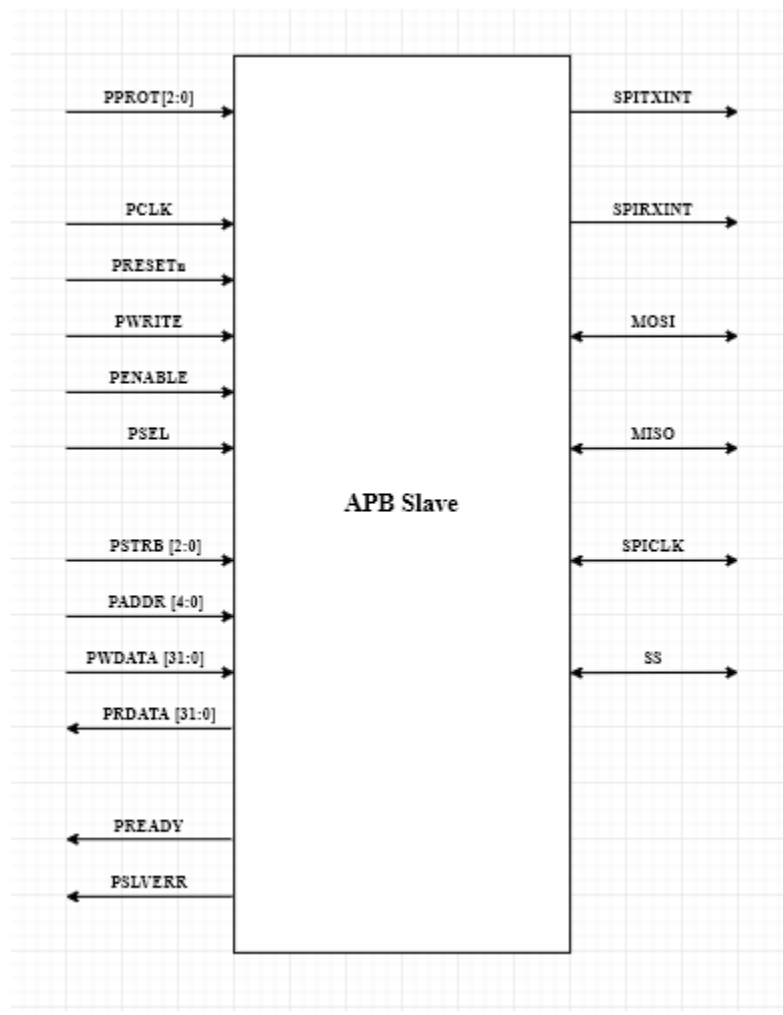


VLSI Technology

4.4.2020

SPI Detail Description

I) Sơ đồ tín hiệu giao tiếp:



Bảng tín hiệu giao tiếp SPI

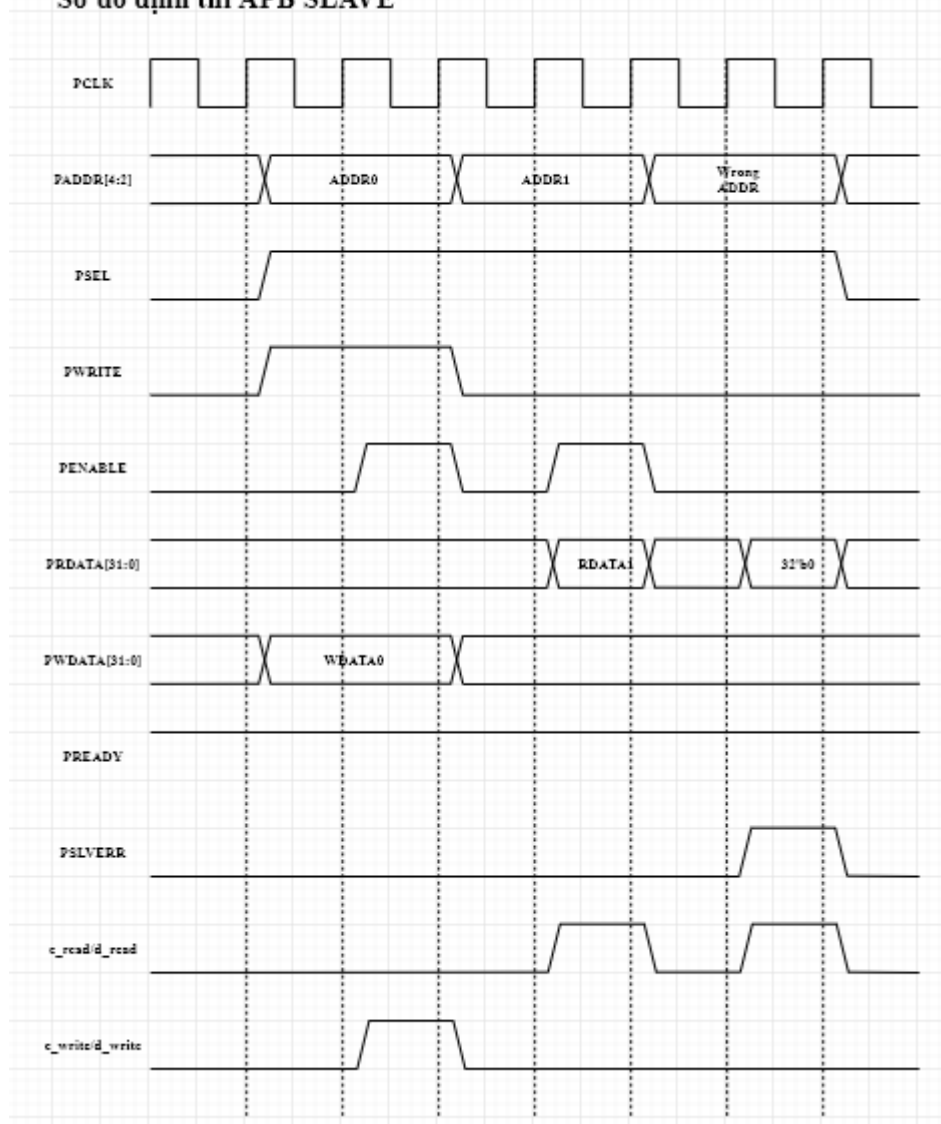
TT	Tín hiệu	Số bit	Chiều	Mô tả
1.	PCLK	1	Ngõ vào	Tín hiệu xung clock từ giao thức APB
2.	PRESETn	1	Ngõ vào	Tín hiệu reset từ giao thức APB, tích cực thấp
3.	PWRITE	1	Ngõ vào	Tín hiệu giúp phân biệt truy cập:

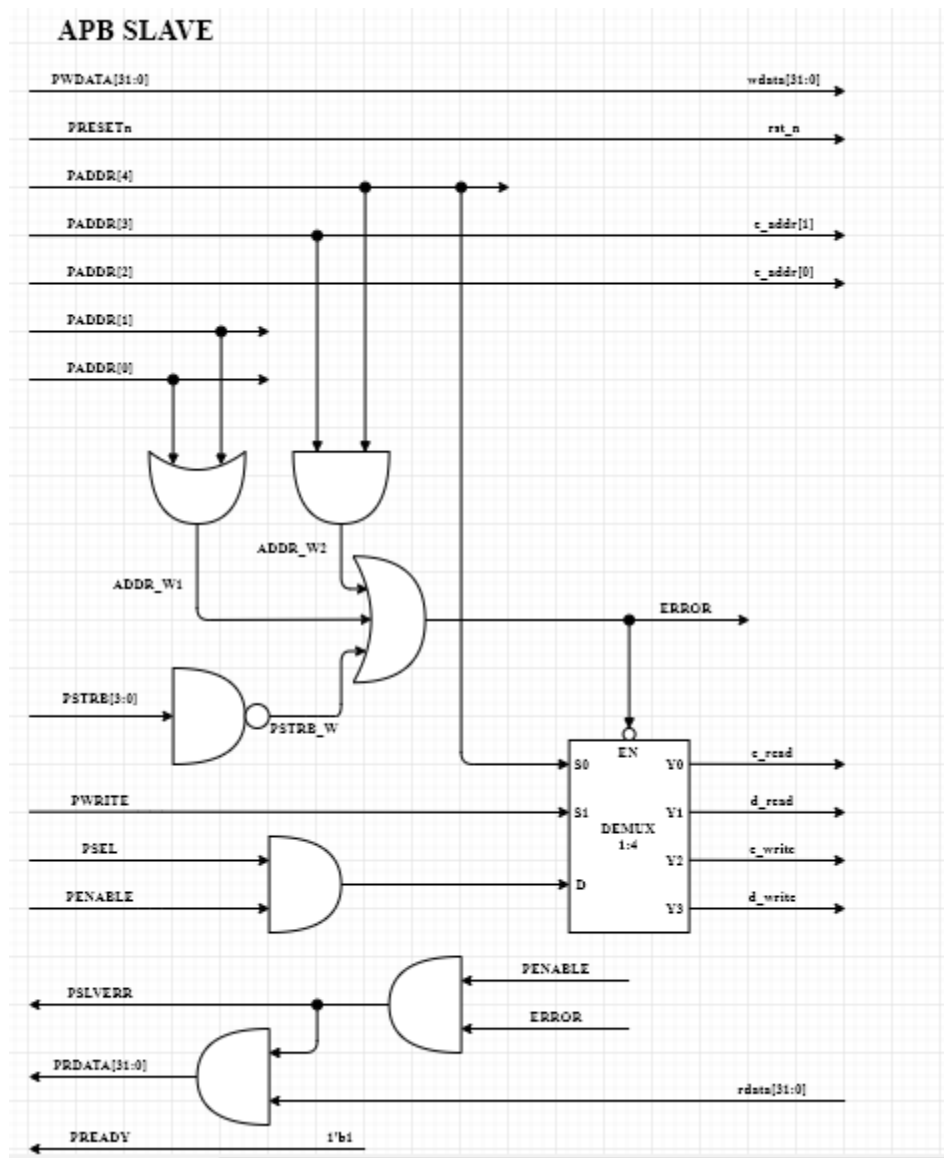
				PWRITE = 0: đọc PWRITE = 1: ghi
4.	PSEL	1	Ngõ vào	Tín hiệu từ giao thức APB chọn ngoại vi SPI
5.	PSTRB	4	Ngõ vào	Tín hiệu báo vị trí hợp lệ của các byte trên dữ liệu ghi, Yêu cầu PSTRB luôn bằng 4'b1111
6.	PPROT	3	Ngõ vào	Không hỗ trợ
7.	PENABLE	1	Ngõ vào	Tín hiệu báo pha ENABLE của một giao thức đọc ghi
8.	PADDR	5	Ngõ vào	Tín hiệu chứa địa chỉ các thanh ghi SPI từ giao thức APB
9.	PWDATA	32	Ngõ vào	Tín hiệu chứa dữ liệu cần truyền
10.	PRDATA	32	Ngõ ra	Tín hiệu chứa dữ liệu đọc được
11.	PREADY	1	Ngõ ra	Tín hiệu báo SPI sẵn sàng nhận hoặc truyền dữ liệu ở cạnh lên PCLK tiếp theo
12.	PSLVERR	1	Ngõ ra	Tín hiệu báo có lỗi đọc ghi trong một APB transfer
13.	MOSI	1	Ngõ ra hoặc vào tùy thuộc vào	Master out Slave in

