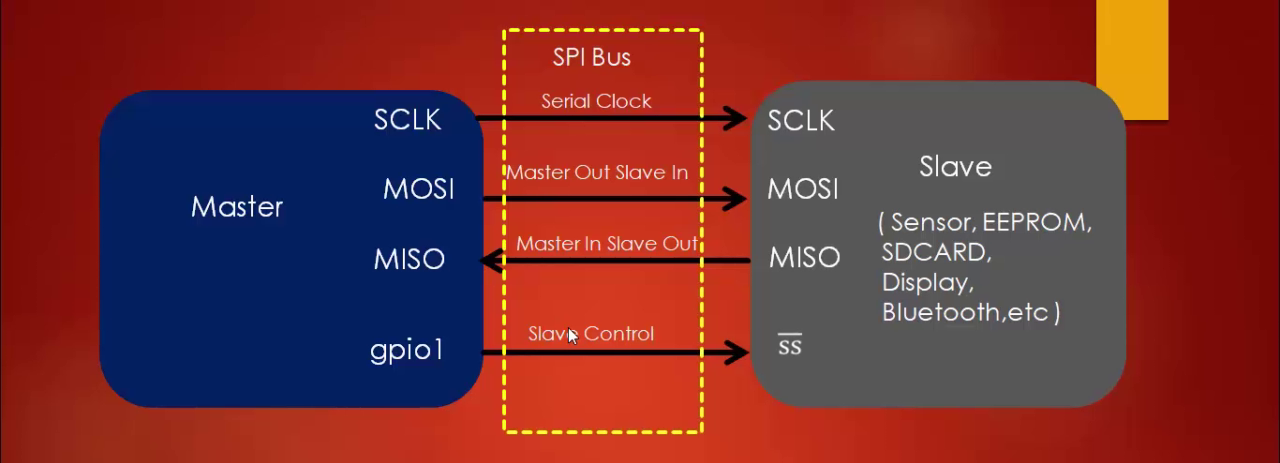
# GIAO TIẾP SPI

## GIỚI THIỆU VÀ CHI TIẾT VỀ BUS SPI

### Giới thiệu về SPI

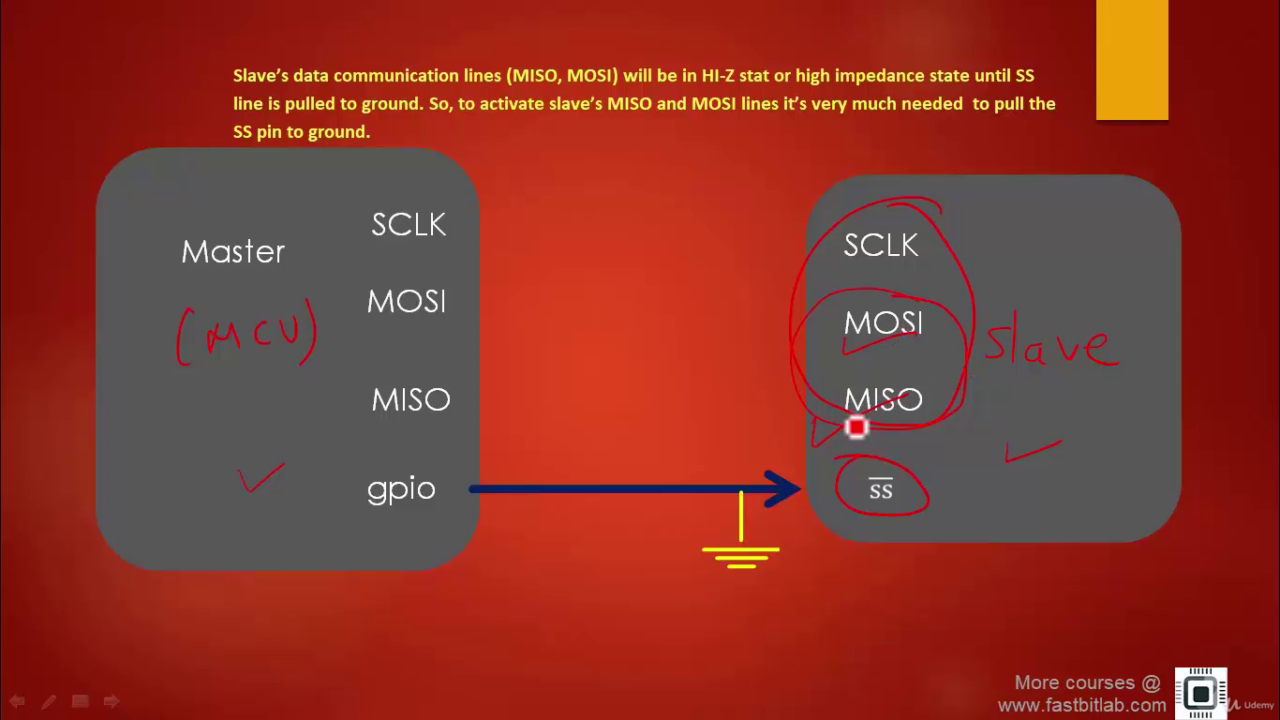
SPI là viết tắt của Serial Peripheral Interface, đây là một giao thức được sử dụng để truyền nhận dữ liệu nối tiếp giữa một master và một hoặc nhiều thiết bị slave trong hệ thống nhúng.



