

FPGA 를 이용한 통신 모듈 구현 방법

하경준*, 도영수*, 전재욱*

*성균관대학교 정보통신대학

g1004jay@g.skku.edu, cok2529@g.skku.edu, jwjeon@skku.edu

Method of Implementing Communication Module Using FPGA

Kyoung-Joon Ha*, Young-Soo Do*, Jae-Wook Jeon*

* College of Information & Communication Engineering, Sungkyunkwan University

요 약

통신은 주로 통신의 시작, 데이터 전송, 오류 검사, 통신의 종료 4 가지 과정을 거쳐 이루어진다. 위 4 가지 과정에 따라 통신 모듈의 상태(state)를 분류하고 상태도(state diagram)를 그릴 수 있다. HDL 언어를 사용하여 상태도를 유한 상태 기계(finite-state machine)로 구현함으로써 통신 모듈을 쉽게 구현할 수 있다. 본 논문은 이러한 방법으로 FPGA 에 통신 모듈을 구현하는 방법을 다루고 있다. 나아가, 이 방법을 이용하여 UART 와 SPI 통신 모듈을 구현하는 실험을 소개한다.

1. 서론

FPGA 는 다른 MCU 를 제어하거나 많은 양의 연산량을 처리하는 등 다양한 용도로 활용될 수 있다. 그 중에서도 대표적으로 통신 모듈을 구현하는데 사용될 수 있다. 본 논문은 FPGA 를 이용하여 통신 모듈을 구현하는 방법과 이 방법을 이용하여 실제 시리얼 통신 모듈을 구현하는 실험을 다루고 있다.

2 장에서는 전반적인 통신의 과정을 살펴보고, 3 장에서는 FPGA 에 통신 모듈을 구현하는 방법을 제시한다. 4 장에서는 3 장에서 소개한 방법으로 UART 와 SPI 통신을 구현하는 실험을 소개한다.

2. 통신의 과정

통신은 주로 통신의 시작, 데이터 전송, 오류 검사, 통신의 종료 4 단계를 통해 이루어진다.

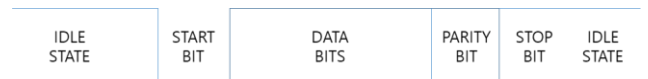
1. 통신의 시작 단계

통신이 이루어지지 않을 때 통신 라인은 유힬 상태(idle state)에 있다. 대부분의 통신은 유힬 상태에서 바로 데이터 전송이 이루어지는 것이 아니라 통신의 시작을 알리는 예비 동작을 취한다. 통신마다 통신의 시작을 알리는 형태가 다르다. <표 1>은 각종 통신에서 통신의 시작 형태를 정리한 것이다.

<표 1> 각종 통신의 시작 형태

통신의 종류	통신의 시작 형태
무선 통신	복잡한 비트의 조합
UART	통신 라인의 전압레벨 감소(start bit)
SPI	SS 라인의 전압레벨 감소
I2C	간단한 프로토콜 위반

(그림 1)은 UART 통신의 프레임이다 [1]. Start bit 는 본격적인 데이터 통신이 이루어지기 전에 통신의 시작을 알리는 역할을 한다.



(그림 1) UART 통신의 프레임

2. 데이터 전송 단계

본격적인 데이터 전송이 이루어지는 단계다. 통신은 데이터 전송 기준에 따라 동기식(synchronous)과 비동기식(asynchronous) 방식으로 나뉜다. 데이터가 송신부와 수신부가 공유하는 클럭 따라 전송되면 동기식 통신이다. 반면, 독립적인 카운터에 의해 전송되면 비동기식 통신이다.

3. 오류 검사 단계

데이터 전송에 오류가 있었는지 검사하는 단계다. 사실 시리얼 통신은 오류 확률이 극히 낮아서 대체로 생략되는 단계다. 다만, UART 통신에서는 parity bit 가

간략하게나마 오류 검사 역할을 한다.