			cài đặt Master/Slave	
14.	MISO	1	Ngõ vào hoặc ra tùy thuộc vào cài đặt Master/Slave	Master in Slave out
15.	SPICLK	1	Ngõ ra hoặc vào tùy thuộc vào cài đặt Master/Slave	Clock của giao tiếp SPI
16.	\overline{SS}	1	Ngõ vào	Tín hiệu chọn Slave.
17.	SPITXINT	1	Ngõ ra	Tín hiệu ngắt SPI transfer
18.	SPIRXINT	1	Ngõ ra	Tín hiệu ngắt SPI Receive

II) Sơ đồ khối tổng quát:

Sơ đồ định thời APB SLAVE



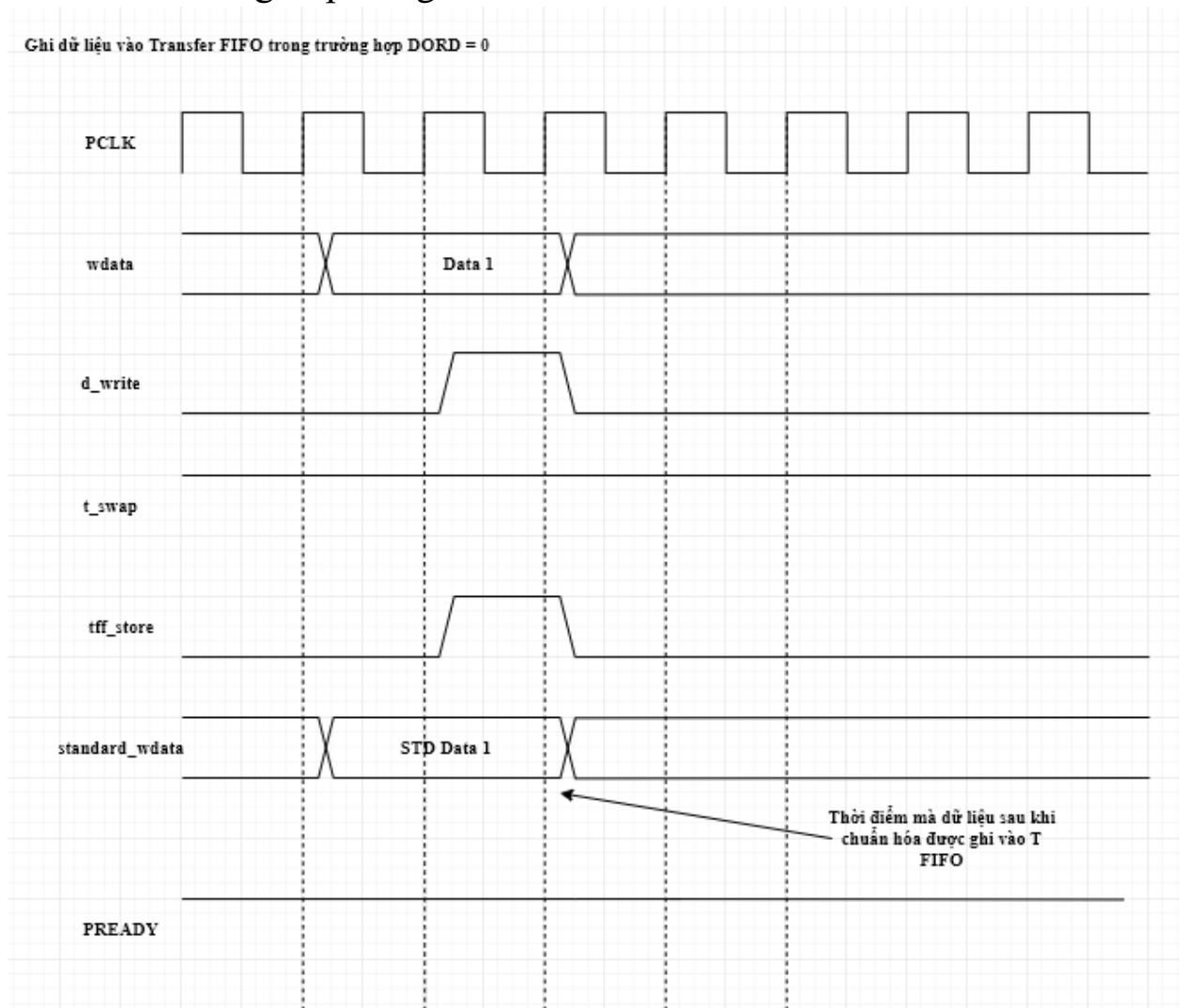


- APB Slave khi phát hiện có lỗi (Lỗi địa chỉ hoặc lỗi PSTRB) sẽ trả lời bằng một chu kỳ tích cực PSLVERR và dữ liệu trên PRDATA[31:0] = 32'b0
- Các tín hiệu c_read, d_read, d_write, c_write, chỉ được tích cực khi không có lỗi xảy ra. Mỗi chu kỳ đọc ghi của APB bus chỉ tích cực các tín hiệu này trong một chu kỳ PCLK
- PREADY luôn được đặt bằng 1

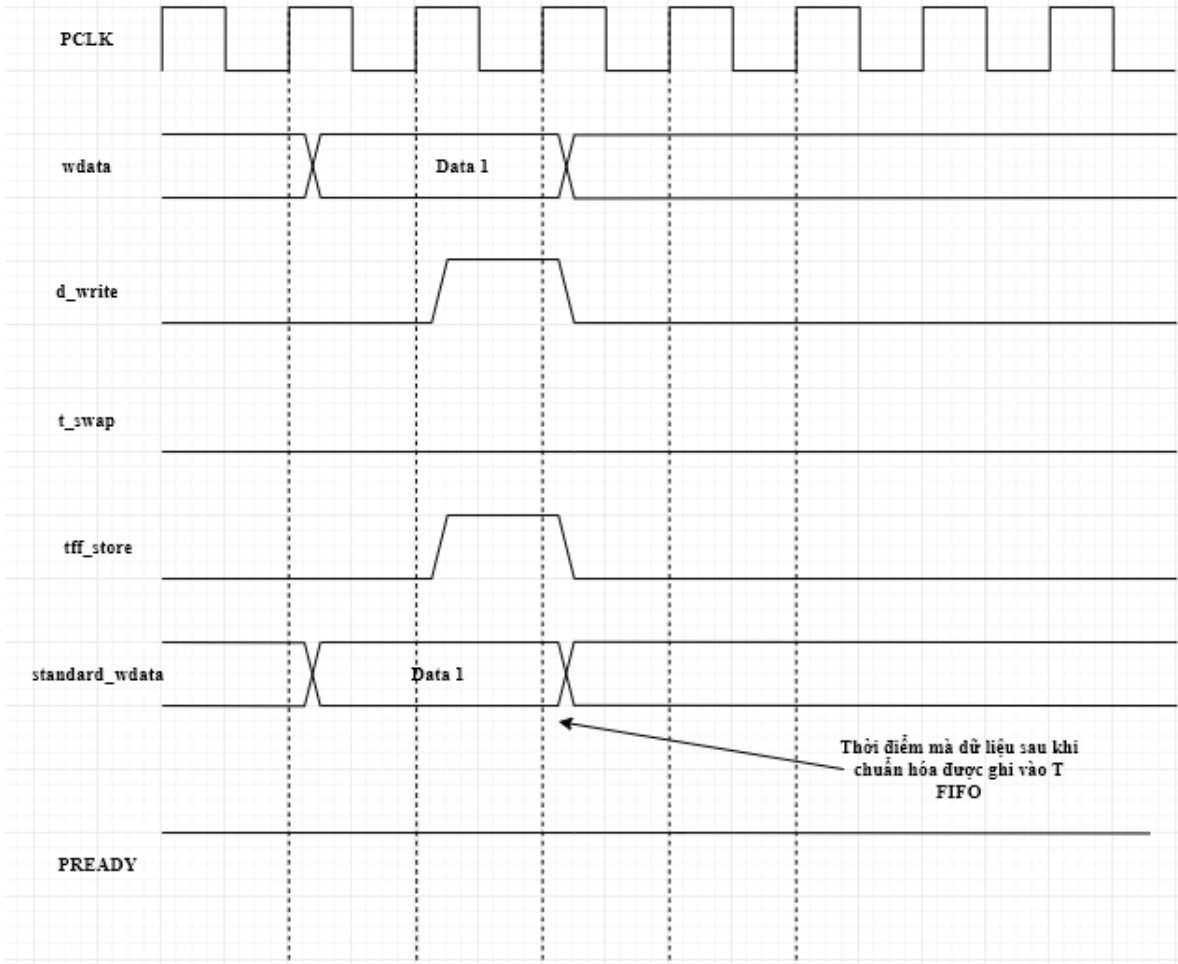
2) **SPI Data:**

- Sơ đồ định thì các trường hợp hoạt động của SPI Data:

