

# FPGA를 이용한 Network Processor용 Protocol 변환장치의 구현 및 흐름제어

## An Implementation of Network Processor Protocol Converter and Flow Control using FPGA

방진민\*(Jin-Min Bang), 조준동\*\*(Jun-Dong Cho), 김석호\*\*\* (Austin.s.Kim)

**Abstract** - Recent trend on high speed packet processing for providing multiple internet services is to use network processor instead of being implemented by legacy ASIC or FPGA. Most frequently used network processor interface is the SPI4.2. This paper address the data-rate conversion interface device between SPI4.2 and SPI3/CSIX, implemented using XILINX XC2VP40 FPGA. Furthermore, we address the methodology and necessity of flow control occurred due to the data rate difference between 10Gbps and 3.2 Gbps.

**Key Words** :Bridge,SPI4.2,SPI3,CSIX,Network Processor,Flow Control

### 1. 서론

Network Processor는 ASIC 또는 FPGA등에 하드웨어가 수행하는 고속의 고정처리 기능을 소프트웨어 기반으로 처리하여 다양한 고속 실시간 Packet 처리 기능을 저비용으로 단시간 내에 개발할 수 있는 장점을 가지고 있어 BSC,ACR,라우터등에 핵심 기술로서 개발 되었다.[7]

한편 현재 구축된 Device간 Interface는 네트워크의 발전 속도를 충족시키지 못하고, Network망 내부에서 병목 현상을 일으키고 있다. 이러한 문제를 해결하기 위하여 초고속 Interface 기술이 발표되고 있다.

Network Processor의 Main Traffic입/출력 단자로는 주로 Physical Layer 소자와 Link Layer 소자 간에 10 Gbps 이상의 성능을 낼 수 있도록 OIF(Optical Interface Forum)에서 권고한 규격인 있는 SPI4.2 Protocol을 사용한다.[1][4][8][9]

그러나 Link Layer Device의 경우 최대 4 Gbps 미만의 SPI3/CSIX등의 Protocol을 사용하는 경우가 많다. 따라서 SPI4.2와 SPI3/CSIX등의 Protocol간 변환장치가 필요하게 되었다.

본 논문은 Protocol간에 정합을 위한 변환장치를 FPGA에 구현하는 방법과 그 흐름제어 방법을 제안 한다.본 논문은 구성은 다음과 같다. 1장에서는 서론을 2.1장에서는 FPGA의 구성,2.2장에서는 흐름제어2.3에서는 구현 결과에 대하여 설명하고 3장에서는 결론을 맺는다.

### 2. 본론

#### 2.1 변환 장치 FPGA 구성 및 흐름제어

##### 2.1.1 Network Processor를 이용한 System 구성

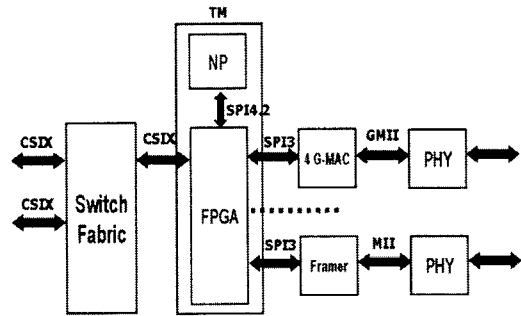


그림 1 Network Processor를 사용한 System 구조 예

그림1은 Network Processor가 Traffic Manager 역할을 하는 System의 구성 예이다.[3]

Network Processor는 SPI4.2 Interface를 가지고 있으나 Network Device의 Ethernet Device의 Traffic Manager 방향 Interface는 SPI3로 구성되어 있으며 Switch Fabric 방향으로 CSIX의 Interface를 가지고 있다. 따라서 이러한 여러 가지 Interface의 정합을 위한 변환 기능을 수행하는 FPGA가 필요하며 본 연구에서는 VHDL을 이용하여 이러한 기능의 FPGA를 구현 하였다.

