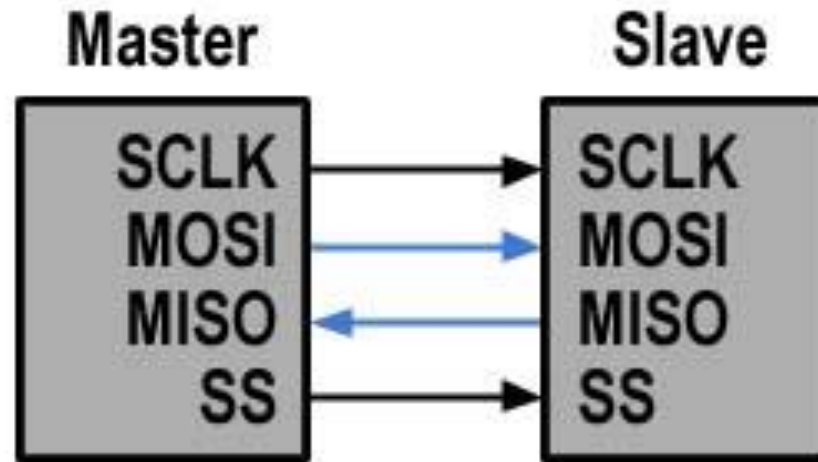


Phụ Lục 6: Giao tiếp SPI

Giới thiệu giao tiếp SPI

- Giao tiếp **SPI** được hãng Motorola giới thiệu giữa năm 1980
- Giao tiếp **SPI** được sử dụng phổ biến để giao tiếp giữa các vi điều khiển, EEPROM, IC thời gian thực, ...
- **SPI** là giao thức nối tiếp đồng bộ đa mục đích, dữ liệu truyền và nhận được diễn ra một cách đồng thời. Một xung clock đồng bộ dùng để dịch và lấy mẫu thông tin trên 2 đường dữ liệu