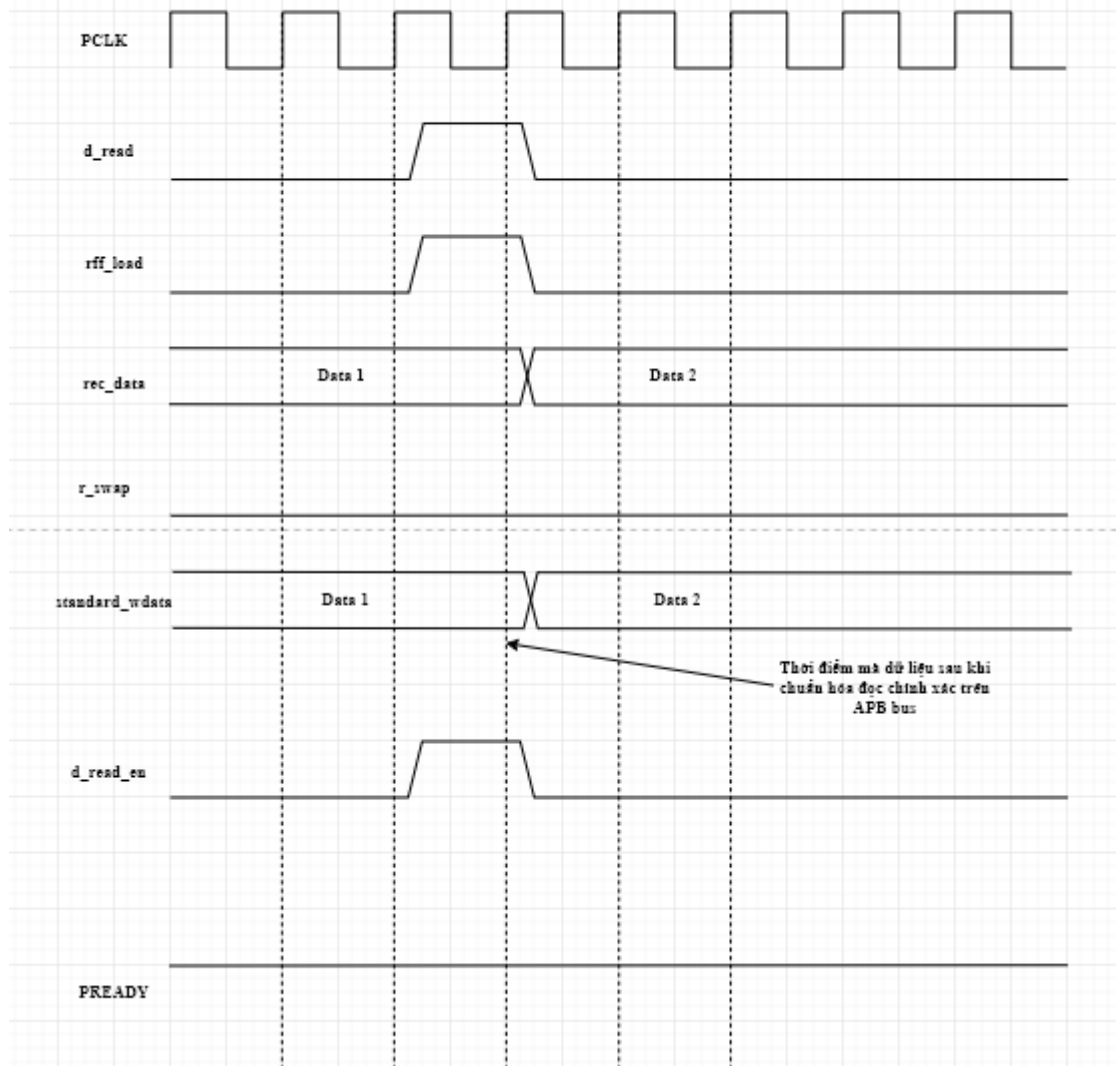
Note: Các tín hiệu điều khiển của SPI khi chỉ sử dụng buffer cũng tương tự như khi dùng FIFO, nên các hình minh họa bên dưới chỉ minh họa trường hợp dùng FIFO

