

# SPI\_BASIC

## REVISION HISTORY

Revision Number	Date	Description of Change	Author
V0.0	8/12/2022	Draft version	Shaoqiang
V0.1	9/21/2022	Add SPI_RDY	Shaoqiang
V1.0	10/09/2022	Change to design SPEC	Shaoqiang

## Table of Contents

SPI_BASIC .....	2
Introduction .....	2
Main features .....	2
Functional Details .....	3
Block Diagram .....	3
I/O description .....	3
State Machine .....	4
Internal SOF detection .....	5
SPI Communication Formats .....	6
TX timeout timer .....	7
COMM_CLEAR Command .....	8
SPI_RDY Behavior .....	8
Error Detection .....	9

## SPI\_BASIC

### Introduction

对HOST来说永远是SPI从机

SPI (Serial Peripheral Interface) is a synchronous communication protocol, which is always a slave device to the host. The main function is to allow communication between the host and daisy chain. Communication between the host and the device via SCK, CSB, MOSI, MISO and SPI\_RDY pins.

常规SPI是四线，这里多了一个SPI\_RDY

### Main features

The SPI module has the following features:

- Support full duplex (HWR007\_SPI\_BASIC) SCLK默认为0, MOSI和MISO默认为1
- Internal SOF detection based on CSB falling edge (HWR001\_SPI\_BASIC) 基于CSB下降沿, 检测内部SOF
- Bit captured on low to high clock transitions and propagated on high to low clock transition (HWR002\_SPI\_BASIC) 时钟上升沿capture bit; 时钟下降沿propagate bit
- For each byte at SPI interface, MSB transmitted first (HWR002\_SPI\_BASIC) SPI DATA BYTE: 先发高bit
- In command frame, MISO will remain "1" (HWR002\_SPI\_BASIC) 对于Command frame来说, RESP恒为0, 因此MISO保持为1 (空闲态)
- Take over the daisy chain communication (except tone communication) of either south or north (HWR002\_SPI\_BASIC)
- Extract rx\_data[8:0] into RX FIFO from A2D\_SPI\_SCLK and A2D\_SPI\_MOSI, MSB is SOF bit (HWR001\_SPI\_BASIC) Extract rx\_data[8:0]到RX\_FIFO中, MSB是SOF bit。  
把TX\_FIFO中的frame传输出去
- Propagate frame in TX FIFO on A2D\_SPI\_SCLK and D2A\_SPI\_MISO (HWR005\_SPI\_BASIC)
- Internal communication clear pattern detection, reset TX FIFO and RX FIFO when COMM\_CLEAR is detected (HWR001\_SPI\_BASIC) 检测到COMM\_CLR, 会reset TX\_FIFO和RX\_FIFO
- Transmit next RX FIFO data and SPI\_SOF bit when COPY\_NXT is detected (HWR005\_SPI\_BASIC) COPY\_NXT检测到之后, 会传输下一个RX\_FIFO data和SPI\_SOF bit给Daisy Chain Transceiver模块?
- Internal 60us timer start to count by entering response frame or receiving tx\_data[7:0] (HWR007\_SPI\_BASIC) 内部的60us timer的作用?
- Additional SPI\_RDY interface to inform the host that it can be written or read (HWR009\_SPI\_BASIC) SPI\_RDY的作用是: 通知host, 自己可以被读写了
- Internal FIFO overflow and underflow fault detection FIFO 空满的判断功能
- The SCLK frequency of SPI interface support 2-6MHz (TR096)
- Transmit FIFO depth is 4 bytes\*2 FIFO的深度: Transmit: 4 bytes\*2  
receive: 8 bytes

为什么两种写法?

- Receive FIFO depth is 8 bytes

## Functional Details

### Block Diagram

TX\_FIFO和RX\_FIFO是针对桥和bridge之间还是桥和device之间，还是二者都有？【从下面的表述中，还是指MCU侧】

回答：从HOST发来的command，经RX移位寄存器，被存在RX\_FIFO中，然后送给Daisy Chain；从菊花链回来的response，被存在TX\_FIFO中，然后通过TX移位寄存器发给MISO

The SPI allows communication between host and daisy chain. In the command frame, the command sent by the host is stored in the Receive FIFO (RX FIFO) after being received by the RX shift register, and then sent to the daisy chain. Response frame of daisy chain will be stored in the Transmit FIFO (TX FIFO) and then sent to the MISO via the TX shift register.

