODE1 decide the SPI_S frame rate as shown in Table 3: SPI_S.

*Table 3: SPI_S Bus Width*

| SPI_S_MODE1 | SPI_S_MODE0 | SPI_S Bus Width |
|---|---|---|
| 0 | 0 | Single (1 bit) |
| 0 | 1 | Dual (2 bits) |
| 1 | 0 | Quad (4 bits) |
| 1 | 1 | Octal (8 bits) |

### 3.7.3 SPI_S Packet Structure

#### 3.7.3.1 *SPI_S Instruction Phase*

Every SPI_S frame starts with an 8-bit instruction field. The user needs to refer to **DWC_ssi databook** section 2.25.1 for details. AKD1500 supports 1, 4, 8, 16 and 32 4-byte-word data length, not 64 words stated in Table 2-5 of section 2.25.1 of **DWC_ssi databook**.

#### 3.7.3.2 *SPI_S Address Phase*

Following the instruction field comes the 24-bit address field with MSB comes in first. In order to address the internal 32-bit space in AKD1500, the 24-bit SPI_S address field is shifted leftward by two bits before being added with an offset of 0xfc000000. Or:

```
AKD1500_ADDR = 32'hfc000000 + {SPI_S_ADDR[23:0], 2'b00}
```

In another word the 24-bit SPI_S address is the word address.

#### 3.7.3.3 *SPI_S Data Phase*

SPI_S receives or transmits in frames of 32 bits. The amount of data to be transmitted or received is defined by the Data Length field of the Instruction phase.

### 3.7.4 SPI_S Access Types

#### 3.7.4.1 *SPI_S Write*

SPI_S Write packet is used for transmitting data to AKD1500 when the targeted address resides in AKD1500 (non-SPI-Master memory). Figure below shows the example pseudo code of SPI_S Write operation to XH_2PMEM at 0xFC800100.

```
SPI_TX(0x82); // Instruction; write transfer; size = 8 words
SPI_TX(0x20); // Word address[23:16]
SPI_TX(0x00); // Word address[15:08]
SPI_TX(0x40); // Word address[07:00]
SPI_TX(0x10); // Data [31:24] of word 0
SPI_TX(0x37); // Data [23:16] of word 0
SPI_TX(0x24); // Data [15:08] of word 0
SPI_TX(0x9F); // Data [07:00] of word 0
SPI_TX(0x0C); // Data [31:24] of word 1
SPI_TX(0x07); // Data [23:16] of word 1
SPI_TX(0xDF); // Data [15:08] of word 1
SPI_TX(0xC4); // Data [07:00] of word 1
…
SPI_TX(0xFE); // Data [07:00] of word 7
```

*Figure 7: SPI_S Write pseudo code*

#### 3.7.4.2 *SPI_S Write Status*

When the external SPI host writes to the AKD1500 SPI-Master memory space, a Write Status polling command is required to check on the status of the SPI_S Write. This is because SPI_S Write cannot be completed right after the command is issued due to the second serial interface at SPI_M. Figure 8: SPI_S Write Status pseudo code**Error! Reference source not found.** shows an example pseudo code of SPI_S Write Status polling operation to SPI_M at 0xFC000200. Note that there is a dummy clock cycle between the Write Status field and the read field. Please refer to Figure 2-93 and Table 2-7 of **DWC_SSI databook** for details of SPI_S Write Status command.

```
SPI_TX(0x80); // Instruction; write transfer; size = 1 word
SPI_TX(0x00); // Word address[23:16]
SPI_TX(0x00); // Word address[15:08]
SPI_TX(0x80); // Word address[07:00]
SPI_TX(0x10); // Data [31:24] of word 0
SPI_TX(0x37); // Data [23:16] of word 0
SPI_TX(0x24); // Data [15:08] of word 0
SPI_TX(0x9F); // Data [07:00] of word 0


RDATA = 0x00;
While(RDATA != 0x80) {
      SPI_TX(0xC8);  // Instruction; write status; 1 wait
cycle
      SPI_DUMMY(1);  // 1 wait dummy cycle
      SPI_RX(RDATA); // Keep reading
}
```

*Figure 8: SPI_S Write Status pseudo code*

#### 3.7.4.3 *SPI_S Read*

SPI_S Read packet is used for receiving data from AKD1500 when the targeted address resides in AKD1500 (non-SPI-Master memory). Figure below shows the example pseudo code of a 4 word SPI_S Read operation to XH_2PMEM at 0xFC800100. Note that there are 8 wait cycles between the address field and the read data field. While the example shows a read of 4 words, SPI read operations can be 1,4, 8, 16 or 32 words. See Table 4: SPI Read operation data packet length (words are 16-bits) for the corresponding read instructions. Please refer to Figure 2-95 in Section 2.25.4.2 of **DWC_SSI databook** for further details of SPI_S Read command.

```
SPI_TX(0x61);  // Instruction; read transfer; 8 wait cycles;
               // size = 4 words
SPI_TX(0x20);  // Word address[23:16]
SPI_TX(0x00);  // Word address[15:08]
SPI_TX(0x40);  // Word address[07:00]
SPI_DUMMY(8);  // 8 wait dummy cycles
SPI_RX(RDATA); // Read data [31:24] of word 0
SPI_RX(RDATA); // Read data [23:16] of word 0
SPI_RX(RDATA); // Read data [15:08] of word 0
SPI_RX(RDATA); // Read data [07:00] of word 0
SPI_RX(RDATA); // Read data [31:24] of word 1
SPI_RX(RDATA); // Read data [23:16] of word 1
SPI_RX(RDATA); // Read data [15:08] of word 1
SPI_RX(RDATA); // Read data [07:00] of word 1
…
SPI_RX(RDATA); // Read data [07:00] of word 3
```

*Figure 9: SPI_S Read pseudo code*

*Table 4: SPI Read operation data packet length (words are 16-bits)*

| SPI Read Command | Data Packet Length |
|---|---|
| 0x60 | 1 word |
| 0x61 | 4 words |
| 0x62 | 8 words |
| 0x63 | 16 words |
| 0x64 | 32 words |

#### 3.7.4.4 *SPI_S Read Request, Read Status and Read1n*

When the external SPI host reads from the AKD1500 SPI-Master memory space, a Read Request and Read Status polling commands are required. This is because the read data for SPI_S Read Request cannot be available right after the command is issued due to the second serial interface at SPI_M. Not until SPI_S Read Status command returns 0x80 can the SPI host issues SPI Read1n command to retrieve the read data. Figure below shows an example pseudo code of SPI_S Read Request and Read Status polling operation to SPI_M at 0xFC000200. Note that there is a dummy clock cycle in both Read Status and Read1n commands. Please refer to Section 2.25.4.2 of **DWC_SSI databook** for details of SPI_S Read Request, Status and Read1n commands.

```
SPI_TX(0x22); // Instruction; read request; size = 8 word
SPI_TX(0x00); // Word address[23:16]
SPI_TX(0x00); // Word address[15:08]
SPI_TX(0x80); // Word address[07:00]

RDATA = 0x00;
While(RDATA != 0x80) {
     SPI_TX(0x48);  // Instruction; read status; 1 wait
cycle
     SPI_DUMMY(1);  // 1 wait dummy cycle
     SPI_RX(RDATA); // Keep reading
}

SPI_TX(0x06); // Instruction; read1n transfer; 1 wait cycle;
              // size = 8 word
SPI_DUMMY(1); // 1 wait dummy cycle
SPI_RX(RDATA); // Read data [31:24] of word 0
SPI_RX(RDATA); // Read data [23:16] of word 0
SPI_RX(RDATA); // Read data [15:08] of word 0
SPI_RX(RDATA); // Read data [07:00] of word 0
SPI_RX(RDATA); // Read data [31:24] of word 1
SPI_RX(RDATA); // Read data [23:16] of word 1
SPI_RX(RDATA); // Read data [15:08] of word 1
SPI_RX(RDATA); // Read data [07:00] of word 1
…
SPI_RX(RDATA); // Read data [07:00] of word 7
```

*Figure 10: SPI_S Read Request and Status pseudo code*

#### 3.7.4.5 *Receive FIFO Burst Threshold Control*

By default SPI_S transmits a single 32-bit word internally every time it receives 32 data bits from the SPI bus. This is because the internal AHB bus has more bandwidth than that of SPI bus such that it is not beneficial to have data words accumulated in the SPI_S receive FIFO before transmitting them on the internal bus. Yet for some applications like programming SPI_M flash device by the external SPI_S host, it is preferred to have data words stored in the receive FIFO before transmitting them internally in bursts. SPI_S' **Receive FIFO Burst Threshold Register** at offset 0x14 assigns the threshold of the burst. Up to the value of 31 can be programmed to this register.

## 3.8 SPI Slave-to-Master (S2M) Pin-mux Access

AKD1500 supports the feature of providing slave-to-master direct SPI access via pin mux so that the external SPI host can access the external SPI memory device. Table 5 shows the pin-mux internal connection when the register field EN_SPI_S2M of AKD1500 configuration register at offset 0x18 bit [16] is set. **Only the single (standard) SPI mode is supported in the pin-mux direct access mode.**

The SPI slave interface can only receive a 24-bit address field. That 24-bit address can be expanded to 32-bits using the **SPI S2M Remap register** (address 0xfce00050) [31:23] to generate a 32-bit address field on SPI master interface. This enables an SPI host to access up to a 32-bit address range of a memory device connected to the SPI master interface. For example, if the external host needs to access the external SPI flash device at address 0x03004000, it needs to program SPI S2M Remap register with 0x03000000 and issue the SPI access command at address 0x004000.

*Table 5: slave-to-master pin-mux internal connection*

| SPI_S | Direction | SPI_M | Direction |
|---|---|---|---|
| SPI_S_2MCS0_N | Input | SPI_M_CS0_N | Output |
| SPI_S_2MCS1_N | Input | SPI_M_CS1_N | Output |
| SPI_S_SCK | Input | SPI_M_SCK | Output |
| SPI_S_IO0 | Input | SPI_M_IO0 | Output |
| SPI_S_IO1 | Output | SPI_M_IO1 | Input |

# 4 AKIDA NEURAL FABRIC

## 4.1 INTRODUCTION

The Akida Neural Fabric is self-contained and managed by a runtime library that can run on a host processor.

