

VLSI Technology
11.2.2020

Serial Peripheral Interface
Specification

Email: quanghungbk1999@gmail.com

I) Đặc điểm thiết kế:

Thiết kế SPI này hỗ trợ các đặc điểm sau:

- Truyền song công (full-duplex)
- Hoạt động ở 2 chế độ: Master hoặc Slave
- Tần số hoạt động tối đa khi ở chế độ Master là $\frac{1}{2} f_{PCLK}$
- Tần số hoạt động tối đa khi ở chế độ Slave ???
- Hỗ trợ tốc độ truyền khác nhau ở chế độ Master
- Hỗ trợ 4 chế độ định thì truyền nhận khác nhau thông qua các bit Clock Polarity và Clock Phase
- Cho phép chọn MSB hay LSB truyền trước
- Giao tiếp APB4
- Chế độ nâng cao: FIFO truyền nhận
 - 2 FIFO truyền nhận riêng biệt với độ sâu là 64
 - Cấu hình thời gian trễ giúp giao tiếp với các ngoại vi SPI khác có tốc độ thấp hơn
- Tùy theo cách sử dụng SPI có thể hoạt động theo 3 cách:
 - Chỉ nhận
 - Chỉ gửi
 - Vừa gửi vừa nhận

II) Mô tả tín hiệu SPI:

4 tín hiệu cơ bản:

1) MISO:

- Tín hiệu MISO được cài đặt là ngõ vào khi ở chế độ Master và ngõ ra cho chế độ Slave.
- Tín hiệu MISO của thiết bị Slave phải ở trạng thái trở kháng cao khi Slave đó không được chọn.

2) MOSI:

- Tín hiệu MOSI được cài đặt là ngõ ra khi ở chế độ Master và ngõ vào khi ở chế độ Slave.

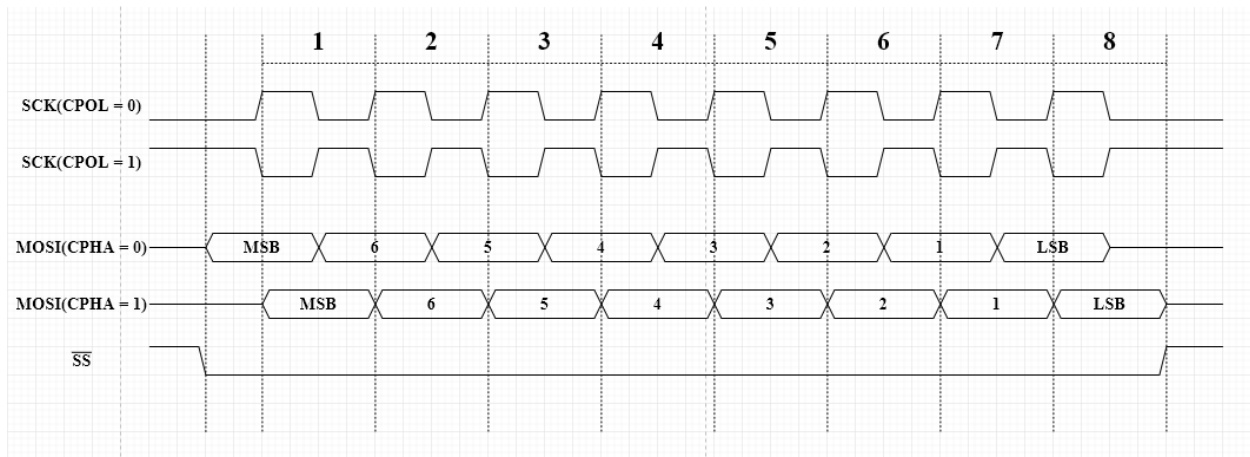
3) SCK:

- Có 4 quan hệ định thì có thể được chọn dựa vào 2 bit điều khiển CPOL và CPHA

- Master và Slave phải được cài đặt ở cùng chế độ định thì.
- Master luôn đặt dữ liệu lên MOSI $\frac{1}{2}$ chu kỳ trước cạnh lên (xuống) của clock.
- 2 bit (SPR0 và SPR1) sẽ chọn tốc độ bit của SPI.
- Bit SPR2X khi được set sẽ tăng gấp đôi tốc độ bit của SPI.

4) \overline{SS} :

- Tín hiệu giúp chọn slave hoạt động, tín hiệu này phải ở mức thấp khi có một transaction diễn ra.
- Tín hiệu \overline{SS} ở Master phải được đặt ở mức cao, nếu không cờ báo lỗi MODF sẽ được đặt.



