

UART 인터페이스의 HDL 설계

김병준* · 민태훈* · 손승일**

*한신대학교 공과대학 정보통신학과

HDL Design of UART Interface

Byung-jun Kim* · Tae-hoon Min* · Seung-il Sohn**

*Dept. of Information and Communication Hanshin University

E-mail : jjun258@nate.com

요 약

인터넷 사용량의 증가와 네트워크 망 기술의 발달로 인해 데이터는 대용량화 되어지는 반면 휴대기기는 고속화와 소형화가 되어 지면서 직렬 포트를 이용한 외부 장치들과 데이터 송·수신이 가능한 인터페이스가 요구되고 있다. 본 논문에서는 16바이트의 버퍼링을 제공하는 UART 인터페이스를 HDL로 설계하여 내부 모듈과 외부 장치들 간의 데이터 전송이 가능하도록 하였고, Modelsim 6.1로 시뮬레이션 하였다.

키워드

UART, MCU,, Divsor Latch, Interrupt

I. 서 론

정보화 시대의 발전에 따라 인터넷 환경과 통신 기기의 기술도 급속도로 발전하고 있다. 특히 인터넷의 사용량 증가와 그에 따른 네트워크 대역폭의 증가로 인해 데이터의 대용량화와 데이터 전송의 고속화가 이루어지고 있는 반면에 휴대기기와 주변기기들은 고속화와 대용량화, 경량화 되어 지고 있다. 이에 따라 직렬 통신을 사용하여 적은 전송 라인으로도 고속의 데이터를 송·수신 할 수 있는 인터페이스가 요구 되고 있다[1].

본 논문에서는 16바이트의 버퍼링을 제공하는 UART 인터페이스를 HDL로 설계하여 내부 모듈과 외부 장치들 간의 데이터 전송이 가능하도록 하였고, 8비트 데이터 송/수신을 기반으로 시뮬레이션 하였다.

II. 본 론

2.1 UART 개요

UART(Universal asynchronous receiver/transmitter)는 일반적으로 컴퓨터나 주변 기기에서 직렬 통신을 위해 사용되고 있는 범용 비동기식 송수신 장치이다. 컴퓨터의 병렬 회로를 통해 8비트 혹은 32비트의 데이터를 직렬 데이터로 변환하여 외부로 전송하고, 직렬 데이터를 병렬 데이터

로 변환하여 전송한다. 본 UART에서 송·수신되는 데이터 프레임은 1개의 start 비트, 데이터 비트, 패리티 비트, 1~2개의 stop 비트로 구성되어 있고 패리티 비트의 경우 내부의 레지스터 컨피규레이션에 따라 생략될 수 있다. 프레임의 구조는 그림 1과 같다[2][3].



그림 1. 프레임 구조

그림 2는 UART의 전체 블록도로 MCU와 UART간의 입·출력 신호들과 UART 내부 모듈에 대해 나타내고 있다. UART는 비동기 통신이기 때문에 송신 비트율과 수신 비트율의 차이에 의해 발생할 수 있는 데이터 오류를 방지하기 위해 클럭 신호를 16분주하여 사용한다. 송신부의 FIFO와 수신부의 FIFO는 각각 16바이트의 버퍼 레지스터로 CPU와 UART의 속도 차이를 해결하기 위해 사용된다. 송신부 쉬프트 레지스터는 FIFO로부터 직렬 데이터를 읽어와 클럭 신호에 의해 쉬프트 되고 start 비트, 패리티 비트, stop 비트가 추가되어 단일 비트 스트림으로 전송하게 된다. 수신 쉬프트 레지스터의 경우 직렬 데이터의 start 비트를 검출하고 데이터의 stop비트

를 검출한 뒤에 패리티 비트, 프레임 에러,오버런 등의 에러를 검사한 뒤에 수신부의 FIFO로 데이터를 보내게 된다[4].

