

NHÓM 1

Bài Tập SoPC

MSSV	Họ tên
1720057	Nguyễn Hữu Đức
1720069	Nguyễn Võ Hoàng Duy
1720134	Bùi Văn Linh
1720135	Đông Phạm Khánh Linh
1720239	Lê Quốc Toàn
1720258	Nguyễn Võ Lam Tuyền
18200006	Nguyễn Trọng Quang Huy
18200051	Phan Tuấn An

Câu 1: Phân biệt khái niệm Master và Slave

Master:

- Là phần cứng hay chương trình điều khiển các phần cứng hay chương trình khác, thường là Slave.
 - Là phần cứng chủ động.
 - Là phần cứng gửi tín hiệu bắt đầu quá trình đọc và ghi lên những Slave.
- Ví dụ:
- CPU
 - DMA Controller
- Master điều khiển Slave bằng gửi ra các tín hiệu:
- Address
 - Write/Read
 - Byte enable
 - Bus data Write

Slave:

- Là phần cứng hay chương trình được điều khiển bởi phần cứng hay chương trình khác, thường là Master.
 - Là phần cứng bị động.
 - Là phần cứng nhận tín hiệu và xử lý và giải quyết yêu cầu đến từ Master.
- Ví dụ:
- RAM
 - ROM
 - I/O
 - Sensor
 - Disk
- Slave trả về kết quả hoặc giao tiếp với Master bằng các tín hiệu:
- Wait request
 - Read data valid
 - Bus data Read

Master và Slave không giao tiếp trực tiếp mà thông qua một hệ thống Bus. Hệ thống Bus có nhiệm vụ phân giải các tín hiệu và là kênh truyền dữ liệu để cho Master và Slave có thể giao tiếp dễ dàng.

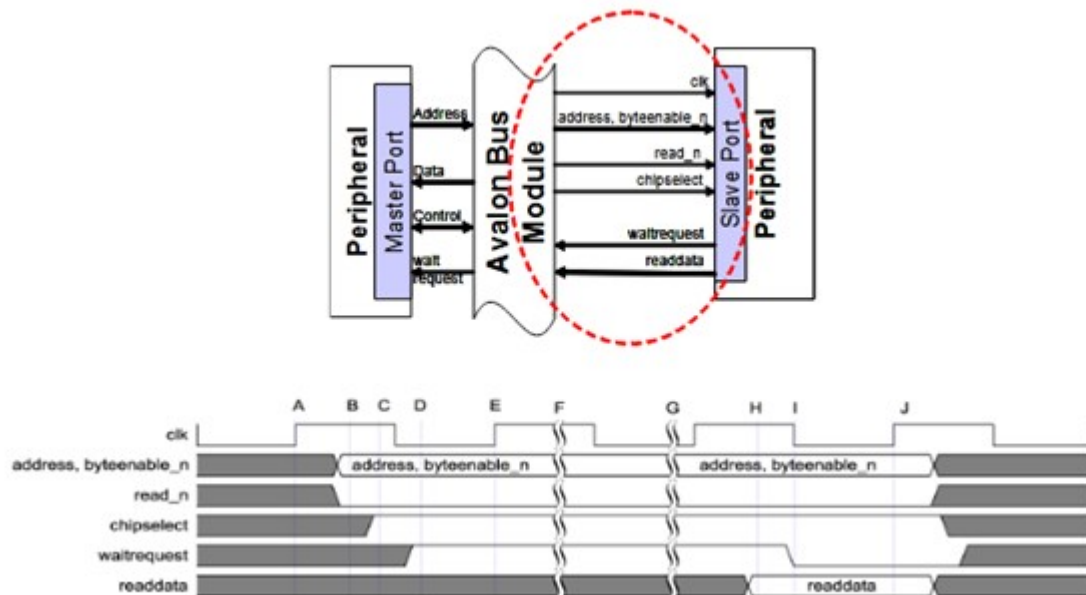
Một Master có thể điều khiển nhiều Slave. Ngược lại, một Slave cũng có thể đáp ứng cho nhiều Master.

=>Master : khởi tạo quá trình truyền dữ liệu.

=>Slave: Đáp ứng với yêu cầu truyền dữ liệu (có nguồn gốc từ Master).
Master và Slave không kết nối trực tiếp với nhau mà phải thông qua Avalon