4. 통신의 종료 단계

오류 검사 단계까지 마치고 통신이 유희 상태에 들어가기 전 예비 동작을 취하는 단계다. UART 통신에는 stop bit 가 통신의 종료를 알리는 역할을 한다.

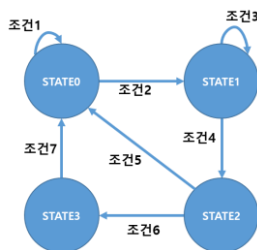
3. FPGA 에 통신 프로토콜 구현하기

유한 상태 기계(finite-state machine)를 이용하면 FPGA 에 통신 모듈을 쉽게 구현할 수 있다. 상태 기계에는 대표적으로 밀리 기계(Mealy machine)와 무어 기계(Moore machine)가 있다 [2]. 본 논문에서는 밀리 기계를 이용할 것이다. 또한, HDL 의 일종인 Verilog-HDL 을 사용할 것이며 소프트웨어 Vivado 를 이용해 시뮬레이션할 것이다 [3].

먼저, FPGA 에 통신 모듈을 구현하기 위한 첫번째 과정은 통신 프로토콜의 프레임(frame)을 보고 통신의 과정을 몇 단계로 나누는 것이다. (그림 1)에 나타나있는 UART 통신에 적용해보면 통신의 과정을 IDLE 단계, START 단계, DATA 단계, PARITY 단계, STOP 단계로 나눌 수 있다.

이후, 통신 프로토콜의 사양(specification)과 타이밍 다이어그램(timing diagram)을 보고 다음 단계로 전환되는 조건을 송신부와 수신부 각각의 입장에서 정리한다. 이 역시 UART 통신에 적용해보자. (그림 1)을 보면, 수신부의 입장에서 IDLE 단계에서 START 단계로 나아가는 조건은 수신 라인이 1 에서 0 으로 떨어지는 것이다.

다음, 지금까지 정리한 내용을 바탕으로 상태도(state diagram)를 그린다. 상태도의 대략적인 모습은 (그림 2)와 같다.



(그림 2) 상태도

정리한 상태도를 바탕으로 HDL 코드를 작성한다. 코드의 형태는 <코드 1>과 같다 [4]. <코드 1>은 밀리 기계를 구현하는 전형적인 코드다.

<코드 1> 밀리 기계를 구현하는 Verilog 코드

```

reg [1:0] state, next_state;
reg out, next_out;
    
```

```

always @ (*)
begin
    case (state)
    S0: begin
        if (A) begin
            next_state = S0;
            next_out = 1'b0;
        end else begin
            next_state = S1;
            next_out = 1'b1;
        end
    end
    S1: begin
        if (B) begin
            next_state = S1;
            next_out = 1'b1;
        end else begin
            next_state = S2;
            next_out = 1'b0;
        end
    end
    default: begin
        next_state = S0;
        next_out = 1'b0;
    end
    endcase
end
    
```

```

always @ (posedge clk or posedge reset)
begin
    if (reset) begin
        state <= S0;
    end else begin
        state <= next_state;
    end
end
    
```

```

always @ (posedge clk or posedge reset)
begin
    if (reset) begin
        out <= 1'b0;
    end else begin
        out <= next_out;
    end
end
    
```

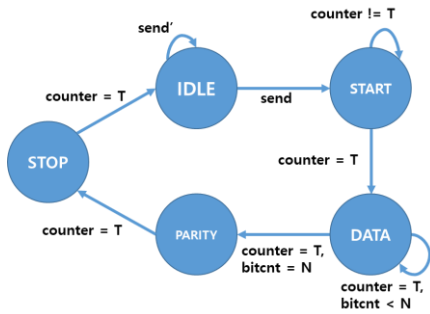
4. UART, SPI 통신의 구현

4 장에서는 3 장에서 제시한 방법으로 UART, SPI 통신 모듈을 구현하는 실험 과정과 결과를 제시한다.

