

MAC에 적용 가능한 Receive FIFO블록의 설계

이동훈*, 손승일*, 이범철**

한신대 정보통신학과*, ETRI**

The Design of RX FIFO Block for MAC

Dong Hun Lee*, Seung Il Sonh

*HanShin University, **Electronics and Telecommunications Research Institute

donghun2@hs.ac.kr, saisonh@hanshin.ac.kr

요 약

MAC RX FIFO은 10Gbps전송 전송속도에서 제공하는 FIFO모듈이다. 10Gbps이상의 전송속도에서는 전송부, 수신부모두 양방향 전송신가능한 Full Duplex(전이중)방식을 사용한다. 기존 FIFO의 임시 버퍼기능 뿐만아니라 프레임 흐름 컨트롤 블록을 적용하여 전송간의 프레임 흐름제어 기능을 수행한다. 하위계층에서 MAC으로부터 전송되는 데이터는 64비트와 데이터 유효 정보를 가진 8비트 데이터 유효 정보비트를 가진다. 이렇게 전송되는 데이터는 MAC RX FIFO에 프레임단위로 저장되어 프레임 간의 구분정보 Codeword프레임을 확인하여 새프레임 데이터를 확인한다. 사용자계층에는 데이터 128비트와 유효 정보비트 16비트로 사용자계층에 전송한다.

본 논문에서는 10G전송속도를 갖는 MAC RX FIFO을 설계한다. VHDL언어를 사용하였고, ModelSim5.6a로 시뮬레이션하여 파형분석과 타이밍 분석하여 정상적인 동작을 확인한다. MAC RX FIFO는 10Gbps전송속도에서 요구되어지는 모듈에서 Flow Control, Pause프레임기능을 갖는 모듈에 적용되어 사용가능할 것으로 사료된다.

1. 서 론

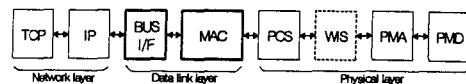
인터넷의 급속적인 발전으로 사용자의 요구에 상응하는 컨텐츠 질의향상으로 인터넷의 속도가 고속화 되어가고 있다. 고화질의 실시간 동영상이나 고용량의 데이터를 고속으로 사용자가 요구에 맞게 이용할 수 있게 되었다. 현재 10Gbps 이더넷망이 관공서, 학교망에 구축되어 사용되고있다. 그러나 그동안 ATM 등의 전송기술에 의해 관심밖에 있던 이더넷이기 때문에, 국내에서 축적된 기술도 미약할 뿐만 아니라 관련된 기술자도 없어, 대부분의 핵심기술들이 외국기술이나 제품을 도입하여 조립하는 실정이다. 네트워크망의 발전 속도를 현재 구축된 인터페이스 디바이스들은 이를 충족시키지 못하고, 네트워크망 내부에서 병목현상을 일으키고 있다. 이를 해결하기 위해 고속의 대역폭을 지원하는 인터페이스 기술이 발표되고 있다. MAC에 적용 가능한 RX FIFO인터페이스는 IEEE 802.3ae-2002 Spec에서 권장하는 규격이다[1][2].

본 연구에서는 10Gbps 이더넷 MAC의 데이터 링크 계층과 물리계층간의 프레임 전송을 위해 임시 버퍼기능 수행할 FIFO을 두어 프레임의 전송간의 흐름제어와 전송중에 Pause프레임의 정보를 보내 전송간의 올바른 프레임이 전달 하고자 한다.

제안한 블록을 VHDL언어를 이용하여 설계하였고, ModelSim5.6a로 시뮬레이션하여 최적화된 설계를 하였다[4].

