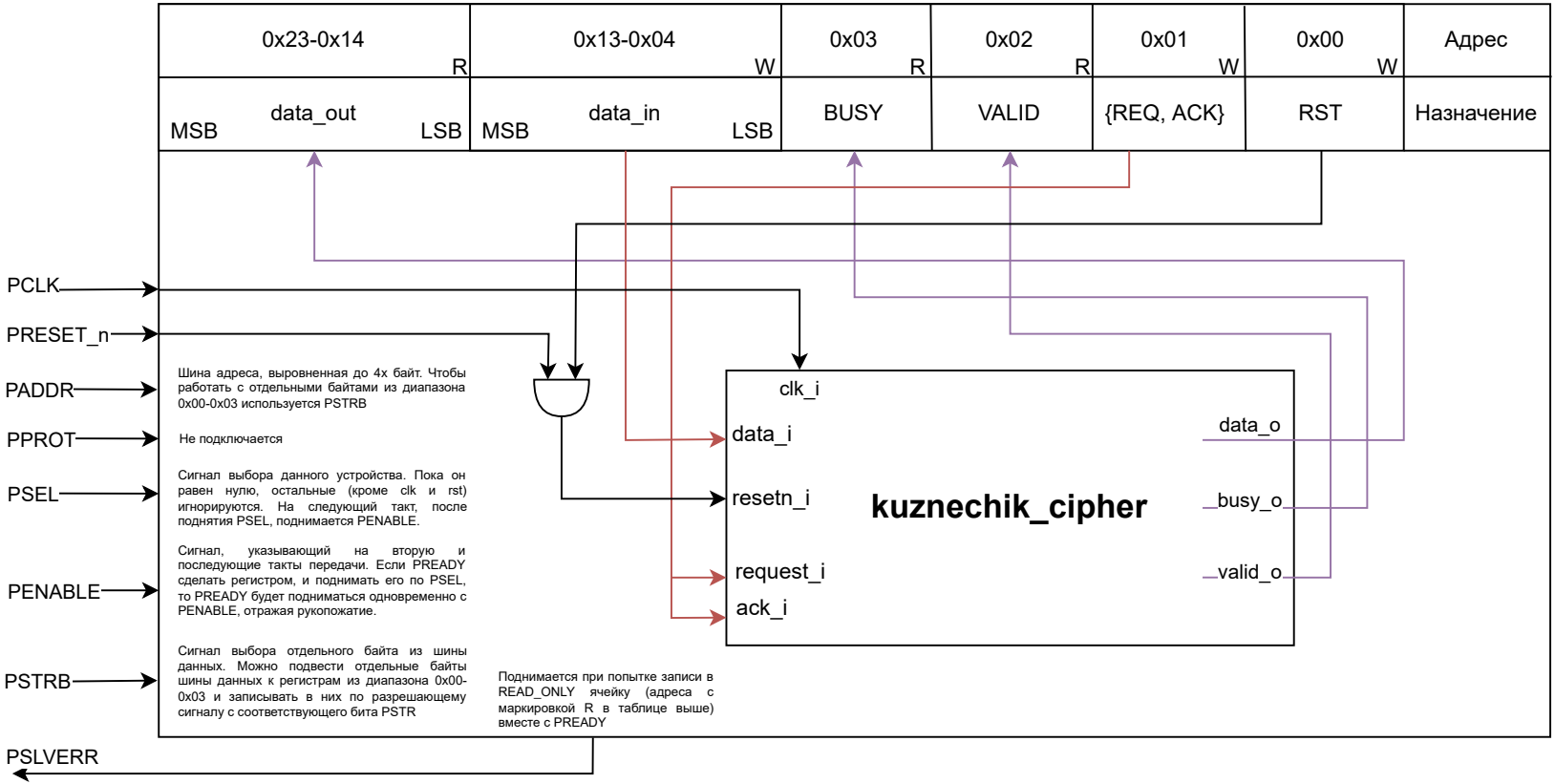
0x23-0x14
R
0x13-0x04
W
0x03
R
0x02
R
0x01
W
0x00
W
Адрес
MSB
data_out
LSB
MSB
data_in
LSB
BUSY
VALID
{REQ, ACK}
RST
Назначение
PCLK
PRESET_n
PADDR
PPROT
PSEL
PENABLE
PSTRB
PSLVERR
Шина адреса, выровненная до 4х байт. Чтобы работать с отдельными байтами из диапазона 0x00-0x03 используется PSTRB
Не подключается
Сигнал выбора данного устройства. Пока он равен нулю, остальные (кроме clk и rst) игнорируются. На следующий такт, после поднятия PSEL, поднимается PENABLE.
Сигнал, указывающий на вторую и последующие такты передачи. Если PREADY сделать регистром, и поднимать его по PSEL, то PREADY будет подниматься одновременно с PENABLE, отражая рукопожатие.
Сигнал выбора отдельного байта из шины данных. Можно подвести отдельные байты шины данных к регистрам из диапазона 0x00-0x03 и записывать в них по разрешающему сигналу с соответствующего бита PSTR
Поднимается при попытке записи в READ_ONLY ячейку (адреса с маркировкой R в таблице выше) вместе с PREADY
clk_i
data_i
resetn_i
request_i
ack_i
kuznechik_cipher
data_o
busy_o
valid_o

