

PLC의 시퀀스 제어를 위한 BIT 연산 프로세서의 구현

유영상\*, 양오  
 청주대학교 이공대학 전자·정보통신·반도체 공학부

An Implementation of Bit Processor for the Sequence Logic Control of PLC

Young Sang Yu\* , Oh Yang  
 Dept. Electronic Engineering, Chongju Univ.

**Abstract** - In this paper, A bit processor for controlling sequence logic was implemented, using a FPGA. This processor consists of program memory interface, I/O interface, parts for instruction fetch and decode, registers, ALU, program counter and etc. This FPGA is able to execute sequence instruction during program fetch cycle, because of divided bus system, program bus and data bus. Also this bit processor has instructions set that 16bit or 32bit fixed width, so instruction decoding time and data memory interface time was reduced.

This FPGA was synthesized by pASIC 2 SpDE and Synplify-Lite synthesis tool of Quick Logic company. The final simulation for worst cases was successfully performed under a Verilog HDL simulation environment. And the FPGA programmed for an 84 pin PLCC package.

Finally, the benchmark was performed to prove that Our FPGA has better performance than DSP(TMS320C32-40MHz) for the sequence logic control of PLC.

1. 서 론

본 논문에서는 FPGA를 이용하여 산업용 자동제어 장치로 널리 사용되고있는 시퀀스 로직 제어용 비트 프로세서를 설계하였다. 이를 위해 VHDL<sup>(1)</sup>을 이용하여 FPGA를 설계하였으며 이 FPGA의 내부에는 프로그램 메모리 인터페이스부, 데이터 메모리 인터페이스부, 입출력(I/O) 인터페이스부, 인스트럭션 페치 및 디코더부, 레지스터 및 ALU부, 프로그램 카운터부 등으로 각각 구성된다. 본 논문에서는 FPGA의 설계시 고속처리의 문제점을 해결하기 위해 프로그램 메모리와 데이터 메모리부를 분리 설계함으로써 인스트럭션을 페치하는 도중에 시퀀스명령을 실행할 수 있는 구조로 하였으며, 클럭 전용핀을 활용하여 40MHz에서도 동작할 수 있도록 하였다. 아울러 프로세서의 명령어들을 시퀀스제어에 적합하도록 16비트 또는 32비트로 고정하여 명령어의 디코딩시간과 데이터메모리의 인터페이스 시간을 줄였다. 이와 같은 기능들을 FPGA로 구현하기 위하여 퀵로직(Quick Logic)사<sup>(2)</sup>의 pASIC 2 SpDE와 Synplify-Lite 합성툴<sup>(3)</sup>을 이용하여 로직을 합성하였다. 또한 Verilog HDL<sup>(4)</sup> 환경에서 최종 시뮬레이션이 성공적으로 수행되었다. 아울러 구현된 FPGA를 84핀 PLCC 형태의 FPGA로 프로그래밍 하였다.

마지막으로, 본 논문에서 설계된 프로세서의 제어성능을 비교하기 위해 DSP(TMS320C32-40MHz)<sup>(4)</sup>를 이용한 시퀀스 제어시스템과 성능을 비교하여 본 논문에서 설계된 시퀀스전용의 비트 프로세서가 우수함을 실험을 통해 확인하였다.

2. 본 론

2.1 시퀀스 로직 컨트롤러의 구조

순차제어나 시퀀스제어 등과 같은 자동제어기기로 널리 사용되고있는 시퀀스 제어기의 구성은 그림 1과 같이 구성된다.<sup>(6)</sup> 시퀀스 제어기는 미리 정해진 사용자 프로그램에 따라 순차적으로 반복하여 처리하는 제어기로서 퍼스널 컴퓨터와 매우 유사한 구조를 가지고 있다. 이러한 측면으로 볼 때 시퀀스 제어기는 제어에 적합한 명령어를 구비할 것과 속도 제어가 가능하도록 리얼타임 처리가 요구된다.<sup>(7)(8)</sup> 아울러 시퀀스 제어기에 연결되는 입출력 기기가 다양할 것과 각각의 입출력에 대한 고신뢰성이 요구되고 있다.

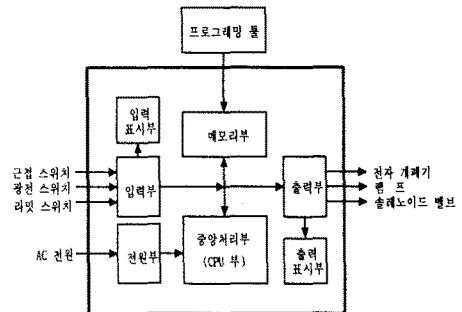


그림 1. 시퀀스 로직 컨트롤러의 구성

이상과 같이 구성된 시퀀스 제어기의 동작은 주로 논리 비트연산, 수치연산 등 여러 가지 처리기능을 가지고 있으며 프로그램의 제어기능에는 사이클릭(cyclic)처리, 정주기 타스크처리(time driven task), 외부 인터럽트 타스크처리 등이 있다.<sup>(9)</sup>