2. MAC용 RX FIFO의 개요

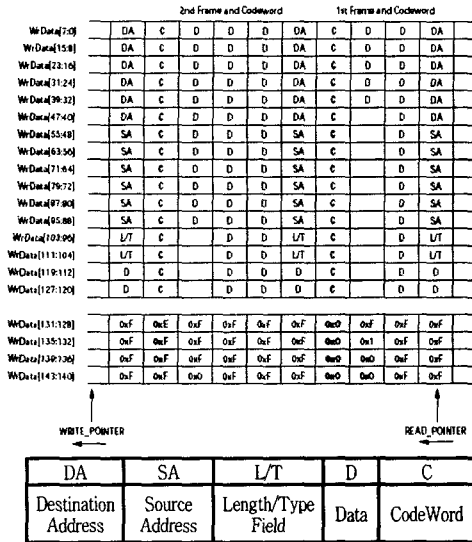
RX FIFO는 하위계층의 MAC으로부터 64비트 데이터와 8비트 데이터 유효 정보비트를 갖고 RX FIFO에 일시적으로 버퍼링 하기위해 사용된다. 그림 1은 10Gbps이더넷의 아키텍처를 보여주고 있다.



[그림 1] 10Gbps 이더넷 아키텍처

그림에서 데이터 링크계층에서 RX FIFO가 위치하여 상위 사용자계층과 하위계층 물리계층간의 10Gbps 전송속도로 전송한다. 프레임이 Cut-through방식으로 DA(Destination Address)정보를 읽어 즉시 프레임을 전송하는 방식을 사용하였고, Codeword비트를 체크하여 새프레임간의 구분하

여 전달 할 수 있게 한다. 그림 2는 데이터 읽기, 쓰기흐름과 프레임 형식을 보여주고 있다[4][5].



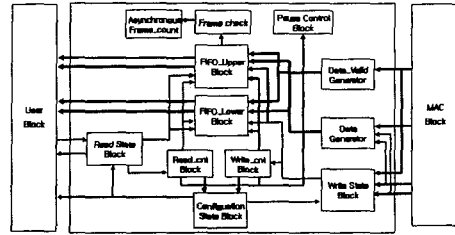
[그림 2] FIFO 데이터 개념도

3. RX FIFO의 설계

하위계층에서 전송되는 프레임 데이터 크기는 64비트와 데이터 유효 정보비트를 8비트를 포함하여 전송된다. 전송된 데이터는 RX FIFO에 유효 정보비트를 체크하여 올바른 데이터가 FIFO입시버퍼에 하위 FIFO, 상위 FIFO블록에 각각 64비트씩, 8비트 정보가 버퍼링되며, 사용자계층에 128비트 데이터와 16비트 유효 정보데이터를 전송한다.

3.1 RX FIFO 제안한 블록도

RX FIFO블록도는 하위계층 MAC블록과 상위계층 사용자블록 사이에 RX FIFO블록이 위치하여 프레임 단위 데이터 전송역할과, GOOD프레임, BAD프레임정보를 함께 전송한다. 세부블록도에서 DATA_Generator블록은 64비트 데이터 전송블록이다. DATA_VALID_Generator블록은 8비트 유효 데이터 정보를 전송한다. WRITE STATE블록은 MAC으로부터 전송되는 Data_Valid 비트가 '0'에서 '1'로 전환할 때 Write_en (FIFO_Upper_en, FIFO_Low_en)정보가 활성화 된다. Write_cnt 블록, Read_cnt블록은 RX FIFO depth유효 카운트 수를 Write_en, Read_en수에 맞게 증가, 감소한다. Configuration_State블록은 RX FIFO의 제어상태 정보를 가지고 있다. Almost_full, Full, Almost_empty, Empty상태정보를 사용자쪽에 전달한다. 그림 3은 제안한 RX FIFO의 블록도를 보여주고 있다.

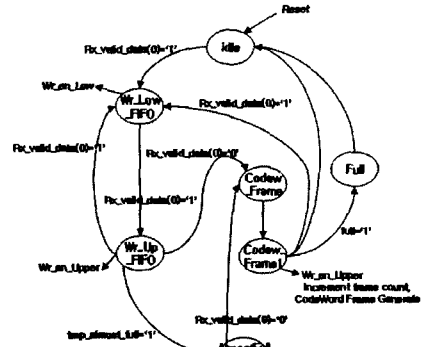


[그림 3] RX FIFO블록도

Pause컨트롤 블록은 Threshold값으로부터 유도된 신호를 사용하여 Nearly_rx_full, Nearly_rx_empty을 활성화 되면 Pause프레임을 MAC에 전송한다. 비동기 프레임 카운트블록은 RX FIFO에 프레임 구분 Codeword비트를 확인하여 Data_valid='0'이면 체크하여 RX FIFO에 Codeword프레임을 생성하여, RX FIFO에 프레임카운트를 증가, 감소한다. Read_State 블록은 사용자쪽에서 Read_en을 활성화하여 데이터 128비트 크기와 16비트 유효비트를 읽어온다[4][5].