2.1 AMBA APB signals

Table 2-1 lists the APB signals.

Table 2-1 APB signal descriptions

Signal	Source	Description
PCLK	Clock source	Clock. The rising edge of **PCLK** times all transfers on the APB.
PRESETn	System bus equivalent	Reset. The APB reset signal is active LOW. This signal is normally connected directly to the system bus reset signal.
PADDR	APB bridge	Address. This is the APB address bus. It can be up to 32 bits wide and is driven by the peripheral bus bridge unit.
PPROT	APB bridge	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.
PSELx	APB bridge	Select. The APB bridge unit generates this signal to each peripheral bus slave. It indicates that the slave device is selected and that a data transfer is required. There is a **PSELx** signal for each slave.
PENABLE	APB bridge	Enable. This signal indicates the second and subsequent cycles of an APB transfer.
PWRITE	APB bridge	Direction. This signal indicates an APB write access when HIGH and an APB read access when LOW.
PWDATA	APB bridge	Write data. This bus is driven by the peripheral bus bridge unit during write cycles when **PWRITE** is HIGH. This bus can be up to 32 bits wide.
PSTRB	APB bridge	Write strobes. This signal indicates which byte lanes to update during a write transfer. There is one write strobe for each eight bits of the write data bus. Therefore, **PSTRB[n]** corresponds to **PWDATA[(8n + 7):(8n)]**. Write strobes must not be active during a read transfer.
PREADY	Slave interface	Ready. The slave uses this signal to extend an APB transfer.
PRDATA	Slave interface	Read Data. The selected slave drives this bus during read cycles when **PWRITE** is LOW. This bus can be up to 32-bits wide.
PSLVERR	Slave interface	This signal indicates a transfer failure. APB peripherals are not required to support the **PSLVERR** pin. This is true for both existing and new APB peripheral designs. Where a peripheral does not include this pin then the appropriate input to the APB bridge is tied LOW.

2.1.1 Data buses

The APB protocol has two independent data buses, one for read data and one for write data. The buses can be up to 32 bits wide. Because the buses do not have their own individual handshake signals, it is not possible for data transfers to occur on both buses at the same time.

3.1 Write transfers

This section describes the following types of write transfer:

- *With no wait states*
- *With wait states*.

3.1.1 With no wait states

Figure 3-1 shows a basic write transfer with no wait states.

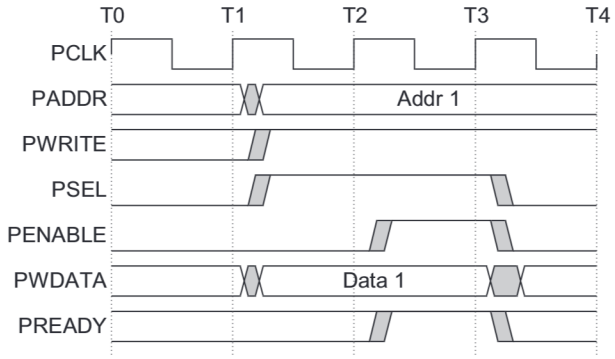

Figure 3-1 Write transfer with no wait states

At T1, a write transfer starts with address **PADDR**, write data **PWDATA**, write signal **PWRITE**, and select signal **PSEL,** being registered at the rising edge of **PCLK**. This is called the Setup phase of the write transfer.