Slave có thể là cảm biến, chip EEPROM, thẻ nhớ SD, màn hình hoặc chip kết nối không dây như RF, Bluetooth, WiFi…

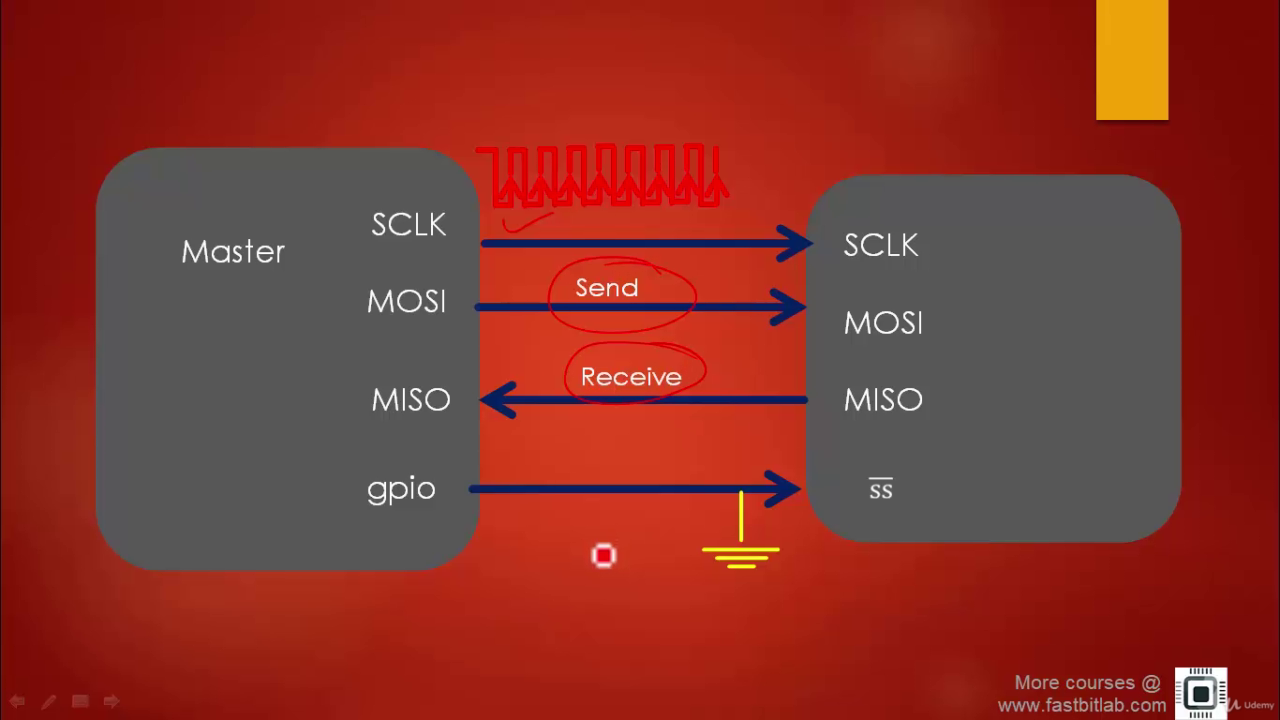
### Sự quan trọng của chân Select Slave (SS)

Để sử dụng được giao tiếp SPI, thì bắt buộc Master và Slave phải hỗ trợ SPI :). Mỗi khi Master muốn giao tiếp với Slave thì việc đầu tiên Master cần làm đó là chọn Slave, Master sẽ sử dụng một chân của GPIO để kéo chân SS xuống mức 0 hoặc nối đất (GND). Đây là một bước rất quan trọng, vì chỉ khi kéo chân SS xuống đất, thì các chân SCLK, MOSI và MISO mới được kích hoạt và truyền thông sẽ không được kích hoạt trên hai đường MOSI và MISO nếu không hai đường này sẽ ở trạng thái Hi-Z (trở kháng cao).



Đây là một lưu ý rất quan trọng, do đó chúng ta cần phải sử dụng một chân GPIO của Master nối với chân SS của Slave, và sau đó phải kéo chân SS của Slave xuống đất thì tiếp theo đó, chúng ta mới có thể truyền nhận được dữ liệu.