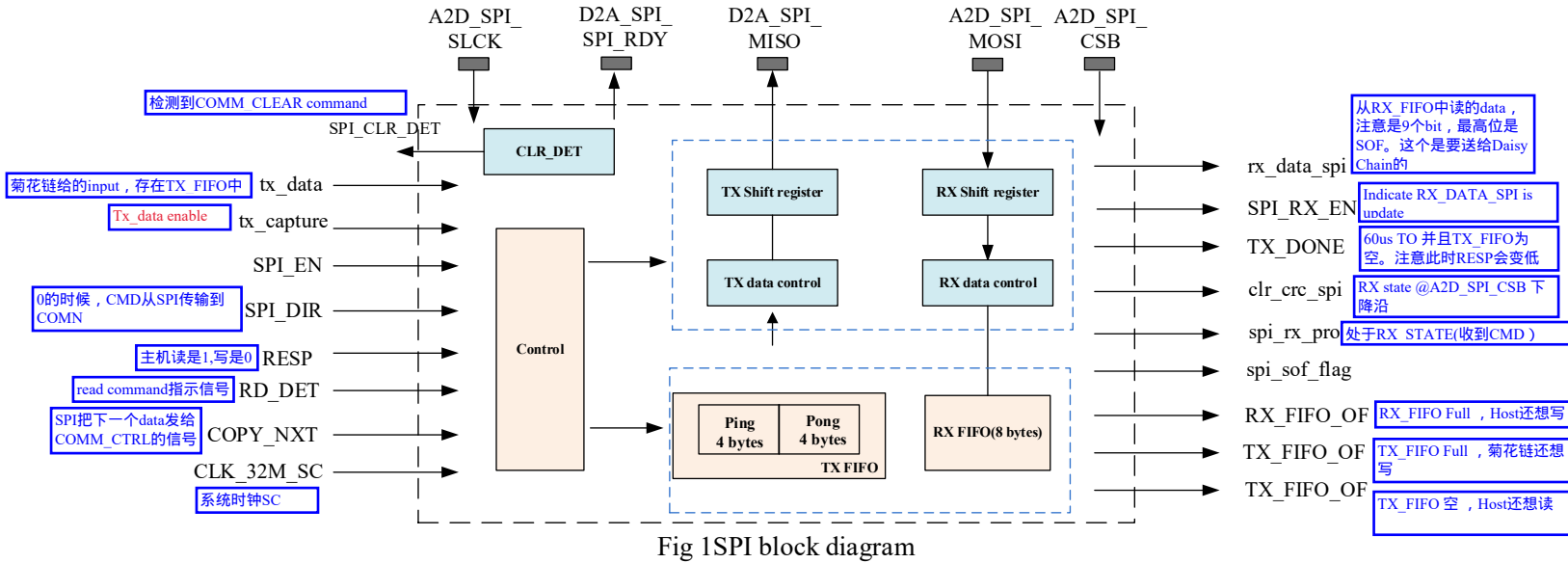


Fig 1 SPI block diagram

## I/O description

Table 1 SPI I/O description

Pin Name	Direction	Width	Duration	Description
A2D_SPI_SCLK	I	1'b	N/A	Clock input
A2D_SPI_MOSI	I	1'b	N/A	Master output slave input, MSB first
A2D_SPI_CS	I	1'b	N/A	Chip selection input
COPY_NXT	I	1'b	N/A	SPI to send next data to COMM_CTRL
RD_DET	I	1'b	N/A	D2A_SPI_RDY (h → l), indicate is read command
SPI_RESP	I	1'b	N/A	RESP = 1, indicate is maser read state RESP = 0, indicate is master write state
tx_data	I	8'b	N/A	Daisy chain input data, store to TX FIFO
tx_capture	I	1'b	N/A	Tx_data enable
SPI_EN	I	1'b	N/A	Enable SPI
SPI_DIR	I	1'b	N/A	"1" for north interface, "0" for south interface
D2A_TX_EN_N	I	1'b	N/A	North interface enable
D2A_TX_EN_S	I	1'b	N/A	South interface enable
CLK_32M_SC	I	1'b	N/A	System clock
CLK_SLOW_SC	I	1'b	N/A	Used to count RX timeout
resetb_CLK	I	1'b	N/A	Power on reset of CLK_32M_SC
CLK_REG_SC	I	1'b	N/A	8M clock