Giản đồ định thì SPI

III) **SPI Registers:**

Thiết kế SPI này sẽ gồm 7 thanh ghi và 2 FIFO, trong đó 2 thanh ghi đệm sẽ dùng chung địa chỉ với FIFO truyền nhận, thanh ghi SPIDSR không cho phép người dùng ghi/đọc dữ liệu nên sẽ không có địa chỉ:

Bảng địa chỉ offset các thanh ghi:

Thanh ghi	Địa chỉ
SPICR	0x0
SPISR	0x4
SPITXCR	0x8
SPIRXCR	0xC
SPITXBUF(T_FF)	0x10
SPIRXBUF(R_FF)	0x14

***Các thanh ghi cấu hình và trạng thái:**

1) SPI Control Register (SPICR)

31:24	SWR	ENHFF	DORD	MSTR	CPOL	CPHA	MCLKSEL	TALK
23:16			FFD5	FFD4	FFD3	FFD2	FFD1	FFD0
15:8	DATALEN4	DATALEN3	DATALEN2	DATALEN1	DATALEN0	OVINTE	SPIRXE	SPITXE
7:0	SPBR7	SPBR6	SPBR5	SPBR4	SPBR3	SPBR2	SPBR1	SPBR0

Bảng mô tả thanh ghi SPICR

Bit	Tên	Giá trị	Mô tả
31	SWR (Software Reset)	0	Toàn bộ các cờ ngắt sẽ bị xóa. SPI không cho phép truyền nhận dữ liệu. Nếu bit SWR bị xóa lúc việc truyền nhận đang diễn ra thì dữ liệu đang gửi trong thanh ghi dịch sẽ vẫn tiếp tục được dịch đi.

		1	Bit SWR được bật lên thì SPI mới được phép hoạt động
30	ENHFF (Enhanced FIFO)	0	Chế độ nâng cao SPI FIFO tắt
		1	Chế độ nâng cao SPI FIFO bật
29	DORD (Data Order)	0	MSB truyền trước
		1	LSB truyền trước
28	MSTR (Master Mode)	0	SPI hoạt động ở chế độ Slave
		1	SPI hoạt động ở chế độ Master
27	CPOL (Polar)	0	Khi không hoạt động SPICLK ở mức thấp
		1	Khi không hoạt động SPICLK ở mức cao
26	CPHA (Phase)	0	Master đặt dữ liệu lên MOSI ½ chu kỳ trước cạnh clock đầu tiên (lên hoặc xuống) và lấy mẫu dữ liệu ở cạnh clock đó.
		1	Master đặt dữ liệu lên MOSI tại cạnh clock đầu tiên và lấy mẫu ở cạnh clock tiếp theo ngược với cạnh clock trước.
25	MCLKSEL (Master Clock Select)	0	Clock dùng để tạo xung SPICLK được lấy từ Bus ngoại vi (pclk của giao tiếp APB4)
		1	Clock dùng để tạo xung SPICLK được lấy từ các nguồn clock khác(osc, PLL, ...)
24	TALK	0	+ Ở chế độ Master: hoạt động của SPI vẫn bình thường tuy nhiên các ngõ ra MOSI, \overline{SS} và SPICLK sẽ được đặt ở trạng thái trở kháng cao. + Ở chế độ Slave: hoạt động của SPI vẫn bình thường tuy nhiên ngõ ra SOMI sẽ được đặt ở trạng thái trở kháng cao.
		1	Các chân MOSI, SOMI được đề cập ở trên sẽ hoạt động bình thường

			Note: bit TALK hỗ trợ việc kết nối nhiều Slave và Master với nhau
23:22	X	x	x
21:16	FFD (FIFO Depth)	bbbbbb	Các bit quy định độ sâu FIFO được sử dụng (độ sâu tối đa là 64) $FIFO\ depth = FFD + 1$
18	OVINTE (Overflow Interrupt Enable)	0 1	Cấm ngắt từ cờ OVF Cờ OVF được bật lên sẽ tạo tín hiệu ngắt SPIRXINT
17	SPIRXE (SPI receive interrupt enable)	0 1	Cấm ngắt từ cờ SPIF Cờ ngắt SPIF được bật lên sẽ tạo tín hiệu ngắt SPIRXINT
16	SPITXE (SPI transfer interrupt enable)	0 1	Cấm ngắt từ cờ WCOL Cờ ngắt WCOL được bật lên sẽ tạo tín hiệu ngắt SPITXINT
15:11	DATALEN (Data Length)	bbbbbb	Độ dài của một dữ liệu được truyền nhận $Data\ length = DATALEN + 1$
10:8	X	x	x
7:0	SPBR (SPI baud rate)	bbbbbbbbb	Các bit quy định tốc độ baud của SPI 00000000: $shift_clock = pre_clock/2$ 00000001: $shift_clock = pre_clock/4$... $shift_clock = pre_clock / (SPBR*2 + 2)$