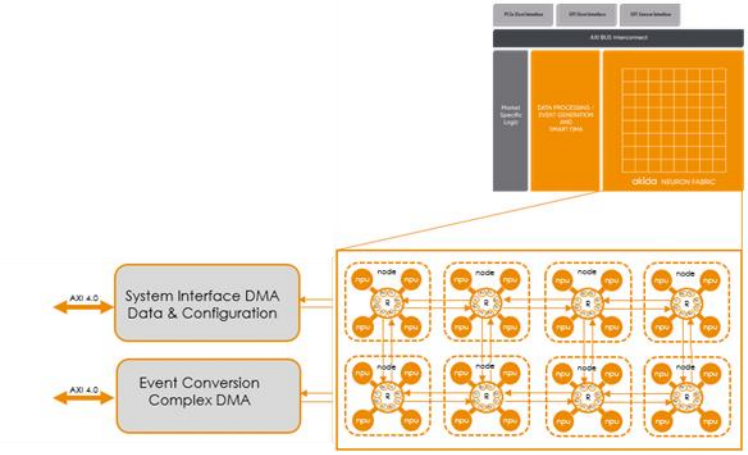



*Figure 11: NPU Mesh Network Diagram*

Groups of four Neural Processing Units are organized in a node. A mesh network transports spike packets and configuration data between the nodes. An AKD1500 Chip provides a 3 row x 3 column mesh.

## 4.2 NPU

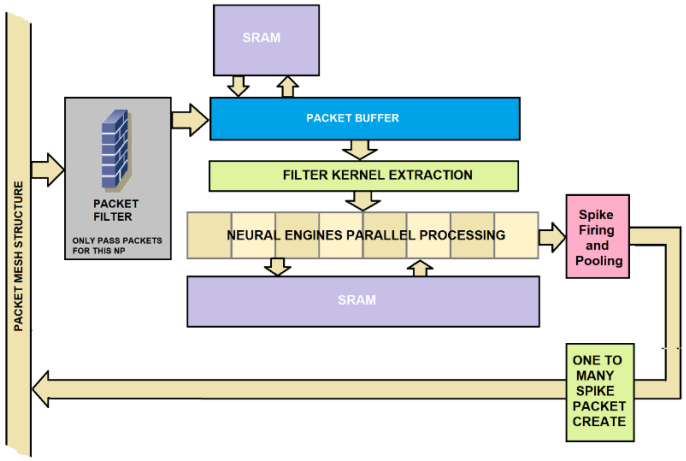



*Figure 12: NPU Block Diagram*

Each AKIDA Node has four configurable NPUs with each NPU having eight Parallel Neural Processing Execution engines (NPE) comprising four Execution Engines. NPUs can be configured for Convolutional or Fully connected operation. NPUs have 100KB of SRAM for kernel and neuron potential storage. That SRAM can be reallocated to near NPU's as required.

Other features include:

- Packet based event propagation with configurable packet size
- Multiple networks can co-exist in same device
- Multiple NODEs can be cascaded together on mesh network to create larger Networks.
- On-chip learning in event domain using BrainChip proprietary learning algorithm

## 4.3 DATA TO EVENTS CONVERTER (HIGH PRECISION CONVOLUTION - HPC)

In addition to the NPU mesh network, there is a High Precision Convolution (HPC) engine. The engine is used primarily to convert 2D arrays of data such as images, and tables into spikes for the NPU mesh network. It can also be configured to do stand-alone convolution separated from the NPU mesh.

The block diagram of the HPC is shown in the figure below. This description will use an image as the 2D input array. There are three channels that each process one of the three input components (RGB). RGB pixels enter the line buffer serially and are output by columns to the variable sized pixel grid. The pixel grid has the same dimensions as the kernel used for the convolution. The operand router connects a pixel and its associated weight to a multiplier in the MAcc. There are nine 8-bit x 8-bit multipliers per channel. If more than 9 multiplies per channel are required, then the nine multiplier subtotals are accumulated by the Acc block, and it will take multiple clock cycles to calculate the kernel potential. The merge block sums the potentials of each channel to give the total potential.

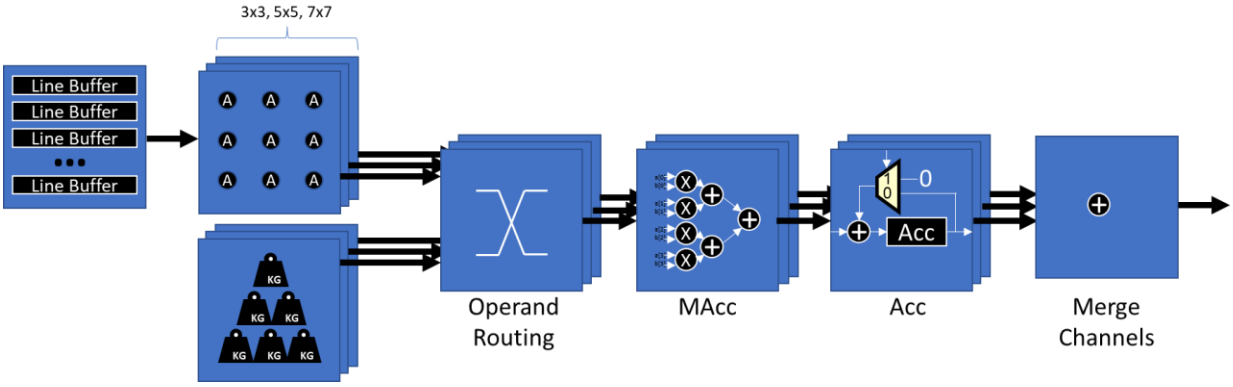



*Figure 13: Convolution Math*

| | |
|---|---|
| Line Buffer | Groups input data into lines of 3,5,7, or 16 lines depending on the kernel size |
| Vertical Stride | Discards line groups that are not used because of vertical striding |
| Horizontal Padding | Adds pad value pixels to the left and or right margins based on configuration |
| Horizontal Stride | Skips horizontally over groups of pixels not used because of horizontal striding |
| Math | Calculates potential of each filter |
| Activation | Converts potentials to activations |
| Horizontal Pooling | Horizontal Stride of 2 pooling (optional) |
| Vertical Pooling | Vertical Stride of 2 pooling (optional) |
| Pixel to Spike | Converts the sequential stream to an event-based spike |

*Figure 14: Simplified Pipeline*

The HPC engine can be configured with the following parameters.

| Parameter | Description |
|---|---|
| Kernel size | 3x3, 5x5, 7x7 |
| Stride | # of pixels or lines to skip between kernel calculations |
| Padding | Top, bottom, left, right side padding |
| Pad Data | Data pattern used for pad |
| Activation | See NPU for activation parameter details |
| Pooling | Optional 1x2, 2x1, or 2x2 |
| | |
| | |

| Kernel Size | Strides | Supported Padding Configurations (left/top,right/bottom) |
|---|---|---|
| 3x3 | 1,2,3 | (0,0), (0,1), (1,1), (1,0) |
| 5x5 | 1,2,3,4,5 | (0,0), (1,0), (0,1), (1,2), (2,2), (2,1), (1,0) |
| 7x7 | 1,2,3,4,5,6,7 | (0,0), (1,0), (0,1), (1,2), (2,2), (2,3), (3,3), (3,2), (2,1), (1,0) |

Note: The configurations in orange are left priority padding. Brainchip software follows right priority padding. The hardware supports either left or right priority padding.

| Input Channels | Kernel Size | Max Filters | Clks per filter | MACC's/Clk |
|---|---|---|---|---|
| 3 (RGB) | 3x3 | 192 | 1 | 27 |
| | 5x5 | 64 | 3 | 25 |
| | 7x7 | 32 | 6 | 24.5 |
| 1 (Gray) | 3x3 | 512* | 1** | 9 |
| | 5x5 | 192 | 1 | 25 |
| | 7x7 | 96 | 2 | 24.5 |

* This number is limited to 512 in order to keep the filter variable at 9-bits.

** There are enough multipliers to calculate three 1-channel 3x3 kernels per clock, but the pipeline is limited to processing only one filter per clock.

# 5 DESIGN DECISIONS

## 5.1 HOST SELECTION

The AKD1500 can be controlled through a 2-lane Gen2 PCIe or SPI host interface. The choice of host will depend on your cost and performance targets.

In PCIe Mode, the AKD1500 can be a PCIe Master (not to be confused with PCIe Root Complex), and can initiate transactions to access data directly from the Host memory. This allows to run applications requiring more memory than the available internal SRAM.

## 5.2 EXTERNAL FLASH MEMORY

Storing models in the AKD1500's external flash memory can improve system performance for some applications. The flow chart below, Figure 15: Is flash required? , will help to determine if the external flash is required for your application.

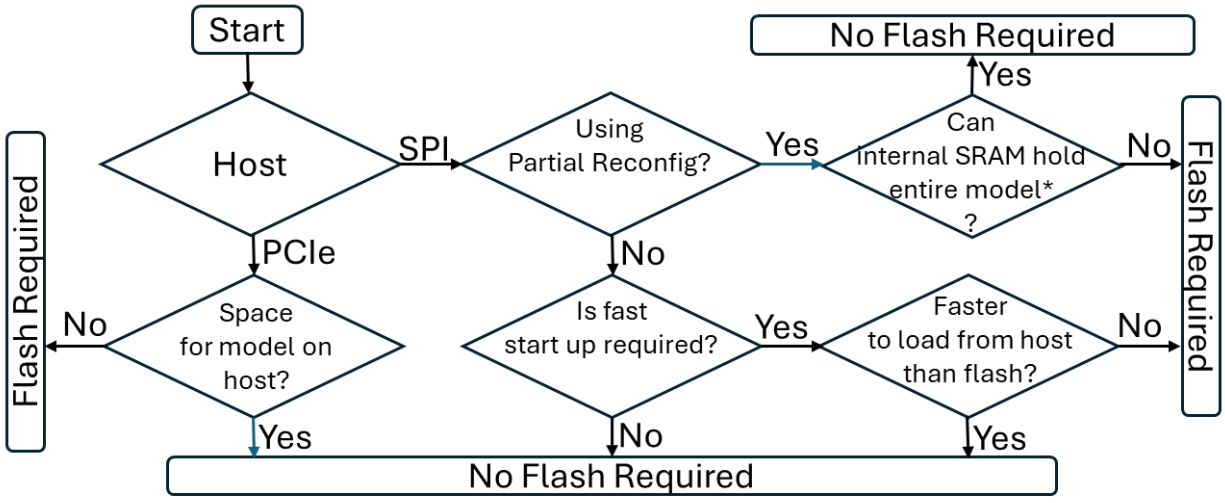



*Figure 15: Is flash required?*

# 6 System Memory Map

There are two AKD1500 system memory maps, one for PCIe host and Akida core and another for SPI host.

Notes

- Blocks with color are alias that have 2 separate address ranges mapped to the same physical module.
- SPI Slave IP can only access 0xFCxx_xxxx memory space.
- Every bus master has limitation on which bus slave it can access. See “OK Master” column for its permission.
- Accesses to a reserved area will have un-defined behavior and error status response.

