



VLSI Technology

NGOẠI VI SPI

Lê Quang Hưng

2020.4.23



**Đây là tài liệu mô tả ngoại vi SPI cho
lõi VG CPU được phát triển bởi VLSI
Technology**



LỜI NÓI ĐẦU

Tài liệu mô tả đặc điểm thiết kế và cấu trúc của ngoại vi SPI:

quanghungbk1999@gmail.com

...

Trân trọng.



MỤC LỤC

LỜI NÓI ĐẦU

MỤC LỤC

DANH SÁCH HÌNH

DANH SÁCH BẢNG

PHẦN MỀM SỬ DỤNG

THUẬT NGỮ VÀ VIẾT TẮT

1 ĐẶC ĐIỂM THIẾT KẾ

1.1 Giới thiệu

1.1.1 Đặc điểm hỗ trợ

1.1.2 Sơ đồ khối giao tiếp SPI

1.1.3 Tóm tắt tín hiệu giao tiếp SPI

1.2 Tổng quan các thanh ghi điều khiển và trạng thái

1.3 FIFO truyền và nhận

1.4 Ngắt SPI

1.4.1 Các bit cho phép ngắt của SPI (SPIINTER)

1.4.2 Các cờ ngắt raw interrupt SPI (SPIRINTR)

1.4.3 Các cờ ngắt interrupt SPI (SPIINTR)

1.4.4 Sơ đồ mô tả hoạt động ngắt SPI

1.5 Hoạt động SPI

1.4.1 Tổng quan hoạt động

1.4.2 Chế độ Master

1.4.3 Chế độ Slave

1.4.3 Định dạng clock

1.4.4 Tốc độ baud và clock dịch

2 CÁC THANH GHI CẤU HÌNH VÀ TRẠNG THÁI SPI

2.1 SPI Control Register (SPICR)



2.2 SPI Baud Rate Register (SPIBR)

2.3 SPI Interrupt Enable Register (SPIINTER)

2.4 SPI Status Register (SPISR)

2.5 SPI Raw Interrupt Register (SPIRINTR)

2.6 SPI Interrupt Register (SPIINTR)

3 HƯỚNG DẪN SỬ DỤNG SPI

Tài liệu tham khảo



DANH SÁCH HÌNH



DANH SÁCH BẢNG



PHẦN MỀM SỬ DỤNG

Phần mềm	Mục đích sử dụng
Microsoft Word 2013	Soạn thảo tài liệu hướng dẫn chi tiết
Microsoft Excel 2013	Tạo bảng tính và mô tả dữ liệu dạng bảng như bảng lệnh, bảng chu kỳ lệnh
Microsoft Visio 2013 Drawio	Vẽ hình minh họa
QuestaSim 10.2c	Mô phỏng RTL code
Quartus Prime Lite Edition 18.1	Khả tổng hợp RTL code



THUẬT NGỮ VÀ VIẾT TẮT

1

ĐẶC ĐIỂM THIẾT KẾ

1.1 Giới thiệu

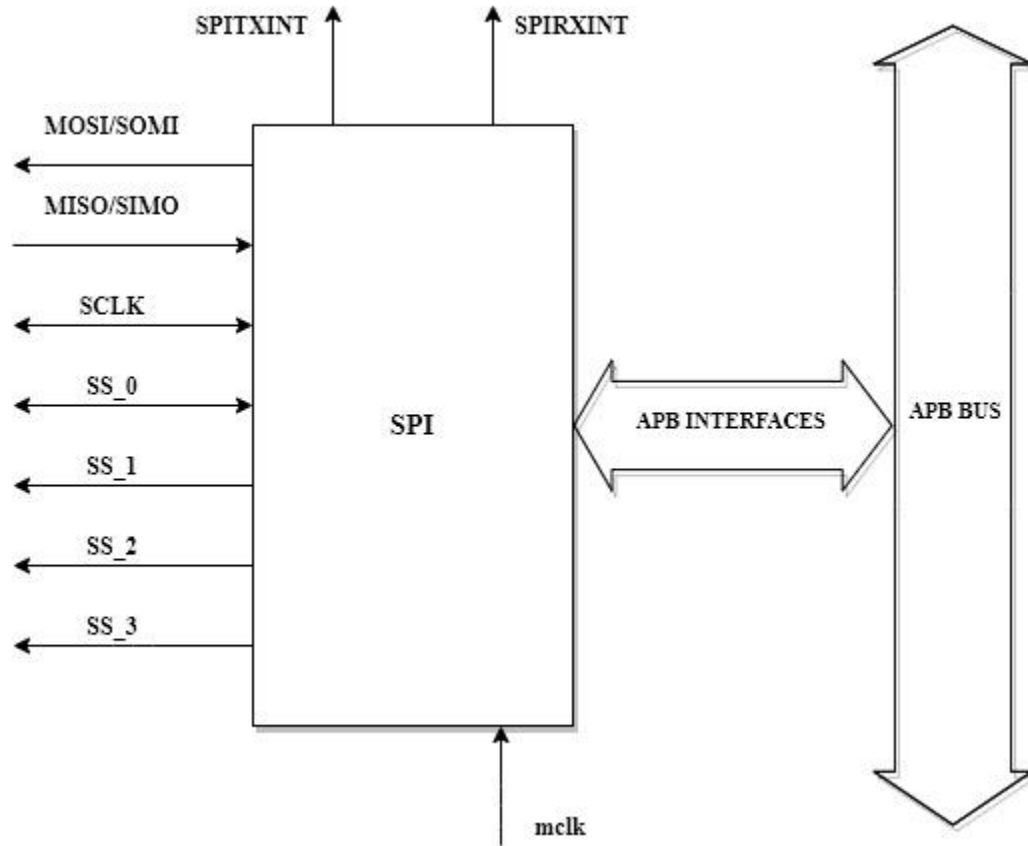
1.1.1 Đặc điểm hỗ trợ

SPI là một chuẩn truyền nhận nối tiếp đồng bộ cho giao tiếp giữa vi điều khiển và các ngoại vi. Một số đặc điểm được thiết kế SPI này hỗ trợ:

- Chế độ Master và Slave
- Chế độ truyền nhận bằng FIFO
- Cấu hình độ rộng của chuỗi dữ liệu truyền nhận (0:32 bit)
- Cấu hình độ trễ giữa các lần nhận và truyền dữ liệu mới khi truyền nhận các gói dữ liệu liên tục
- Cấu hình 256 tốc độ truyền nhận khác nhau
- Cấu hình MSB hay LSB truyền trước
- Hỗ trợ giao tiếp APB4
- Hỗ trợ giao tiếp với nhiều Slave (4)
- 4 định dạng clock truyền nhận (cài đặt thông qua 2 bit CPHA và CPOL)

1.1.2 Sơ đồ khối giao tiếp SPI

Hình 1-1: Sơ đồ khối tín hiệu giao tiếp SPI



1.1.3 Tóm tắt tín hiệu giao tiếp SPI

Bảng 1-1: Bảng tín hiệu giao tiếp SPI

TT	Tên tín hiệu	Số bit	Chiều	Mô tả
1	MOSI/SOMI	1	Ngõ ra	Ngõ ra dữ liệu khi ở chế độ Master/Slave
2	MISO/SIMO	1	Ngõ vào	Ngõ vào dữ liệu khi ở chế độ Master/Slave
3	SCLK	1	Vào/ra	Đường truyền tín hiệu clock dịch đồng bộ Ngõ ra khi ở chế độ Master Ngõ vào khi ở chế độ Slave

4	SS_0	1	Vào/ra	<p>Tín hiệu chọn slave 0/ chọn SPI khi nó ở chế độ Slave</p> <p>Ngõ ra khi ở chế độ Master, tín hiệu này sẽ chọn slave kết nối với nó khi được tích cực mức thấp</p> <p>Ngõ vào khi ở chế độ Slave và phải được tích cực mức thấp thì SPI mới hoạt động</p>
5	SS_1	1	Ngõ ra	<p>Tín hiệu chọn Slave 1</p> <p>Khi ở chế độ Master, tín hiệu này sẽ chọn slave kết nối với nó khi được tích cực mức thấp</p>
6	SS_2	1	Ngõ ra	<p>Tín hiệu chọn Slave 2</p> <p>Khi ở chế độ Master, tín hiệu này sẽ chọn slave kết nối với nó khi được tích cực mức thấp</p>
7	SS_3	1	Ngõ ra	<p>Tín hiệu chọn Slave 3</p> <p>Khi ở chế độ Master, tín hiệu này sẽ chọn slave kết nối với nó khi được tích cực mức thấp</p>
8	mclk	1	Ngõ vào	Ngõ vào clock thứ 2 ngoài clock pclk từ APB bus, được dùng để tạo clock dịch SCLK
9	SPITXINT	1	Ngõ ra	Tín hiệu báo ngắt truyền từ SPI, tích cực mức cao
10	SPIRXINT	1	Ngõ ra	Tín hiệu báo ngắt nhận từ SPI, tích cực mức thấp
APB interfaces				
11	pclk	1	Ngõ vào	Tín hiệu xung clock từ APB bus
12	preset_n	1	Ngõ vào	Tín hiệu reset từ APB bus, tích cực mức thấp
13	paddr	Xem phần mô tả	Ngõ vào	Tín hiệu địa chỉ từ APB bus

				Độ rộng tín hiệu được quy định trong RTL code thông qua thông số parameter SPI_PADDR_WIDTH
14	pprot	3	Ngõ vào	Không hỗ trợ
15	psel	1	Ngõ vào	Tín hiệu chọn SPI, cho biết đang có một yêu cầu truyền nhận dữ liệu giữa APB bus và SPI Tích cực mức cao
16	penable	1	Ngõ vào	Tín hiệu cho phép, cho biết đang có chu kỳ thứ 2 của một truyền nhận APB Tích cực mức cao
17	pwrite	1	Ngõ vào	Báo hiệu một yêu cầu ghi từ APB bus khi nó được tích cực mức cao Báo hiệu một chu kỳ đọc dữ liệu từ SPI khi nó được tích cực mức thấp
18	pwdata	32	Ngõ vào	Dữ liệu cần ghi vào SPI
19	pstrb	4	Ngõ vào	Tín hiệu cho biết các byte nào trong 4 byte của pwdata được sử dụng để ghi vào SPI Thiết kế SPI này yêu cầu cả 4 bit của pstrb phải được tích cực mức cao thì mới hợp lệ
20	pready	1	Ngõ ra	Tín hiệu báo dữ liệu đã sẵn sàng, dùng để mở rộng một truyền nhận APB Thiết kế SPI này luôn tích cực pready ở mức cao
21	prdata	31	Ngõ ra	Dữ liệu đọc ra từ SPI
22	pslverr	1	Ngõ ra	Báo hiệu có lỗi xảy ra trong một truyền nhận APB. Dữ liệu trả lại trên prdata là 32'b0 Tích cực mức cao

1.2 Tổng quan các thanh ghi và FIFO

Bảng 1-2: Bảng mô tả các thanh ghi và FIFO

Tên	Địa chỉ offset	Kích thước	Mô tả	Ghi chú
SPICR	0x00	32b	SPI Control Register	Đọc/ghi
SPIBR	0x04	32b	SPI Baud Rate Register	Đọc/ghi
SPIINTER	0x08	32b	SPI Interrupt Enable Register	Đọc/ghi
SPISR	0x0C	32b	SPI Status Register	Chỉ đọc
SPIRINTR	0x10	32b	SPI Raw Interrupt Register	Chỉ đọc
SPIINTR	0x14	32b	SPI Interrupt Register	Chỉ đọc

1.3 FIFO truyền và nhận

Bảng 1-3: Bảng mô tả FIFO truyền và nhận

Tên	Địa chỉ offset	Kích thước mỗi ô nhớ	Mô tả
SPITXFIFO	0x18	32b	SPI Transfer FIFO
SPIRXFIFO	0x1C	32b	SPI Receive FIFO

Độ sâu của 2 FIFO sẽ được cấu hình trong RTL code thông qua thông số parameter SPI_FIFO_DEPTH.

