

비트 및 워드 연산용 초고속 프로세서 설계

허재동\* . 양오  
 청주대학교 이공대학 전자공학과

The Design of High Speed Bit and Word Processor

Jae Dong Her\* . Oh Yang  
 Dept. of Electronic Engineering, Chongju Univ.

**Abstract** - This paper presents the design of high speed bit and word processor for sequence logic control using a FPGA. This FPGA is able to execute sequence instruction during program fetch cycle, because the program memory was separated from the data memory for high speed execution at 40MHz clock. Also this processor has 274 instructions set with a 32bit fixed width, so instruction decoding time and data memory interface time was reduced.

This FPGA was synthesized by V600EHQ240 and Foundation tool of Xilinx company. The final simulation was successfully performed under Foundation tool simulation environment. And the FPGA programmed by VHDL for a 240 pin HQFP package.

Finally, the benchmark was performed to prove that the designed for bit and word processor has better performance than Q4A of Mitsubishi for the sequence logic control.

1. 서론

반도체 기술의 비약적인 발달과 디지털 논리회로의 고집적화로 마이크로프로세서를 이용한 응용기술은 계속 발전하고있다. 마이크로프로세서의 영향은 퍼스널 컴퓨터나 범용 마이크로컴퓨터의 CPU에서 가전제품, 사무용 기기, 산업용 기기 등과 같이 광범위하게 사용되고 있다. 그 동안 프로그래머블로직 컨트롤러의 설계를 위해 쉽게 사용할 수 있고 구현하기 쉬운 범용의 마이크로 프로세서를 채택하여 기능을 구현하였다. 그러나 산업현장에서의 시퀀스 제어 및 공장 자동화 등과 같은 특장 응용 분야에서는 범용의 프로세서에서 취급하는 바이트나 워드단위의 데이터가 아닌 비트 단위의 데이터 처리를 주로 이용하기 때문에, 이를 범용의 프로세서를 이용하여 구현할 경우 비트 데이터를 구성하기 위해 마스크(Mask), 쉬프트(Shift) 등의 불필요한 동작으로 인해 메모리의 용량 증가와 동작 시간상의 손해를 보게 된다. 이에 본 논문에서는 비트 및 워드 프로세서를 설계하여 사용자 프로그램의 용량 증가를 해결하고 시퀀스 명령처리에 대한 고속화를 꾀하였다. 시퀀스 명령어에 대한 고속처리를 위해서 적절한 타이밍설계와 프로그램 메모리와 데이터 메모리의 버스를 분리(Harvard Architecture) 및 명령어 디코딩 시간을 줄이기 위해 RISC(32bit) 구조 등을 이용하여 명령어 처리속도의 고속화를 꾀하는데 주안점을 두었다.<sup>(1)</sup> 이와 같은 기능들을 VHDL<sup>(2)</sup>를 이용하여 FPGA로 구현하기 위하여 Xilinx사의 V600EHQ240<sup>(3)</sup>과 Foundation 4.2i 합성툴을<sup>(4)</sup> 이용하여 로직을 구현하였다.

마지막으로, 본 논문에서 설계된 프로세서의 제어성능을 비교하기 위해 Mitsubishi사의 Q4A를<sup>(5)</sup> 이용한 시퀀스 제어시스템과의 성능을 비교하여 본 논문에서 설계

된 시퀀스전용의 비트 및 워드 프로세서가 우수함을 실험을 통해 확인하였다.

2. 본론

2.1 고속처리를 위한 비트 및 워드 프로세서 설계

2.1.1 개요 및 구성

비트 및 워드 프로세서에 대한 고속 처리를 위해 프로그램 메모리와 데이터 메모리의 버스를 분리하여 설계하였으며, 고속 프로세서의 설계하기 위한 내부의 구성도는 그림 1과 같다.