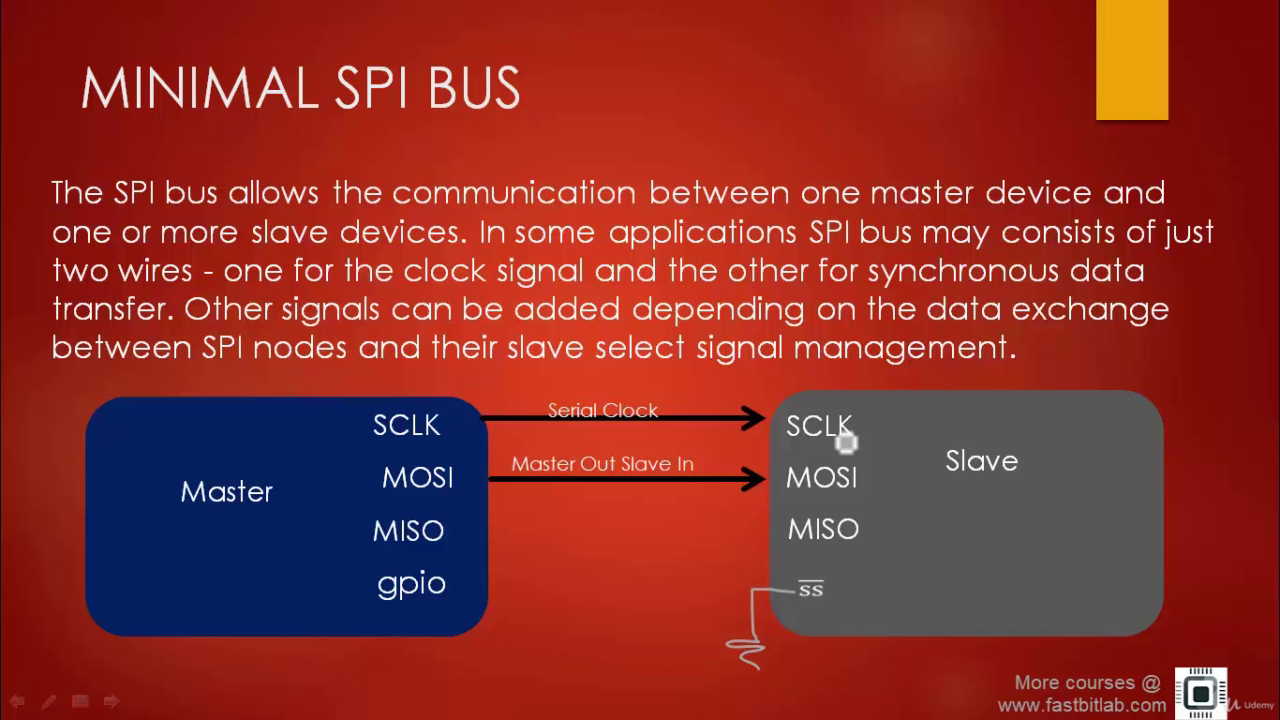
Một lưu ý nữa là, để truyền nhận dữ liệu thì xung Clock phải được gửi đồng thời với dữ liệu. Tất cả dữ liệu truyền nhận sẽ được đồng bộ với xung Clock.



**Chốt: Việc kéo chân SS xuống mức 0 (GND) là vô cùng quan trọng trong giao tiếp SPI. Nếu không thực hiện, thì các chân của Slave sẽ ở trạng thái Hi-Z và Slave sẽ không đáp ứng được bất kỳ dữ liệu nào được gửi bởi Master.**

### Cấu hình tối thiểu của SPI

Giao tiếp SPI cho phép truyền thông giữa một Master với một hoặc nhiều thiết bị Slave. Do đó, trong một vài ứng dụng, SPI có thể chỉ có 2 dây, một cho tín hiệu Clock và một cho truyền dữ liệu đồng bộ. Các tín hiệu này có thể được thêm tùy vào sự trao đổi dữ liệu giữa các nút SPI và quả lý tín hiệu chọn Slave (SS).



Đây là một điều quan trọng cần phải nhớ.

Giả sử chúng ta có một Master và một Slave, và Slave chỉ có nhận dữ liệu và không bao giờ gửi dữ liệu về Master. Trong trường hợp này chúng ta chỉ cần sử dụng hai đường là SCLK và MOSI, lúc này SS của chúng ta sẽ luôn nối đất để cho phép Slave kích hoạt SPI. Việc sử dụng bốn đường lúc này là không cần thiết. Do đó tùy thuộc vào ứng dụng chúng ta sử dụng mà có thể tùy chỉnh các chân của SPI, và nó có thể có thêm nhiều chân tùy vào ứng dụng và số lượng Slave.

### Cách hoạt động của SPI

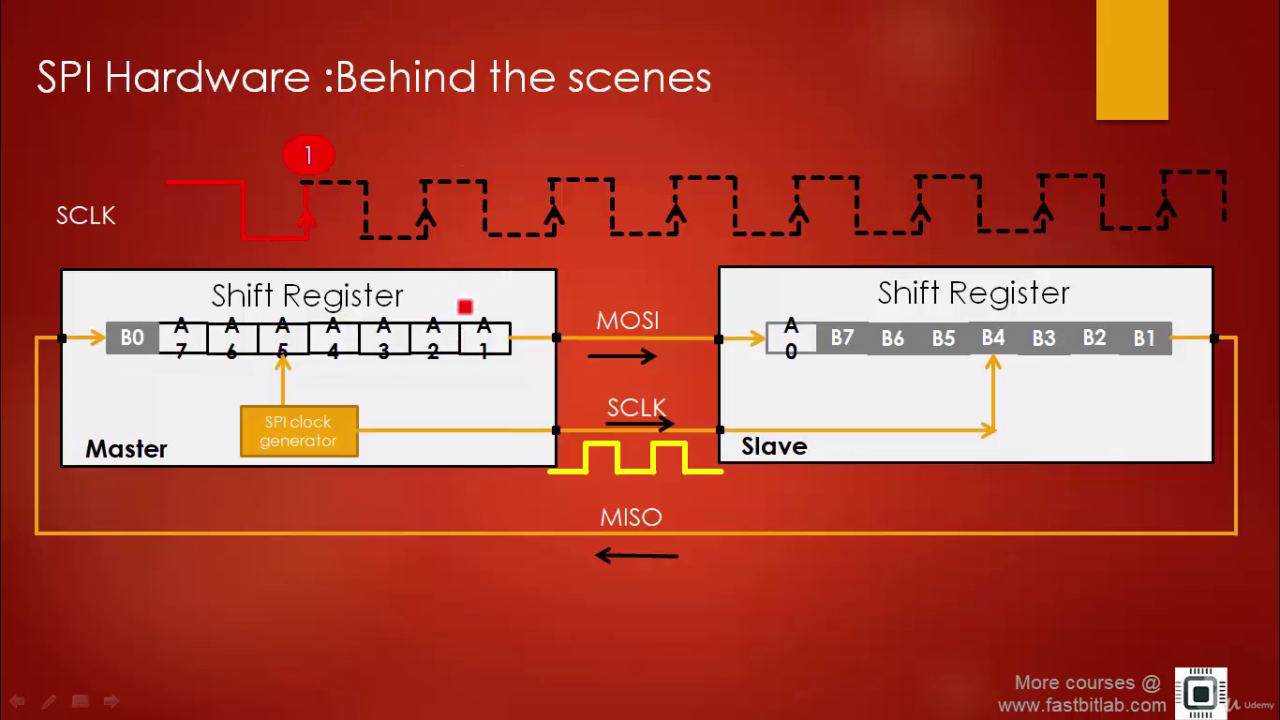
Phần này chúng ta sẽ tìm hiểu một cách chính xác những gì mà giao tiếp SPI diễn ra giữa Master và Slave. Làm sao để chúng trao đổi được dữ liệu?