Cả 2 FIFO truyền và nhận đều có các bit trạng thái (SPITXST và SPIRXST) cho biết số dữ liệu hiện đang có trong FIFO.

Người dùng có thể cấu hình thời điểm truyền dữ liệu mới từ FIFO sang thanh ghi dịch SPI. Các bit SPITXDL (8 bit) trong thanh ghi SPITXCR quy định khoảng thời gian nghỉ giữa các lần nhận và truyền một ô nhớ dữ liệu mới. Khoảng thời gian nghỉ này được xác định theo số chu kỳ của clock APB bus (PCLK) với giá trị lớn nhất là 255 chu kỳ và nhỏ nhất là 0 chu kỳ. Việc thiết lập khoảng thời gian delay trên chỉ nên được thực hiện khi SPI hoạt động ở chế độ Master, chế độ Slave khoảng thời gian delay này nên bằng 0.

Việc ghi dữ liệu vào FIFO truyền sẽ bắt đầu quá trình truyền nhận dữ liệu, nội dung FIFO truyền đọc ra luôn bằng 0.

SPI không cho phép người dùng ghi dữ liệu vào FIFO nhận, dữ liệu nhận được sau quá trình truyền nhận sẽ tự động lưu vào FIFO nhận.

1.4 Ngắt SPI

SPI chỉ tạo ra 2 tín hiệu ngắt là SPITX và SPIRX gửi đến lõi điều khiển. Mỗi tín hiệu ngắt này được hợp thành từ 4 cờ ngắt của FIFO truyền và nhận, riêng SPIRX có thêm cờ SPITRCRINT để báo việc truyền nhận dữ liệu kết thúc.

1.4.1 Các bit cho phép ngắt của SPI (SPIINTER)

Bảng 1-4-1: Bảng mô tả các bit cho phép ngắt của SPI

TT	Tên	Mô tả
1	SPITXFINTE	Cho phép ngắt từ cờ SPITXFRINT
2	SPIRXFINTE	Cho phép ngắt từ cờ SPIRXFRINT
3	SPITXOINTE	Cho phép ngắt từ cờ SPITXORINT
4	SPIRXOINTE	Cho phép ngắt từ cờ SPIRXORINT
5	SPITXEINTE	Cho phép ngắt từ cờ SPITXERINT
6	SPIRXEINTE	Cho phép ngắt từ cờ SPIRXERINT
7	SPITXUINTE	Cho phép ngắt từ cờ SPITXURINT
8	SPIRXUINTE	Cho phép ngắt từ cờ SPIRXURINT
9	SPITRCINTE	Cho phép ngắt từ cờ SPITRCRINT

1.4.2 Các cờ ngắt raw interrupt SPI (SPIRINTR)

Bảng 1-4-2: Bảng mô tả các cờ ngắt raw interrupt của SPI

TT	Tên	Mô tả
1	SPITXFRINT	Báo dữ liệu đã được ghi đầy vào FIFO truyền (FIFO đầy)
2	SPIRXFRINT	Báo dữ liệu đã được ghi đầy vào FIFO nhận (FIFO đầy)
3	SPITXORINT	Báo FIFO truyền đã đầy nhưng vẫn có yêu cầu ghi dữ liệu mới. Dữ liệu cũ chưa được truyền đi đã bị thay thế bởi dữ liệu mới ghi vào
4	SPIRXORINT	Báo FIFO nhận đã đầy nhưng vẫn có yêu cầu ghi dữ liệu mới. Dữ liệu cũ chưa được đọc đã bị thay thế bởi dữ liệu mới nhận được

5	SPITXERINT	Báo dữ liệu đã được truyền hết từ FIFO truyền (FIFO rỗng)
6	SPIRXERINT	Báo dữ liệu đã được đọc hết từ FIFO nhận (FIFO rỗng)
7	SPITXURINT	Báo FIFO truyền đã rỗng nhưng vẫn có yêu cầu đọc dữ liệu ra. Dữ liệu được truyền đi trên MOSI/SOMI là dữ liệu rác
8	SPIRXURINT	Báo FIFO nhận đã rỗng nhưng vẫn có yêu cầu đọc dữ liệu ra Dữ liệu đọc được trên APB bus là dữ liệu rác
9	SPITRCRINT	Báo việc truyền nhận dữ liệu đã kết thúc

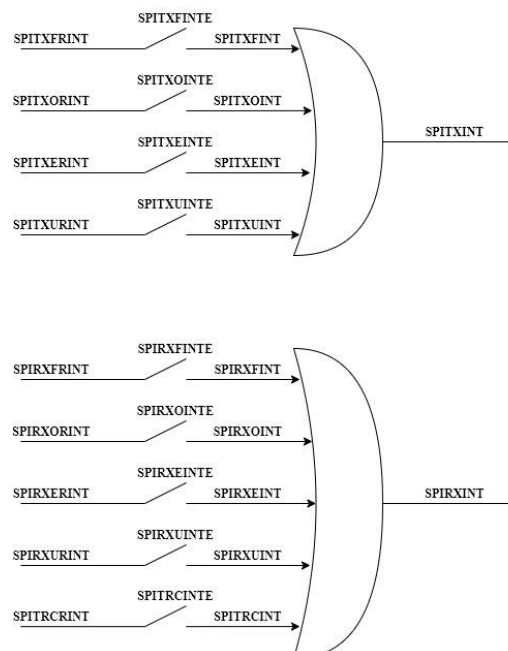
1.4.3 Các cờ ngắt interrupt SPI (SPIINTR)

Các cờ ngắt SPI trong thanh ghi SPIINTR được tạo từ phép AND của cờ raw interrupt trong thanh ghi SPIRINTR và bit cho phép tương ứng của nó trong thanh ghi SPIINTER.

Các cờ ngắt này báo cho người dùng biết cờ ngắt raw interrupt nào đã tạo ra ngắt SPITX và SPIRX gửi đến lõi điều khiển.