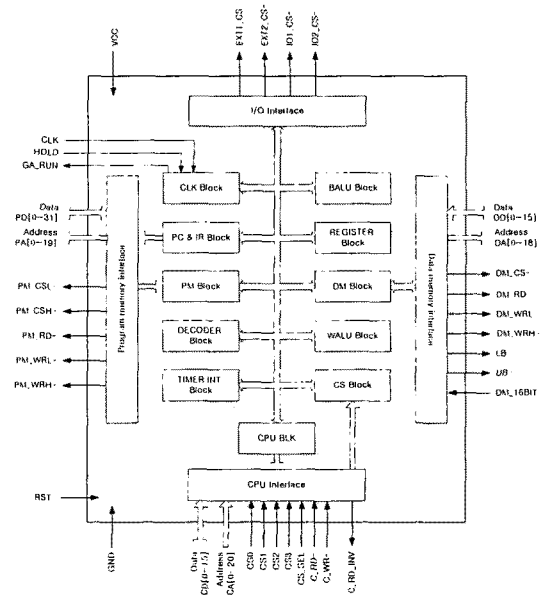


그림 1. 비트 및 워드 프로세서의 내부 구성도

그림 1에서의 CS 블록에서는 범용 프로세서의 어드레스 버스가 연결되며 이 어드레스는 시퀀스 제어용 프로세서 내부에서 선택신호를 만드는 어드레스로 사용된다. PM 블록에서는 인스트럭션 레지스터를 32비트로 구성하기 위해 16비트 메모리 2개를 사용하였고 이를 위해 WR 신호를 High 16비트와 Low 16비트로 분리하였다. DM 블록에서는 DM\_16BIT가 0일 때는 8비트로 '1'일 때는 16비트로 인터페이스 하게 하였으며 8비트로 사용할 때 쓰기에서 High와 Low 번지를 분리하여 제어하기 위해 DM\_WRH와 DM\_WRL-신호를 사용했다. 또한 16비트 메모리를 사용할 때 상위 8비트와 하위 8비트를 제어하기 위해 LB-와 UB- 신호를 사용하였다. CPU 블록에서는 CS\_SEL 신호로 범용프로

세서에서 직접 Address가 Decoder된 CS0, CS1, CS3과 Decoder되지 않은 CA 신호를 선택할 수 있게 설계하였다. Timer interrupt 블록에서는 타이머 인터럽트 수행을 위한 레지스터들을 제어하고 있으며 인터럽트 발생시 인터럽트 루틴을 수행하게 한다. CLK 블록에는 비트 및 워드 프로세서를 제어하는 HOLD와 GA\_RUN이 연결된다. 만약 외부로부터 HOLD의 입력 신호가 low가 되면 비트 및 워드 프로세서가 처리할 수 있는 명령어의 경우에는 GA\_RUN을 high로 출력하고 명령어를 수행한다. 하지만 수행 불가능한 명령어일 경우에는 GA\_RUN을 low로 출력하여 범용 프로세서에 제어권을 넘겨주게 된다. 그림 1에서 설계된 프로세서는 총 20비트로 구성된 프로그램 카운터를 기본으로 하여 동작하며 HOLD가 high가 되면 정지하는 구조로 되어 있다. 또한 내부 클럭 발생을 위해 외부 클럭 40MHz를 입력하였다. 그리고 비트 ALU와 워드 ALU를 따로 설계하였다.

### 2.1.2 프로세서의 설계사양

먼저 FPGA를 이용한 비트 및 워드 연산용 고속 프로세서를 설계하기 위한 설계사양을 표 1에 나타내었다. 고속 처리를 위한 설계와 설계한 FPGA의 디버깅을 위하여 한 명령어씩 수행하는 싱글 스텝런(single step run), 프로그램 카운터의 내용과 프로그램을 정지하고 자하는 어드레스가 일치했을 경우 정지하는 방법(pc break run), 정지하고자 하는 데이터 어드레스가 일치했을 경우 정지하는(dm break run)방법 등의 기능이 필요로 하다. 이와 같은 디버그 기능은 실시간 디버그 기능을 구현하므로 실제 상황과 동일한 방법으로 실현할 수 있는 장점을 가지고 있다.

