SPI Slave APB Master IP

Revision 0.4

January 6, 2016

Table of Contents

Table of Figures

Table of Tables

1 Revision History

Revision	Date	Author	Change Description
0.1	11/01/12	Dave Charneski	First draft.
0.2	11/01/12	Dave Charneski	Made several grammar/stye changes and improved readability of timing diagrams in printed copy.
0.3	12/8/15	Bill Elliott	Added parameter definition table. Updated to the latest architecture.
0.4	1/5/16	Bill Elliott	More timing details and expanded timing diagrams.

2 Referenced Documents

1. SPI Slave APB Master Requirements REV7
2. ARM AMBA Specification Rev 2.0

3 Introduction

The Intrinsix Corp SPI Slave APB Master module provides a configurable, streamlined, high performance SPI slave interface implemented in Verilog HDL which is compatible with many common SPI master protocols and an industry standard APB2 master interface as defined in Chapter 5 of ARM AMBA Specification Rev 2.0.

It is anticipated that this IP will be used most often as an interface to programming registers in applications where simplicity and a small footprint are desired. The SPI Slave APB Master module is also capable of operating in predominantly analog applications where there are no dedicated, free-running digital clocks elsewhere in the design.

Features

- **SPI Slave Interface**
  - Single-Channel
  - Single Data Rate
  - Independent Read, Write Transfers
- **All SPI Bus Modes Supported**
  - Mode 0, 1, 2 or 3 configurable at compile time via Verilog parameters
- **Command, Address, Data SPI Protocol**
  - Address and data field widths configurable at compile time via Verilog parameters
- **ARM AMBA APB2 Master Interface**
  - PCLK frequency division logic configurable at compile time via Verilog parameter and define statements to support slow APB slaves
- **Operates From SPI Serial Clock Only – No Other Clocks Required**
- **Minimal Area Design**
  - Command, address and data fields are available to the APB layer directly from the input serial shift register. No additional buffering or registering is inserted between layers.

4 Architecture Overview

The SPI Slave APB Master top-level block diagram is given below.

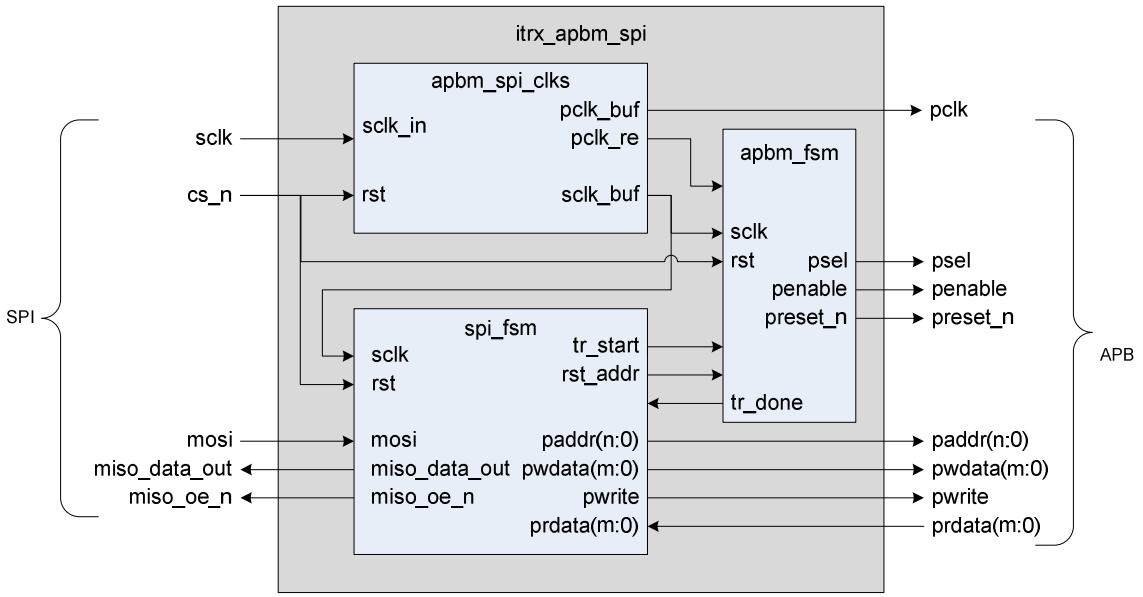

Figure 4-1 : SPI Slave APB Master Top-Level Block Diagram

The top-level architecture consists of the abpm_spi_clks block, where all internal clocks are created and buffered, and two finite state machine blocks, spi_fsm and abpm_fsm.

The abpm_spi_clks block receives SCLK and CS_N from the SPI bus and generates the buffered APB clock, PCLK, and a buffered version of SCLK. Note that CS_N serves as the IP's global asynchronous reset. Given that all internal clocks are derived from SCLK, the itrx_abpm_spi module and connected APB slave modules do not require any additional clocks for full and proper operation. As more fully described in the Timing and Functional Description section, PCLK frequency selection is parameterized and can be set to run at $f_{SCLK}/1$, $f_{SCLK}/2$, $f_{SCLK}/4$ or $f_{SCLK}/8$. The buffered SCLK and PCLK clock domains are fully synchronized so that data can be passed between them without timing violations. The pready and perror signals from the APB spec are not supported.

The itrx_apbm_spi block receives all of the SPI bus signals and generates the APB interface to the core logic. More details about itrx_apbm_spi are discussed in the Architecture Detail section. The itrx_apbm_spi signal interface description is given below.