Sơ đồ nối dây

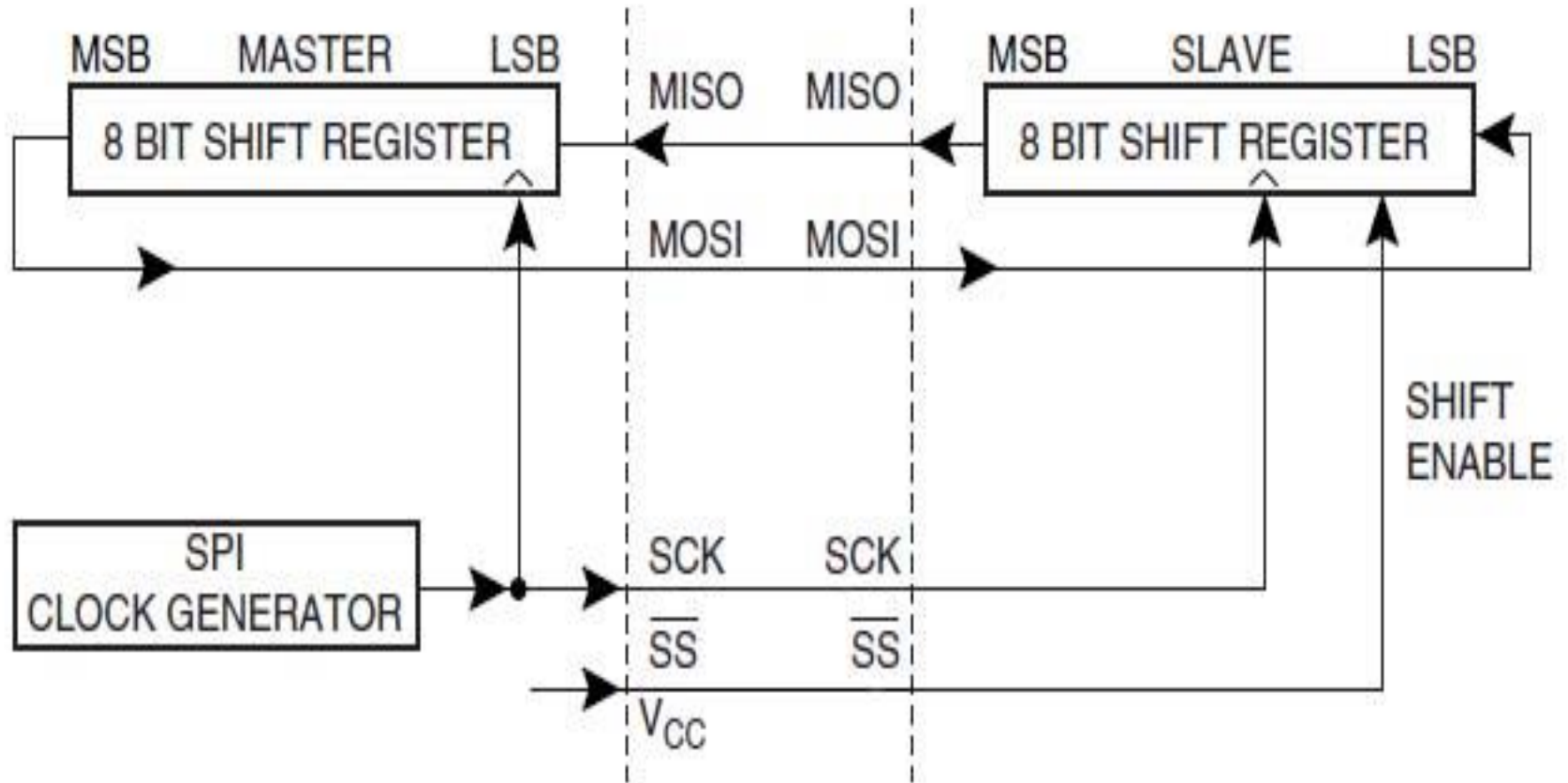


- Giao tiếp SPI sử dụng mối quan hệ Master-Slave
- Do thiết bị Slave không được định địa chỉ nên khi giao tiếp 1 Master và nhiều Slave ta cần đường chọn chip SS để chọn Slave nào cần truyền nhận dữ liệu

Mô tả các chân

- **MOSI** (Master Output Slave Input): Chân này được sử dụng để truyền dữ liệu khi thiết bị là Master và nhận dữ liệu khi cấu hình là Slave
- **MISO** (Master Input Slave Output): Chân này được sử dụng để nhận dữ liệu khi thiết bị là Master và truyền dữ liệu khi cấu hình là Slave
- **SCK**: Chân SCK cấp xung đồng bộ để truyền nhận dữ liệu với một Slave nào đó được chọn
- **SS**: Tín hiệu này được tạo bởi Master để lựa chọn thiết bị Slave muốn truyền nhận. Đối với thiết bị Master chân này được cấu hình là chân xuất (output) và là chân nhập (input) với Slave

Nguyên lý hoạt động (1)