표 1. 시퀀스 로직 제어용 고속 프로세서의 설계사양

프로세서 비트 처리용 명령어 수		118 종류
프로세서 워드 처리용 명령어 수		156 종류
프로세서 내부 데이터 버스		32/64 비트
프로세서 외부 프로그램 메모리 용량		32비트 × 1 Mbytes
프로세서 외부 데이터 메모리 용량		16비트 × 512 kbytes
워드 명령	내부 처리	75 ns
	16비트 단위로 외부 데이터 메모리 쓰기 처리속도	100 ns
	16비트 단위로 외부 데이터 메모리 읽기 처리속도	125 ns
	32비트 단위로 외부 데이터 메모리 쓰기 처리속도	175 ns
처리 속도	32비트 단위로 외부 데이터 메모리 읽기 처리속도	175 ns
	32비트 단위로 외부 데이터 메모리 읽기 처리속도	200 ns
	32비트 단위로 외부 데이터 메모리 읽기 처리속도	200 ns
비트 명령	내부 처리	75 ns
	외부 데이터 메모리 읽기 또는 쓰기	100 ns
	외부 데이터 메모리 읽기와 쓰기	175 ns
실시간 디버그(Debug) 기능		1 스텝 운전 모드
		일정번지 데이터 메모리 브레이크 운전 모드
		일정번지 프로그램 메모리 브레이크 운전 모드
인터럽트 기능		32bit Timer Interrupt
다른 MPU와의 인터페이스 기능		프로그램 RUN-STOP 지령

### 2.2 합성 및 시뮬레이션

본 논문에서 제안된 각각의 모듈을 톱다운 방식으로 프로그램 메모리 인터페이스부, 데이터 메모리 인터페이스부, 디코더부, I/O 인터페이스부, 비트 ALU, 워드 ALU 및 레지스터부, 타이머 블록 등 각각의 모듈을 설계하여 VHDL<sup>[2]</sup>로 기술하였다. 이와 같이 기술된 VHDL을 Xilinx사에서 제공되는 Foundation V4.2i의 로직합성툴을 이용하여 로직을 합성하였고 최고속도 40MHz를 갖고 600,000 게이트에 해당되며 핀의 형태는 240핀 HQFP 형태인 V600EHQ240을 사용하였다. 이때 배치 및 배선의 과정을 통해 FPGA의 사용율은 6,912개의 셀중 5,351개를 사용함으로써 77.4%가 됨을 확인하였다.

아울러 테스트 벡터를 원활히 작성하고 시스템 레벨상에서 시뮬레이션을 하기 위해 Xilinx사의 Foundation V4.2i를<sup>[5]</sup> 사용하여 테스트 벡터를 작성한 후 포스트 시뮬레이션(post simulation)을 하였다. 그림 2와 그림 3은 배치 및 배선을 마친 후 포스트 시뮬레이션의 워드 명령어와 비트 명령어 결과를 나타내고 있다. GA-RUN이 High가 되는 것은 현재 프로세서가 명령어를 처리하고 있음을 나타내고 있고, CLK은 40MHz가 입력되었다. 그림 2에서 처리명령어는 시뮬레이션 결과 아래에 수행시간과 함께 나타내었다. 워드 명령어에서 MOV, ADD는 75ns가 소요되며, BCD 변환은 200ns, SHLL은 75ns임을 알 수 있다. BCD 변환에서는 T3클럭이 길게 늘어진 것은 BCD로 변환하는 시간을 늘리기 위해 Wait를 주었기 때문이다. 비트 및 워드 명령어의 수행시간을 늘리기 위해서 워드 명령어는 Wait를 사용하였으며 비트 명령어에서는 클럭수를 늘리는 방식으로 수행시간을 제어하게 설계하였다.