1.4.4 Sơ đồ mô tả hoạt động ngắt SPI

Hình 1-4-3: Sơ đồ mô tả hoạt động ngắt SPI



1.5 Hoạt động SPI

1.5.1 Tổng quan hoạt động

SPI là một chuẩn giao tiếp nối tiếp được đồng bộ dựa vào xung clock dịch SCLK. Trong đó, chỉ có Master có thể bắt đầu một chuỗi truyền nhận dữ liệu do nó điều khiển SCLK. Để Master và Slave có thể giao tiếp thành công với nhau thì chúng phải được cài đặt ở cùng định dạng clock dịch, cùng chiều gửi dữ liệu và độ dài dữ liệu.

Thanh ghi SPIBR của SPI chứa các bit quy định tốc độ của xung clock SCLK. Định dạng clock được chọn thông qua 2 bit CPOL và CPHA của thanh ghi SPICR.

Độ dài của một dữ liệu truyền nhận được quy định bởi các bit DATALEN trong thanh ghi SPICR. Chiều truyền dữ liệu là MSB trước hay LSB trước được quy định thông qua bit DORD cũng của thanh ghi này.

1.5.2 Chế độ Master

Trong chế độ Master (MSTR = 1), SPI cung cấp clock dịch cho Slave qua chân SCLK để điều khiển toàn bộ hoạt động của giao tiếp. SPI truyền dữ liệu tới Slave qua chân MOSI(SOMI) và nhận dữ liệu đồng thời ở chân MISO(SIMO). Chân SS_0 lúc này được cấu hình là ngõ ra.

Khi bắt đầu một chuỗi truyền nhận dữ liệu, SPI sẽ kéo tín hiệu slave_select, được đưa tới 1 trong 4 chân SS_0, SS_1, SS_2 và SS_3, xuống mức thấp để chọn Slave mà nó muốn giao tiếp.

Sau khi có một dữ liệu được ghi vào FIFO truyền, SPI sẽ bắt đầu chuỗi truyền dữ liệu bằng cách kéo tín hiệu slave_select xuống mức thấp và tạo xung clock dịch SCLK. Sau khi truyền xong một ô dữ liệu, nếu FIFO vẫn còn dữ liệu cần truyền, sau một khoảng thời gian delay (được quy định trong các bit SPITXDL), SPI sẽ tiếp tục tạo xung SCLK để truyền ô dữ liệu tiếp theo. Sau khi truyền xong ô dữ liệu cuối cùng, cờ SPITXFINT được bật lên, báo FIFO truyền đã hết dữ liệu. Khi hoàn thành việc truyền và nhận dữ liệu cuối cùng, cờ SPITRCRINT sẽ được bật lên. Lúc này, SPI sẽ kết thúc chuỗi truyền nhận bằng cách đưa shift_clock về trạng thái nghỉ và kéo tín hiệu slave_select lên lại mức cao.

Khi bit TALK bị xóa đi về 0, SPI vẫn nhận dữ liệu tuy nhiên việc truyền bị cấm (ngõ ra MOSI(SOMI) và ngõ ra slave_slect sẽ được đặt ở trạng thái trở kháng cao). Ở chế độ Master, bit TALK hỗ trợ nhiều Master cùng kết nối vào giao thức SPI có thể nhận dữ liệu cùng lúc.

1.5.3 Chế độ Slave

Trong chế độ Slave ($MSTR = 0$), SPI nhận clock dịch từ Master của nó qua chân SCLK. SPI truyền dữ liệu tới Master qua chân SOMI(MOSI) và nhận dữ liệu đồng thời ở chân SIMO(MISO). Chân SS₀ lúc này được cấu hình là ngõ vào.

Chỉ khi chân SS₀ của SPI được kéo xuống mức thấp thì clock dịch mới được đưa vào SPI và dữ liệu trên SOMI(MOSI) và SIMO(MISO) mới được lái. Khi SS₀ ở mức cao, SOMI và SIMO sẽ ở trạng thái trở kháng cao, SCLK ở trạng thái IDLE của nó.

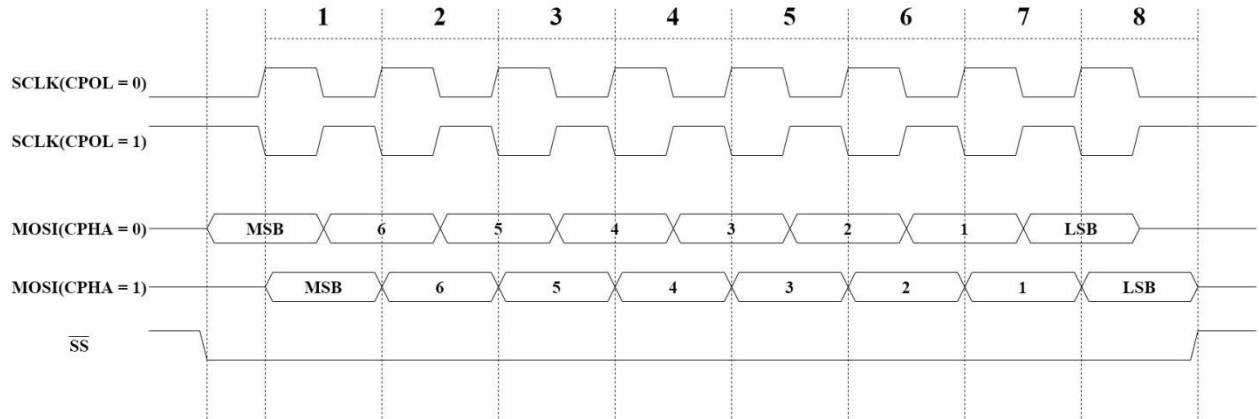
Khi bit TALK bị xóa đi về 0, SPI vẫn nhận dữ liệu tuy nhiên việc truyền bị cấm (ngõ ra SOMI(MISO) sẽ được đặt ở trạng thái trở kháng cao). Bit TALK hỗ trợ nhiều Slave cùng kết nối vào giao thức SPI có thể nhận dữ liệu cùng lúc.