- P\_F1W: Enter the write TX FIFO1 state from the RX state
- F1RR2W: Read TX FIFO1 and write TX FIFO2
- TO\_F1R\_F2R: TX FIFO1 is not empty while TX FIFO2 is timeout
- F2R: Read TX FIFO2 only
- F1R\_F2R: TX FIFO1 is not empty while TX FIFO2 is full
- F2W: Write TX FIFO2 only
- F1W: Write TX FIFO1 only
- F1WF2R: Read TX FIFO2 and write TX FIFO1
- TO\_F2R\_F1R: TX FIFO2 is not empty while TX FIFO1 is timeout
- F1R: Read TX FIFO1 only
- F2R\_F1R: TX FIFO1 is not empty while TX FIFO2 is full
- END1: F1W/P\_F1W state but TX FIFO1 empty & TX FIFO1 timeout or enter F2W state but TX FIFO2 empty & TX FIFO2 timeout
- END2: F1R state and FIFO1 is empty or F2R state and FIFO2 is empty

### Internal SOF detection

(HWR001\_SPI\_BASIC)

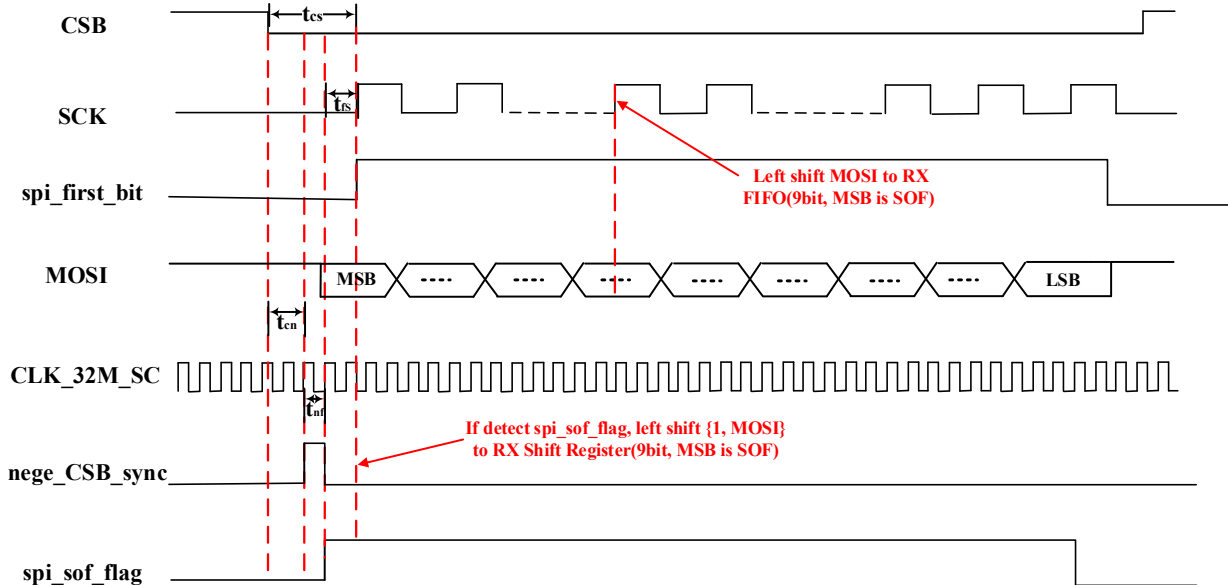


Fig 3 Schematic diagram of internal SOF detection

The calculation formula of  $t_{cs}$  is as follows:

$$t_{cs} = t_{cn} + t_{nf} + t_{fs}$$

$t_{cn}$  is the time between the falling edge of CSB and the rising edge of CSB\_sync obtained by internal synchronization, with a minimum of 2 CLK\_32M\_SC. The next clock cycle produces spi\_sof\_flag,  $t_{nf}=1$

CLK\_32M\_SC. To ensure spi\_sof\_flag stably sampled on the SCK rising edge,  $t_{fs} > 1 \text{ CLK\_32M\_SC}$  is required, so  $t_{cs} > 4 \text{ CLK\_32M\_SC}$  (125ns) is obtained. To broaden the time,  $t_{cs} \geq 200\text{ns}$ .