3.2 RX FIFO의 쓰기 FSM

RX FIFO쓰기시 상태를 나타내고 있다. Reset이 발생상태면 Idle상태에 머무른다. Rx_valid_data(0)가 '0'에서 '1'로 Toggle될때 먼저 Low FIFO의 Wr_en_low신호가 활성화된다. 다시 Rx_valid_data(0)정보가 '0'에서 '1'로 Toggle되면 두 번째 UP_FIFO의 Wr_en_upper신호가 활성화 된다. Wr_up_fifo 상태에서 Rx_valid_data(0)가 '0'값이면 Code_frame상태로 천이하고 RX FIFO에 128비트 Codeword프레임이 쓰여지고 하나의 프레임을 쓰여지게된다. 그림 4는 RX FIFO의 쓰기 상태를 나타낸다.



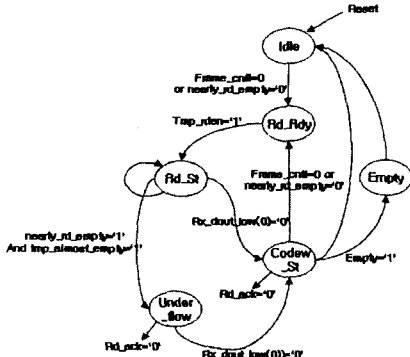
[그림 4] TX FIFO의 쓰기시 상태

반복적으로 정상 Rx_valid_data(0)가 '1'값이거나 Nearly_rx_full, full상태가 아니면 Wr_low_fifo 반복 천이하여 수행한다. 만약 tmp_almost_full신호가 활성화되거나 full신호가 활성화

되면 각각 Almost_fullSt, FullSt상태로 천이하여 Idle상태로 천이하게된다[4][5].

3.3 RX FIFO의 읽기 FSM

RX FIFO의 읽기시 상태를 나타내고 있다. Reset 신호가 활성화 되면 Idle상태에 머무른다. Frame_cnt신호가 '0'이 아니거나 Nearly_rd_empty신호가 '0'이 아니면 Rd_Rdy상태로 천이한다. 그림 5은 RX FIFO의 읽기시 상태를 나타낸다.

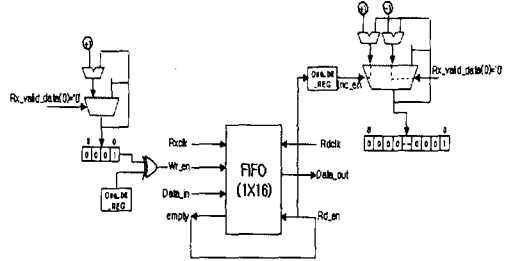


[그림 5] RX FIFO의 읽기시 상태

그리고 Tmp_rd_en신호가 활성화 되면 Rd_St상태로 천이한다. RX FIFO에서 읽어온 데이터에서 Rx_dout_low(0)가 '0'값이면 Codew_ST로 천이하고 128비트 유효 Codeword 프레임 읽어진다. 정상적인 반복수행에서 Frame_cnt신호가 '0'이 아니거나 Nearly_rd_empty신호가 '0'이 아니면 Rd_Rdy상태로 천이하여 반복수행을 한다. 만약 Nearly_rd_empty와 Tmp_almost_empty신호가 활성화되거나 Empty신호가 활성화상태이면 각각 Under_flowST, EmptyST상태로 천이하게되고 Idle로 천이한다[4][5].