##### 2.1.2 변환장치 FPGA 구성

그림 2는 SPI3 1 Port와 CSIX 1 Port를 사용하여 구성된 FPGA기능Block도이다.PMD(Physical Medium Independent)로부터 SPI3 Rx를 통하여 수신된 Packet은 FIFO에 저장된

#### 저자 소개

- \* 방진민 : 成均館大學 電子電氣學科 碩士課程  
三星電子 通信研究所 責任 研究員
- \*\* 조준동 : 成均館大學 電子電氣學科 教授 · 工博
- \*\*\* 김석호 : 三星電子 通信研究所 任員 · 工博

후 SPI4.2 Tx를 통하여 Network Processor에 전달된다. Network Processor는 이렇게 전달된 Packet을 CSIX Format인 C-Frame화 한다. C-Frame은 다시 SPI4.2 Rx를 통하여 FPGA로 전달되어 FIFO에 저장된 후 CSIX Tx를 통하여 Switch Fabric에 전달된다.

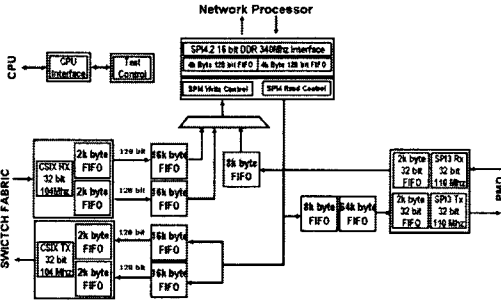


그림 2 변환장치 FPGA 기능 Block도

반대로 Switch Fabric에서 CSIX Rx를 통하여 FPGA에 전달되는 Egress C-Frame은 FPGA FIFO에 저장 후 SPI4.2 Tx를 통하여 Network Processor로 전달된다. Network Processor는 이러한 C-Frame을 Packet화하여 SPI4.2 Rx를 통해 FPGA에 전달하며 FPGA는 Logical Port(이후 LP) FIFO에 저장 후 SPI3 Tx를 통하여 PMD에 전달한다.

## 2.2 흐름 제어

### 2.2.1 FPGA 흐름 제어

Network Processor는 FPGA로부터 유입되는 Data의 량이 처리 한계를 넘으면 SPI4.2 Tx Status Bit 라고 불리는 Flow Control용 Bus(2 Bit)를 이용하여 Back Pressure(이후 BP)를 FPGA에 전달한다. FPGA는 해당 LP에 BP가 발견되면 즉시 전송을 멈추게 된다.

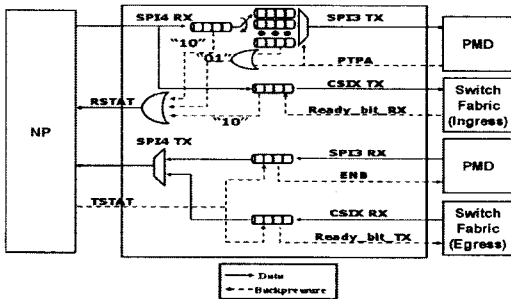


그림 3 흐름제어

반대로 Network Processor에서 FPGA의 SPI4.2 Rx를 통한 Data 전달 방향에 흐름제어는 FPGA에서 Network Processor에 보내는 Rx Status를 이용한다. 이때 SPI3 방향으로 HOL(Head of Line Blocking)발생을 피하기 위하여 LP 별로 Packet의 관리가 필요하므로 FIFO 구조 역시 LP별로 구분하여 구성하였다.

### 2.2.2 Logical Port FIFO Threshold

그림4와 같이 SPI4.2를 통하여 10 Gbps이상의 Data가 SPI3로 전달 시 FIFO Overflow가 발생하게 되며 이를 방지하기 위하여 Network Processor에 BP를 전달해야 한다. 따라서 한 LP의 BP 정보를 Network Processor에 전달하여 BP가 반영되기까지는 다음과 같은 시간이 소요 된다.[1]

$$T(BP Send) = (4 * SPI4.2 Clock Period * (Total LP Number + 2)) + Network Processor Processing Time \quad (1)$$