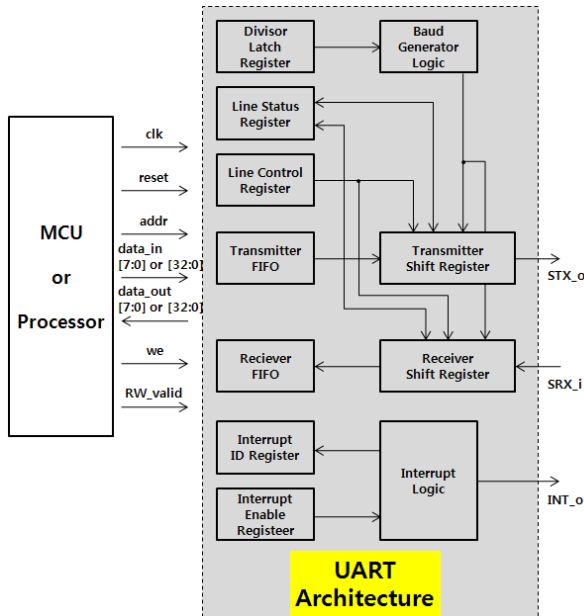


그림 2. UART 전체 블록도

2.2 UART 모듈 설계

2.2.1 초기화

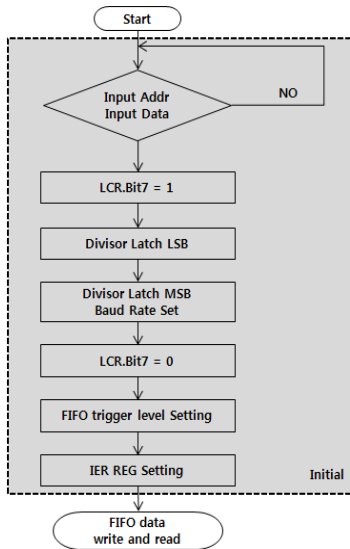


그림 3. UART 초기화 순서도

UART의 초기화 단계에서는 `addr_i`와 `data_i`의 값을 읽어 들인다. `addr_i`값은 각 레지스터를 선택하고, `data_i`값은 레지스터 값을 설정한다. Line Control Register의 bit 7번째를 1로 설정하여 Divisor Latch로 접근하여 Divisor Latch의

LSB와 MSB를 통하여 Baud Rate를 만들어준다. 다시 Divisor Latch의 접근을 불허하기 위해 Line Control Register의 bit 7번째를 0으로 설정하여 전송 엔진을 시작하고, 데이터를 주고받는다. FIFO Trigger Level를 설정하여 시스템이 충분히 빠르게 응답한다면 14bytes로 설정해야 시스템에 인터럽트가 더 적게 생성된다. 다음으로 인터럽트에 적절한 비트를 설정하여 원하는 인터럽트를 활성화하면, FIFO를 통해 데이터를 읽고 쓸 수 있다[5].

2.2.2 송신부

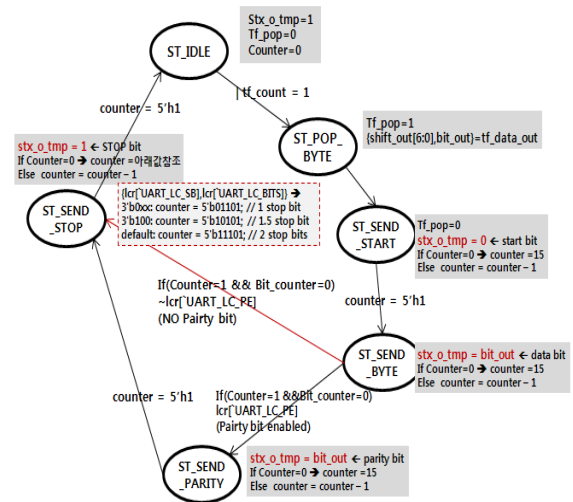


그림 4. 송신부 상태 머신

송신부의 상태머신은 Enable신호가 활성화되어야 동작한다. `tf_count = 0` 이면 `ST_IDLE`에서 `ST_POP_BYTE` 상태로 가며, `ST_POP_BYTE`상태에서는 `lcr[1:0]`에 대해 패리티를 계산하여 저장한다. 다음 상태 머신에서는 `start bit`, `data`, `parity`, `stop bit` 순으로 `stx_o_tmp`에 저장한다. 단, 설정 값에 따라 `parity bit`의 존재 유무 및 설정이 달라질 수 있으며, `counter`값에 따라 `stop bit`의 크기가 달라진다. Break Control bit를 통해 `stx_o_tmp`를 `STX_PAD_O`로 데이터를 전송한다.