1.5.3 Định dạng clock

Có 4 định dạng clock được sử dụng trong giao thức SPI. Định dạng này sẽ được chọn thông qua 2 bit CPOL và CPHA trong thanh ghi SPICR

- CPOL: quy định mức logic của xung SCLK khi không hoạt động (IDLE)
 - CPOL = 0: SCLK ở mức 0 khi không hoạt động, khi bắt đầu xung truyền (thoát IDLE) thì cạnh dẫn sẽ là cạnh lên và cạnh theo sau là cạnh xuống
 - CPOL = 1: SCLK ở mức 1 khi không hoạt động, khi bắt đầu xung truyền (thoát IDLE) thì cạnh dẫn sẽ là cạnh xuống và cạnh theo sau là cạnh lên
- CPHA: quy định thời điểm lấy mẫu và thiết lập dữ liệu theo clock dịch
 - CPHA = 0: bit dữ liệu cần truyền sẽ được đưa ra chân MOSI/SOMI ở cạnh theo sau, bit dữ liệu nhận vào MISO/SIMO sẽ được chốt ở cạnh dẫn. Ở chu kỳ xung dịch đầu tiên, bit dữ liệu đầu tiên phải được đưa lên đường truyền trước khi cạnh dẫn xuất hiện.
 - CPHA = 1: bit dữ liệu cần truyền sẽ được đưa ra chân MOSI/SOMI ở cạnh dẫn, bit dữ liệu nhận vào MISO/SIMO sẽ được chốt ở cạnh theo sau.

Hình 1-5-3: Giản đồ các định dạng clock (SPI Master, 8 bit, MSB truyền trước)



1.5.4 Tốc độ baud và clock dịch

Tùy vào chế độ Master hay Slave mà SPI có thể nhận nguồn clock dịch từ bên ngoài hoặc truyền clock dịch ra ngoài.

Tốc độ baud của SPI có thể được cài đặt nhờ 8 bit SPIBR của thanh ghi SPIBR, tốc độ baud được cài đặt chính là tần số xung clock dịch của Master.

- Chế độ Master: SPI truyền clock dịch ra chân SLK, tốc độ clock dịch này không được vượt quá $\frac{F_{pre_clock}}{2}$???
- Chế độ Slave: SPI nhận clock dịch bên ngoài từ chân SLK, tốc độ clock dịch này không được vượt quá $\frac{F_{pre_clock}}{2}$???

Công thức bên dưới minh họa cách tính tốc độ baud SPI:

$$\text{SPI baud rate} = \frac{F_{pre_clock}}{2(1 + SPIBR)}$$

Trong đó:

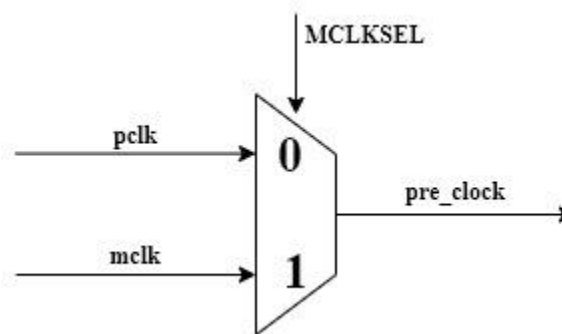
- F_{pre_clock} là tần số của clock nguồn được chọn để tạo clock dịch
- SPIBR là nội dung thanh ghi SPIBR của Master

(1)

Từ công thức trên, t có thể thấy tốc độ tối đa mà SPI có thể tạo ra là $\frac{F_{pre_clock}}{2}$ (SPIBR = 0) và tốc độ tối thiểu là $\frac{F_{pre_clock}}{512}$ (SPIBR = 255)

Hình minh họa clock nguồn của SPI Master:

Hình 1-5-4: Clock nguồn SPI Master



Trong đó:

- pclk: là clock của APB bus
- mclk: là clock được lấy từ các nguồn khác pclk (osc, PLL, ...)
- MCLKSEL: bit chọn nguồn clock tạo xung SCLK trong thanh ghi SPICR
- pre_clock: clock dùng để tạo xung SCLK

2 CÁC THANH GHI CẤU HÌNH VÀ TRẠNG THÁI SPI

SPI chứa các thanh ghi điều khiển và trạng thái sau:

Cấu hình:

- [SPICR: SPI Control Register \(0x00\)](#)
- [SPIBR: SPI Baud Rate Register \(0x04\)](#)
- [SPIINTE: SPI Interrupt Enable Register \(0x08\)](#)

Trạng thái:

- [SPISR: SPI Status Register \(0x0C\)](#)
- [SPIRINTR: SPI Raw Interrupt Register \(0x10\)](#)
- [SPIINTR: SPI Interrupt Register \(0x14\)](#)

2.1 SPI Control Register (SPICR)

Hình 2-1: Thanh ghi SPICR

31:24	SPIE	SWR	DORD	MSTR	CPOL	CPHA	MCLKSEL	TALK
23:16	SPITXDL7	SPITXDL6	SPITXDL5	SPITXDL4	SPITXDL3	SPITXDL2	SPITXDL1	SPITXDL0
15:8	SS1	SS0					SPITXRST	SPIRXRST
7:0				DATALEN4	DATALEN3	DATALEN2	DATALEN1	DATALEN0

Bảng 2-1: Bảng mô tả thanh ghi SPICR

Bit	Tên	Giá trị mặc định	Mô tả
-----	-----	------------------	-------

31	SPIE	0	<p>SPI enable:</p> <p>0: SPI không được phép hoạt động, tất cả các thanh ghi sẽ được reset về giá trị mặc định</p> <p>1: SPI được phép hoạt động</p>
30	SWR	0	<p>Software reset: Bit này tích cực mức 0</p> <p>0: SPI xóa tất cả các cờ ngắt về trạng thái mặc định, con trỏ đọc và ghi của cả FIFO truyền và nhận đều bị xóa về 0. SPI sẽ hoàn thành việc truyền nhận gói dữ liệu hiện tại. Sau đó, clock dịch SCLK bị buộc phải về trạng thái IDLE. Tất cả các bit cấu hình hoạt động và cho phép ngắt của SPI vẫn được giữ nguyên.</p> <p>1: SPI sẵn sàng truyền nhận dữ liệu</p>
29	DORD	0	<p>Data order:</p> <p>0: MSB truyền trước</p> <p>1: LSB truyền trước</p>
28	MSTR	0	<p>Master mode:</p> <p>0: SPI hoạt động ở chế độ Slave</p> <p>1: SPI hoạt động ở chế độ Master</p>
27	CPOL	0	<p>Clock polarity:</p> <p>0: Quy định mức logic của xung SCLK khi không hoạt động (IDLE) là 0</p> <p>1: Quy định mức logic của xung SCLK khi không hoạt động (IDLE) là 1</p>
26	CPHA	0	<p>Clock phase: quy định thời điểm lấy mẫu và thiết lập dữ liệu theo clock dịch</p>