Trong bất cứ ngoại vi SPI nào của vi điều khiển, chúng ta sẽ luôn thấy thanh ghi dịch (Shift Register). Truyền thông SPI cơ bản được dựa trên sự ghi dịch của thanh ghi. Đây là một kỹ thuật trao đổi dữ liệu rất đơn giản, chúng ta hãy cùng xem các hoạt cảnh của chúng.

Đầu tiên chúng ta sẽ có 2 thiết bị là Master và Slave, và ngoại vi SPI của Master và Slave là 1 thanh ghi dịch 8-bit. Và MSB của Master là từ A7 về A4, và LSB từ A3 về A0, tương tự Slave MSB từ B7 về B4, và LSB từ B3 về B0. MOSI của Master được nối với MOSI của Slave, tương tự với chân Clock và MISO.

Chúng ta đều biết rằng SPI là giao thức truyền thông nối tiếp đồng bộ, chính vì vậy Clock là bắt buộc đề gửi hoặc đẩy dữ liệu. Trên chân Clock, dữ liệu sẽ được đồng bộ với chúng và được tạo ra trên chân Clock này.

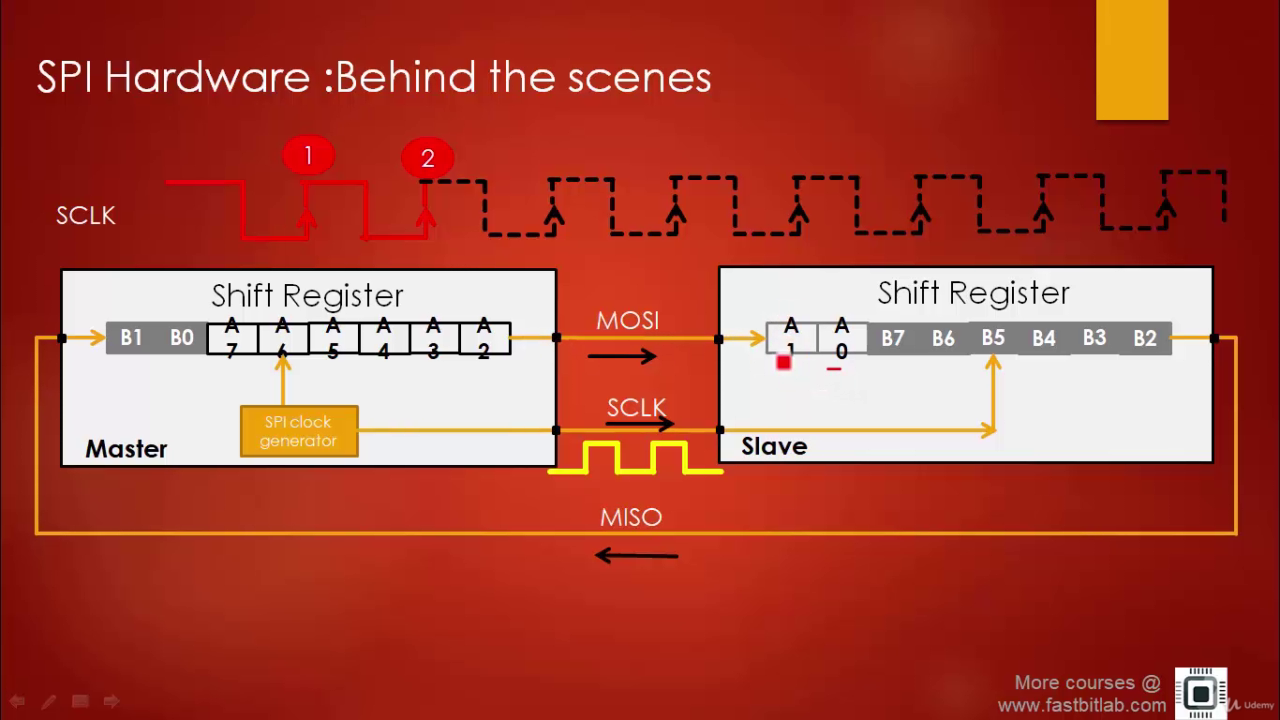
Giả sử chúng ta sẽ truyền dữ liệu từ Master đến Slave, dữ liệu chúng ta có là từ A0 đến A7. Chuyện gì sẽ xảy ra sau một xung đầu tiên.