2) SPI Status Register (SPISR)

31:24	TXFFINT		TXFFST5	TXFFST4	TXFFST3	TXFFST2	TXFFST1	TXFFST0
23:16	RXFFINT		RXFFST5	RXFFST4	RXFFST3	RXFFST2	RXFFST1	RXFFST0
15:8	TXFFOVINT	RXFFOVINT	TXFFUNINT	RXFFUNINT				
7:0						WCOL	OVF	SPIF

Bảng mô tả thanh ghi SPISR

Bit	Tên	Giá trị	Mô tả
31	TXFFINT (Transfer FIFO Full FFD flag)	0 1	Cờ báo số lượng ô nhớ được ghi vào, trong FIFO truyền, đã bằng với dung lượng cài đặt trong các bit FFD
29:24	TXFFST (Transfer FIFO Status)	bbbbbb	Nội dung con trỏ ghi của FIFO truyền
23	RXFFINT (Receive FIFO full FFD flag)	0 1	Cờ báo số lượng ô nhớ được ghi vào, trong FIFO nhận, đã bằng với dung lượng cài đặt trong các bit FFD
21:16	RXFFST (Receive FIFO Status)	bbbbbb	Nội dung con trỏ ghi của FIFO nhận
15	TXFFOVINT (Transfer FIFO overflow flag)	0 1	Cờ báo overflow ở FIFO truyền (con trỏ ghi đã đuổi kịp con trỏ đọc nhưng vẫn còn yêu cầu ghi)
14	RXFFOVINT	0	

	(Receive FIFO overflow flag)	1	Cờ báo overflow ở FIFO nhận (con trỏ ghi đã đuổi kịp con trỏ đọc nhưng vẫn còn yêu cầu ghi)
13	TXFFUNINT (Transfer FIFO underflow flag)	0 1	Cờ báo underflow ở FIFO truyền (con trỏ đọc đã đuổi kịp con trỏ ghi nhưng vẫn còn yêu cầu đọc)
12	RXFFUNINT (Receive FIFO underflow flag)	0 1	Cờ báo underflow ở FIFO nhận (con trỏ đọc đã đuổi kịp con trỏ ghi nhưng vẫn còn yêu cầu đọc)
11:3	x	x	x
2	WCOL (Write Collision flag)	0 1	Khi không sử dụng FIFO, SPI sử dụng 1 buffer truyền nhằm chứa dữ liệu cần gửi, cờ WCOL báo người dùng biết dữ liệu trước đó trong thanh ghi này vẫn chưa được gửi nhưng đã có dữ liệu mới ghi vào
1	OVF (Overflow flag)	0 1	Khi không sử dụng FIFO, SPI sử dụng 1 buffer nhận để chứa dữ liệu nhận được, cờ OVF báo người dùng biết dữ liệu nhận được trước đó vẫn chưa được đọc ra nhưng đã có dữ liệu mới ghi vào
0	SPIF (SPI transfer complete flag)	0 1	Cờ SPIF được bật lên nhằm báo người dùng biết một chu kỳ truyền nhận vừa kết thúc. Ở chế độ Slave cờ SPIF chỉ có thể xóa bởi người dùng. Ở chế độ Master, cờ SPIF sẽ tự động xóa khi có một yêu cầu bắt đầu truyền nhận mới

3) SPI FIFO transfer control register (SPITXCR):

31:24	TXFFDL7	TXFFDL6	TXFFDL5	TXFFDL4	TXFFDL3	TXFFDL2	TXFFDL1	TXFFDL0
23:16								FFSTA
15:8	TXFFINTE	TXFFOVE	TXFFUNE					
7:0								TXFFRST