			<p>0: Lấy mẫu dữ liệu ở cạnh clock đầu tiên và đưa dữ liệu lên đường truyền cạnh clock theo sau. Bit đầu tiên được gửi đi phải được đưa lên đường truyền $\frac{1}{2}$ chu kỳ trước cạnh clock đầu tiên</p> <p>1: Đưa dữ liệu lên đường truyền ở cạnh clock đầu tiên và lấy mẫu dữ liệu ở cạnh clock tiếp theo</p>
25	MCLKSEL	0	<p>Master clock select:</p> <p>0: Clock dùng để tạo xung SCLK được lấy từ APB bus (pclk)</p> <p>1: Clock dùng để tạo xung SCLK được lấy từ các nguồn clock khác (osc, PLL, ...)</p>
24	TALK	0	<p>Slave transmit enable: tích cực mức 0</p> <p>0: SPI được phép truyền và nhận dữ liệu</p> <p>1: SPI vẫn được phép nhận dữ liệu. Tuy nhiên, việc truyền dữ liệu sẽ bị cấm. Tùy vào chế độ hoạt động mà việc cấm truyền sẽ được cài đặt khác nhau</p> <p>+ Chế độ Master: chân MOSI(SOMI) và tín hiệu slave_select gửi đến một trong các chân SS sẽ được đặt ở trạng thái trở kháng cao.</p> <p>+ Chế độ Slave: chân SOMI(MOSI) được đặt ở trạng thái trở kháng cao</p>
23:16	SPITXDLx	8'b0	<p>SPI transfer delay:</p> <p>Quy định thời gian delay giữa các lần kết thúc việc gửi một gói dữ liệu và bắt đầu truyền một gói dữ liệu mới. Thời gian delay được tính theo chu kỳ clock pclk của APB bus.</p> <p>Tối thiểu: 0 chu kỳ pclk</p> <p>Tối đa: 255 chu kỳ pclk</p>

15:14	SSx	2'b0	Slave select: Chọn Slave cần giao tiếp khi SPI hoạt động ở chế độ Master: 2'b00: SS_0 2'b01: SS_1 2'b10: SS_2 2'b11: SS_3
13:10	Reversed	4'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
9	SPITXRST	1	SPI transfer reset: tích cực mức 0 0: Xóa con trỏ đọc và ghi của FIFO truyền về 0 1: Cho phép FIFO truyền hoạt động
8	SPIRXRST	1	SPI receive reset: tích cực mức 0 0: Xóa con trỏ đọc và ghi của FIFO nhận về 0 1: Cho phép FIFO nhận hoạt động
7:5	Reversed	3'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
4:0	DATALENx	5'b00111	Data length: Quy định độ dài của một gói dữ liệu: $Data\ length = DATALEN + 1$

2.2 SPI Baud Rate Register (SPIBR)

Hình 2-2: Thanh ghi SPIBR

31:24								
23:16								
15:8								
7:0	SPBR7	SPBR6	SPBR5	SPBR4	SPBR3	SPBR2	SPBR1	SPBR0

Bảng 2-2: Bảng mô tả thanh ghi SPIBR

Bit	Tên	Giá trị mặc định	Mô tả
31:8	Reversed	24'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
7:0	SPIBRx	8'b0	<p>SPI baud rate divisor:</p> <p>Quy định tốc độ baud của SPI, được tính bằng công thức:</p> $\text{SPI baud rate} = \frac{F_{pre_clock}}{2(1 + SPIBR)}$ <p>Tham khảo: (1)</p>

2.3 SPI Interrupt Enable Register (SPIINTER)

Hình 2-3-1: Thanh ghi SPIINTER

31:24	SPITRCINTE						
23:16							
15:8				SPITXFINT	SPITXOINTE	SPITXEINTE	SPITXUINTE
7:0				SPIRXFINT	SPIRXOINTE	SPIRXEINTE	SPIRXUINTE

Bảng 2-3-2: Bảng mô tả thanh ghi SPIINTER

Bit	Tên	Giá trị mặc định	Mô tả
-----	-----	------------------	-------

31	SPITRCINTE	1	SPI transfer and receive complete flag enable: 0: Cấm ngắt từ cờ SPITRCRINT 1: Cho phép ngắt từ cờ SPITRCRINT
30:12	Reversed	19'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
11	SPITXFINTE	0	SPI transfer full flag interrupt enable 0: Cấm ngắt từ cờ SPITXFRINT 1: Cho phép ngắt từ cờ SPITXFRINT
10	SPITXOINTE	0	SPI transfer overflow flag interrupt enable 0: Cấm ngắt từ cờ SPITXORINT 1: Cho phép ngắt từ cờ SPITXORINT
9	SPITXEINTE	0	SPI transfer empty flag interrupt enable 0: Cấm ngắt từ cờ SPITXERINT 1: Cho phép ngắt từ cờ SPITXERINT
8	SPITXUINTE	0	SPI transfer underflow flag interrupt enable 0: Cấm ngắt từ cờ SPITXURINT 1: Cho phép ngắt từ cờ SPITXURINT
7:4	Reversed	4'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
3	SPIRXFINTE	0	SPI receive full flag interrupt enable 0: Cấm ngắt từ cờ SPIRXFRINT 1: Cho phép ngắt từ cờ SPIRXFRINT
2	SPIRXOINTE	0	SPI receive overflow flag interrupt enable 0: Cấm ngắt từ cờ SPIRXORINT 1: Cho phép ngắt từ cờ SPIRXORINT
1	SPIRXEINTE	0	SPI receive empty flag interrupt enable