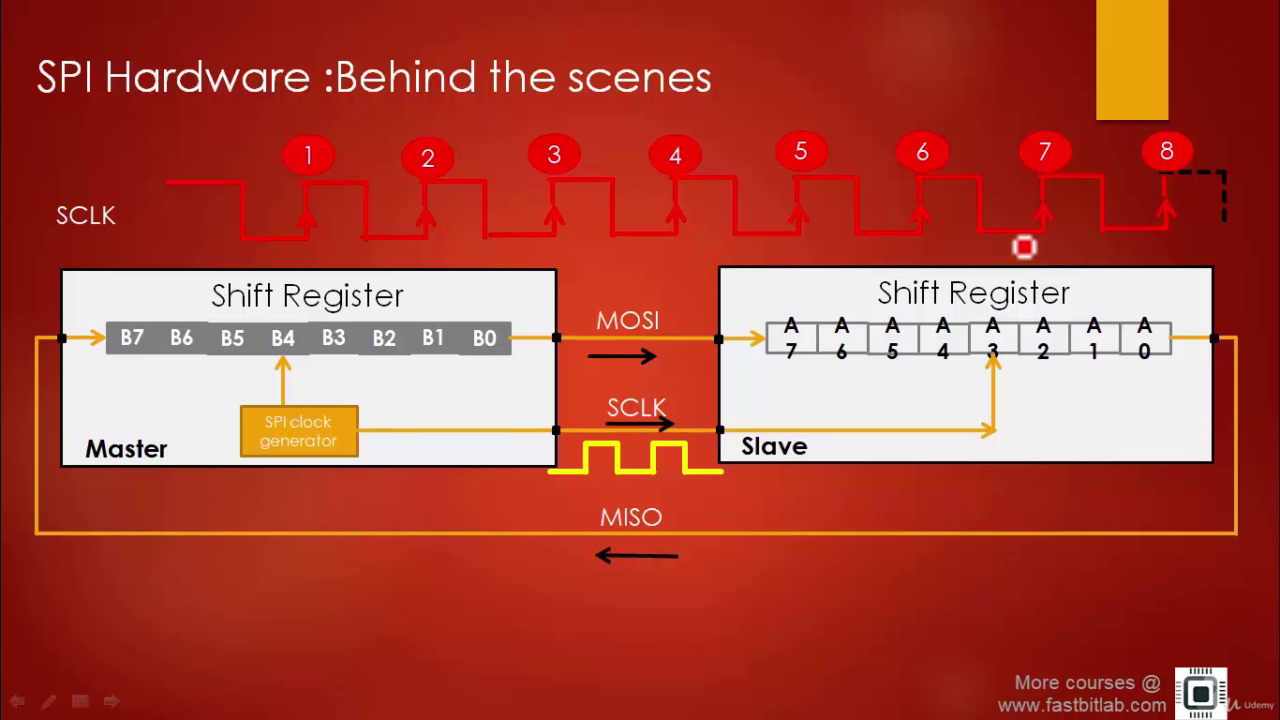
A0 trong LSB của thanh ghi dịch Master đã được chuyển ra khỏi MOSI và đến vị trí MSB của thanh ghi dịch Slave và dữ liệu đã di chuyển được 1 bước. Đồng thời dữ liệu B0 của Slave cũng được chuyển ra khỏi MISO và đến vị trí MSB của thanh ghi dịch Master.

Điều này có nghĩa là, trong mỗi một chu kỳ của Clock, sẽ có sự dịch chuyển giữa hai thanh ghi dịch của Master và Slave. Đó là một nguyên tắc cơ bản của thanh ghi dịch.

Tiếp tục, sau chu kỳ đồng hồ thứ hai. A0 đã được di chuyển và A1 được ghi dịch vào vị trí MSB của Slave, đồng thời B0 được di chuyển và B1 được ghi dịch vào vị trí MSB của Master, cứ như vậy cho đến khi hết 8 chu kỳ Clock.



Sau khi hết chu kỳ Clock thứ 8. Dữ liệu trong thanh ghi dịch của Master và Slave đã được trao đổi với nhau.



Master sẽ luôn nhận được dữ liệu sau khi gửi dữ liệu tới Slave nếu chúng ta nối MISO. Bởi vì bất cứ khi nào master gửi dữ liệu đến thanh ghi dịch của Master thì đều có sự chuyển dữ liệu ngược lại qua đường MISO và đến thanh ghi dịch của Master.

Đây chính là cách mà SPI diễn ra mà chúng ta cần phải nhớ và nắm rõ!

Một điều cần lưu ý nữa chính là, chỉ có Master tạo xung Clock để truyền dữ liệu đồng bộ, trong khi đó Slave thì không.