Câu 2: Giải thích các dạng sóng của hình bên dưới

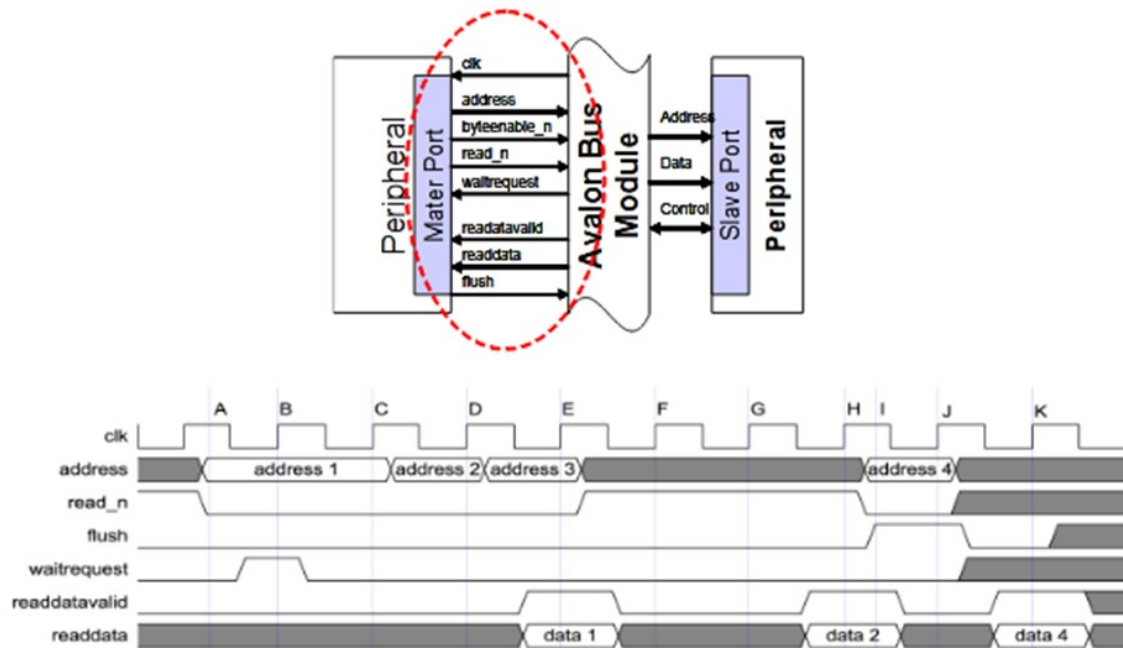
Slave



- (A) Chu kỳ bus đầu tiên bắt đầu trên cạnh lên của xung *clk*.
- (B) Có tín hiệu *address*, *read_n* từ Avalon bus đến slave.
- (C) Avalon bus giải mã tín hiệu *address* sau đó xác nhận *chipselect*.
- (D) Slave port xác nhận tín hiệu *waitrequest* và chờ trước khi cạnh lên tiếp theo của *clk*.
- (E) Avalon bus tiếp tục chờ ở cạnh lên của *clk*. Tín hiệu *waitrequest* được xác nhận, vì vậy tín hiệu *readdata* không được ghi lại ở cạnh trên của xung *clk* này.
- (F-G) Với tín hiệu *waitrequest* được xác nhận xuyên suốt, vô số chu kỳ bus sẽ trôi qua.
- (H) Slave port trình bày dữ liệu *readdata*.
- (I) Slave port cảnh báo tín hiệu *waitrequest*.

(J) Module Avalon bus ghi lại tín hiệu *readdata* trên cạnh lên tiếp theo của *clk* và quá trình chuyển đọc kết thúc ở đây. Chu kỳ bus tiếp theo có thể là sự bắt đầu của một chu kỳ bus khác.

Master



(A) Master bắt đầu chuyển đọc bằng cách trình bày *address* và *read_n* cho giai đoạn chuyển address mới.

(B) Module Avalon bus đã xác tín hiệu *waitrequest* nên master port đợi và giữ nguyên tín hiệu *address* và *read_n* cho một chu kỳ bus khác.

(C) *Waitrequest* không được xác nhận, vì vậy module Avalon bus ghi lại tín hiệu *address* ở cạnh lên của xung *clk*. *readdatavalid* không được xác nhận, vì vậy master không ghi lại *readdata*.

(D) Module Avalon bus ghi lại tín hiệu *address* mới ở cạnh lên của *clk*. *readdatavalid* cũng không được xác nhận, vì vậy master cũng không ghi lại *readdata*.

(E) Module Avalon bus ghi lại tín hiệu *address* mới ở cạnh lên của *clk* (thực hiện tổng cộng ba lần chuyển đang chờ xử lý). *readdatavalid* được xác nhận, do đó, master ghi lại *readdata* hợp lệ.

(F) *readdatavalid* không được xác nhận, vì vậy master không ghi được *readdata* hợp lệ.

(G) *readdatavalid* không được xác nhận, vì vậy master không ghi được *readdata*

(H) *readdatavalid* được xác nhận, do đó master ghi được *readdata* hợp lệ.

(I) Master trình bày *address* và *read_n* cho một lần chuyển đọc mới.

(J) *readdatavalid* không được xác nhận, vì vậy master không ghi được *readdata.flush* được xác nhận, vì vậy module Avalon bus xả quá trình chuyển đang chờ xử lý. Module Avalon bus ghi tín hiệu *address* mới.

(K) *readdatavalid* được xác nhận, vì vậy master ghi lại *readdata* hợp lệ. Không có quá trình nào đang chờ xử lý.