Signal Name	Direction	Description
SCLK	In	SPI Serial Clock:
CS_N	In	SPI Chip Select (Active Low): When CS_N = '1', all flip-flips in the module will be held in the reset state. When CS_N = '0', all flip-flops come out of reset and are responsive to PCLK, and SCLK. Note that CS_N does not generate a PRESET_N signal.
MOSI	In	SPI Master Out Slave In: The MOSI input is the serial data input from the SPI master.
MISO_DATA_OUT	Out	SPI MISO (Master In Slave Out) Data Output: The MISO_DATA_OUT output is the serial data output and must be connected to the data input of a tri-state output pad. The output of the tri-state pad must be connected to the MISO pin.
MISO_OE_N	Out	SPI MISO Output Enable (Active Low): The MISO_OE_N output is the output enable control for MISO tri-state output pad. When MISO_OE_N = '0', the tri-state output pad must drive the MISO pin of the device with MISO_DATA_OUT. When MISO_OE_N = '1', the tri-state output pad must put the MISO pin into the high impedance state.
PCLK	Out	APB Clock: The PCLK output is the clock which serves the APB domain. Via module parameter settings in the RTL, PCLK frequency can be programmed at compile time to operate at $f_{SCLK}/1$, $f_{SCLK}/2$, $f_{SCLK}/4$ or $f_{SCLK}/8$.
PSEL	Out	APB Slave Select (Active High): The PSEL output selects the APB Slave (active high) and stays asserted throughout the two-cycle APB access.
PENABLE	Out	APB Enable (Active High): The PENABLE output is asserted during the second half of the APB access to indicate that either PWDATA (during writes) or PRDATA (during reads) is valid.
PADDR(n:0)	Out	APB Address: The PADDR output is the APB slave address which receives PWDATA during writes and from which PRDATA is retrieved during reads. The PADDR bus width is parameterized (i.e. fixed at compile time) as described in section the Timing and Functional Description.
PWDATA(m:0)	Out	APB Write Data: The PWDATA output is the data written to the APB slave and is available directly from the output of the MOSI shift register immediately after shifting into the device. The PWDATA bus width is parameterized (i.e. fixed at compile time) as described in theTiming and Functional Description section .
PRDATA(m:0)	In	APB Read Data: The PRDATA input is the data read from the APB slave and is available to the MISO output shift register upon completion of the APB enable phase. The PRDATA bus width is parameterized (i.e. fixed at compile time) as described

		inTiming and Functional Description section.
PWRITE	Out	APB Write/Read Control: When PWRITE = '1', an APB write operation is performed. When PWRITE = '0', an APB read operation is performed.
PRESET_N	Out	APB Reset: When PRESET_N = '0', the APB slave enters the reset state. When PRESET = '1', the APB slave leaves the reset state and becomes responsive to PCLK. PRESET_N is asserted when a write operation to the APB_RST_ADDR is performed. APB_RST_ADDR is determined at compile time by a module parameter.

Table 4-1 : Module Interface Signals

Param Name	Range	Default	Description
ADDR_BITS_N	1 to 11	3	Number of bits in the SPI address field. Size of the APB address bus.
DATA_BITS_M	Min = 1	8	Number of bits in the SPI data field. Size of the APB data bus
PCLK_DIV	1, 2, 4, 8	1	PCLK divide ratio. $f_{pclk} = f_{sclk}$ / PCLK_DIV
APB_RST_ADDR	0 to $2^{ADDR\_BITS\_N}$-1	0	Writing to this address will assert PRESET_N.
APB_RST_DATA	0 to $2^{DATA\_BITS\_N}$-1	0	Hard-coded value returned when reading from APB_RST_ADDR. Mode is active if APB_RST_RO is set.
APB_RST_RO	0, 1	0	Activates APB_RST_DATA as RO. Otherwise reads from that address are normal reads.
CPOL_MODE	0, 1	0	SPI CPOL mode.
CPHA_MODE	0, 1	1	SPI CPHA mode.

Table 4-2 : Verilog parameters used to configure the module

5 SPI Clock Modes

Traditionally, the relationship between the SPI clock and serial data has been defined by two one-bit values – the clock polarity, CPOL, and the clock phase, CPHA. This timing is illustrated below.

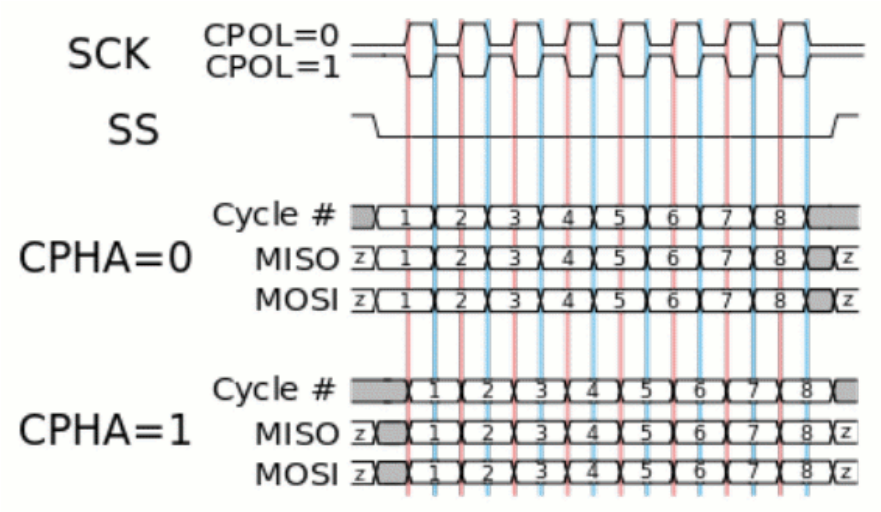

Figure 5-1 : SPI Clock Mode Timing

When CPHA=0, data is first sampled on the first SCLK edge (whether rising or falling), but it must be stable ½ clock cycle in advance. When CPHA=1, first data is presented on the first SCLK edge (whether rising or falling) and it is sampled on the second SCLK edge (whether rising or falling). The value of the CPHA and CPOL bits will be stored as parameters in a Verilog header file.

