introspect technology

Parallel Test Interface Design Guide for SV3D09

Pinout & timing, and programming examples

November 21, 2023

EN-G077E-E-23325

Pinout

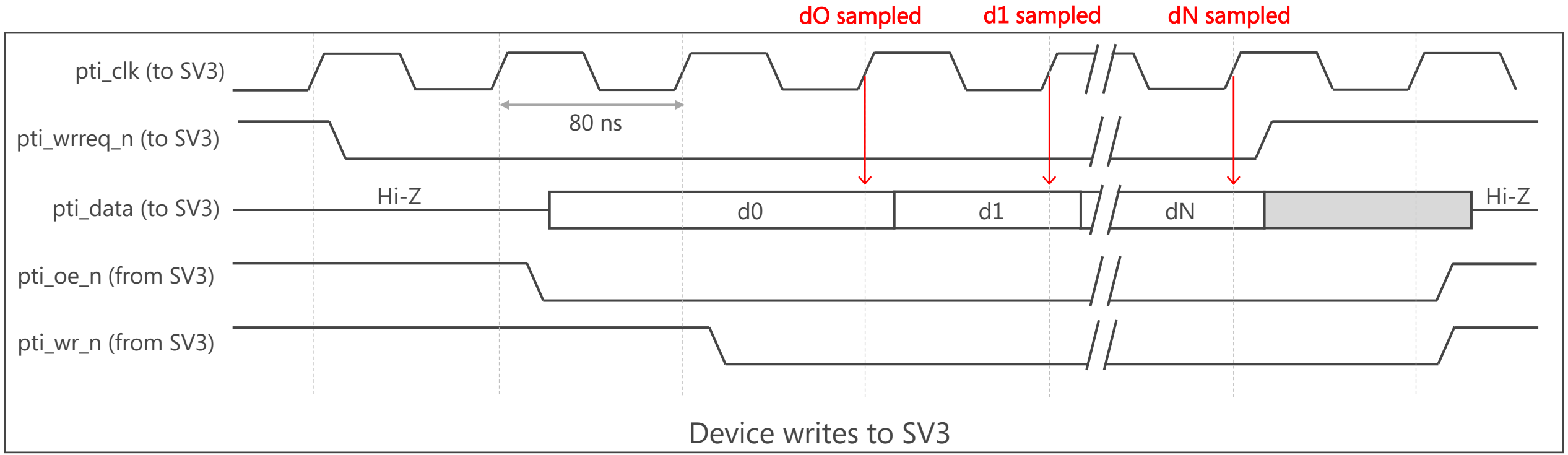
PTI pinout on SV3D J5

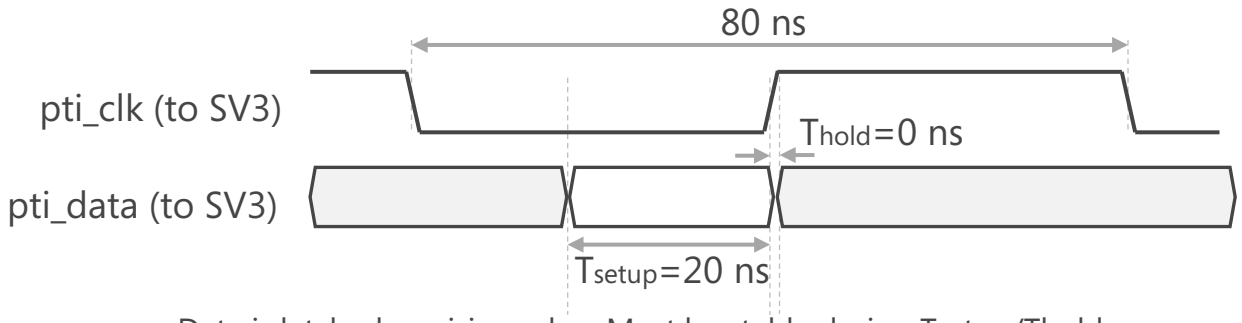
SV3D Header	Pin name	Pin #	Function	Direction
J5	GPIO_5_0	13	pti_data[0]	IO
J5	GPIO_5_7	14	pti_data[1]	IO
J5	GPIO_5_8	15	pti_data[2]	IO
J5	GPIO_5_15	16	pti_data[3]	IO
J5	GPIO_5_16	17	pti_data[4]	IO
J5	GPIO_5_23	18	pti_data[5]	IO
J5	GPIO_5_1	19	pti_data[6]	IO
J5	GPIO_5_6	20	pti_data[7]	IO
J5	GPIO_5_9	21	pti_data[8]	IO
J5	GPIO_DQS_5_2	25	pti_data[9]	IO
J5	GPIO_5_5	26	pti_data[10]	IO
J5	GPIO_5_10	27	pti_data[11]	IO
J5	GPIO_DQS_5_13	28	pti_data[12]	IO
J5	GPIO_5_22	30	pti_data[13]	IO
J5	GPIO_5_3	31	pti_data[14]	IO
J5	GPIO_5_4	32	pti_data[15]	IO
J5	GPIO_5_11	33	pti_data[16]	IO
J5	GPIO_5_18	35	pti_data[17]	IO
J5	GPIO_5_21	36	pti_data[18]	IO
J5	GPIO_4_0	37	pti_data[19]	IO
J5	GPIO_4_7	38	pti_data[20]	IO
J5	GPIO_5_12	40	pti_data[21]	IO
J5	GPIO_5_19	41	pti_data[22]	IO
J5	GPIO_5_20	42	pti_data[23]	IO
J5	GPIO_4_1	43	pti_data[24]	IO
J5	GPIO_4_8	45	pti_data[25]	IO
J5	GPIO_4_15	46	pti_data[26]	IO
J5	GPIO_4_16	47	pti_data[27]	IO
J5	GPIO_4_23	48	pti_data[28]	IO
J5	GPIO_4_6	50	pti_data[29]	IO
J5	GPIO_4_9	51	pti_data[30]	IO
J5	GPIO_4_14	52	pti_data[31]	IO
J5	GPIO_4_17	53	pti_clk	I
J5	GPIO_DQS_4_22	54	pti_wrreq_n	I
J5	GPIO_DQS_4_2	55	pti_rdreq_n	I
J5	GPIO_4_5	56	pti_wr_n	O