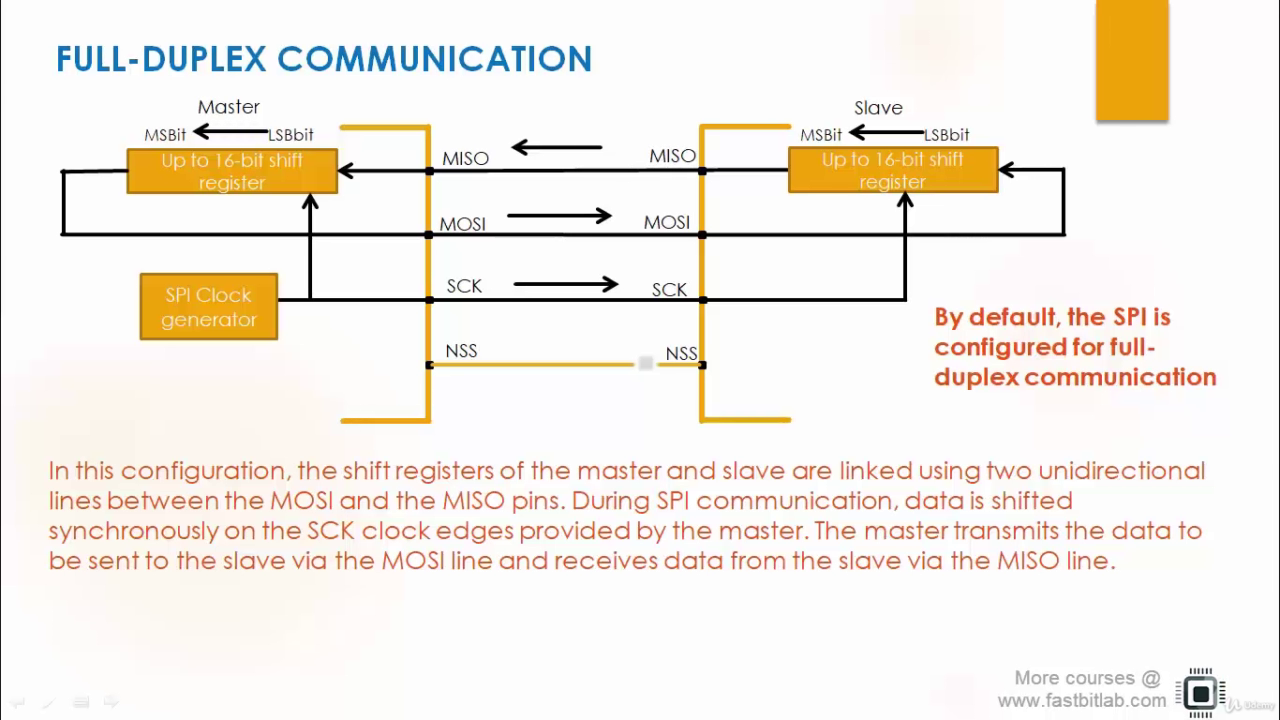
## CẤU HÌNH BUS SPI VÀ SƠ ĐỒ KHỐI SPI STM32

### Cấu hình bus SPI

Bus SPI cho phép vi điều khiển giao tiếp sử dụng với các cấu hình khác nhau, tùy thuộc vào mục tiêu sử dụng và yêu cầu của thiết bị.

#### Giao tiếp song công hoàn toàn (Full-duplex communication)

Phải nhớ rằng SPI là giao tiếp song công hoàn toàn, do đó chúng ta sẽ có hai đường dữ liệu truyền nhận, khi chúng ta thực hiện truyền dữ liệu, thì chúng ta cũng có thể nhận dữ liệu.