In order to simplify the internal clock architecture, the SPI serial clock, SCLK, will be inverted as required such that data is always sampled (or appears to be sampled, in the case of reads) on the rising edge of SCLK. Therefore, when CPHA = '0' and CPOL = '1' (sample data on falling edge), SCLK is inverted internally so that the module is effectively operating as if CPOL = '0' (sample data on rising edge). In like manner, when CPHA = '1' and CPOL = '0' (sample data on falling edge), SCLK is inverted internally so that the module is effectively operating as if CPOL = '1' (sample data on rising edge). So from a synthesis standpoint, the itrx_apbm_spi logic is posedge on the gated, buffered and possibly inverted SCLK, with these exceptions:

- Flip flip that samples MOSI
- PCLK gating logic

The SPI clock mode definitions are given below for reference.

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Figure 5-2 : SPI Clock Mode Definition

6 APB Interface Signals

For reference purposes, the APB2 write and read timing diagrams have been excerpted from ARM AMBA Specification Rev 2.0 and reprinted below.

Since the traditional 4-wire SPI protocol does not support an indeterminate number of wait states, as does APB3 and APB4. This limitation dictates that the APB layer must adhere to the APB2 specification in order to be fully compliant with any established APB standard. Therefore, as defined in Chapter 5 of AMBA Specification, Rev 2.0, read and write operations each consist of two PCLK periods and the APB pready signal from the slave is not supported. However, in cases where PCLK will not be operated at the SCLK frequency, the number of SCLK periods required for the APB Setup and Enable fields will increase with each increase in the PCLK_DIV.

The perror APB signal is not supported.

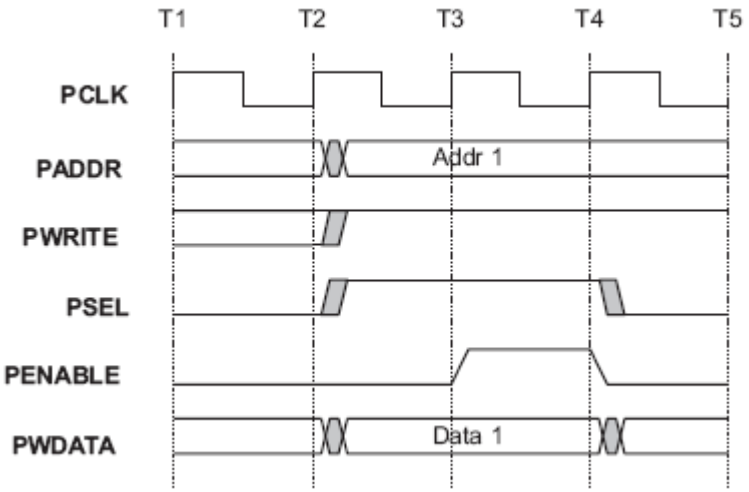

Figure 6-1 : APB2 Write Timing

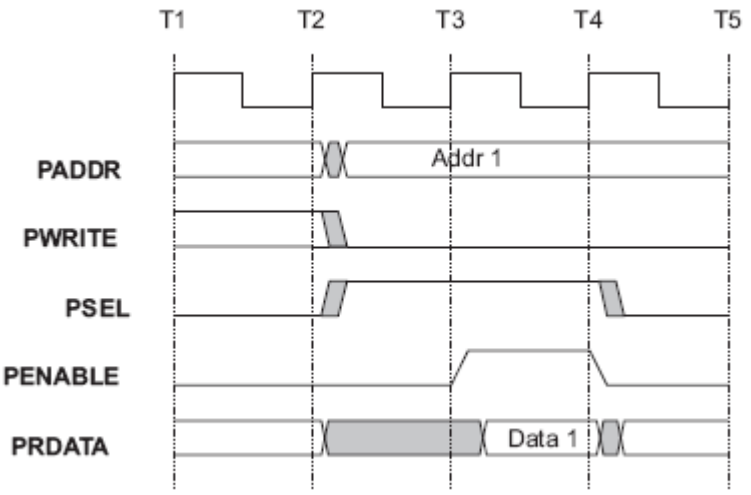

Figure 6-2 : APB2 Read Timing

7 SPI-APB Timing and Module Functional Description

Figures Figure 7-1 through Figure 7-6 show read and write timing for both SPI CPHA modes, and for PCLK_DIV equal to 1 and 8.

PCLK_DIV=1 mode, detailed in Figures Figure 7-1 through Figure 7-4, is distinct from the other PCLK_DIV modes in that the PCLK is a buffered version of SCLK rather than generated from a divide counter. Although, PCLK is always posedge and never negedge, the idle state of PCLK will be high as shown in the figures, when PCLK_DIV equals 1 and CPHA equals 0. In all other modes, PCLK idles low.

Figures Figure 7-5 and Figure 7-6 with PCLK_DIV equal 8 are representative of all the divided clock modes (PCLK_DIV≠1). These figures show SPI CPOL=0, but SPI CPOL=1 modes have identical timing, simply with SCLK inverted. PCLK is not inverted.

Examining the details of the MOSI signal timing, the write transfer consists of a R/W bit following by an address field, a data field, APB Setup and Enable period plus optional pad bits at the end of the transfer. Only one set of R/W, Address, Data and APB fields will be transmitted for each assertion of CS_N. Each assertion of CS_N comprises one transfer and the falling edge of CS_N indicates the start of a transfer. When CS_N = '1', the module enters the reset state.

The number of active bits within a transfer is given by the following equation. An active bit is defined as a bit with significance that is captured for use by the core logic.

ACTIVE_BITS = COMMAND_LEN + ADDR_LEN + DATA_LEN + APB_LEN,

where: COMMAND_LEN = 1 (for the R/W bit),
ADDR_LEN = n = address field length,
DATA_LEN = m = data field length,
APB_LEN = (2 * PCLK_DIV)

The ADDR_LEN, DATA_LEN, and PCLK_DIV values are captured as Verilog parameters which are set at compile time and, therefore, also fixed within the synthesized design. The APB clock, PCLK, is derived from SCLK and its frequency can be reduced up to a factor of 8 to accomodate slower APB slaves via selection of the appropriate PCLK_DIV. The PCLK_DIV value will be determined by the use of module parameters which will choose the appropriate net assignment to PCLK for the value of PCLK_DIV chosen. PCLK_DIV values of 1, 2, 4 and 8 will be provided.

