Review (2020.04.18)

**1/ Về hình thức các tài liệu file word**

**Trước hết, mạnh dạn ghi tên họ, email, … đóng dấu tên bạn và những người làm cùng bạn vào bất cứ tài liệu nào do bạn tạo ra để khẳng định chủ quyền (không cần để tên những người không trực tiếp làm, ví dụ không cần để tên “Nguyễn Quân chỉ chém gió vào :D”)**

**Chú ý, sau này RTL code cũng vậy.**

**1.0 Gom 2 tài liệu mô tả spec vào trong 1 file duy nhất, mọi chỉnh sửa sau này chỉ làm trên 1 file (không tạo nhiều file v1 v2, v3, …., khi upload thì ghi rõ các chú thích thay đổi để khi cần có thể lấy lại version mong muốn)**

**1.1 Dùng format chung của dự án, thao khảo ở link sau:**

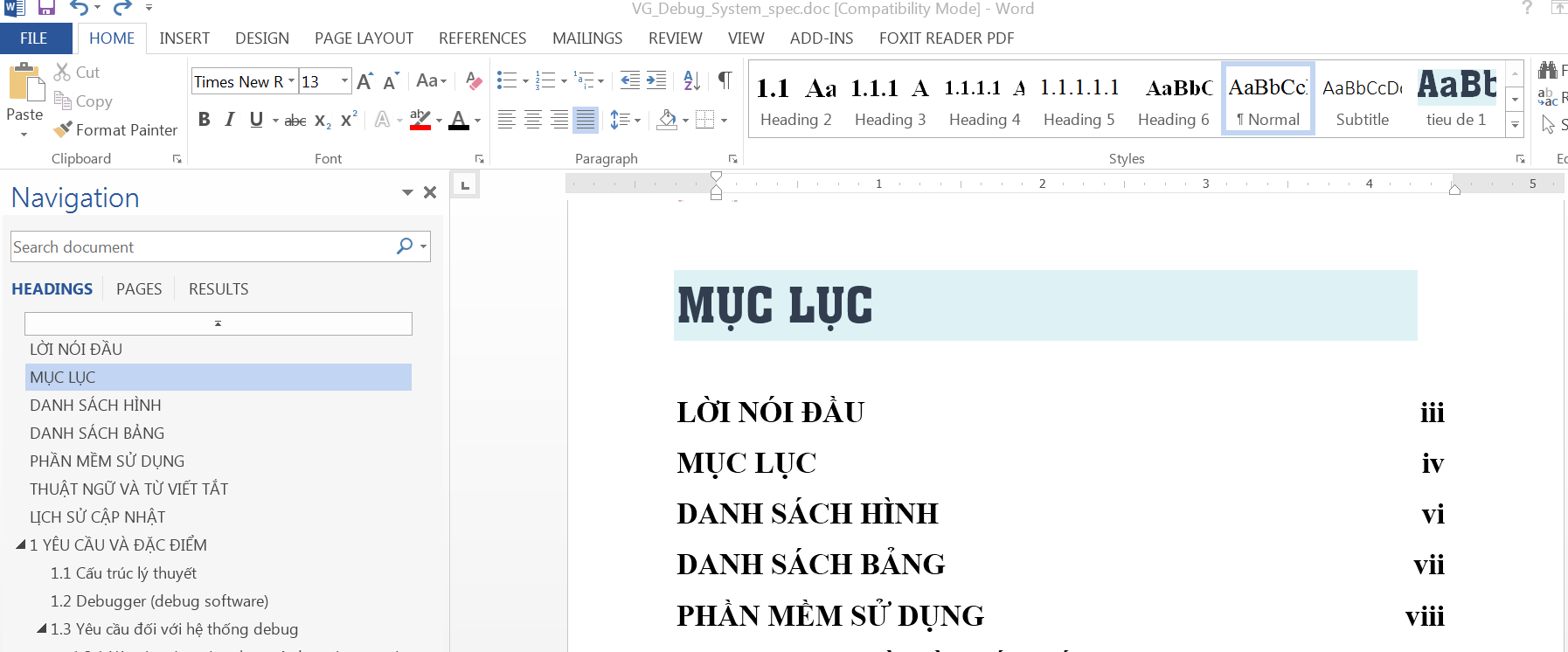
<https://github.com/nguyenquanicd/VG_Specification/tree/master/001_General_Spec>