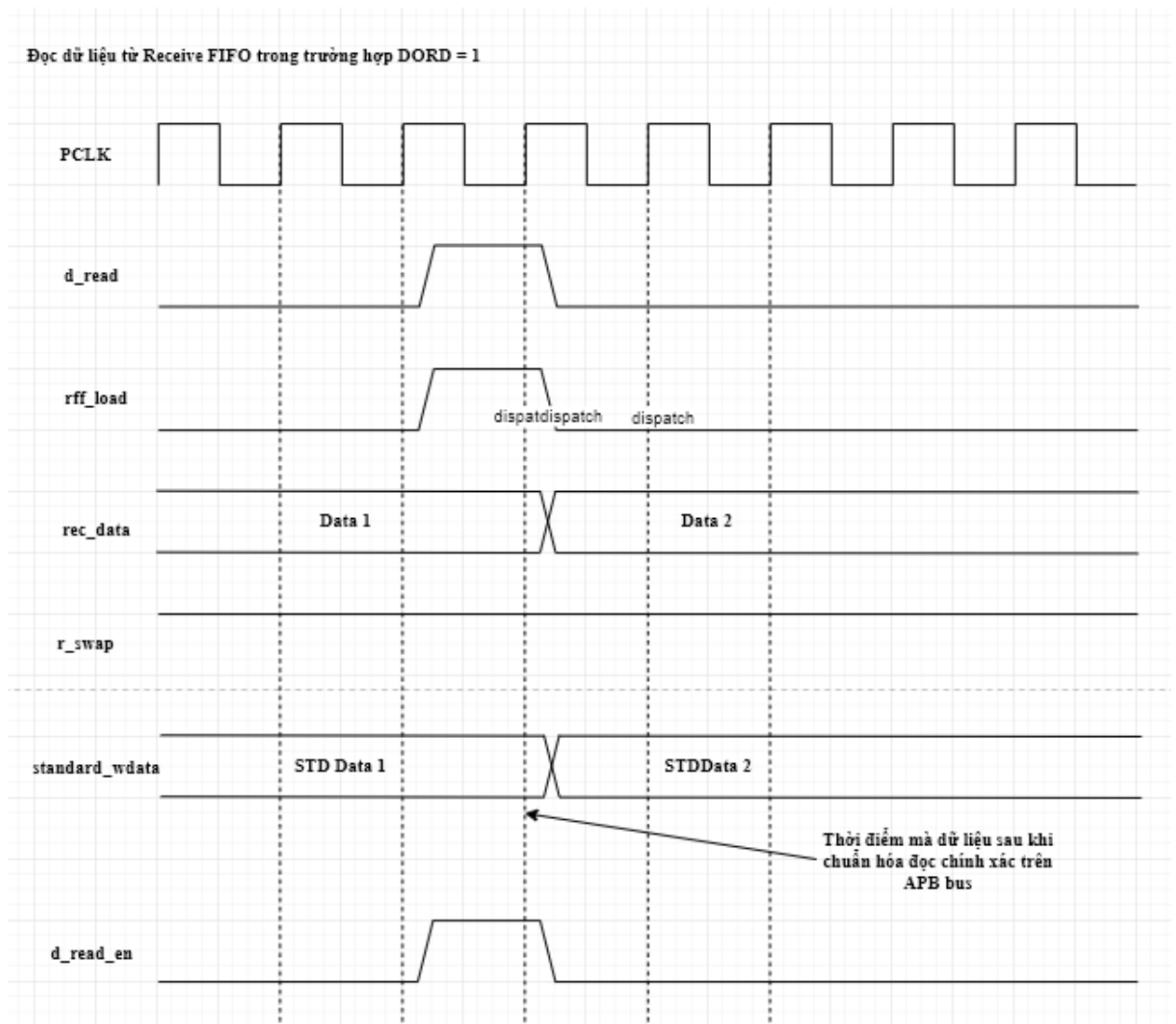


Ghi dữ liệu vào Transfer FIFO trong trường hợp DORD = 1



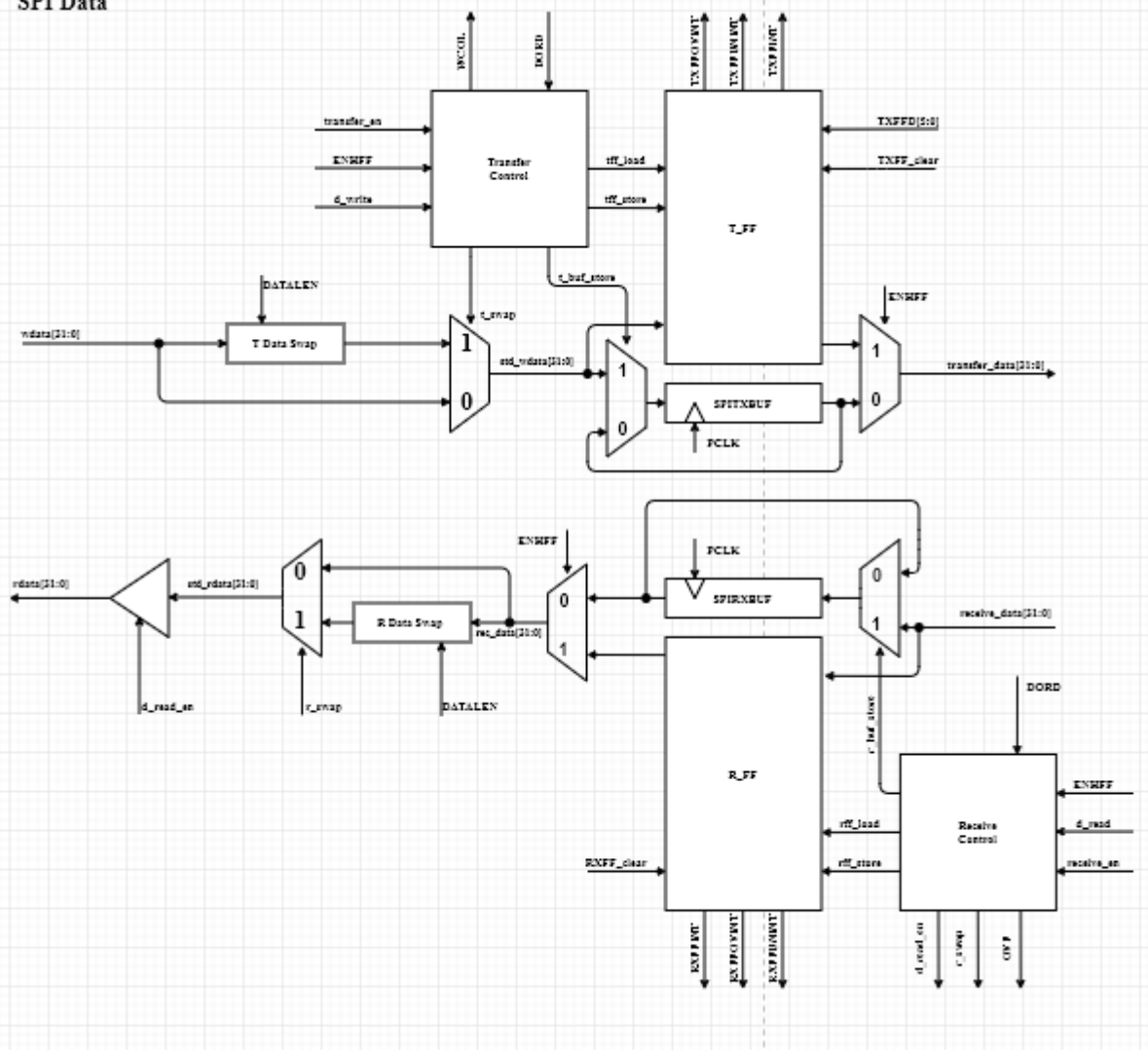
Đọc dữ liệu từ Receive FIFO trong trường hợp DORD = 0



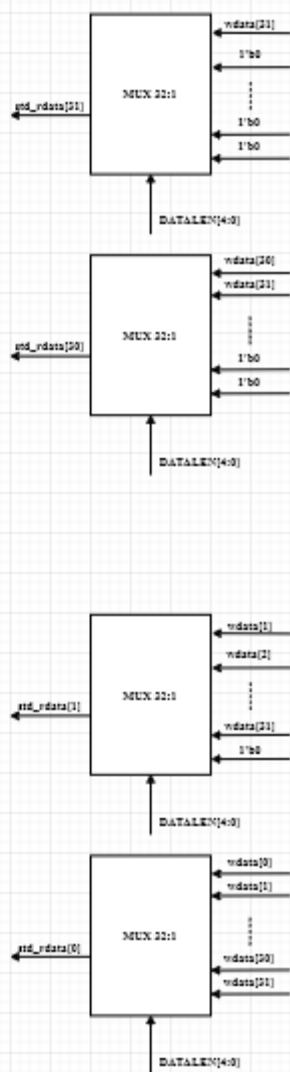


- Sơ đồ khối chi tiết các thành phần của SPI Data:

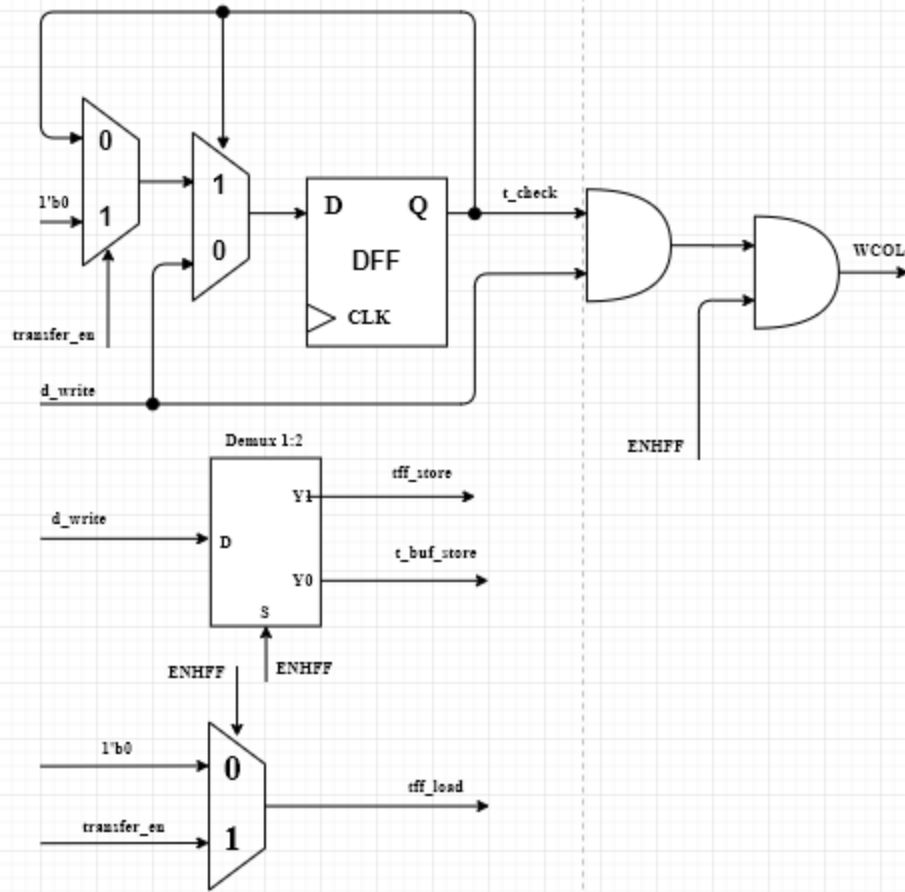
SPI Data

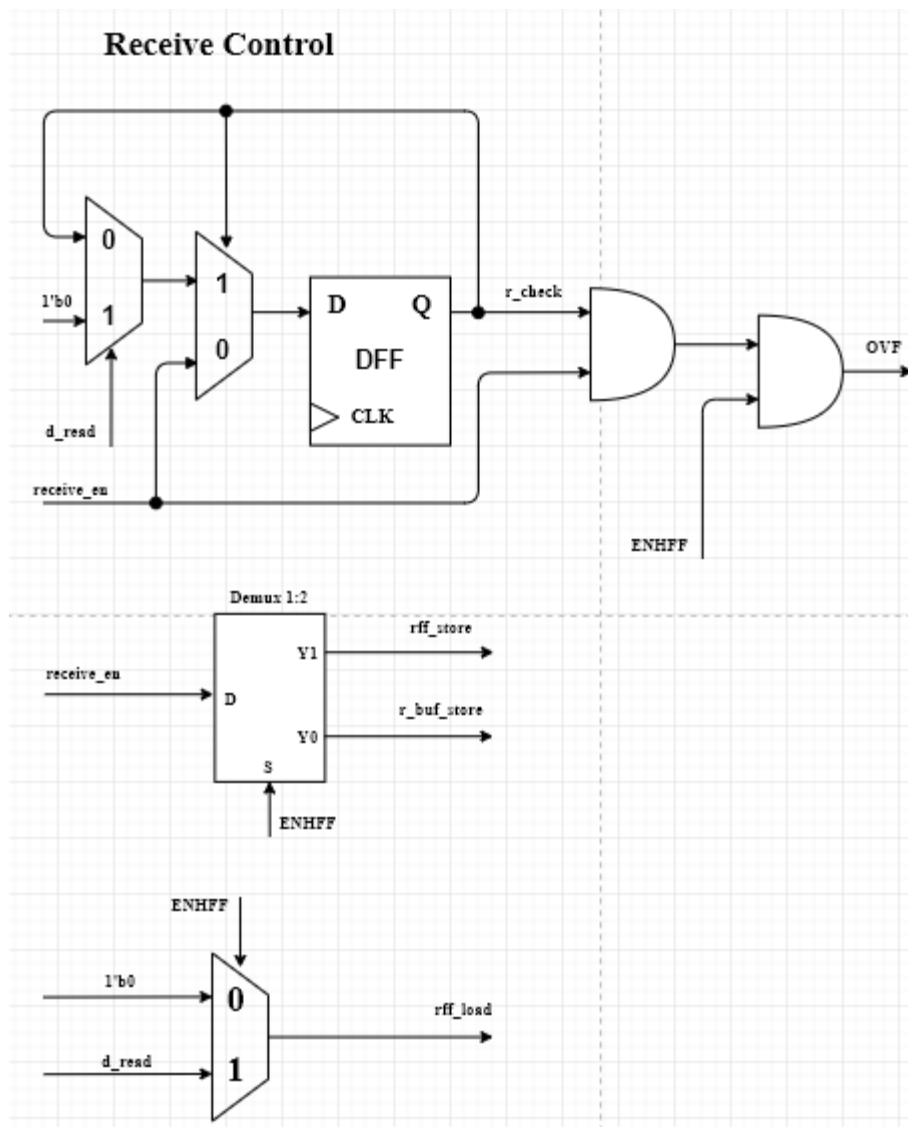


R Data Swap

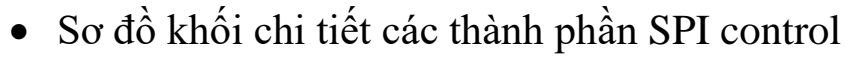


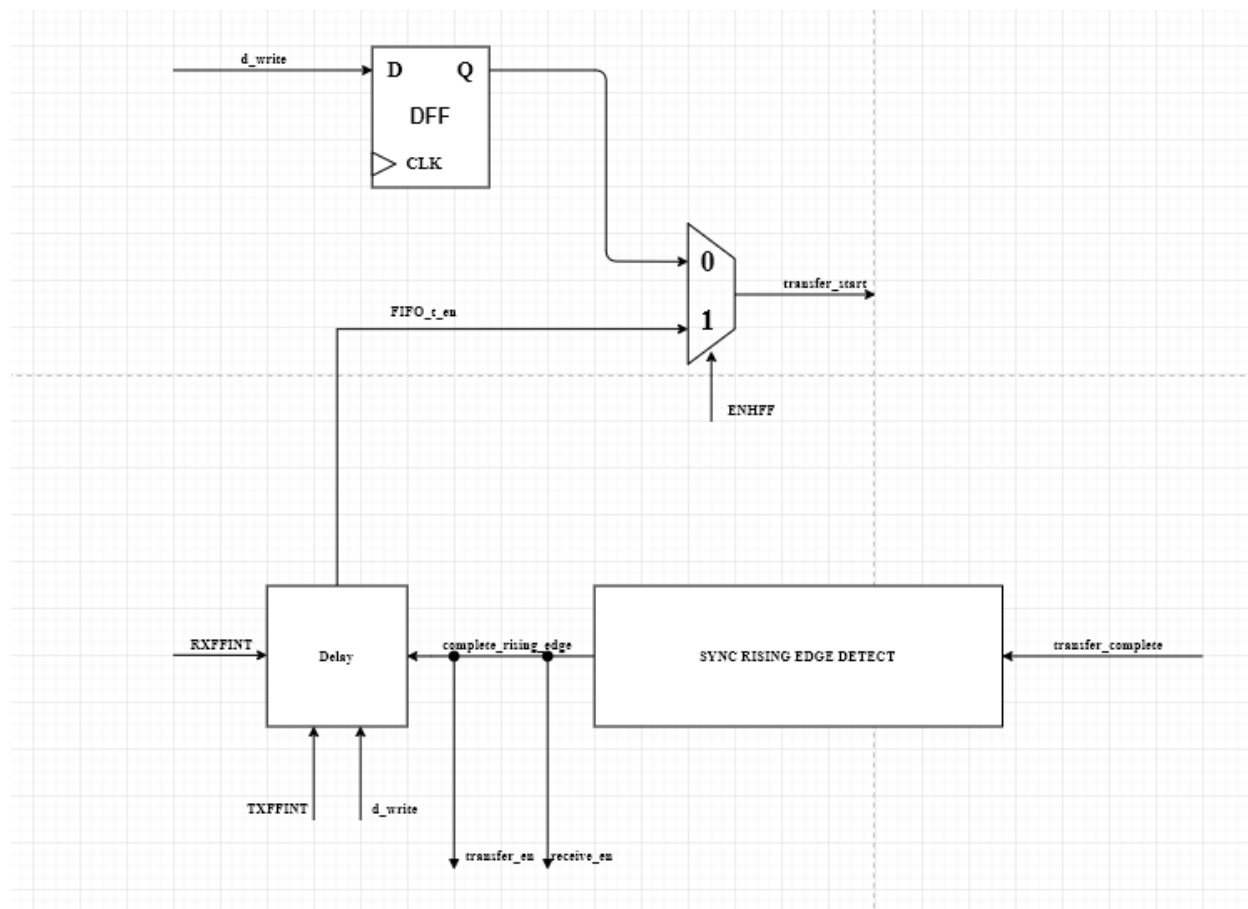
Transfer Control



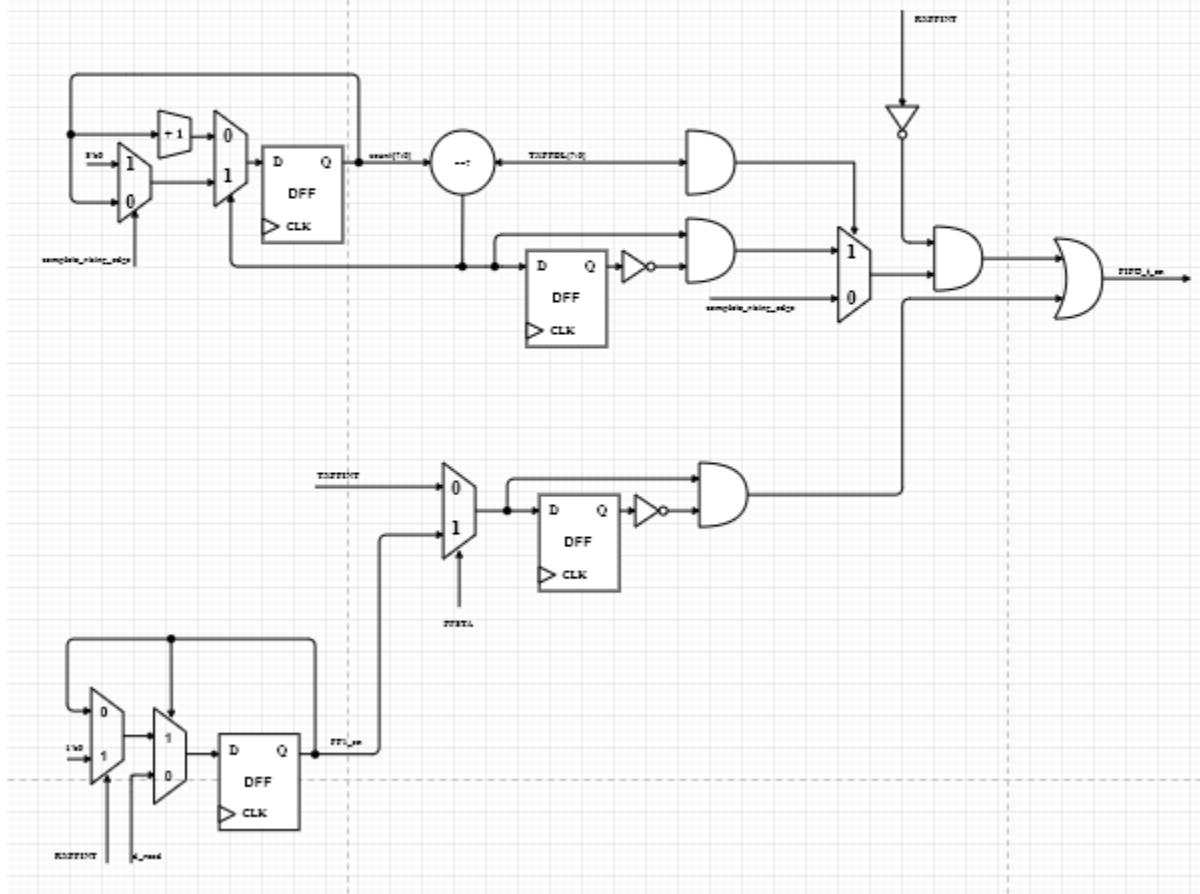


- Tín hiệu transfer_en đến từ SPI Control giúp SPI Data biết rằng mình cần phải load một ô dữ liệu mới từ FIFO truyền lên đường truyền transfer_data[31:0] khi nó ở chế độ Master.
- Khi không sử dụng FIFO, SPITXBUF ghi đè dữ liệu mới mỗi khi có một xung d_write tích cực mức cao.
- Tín hiệu Receive_en đến từ SPI Control giúp SPI Data biết rằng có một dữ liệu mới được nhận và sẵn sàng được ghi vào FIFO nhận hoặc SPIRXBUF
- Khối T Data Swap được sử dụng để chuyển dữ liệu cần truyền về dạng left-justified khi ở chế độ dịch MSB trước

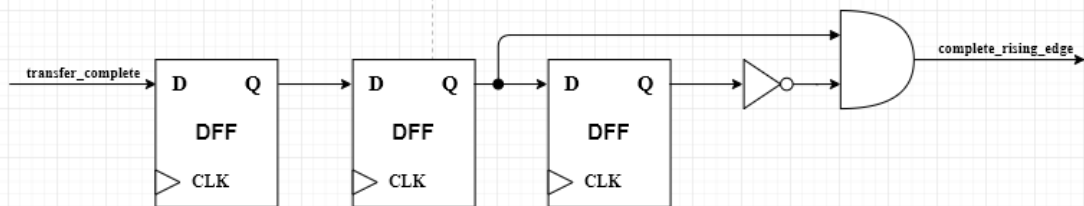




Delay



SYNC RISING EDGE DETECT



- Việc ghi từ APB vào SPISR (SPI Status Register) không làm thay đổi nội dung thanh ghi này
- Khối SYNC RISING EDGE DETECT được dùng để đồng bộ tín hiệu shift_complete được gửi từ miền clock dịch sang. Tín