In order to minimize area, the R/W, address and data fields are captured by an input shift register and transmitted directly to the APB output ports, PWRITE, PADDR and PWDATA, respectively, without any additional registering or buffering.

Upon capturing the last bit of the data field, D0, the internal control signal, tr_start is asserted to initiate the APB transfer. The APB transfer length is fixed at two PCLK cycles. However, in cases where PCLK will not be operated at the SCLK frequency, the number of SCLK periods required for the APB Setup and Enable fields will increase with each increase in the PCLK_DIV.

After the APB transaction is completed, the host system is allowed to insert dummy SCLK cycles to pad the transfer as desired. The transfer is then terminated by the deassertion of CS_N which drives all output signals to their reset state.

Read timing is similar to write timing except that the APB transaction is inserted between the address and data field transmissions and the data for the transfer appears on MISO_DATA_OUT versus MOSI. Accordingly, the internal signal, APB_START, is asserted upon capture of the last address field bit, A0, to initiate the APB transaction. The APB read data, PRDATA, is loaded into an output shift register upon assertion of PENABLE at the end of the APB read transaction and the data is then shifted out via MISO_DATA_OUT while MISO_OE_N is asserted to remove the bus from tri-state.

PAD_BITS, which are all SCLK edges occurring after completion of the ACTIVE_BITS transfer and until the deassertion and reassertion of CS_N, will be ignored other than continuing to the PCLK signal. This allows SPI masters to pad transfers as appropriate for the host controller architecture, for example, to achieve a convenient word length such as a convenient power of two or byte multiple. Read and write transfers occur independently, thus allowing any combination of consecutive and/or interspersed read and write transfers. Thus, the total number of bits per transfer is defined as follows.

Another consideration in generating pad bits is avoidance of creating a PCLK runt pulse if the SCLK stops while PCLK is in the positive phase. If the APB slave is sensitive to shortened pulses, the SCLK pad bits should be extended to create a full PCLK positive phase pulse.

TOTAL_BITS = ACTIVE_BITS + PAD_BITS = SPI transfer length

In the modes where PCLK_DIV≠1, the clock divider generates PCLK with a phase such that the PCLK positive edge coincides with the earliest possible start of the APB transfer. For a write transfer, this is the active SCLK edge when the last input data bit has been captured. For a read transfer, this is the active SCLK edge when the last address bit has been captured.