3.4 비동기 Frame Count블록

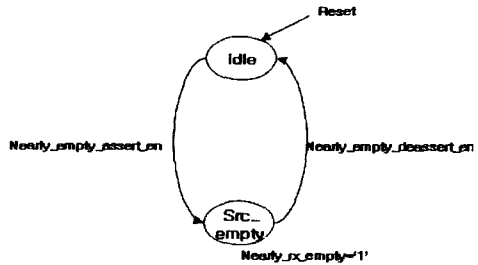
비동기 Frame Count블록은 쓰기 클럭, 읽기 클럭이 독립적으로 발생되어짐에 따라 Write시 프레임 카운트와 Read프레임 카운트 동기화위해 하나 이 FIFO을 생성하여 프레임카운트를 증가, 감소한다. 쓰기상태시 Codeword값을 체크하여 쓰기 프레임 카운트 수를 증가하고 FIFO(1x16)을 두어 한 클럭 wr_en신호가 활성화되어 FIFO에 쓰고 empty신호가 Rd_en 신호를 활성화상태이면 한클럭 tmp_rd_en신호시 Frame_cnt 임시버퍼에 동기화된 프레임카운트가 적용되어 프레임 카운트가 증가하고 Rd_ST FSM 동작상태에서 Codew_st이면 프레임 카운트를 감소시켜 동기화된 Frame_cnt의 카운트수를 셀 수 있다. 그림 6은 프레임 카운트 블록을 나타내고 있다[3][4].



[그림 6] Frame Counter블록

3.5 Nearly_Rx_Threshold 처리부

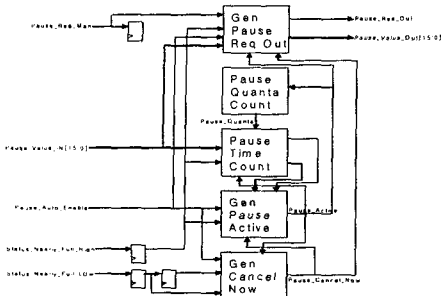
초기 Reset에의해 Idle상태에 있으면서 Nearly_Rx_empty신호가 비활성하고, FIFO의 비어있는 크기와 Nearly_Rdy_empty Assert 초기값과 같아졌을때 Nearly_empty_assert_en신호가 활성화되어 Src_empty가 상태로 천이 하게되고 Src_empty상태에서 FIFO의 비어있는 크기와 초기설정된 Nearly_empty_deassert 초기값과 같아졌을때 Nearly_empty_deassert_en신호는 활성화되어 상태머신을 Idle로 천이하게 된다. 그림 7은 Nearly_Rx_Threshold블록은 나타낸다[3][4].



[그림 7] Nearly_Rx_Treshold 상태

3.6 PAUSE Control 블록

Pause Frame Control블록은 사용자쪽에서 MAC으로 전송하는 프레임의 전송을 제어하기위한 기능을 수행한다. Pause 프레임은 수신쪽 FIFO에서 Threshold값으로 유도된 신호를 사용하여 Rx FIFO에 Nearly_Rx_full신호가 High가 되면 Flow Control Select set되어 자동적으로 Pause_auto_enable신호가 High되고, Pause_value_in [15:0]에 Pause Time의 최소 512 bit-times 값으로 Pause priod (8클럭 X 64비트)단위로 Pause Frame time값을 갖고, MAC하위계층에 Pause프레임을 전송한다. 그림 8은 Pause 컨트롤 블록을 보여주고 있다.

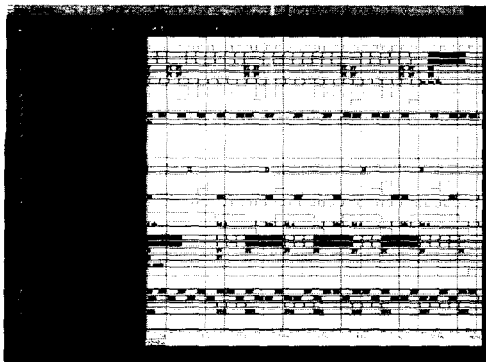


[그림 8] Pause Control블록

Rx FIFO에 Nearly_rd_empty신호가 Low가 되면 Pause_request 취소를 위한 Pause_value_in[15:0]에 Zero값이 만들어지고 Pause_request가 Cancel된 다[4].