J5	GPIO_4_10	57	pti_rd_n	O
J5	GPIO_4_21	60	pti_oe_n	O
J5	GPIO_4_3	61	pti_reset_n	I

PTI pinout on SV3M J65 (side connector)

SV3M Header	Pin name	Pin #	Function	Direction
J65	GPIO_5_0	93	pti_data[0]	IO
J65	GPIO_5_7	104	pti_data[1]	IO
J65	GPIO_5_8	116	pti_data[2]	IO
J65	GPIO_5_15	140	pti_data[3]	IO
J65	GPIO_5_16	60	pti_data[4]	IO
J65	GPIO_5_23	58	pti_data[5]	IO
J65	GPIO_5_1	81	pti_data[6]	IO
J65	GPIO_5_6	92	pti_data[7]	IO
J65	GPIO_5_9	105	pti_data[8]	IO
J65	GPIO_DQS_5_2	80	pti_data[9]	IO
J65	GPIO_5_5	69	pti_data[10]	IO
J65	GPIO_5_10	68	pti_data[11]	IO
J65	GPIO_DQS_5_13	129	pti_data[12]	IO
J65	GPIO_5_22	59	pti_data[13]	IO
J65	GPIO_5_3	57	pti_data[14]	IO
J65	GPIO_5_4	56	pti_data[15]	IO
J65	GPIO_5_11	45	pti_data[16]	IO
J65	GPIO_5_18	71	pti_data[17]	IO
J65	GPIO_5_21	72	pti_data[18]	IO
J65	GPIO_4_0	47	pti_data[19]	IO
J65	GPIO_4_7	48	pti_data[20]	IO
J65	GPIO_5_12	128	pti_data[21]	IO
J65	GPIO_5_19	82	pti_data[22]	IO
J65	GPIO_5_20	70	pti_data[23]	IO
J65	GPIO_4_1	46	pti_data[24]	IO
J65	GPIO_4_8	44	pti_data[25]	IO
J65	GPIO_4_15	117	pti_data[26]	IO
J65	GPIO_4_16	83	pti_data[27]	IO
J65	GPIO_4_23	84	pti_data[28]	IO
J65	GPIO_4_6	35	pti_data[29]	IO
J65	GPIO_4_9	33	pti_data[30]	IO
J65	GPIO_4_14	141	pti_data[31]	IO
J65	GPIO_4_17	96	pti_clk	I
J65	GPIO_DQS_4_22	94	pti_wrreq_n	I
J65	GPIO_DQS_4_2	36	pti_rdreq_n	I
J65	GPIO_4_5	34	pti_wr_n	O
J65	GPIO_4_10	32	pti_rd_n	O
J65	GPIO_4_21	95	pti_oe_n	O
J65	GPIO_4_3	23	pti_reset_n	I