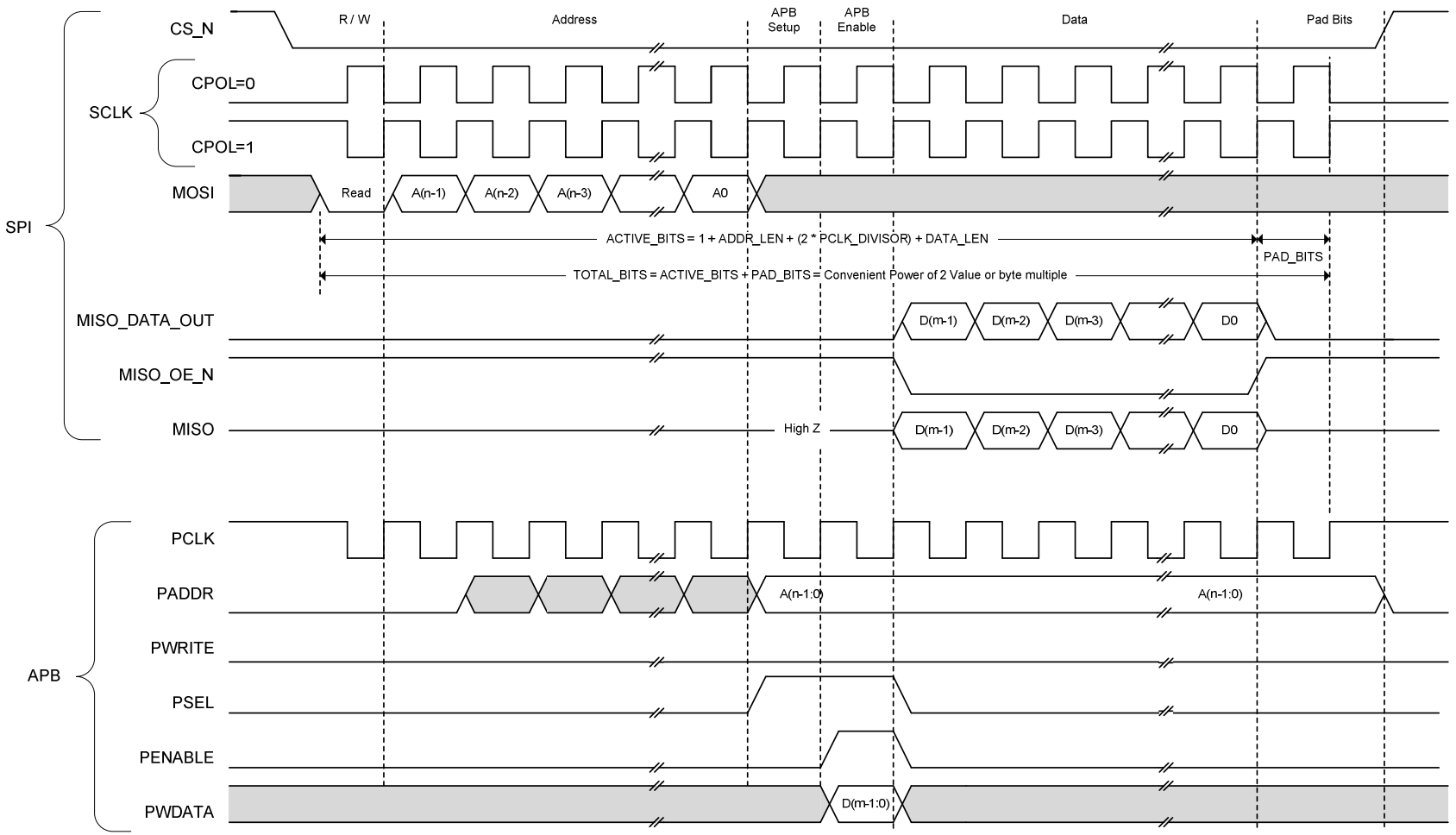

Figure 7-1 : Read Transfer Timing, PCLK_DIV=1, CPHA=0

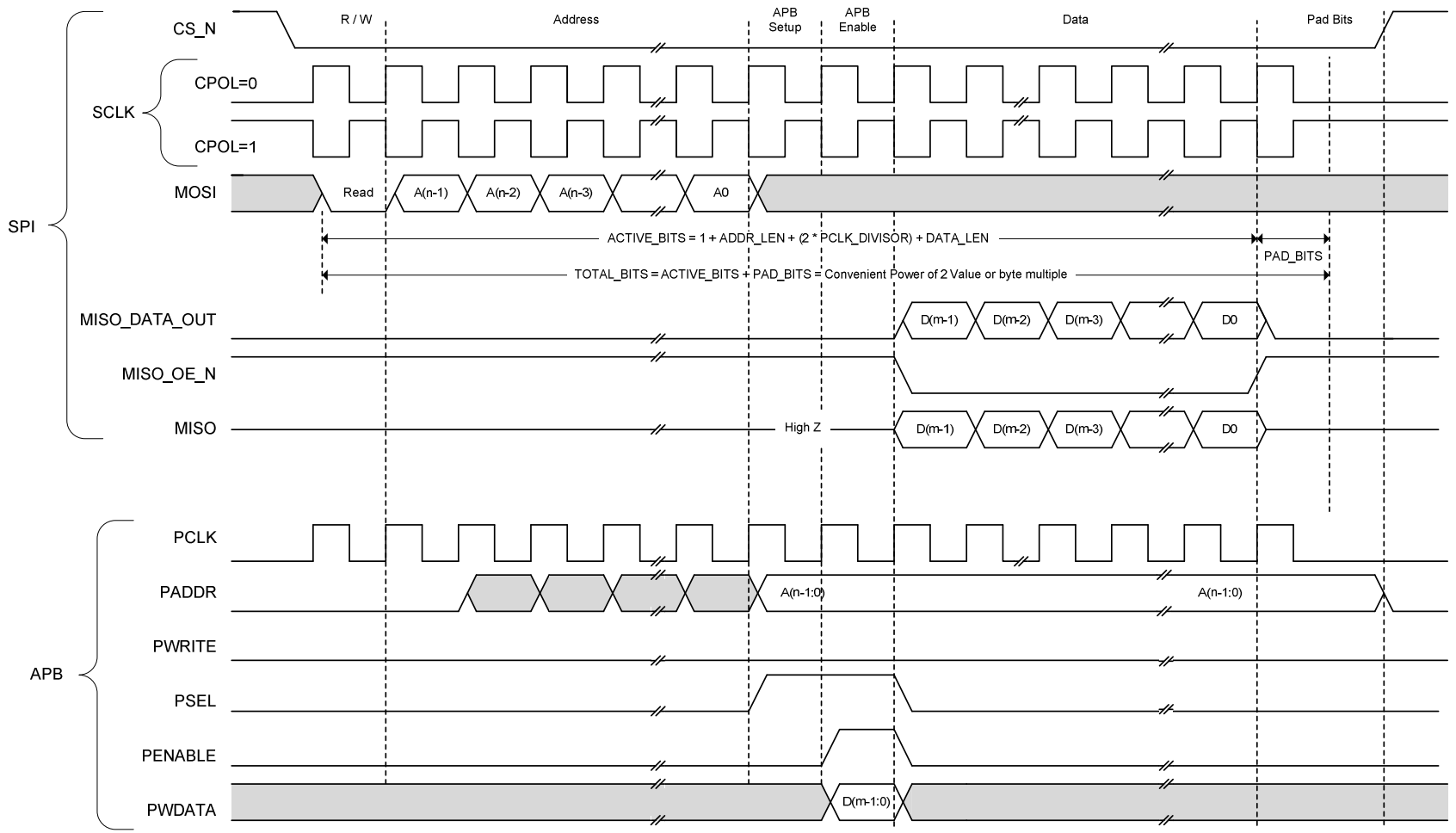

Figure 7-2 : Read Transfer Timing, PCLK_DIV=1, CPHA=1

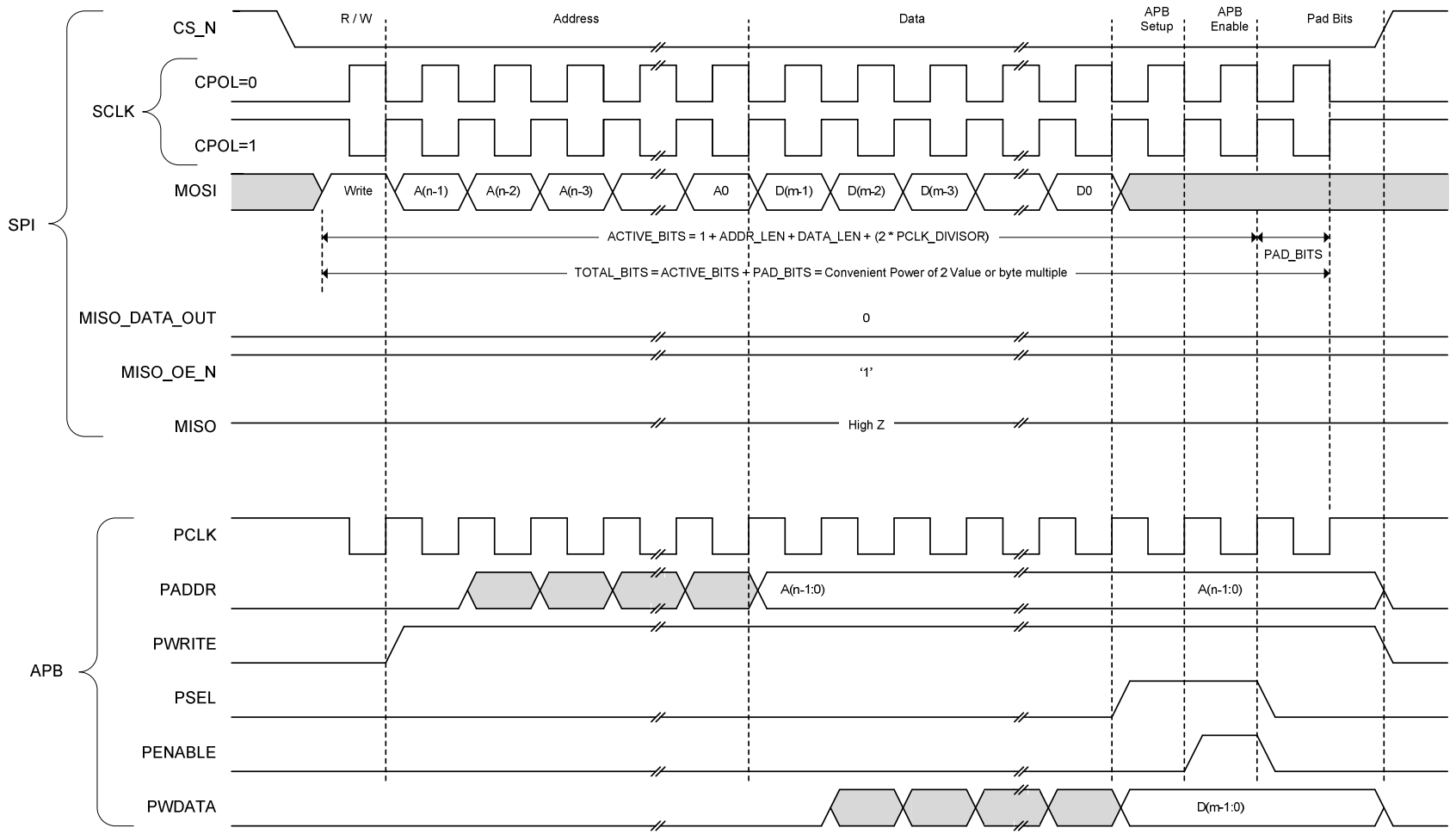

Figure 7-3 : Write transfer timing, PCLK_DIV=1, CPHA=0

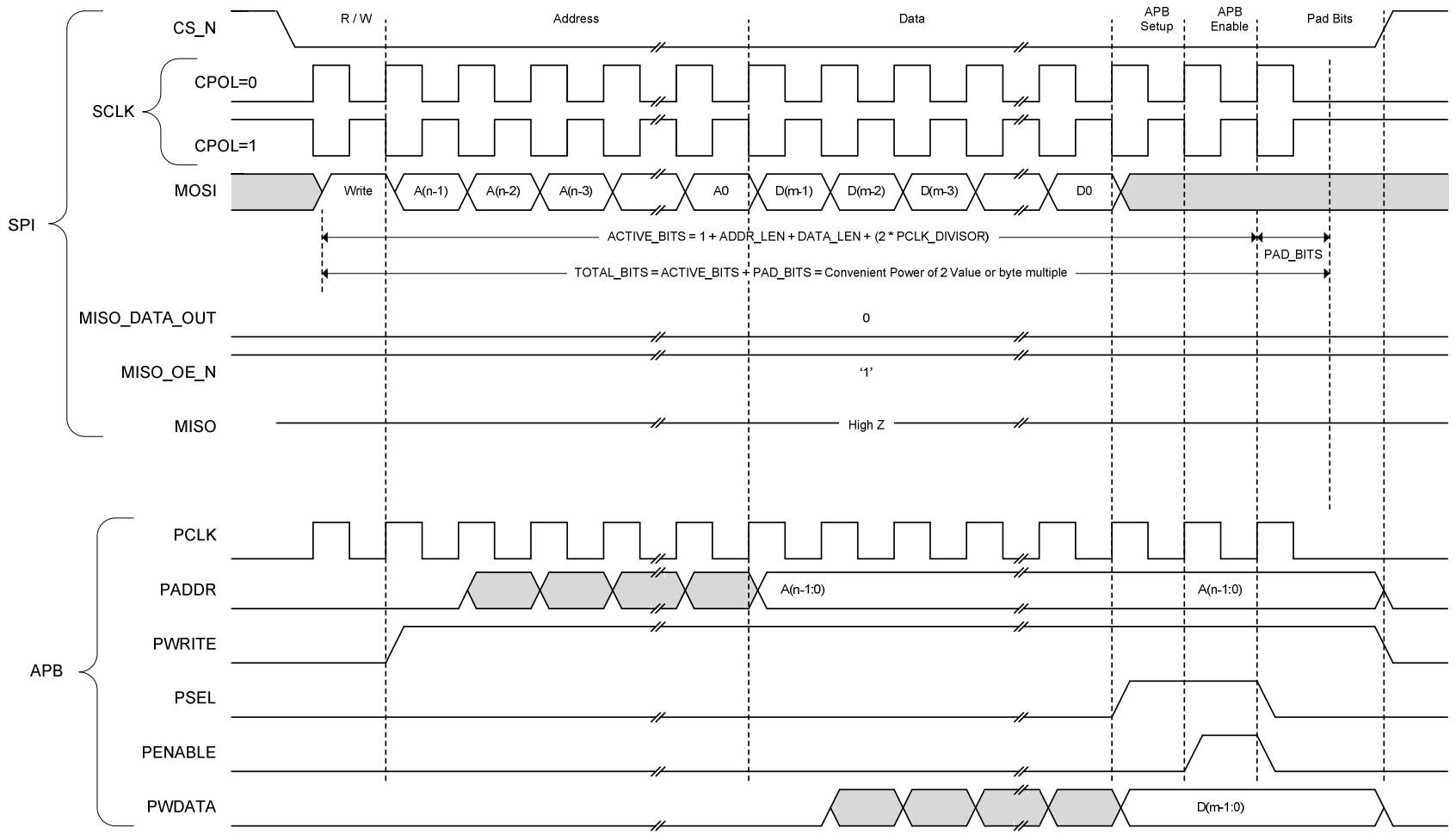

Figure 7-4 : Write transfer timing, PCLK_DIV=1, CPHA=1

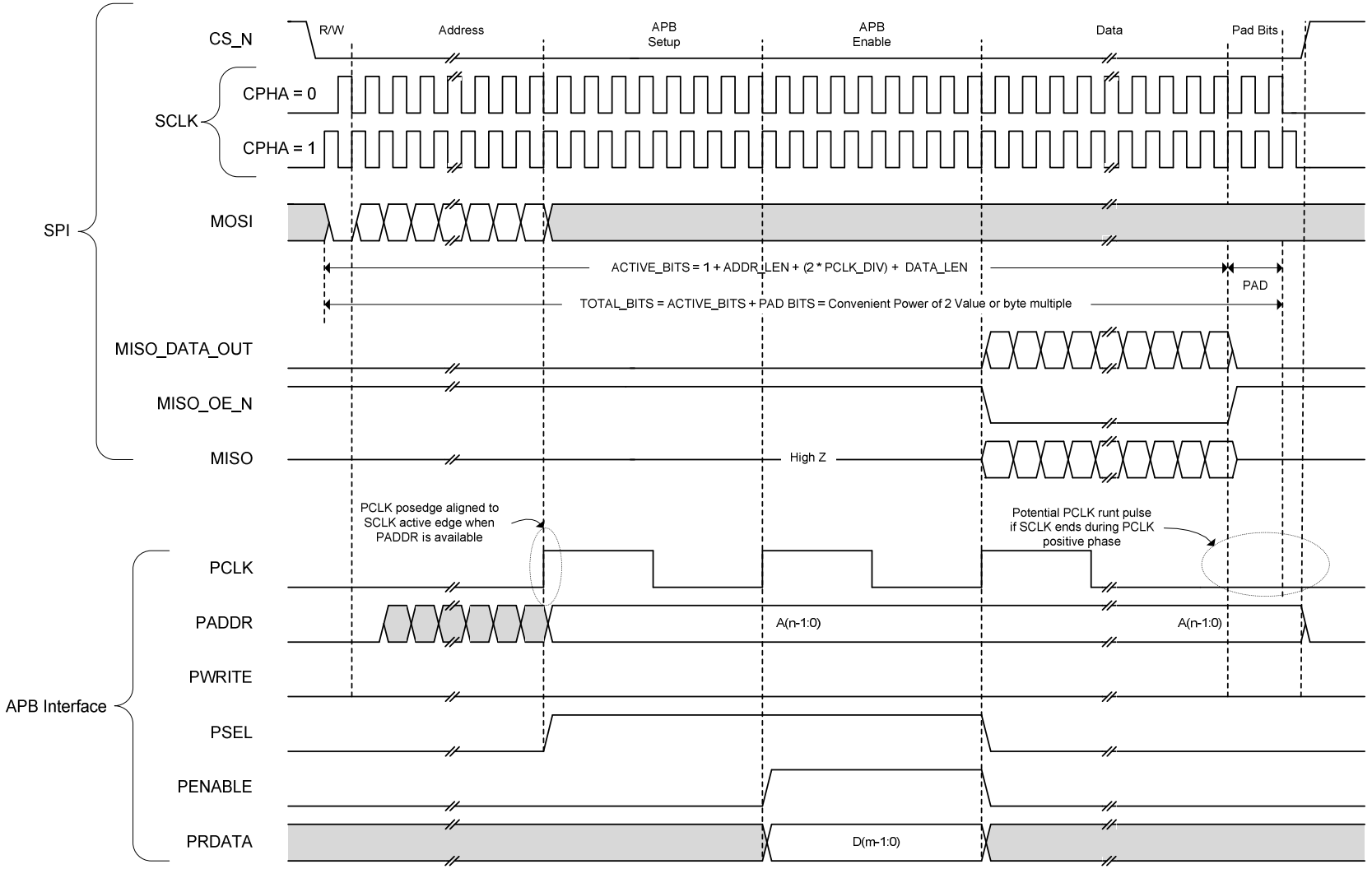

Figure 7-5 : Read Transfer Timing, PCLK_DIV=8

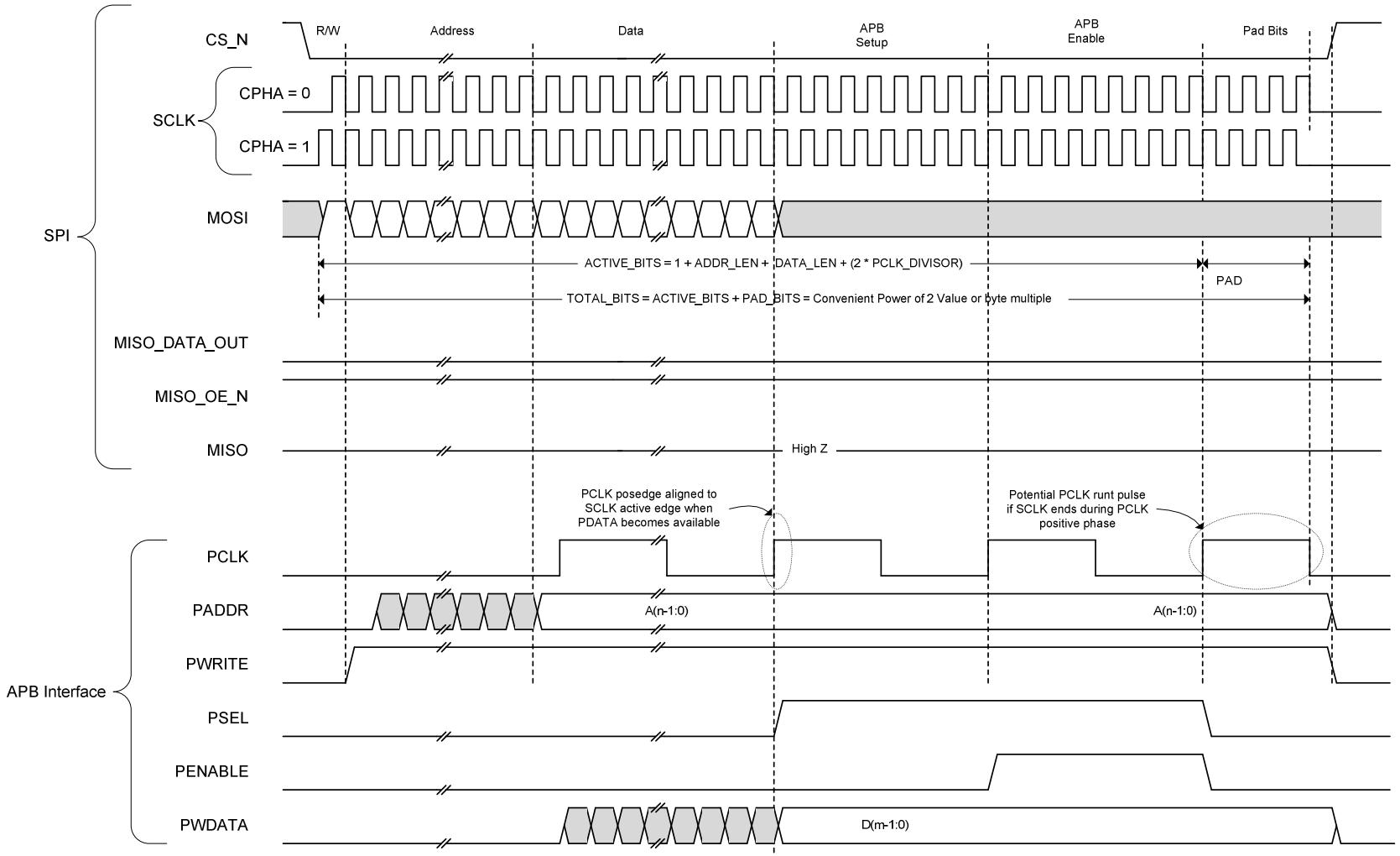

Figure 7-6 : Write Transfer Timing, PCLK_DIV=8

8 Architecture Detail

As discussed in the Architecture Overview section and show in Figure 4-1, the top level of the module consists of the following blocks:

itrx_apbm_clks block : where all internal clocks are created and buffered