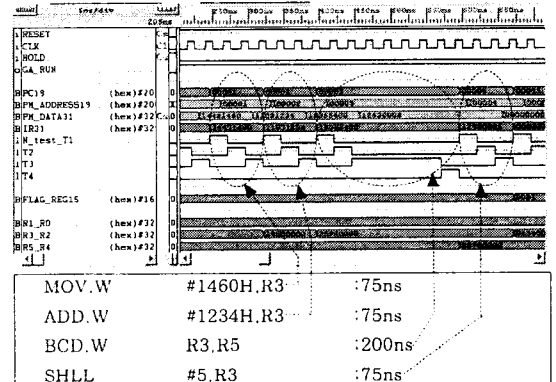


그림 2. 워드 명령어에 대한 시뮬레이션 결과

그림 3에서 수행명령어와 수행시간을 결과 아래에 나타내었다. DM\_WRL-와 DM\_WRH-의 신호가 동시에 low로 되는 것은 시스템 설계시 DM 메모리를 8비트가 아닌 16비트로 구성하였기 때문에 외부 쓰기 신호인 DM\_WRL-, DM\_WRH-중 하나를 선택할 수 있도록 설계하였기 때문이다.

이 결과에서 비트 명령어는 내부처리 일때는 75ns가 소요되며, 외부 데이터 메모리를 읽을 때는 100ns, 외부 메모리 읽기와 쓰기 일때에는 175ns임을 알 수 있다. 또한 펄스처리 명령어인 PLS는 상승에지가 검출되면 Flag(11)인 EF(Edge flag)을 1과 데이터 메모리에 0을 출력하고 반대로 상승에지의 상태가 검출되지 않으면 EF에 0을 데이터 메모리에 1을 출력하는 동작을 한다. 그림 3에서의 시뮬레이션 결과에서는 상승에지가 검출되어 FLAG(11)비트가(EF) 1로 외부데이터 메모리에는 0이 출력됨을 알 수 있다. 이와 같은 펄스처리 시간은 총 7사이클로써 처리속도가 175ns임을 알 수 있다.

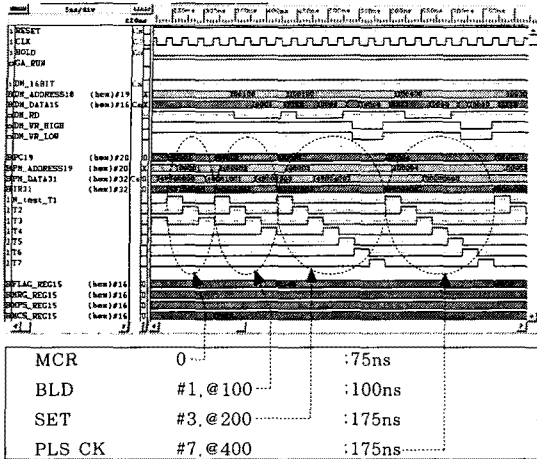


그림 3. 비트 명령어에 대한 시뮬레이션 결과

### 2.3 실험 및 결과

FPGA를 이용한 비트 및 워드 프로세서의 성능을 실험을 통해 평가하기 위해 그림 4와 같이 시스템을 구성하였다. 그림 4에서 H8S/2148은<sup>[6][7]</sup> HITACHI사의 범용 마이크로 콘트롤러이며 FPGA를 제어하기 위해 사용하였으며 외부 클럭은 시리얼 통신시 38,400bps의 전송속도에 적합한 19.6608MHz를 사용하였다. Buffer는 74ALVC164245로 신호선들의 전압 레벨을 (3.3V~5V) 변환해 주기 위해 사용하였으며, 시퀀스 제어용 프로세서의 프로그램 메모리와 데이터 메모리는 액세서타임이 55ns인 CMOS SRAM(K6T4016V3C-TB55)를 각각 2개와 1개를 사용하였다. 또한 시퀀스 제어의 1스캔이 완료되면 입력 카드로부터는 입력을 읽고 출력카드로 연산된 데이터를 출력 할 수 있는 구조로 입출력을 각각 설계하였다.