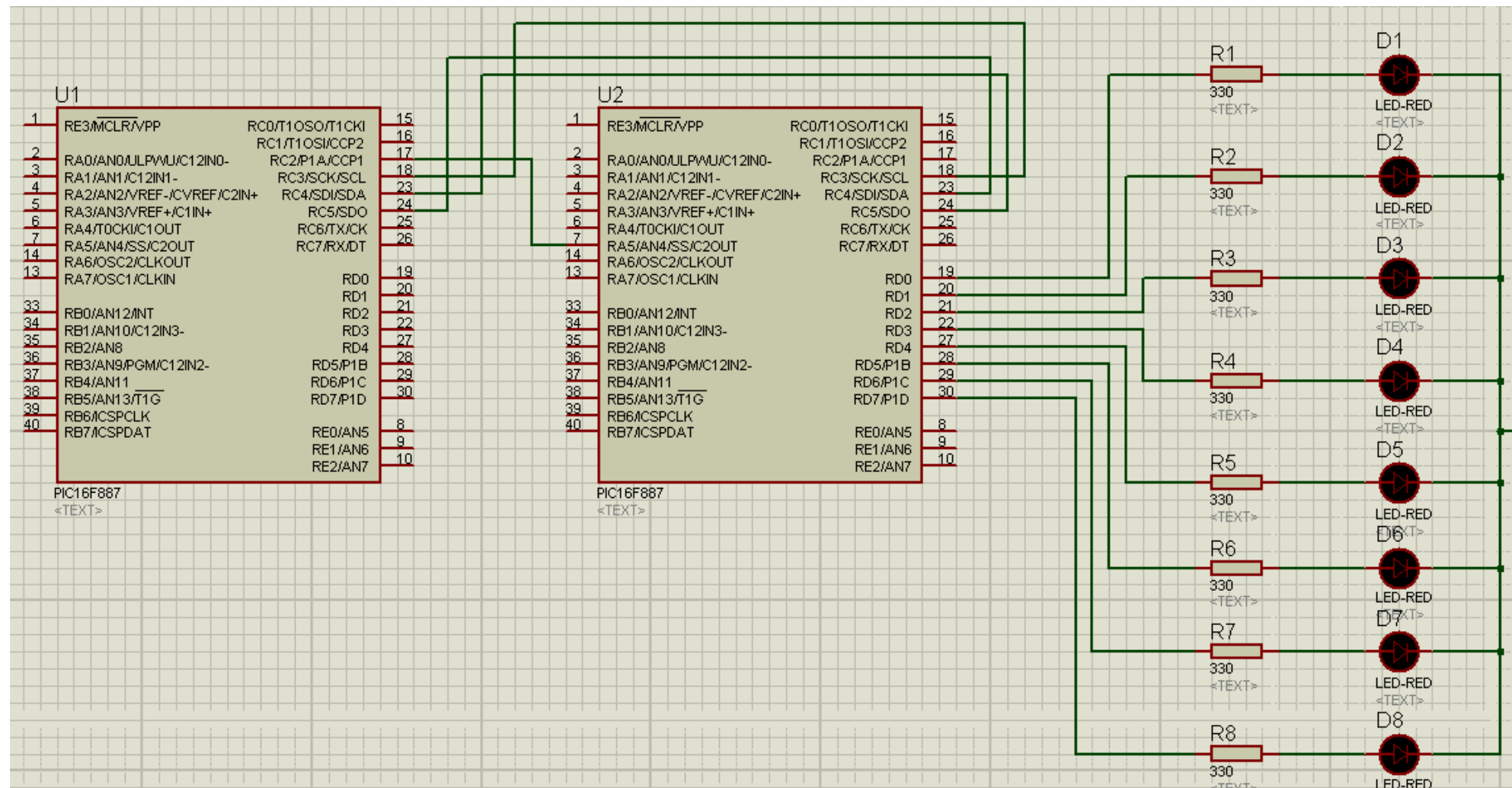
Nguyên lý hoạt động (2)

- Cả Master và Slave đều có thanh ghi dịch nối tiếp ở bên trong
- Thiết bị Master bắt đầu việc trao đổi dữ liệu bằng cách truyền đi một byte vào thanh ghi dịch của nó, sau đó byte dữ liệu sẽ được đưa sang Slave theo đường tín hiệu MOSI (SDI)
- Slave sẽ truyền dữ liệu nằm trong thanh ghi dịch của chính nó ngược trở về Master thông qua đường tín hiệu MISO (SDO)
- Việc đọc và ghi dữ liệu vào Slave diễn ra cùng một lúc nên tốc độ trao đổi dữ liệu diễn ra rất nhanh.

Do đó, giao thức SPI là một giao thức rất có hiệu quả

Ví dụ 1

Thiết kế và lập trình mạch giao tiếp SPI ở chế độ 1 Master - 1 Slave?



Ví dụ 1

```
#include <16F887.h>
```

```
#FUSES NOWDT, HS
```

```
#use delay(clock=20000000)
```

```
int8 data;
```

```
void main()
{
    set_tris_c(0x00);
    output_high(pin_c2);
    data = 0;
    setup_spi( SPI_MASTER | SPI_L_TO_H | SPI_CLK_DIV_16);
    delay_ms(1000);
    while(1)
    {
        output_low(pin_c2); //Chân C2 dùng Select chip
        spi_write(data);
        output_high(pin_c2);
        delay_ms(1000); //Tạo trễ giữa 2 lần truyền
        data = data+1;
        if (data==255)
        {
            data = 0;
        }
    }
}
```


Ví dụ 1

```
#include <16F887.h>
#FUSES NOWDT, HS
#use delay(clock=20000000)
int8 data;

#int_ssp
void ssp_isr(void)
{
    data = spi_read();
    delay_ms(10);
}

void main()
{
    set_tris_d(0x00);
    set_tris_a(0xFF);
    data = 0;

    enable_interrupts(GLOBAL);
    enable_interrupts(int_ssp);

    setup_spi(SPI_SLAVE|SPI_L_TO_H|SPI_CLK_DIV_16);

    while(1)
    {
        output_d(data); //Hien thi ra Port D
    }
}
```

Ví dụ 2

Thiết kế và lập trình mạch giao tiếp SPI ở chế độ 1 Master - 3 Slave theo kiểu kết nối song song nhiều Slave?