itrx_apbm_spi_fsm block: contains the SPI FSM and serializes/deserializes words
itrx_apbm_fsm block: contains the APBM FSM

Given the detailed description of SPI and APB timing in the previous section as a prerequisite, the details of the finite state machine behavior of the itrx_apbm_spi module is given below.

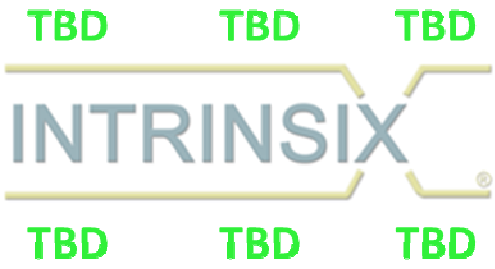

Figure 8-1 : SPI Finite State Machine

Figure 8-2 : APBM Finite State Machine

The SPI FSM interacts with the SPI interface, parallelizing the address bits and parallelizing data bits on a write operation and serializing the data bits on a read operation. When the address and data are ready for a write operation, and when the address is ready for a read operation, the APBM FSM is signaled to start the transfer with a tr_start signal. To cause the APB transfer, the APBM FSM generates the psel and penable signals with the correct timing. The tr_done signal from the APBM indicates the transfer is done. During a read operation, the tr_done signal starts the serialization of the data in the SPI FSM. During a write operation, tr_done indicates that both FSMs are finished with the operation.

The rst_addr signal from spi_fsm to apbm_fsm, which asserts when the addr matches the APB_RST_ADDR parameter, signals the APBM FSM to generate a synchronous preset_n signal to the APB bus.

9 Test Considerations

In applications requiring a relatively small address space, reduced area and little to no additional digital logic in the overall design, it is anticipated that scan testing of the SPI Slave APB Master IP will not be necessary as it may be possible to achieve very high fault coverage via functional vectors that exercise all data bits and the entire address space

In larger designs where scan is required, additional DFT logic will be necessary. Specifically, the addition of a SCAN_MODE pin and associated muxing logic to route SCLK around the PCLK divsion and SCLK inversion logic to all flip-flops will be required. In scan mode, SCLK will then serve as the scan clock. Given that CS_N serves as the global asynchronous reset in normal mode and is not gated in any way, it can serve as the scan reset pin. The scan chain through the IP can then be connected to the rest of the host design via the usual methods.

*Due to this additional complexity, addition of scan test compatibility as a requirement for this IP is **TBD**.*