Table 6: System Memory Map

| Name | Address | Size | OK Master | Note |
|---|---|---|---|---|
| | 0x0000_0000 0x1FFF_FFFF | | | reserved |
| XH_2PMEM | 0x2000_0000 0x7FFF_FFFF | 1 MB | PCIe, Akida | |
| SPI Master Memory | 0x8000_0000 0xBFFF_FFFF | 1 GB | PCIe, Akida | External SPI Master flash / memory direct access. Address[31] is inverted before going to SPI Master IP; so it starts with 0. |
| PCIe Slave | 0xC000_0000 0xEFFF_FFFF | 1 GB | Akida | To direct access the PCIe host memory. |
| | | | | reserved |
| PCIe DBI | 0xF8C0_0000 0xF8FF_FFFF | 4 MB | PCIe | |
| | | | | reserved |
| SPI Master Memory | 0xFC00_0000 0xFC7F_FFFF | 8 MB | SPI | Use System Conf. Reg. 0x50 to change the upper address[31:23] of SPI Master IP slave AHB bus. |
| XH_2PMEM | 0xFC80_0000 0xFCBF_FFFF | 1 MB | PCIe, SPI, Akida | Alias of 0x2000_0000 area |
| Akida APB | 0xFCC0_0000 0xFCCF_FFFF | 1 MB | PCIe, SPI | See Akida Core spec. |
| | | | | reserved |
| Sys. Conf. Register | 0xFCE0_0000 0xFCE0_0FFF | 4 KB | PCIe, SPI | Alias of 0xF000_0000 |
| Clock Reset | 0xFCE0_1000 0xFCE0_1FFF | 4 KB | PCIe, SPI | Alias of 0xF000_1000 |

| | | | | |
|---|---|---|---|---|
| | | | | |
| | | | | reserved |
| SPI Slave CFG | 0xFCF1_0000<br>0xFCF1_FFFF | 64 KB | PCIe, SPI | |
| SPI Master CFG | 0xFCF2_0000<br>0xFCF2_FFFF | 64 KB | PCIe, SPI | Alias of<br>0xF3FF_0000 |
| | | | | reserved |
| | | | | |

# 7 System Configuration Registers

**Acronym**:

- RO: Read Only
- RW: Read and Write
- R2C: Read to Clear

The following registers can be accessed through Mesh Network or SPI bus.

*Table 7: Chip Info (offset 0x10)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| Version | 3:0 | RO | 0x1 | The version of the AKD1500 Chip.<br>"1" → the first version. |
| | 7:4 | RO | 0x0 | reserved |
| OP_MODE | 9:8 | RO | 0x? | This read the status of the chip strap input pin value for operation mode decision. |
| | 11:10 | RW | 0x0 | reserved |
| PCIE_HOST_SEL | 12 | RO | 0x? | '0' → SPI Host mode. PCIe module is power down.<br>'1' → PCIe Host mode. |
| SEL_CLK | 13 | RO | 0x? | '0' → Xtal OSC is used as input.<br>'1' → Digital clock is used as input. |
| | 31:14 | RO | 0x0 | Reserved |

*Table 8: Dual Port Memory Controller (offset 0x14)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| LP_DELAY | 7:0 | RW | 0x02 | Number of idle clocks before putting SRAM blocks into low power (LP) mode. It is count of 256 clock granularities.<br><br>"0x01" → 1 x 256 clock delay.<br>"0x02" → 2 x 256 clock delay.<br>… |
| LP_REC_TIME | 11:8 | RW | 0x3 | Recovery time from LP mode (GF 32Kx32 SRAM needs 6.2ns). |
| | 14:12 | | 0x0 | reserved |
| LP_MODE_EN | 15 | RW | 0x1 | '0' = SRAM blocks are never put into low power mode. |

| | | | | |
|---|---|---|---|---|
| | | | | ‘1’ = allowed SRAM blocks to be put into low power mode. |
| X_AGING_DELAY | 19:16 | RW | 0x2 | Number of continuous transfers before AXI access loses the priority to AHB access. |
| | 23:20 | RO | 0x0 | reserved |
| H_AGING_DELAY | 27:24 | RW | 0x4 | Number of continuous transfers before AHB access loses the priority to AXI access. |
| | 31:28 | RO | 0x0 | reserved |

*Table 9: Control Signals (offset 0x18)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| EN_BUS_RESP | 0 | RW | 0x0 | To configure 2P Memory Controller bus interfaces.<br><br>‘0’ → Never send error response to AXI or AHB accesses.<br><br>‘1’ → Send error response to AXI or AHB accesses if address decoding is out of range with EN_MEM_RANGE = 1. |
| EN_MEM_RANGE | 1 | RW | 0x0 | To configure 2P Memory Controller bus interfaces.<br><br>‘0’ → SRAM address range is alias. No address range check.<br><br>‘1’ → SRAM address range is checked, and no SRAM write is performed on out-of-range writes. |
| | 3:2 | R | 0x0 | reserved |
| X_2HBM_NO_WBURST | 4 | RW | 0x0 | To enable or disable AHB write burst in AXI to AHB converter of the BM.<br><br>‘0’ → AXI to AHB converter can use AHB write burst output when AXI has no empty write.<br><br>‘1’ → Translate all the AXI write burst to AHB single accesses. |
| X_2HBM_NO_RBURST | 5 | RW | 0x0 | To enable or disable AHB read burst in AXI to AHB converter of the BM.<br><br>‘0’ → AXI to AHB converter can use AHB read burst output. |

| | | | | |
|---|---|---|---|---|
| | | | | '1' → Translate all the AXI read burst to AHB single accesses. |
| | 15:6 | R | 0x0 | Reserved |
| EN_SPI_S2M | 16 | RW | 0x0 | To configure SPI master bus to connect to SPI slave bus directly.<br><br>'0' → external SPI master bus is connected to SPI Master module (normal operation).<br><br>'1' → external SPI master bus is connected to SPI slave bus directly; so the host at SPI slave bus can access SPI master bus using 1 bit protocol. |
| SPIM_DQS_PD | 17 | RW | 0x1 | '1' → Enable SPI Master Data Strobe input pulled down. Use it when DQS pin is not used.<br><br>'0' → Do not enable SPI Master Data Strobe input pulled down. |
| | 19:18 | | | |
| SPIM_DO_SWAP | 20 | RW | 0x0 | '0' → The AHB Write Data bus (h_2spim_hwdata) to SPI Master has NO byte swap.<br><br>'1' → The AHB Write Data bus (h_2spim_hwdata) to SPI Master has byte swap. |
| SPIM_DI_SWAP | 21 | RW | 0x1 | '0' → The AHB Read Data bus (h_2spim_hrdata) from SPI Master has NO byte swap.<br><br>'1' → The AHB Read Data bus (h_2spim_hrdata) from SPI Master has byte swap. |
| | 31:22 | R | 0x0 | reserved |

*Table 10: GPIO Pull Up Enable (offset 0x20)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| GPIO_PU | 27:0 | RW | 0x0 | '1': enable GPIO[i] pad pull-up resistor.<br>'0': disable GPIO[i] pad pull-up resistor. |

| | | | | |
|---|---|---|---|---|
| | | | | Note: These signals connect to the IO cell PU directly; no gating. |
| | 31:28 | RO | 0x0 | reserved |

*Table 11: GPIO Pull Down Enable (offset 0x24)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| GPIO_PD | 27:0 | RW | 0x0 | '1': enable GPIO[i] pad pull-down resistor.<br>'0': disable GPIO[i] pad pull-down resistor.<br><br>Note: These signals connect to the IO cell PD directly; no gating. |
| | 31:28 | RO | 0x0 | reserved |

*Table 12: GPIO Output Data (offset 0x30)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| GPIO IN | 27:0 | R | 0x? | Status of GPIO Pin |
| | 31:28 | RO | 0x0 | reserved |

*Table 13: GPIO Output Data (offset 0x34)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| GPIO OUT | 27:0 | RW | 0x0 | GPIO Output value |
| | 31:28 | RO | 0x0 | reserved |

*Table 14: GPIO Output Data (offset 0x38)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| GPIO OE (active high) | 27:0 | RW | 0x0 | Output Enable for each GPIO Pin<br><br>'1': The corresponding GPIO output pin is enabled.<br><br>'0': The corresponding GPIO output pin is tri stated. |
| | 31:28 | RO | 0x0 | reserved |

*Table 15: GPIO Mux Enable (offset 0x3C)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| GPIO MUX OK | 27:0 | RW | 0x0 | To control GPIO to use shared IO.<br><br>‘1’: The shared IO pin is selected for the corresponding GPIO signal.<br>‘0’: The shared IO pin is selected for another signal (not GPIO signal).<br><br>Note: these signals do not control PU or PD. |
| | 31:28 | RO | 0x0 | reserved |

*Table 16: GPIO Drive Strength (offset 0x40)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| GPIO_DRV | 27:0 | RW | 0x0 | Drive strength select for each GPIO pin:<br>1’b0: 6 mA<br>1’b1: 12 mA |
| | 31:28 | RO | 0x0 | reserved |

*Table 17: SPI Slave to SPI Master Memory Remap (offset 0x50)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| | 22:0 | | | reserved |
| SPI_S_REMAP | 31:23 | RO | 0x0 | This allows SPI Slave IP to access 1 GB external memory in SPI Master bus using 8 MB of window in 0xFC00_0000 to 0xFC7F_FFFF range.<br><br>The [31:23] bits (8 MB range) become the new haddr[31:23] of SPI Master IP slave AHB address bus. |