At T2, enable signal **PENABLE**, and ready signal **PREADY**, are registered at the rising edge of **PCLK**.

When asserted, **PENABLE** indicates the start of the Access phase of the transfer.

When asserted, **PREADY** indicates that the slave can complete the transfer at the next rising edge of **PCLK**.

The address **PADDR**, write data **PWDATA**, and control signals all remain valid until the transfer completes at T3, the end of the Access phase.

The enable signal **PENABLE**, is deasserted at the end of the transfer. The select signal **PSEL**, is also deasserted unless the transfer is to be followed immediately by another transfer to the same peripheral.

3.1.2 With wait states

Figure 3-2 on page 3-3 shows how the slave can use the **PREADY** signal to extend the transfer. During an Access phase, when **PENABLE** is HIGH, the slave extends the transfer by driving **PREADY** LOW. The following signals remain unchanged while **PREADY** remains LOW:

- address, **PADDR**
- write signal, **PWRITE**
- select signal, **PSEL**
- enable signal, **PENABLE**
- write data, **PWDATA**
- write strobes, **PSTRB**
- protection type, **PPROT.**

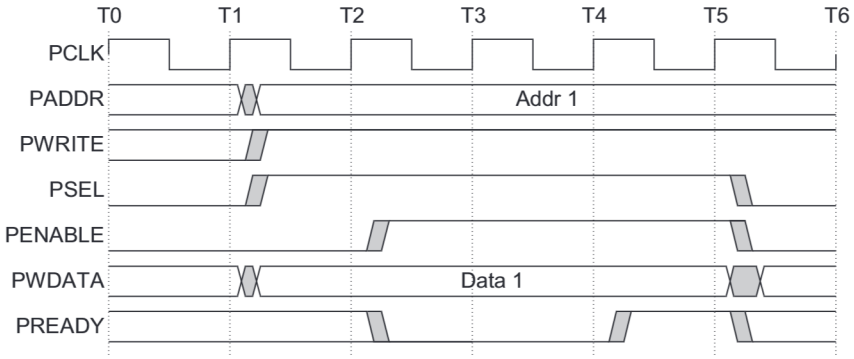

Figure 3-2 Write transfer with wait states

PREADY can take any value when **PENABLE** is LOW. This ensures that peripherals that have a fixed two cycle access can tie **PREADY** HIGH.

Note

It is recommended that the address and write signals are not changed immediately after a transfer, but remain stable until another access occurs. This reduces power consumption.

3.2 Write strobes

The write strobe signals, **PSTRB**, enable sparse data transfer on the write data bus. Each write strobe signal corresponds to one byte of the write data bus. When asserted HIGH, a write strobe indicates that the corresponding byte lane of the write data bus contains valid information.

There is one write strobe for each eight bits of the write data bus, so **PSTRB[n]** corresponds to **PWDATA[(8n + 7):(8n)]**. Figure 3-3 shows this relationship on a 32-bit data bus.

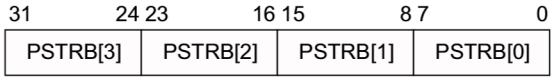

Figure 3-3 Byte lane mapping

Note

For read transfers the bus master must drive all bits of **PSTRB** LOW.

3.3 Read transfers

Two types of read transfer are described in this section:

- *With no wait states*
- *With wait states*.

3.3.1 With no wait states

Figure 3-4 shows a read transfer. The timing of the address, write, select, and enable signals are as described in *Write transfers* on page 3-2. The slave must provide the data before the end of the read transfer.