**1.2 Các headline, caption của bảng/hình, font, … phải theo định dạng chung đã có (tạo sẵn) trong format trên. Dưới đây là nơi có thể tìm thấy các định dạng (format) đã tạo sẵn**

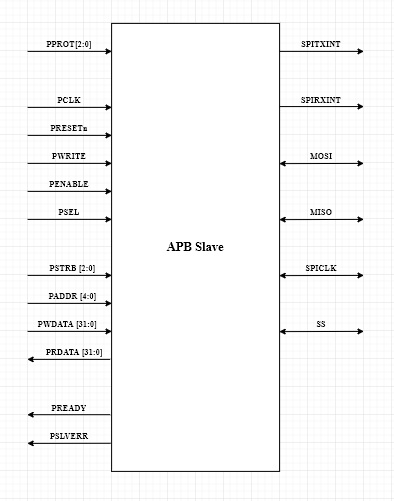


**1.3 Tài liệu phải có phần mục lục, mục lục hình, mục lục bảng**

Khi tạo các mục lục cần kiểm tra lại qua **View/Navigation Pane** để đảm bảo như ý muốn



**1.4 Các hình vẽ cắt trong file word cần bỏ dòng kẻ chỉ để nền trắng.**

****

**1.5 Tên các tín hiệu nên để đúng như khi viết RTL code nếu có thể.**

Tên các block thì để như RTL code nếu có thể (ko bắt buộc).

**1.6 Tên các tín hiệu và module dùng một trong 2 cách**

1. Đặt tên hoàn toàn chữ thường (không dùng chữ in hoa) và dấu “\_”. Ví dụ: preset\_n (Khuyến khích dùng cách này)
2. Đặt tên Có in hoa chữ cái đầu tiên của 1 cụm từ có nghĩa và không dùng dấu “\_”. Ví dụ: PResetN

Khuyến khích dùng cách 1 vì khi cần dùng script thao tác cắt chuỗi thì việc phát hiện dấu “\_” sẽ giúp cắt dễ hơn.

**1.7 Chữ in hoa toàn bộ chỉ dùng cho khai báo hằng số parameter, localparam, giá trị trong enum, …. và bắt đầu bằng tên viết tắt của module.**

<module name>\_<parameter name>

Ví dụ: parameter SPI\_PADDR\_WIDTH = 12;

**1.8 Với mỗi hình vẽ cho design/spec/logic đưa ra cần mô tả các nội dung sau**

* Mục đích: hình này thể hiện cái gì? Ví dụ: Hình này là logic chi tiết của khối APB interface
* Chức năng: thực thi chức năng gì? và liên kết với các phần logic nào khác?. Ví dụ: Logic này thực thi việc đọc/ghi vào thành ghi A. Nó sẽ lấy dữ liệu ghi và điều khiển từ khối APB interface. Sau đó, giá trị thanh ghi này sẽ đưa đến khối C và D.
* Không cần mô tả lại mạch logic đã có trong hình mà tập trung giải thích vì sao lại thiết kế như vậy.
* Ví dụ: “PREADY luôn được đặt bằng 1” vì APB interface luôn hoàn thành việc đọc/ghi khi Psel=Penable=1

**1.9 Hoàn toàn dùng thuần Việt (trừ các thuật ngữ tiếng Anh ko cần dịch).**

Ví dụ, không dùng “note: ….”, chuyển thành “chú ý: …”

Bên cạnh đó, “chú ý” chỉ là những nội dung bổ sung thêm bên cạnh nội dung chính. Một đoạn không mô tả bất cứ nội dung nào khác ngoài “chú ý” thì chú ý này hãy để là nội dung chính, và không cần “note”.

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

**1.10 Hãy tổng ra soát lại các câu chữ, với tài liệu kỹ thuật cần dùng câu đơn giản (câu đơn) và dễ hiểu, đảm bảo phân tách rõ trạng ngữ, chủ ngữ, và vị ngữ.**

Cái này là tùy khả năng mỗi người nhưng việc chú ý thay đổi sẽ giúp ích nhiều.