hiệu tạo ra là một xung tích cực mức cao

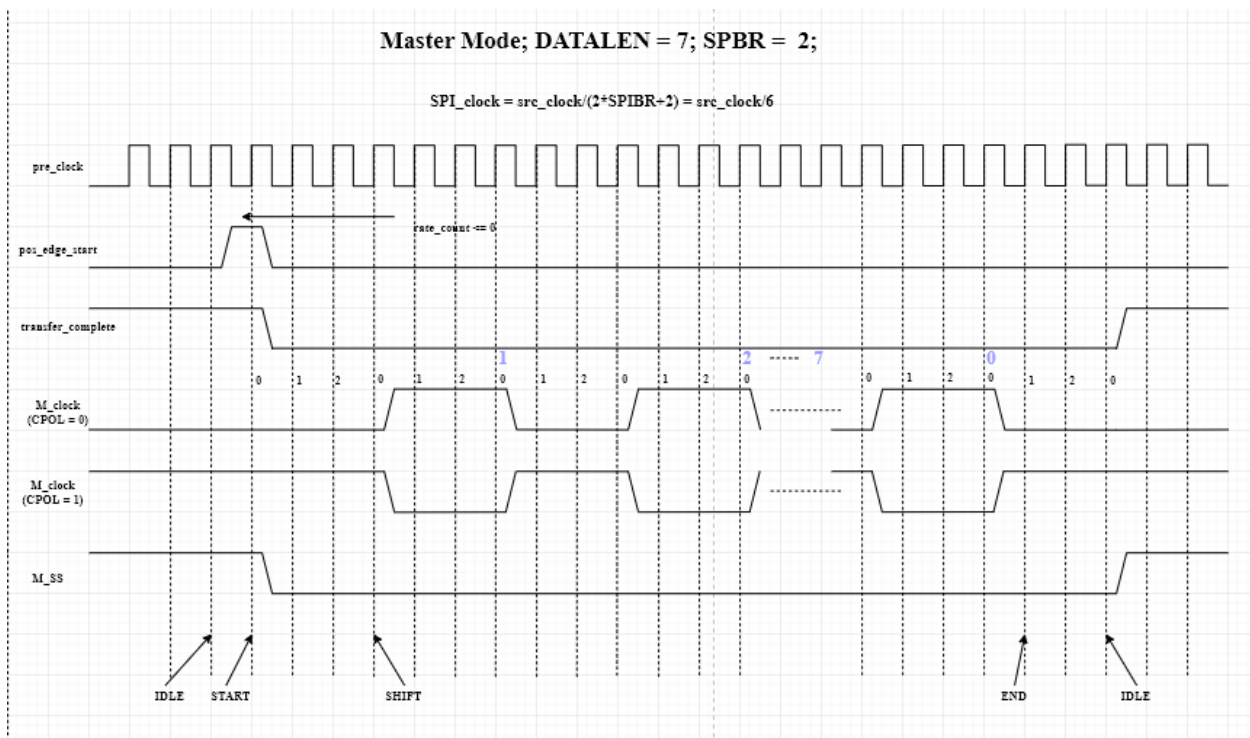
complete_rising_edge báo hiệu cho SPI biết là một chuỗi dữ liệu vừa được gửi xong.

- Khối Delay tạo delay giữa các lần kết thúc truyền nhận và bắt đầu sự kiện truyền mới nhằm chờ Slave kịp xử lý, đưa dữ liệu mới lên đường truyền sau khi một chu kỳ truyền nhận kết thúc

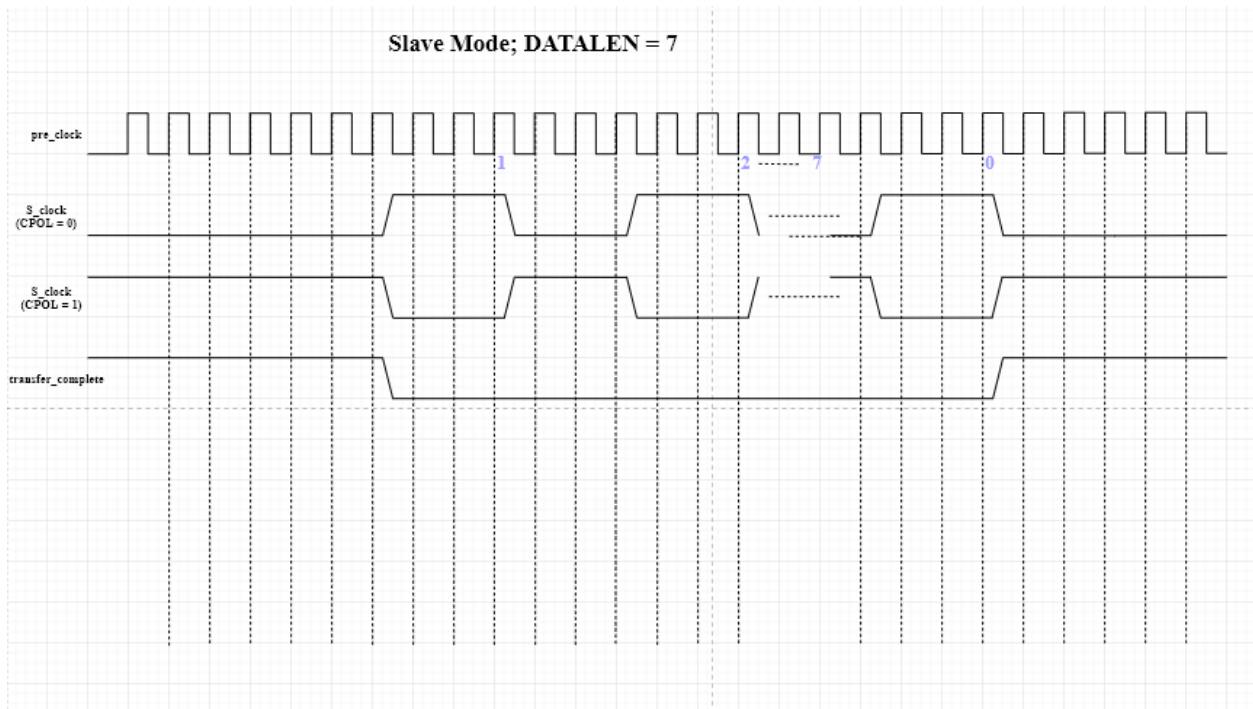
4) **Shift Control Clock:**

- Sơ đồ định thì cho một hoạt động của khối Shift Control Clock

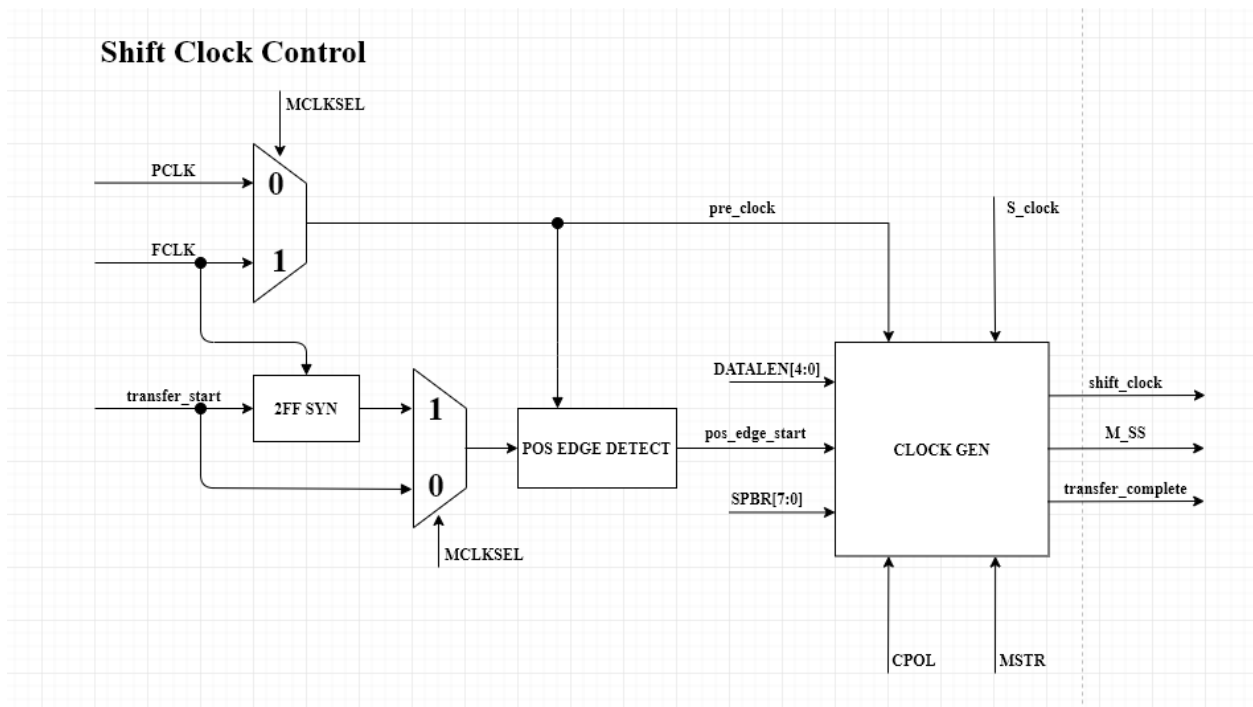
a) Minh họa cho ví dụ hoạt động ở chế độ Master, độ dài dữ liệu là 8 bit, tốc độ truyền là $\text{src_clock}/6$