Bảng mô tả thanh ghi SPITXCR

Bit	Tên	Giá trị	Mô tả
31:24	TXFFDL	bbbbbbb	Các bit này quy định thời gian delay từ lúc phát hiện một chu kỳ gửi nhận kết thúc đến lúc bắt đầu một chu kỳ gửi nhận mới. Hỗ trợ delay từ 0:255 chu kỳ PCLK
23:15	x	x	x
16	FFSTA (FIFO Transfer Start)	0 1	SPI bắt đầu chu kỳ truyền đầu tiên, ở chế độ FIFO, sau khi cờ TXFFINT được bật lên SPI bắt đầu chu kỳ truyền đầu tiên, ở chế độ FIFO, sau khi có một yêu cầu ghi vào FIFO truyền.
15	TXFFINTE	0 1	Cấm ngắt từ cờ TXFFINT Cờ TXFFINT được bật lên sẽ tạo ngắt SPITXINT
14	TXFFOVE	0 1	Cấm ngắt từ cờ TXFFOVINT Cờ TXFFOVINT được bật lên sẽ tạo ngắt SPITXINT
13	TXFFUNE	0 1	Cấm ngắt từ cờ TXFFUNINT Cờ TXFFUNINT được bật lên sẽ tạo ngắt SPITXINT
12:8	x	x	x
7:1	x	x	x
0	TXFFRST	0	

		1	Con trỏ đọc, ghi của FIFO truyền sẽ bị xóa về 0
--	--	---	---

4) SPI FIFO receive control register (SPIRXCR):

31:24							
23:16							
15:8	RXFFINTE	RXFFOVE	RXFFUNE				
7:0							RXFFRST

Bảng mô tả thanh ghi SPIRXCR

Bit	Tên	Giá trị	Mô tả
31:16	x	x	X
15	RXFFINTE	0	Cấm ngắt từ cờ RXFFINT
		1	Cờ RXFFINT được bật lên sẽ tạo ngắt SPIRXINT
14	RXFFOVE	0	Cấm ngắt từ cờ RXFFINT
		1	Cờ RXFFOVF được bật lên sẽ tạo ngắt SPIRXINT
13	RXFFUNE	0	Cấm ngắt từ cờ RXFFUNINT
		1	Cờ RXFFUNINT được bật lên sẽ tạo ngắt SPIRXINT
12:1	x	x	x
0	RXFFRST	0	Con trỏ đọc, ghi của FIFO nhận sẽ bị xóa về 0
		1	

IV) Các cách sử dụng chế độ FIFO:

FIFO hỗ trợ hoạt động theo 3 chế độ:

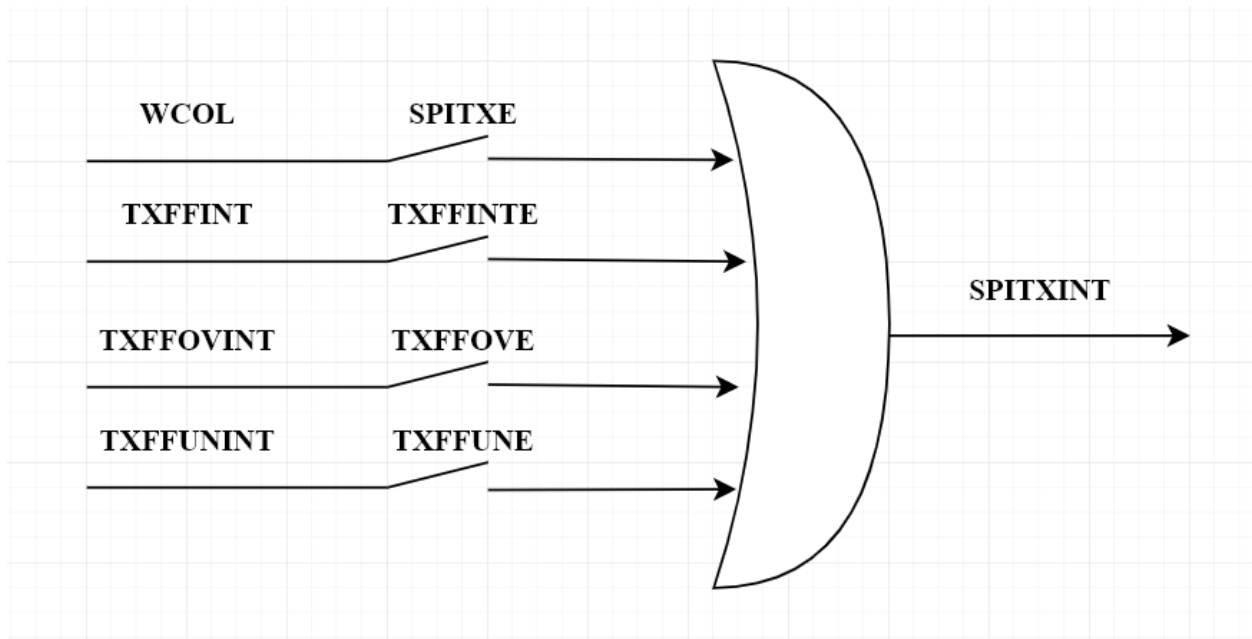
- **Truyền và nhận:** người dùng phải cho phép 2 cờ ngắt TXFFINT và RXFFINT, số lượng dữ liệu muốn gửi đi mỗi lần gửi nhận phải được ghi vào các bit TXFFD và RXFFD (tối đa 64 ở cả 2 FIFO). Sau khi ghi đủ số dữ liệu cần gửi (cờ TXFFINT được bật), phần mềm chờ đến khi có ngắt RXFFINT thì mới được đọc. Sau khi hoàn thành 1 vòng truyền nhận thì lần lượt bật tắt các bit TXFFRST và RXFFRST rồi bắt đầu vòng truyền nhận mới.
- **Chỉ gửi:** người dùng phải cho phép cờ ngắt TX