(Trong công việc thực tế, ở các cty, việc mô tả một tài liệu kỹ thuật, spec chính xác là yêu cầu đầu tiên trước khi RTL code, một lỗi hiểu nhầm hay sai từ mô tả spec sẽ gây tốn thời gian và tiền bạc nhất vì nó ảnh hưởng dây chuyền đến các khâu tiếp theo)

**1.11 Có một mục để các đường link đến các tài liệu tham khảo, source code, … hoặc các link liên quan. Ví dụ link Github.**

**1.12 Các nhận xét đều khổng thể đầy đủ với tất cả các trường hợp, hãy cố gắng tạo một tài liệu ngày càng hoàn chỉnh vì sau này nó sẽ đến tay người dùng và bạn chính là một nhà cung cấp design/IP.**

**1.13 Logo của nhóm nếu bạn cần xài**

<https://github.com/nguyenquanicd/VG_Specification/tree/master/000_CommonRef/VlsiTech_CommonInfo>

**1.14 Điều chỉnh lại mô tả các bit thanh ghi, ở đây chỉ là 1 ví dụ:**

Bỏ cột giá trị, để chung với mô tả vì mỗi lần viết, canh dòng rất mất thời gian.

Thêm cột default hoặc mô tả giá trị default trong bảng thanh ghi.

Thường (không phải tất cả), Trạng thái tích cực là trạng thái thực thi chức năng như ý nghĩa của tên bit và mức tích cực là 1 nếu không có chú thích khác. Điều này giúp người dùng nhanh chóng nắm bắt và hạn chế sai sót.

|  |  |  |  |
| --- | --- | --- | --- |
| 31 | SWR  (Software Reset) | 0  1 | 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.  Bit SWR được bật lên thì SPI mới được phép hoạt động |

Thành

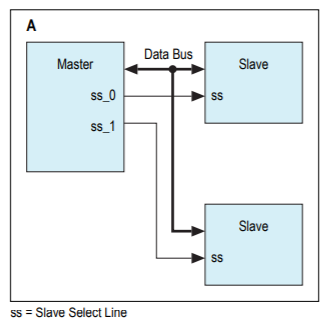
|  |  |  |
| --- | --- | --- |
| 31 | SWR | Software reset. Bit này tích cực mức 0.  0: Toàn bộ các cờ ngắt sẽ bị xóa. SPI không cho phép truyền nhận dữ liệu. Nếu bit SWR bị xóa lúc việc truyền nhận đang diễn ra thì dữ liệu đang gửi trong thanh ghi dịch sẽ vẫn tiếp tục được dịch đi.  1: Bit SWR được bật lên thì SPI mới được phép hoạt động |

Ví dụ khác trong mô tả bit, ở đây thuật ngữ “hoạt động bình thường” là rất chung chung và không được định nghĩa thế nào là “bình thường”? cần điều chỉnh lại tập trung và các chức năng mà bit này sẽ ảnh hưởng khi nó bằng 0 hoặc bằng 1. Ví dụ, nếu chỉ ảnh hưởng đến cấu hình chân thì chỉ nói về điều đó. Còn nếu còn ảnh hưởng đến hoạt động khác thì hãy chú ý thêm.

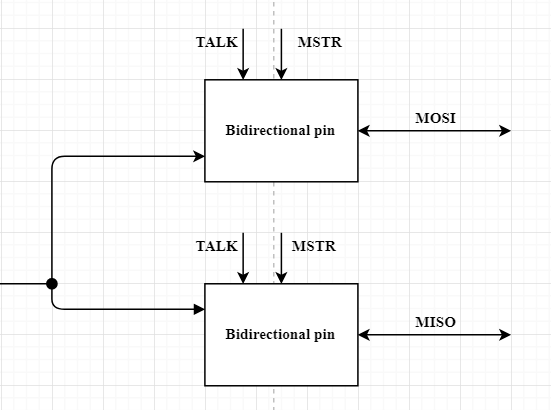
|  |  |  |  |
| --- | --- | --- | --- |
| 24 | TALK | 0  1 | + Ở chế độ Master: hoạt động của SPI vẫn bình thường tuy nhiên các ngõ ra MOSI, và SPICLK sẽ được đặt ở trạng thái trở kháng cao.  + Ở chế độ Slave: hoạt động của SPI vẫn bình thường tuy nhiên ngõ ra SOMI sẽ được đặt ở trạng thái trở kháng cao.  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 |

**2/ Về design của SPI**

1. **Hỗ trợ nhiều giao tiếp với nhiều SPI slave thì SS cần nhiều bit, cần có một bit enable riêng cho từng tín hiệu SS**