			0: Cấm ngắt từ cờ SPIRXERINT 1: Cho phép ngắt từ cờ SPIRXERINT
0	SPIRXUINTE	0	SPI receive underflow flag interrupt enable 0: Cấm ngắt từ cờ SPIRXURINT 1: Cho phép ngắt từ cờ SPIRXURINT

2.4 SPI Status Register (SPISR)

Hình 2-4-1: Thanh ghi SPISR

31:24							
23:16							
15:8		SPITXST5	SPITXST4	SPITXST3	SPITXST2	SPITXST1	SPITXST0
7:0		SPIRXST5	SPIRXST4	SPIRXST3	SPIRXST2	SPIRXST1	SPIRXST0

Bảng 2-4-2: Bảng mô tả thanh ghi SPISR

Bit	Tên	Giá trị mặc định	Mô tả
31:14	Reversed	18'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
13:8	SPITXST _x	6'b0	SPI transfer status Trạng thái của FIFO truyền 6'000000: FIFO truyền rỗng 6'b000001: FIFO truyền có 1 gói dữ liệu ...
7:6	Reversed	2'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
5:0	SPIRXST _x	6'b0	SPI receive status Trạng thái của FIFO nhận 6'000000: FIFO nhận rỗng 6'b000001: FIFO nhận có 1 gói dữ liệu

			...
--	--	--	-----

2.5 SPI Raw Interrupt Register (SPIRINTR)

Bảng 2-5-1: Thanh ghi SPIRINTR

31:24	SPITRCRINT						
23:16							
15:8				SPITXFRINT	SPITXORINT	SPITXERINT	SPITXURINT
7:0				SPIRXFRINT	SPIRXORINT	SPIRXERINT	SPIRXURINT

Bảng 2-5-2: Bảng mô tả thanh ghi SPIRINTR

Bit	Tên	Giá trị mặc định	Mô tả
31	SPITRCRINT	0	SPI transfer and receive complete flag raw interrupt 0: Việc truyền nhận dữ liệu chưa kết thúc 1: Việc truyền nhận dữ liệu đã kết thúc
30:12	Reversed	19'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
11	SPITXFRINT	0	SPI transfer full flag raw interrupt 0: FIFO truyền chưa đầy 1: FIFO truyền đầy
10	SPITXORINT	0	SPI transfer overflow flag raw interrupt 0: FIFO truyền không bị overflow 1: FIFO truyền bị overflow
9	SPITXERINT	0	SPI transfer empty flag raw interrupt

			0: FIFO truyền vẫn còn dữ liệu 1: FIFO truyền rỗng
8	SPITXURINT	0	SPI transfer underflow flag raw interrupt 0: FIFO truyền không bị underflow 1: FIFO truyền bị underflow
7:4	Reversed	4'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
3	SPIRXFRINT	0	SPI receive full flag raw interrupt 0: FIFO nhận chưa đầy 1: FIFO nhận đầy
2	SPIRXORINT	0	SPI receive overflow flag raw interrupt 0: FIFO nhận không bị overflow 1: FIFO nhận bị overflow
1	SPIRXERINT	0	SPI receive empty flag raw interrupt 0: FIFO nhận vẫn còn dữ liệu 1: FIFO nhận rỗng
0	SPIRXURINT	0	SPI receive underflow flag raw interrupt 0: FIFO nhận không bị underflow 1: FIFO nhận bị underflow

2.6 SPI Interrupt Register (SPIINTR)

Hình 2-6-1: Thanh ghi SPIINTR

31:24	SPITRCINT						
23:16							
15:8				SPITXFINT	SPITXOINT	SPITXEINT	SPITXUINT
7:0				SPIRXFINT	SPIRXOINT	SPIRXEINT	SPIRXUINT

Bảng 2-6-2: Bảng mô tả thanh ghi SPIINTR

Bit	Tên	Giá trị mặc định	Mô tả
31	SPITRCINT	0	SPI transfer and receive complete full flag interrupt 0: Cờ SPITRCINT không tạo ngắt SPITX 1: Cờ SPITRCINT tạo ngắt SPITX
30:12	Reversed	19'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
11	SPITXFINT	0	SPI transfer full flag interrupt 0: Cờ SPITXFRINT không tạo ngắt SPITX 1: Cờ SPITXFRINT tạo ngắt SPITX
10	SPITXOINT	0	SPI transfer overflow interrupt 0: Cờ SPITXORINT không tạo ngắt SPITX 1: Cờ SPITXORINT tạo ngắt SPITX
9	SPITXEINT	0	SPI transfer empty interrupt 0: Cờ SPITXERINT không tạo ngắt SPITX 1: Cờ SPITXERINT tạo ngắt SPITX
8	SPITXUINT	0	SPI transfer underflow interrupt

			0: Cờ SPITXURINT không tạo ngắt SPITX 1: Cờ SPITXURINT tạo ngắt SPITX
7:4	Reversed	4'b0	Các bit không sử dụng, không thể ghi vào và luôn đọc ra giá trị 0
3	SPIRXFINT	0	SPI receive full flag interrupt 0: Cờ SPIRXFRINT không tạo ngắt SPIRX 1: Cờ SPIRXFRINT tạo ngắt SPIRX
2	SPIRXOINT	0	SPI receive overflow interrupt 0: Cờ SPIRXORINT không tạo ngắt SPIRX 1: Cờ SPIRXORINT tạo ngắt SPIRX
1	SPIRXEINT	0	SPI receive empty interrupt 0: Cờ SPIRXERINT không tạo ngắt SPIRX 1: Cờ SPIRXERINT tạo ngắt SPIRX
0	SPIRXUINT	0	SPI receive underflow interrupt 0: Cờ SPIRXURINT không tạo ngắt SPIRX 1: Cờ SPIRXURINT tạo ngắt SPIRX

3 HƯỚNG DẪN SỬ DỤNG SPI

Sau khi bị cấm hoạt động ($SPIE = 0$), SPI sẽ được đưa về trạng thái cấu hình mặc định sau:

- Cấu hình Slave ($MSTR = 0$)
- SPI không được phép tạo hay nhận clock dịch SCLK, các cờ ngắt đều bị xóa tích cực, FIFO truyền và nhận đều bị xóa dữ liệu ($SWR = 0$)
- Cho phép truyền dữ liệu ($TALK = 0$)
- Clock nguồn được chọn để tạo xung SCLK nếu đổi sang chế độ Master là pclk ($MCLKSEL = 0$)
- MSB truyền trước ($DORD = 0$)
- Độ dài dữ liệu là 1 bit ($DATALEN = 5'b0$)
- Các cờ ngắt SPI đều bị cấm trừ cờ SPITRCRINT
- Thời gian delay giữa các lần kết thúc việc gửi một gói dữ liệu và bắt đầu truyền một gói dữ liệu mới là 0 chu kỳ pclk ($SPITXDL = 8'b0$).