A. UART 통신

(그림 1)에 나타난 것처럼 UART 통신은 IDLE 단계, START 단계, DATA 단계, PARITY 단계, STOP 단계 순으로 진행된다. 나아가, UART 통신은 비동기식이기 때문에 송신부와 수신부가 가지고 있는 카운터에 의해 통신이 이루어진다. 이 정보들을 종합하여 상태도를 그렸다. (그림 3)은 송신 모듈의 상태도이며, <표

2>는 각 상태에 대한 설명이다. (그림 4)는 수신 모듈의 상태도이며, <표 3>은 각 상태에 대한 설명이다.

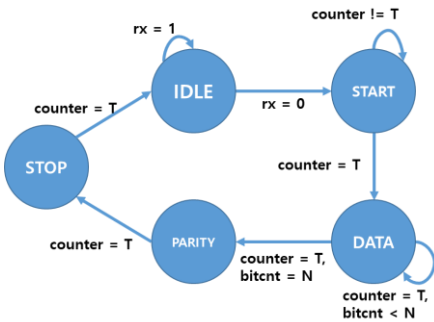


(그림 3) UART 송신 모듈의 상태도

(send = 송신 명령 신호, bitcnt = 송신한 비트의 개수, T = 한 비트당 필요한 클럭 사이클의 수, N = 한 프레임당 데이터의 비트 수)

<표 2> (그림 3) 상태도에 대한 설명

상태명	설명
IDLE	통신 시작 전 유휴 상태다. 송신 명령 신호가 들어오면 유휴 상태에서 벗어난다.
START	송신 모듈이 수신 모듈에게 데이터를 전송하기 전 start bit 를 보낸다. 카운터 값이 T 에 도달하면 데이터 송신을 시작한다.
DATA	데이터를 송신한다. 보 레이트(baud rate)에 따라 T 값을 설정하고 카운터 값이 T 에 도달할 때마다 한 비트씩 보낸다.
PARITY	수신 모듈이 오류 검사를 할 수 있도록 parity bit 를 보낸다.
STOP	데이터 전송이 끝나고 유휴 상태에 들어가기 전 stop bit 를 보낸다.



(그림 4) UART 통신 수신 모듈의 상태도

(rx = 수신된 비트 값, T = 한 비트당 필요한 클럭 사이클의 수, N = 한 프레임당 데이터의 비트 수)

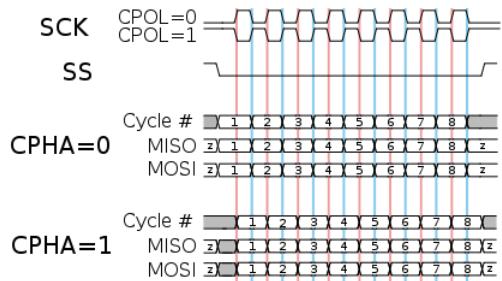
<표 3> (그림 4) 상태도에 대한 설명

상태명	설명
IDLE	유휴 상태다. 송신 모듈로부터 start bit 가 수신되면 유휴 상태에서 벗어난다.
START	송신 모듈로부터 start bit 를 받는 상태다.
DATA	송신 모듈로부터 데이터를 받는 상태다.
PARITY	송신 모듈로부터 parity bit 를 받는 상태다.
STOP	송신 모듈로부터 stop bit 를 받는 상태다. 이후 유휴 상태에 들어가게 된다.

(그림 3, 4)의 상태도를 <코드 1>의 형태로 작성하였다. 이후 작성한 코드에 대한 시뮬레이션을 진행하였다. (그림 5)는 시뮬레이션으로 나온 입출력 신호들의 파형(waveform)이다. 8 비트 데이터(0xA3)를 송수신하는 상황을 시뮬레이션한 것이다. 송신 모듈의 경우 LSB 부터 차례대로 송신하는 것을 관찰할 수 있다. 수신 모듈의 경우 rx_data 에 0xA3 이 저장되는 것을 관찰할 수 있다.