2.2.3 수신부

수신부는 수신부와 마찬가지로 Enable신호가 활성화되어야 동작한다. Enable신호가 활성화되면, `srx_pad_i` 값의 `start bit`값을 검출하고, Line Control Register[1:0]에서의 비트수 파악 후 입력받은 데이터를 저장한다. 데이터 확인 후 `LCR[3]`을 통해 `parity bit`의 존재를 파악하여 `parity bit`가 존재한다면 `parity`를 체크하여 올바르게 전송되었는지 확인하고, `stop bit`를 검출한다. 마지막으로 `praming` 에러와 `break` 에러 확인 후 이상이 없다면 데이터를 출력시킨다.

2.2.4 인터럽트

인터럽트에는 Receiver Line Status, Receiver Data Available, Timeout Indication, Transmitter Holding Register empty, Modem Status 가 있다. Receiver Line Status의 경우 패리티, 오버런 또는 프레임 에러나 Break 인터럽트가 발생할 경우 생성되고, Line Status Register를 읽을 경우 인터럽트가 제거 된다. Receiver Data Available은 FIFO가 트리거 레벨에 도달할 경우 생성되고, FIFO가 트리거 레벨 아래로 떨어질 경우 제거된다. Timeout Indication은 하나 이상의 데이터가 FIFO에 있으며 4 개의 character time 동안 FIFO에 데이터의 입출력이 없을 경우 생성되고, 수신 버퍼 레지스터를 읽을 경우 제거 된다. Transmitter Holding Register empty은 Transmitter Holding Register 가 비었을 경우 발생하고, Transmitter Holding Register에 데이터를 쓰거나 IIR을 읽을 경우 제거된다. Modem Status은 CTS, DSR, RI, DCD가 발생할 경우 생성되고, Modem Status Register 를 읽을 경우 제거 된다.

각각의 인터럽트에는 우선순위가 존재한다. Receiver Line Status가 가장 높고, Receiver Data Available과 Timeout Indication이 두 번째, Transmitter Holding Register empty가 세 번째, Modem Status가 가장 낮은 순서이다.

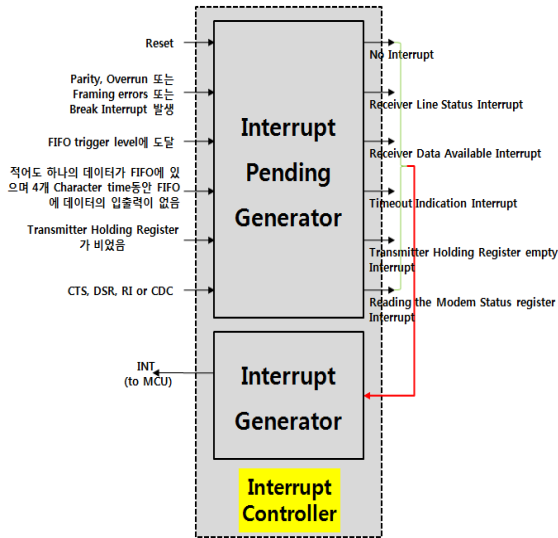


그림 5. 인터럽트 블록도

III. 설계 모듈의 시뮬레이션

3.1 송신부 모듈의 파형분석

HDL을 이용하여 설계한 UART를 시뮬레이션 하였다. 테스트벤치를 작성하여 그림6과 같은 파형과 같은 결과가 나왔다. 송신부의 앞단에서는 초기화 단계로써 그림3과 같은 초기화 순서도

따랐으며, 초기화 후 데이터를 receive buffer에 aa, 02~0f까지 주어 가장 높은 인터럽트 상태를 읽어 그에 해당하는 서비스를 검증하였다.