$T(BP Send)$  : BP Propagation Delay Of FPGA To Network Processor

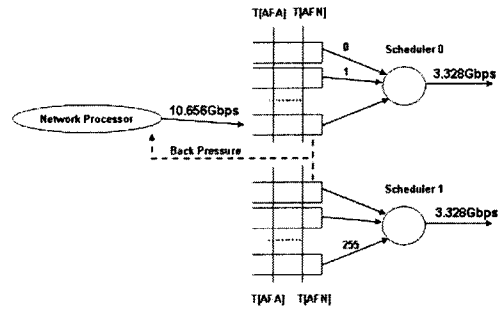


그림 4 Threshold에 의한 흐름제어

FIFO에 일정한 량의 Data가 누적되면 BP를 발생하는데 이를  $T[Afa]$ (Almost Full Assert Threshold)라고 한다. 이러한 Threshold에 근접하여 BP가 발생한 후, BP를 Network Processor에 전달하는 동안 누적되는 Data의 량은 다음과 같다.[10]

$$D(Afassert) = T(BP Send) * (Th(SPI4.2) - Th(SPI3)) \quad (2)$$

$D(Afassert)$ : Accumulated Total Data Size When BP Assertion

$T(BP Send)$ : Maximum BP Propagation Time

$Th(SPI4.2)$ : SPI4.2 Throughput

$Th(SPI3)$ : SPI3 Throughput

위와 같이 FPGA로부터 BP를 받은 Network Processor는 전송을 멈추게 되며 SPI3 Tx로는 계속해서 Data가 전송되므로 특정시점에서 Network Processor에 선언한 BP를 Negate하는 Threshold가 있어야 하는데 이를  $T[Afn]$ (Almost Full Negate Threshold)라고 한다.

Network Processor에 BP 해지를 알리는 동안 SPI3를 통해 전송되는, 즉 FIFO에서 감소하는 Data의 량은 다음과 같이 구할 수 있다.

$$D(AfNegate) = T(BP Send) * Th(SPI3) \quad (3)$$

$D(AfNegate)$ : Decreased Total Data Size When BP Negate

$T(BP)$ : BP Propagation Time

$Th(SPI3)$ : SPI3 Throughput

위의 식 (3),(4)에서  $D(AfAssert) > D(AfNegate)$  이므로 LP당 최소fifo Size는 두 값을 더하면 된다.

$$\text{Minimum FIFO Size/LP} = D(Af \text{ Assert}) + D(Af \text{ Negate}) \quad (4)$$

위의 식으로부터 256 LP에 333Mhz의 SPI4.2 Clock 사용 시 실제로 값을 구해보면

$$\begin{aligned} \text{Minimum FIFO Size/LP} &= 3.754 \text{ K Byte} + 1.705 \text{ K Byte} \\ &= 5.459 \text{ Kbyte /LP} \end{aligned} \quad (5)$$

가 되므로 최소한 5.6 Kbyte이상의 LP FIFO가 필요하다.

### 2.2.3 Traffic Simulation Modeling

본 논문의 FPGA에서는  $T[Afa]$ 보다 커지는 순간과  $T[Afn]$ 보다 작아지는 순간에 FIFO 정보를 피드백하여 데이터의 손실을 최소화 하고 FIFO의 활용을 최대화 하도록 하였다. [10]

그림 5는 Network Simulator인 OPNET을 이용하여 본 FPGA의 FIFO 형상 및 Network Processor의 기능을 Modeling 한 것이다. 이러한 Model을 이용하여 설정한 Threshold에 따른 결과를 Traffic Simulation 하였다.

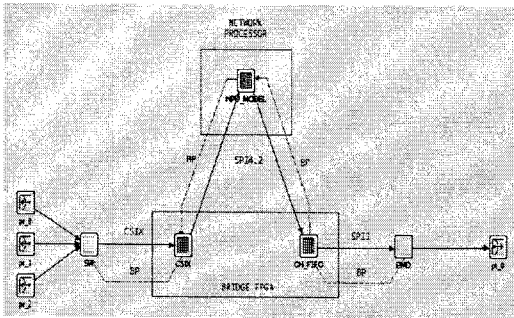


그림 5 OPNET에 의한 System Model

최소 요구 FIFO Size가 식 (6)과 같이 5.459 Kbyte/LP이며 Memory의 할당에 편리를 위하여 8 KByte/LP로 결정했을 때  $T[Afa]$ 와  $R[Afn]$ 을 결정해야 한다.