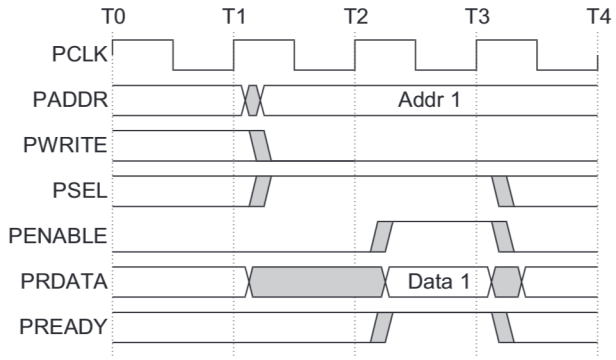

Figure 3-4 Read transfer with no wait states

3.3.2 With wait states

Figure 3-5 shows how the **PREADY** signal can extend the transfer. The transfer is extended if **PREADY** is driven LOW during an Access phase. The protocol ensures that the following remain unchanged for the additional cycles:

- address, **PADDR**
- write signal, **PWRITE**
- select signal, **PSEL**
- enable signal, **PENABLE**
- protection type, **PPROT**.

Figure 3-5 shows that two cycles are added using the **PREADY** signal. However, you can add any number of additional cycles, from zero upwards.

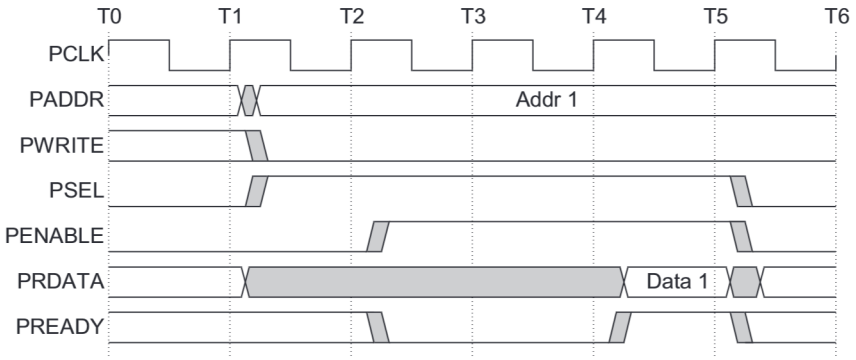

Figure 3-5 Read transfer with wait states

3.4 Error response

You can use **PSLVERR** to indicate an error condition on an APB transfer. Error conditions can occur on both read and write transactions.

PSLVERR is only considered valid during the last cycle of an APB transfer, when **PSEL**, **PENABLE**, and **PREADY** are all HIGH.

It is recommended, but not mandatory, that you drive **PSLVERR** LOW when it is not being sampled. That is, when any of **PSEL**, **PENABLE**, or **PREADY** are LOW.

Transactions that receive an error, might or might not have changed the state of the peripheral. This is peripheral-specific and either is acceptable. When a write transaction receives an error this does not mean that the register within the peripheral has not been updated. Read transactions that receive an error can return invalid data. There is no requirement for the peripheral to drive the data bus to all 0s for a read error.

APB peripherals are not required to support the **PSLVERR** pin. This is true for both existing and new APB peripheral designs. Where a peripheral does not include this pin then the appropriate input to the APB bridge is tied LOW.

3.4.1 Write transfer

Figure 3-6 shows an example of a failing write transfer that completes with an error.

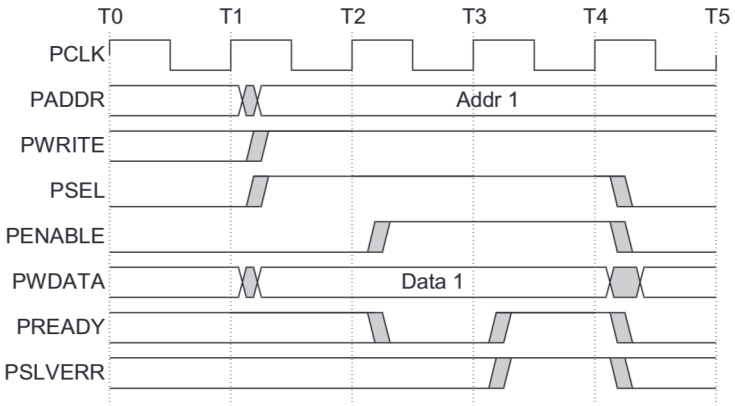

Figure 3-6 Example failing write transfer

3.4.2 Read transfer

A read transfer can also complete with an error response, indicating that there is no valid read data available. Figure 3-7 shows a read transfer completing with an error response.