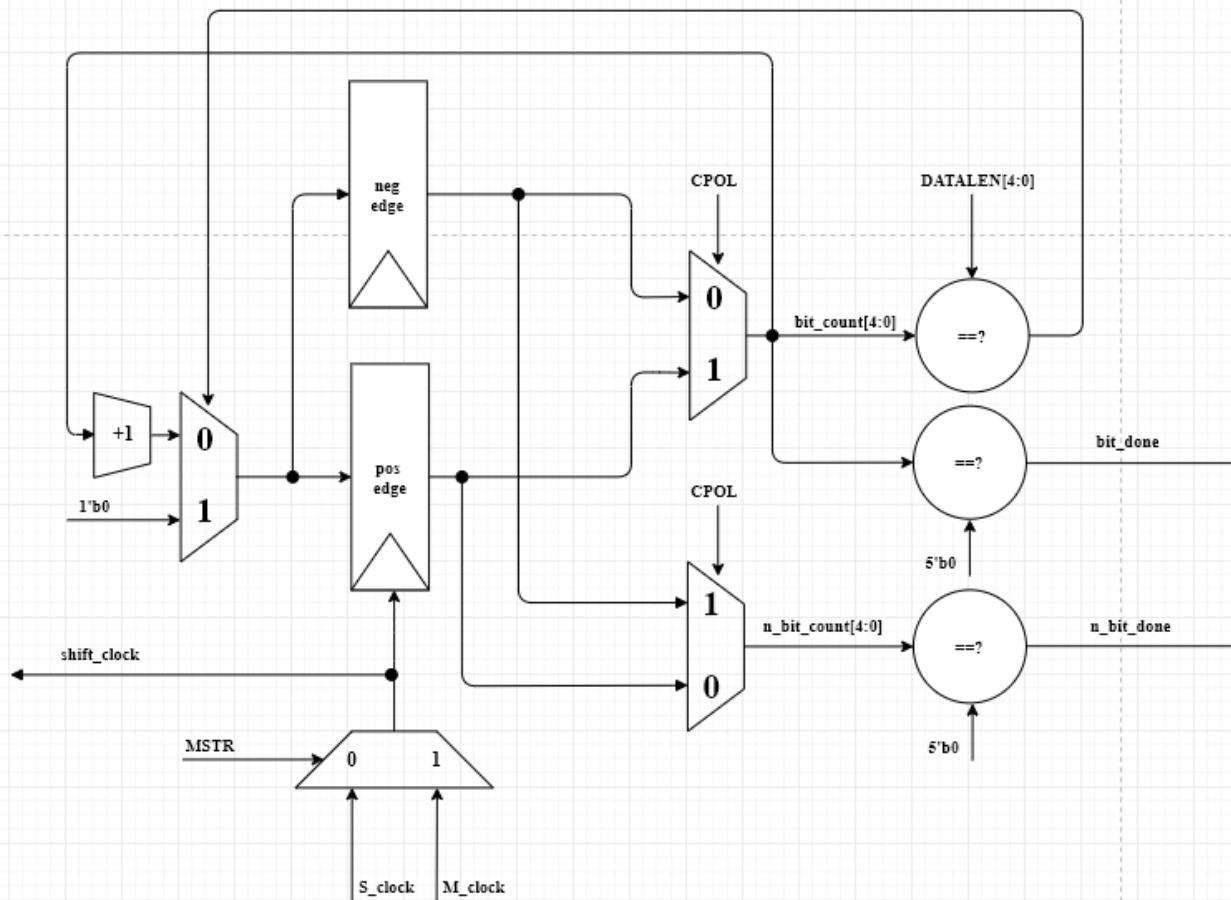


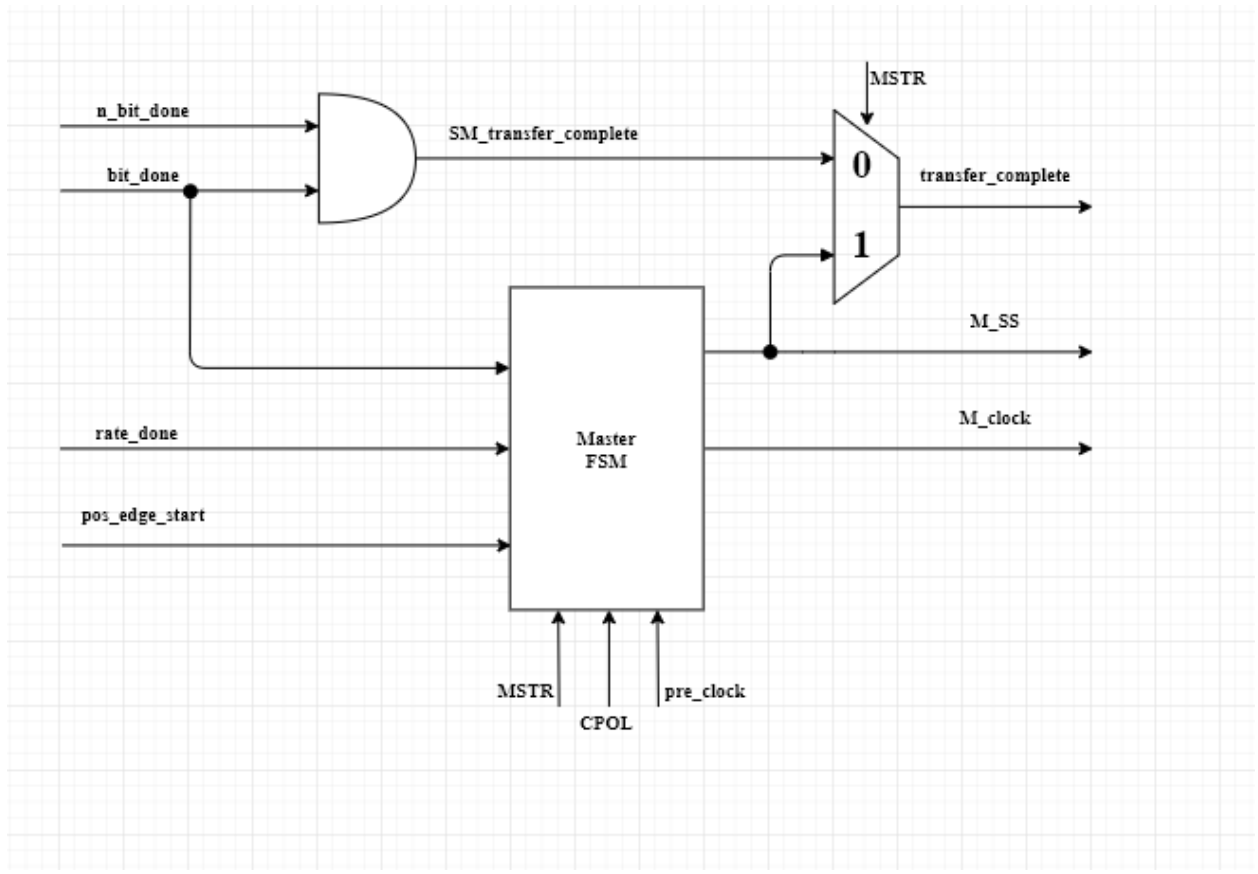
a) Minh họa cho ví dụ hoạt động ở chế độ Slave



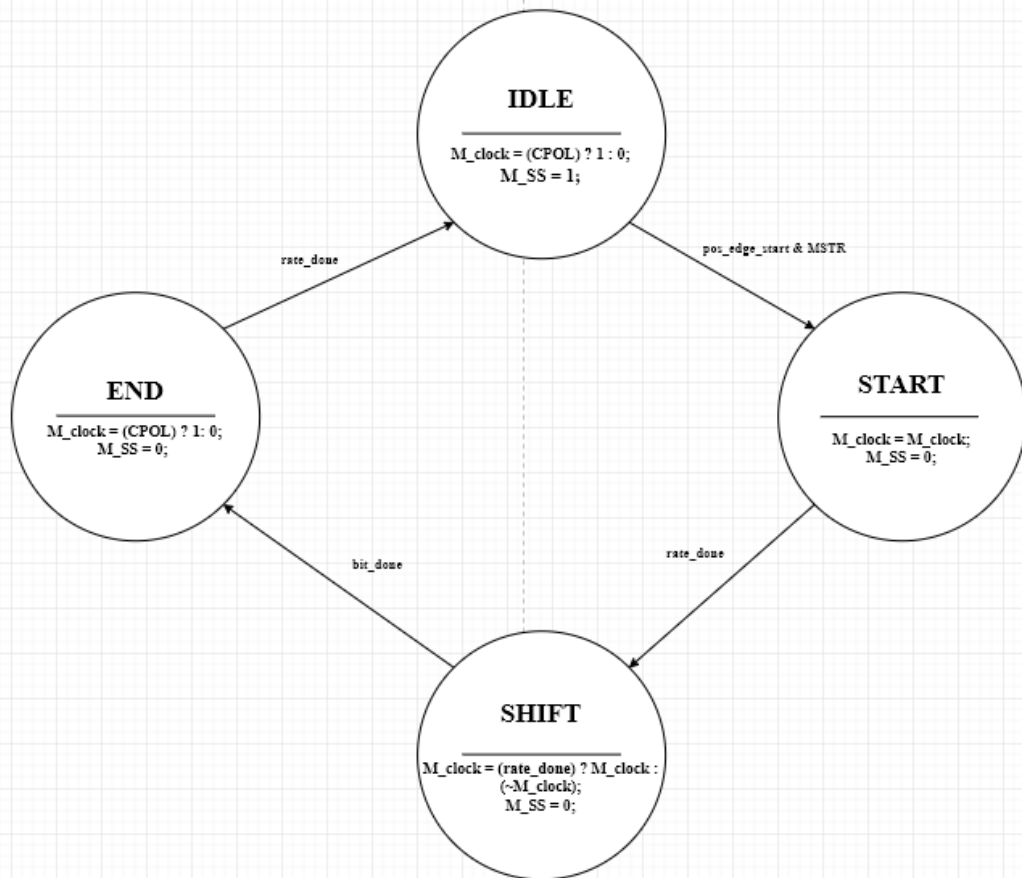
- Sơ đồ khối chi tiết của Shift Clock Control