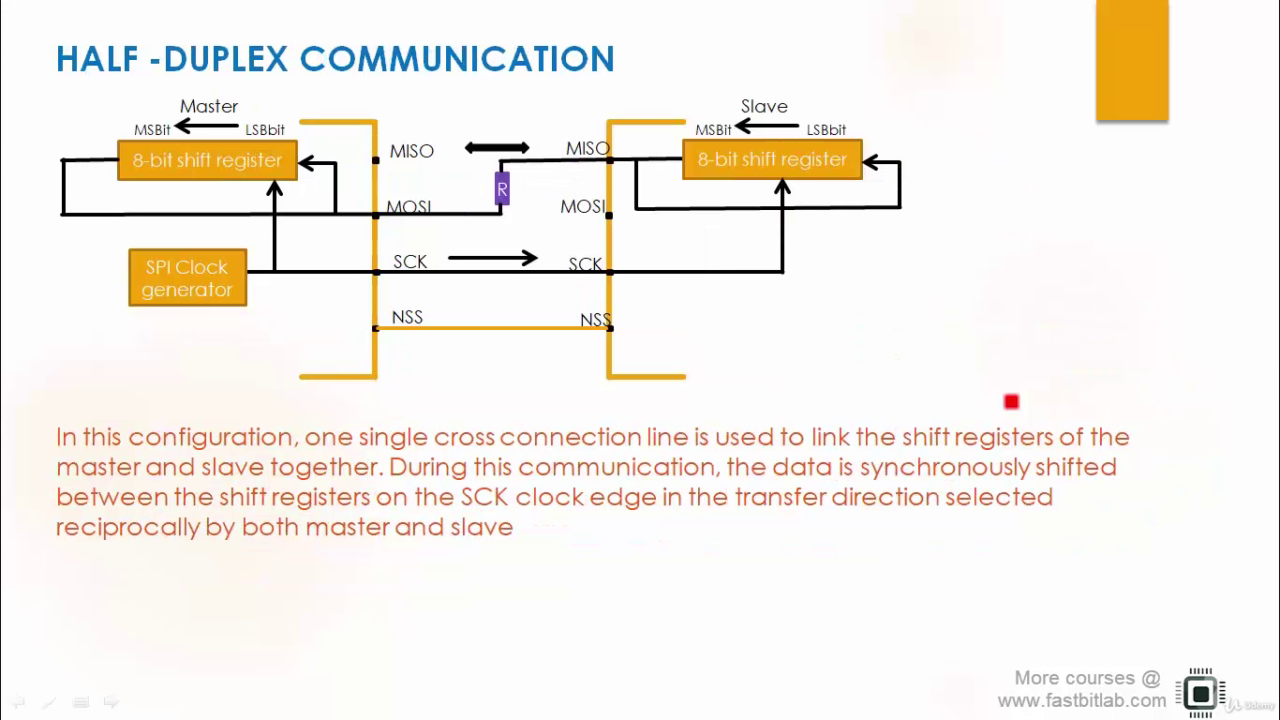


Trong giao tiếp song công hoàn toàn, các thanh ghi dịch của master và slave được liên kết bằng hai đường đơn hướng giữa các chân MOSI và MISO. Trong quá trình giao tiếp SPI, dữ liệu được dịch chuyển đồng bộ trên các cạnh của clock do master cung cấp. Master truyền dữ liệu được gửi đến slave thông qua đường MOSI và nhận dữ liệu từ slave thông qua đường MISO.

Mặc định thì SPI được cấu hình giao tiếp song công.

#### Giao tiếp bán song công (Half-Duplex communication)

Với cách giao tiếp này thì chúng ta có thể tiết kiệm được một vài chân.



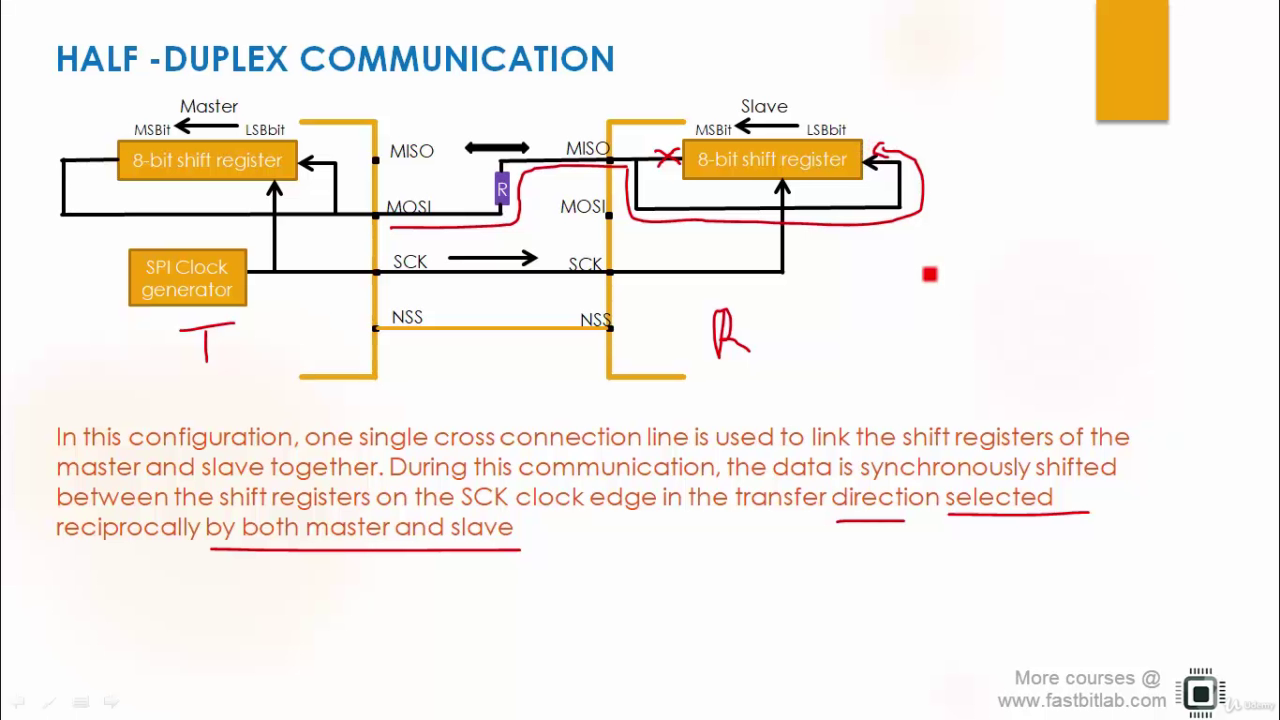
Trong cấu hình này, một đường kết nối chéo duy nhất được sử dụng để liên kết thanh ghi dịch của master và slave với nhau. Vì vậy, trong quá trình giao tiếp này, dữ liệu đồng bộ được dịch chuyển giữa thanh ghi dịch trên sườn Clock SCK theo hướng được truyền được chọn đối với cả master và slave.