2.2 비트 프로세서 설계

2.2.1. 개요 및 구성

본 논문에서 설계된 비트 프로세서는 고속 데이터 처리를 위해 프로그램 메모리와 데이터 메모리의 버스를 분리하여 설계하였으며(Harvard Architecture), 경제성을 고려하여 프로그램 메모리와 데이터 메모리를 각각 8 비트의 구조로 설계하였으며 중소형의 시퀀스 로직에 적합한 메모리 구조를 위해 어드레스는 32 Kbyte의 공간을 가지도록 하였다. 또한, 본 FPGA의 내부에는 프로그램 메모리 인터페이스부, 데이터 메모리 인터페이스부, 입출력(I/O) 인터페이스부, 인스트럭션 페치 및 디코더부, 레지스터 및 ALU부, 프로그램 카운터부, 클럭 발생부, 디버그 콘트롤부 등으로 각각 구성되며, 내부의 구성도는 그림 2와 같다.

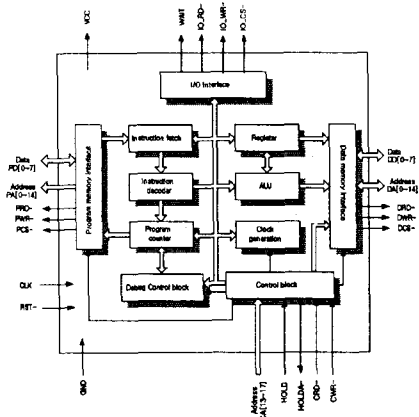


그림 2. 비트 프로세서의 내부 구성도

클럭 발생부는 외부로부터 40MHz 클럭을 입력받아 그림 3과 같이 4개의 상태(Q1, Q2, Q3, Q4)로 분주되며 프로세서 내부처리 명령어와 외부 메모리에 대한 입력명령의 경우에는 T1~T2의 사이클로 분주되고 외부메모리의 입출력명령어의 경우에는 T1~T4로 각각 분주된다. 또한, 외부메모리의 필스처리 명령어 그룹의 경우에는 T1~T8로 각각 분주된다.

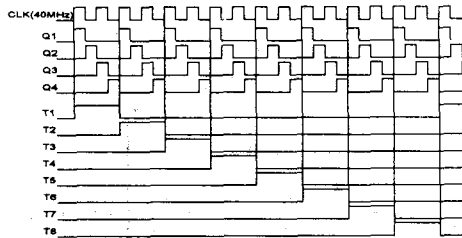


그림 3. 클럭 발생부의 타임차트

프로그램 메모리 인터페이스부는 16비트의 어드레스 버스와 8비트의 데이터버스로 구성된다. 여기서 데이터 버스는 메모리의 수를 줄이고 경제성을 고려하기 위해 8비트로 하였으며, 선택신호와 읽기/쓰기 제어신호가 연결되어 있다.

또한 인스트럭션 페치부에서는 그림 4와 같이 프로그램 메모리로부터 명령어를 페치한 정보를 일시 저장하기 위한 것으로써 8비트의 Pre-IRH라는 프리페치 인스트럭션 레지스터가 있고 이 레지스터는 고속 수행을 위한 구조로써 데이터 메모리를 처리하는 도중에 미리 프로그램 메모리로부터 데이터를 래치하며 T1의 상승에지에서 32비트의 인스트럭션 레지스터에 래치하고 바로 이어 디코더 블록에서 각각의 명령어에 대한 분석을 수행한다.

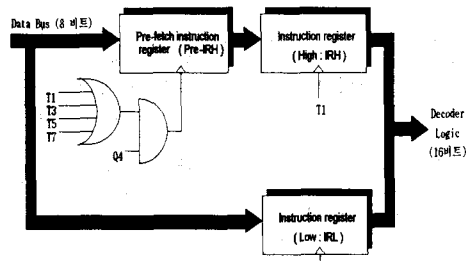


그림 4. 인스트럭션 페치 레지스터의 내부 구성도

## 2.2.2. 비트 프로세서 명령어 및 기능

설계된 비트 프로세서의 명령어 및 기능 그리고 실제 PLC에서 사용되는 래더 다이어그램 기호는 표 1과 같다.

표 1. 비트 프로세서 명령어 및 기능

명령어	기능	기호	구분
NOP	무처리		내부처리 명령어
NOT	논리부정		
MC	마스터 콘트롤		
MCR	마스터 콘트롤		
MPS	분기개시 (PUSH)		
MRD	분기 (READ)		
MPP	분기종료 (POP)		
ANB	논리블럭간 AND		
ORB	논리블럭간 OR		
END	시퀀스 프로그램종료		
LD	논리연산		외부메모리 입력명령
LDI	논리부정연산		
AND	논리곱		
ANI	논리곱부정		
OR	논리합		
ORI	논리합부정		외부메모리 출력명령
PLS	1스캔의 상승펄스		
PLF	1스캔의 하강펄스		
OUT	점점출력		외부메모리 출력명령
SET	점점ON		
RST	점점OFF		