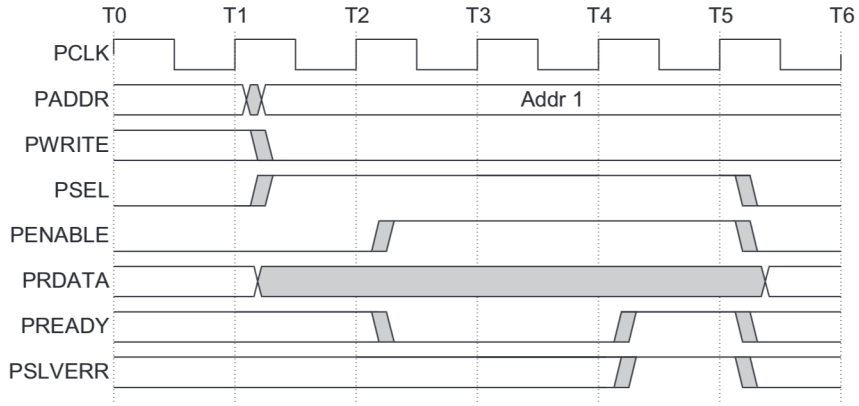

Figure 3-7 Example failing read transfer

3.4.3 Mapping of PSLVERR

When bridging:

From AXI to APB An APB error is mapped back to **RRESP**/**BRESP** = SLVERR. This is achieved by mapping **PSLVERR** to the AXI signals **RRESP[1]** for reads and **BRESP[1]** for writes.

From AHB to APB **PSLVERR** is mapped back to **HRESP** = ERROR for both reads and writes. This is achieved by mapping **PSLVERR** to the AHB signal **HRESP[0]**.

4.1 Operating states

Figure 4-1 shows the operational activity of the APB.

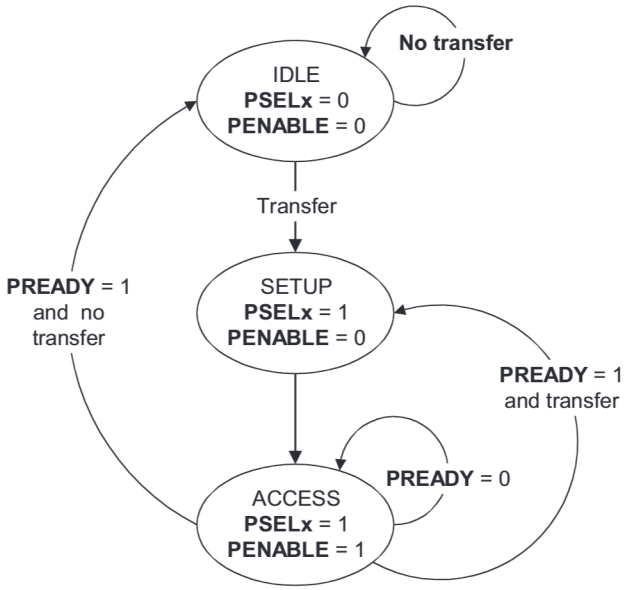

Figure 4-1 State diagram

The state machine operates through the following states:

IDLE This is the default state of the APB.

SETUP When a transfer is required the bus moves into the SETUP state, where the appropriate select signal, **PSELx**, is asserted. The bus only remains in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock.

ACCESS The enable signal, **PENABLE**, is asserted in the ACCESS state. The address, write, select, and write data signals must remain stable during the transition from the SETUP to ACCESS state.

Exit from the ACCESS state is controlled by the **PREADY** signal from the slave:

- If **PREADY** is held LOW by the slave then the peripheral bus remains in the ACCESS state.
- If **PREADY** is driven HIGH by the slave then the ACCESS state is exited and the bus returns to the IDLE state if no more transfers are required. Alternatively, the bus moves directly to the SETUP state if another transfer follows.