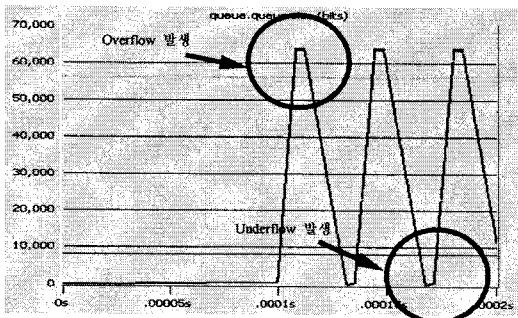


그림 6 Parameter 이상 설정에 따른 FIFO 사용률

그림 6은 Parameter 설정 잘못에 의한 Overflow Underflow의 예이며 그림7은 중간에 2 Gbps로 Throughput 저하가 발생하는 것을 나타낸다.

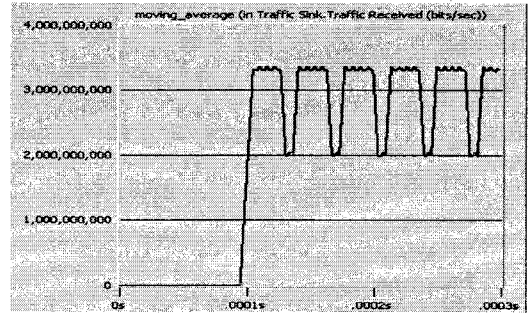


그림 7 Parameter 이상 설정에 따른 SPI3 Throughput

Throughput 저하와 문제를 개선을 위하여 여러 가지 Parameter 값을 표 1과 같이 입력하여 Simulation 결과 그림 8과 같은 결과를 얻을 수 있다.

	T[Afa]	T[Afn]
Base0	2600 Byte	2500 Byte
Base1	4200 Byte	1800 Byte
Base2	6200 Byte	1000 Byte
Base3	7200 Byte	500 Byte
Base4	7800 Byte	100 Byte

표1 Parameter 설정 예시

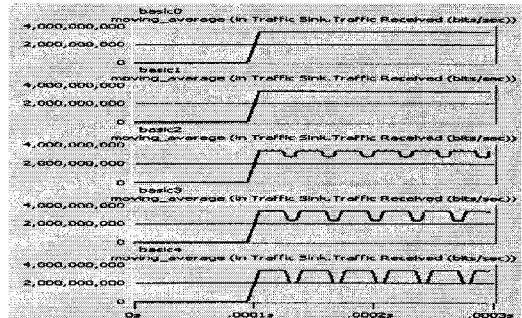


그림 8 Parameter 설정 변화에 따른 SPI3 Throughput

한편 상기 Simulation과 같은 Throughput 저하를 막기 위한 Parameter값을 계산하면 다음과 같다.

$$T[Afa] < \text{FIFO Size} - D(Af \text{ assert}) \quad (7)$$

$$T[Afn] > D(Af \text{ Negate}) \quad (8)$$

식 7과 8에 의하여 실제로 구하면

$$T[Afa] < 64000 \text{ Bit} - 30037 \text{ Bit} = 33963 \text{ Bits}$$

$$T[Afn] > 13641 \text{ Bits}$$

위와 같은 Parameter를 적용한 결과 그림 9은 Overflow와 Underflow가 발생하지 않고 있음을 나타내고 그림 10은 SPI3의 Full Throughput을 나타내고 있음을 볼 수 있다.