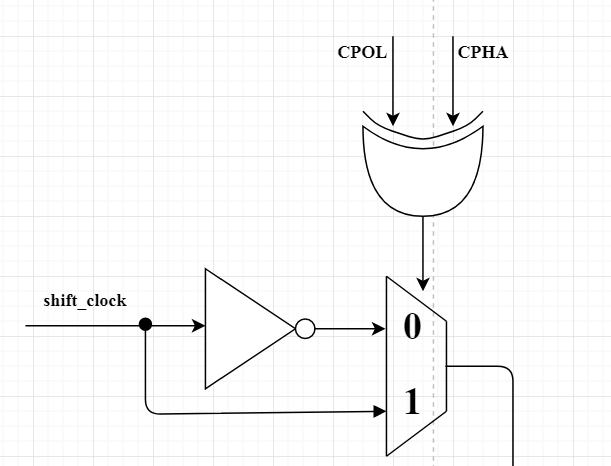


1. **Chỉ khi chân SS tích cực thì MOSI và MISO mới được lái. Bên cạnh đó với SPI, MOSI và MISO là 1 chiều? (chỉ 2 chiều với QSPI ???)**



1. **Xem xét các logic đặt phù hợp với các block đã được phân chia**

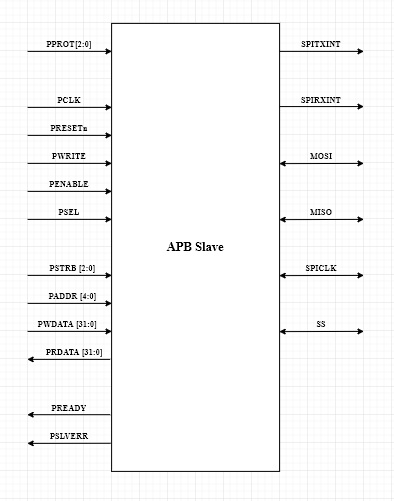
**Chuyển mạch tạo shft clock về khối clock control, tajid dây tạo 2 tín hiệu shift\_clock và shift\_lock\_n (bù của shift\_clock) để nối đến khối dịch dữ liệu**



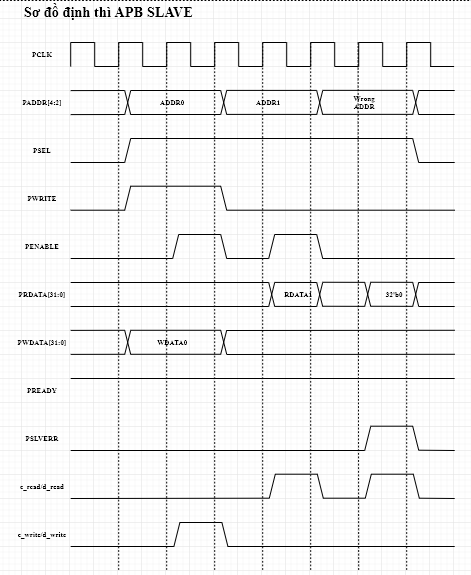
1. **Độ rộng các tín hiệu**

PADDR[4:0] chuyển thành paddr[SPI\_PADDR\_WIDTH-1:0], đặt một parameter cho độ rộng address.

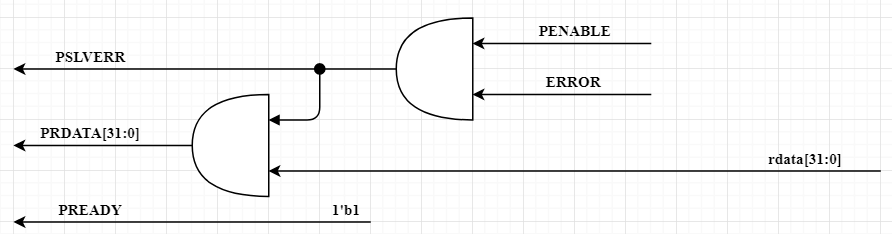
PSTRB[2:0] là pstrb[3:0]

****

1. **Thiếu PENABLE=1, dù address đưa đến địa chỉ không dùng nhưng penable vẫn có**



1. **Xem lại logic prdata, với logic hiện tại, khi nào có error thì prdata mới có giá trị để đọc ???**



1. **Để đảm bảo timing khi kết nối hệ thống, tất cả các ngõ ra của module cần phải trực tiếp từ FF (yêu cầu bắt buộc trong Soc này, trừ trường hợp đặc biệt thì phải thảo luận)**

Ở đây prdata, pslverr, phải từ FF

pready=1 thì không cần

1. **Gom phần logic decoder đọc ghi các thanh ghi về APB interface**

Cố gắng đơn giản và dễ hình dung. Xem minh họa dưới đây.