The RX shift register and RX FIFO have a storage unit of 9bit, bit[8] stores the SOF flag, and bit[7:0] is one byte of data received. At each SCLK rising edge, the MOSI is shifted left from the lowest position into the shift register. When the first bit(MSB) of each byte detects spi\_sof\_flag, it shifts {1, MOSI} left into the shift register. For other non-MSB bits, only the MOSI is shifted.

### SPI Communication Formats

#### (HWR007\_SPI\_BASIC, HWR005\_SPI\_BASIC, HWR002\_SPI\_BASIC)

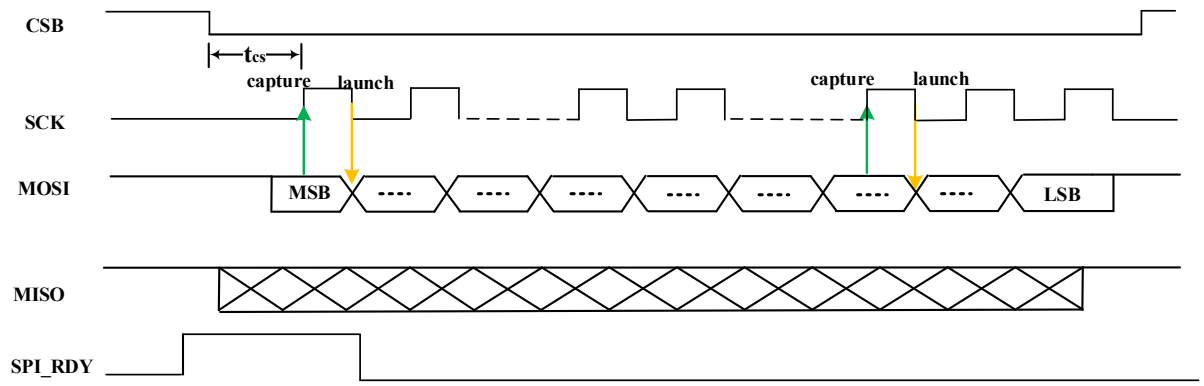
Although the SPI interface supports full duplex, since the daisy chain only supports half duplex, it is actually half duplex. It means that only command frame or response frame is transmitted between the MCU and the device at a given time. During master reading mode, MOSI is ignored by the device. SPI will capture data on low to high clock transitions and propagate on high to low clock transition. For each byte at SPI interface, MSB transmitted first. The communication timing diagram is shown in Figure 4.

#### Command Frame

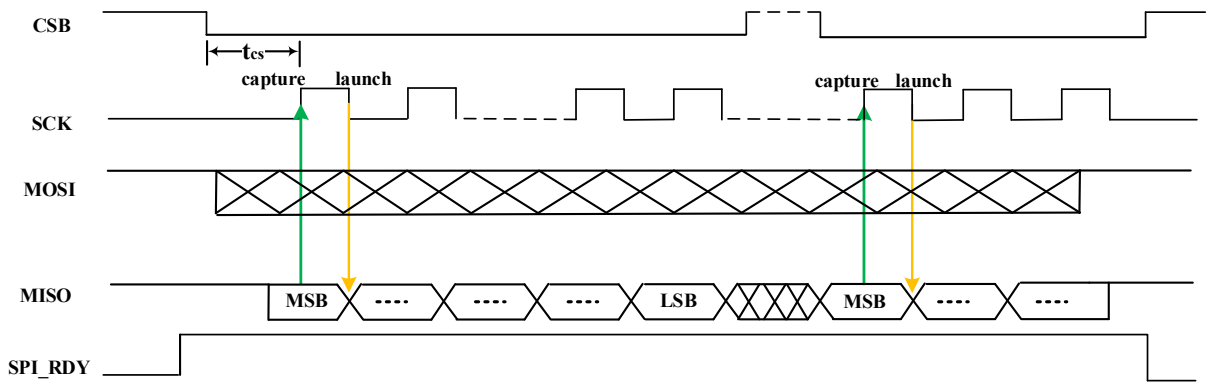
When command is sent from the MCU to the device, the MISO will remain 1. In command frame, the interval between the falling edge of CSB and the first rising edge of the SCK ( $t_{cs}$ ) must be greater than 200ns. Pulling CSB high during the transmission of a command frame is not allowed, if this is done, the frame command will be considered an invalid command.