PTI Write Timing

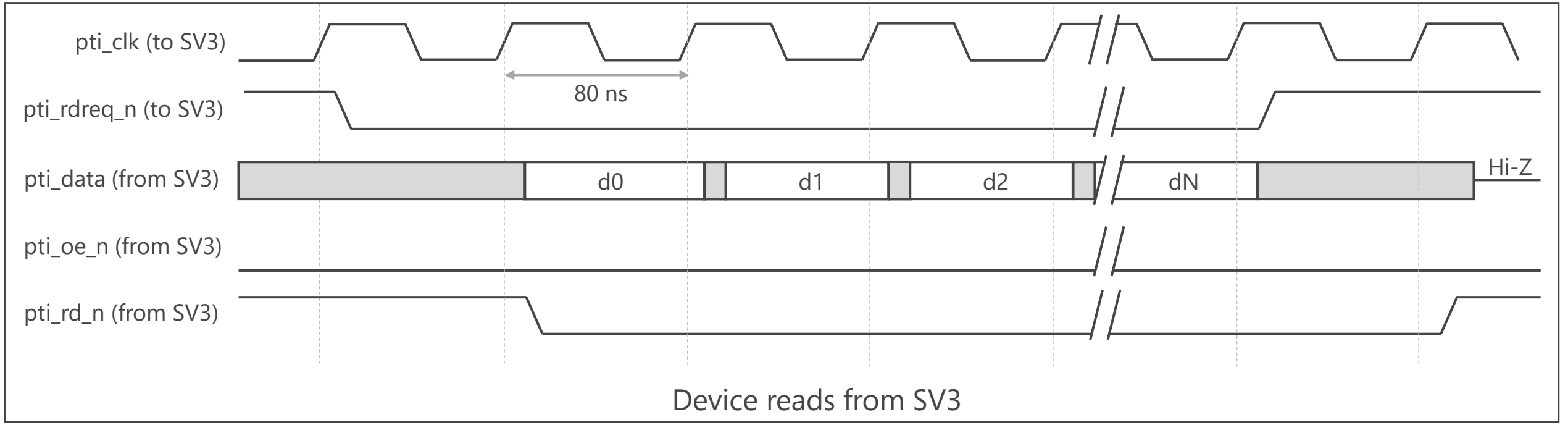




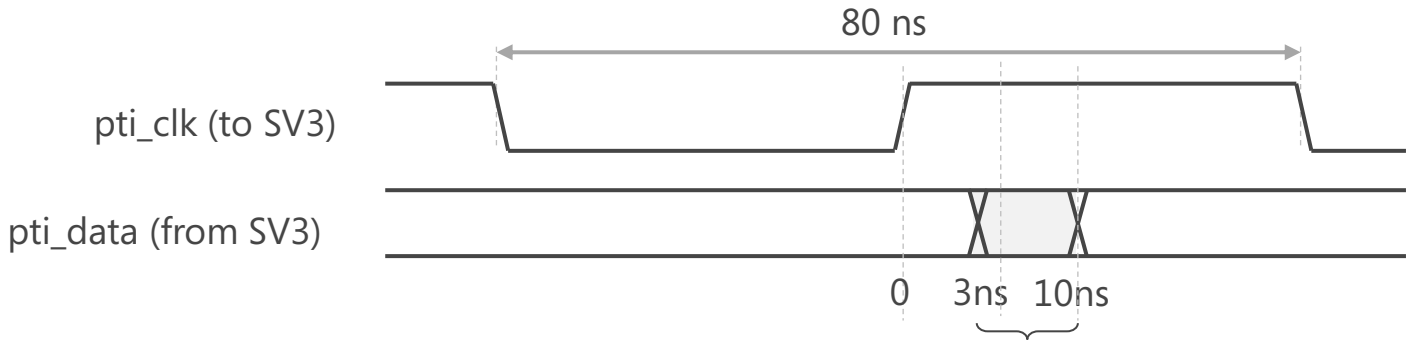
Data is latched on rising edge. Must be stable during Tsetup/Thold.