Master FSM



- Các vấn đề ở khối Shift Clock Control:
 - Khi nguồn clock được chọn để tạo clock dịch là PCLK thì tín hiệu `transfer_start` được gửi từ SPI Control sang không cần đồng bộ. Tuy nhiên nếu nguồn clock được chọn là FCLK thì tín hiệu `transfer_start` cần được đồng bộ. Vấn đề ở đây là tín hiệu `transfer_start` chỉ được tích cực trong một chu kỳ PCLK do đó để đồng bộ đưa thì clock FCLK phải thỏa điều kiện:

$$T_{PCLK} \geq 2T_{FCLK}$$

$$\Rightarrow FCLK \geq 2PCLK \quad (1)$$

- o Tương tự cho việc đồng bộ tín hiệu transfer_complete được gửi từ khối Shift Clock Control sang khối SPI Control (tín hiệu này chỉ bằng 0 khi việc dịch dữ liệu đang diễn ra), để thỏa mãn yêu cầu đồng bộ tần số dịch phải thỏa điều kiện:

a) Chế độ Master:

$$(DATALEN + 1.5)T_{\text{shift_clock}} \geq 2T_{\text{PCLK}}$$

$$\Rightarrow \text{shift_clock} \leq \frac{(DATALEN + 1.5)PCLK}{2}$$

$$\Rightarrow \text{pre_clock} \leq (DATALEN + 1.5)PCLK$$

Với clock nguồn là PCLK thì công thức trên luôn thỏa. Tuy nhiên khi clock nguồn là FCLK:

$$FCLK \leq (DATALEN + 1.5)PCLK \quad (2)$$

b) Chế độ Slave:

$$(DATALEN + 0.5)T_{\text{shift_clock}} \geq 2T_{\text{PCLK}}$$

$$\Rightarrow \text{shift_clock} \leq \frac{(DATALEN + 0.5)PCLK}{2}$$

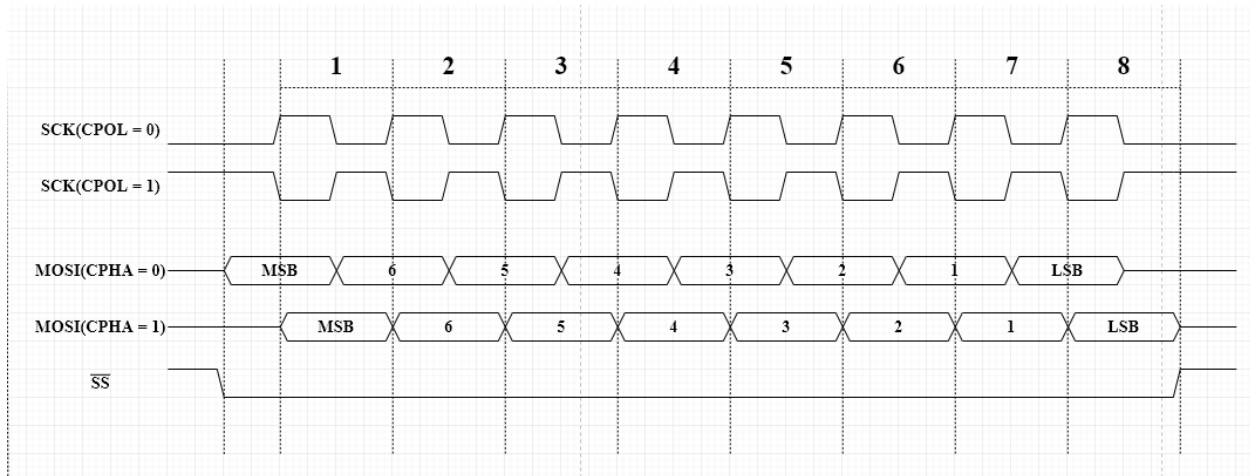
$$\Rightarrow S_clock \leq \frac{(DATALEN + 0.5)PCLK}{2} \quad (3)$$

Từ (1) (2) (3) ta có các điều kiện cho các nguồn clock:

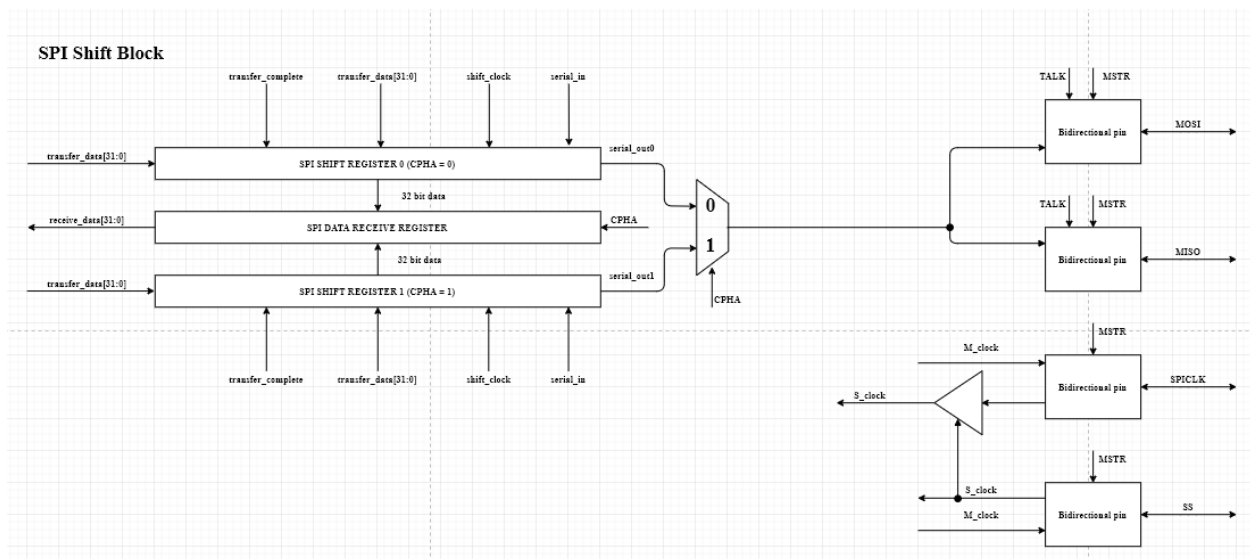
- $2PCLK \leq FCLK \leq (DATALEN + 1.5)PCLK$
- $S_clock \leq \frac{(DATALEN + 0.5)PCLK}{2}$

5) **SPI Shift Block:**

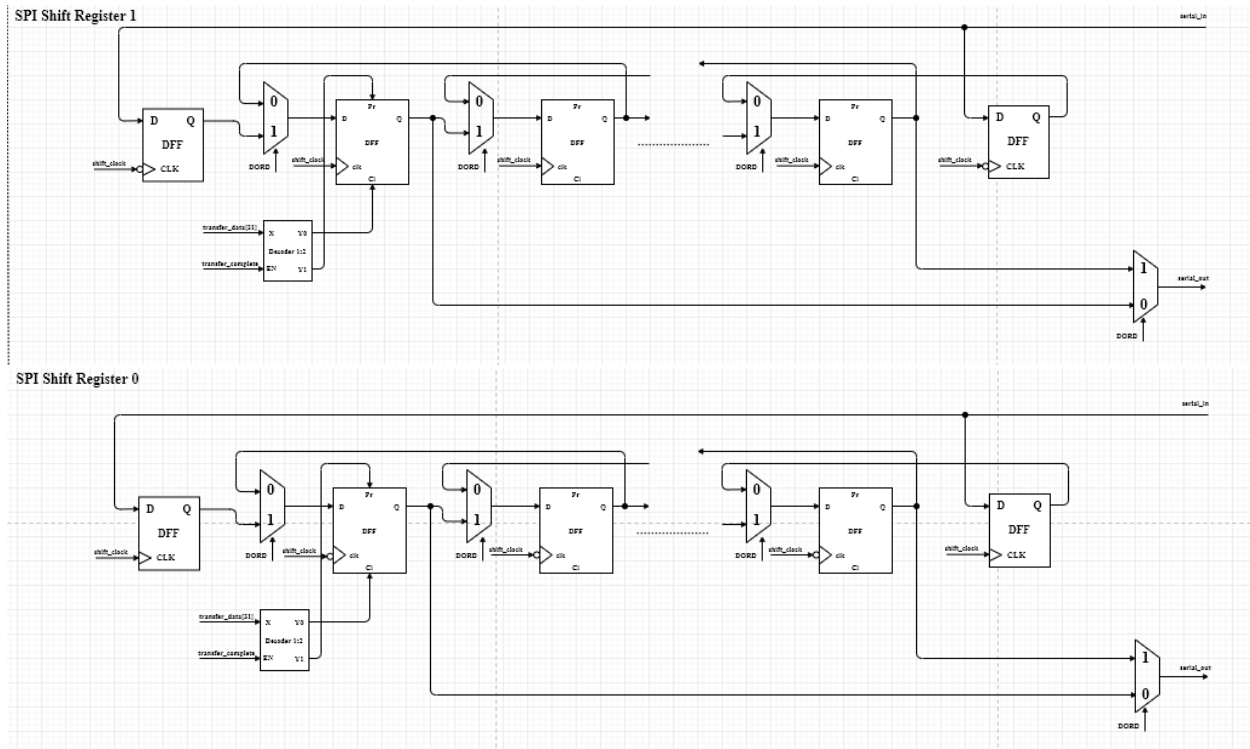
- Sơ đồ định thì khi SPI hoạt động ở chế độ Master, truyền 8 bit



- Sơ đồ chi tiết khối SPI Shift Block



- Tín hiệu S_clock chỉ được truyền vào SPI khi S_clock bằng 0
- Dữ liệu nhận được sau một chu kỳ truyền nhận sẽ được truyền song song vào thanh ghi SPI DATA RECEIVE REGISTER nhờ xung cạnh lên của tín hiệu transfer_complete



- Khi `transfer_complete` bằng 1 dữ liệu từ `transfer_data[31:0]` sẽ được đưa vào thanh ghi dịch, tùy vào bit `CPHA` mà thanh ghi dịch 0 hay 1 sẽ được chọn