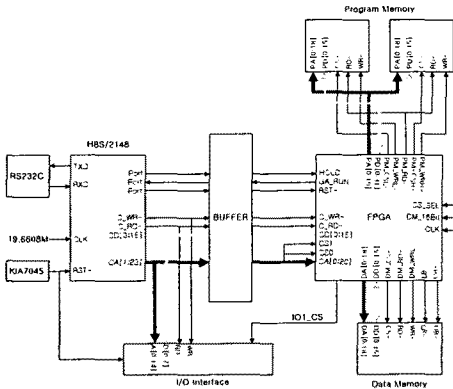


그림 4. 비트 및 워드 프로세서의 시스템 구성

비트 및 워드 프로세서와 Mitsubishi사의 Q4A와 비교 실험한 결과는 표 2와 같다.

### 3. 결론

본 논문에서는 FPGA를 이용하여 PLC의 시퀀스 제어 위한 비트 및 워드 프로세서를 설계하였다. 설계시 고속처리의 문제점을 해결하기 위해 프로그램 메모리 버스와 데이터 메모리 버스는 하버드 구조로 설계하였으며 명령어를 패치하여 Decoder 시간을 줄이기 위해 RISC 구조로 설계하였다. 또한 이를 위해 VHDL을 이용하여

표 2. 본 논문에서 설계된 프로세서와 Q4A의 명령어 수행시간 비교

비트 명령어 종류	본 논문에서 설계된 프로세서	Q4A
ANB, ORB, MC, MCR, MPS, MRD, MPP, INV	75ns	75ns
LD, LDI, AND, ANI, OR, ORI, XOR	100ns	75ns
LDP, LDF, ANDP, ANDF, ORP, ORF, PLF, PLS	175ns	2500us
MOV	16비트	225ns
	32비트	375ns
ADD, SUB	16비트	600ns
	32비트	825ns
MUL	16비트	1250ns
	32비트	1825ns
DIV	16비트	1250ns
	32비트	1825ns
BCD	16비트	725ns
	32비트	1050ns
BIN	16비트	725ns
	32비트	1050ns
BADD, BSUB	16비트	600ns
	32비트	825ns
AND, OR, XOR	16비트	600ns
	32비트	825ns
ROL, ROR	16비트	600ns
	32비트	825ns
RCL, RCR	16비트	600ns
	32비트	825ns
JMP	150ns	3000ns

FPGA를 설계하였으며 Xilinx사의 240핀 HQFP 형태인 V600EHQ240 디바이스와 Foundation V4.2i 합성툴을 이용하여 시뮬레이션을 성공적으로 수행하였다. 본 논문에서 설계된 프로세서의 우수성을 보이기 위해 Mitsubishi사의 Q4A와 비교하여 비트 명령어에서의 펄스 처리 명령어일 경우 약 14배가 빠르며 워드 명령어일 경우에도 약 2배 이상 빠름을 확인하였다.

### (참 고 문 헌)

- [1] M. Morris Mano, "Computer System Architecture", 1997.
- [2] R. Lipsett, C. Schaefer, "VHDL : Hardware Description and Design," KALA, 1991.
- [3] Xilinx, "VirtexTM-E 1.8V Fidle Programmable Gate Arrays", 2001.
- [4] 이준성/서강수, "Xilinx Foundation을 이용한 디지털 시스템 설계", 북두출판사, 2001.
- [5] MITSUBISHI, "QCPU(Q Mode)/QnACPU Programming Manual(Common Instructions)" 1999.
- [6] HITACHI Semiconductor, "Hitachi Single-Chip Microcomputer H8S/2144 Series, H8S/2148 Series Hardware Manual", 1997.
- [7] HITACHI Semiconductor, "Hitachi Single-Chip Microcomputer H8S/2144 Series, H8S/2148 Series Programming Manual", 1997.