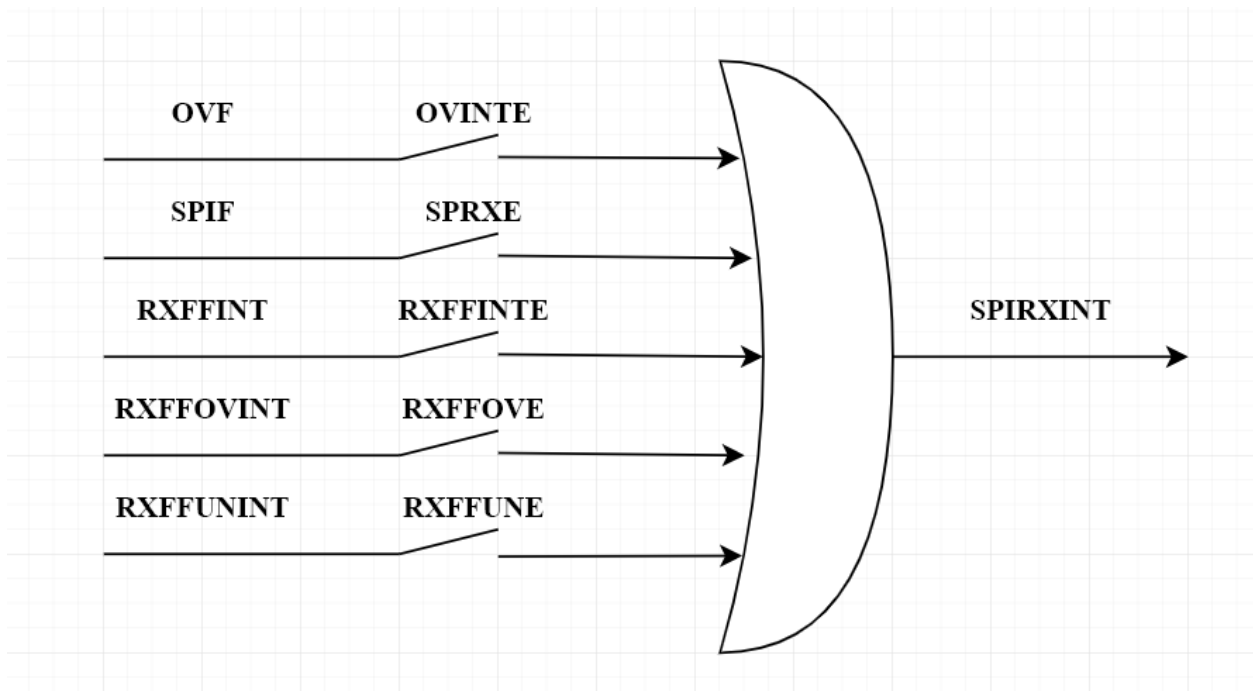
V) **Ngắt và Clock:**

1) **Ngắt:**

Ngoại vi SPI có thể tạo ra 2 nguồn ngắt tổng quát SPIRXINT và SPITXINT

Sơ đồ nguyên lý ngắt SPI:





2) Nguồn clock:

- Bit MCLKSEL sẽ chọn nguồn clock dùng để tạo xung dịch cho SPI khi nó ở chế độ MASTER

MCLKSEL = 0: Clock dịch được tạo ra từ clock của bus ngoại vi APB

MCLKSEL = 1: Clock dịch được tạo ra từ các nguồn clock khác (PLL, OSC, ...)

- Clock hoạt động của SPI khi ở chế độ Master:

$$\text{shift_clock} = \frac{\text{pre_clock}}{2 * \text{DATALEN} + 2}$$