introspect technology

PTI Read Timing



Device reads from SV3



Data from the SV3 transitions in this window.

introspect technology

PTI Pin Descriptions

Name	I/O	Description
pti_data[31:0]	IO	32-bit bidirectional data bus
pti_clk	I	12.5 MHz free-running clock. All signals are synchronous to this clock.
pti_reset_n	I	Reset. Active low. Take PTI out of reset after power-up by setting pti_reset_n=1.
pti_wrreq_n	I	Request to write. Active low. The device sets pti_wrreq_n=0 to request to write to the SV3.
pti_rdreq_n	I	Request to read. Active low. The device sets pti_rdreq_n=0 to request to read from the SV3.
pti_rd_n	O	Read. Active low. The SV3 sets pti_rd_n=0 in response to a read request to let the device know when to read the pti_data. The device must register the pti_data on the rising edge of pti_clk when pti_rdreq_n=0 and pti_rd_n=0.
pti_wr_n	O	Write. Active low. The SV3 sets pit_wr_n=0 in response to a write request to let the device know that the pti_data has been written to the SV3. The device must present the next pti_data on the rising edge of pti_clk when pti_wrreq_n=0 and pti_wr_n=0.
pti_oe_n	O	Output Enable. Active low. The SV3 sets pti_oe_n=0 in response to a write request. The device must drive pti_data when pti_oe_n=0. The device must tristate pti_data when pti_oe_n=1.

I = Input to SV3
O = Output from SV3

Example 1: writing 1024 bits to SV3 via PTI, SPI commands

```
# PTI write chunk #1 (1024 bits / 128 bytes)
#-----------------------------------------------------------

# SPI command: Set PTI transfer length to 128 bytes (0x80 bytes)
send_spi_bytes(01 01 C4 00 00 00 00 80)
send_spi_bytes(FF FF FF FF FF FF FF FF)

# SPI command: Arm PTI transfer. The SV3 is now waiting to receive 128 bytes over the PTI interface.
# The SV3 will remain busy until all data is received.
send_spi_bytes(01 01 C6 00 00 00 00 02)
send_spi_bytes(FF FF FF FF FF FF FF FF)

# PTI write 128 bytes (32 x 32-bit words)

# SPI command: Process 'single-chunk' of size 128 bytes (0x80 bytes).
# Sends the chunk to User Pattern Slot 5, in this example.
send_spi_bytes(01 02 32 00 00 00 00 05)
Send_spi_bytes(FF FF FF FF FF FF FF FF)
send_spi_bytes(01 03 60 00 00 00 00 80)
send_spi_bytes(FF FF FF FF FF FF FF FF)

# SPI command: Read back length of pattern in User Pattern Slot 5, in this example.
send_spi_bytes     (00 03 11 00 00 00 00 00)
Expected response: (07 07 07 07 07 07 07 07)
send_spi_bytes     (FF FF FF FF FF FF FF FF)
Expected response: (00 00 00 80 00 00 00 07)
```