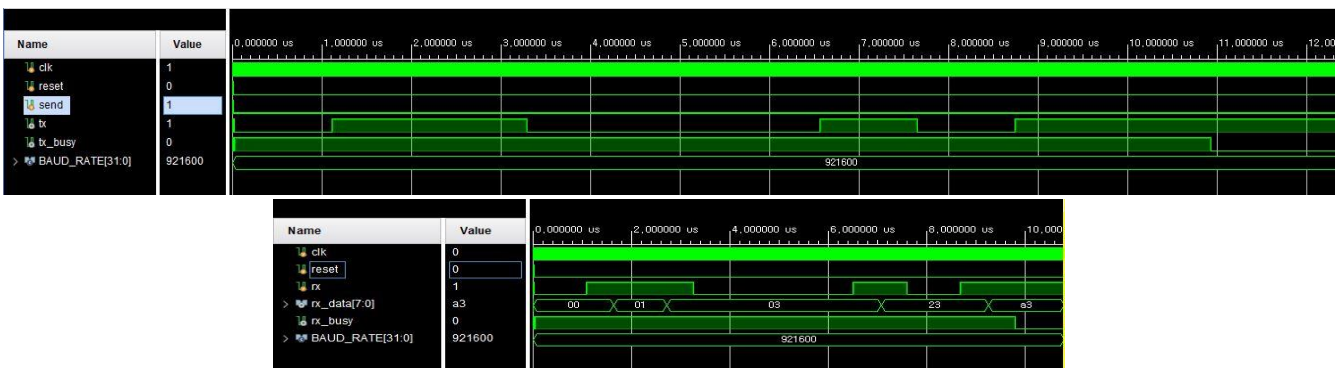
B. SPI 통신

(그림 6)은 SPI 통신의 타이밍 다이어그램(timing diagram)이다 [5]. SPI 통신 역시 2 장에서 서술한 통신의 과정대로 진행된다. 다만 오류 검사 단계가 생략된다. 즉, 통신의 시작, 데이터 전송, 통신의 종료 단계 순으로 진행된다.<표 4>는 각 단계의 구체적인 프로토콜을 나타낸 것이다.