*Table 18: SPI IO Config (offset 0x54)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| SPIS_RXEN_ON | 0 | RW | 0x0 | ‘1’: Force the IO input buffer is always enabled for SPI Slave IO.<br>‘0’: Normal mode, IO input buffer is enabled only when is needed for SPI Slave IO. |
| SPIM_RXEN_ON | 1 | RW | 0x0 | ‘1’: Force the IO input buffer is always enabled for SPI Master IO. |

| | | | | |
|---|---|---|---|---|
| | | | | '0': Normal mode, IO input buffer is enabled only when is needed for SPI Master IO. |
| SPIM_CLK_ON | 2 | RW | 0x1 | '1': Enable SPI Master CLK output.<br>'0': Disable SPI Master CLK output. |
| SPIM_CLKX_ON | 3 | RW | 0x0 | '1': Enable SPI Master CLKX output.<br>'0': Disable SPI Master CLKX output. |
| SPIS_IO_SLWEN | 4 | RW | 0x1 | '1': Enable SPI Slave IO slew rate control.<br>'0': Disable SPI Slave IO slew rate control. |
| | 7:5 | | | reserved |
| SPIM_IO_SLWEN | 8 | RW | 0x1 | '1': Enable SPI Master IO slew rate control.<br>'0': Disable SPI Master IO slew rate control. |
| SPIM_CS_SLWEN | 9 | RW | 0x1 | '1': Enable SPI Master CS slew rate control.<br>'0': Disable SPI Master CS slew rate control. |
| SPIM_RST_SLWEN | 10 | RW | 0x1 | '1': Enable SPI Master RST slew rate control.<br>'0': Disable SPI Master RST slew rate control. |
| SPIM_DQS_SLWEN | 11 | RW | 0x1 | '1': Enable SPI Master DQS slew rate control.<br>'0': Disable SPI Master DQS slew rate control. |
| | 15:12 | | | reserved |
| SPIS_IO_DRV | 17:16 | RW | 0x3 | SPI Slave IO drive strength select:<br>2'b00: 3 mA<br>2'b01: 6 mA<br>2'b10: 9 mA<br>2'b11: 12 mA |
| | 21:18 | | | reserved |
| SPIM_DQS_DRV | 23:22 | RW | 0x3 | SPI Master DQS drive strength select:<br>2'b00: 3 mA<br>2'b01: 6 mA<br>2'b10: 9 mA<br>2'b11: 12 mA |
| SPIM_IO_DRV | 25:24 | RW | 0x3 | SPI Master IO drive strength select:<br>2'b00: 3 mA<br>2'b01: 6 mA<br>2'b10: 9 mA<br>2'b11: 12 mA |
| SPIM_CS_DRV | 27:26 | RW | 0x3 | SPI Master CS drive strength select:<br>2'b00: 3 mA<br>2'b01: 6 mA<br>2'b10: 9 mA<br>2'b11: 12 mA |
| SPIM_RST_DRV | 29:28 | RW | 0x3 | SPI Master RST drive strength select:<br>2'b00: 3 mA<br>2'b01: 6 mA<br>2'b10: 9 mA<br>2'b11: 12 mA |
| SPIM_CLK_DRV | 31:30 | RW | 0x3 | SPI Master CLK drive strength select:<br>2'b00: 3 mA<br>2'b01: 6 mA<br>2'b10: 9 mA<br>2'b11: 12 mA |

*Table 19: Control Signals (offset 0x60)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| EP_APP_LTSSM_EN | 0 | RW | 0x1 | '1'= Start PCIe EP Training for PCIE_HOST_SEL=1.<br><br>Note:<br>When PCIE_HOST_SEL=0 this signal is always 0. |
| | 31:1 | R | | reserved |

*Table 20: REG2DBI Control (offset 0x70)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| REG2DBI_EN | 0 | RW | 0x0 | '1'= Switch PCIe AXI DBI bus to REG2DBI bus.<br><br>'0'= Normal operation. PCIe AXI DBI bus is connected to the AXI bus matrix. |
| | 3:1 | R | | reserved |
| REG2DBI_WDONE | 4 | R | 0x1 | '1'= AXI write transfer from REG2DBI done.<br>'0'= AXI write transfer from REG2DBI pending. |
| REG2DBI_RDONE | 5 | R | 0x1 | '1'= AXI write transfer from REG2DBI done.<br>'0'= AXI write transfer from REG2DBI pending. |
| | 31:6 | R | | reserved |

*Table 21: REG2DBI Read Data (offset 0x74)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| REG2DBI_RDATA | 31:0 | R | 0x0 | Last AXI Read Data bits. |

*Table 22: REG2DBI Write Data (offset 0x78)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| REG2DBI_WDATA | 31:0 | RW | 0x0 | AXI Write Data bits. |

*Table 23: REG2DBI Command and Address (offset 0x7C)*

| Field | Bits | Type | Reset value | Description |
|---|---|---|---|---|
| REG2DBI_ADDR | 21:0 | RW | 0x0 | AXI AWADDR[21:0] or ARADDR[21:0] address bits. |

| | | | | |
|---|---|---|---|---|
| | 23:22 | R | | reserved |
| REG2DBI_WSTRB | 27:24 | RW | 0xF | AXI WSTRB |
| REG2DBI_WREQ | 28 | W | 0x0 | ‘1’=to generate one REG2DBI AXI write transfer.<br>‘0’=no AXI write request.<br><br>Note: a write with this bit = ’1’ will create 1 clock pulse to start AXI write transfer. |
| REG2DBI_RREQ | 29 | W | 0x0 | ‘1’=to generate one REG2DBI AXI read transfer.<br>‘0’=no AXI read request.<br><br>Note: a write with this bit = ’1’ will create 1 clock pulse to start AXI read transfer. |
| | 31:30 | R | | reserved |

# 8 ELECTRICAL SPECIFICATION

## 8.1 ABSOLUTE MAXIMUM CONDITIONS

*Table 24: Absolute Max Conditions*

| Symbol | Parameter | Min | Max | Units | Notes |
|---|---|---|---|---|---|
| VDD08 | Core Power | -0.5 | 1.000 | V | |
| VP08, VPTX008, VPTX108 | PCIe Analog 0.8V | -0.5 | 1.000 | V | |
| VPH33 | PCIe PHY high voltage | -0.5 | 3.400 | V | |
| DVDD08 | PCIe Digital Core Power | -0.5 | 1.000 | V | |
| AVDD18 | PCIe Analog 1.8V | -0.5 | 2.000 | V | |
| AVD_PLL | Analog PLL Power | -0.5 | 1.000 | V | |
| VDE18 | 1.8V Only Power | -0.5 | 2.000 | V | |
| 1.8V PCIe | Dual Voltage Pad 1.8V | -0.5 | 2.000 | V | |
| 3.3V PCIe | Dual Voltage Pad 3.3V | -0.5 | 3.400 | V | |
| $T_{STG}$ | Storage temperature | –65 | 150 | °C | |
| $T_{LEAD}$ | Lead temperature during soldering | – | See note 1 | °C | |
| $V_{ESD}$ | Electrostatic discharge voltage (human body model) | –2000 | 2000 | V | 2, 3 |

Notes: 1. Compliant with JEDEC Standard J-STD-020C (for small-body, Sn-Pb or Pb assembly), RoHS, and the European directive on Restrictions on Hazardous Substances (RoHS) 2002/95/EU.

2. All specified voltages are with respect to $V_{SS}$. During infrequent, nonperiodic transitions, the voltage potential between $V_{SS}$ and the $V_{CC}$ may undershoot to –2.0V for periods less than 20ns, or overshoot to $V_{CC,max}$ + 2.0V for periods less than 20ns.

3. JEDEC Standard JESD22-A114A (C1 = 100pF, R1 = 1500Ω, R2 = 500Ω)

## 8.2 RECOMMENDED OPERATING CONDITIONS – DEVICE SPECIFIC

### 8.2.1 DC Electrical Characteristics

All specifications are measured at the package ball

*Table 25: Power*

| Parameter | Symbol | Rating | | | Units |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| Core - VDDI Power Supply (Dig) | $V_{VDD08}$ | 0.76 | 0.80 | 0.84 | V |
| | $I_{VDD08}$ | | 1.25 | 2.0 | A |
| LVCMOS 1.8V IO | $V_{VDE18}$ | 1.71 | 1.8 | 1.89 | V |
| | $I_{VDE18}$ | | | 310 | mA |
| VP08 | $V_{VP08}$ | 0.76 | 0.80 | 0.86 | V |
| | $V_{VP08\_SLEW}$ | | | 0.1 | V/µS |
| | $I_{VP08}$ | | | 76.3 | mA |
| VPTX008, VPTX108 | $V_{VPTX108}$, $V_{VPTX108}$ | 0.76 | 0.80 | 0.86 | V |
| | $V_{VPTXX08\_SLEW}$ | | | 0.1 | V/µS |
| | $I_{VPTX008}$, $I_{VPTX008}$ | | | 33.8 | mA |
| VPH33 | $V_{VPH33}$ | 3.14 | 3.3 | 3.56 | V |
| | $V_{VPH33\_SLEW}$ | | | 0.1 | V/µS |
| | $I_{VPH33\ (x2\ lane)}$ | | 30 | 37.8 | mA |
| | $I_{VPH33\ (x1\ lane)}$ | | 21 | 35 | mA |
| Master Clock PLL | $V_{DDAVDPLL}$ | 0.76 | 0.80 | 0.84 | V |
| | $I_{VDDAPLL}$ | | 2.5 | 5 | mA |
| Junction Temperature | Tj | -40 | | 125 | ◦C |

### 8.2.2 6.2.1 IO Characteristics

| Parameter | Symbol | Min | Typ | Max | Units | Notes |
|---|---|---|---|---|---|---|
| Input Low Level | $V_{IL}$ | 0 | | 0.54 | V | |
| Input High Level | $V_{IH}$ | 1.26 | | 1.8 | V | |
| Hysteresis | $V_{HYS}$ | 0.18 | | | V | |
| Output Low Level | $V_{OL}$ | 0 | | 0.45 | V | |
| Output High Level | $V_{OH}$ | 0.99 | | 1.8 | V | |
| Leakage | $I_L$ | | | 10 | µA | |
| Pullup/Pulldown | $R_{PUD}$ | | 50 | | Ω | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

### 8.2.3 AC Electrical Characteristics

*Table 26: Reference Clocks*

| Parameter | Symbol | Rating | | | Units | Notes |
|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | | |
| External Clock input for software compatibility | $F_{REF\ API}$ | 25 | 25 | 25 | MHz | 1,2 |
| External Clock input - custom | $F_{REF\ custom}$ | 7 | | 200 | MHz | 1,2 |
| | | | | | | |
| Reset pulse width | $T_{RST(Min)}$ | | | 1 | µS | |
| Xtal Frequency | $F_{XTAL}$ | 20.1 | 25 | 30 | MHz | 1,2 |
| Xtal Internal Shunt Cap. | $C_0$ | | 7 | | pF | 4 |
| Xtal Total Load Cap. | $C_L$ | | 12 | | pF | 4 |
| Xtal Feedback Resistor | $R_F$ | | 1M | | Ω | 4 |
| Xtal Equivalent Resistance | ESR | | 60 | | Ω | |
| Xtal Frequency Tolerance | $F_{XTAL\_TOL}$ | | | 100 | PPM | |

Notes:

1. For compatibility with existing software, the xtal or clock input must be 25MHz. With software customization, it is possible to change the $F_{REF}$ to other ranges. If not using the software compatible value, you must bypass the PLL until it has been programmed to the desired custom settings.
2. Crystal and clock input are mutually exclusive. Only one clocking method is allowed.
3. Reset pulse requirement is driven by PLL specifications
4. Refer to Figure 16: Recommended Xtal Configuration. $C_L = C_{PCB\ STRAY} + (C_{L1} \times C_{L2})/(C_{L1}+C_{L2})$. $C_0$ and ESR are internal properties of the xtal.
5. Xtal values used for qualification: $C_{L1} = C_{L2} = 7pF$, $R_F = 1M\ \Omega$, Xtal = ABM10W-25.0000MHZ-7-B1U-T3. Optimum values will vary depending on board layout.