#### Response Frame

In response frame, the host must send 8 SCK clock cycles when it detects that SPI\_RDY is 1, otherwise it may receive an incomplete response frame. MISO is wrong when SPI\_RDY is low. If host sends SCK when SPI\_RDY is low, SPI\_BASIC will generate a TX\_FIFO\_OF error.



(a)



(b)

Fig 4SPI communication timing diagram of (a) command frame, and (b) response frame.

### TX timeout timer

(HWR002\_SPI\_BASIC, HWR007\_SPI\_BASIC)

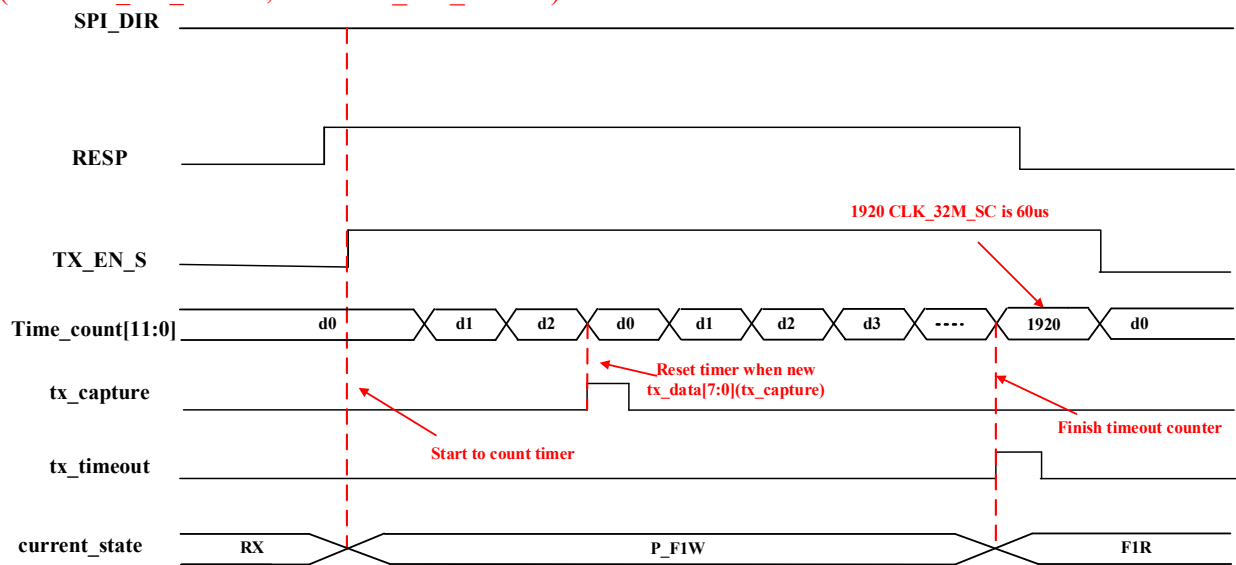


Fig 5 The timing diagram of TX timeout timer

The condition for SPI\_BASIC to enter TX from RX is:

$$\text{TX\_state} = \text{SPI\_EN} \& (\text{SPI\_DIR} ? \text{TX\_EN\_N} : \text{TX\_EN\_S}) \& \text{RESP}$$

When SPI\_EN is 1 and RESP is 1, SPI\_DIR is required. If SPI\_DIR is 0, SPI receives data from the south port of the daisy chain. When TX\_EN\_S is 1, SPI enters TX. In contrast, if SPI\_DIR is 1, SPI receives data from the north port of the daisy chain. When TX\_EN\_N is 1, SPI enters TX.

The counter starts counting at the beginning of entering TX. When a new tx\_data [7:0] is received the counter will restart counting. When the counter reaches 1920, the timeout flag will be generated and exit the write TX FIFO mode.

### COMM\_CLEAR Command

The COMM\_CLEAR command is defined as: eight consecutive bits of 0 are detected after the falling edge of the CSB as shown in Fig 3. COMM\_CLEAR command will be detected during the entire process when CSB is low, if detected, will reset TX and RX FIFO. After receiving the COMM\_CLEAR command, the SPI immediately enters the receive command mode, which means that the host must send a new command but cannot receive the response frame at this time.