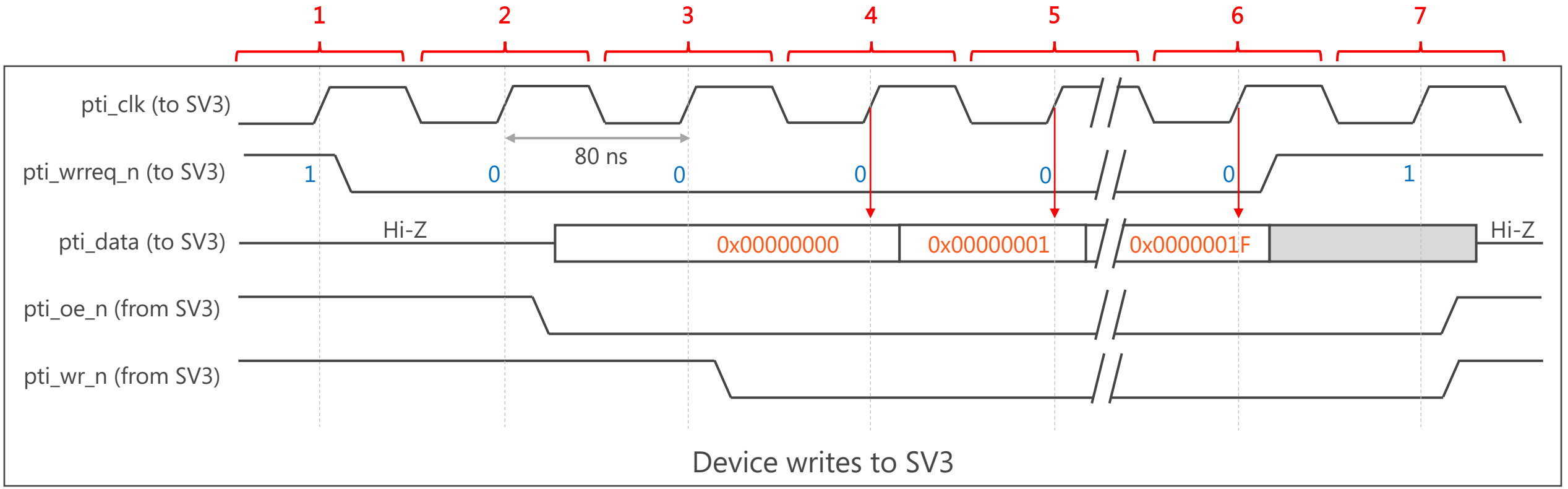

introspect technology

Example 1: writing 1024 bits to SV3 via PTI, PTI transfer

```
# PTI write chunk #1 (1024 bits / 128 bytes)
#-----------------------------------------------------

# PTI write 128 bytes (32 x 32-bit words)
# Note: C is clock (0 for first half of UI, 1 for second half of UI), Z is tristate.
# The clock must be continuously running!
+pti_clk (12.5 MHz)
|+pti_reset_n
||+pti_rdreq_n
|||+pti_wrreq_n
||||+pti_data[31]                    +pti_data[0]
|||||                                |
C111ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ (default state)
C110ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ <- This data is not transferred.
C110000000000000000000000000000000 <- This data is not transferred.
C110000000000000000000000000000000 <- 1st word transferred in PTI write: pti_data[31:0] = 0x00000000 = 0
C110000000000000000000000000000001 <- 2nd word transferred in PTI write: pti_data[31:0] = 0x00000001 = 1
C110000000000000000000000000000010 <- 3rd word transferred in PTI write: pti_data[31:0] = 0x00000010 = 2
C110...
C110000000000000000000000000011101 <- 30th word transferred in PTI burst: pti_data[31:0] = 0x0000001D = 29
C110000000000000000000000000011110 <- 31st word transferred in PTI burst: pti_data[31:0] = 0x0000001E = 30
C110000000000000000000000000011111 <- 32nd word transferred in PTI burst: pti_data[31:0] = 0x0000001F = 31
C111ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ (return to default state)
```

Example 1: writing 1024 bits to SV3 via PTI, PTI Timing



introspect technology

Example 2: writing 8188 bytes to SV3 via PTI, SPI commands

```
# PTI write chunk #1 (8188 bytes)
#-------------------------------------------------------
# SPI command: Set PTI transfer length to 8188 bytes (0x1FFC bytes)
send_spi_bytes(01 01 C4 00 00 00 1F FC)
send_spi_bytes(FF FF FF FF FF FF FF FF)

# SPI command: Arm PTI transfer. The SV3 is now waiting to receive 8188 bytes over the PTI interface.
# The SV3 will remain busy until all data is received.
send_spi_bytes(01 01 C6 00 00 00 00 02)
send_spi_bytes(FF FF FF FF FF FF FF FF)

# PTI write 8188 bytes (2047 x 32-bit words)

# SPI command: Process 'single-chunk' of size 8188 bytes (0x1FFC bytes).
# Sends the chunk to User Pattern Slot 7, in this example..
send_spi_bytes(01 02 32 00 00 00 00 07)
Send_spi_bytes(FF FF FF FF FF FF FF FF)
send_spi_bytes(01 03 60 00 00 00 1F FC)
send_spi_bytes(FF FF FF FF FF FF FF FF)

# SPI command: Read back length of pattern in User Pattern Slot 7, in this example.
# Pattern has been padded 2 times in memory, so that it becomes a multiple of 64 bits.
# Note 8188 bytes read -> 16376 bytes stored in memory.
send_spi_bytes     (00 03 11 00 00 00 00 00)
Expected response: (07 07 07 07 07 07 07 07)
send_spi_bytes     (FF FF FF FF FF FF FF FF)
Expected response: (00 00 00 3F F8 00 00 07)
```