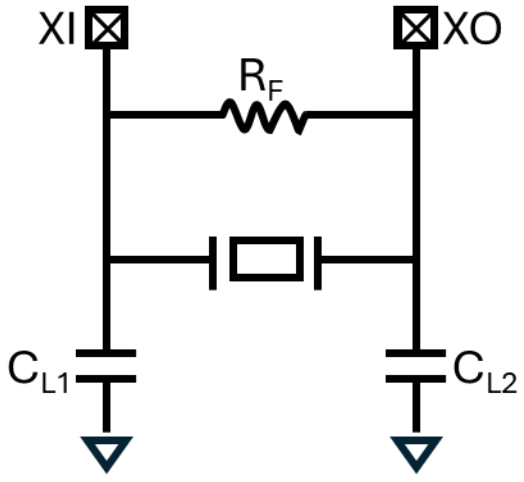
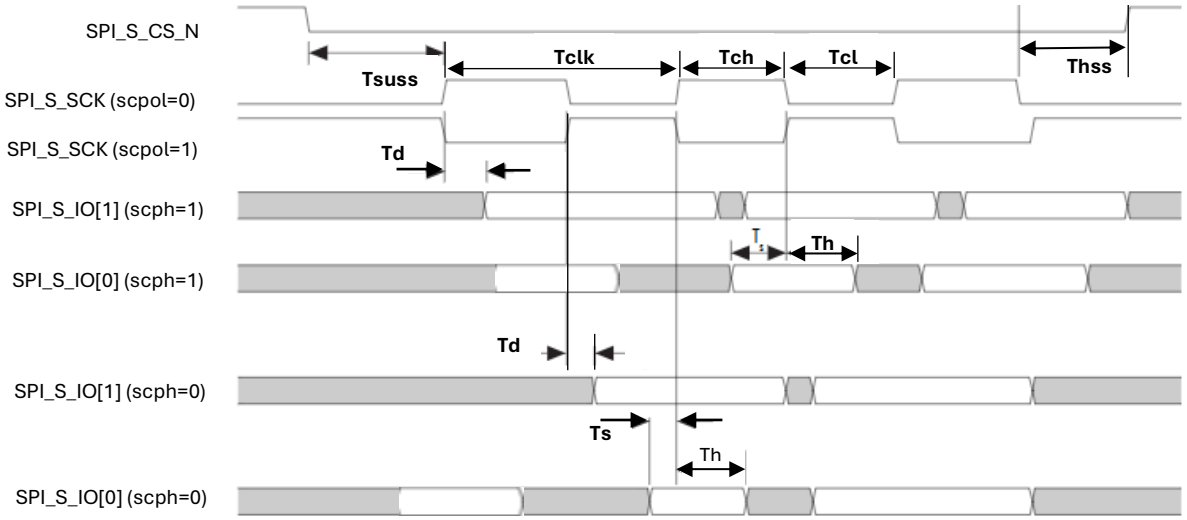



*Figure 16: Recommended Xtal Configuration*

# 9 TIMING SPECIFICATIONS

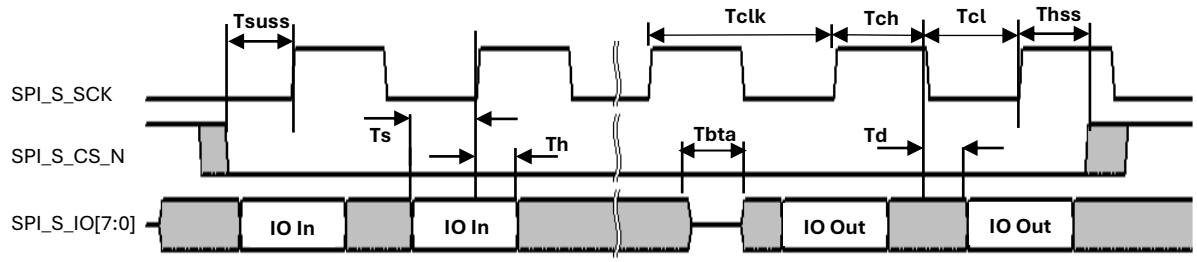
## 9.1 SPI SLAVE INTERFACE

### 9.1.1 SPI Slave Single Lane Mode




| Symbol | Description | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $T_{clk}$ | SCK Clock Period | 15 | — | ns | 1 |
| $T_{ch}$ | SCK Clock High Time | 7.5 | — | ns | |
| $T_{cl}$ | CLK Clock Low Time | 7.5 | — | ns | |
| $T_s$ | SPI_S_IO0 (MOSI) Setup time | 2 | — | ns | |
| $T_h$ | SPI_S_IO0 (MOSI) Hold time | 1 | — | ns | |
| $T_d$ | SPI_S_IO1 (MISO) Output Delay | 1.5 | 5.5 | ns | |
| $T_{suss}$ | Setup time SPI_SS valid before first clock edge | 5 | — | ns | |
| $T_{hss}$ | Hold time SPI_SS valid after last clock edge | 3 | — | ns | |
| $C_{OUT}$ | Maximum capacitive load | - | 10 | pf | |

### 9.1.2 SPI Slave Dual/Quad/Octal Mode



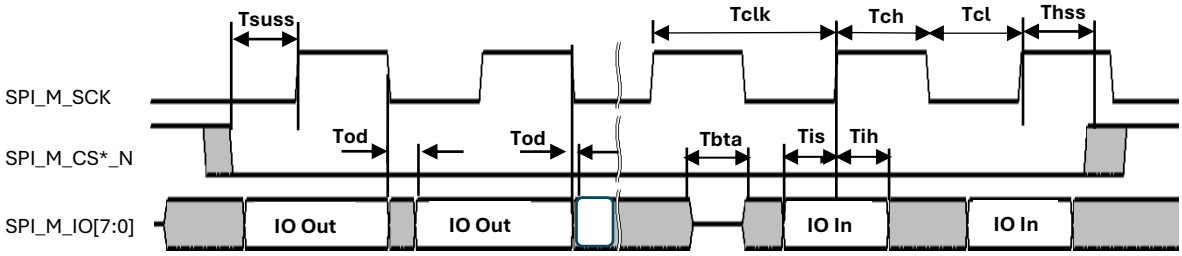

| Symbol | Description | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $T_{clk}$ | SCK Clock Period | 15 | — | ns | 1 |
| $T_{ch}$ | SCK Clock High Time | 7.5 | — | ns | |
| $T_{cl}$ | CLK Clock Low Time | 7.5 | — | ns | |
| $T_s$ | SPI_S_IO[7:0] Input Setup time | 2 | — | ns | |
| $T_h$ | SPI_S_IO[7:0] input Hold time | 1 | — | ns | |
| $T_d$ | SPI_S_IO[7:0] Output Delay | 1.5 | 5.5 | ns | |
| $T_{suss}$ | Setup time SPI_SS valid before first clock edge | 5 | — | ns | |
| $T_{hss}$ | Hold time SPI_SS valid after last clock edge | 3 | — | ns | |
| $T_{bta}$ | Bus Turnaround time | - | 3.5 | ns | |
| $C_{OUT}$ | Maximum capacitive load | - | 10 | pf | |

Notes:

1. The SCK frequency is also limited by internal clock settings. Please refer to section 3.7.1.

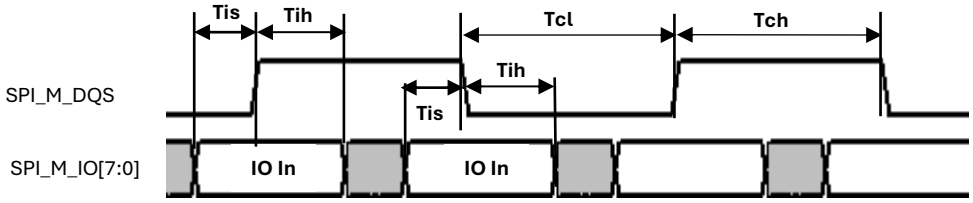
## 9.2 SPI MASTER INTERFACE

### 9.2.1 SPI Master Single Data Rate Mode



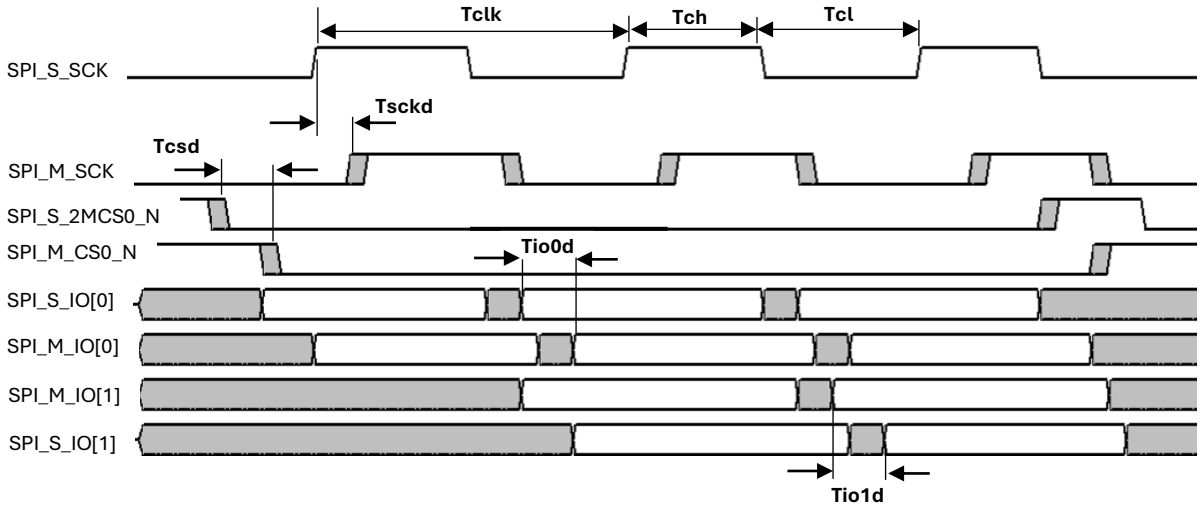
| Symbol | Description | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $T_{clk}$ | SCK Clock Period | 10 | — | ns | |
| $T_{ch}$ | SCK Clock High Time | 5 | — | ns | |
| $T_{cl}$ | CLK Clock Low Time | 5 | — | ns | |
| $T_{od}$ | SPI_M_IO[7:0] Output Delay | 0 | 1.8 | ns | |
| $T_{is}$ | SPI_M_IO[7:0] input Setup time | 1.25 | — | ns | |
| $T_{ih}$ | SPI_M_IO[7:0] Input Hold Time | 0 | - | ns | |
| $T_{suss}$ | Setup time SPI_SS valid before first clock edge | 1.8 | — | ns | |
| $T_{hss}$ | Hold time SPI_SS valid after last clock edge | 5 | — | ns | |
| $T_{bta}$ | Bus Turnaround time | - | 3.5 | ns | |
| $C_{OUT}$ | Maximum capacitive load | - | 10 | pf | |