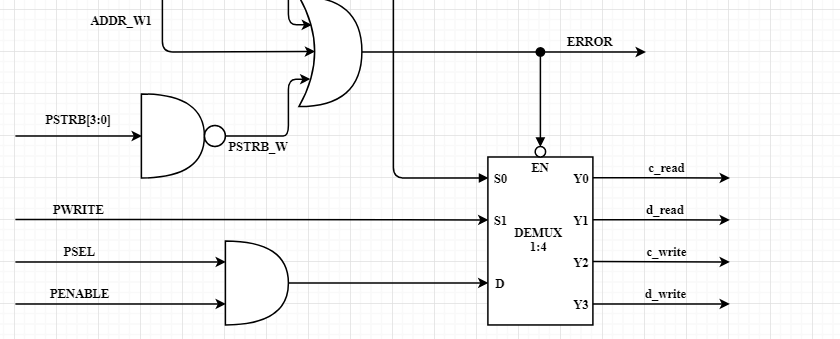
reg\_we có thể dùng penable hoặc không

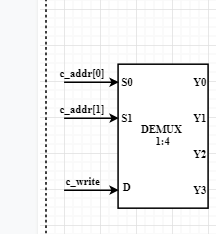
Có dùng: giúp gate clock hiệu quả hơn nếu gate clock được dùng vì reg\_we chỉ tích cực 1 chu kỳ. Nếu dùng thì có thể chọn penable hoặc ~penable

Không dùng thì register sẽ cập nhật trong cả 2 chu kỳ

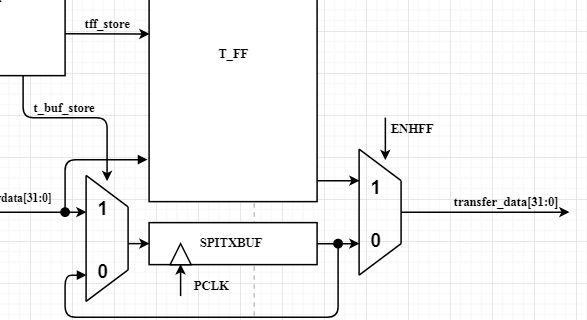


Cái trên khác design hiện tại, ví dụ decode cho write 1 phần ở APB, 1 phần ở CTL, …





1. **Xem xét bỏ chức năng hỗ trợ Buffer trên cả TX/RX (chỉ hỗ trợ FIFO) vì hiệu quả sử dụng ko nhiều**



Ở đây, trong cả 2 trường hợp, vẫn cần 1 chu kỳ để data[31:0] đến transfer\_data.

1. **serial\_in lấy từ đâu?**
2. **Xem xét lại mạch swap và serial\_out. Cần điều chỉnh để serial\_out đi ra từ 1 FF không phải mạch tổ hợp (MUX).**

Đề xuất như sau:

serial\_out luôn nối đến bit 0 của thanh ghi dịch

Mạch swap sẽ dựa trên DATALEN và DORD để xếp lại thứ tự bit trước khi lưu vào FIFO

1. **Phân chia lại các thanh ghi**

Tách nhóm điều khiển và trạng thái ngắt ra thành 3 thanh ghi

Thanh ghi enable

Thanh ghi raw interrupt

THanh ghi interrupt

Tách baud rate thành thanh ghi riêng (tuy không dùng hết 32 bit nhưng cần xem xét khả năng hỗ trợ độg rộng bit lớn hơn trong tương lai)

FIFO depth sẽ không phải là các bit cấu hình thanh ghi, nó chỉ là parameter trong RTL code. Một thông tin cần hỗ trợ người dùng là có bit trạng thái thể hiện TX FIFO và RX FIFO đang có báo nhiêu dữ liệu

Cần thêm bit enable cho toàn bộ SPI (đây là bit cơ bản của tất cả các IP). CHỉ khi bit này tích cực thì SPI mới được phép hoạt động.

1. **Mô tả flow lập trình sử dụng SPI**

Phần này là một spec quan trọng, không chỉ ảnh hưởng đến user mà còn ảnh hưởng đến design.

Ví dụ: Baudrate sẽ được lập trình khi nào? trước khi SPI được enalbe hay sau khi SPI được enable? và những bit cấu hình khác … Hay trong quá trình SPI đang truyền nhận dữ liệu các bit cấu hình có được phép lập trình lại hay không?