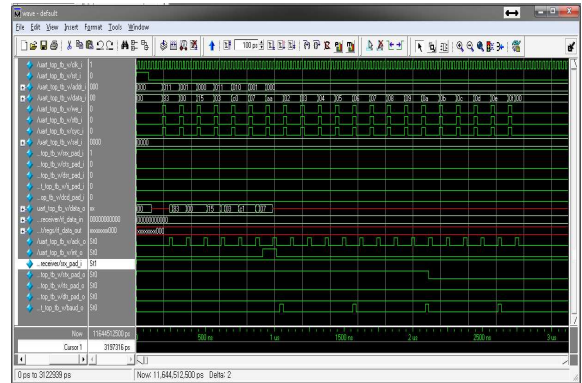


그림 6. 송신부 모듈의 파형

3.2 수신부 모듈의 파형분석

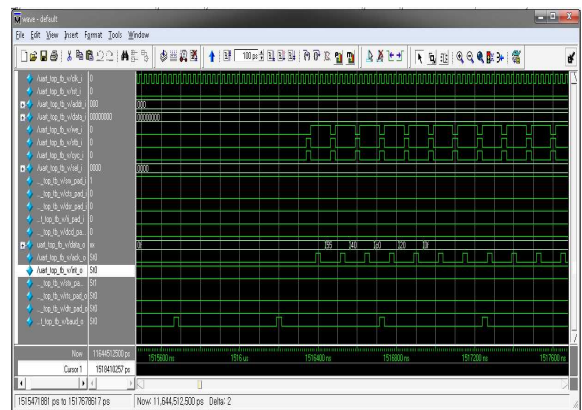


그림 7. 수신부 모듈의 파형

수신부의 경우 데이터의 포맷은 start bit, data, parity, stop bit의 형식을 띤다. 이 형식에 맞게 stx_pad_i에 입력한다. 입력된 값은 수신부의 상태 머신을 따라 진행되어 data_o의 출력이 나온다. data_o의 출력은 start, parity, stop bit를 뺀 data 값만이 출력된다.

IV. 결 론

UART는 거의 모든 마이크로프로세서에 들어 있는 마이크로칩으로 8비트 혹은 32비트의 병렬 데이터를 하나의 직렬 데이터로 바꾸어 전송하거나, 직렬 데이터를 컴퓨터가 처리할 수 있는 병렬 데이터로 바꾸어 주는 인터페이스이다.

본 논문에서는 실제 송·수신 가능한 모듈을 설계하였고, 8비트의 데이터와 32비트의 데이터

송·수신 모두를 지원하도록 설계 하였다. 또한 최대 40Mbps의 전송속도를 지원한다.

참고문헌

- [1] 김성중, 손승일 "UART 디바이스의 VHDL 설계" 한국해양정보통신학회 춘계종합학술대회2004
- [2] 홍건표 저 "개발자들을 위한 ARM 프로세서" 2006.
- [3] 차영배 저 "MSP430F1611" 2009
- [4] UART IP Core Specification, Jacob Gorban. 8, 2002
- [5] J. Smith "HDL Chip Design" 1996

저자소개



김 병 준(Byung-Jun Kim)

2012년 한신대학교 정보통신학과(학사)
2012년 ~ 현재 한신대학교 정보통신학과(석사)

※ 관심분야 : ATM 통신 및 보안, ASIC 설계



민 태 훈(Tae-Hoon Min)

2011년 한신대학교 정보통신학과(학사)
2011년 ~ 현재 한신대학교 정보통신학과(석사)

※ 관심분야 : ATM 통신 및 보안, ASIC 설계



손 승 일(Seung-II Sohn)

1989년 연세대학교 전자공학과(학사)
1991년 연세대학교 대학원 전자공학과(석사)
1998년 연세대학교 대학원 전자공학과(박사)

1998~2002년 호남대학교 컴퓨터공학과 조교수
2002년 ~ 현재 한신대학교 정보통신학과 조교수

* 관심분야 : ATM 통신 및 보안, ASIC 설계