Chún ta cần nhớ rằng trong giao tiếp bán song công chỉ có một dòng dữ liệu. Và cách kết nối được giới thiệu như trong hình. MOSI của master phải được nối với MISO của slave và phải có 1 điện trở ở giữa, cụ thể là 1kΩ, … Chân MISO của master và MOSI của slave không được kết nối. Chính vì vậy, Vi điều khiển có thể sử dụng chân đó cho mục đích khác, và SCK là cần thiết cũng như NSS cũng không cần thiết nếu chỉ có 1 slave.

Bất cứ khi nào mà master muốn gửi dữ liệu, thì master phải ở chế độ phát và slave phải ở chế độ nhận, do đó chúng ta cần phải cấu hình trong chương trình. Việc chuyển dữ liệu được xảy ra theo hướng được chọn bởi cả master và slave.

Nếu master là T thì slave phải là R. Ngược lại nếu master ở chế độ nhận thì slave ở chế độ phát.

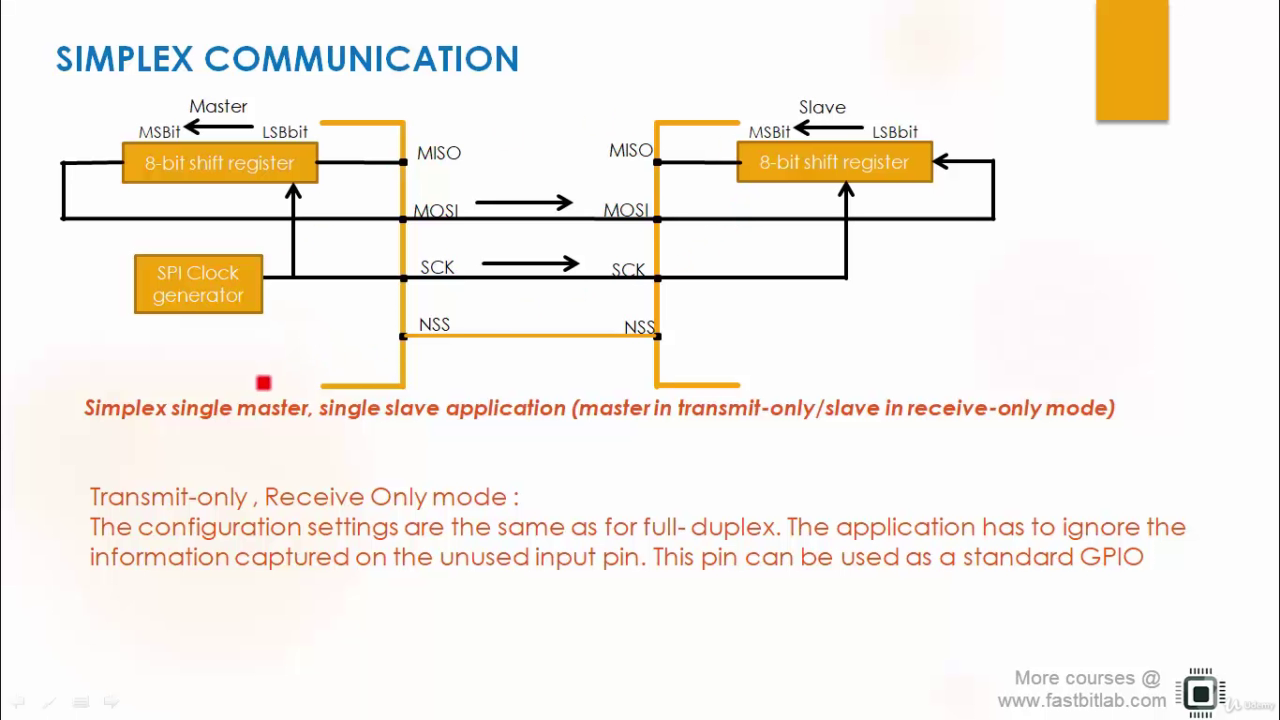
Giả sử master muốn gửi dữ liệu, thì dữ liệu sẽ được chuyển từ MOSI sang MISO theo hướng mũi tên, lúc này phần X sẽ ở trạng thái Hi-Z (trở kháng cao).



Hầu hết vi điều khiển cho phép chúng ta cấu hình SPI ở chế độ bán song công, nhưng có một vài vi điều khiển không cho phép chúng ta cấu hình chúng. Vì vậy chúng ta cần kiểm tra manual của vi điều khiển để xem chúng có hỗ trợ bán song công hay không.

#### Giao tiếp đơn giản (Simplex communication)

Giống như với tên, đây là một chế độ đơn giản, trong chế độ này, chúng ta sẽ có hai loại giao tiếp là, một là chỉ truyền và chỉ nhận. Ví dụ, trong hình, là giao tiếp đơn giản, master chỉ ở chế độ truyền và slave chỉ ở chế độn nhận. Do đó MOSI của hai thiết bị được kết nối với nhau và MISO không được kết nối, lúc này master sẽ không bao giờ nhận bất cứ dữ liệu nào từ slave.



Việc cấu hình giống như với giao tiếp song công hoàn toàn. Ứng dụng sẽ bỏ qua thông tin được ghi lại trên chân không sử dụng. và nó có thể được sử dụng như GPIO tiêu chuẩn. Do đó troing giao tiếp đơn giản, chúng ta có thể cấu hình thiết bị ở chế độ chỉ truyền hoặc chỉ nhận.

### Sơ đồ khối SPI STM32