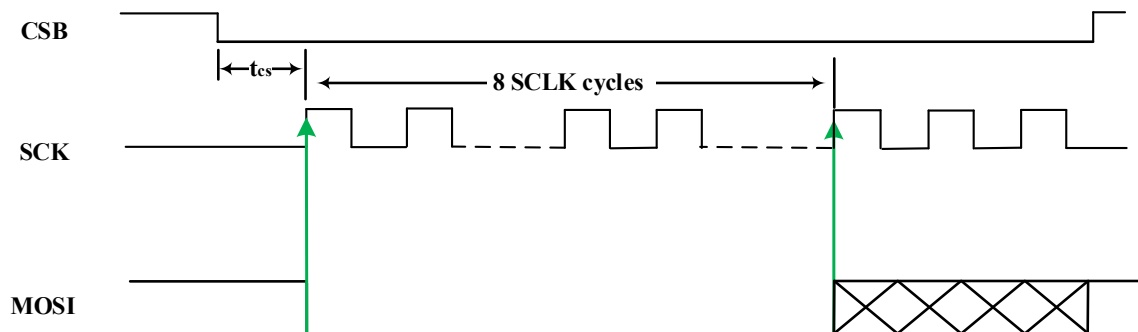


Fig 6 SPI COMM\_CLEAR command

### SPI\_RDY Behavior

(HWR010\_SPI\_BASIC, HWR011\_SPI\_BASIC)

SPI\_RDY is required because the depth of the TX FIFO is limited. Data overflow occurs if the host requests more than 256 bytes and the host does not service (read data) the device in time. SPI\_RDY indicates to the host that a certain amount of data is ready to be read or written.

In the command frame, when the data in the RX FIFO is no more than 2 bytes, SPI\_RDY will become high. When the data in the RX FIFO is greater than 4 bytes, SPI\_RDY will become low. When the SPI



receives the COMM\_CLEAR command, it will enter the receive command mode, and SPI\_RDY will become high regardless of the previous state.

In the response frame, SPI will receive the data sent by the daisy chain. When TX FIFO (ping or pong FIFO) is full or the TX timeout timer is expired (TX timeout is defined as no response from the daisy chain within 60us in response frame), SPI\_RDY will be pulled high, indicating that host can read the response frame stored by the SPI. SPI\_RDY will continue be high until the current FIFO is empty. Note that if another TX FIFO is full before the current FIFO is empty, TX\_FIFO\_OF error will be generated when daisy chain sends new data. When the TX FIFOs are all empty, the SPI will automatically enter the receive command mode and SPI\_RDY will become high after 6us. The behavior of SPI\_RDY is summarized in Table 1.

Note:

When SPI\_RDY is low, it lasts at least 2us before it becomes high.

Table1. SPI\_RDY Behavior Summary

	HIGH -> LOW	LOW -> HIGH
Host write	RX FIFO has >4 bytes	RX FIFO has <=2 bytes
Host read	Device receive first byte of read command	Ping or Pong FIFO is full
	Ping or Pong FIFO being read becomes empty (the last byte in the FIFO is read)	TX FIFO time out happened
		TX FIFO timeout and TX FIFO are all read empty, after 6us

## Error Detection

### (HWR009\_SPI\_BASIC)

SPI\_BASIC will detect three kinds of errors. Errors include the following:

**RX\_FIFO\_OF**: in command frame, when RX FIFO is full but host still writes data to SPI\_BASIC.

`lpspi_rx_full && lpspi_end_byte`

**TX\_FIFO\_OF**: in response frame, host did not read the data from the TX FIFO in time, when the TX FIFOs are all full but the daisy chain still writes data to SPI\_BASIC.

`(lpspi_tx_full && lpspi_tx_full2 && tx_capture) ||`

`((current_state == F2R_F1R || current_state == F1RF2R) && tx_capture )`

**TX\_FIFO\_UF:** in response frame, host sends SCK clocks when SPI\_RDY is low.

$(\sim D2A\_SPI\_SPI\_RDY) \&\& (current\_state \neq RX) \&\& pose\_SCLK$