### 9.2.2 SPI Master DQS Dual Data Rate




| Symbol | Description | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $T_{clk}$ | SCK Clock Period | 10 | — | ns | |
| $T_{ch}$ | SCK Clock High Time | 5 | — | ns | |
| $T_{cl}$ | CLK Clock Low Time | 5 | — | ns | |
| $T_{is}$ | SPI_M_IO[7:0] input Setup time | 0.5 | — | ns | |
| $T_{ih}$ | SPI_M_IO[7:0] Input Hold Time | 1.7 | - | ns | |

## 9.3 SPI DIRECT ACCESS (SPI SLAVE TO SPI MASTER FEEDTHROUGH)




| Symbol | Description | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $T_{clk}$ | SCK Clock Period | N/A | — | ns | 1 |
| $T_{ch}$ | SCK Clock High Time | N/A | — | ns | |
| $T_{cl}$ | CLK Clock Low Time | N/A | — | ns | |
| $T_{sckd}$ | SPI_S_SCK to SPI_M_SCK Delay | 1.5 | 4 | ns | |
| $T_{csd}$ | SPI_S_2MCS0_N to SPI_M_CS0_N Delay | 3.5 | 6.5 | ns | |
| $T_{io0d}$ | SPI_S_IO[0] to SPI_M_IO[0] Delay | 3.5 | 6.5 | ns | |
| $T_{io1d}$ | SPI_M_IO[1] to SPI_S_IO[1] Delay | 2.5 | 4.5 | ns | |
| $C_{OUT}$ | Maximum capacitive load | - | 10 | pf | |

Notes:

1. The AKD1500 is only adding delays to the SPI signals. The minimum clock period depends on the Host's SPI Master and the Device's SPI Slave specifications.

# 10 Power Sequence Recommendations

## 10.1 Power On Sequence

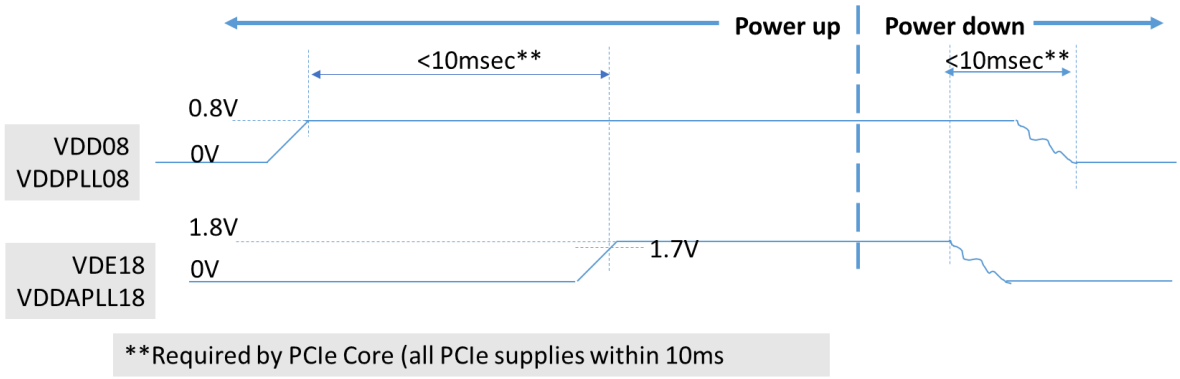



*Figure 17: Power Sequence Diagram*

### 10.1.1 General

Turn on from internal power supply to external power supply. (i.e. 0.8V core voltage first, VDE18 last.)

## 10.2 Power off sequence

### 10.2.1 General

Turn off from external power supply to internal power supply (i.e. VDE18 shuts off first, 0.8 shuts off last).

# 11 LAYOUT CONSIDERATIONS

## 11.1 GROUND

We recommend a solid common ground plane for all analog and digital signals.

## 11.2 RECOMMENDED TEST POINTS

- PWR_GOOD. This will allow verification that the AKD1500 is out of reset.
- The 0.8V, 1.8V, and 3.3V (if applicable) power supplies. This will allow verification that the device is getting correct and clean power.
- All SPI_S and SPI_M signals that are used in your design. This will help to pinpoint HW/SW issues and facilitate faster bring up.
- SPI_S_SCKX. This pin will output the system clock divided by 16 when OP_MODE1 is pulled high. This is a sign of life indicator that is useful for design bring up.
- OSC_XI pin. Recommended if the device clock source is external. If the xtal is used, the xtal trim caps can be used as test points to verify the xtal is running.

## 11.3 DECOUPLING

We recommend one 01005 bypass cap on each power ball under the AKD1500. The 01005 body size allows the caps to be located directly between the vias for the power and ground balls of the device. This will minimize inductance and maximize the effectiveness of the bypass caps. There are a total of 43 such caps. If this is a cost issue for your design and you choose to modify this recommendation, you should run a power analysis simulation of your PCB design to guarantee power integrity. We recommend always adding the footprints for fully bypassing every power pin as described above, even if you don't intend to install the parts. This will make debug possible if it becomes necessary.

Analog supplies (VDDPLL08, VDDAPLL18, VP08, VPH33, VPTX008, VPTX108) should be filtered using a circuit similar to the one shown in Figure 18: Filtering analog supply lines.

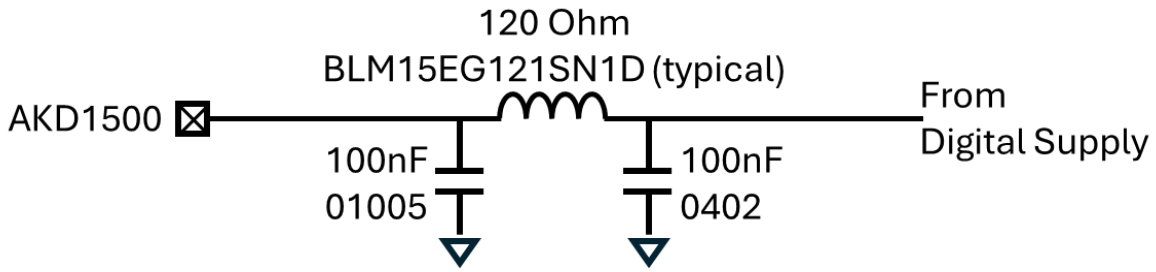



*Figure 18: Filtering analog supply lines*

A conservative implementation for bulk decoupling uses a set of 10uF 0603, 22uF 0603, 100uF 1202 caps on the 1.8V supply, and two sets on the 0.8V supply. The bulk decoupling caps should be within an inch of the AKD1500.

## 11.4 PCIe

- Differential traces to edge connector should be 5 to 7 mils with a typical spacing of one to two times the trace width.
- Target transmission line impedance between 68 and 105 ohms (85 ohms will match the PCIe edge connector).
- Coupling caps on the p and n legs of the TX lanes should be 100nF and 0201 form factor. The coupling caps should be side-by-side in line with the differential lane routes. See Figure 19: PCIe lane routing.
- Maintain a continuous reference plane under PCIe differential pairs.
- We recommend that the TX lane coupling caps have an anti-pad in their reference plane. See Figure 19: PCIe lane routing.
- Skew between traces in differential pairs should be 5mils or less.
- Skew between lanes should be less than 5000 mils.
- Vias in the differential pairs should be minimized and balanced. 2 vias per leg max.
- Avoid 90° angles on traces.
- Remove inner planes and routes on PCIe edge connector.

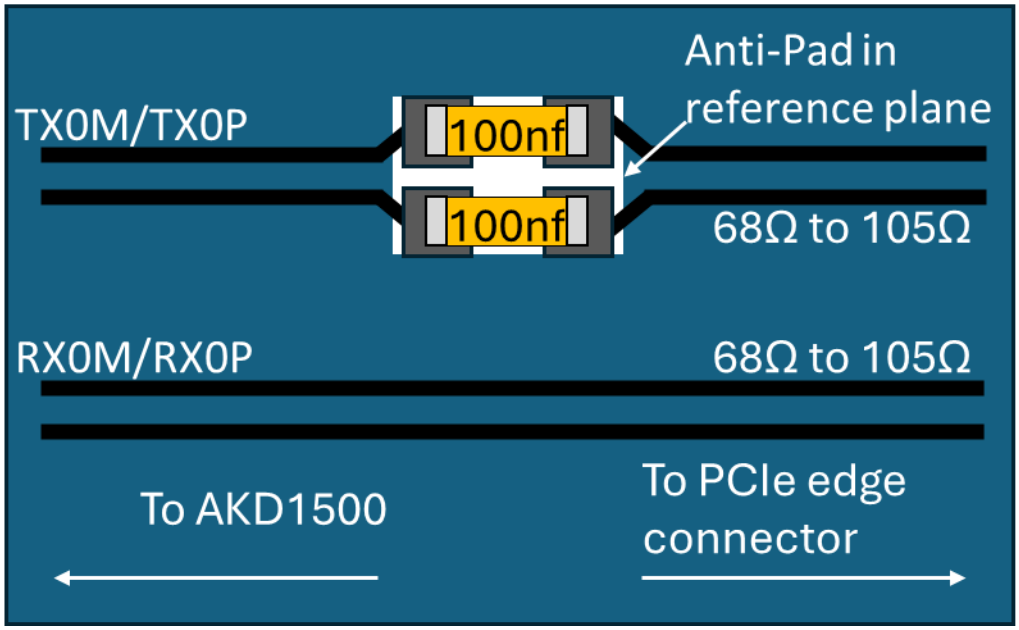



*Figure 19: PCIe lane routing*

## 11.5 SPI_S and SPI_M

- Avoid trace stubs on SPI clocks.
- Use source termination at the source for longer SPI clock lines.
- Daisy chain or use the auxiliary SPI clock (SPI_M_SCKX) when connecting multiple SPI loads
- Maintain a continuous reference plane under the SPI clocks