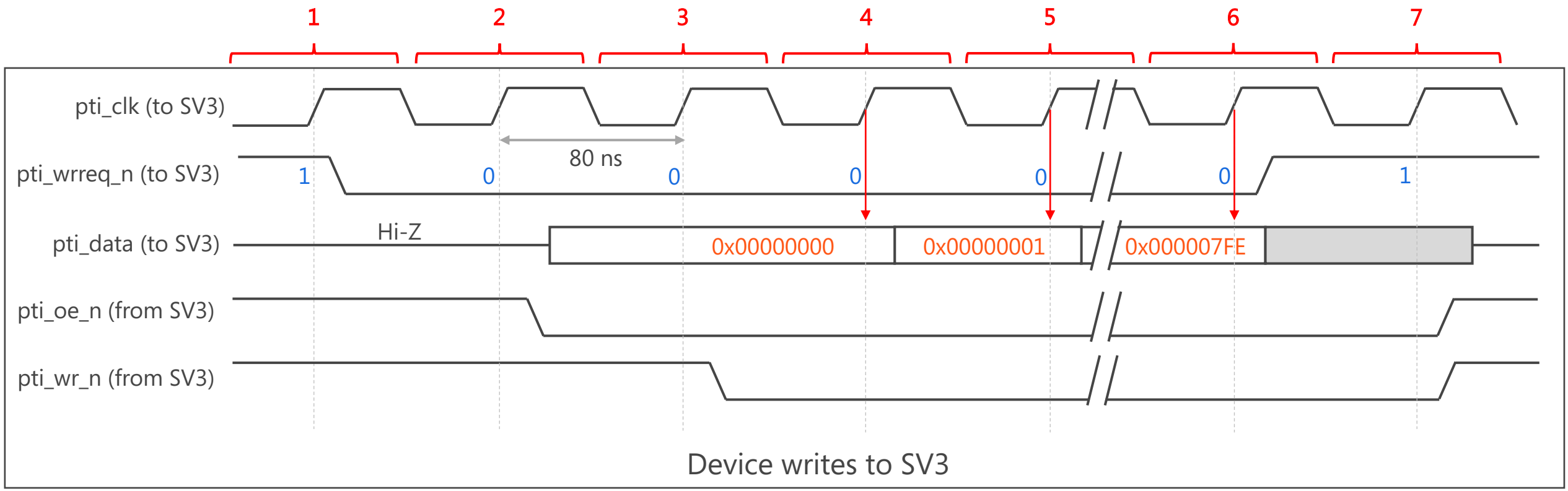

introspect technology

Example 2: writing 8188 bytes to SV3 via PTI, SPI commands

```
# PTI write chunk #1 (8188 bytes)
#-----------------------------------------------------

# PTI write 8188 bytes (2047 32-bit words)
# Note: C is clock (0 for first half of UI, 1 for second half of UI), Z is tristate.
# The clock must be continuously running!
+pti_clk (12.5 MHz)
|+pti_reset_n
||+pti_rdreq_n
|||+pti_wrreq_n
||||+pti_data[31]                  +pti_data[0]
|||||                             |
C111ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ (default state)
C110ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ <- This data is not transferred.
C110000000000000000000000000000000 <- This data is not transferred.
C110000000000000000000000000000000 <- 1st word transferred in PTI write: pti_data[31:0] = 0x00000000 = 0
C110000000000000000000000000000001 <- 2nd word transferred in PTI write: pti_data[31:0] = 0x00000001 = 1
C110000000000000000000000000000010 <- 3rd word transferred in PTI write: pti_data[31:0] = 0x00000010 = 2
C110...
C110000000000000000000011111111100 <- 2045th word transferred in PTI burst: pti_data[31:0] = 0x000007FC = 2044
C110000000000000000000011111111101 <- 2046th word transferred in PTI burst: pti_data[31:0] = 0x000007FD = 2045
C110000000000000000000011111111110 <- 2047th word transferred in PTI burst: pti_data[31:0] = 0x000007FE = 2046
C111ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ (return to default state)
```

Example 2: writing 8188 bytes to SV3 via PTI, PTI Timing



introspect technology


introspect
technology
To Learn More Visit:
INTROSPECT.CA