4. RX FIFO 읽기시 시뮬레이션 파형

하위계층에서 MAC으로부터 전송된 DATA와 DATA_VALID정보 데이터가 RX FIFO에 임시버퍼링 되어 사용자쪽에 읽어져나가는 파형을 보여 주고있다. MAC으로부터 전송하는 데이터가 Rx_valid_data(0)신호가 활성화되면 Write_en신 호가 각각 상위, 하위FIFO에 Write_en활성화 신호가 활성화되어 쓰여지게된다. 이렇게 하위, 상위 FIFO에 128비트 유효로 쓰여지게되고 상위FIFO상 태에서 Write_en카운트가 증가한다. 하위 FIFO의 Rx_valid_data(0)신호가 Low로 활성화되면 128비 트 유효 Codeword프레임이 쓰인다. 그림 9는 RX FIFO쓰기, 읽기시 파형을 보여준다.



[그림 9] RX FIFO의 쓰기, 읽기시 파형

만약 tmp_almost_full신호가 활성화되면 FIFO에 하나의 depth가 남게되고, Codeword프레임이 쓰여져야 한다. 다음은 RX FIFO읽기시 Frame_cnt신 호가 '0'이 아니거나 Nearly_rd_empty신호가 '0' 이 아니면 Rd_Rdy상태로 천이하고, Tmp_rd_en 신호가 활성화 되면 Rd_St로 천이하여 RX FIFO의

128비트 데이터와 16비트 유효 정보 비트가 읽어진다. 만약 Nearly_rd_empty 와 Tmp_almost_empty가 활성화 되거나 empty신호가 활성화 되면 각각 Underflow신호, empty신호가 활성화 된다[4][5].

5. 결론

본 논문의 MAC용 TX FIFO는 10Gbps의 전송속도를 지원하도록 설계하였다. 기존 10Gbps이하 전송속도에서는 Collision Detect기능이 수행하여 전송속도에 영향을 주었는데 10Gbps에서는 초고속으로 전송하기 때문에 Collision Detect기능이 빠지고 프레임 전송에 대한 Pause Control기능을 추가하여 Flow Control호를 제어에 따라 프레임 전송 10Gbps 전송속도로 최적화되어 FIFO 임시 저장된 후 MAC에게 프레임을 전송하는 기능을 수행한다.

본 논문을 구현하기위해 Xilinx 5.1i 이용하여 VHDL언어를 이용하였고, ModelSim 5.6a시뮬레이션 프로그램을 사용하여 파형분석과 타이밍 분석한 후에 정상적인 동작을 확인하였다. 설계 모듈은 기존 FIFO의 수행보다는 프레임간의 Flow Control기능, Pause Frame Control기능의 전송간의 흐름제어 기능을 추가하여 설계하고, MAC모듈 인터페이스와 RX FIFO의 상태 Threshold값으로부터 유도된 신호에 의해 Back Pressure기능을 추가하여 구현하였다. MAC용 RX FIFO의 응용분야는 10Gbps고속 전송속도가 요구되어지는 모듈에서 Flow Frame control 기능과, Pause Frame Control 기능을 수행하는 모듈에 적용되어 사용이 가능할 것으로 사료된다.

참고문헌

- [1] 박노식, 손승일, 최익성, 이범철 "System Packet Interface 모듈의 성능평가", 한국인터넷 정보학회 추계 학술발표 논문집, pp247-250, November 2003.
- [2] 윤중호 저, "네트워크 엔지니어를 위한 최신 이더넷", 교학사, 2002. pp.336-354, 서론
- [3] 한국전자통신연구원 개발 프로젝트 : NP 스트리밍 정합 개발서 pp.121-123, pp.96 연구수행기관 : 한신대학교 정보통신학과 박노식, 손승일.
- [4] http://www.xilinx.com/ipcenter/catalog/reference/designs/docs/ten_gig_eth_fifo.pdf Xilinx Co. Specification참조
- [5] http://www.xilinx.com/ipcenter/catalog/logicore/docs/ten_gig_eth_mac.pdf Xilinx Co. Specification참조