# 12 Ball Assignments

## 12.1 Ball Map

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | GND | VPH 33 | VP08 | VPTX108 | GND | REFPADCLK_P | TX0P | RX0P | RX1P | TX1P | GND | SPI_S_SCK | GND | A |
| B | GND | RESREF | GND | VPTX008 | GND | REFPADCLK_M | TX0M | RX0M | RX1M | TX1M | GND | SPI_S_MODE0 | SPI_S_MODE1 | B |
| C | GND | GND | GND | GND | GND | GND | GND | GND | GND | GND | SPI_S_IO0 | GND | SPI_S_IO1 | C |
| D | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDE18 | GND | SPI_S_IO2 | SPI_S_IO3 | SPI_S_IO4 | D |
| E | GND | GND | GND | GND | GND | GND | GND | GND | VDE18 | GND | SPI_S_IO5 | GND | SPI_S_IO6 | E |
| F | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDE18 | GND | SPI_S_IO7 | SPI_S_2MCS0N | SPI_S_CS_N | F |
| G | GND | GND | GND | GND | GND | GND | GND | GND | VDE18 | VDDAPLL18 | GND | SPI_S_2MCS1N | GND | G |
| H | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDD08 | VDE18 | VDDPLL08 | AVS_PLL | GND | OSC_XI | H |
| J | GND | GND | GND | GND | GND | GND | GND | VDE18 | VDE18 | VDE18 | GND | GND | OSC_XO | J |
| K | VPD | TESTMODE | VDE18 | VDE18 | VDE18 | VDE18 | VDE18 | VQPS_eFUSE | VDDIO_eFUSE | GND | SPI_M_SCKX | SPI_M_CS1_N | GND | K |
| L | TAP_SEL | PWR_GOOD | GND | GND | GND | GND | GND | GND | GND | SPI_M_IO7 | SPI_M_IO6 | SPI_M_IO1 | SPI_M_IO0 | L |
| M | GND | SLEEP | OP_MODE0 | GPIO_03 | GPIO_01 | FUSE_CK | FUSE_EN | GND | SPI_M_CS0_N | SPI_M_IO4 | SPI_M_IO2 | GND | SPI_M_SCK | M |
| N | GND | PCIE_PERST_N | OP_MODE1 | PCIE_HOST_SEL | GPIO_02 | GPIO_00 | SEL_CLK | SPI_M_RST_N | SPI_M_IO5 | GND | SPI_M_IO3 | SPI_M_DQS | GND | N |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | |

*Figure 20: Ball Map*

## 12.2 Ball Description

| Interface | Pin | Symbol | Description |
|---|---|---|---|
| Configuration | M3 | OP_MODE0 | Source of internal clock: '0'=Xtal. '1'=Reserved |
| Configuration | N3 | OP_MODE1 | 1'=SAFE Mode (PLL setup bypassed); '0'=Normal mode. |
| Configuration | N4 | PCIE_HOST_SEL | '1' = PCIe is Host; '0' = SPI Slave is host (SAFE Mode=0) |
| Configuration | L2 | PWR_GOOD | Power Good – AKD1500 master reset |
| Configuration | N7 | SEL_CLK | 1'=select digital CLK input for OSC_XI_CLK pad; '0'=select OSC (XI/XO) |
| Configuration | M2 | SLEEP | 0'=normal mode; '1'=low power |
| GPIO | N6 | GPIO_00 | General purpose IO |

| Interface | Pin | Symbol | Description |
|---|---|---|---|
| GPIO | M5 | GPIO_01 | General purpose IO/PLL lock |
| GPIO | N5 | GPIO_02 | General purpose IO/IRQ High priority |
| GPIO | M4 | GPIO_03 | General purpose IO/IRQ |
| OSC | H13 | OSC_XI | Xtal Oscillator feedback/Ext clock input |
| OSC | J13 | OSC_XO | Xtal Oscillator output |
| PCIe | N2 | PCIE_PERST_N | PCIe slot reset |
| PCIe | B6 | REFPADCLK_M | PCIe -ref clock |
| PCIe | A6 | REFPADCLK_P | PCIe +ref clock |
| PCIe | B2 | RESREF | PCIe Calibration Resistor |
| PCIe | B8 | RX0M | PCIe -Rx Lane 0 |
| PCIe | A8 | RX0P | PCIe +Rx Lane 0 |
| PCIe | B9 | RX1M | PCIe -Rx Lane 1 |
| PCIe | A9 | RX1P | PCIe +Rx Lane 1 |
| PCIe | B7 | TX0M | PCIe -Tx Lane 0 |
| PCIe | A7 | TX0P | PCIe +Tx Lane 0 |
| PCIe | B10 | TX1M | PCIe -Tx Lane 1 |
| PCIe | A10 | TX1P | PCIe +Tx Lane 1 |
| PCIe | A3 | VP08 | Low Voltage Supply. This signal can be supplied from the same off-chip source as the chip 0.8V VDD core supply, but must be isolated on the board through a ferrite bead and must not be connected to any other core supply in the package or on the die |
| PCIe | A2 | VPH33 | High Voltage Supply.<br>High voltage supply of the PHY. This signal is 3.3V. This power supply should be isolated on the board through a ferrite bead and must not be shared with any other supplies in the package<br>or on the die. |
| PCIe | B4 | VPTX008 | Transmitter Supply Voltage.<br>Similar to VP08, this supply can come from the same off-chip source as the chip 0.8V VDD supply, but must be isolated on the board through a ferrite bead and must not be connected to any other core supply in the package or on the die.<br>Sharing VPTXx08 and VP08 connections adds RX asynchronous noise on the transmitter power supply. Keep them isolated until out on the PCB<br>board |

| Interface | Pin | Symbol | Description |
|---|---|---|---|
| PCIe | A4 | VPTX108 | Transmitter Supply Voltage.<br>Similar to VP08, this supply can come from the same off-chip source as the chip 0.8V VDD supply, but must be isolated on the board through a ferrite bead and must not be connected to any other core supply in the package or on the die.<br>Sharing VPTXx08 and VP08 connections adds RX asynchronous noise on the transmitter power supply. Keep them isolated until out on the PCB<br>board |
| SPI Master | M9 | SPI_M_CS0_N | SPI Master Chip Sel 0 |
| SPI Master | K12 | SPI_M_CS1_N | SPI Master Chip Sel 1 |
| SPI Master | N12 | SPI_M_DQS | SPI data strobe in some modes |
| SPI Master | L13 | SPI_M_IO0 | SPI Master Data |
| SPI Master | L12 | SPI_M_IO1 | SPI Master Data |
| SPI Master | M11 | SPI_M_IO2 | SPI Master Data |
| SPI Master | N11 | SPI_M_IO3 | SPI Master Data |
| SPI Master | M10 | SPI_M_IO4 | SPI Master Data |
| SPI Master | N9 | SPI_M_IO5 | SPI Master Data |
| SPI Master | L11 | SPI_M_IO6 | SPI Master Data |
| SPI Master | L10 | SPI_M_IO7 | SPI Master Data |
| SPI Master | N8 | SPI_M_RST_N | SPI Master reset |
| SPI Master | M13 | SPI_M_SCK | SPI Master Clock (primary) |
| SPI Master | K11 | SPI_M_SCKX | SPI Master Clock (secondary) |
| SPI Slave | F12 | SPI_S_2MCS0_N | SPI Master I/F Chip Sel 0 |
| SPI Slave | G12 | SPI_S_2MCS1_N | SPI Master I/F Chip Sel 1 |
| SPI Slave | F13 | SPI_S_CS_N | SPI Slave Chip Sel |
| SPI Slave | C11 | SPI_S_IO0 | SPI Slave Data |
| SPI Slave | C13 | SPI_S_IO1 | SPI Slave Data |
| SPI Slave | D11 | SPI_S_IO2 | SPI Slave Data |
| SPI Slave | D12 | SPI_S_IO3 | SPI Slave Data |
| SPI Slave | D13 | SPI_S_IO4 | SPI Slave Data |
| SPI Slave | E11 | SPI_S_IO5 | SPI Slave Data |
| SPI Slave | E13 | SPI_S_IO6 | SPI Slave Data |
| SPI Slave | F11 | SPI_S_IO7 | SPI Slave Data |
| SPI Slave | B12 | SPI_S_MODE0 | SPI Frame format input signal. Indicates the mode in which the SPI transfer is happening. '00'=Standard SPI mode; '01'=SPI Dual Mode; '10'=SPI Quad mode; '11'=SPI Octal mode |
| SPI Slave | B13 | SPI_S_MODE1 | |
| SPI Slave | A12 | SPI_S_SCK | SPI Slave Clock |
| Test | M6 | FUSE_CK | Do not connect |

| Interface | Pin | Symbol | Description |
|---|---|---|---|
| Test | M7 | FUSE_EN | Do not connect |
| Test | L1 | TAP_SEL | '0'=normal operation |
| Test | K2 | TESTMODE | ‘0'=normal operation |
| Test | K9 | VDDIO_eFUSE | Do not connect |
| Test | K1 | VPD | Do not connect |
| Test | K8 | VQPS_eFUSE | Do not connect |
| Power | H11 | AVS_PLL | VSS for PLL |
| Power | A1 | GND | |
| Power | A11 | GND | |
| Power | A13 | GND | |
| Power | A5 | GND | |
| Power | B1 | GND | |
| Power | B11 | GND | |
| Power | B3 | GND | |
| Power | B5 | GND | |
| Power | C1 | GND | |
| Power | C10 | GND | |
| Power | C12 | GND | |
| Power | C2 | GND | |
| Power | C3 | GND | |
| Power | C4 | GND | |
| Power | C5 | GND | |
| Power | C6 | GND | |
| Power | C7 | GND | |
| Power | C8 | GND | |
| Power | C9 | GND | |
| Power | D10 | GND | |
| Power | E1 | GND | |
| Power | E10 | GND | |
| Power | E12 | GND | |
| Power | E2 | GND | |
| Power | E3 | GND | |
| Power | E4 | GND | |
| Power | E5 | GND | |
| Power | E6 | GND | |
| Power | E7 | GND | |
| Power | E8 | GND | |
| Power | F10 | GND | |
| Power | G1 | GND | |
| Power | G11 | GND | |