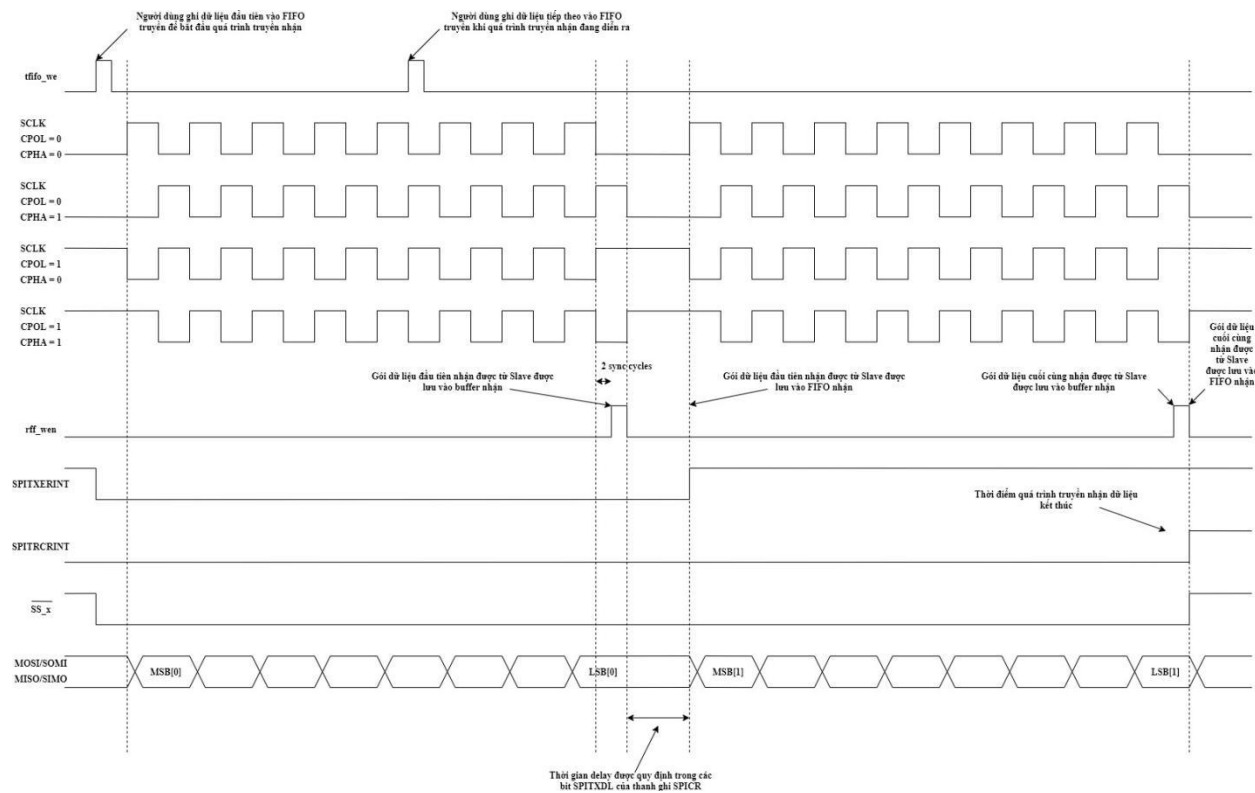
Để thay đổi cấu hình và sử dụng SPI, người dùng phải thực hiện các bước sau:



- Bước 1: Bật bit SPIE (SPICR.31) lên 1 để cho phép SPI hoạt động (nếu SPIE được bật sẵn thì có thể bỏ qua bước này).
- Bước 2: Xóa bit SWR (SPICR.30) về 0 để buộc SPI dừng việc truyền nhận dữ liệu (clock dịch SCLK bị buộc đưa về trạng thái IDLE). Nếu có gói dữ liệu đang truyền khi bit này bị xóa về 0, thì tiếp tục hoàn thành việc truyền nhận gói dữ liệu đó rồi mới đưa clock dịch SCLK về trạng thái IDLE.
- Bước 3: Bắt đầu cấu hình hoạt động, tốc độ baud, và các bit cho phép ngắt lần lượt trong các thanh ghi SPICR, SPIBR và SPIINTER.
- Bước 4: Bật bit SWR (SPICR.30) lên 1 để cho phép SPI tạo hoặc nhận xung dịch SCLK.
- Bước 5: Ghi dữ liệu vào FIFO truyền. Ở chế độ Master, việc ghi dữ liệu vào FIFO truyền sẽ bắt đầu quá trình truyền nhận.
- Bước 6: Đọc dữ liệu từ FIFO nhận sau khi quá trình truyền nhận dữ liệu kết thúc ($\text{SPITRCRINT} = 1$ (SPIRINTR.31)).

Khi SPI hoạt động ở chế độ Slave, gói dữ liệu đầu tiên cần truyền phải được đưa vào FIFO truyền trước khi clock dịch từ Master được đưa tới.

Hình 3-1: Hình minh họa SPI hoạt động ở chế độ Master (DATALEN = 7, DORD = 0, SPI truyền 2 gói dữ liệu liên tiếp)



Hình 3-2: Hình minh họa SPI hoạt động ở chế độ Slave (DATALEN = 7, DORD = 0, Master truyền 2 gói dữ liệu liên tiếp)