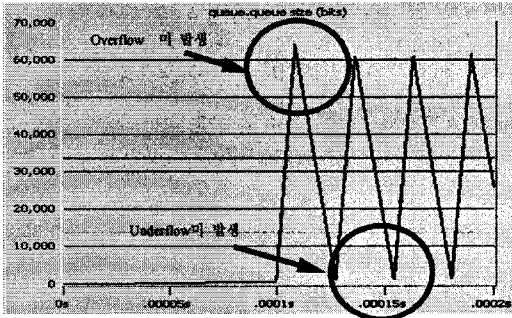


그림 9 Threshold 수정에 따른 FIFO 사용자

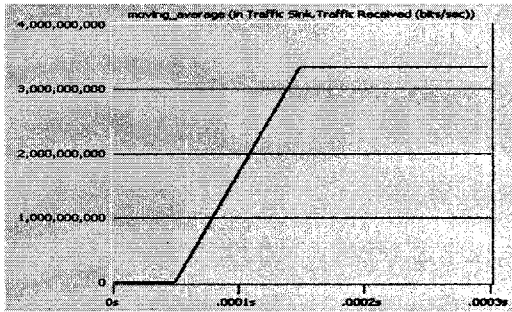


그림10 Threshold 수정 따른 SPI3 Throughput

### 2.3 RTL Simulation 결과 및 구현

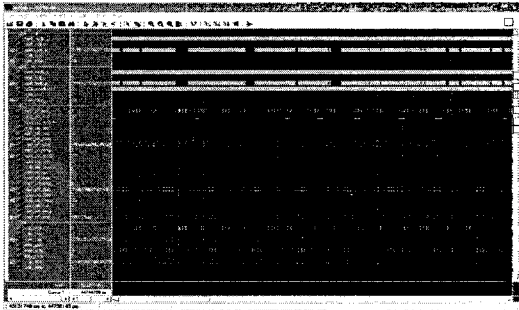


그림11 FPGA Logic Simulation



그림12 FPGA Placement

본 논문의 장치는 Xilinx FPGA인 XC2VP-40 1152 Series 를 이용하여 VHDL로 구현하였다. 그림 11는 구현된 FPGA

의 RTL Simulation을 Modelsim을 이용하여 실행한 결과이다. SPI4.2 Rx,Tx,SPI3,CSIX 순으로 Wave를 나타내고 있다. 또한 그림 12는 Synthesis 및 Placement와 Routing 후의 배치도이다.

### 3. 결론

본 논문에서는 Network Process의 고속 Interface로 사용하는 SPI4.2와 Switch Interface인 CSIX 및 PMD의 일반적 Interface인 SPI3간에 Interface 교환을 해주는 변환장치의 설계 방법에 대하여 논하였다.

또한 본 변환장치에 있어 가장 중요한 문제인 고속 Interface와 저속 Interface간 발생하는 흐름제어 문제에서 LP별로 가능한 최적의 FIFO Size를 결정하고 그에 맞는 Threshold를 찾는 방법과 Traffic Simulation 하는 방법을 제안 하였다.

이와 같은 변환장치는 향후 다양한 Interface 요청에 따라 UTOPIA, SONET, XAUI등 Application에 따라 Interface추가 구현이 필요 할 것으로 보이며, 또한 Dual, Triple SPI4.2 등의 구성도 가능하다.

### 참 고 문 헌

- [1] Optical Internetworking Forum, "System Packet Interface Level 4 (SPI-4) Phase 2 Revision 1: 1: Oc-192 System Interface For Physical And Link Layer Devices", Oif-SPI-4-02.1, October 15, 2003.
- [2] Optical Internetworking Forum, "System Packet Interface Level 3: Oc-48 System Interface For Physical And Link Layer Devices", Oif-SPI3-01.0, June 2000.
- [3] Network Processor Forum "CSIX-L1: Common Switch Interface Specification-L1" August 5, 2000.
- [4] Xilinx "SPI-4.2 Core V6.0 Product Specification" July 31, 2003.
- [5] Xilinx "SPI-4.2 To Quad SPI-3 Bridge" June 24, 2003
- [6] Xilinx "Virtex-Ii Pro™Platform FPGAs Introduction And Overview" March 24, 2003.
- [7] 김선일, "A Study On Network Processors For The Next Generation Internet" 과학기술원 논문집 제 4권2002.
- [8] 손승일, "A Design Of SPI-4.2 Interface Core", 한국 해양 정보통신학회논문지 제8권 제6호,2004.
- [9] 박노식, 손승일, 이범철 "Design For Receive Unit Of System Packet Interface Level 4 Phase 2", 한국 해양 정보통신학회논문지 제8권 제1호,2004.
- [10] 서주하, 정부영, 유현희 "A Flow Control Based Queue Dynamics Estimate Information", 산업기술연구소 논문집 제19집,1999.