| Interface | Pin | Symbol | Description |
|---|---|---|---|
| Power | G13 | GND | |
| Power | G2 | GND | |
| Power | G3 | GND | |
| Power | G4 | GND | |
| Power | G5 | GND | |
| Power | G6 | GND | |
| Power | G7 | GND | |
| Power | G8 | GND | |
| Power | H12 | GND | |
| Power | J1 | GND | |
| Power | J11 | GND | |
| Power | J12 | GND | |
| Power | J2 | GND | |
| Power | J3 | GND | |
| Power | J4 | GND | |
| Power | J5 | GND | |
| Power | J6 | GND | |
| Power | J7 | GND | |
| Power | K10 | GND | |
| Power | K13 | GND | |
| Power | L3 | GND | |
| Power | L4 | GND | |
| Power | L5 | GND | |
| Power | L6 | GND | |
| Power | L7 | GND | |
| Power | L8 | GND | |
| Power | L9 | GND | |
| Power | M1 | GND | |
| Power | M12 | GND | |
| Power | M8 | GND | |
| Power | N1 | GND | |
| Power | N10 | GND | |
| Power | N13 | GND | |
| Power | D1 | VDD08 | 0.8V core voltage |
| Power | D2 | VDD08 | 0.8V core voltage |
| Power | D3 | VDD08 | 0.8V core voltage |
| Power | D4 | VDD08 | 0.8V core voltage |
| Power | D5 | VDD08 | 0.8V core voltage |
| Power | D6 | VDD08 | 0.8V core voltage |
| Power | D7 | VDD08 | 0.8V core voltage |

| Interface | Pin | Symbol | Description |
|---|---|---|---|
| Power | D8 | VDD08 | 0.8V core voltage |
| Power | F1 | VDD08 | 0.8V core voltage |
| Power | F2 | VDD08 | 0.8V core voltage |
| Power | F3 | VDD08 | 0.8V core voltage |
| Power | F4 | VDD08 | 0.8V core voltage |
| Power | F5 | VDD08 | 0.8V core voltage |
| Power | F6 | VDD08 | 0.8V core voltage |
| Power | F7 | VDD08 | 0.8V core voltage |
| Power | F8 | VDD08 | 0.8V core voltage |
| Power | H1 | VDD08 | 0.8V core voltage |
| Power | H2 | VDD08 | 0.8V core voltage |
| Power | H3 | VDD08 | 0.8V core voltage |
| Power | H4 | VDD08 | 0.8V core voltage |
| Power | H5 | VDD08 | 0.8V core voltage |
| Power | H6 | VDD08 | 0.8V core voltage |
| Power | H7 | VDD08 | 0.8V core voltage |
| Power | H8 | VDD08 | 0.8V core voltage |
| Power | G10 | VDDAPLL18 | 1.8V PLL voltage - isolate with ferrite bead |
| Power | H10 | VDDPLL08 | 0.8V PLL core voltage - isolate with ferrite bead |
| Power | D9 | VDE18 | 1.8V i/o voltage |
| Power | E9 | VDE18 | 1.8V i/o voltage |
| Power | F9 | VDE18 | 1.8V i/o voltage |
| Power | G9 | VDE18 | 1.8V i/o voltage |
| Power | H9 | VDE18 | 1.8V i/o voltage |
| Power | J10 | VDE18 | 1.8V i/o voltage |
| Power | J8 | VDE18 | 1.8V i/o voltage |
| Power | J9 | VDE18 | 1.8V i/o voltage |
| Power | K3 | VDE18 | 1.8V i/o voltage |
| Power | K4 | VDE18 | 1.8V i/o voltage |
| Power | K5 | VDE18 | 1.8V i/o voltage |
| Power | K6 | VDE18 | 1.8V i/o voltage |
| Power | K7 | VDE18 | 1.8V i/o voltage |

*Figure 21: Ball Descriptions*

# 13THERMAL

| Symbol | Value | Units | Description |
|---|---|---|---|
| $\theta_{jb}$ | 14.59 | °C/W | Thermal resistance junction to board |
| $\theta_{ja}$ | 28.79 | °C/W | Thermal resistance junction to ambient |

Conditions: Ta=85(℃), Power=1.2(W)

*Figure 22: Thermal Characteristics*

# 14 MECHANICAL DIMENSIONS FOR MFCLFBGA169 7X7MM

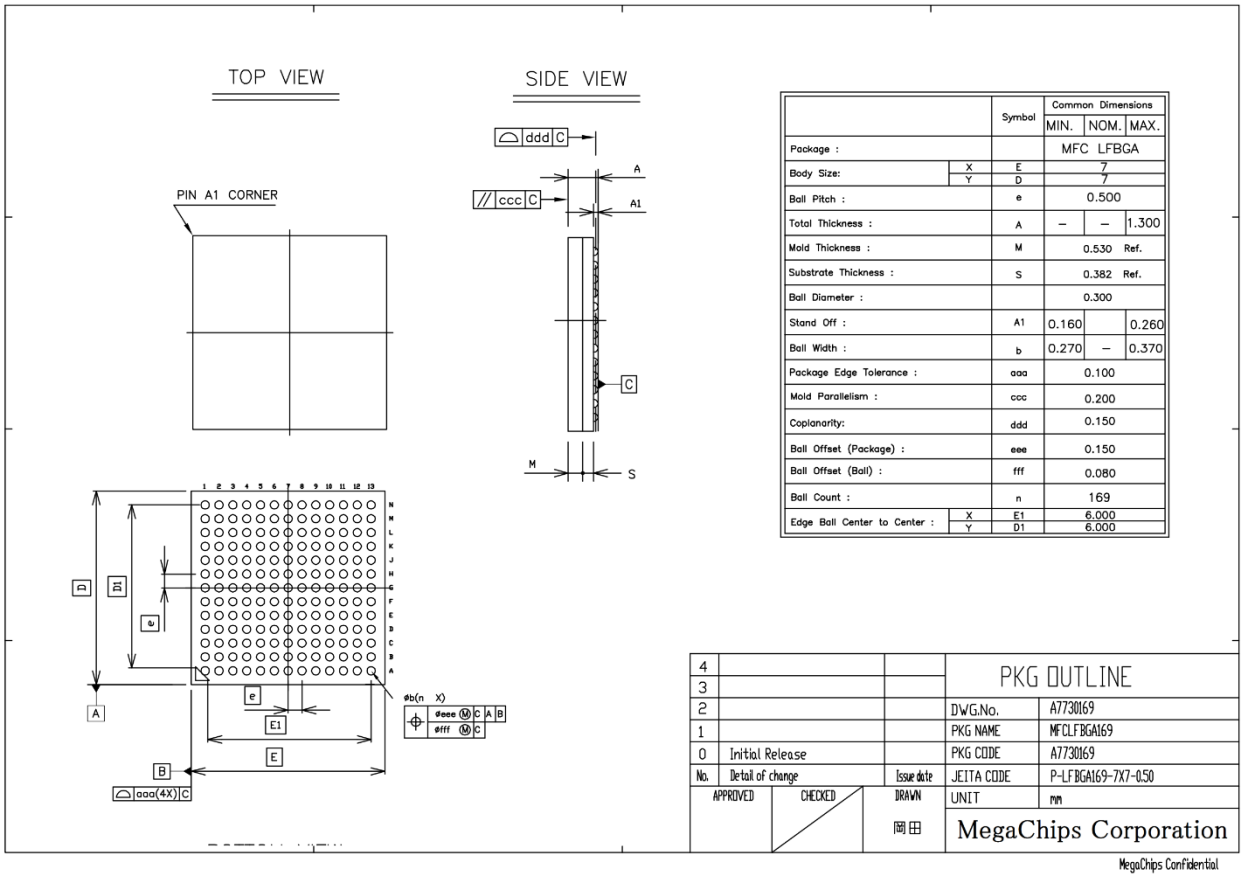



*Figure 23: Mechanical Package Dimensions (MFCLFBGA169)*

# 15 Shelf-Life and Ball Composition

## 15.1 Moisture Sensitivity Level

The MSL is 3. The shelf-life in a Moisture Barrier Bag is 12-months. After this time, it is recommended to bake the chips for 24 hrs. at (125±5) ℃ in heat–resistant containers.

## 15.2 Ball Composition

Ball Composition is Sn-1.2Ag-0.5Cu-0.05Ni

# 16 Application Examples

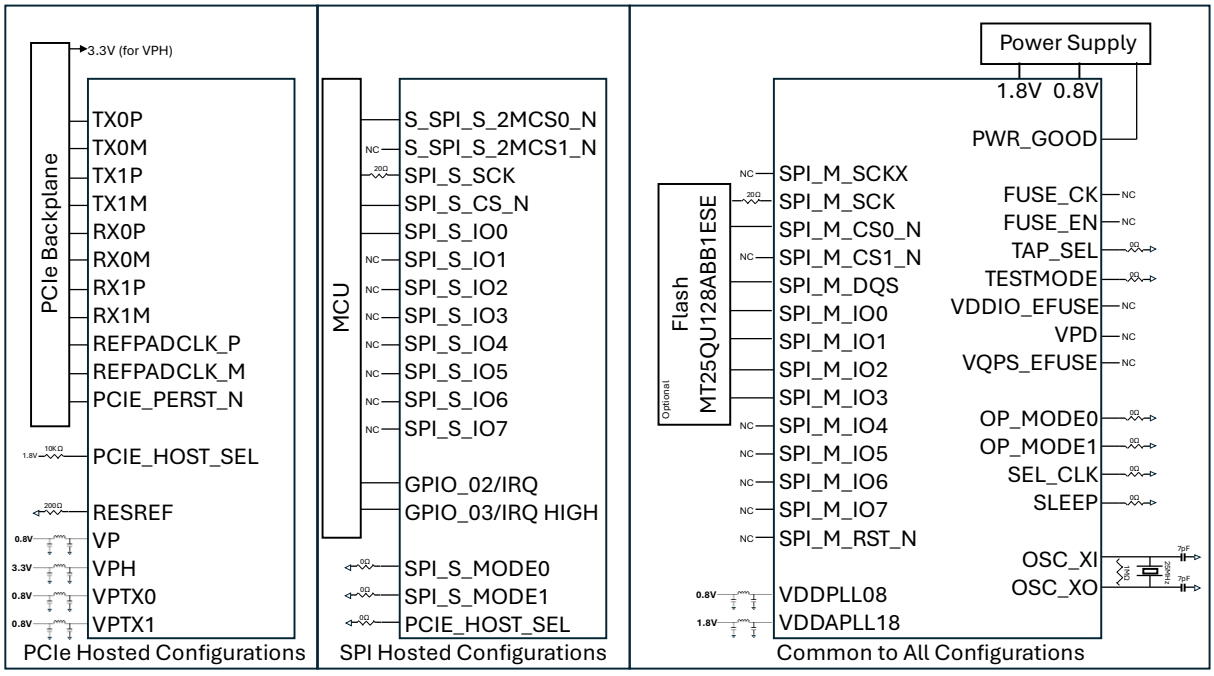



*Figure 24: Application Examples*