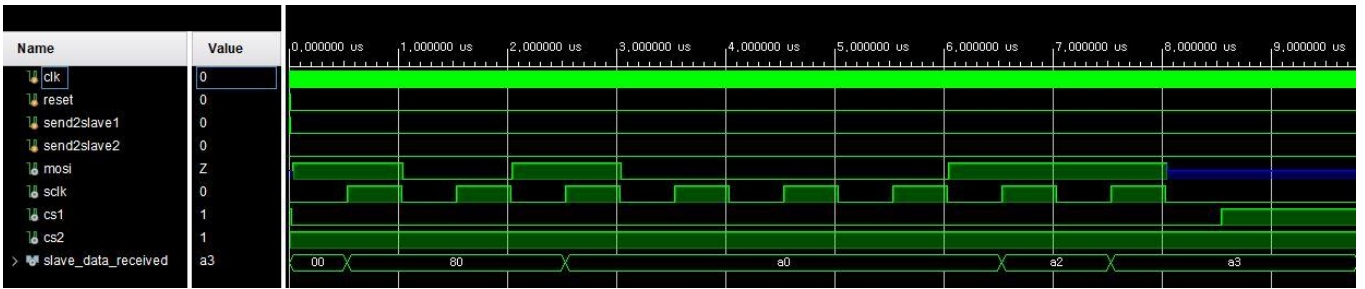


(그림 6) SPI 통신의 타이밍 다이어그램

라이선스 : free license



(그림 5) 위 : UART 송신 모듈 시뮬레이션 결과 / 아래 : UART 수신 모듈의 시뮬레이션 결과

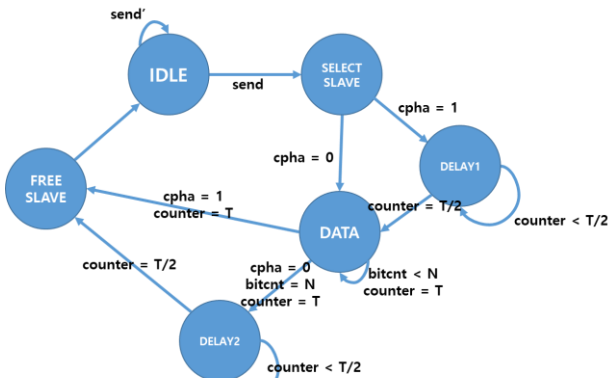


(그림 8) SPI 통신 시뮬레이션 결과

<표 4> SPI 통신의 프로토콜

통신의 단계	프로토콜
통신의 시작	SS 라인의 전압레벨 감소
데이터 전송	마스터가 생성하는 클럭에 맞추어 비트 전송 및 수신
통신의 종료	SS 라인의 전압레벨 증가

SPI 통신은 마스터와 슬레이브 사이의 통신으로 이루어진다. 나아가, 동기식 통신이기 때문에 마스터가 생성하는 클럭을 기준으로 데이터 전송이 이루어진다. (그림 7)은 저자가 구상한 마스터의 상태도이며, <표 5>는 각 상태에 대한 설명이다.



(그림 7) SPI 마스터의 상태도

(send = 송신 명령 신호, bitcnt = 송신한 비트의 개수, T = SCK 한 주기/시스템 클럭 한 주기, N = 한 프레임당 데이터의 비트 수)

<표 5> (그림 7) 상태도에 대한 설명

통신의 단계	프로토콜
IDLE	통신 시작 전 유휴 상태다. 송신 명령 신호가 들어오면 유휴 상태에서 벗어난다.
SELECT SLAVE	마스터가 통신하고자 하는 슬레이브에 해당하는 SS 라인의 전압레벨을 끌어내린다.
DELAY1	CPHA = 1 일 때 SS 라인의 전압레벨 감소 후 데이터 전송이 이루어지기 전 지연이 있다.
DATA	데이터를 송신한다. 카운터 값이 T에 도달할 때마다 한 비트씩 보낸다.
DELAY2	CPHA = 0 일 때 데이터 전송 단계가 끝나고 SS 라인의 전압레벨이 다시 올라가기 전 지연이 있다.
FREE SLAVE	마스터가 SS 라인 제어를 중단한다.

(그림 7)의 상태도를 <코드 1>의 형태로 작성하였다. 이후 작성한 코드에 대한 시뮬레이션을 진행하였다. 시뮬레이션 결과는 (그림 8)과 같으며, 8 비트 데이터(0xA3)를 송수신하는 상황을 시뮬레이션한 것이다.

5. 결론

3 장에서는 FPGA 에 통신 모듈을 구현하는 방법을 제시하였다. 4 장에서는 이 방법을 이용하여 UART 와 SPI 통신 모듈을 구현하는 실험의 과정과 결과를 소개하였다. 본 논문에서 제시한 방법을 이용하면 FPGA 에 쉽게 통신 모듈을 구현할 수 있음을 보였다.

Acknowledgement This work was supported in part by the Ministry of Science and ICT (MSIP), South Korea, through the G-ITRC Support Program supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2018- 20150-00742.

참고문헌

- [1] Peña, E., & Legaspi, M. G. (2020, December). UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices.
- [2] Charles Roth. (2015, February) Digital Systems Design Using Verilog International Edition.
- [3] Xilinx. Vivado Design Suite - HLx Edition 2020. 3. Available : <https://www.xilinx.com/support/download>
- [4] Golson, S. (1998). State Machine Design Techniques for Verilog and VHDL.
- [5] Cburnett. (2006, December). SPI timing diagram. Wikimedia Commons.