## 2.3 합성 및 시뮬레이션

본 논문에서 제안된 각각의 모듈을 톱다운 방식으로 프로그램 메모리 및 데이터 메모리 인터페이스부, 인스트럭션페치 및 디코더부, I/O 인터페이스부, 비트 ALU 및 레지스터부, 콘트롤 블록 등 각각의 모듈을 설계하여 VHDL<sup>(1)</sup>로 기술하였다. 이와 같이 기술된 VHDL을 퀵 로직(Quick Logic)<sup>(2)</sup>사에서 제공되는 Synplify-Lite의 로직합성 툴<sup>(3)</sup>을 이용하여 로직을 합성하였으며 핀의 형태는 84핀 PLCC 형태이며 또한, 최고속도 40 MHz를 갖고 7,000 게이트에 해당되는 QL2007-2PF84C를 사용하였다. 이때 배치 및 배선의 과정을 거쳤다. 또한 FPGA의 사용율은 480개의 셀중 366개를 사용함으로써 76.3%가 뒀을 확인하였다.

아울러 테스트 벡터를 원활히 작성하고 시스템 레벨상에서 시뮬레이션을 하기 위해 Verilog HDL<sup>(4)</sup>을 사용하여 테스트 벡터를 작성한 후 포스트 시뮬레이션(post simulation)을 하여 각각의 시퀀스 명령어에 대한 동작이 정상적으로 수행됨을 확인하였다.

## 2.4 실험 및 결과

설계된 FPGA에 대한 타당성을 검토하며 또한, 이 FPGA를 이용한 프로그램어블 로직 컨트롤러를 설계하여 FPGA의 성능을 실험을 통해 평가하고자한다.

아래의 그림 5와 그림 6은 파형 오른쪽쪽의 명령을 수행 시 발생하는 외부 메모리 제어 신호들을 나타낸다. 그림 5는 위에서부터 각각 HOLDA-, DWR-, DRD-, DA0에 대한 파형을, 그림 6은 HOLD, HOLDA-, DRD-, DWR- 신호에 대한 파형을 각각 나타낸다.

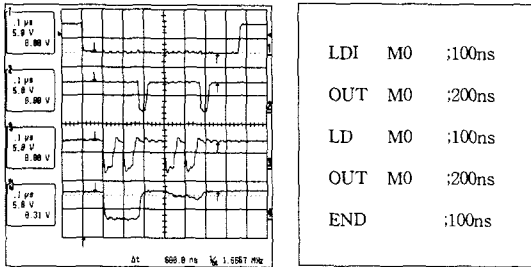


그림 5. 읽기와 쓰기 및 데이터에 대한 실험 결과

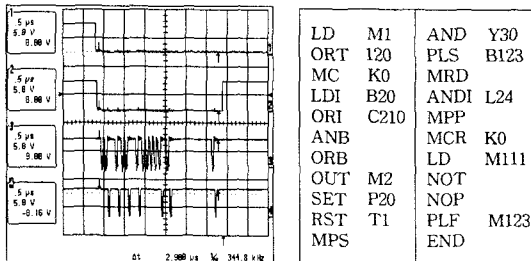


그림 6. 전체 명령에 대한 실험 결과

표 2. 비트 프로세서와 DSP의 명령 수행시간 비교

명령어	본 논문에서 설계된 프로세서	DSP를 이용한 경우	명령어	본 논문에서 설계된 프로세서	DSP를 이용한 경우
LD	100ns	1.9us	PLS	400ns	3.4us
LDI	100ns	1.9us	PLF	400ns	3.4us
OR	100ns	1.9us	NOP	100ns	1.3us
ORI	100ns	1.9us	NOT	100ns	1.3us
AND	100ns	1.9us	ANB	100ns	1.9us
ANDI	100ns	1.9us	ORB	100ns	1.6us
OUT	200ns	2.0us	MPP	100ns	1.6us
SET	200ns	1.8us	MRD	100ns	1.6us
RST	200ns	1.8us	MPS	100ns	1.6us
MC	100ns	1.6us	END	100ns	2.9us
MCR	100ns	1.9us			

프로그램 메모리와 데이터 메모리가 각각 32비트 구조를 갖는 DSP(TMS320C32-40MHz)와 벤치마크를 하기 위해 비트 프로세서와 동일한 동작을 수행하도록 어셈블러<sup>[10]</sup>를 이용해 DSP용 프로그램을 작성하였고, 동일한 40MHz의 클럭을 인가하여 각각 명령에 대한 수행 시간을 측정하였다. 여기서 비트 프로세서에서 비트 결과를 임시로 저장하는 내부 레지스터를 구성하는데

있어서, DSP의 특성상 이를 32비트 메모리 공간에 1비트 결과를 저장하도록 구성하여 해당 비트를 마스크 하는데 걸리는 시간소비를 최소화하였다. 주요 명령에 대한 벤치마크 결과는 표 2와 같다. 표 2의 결과에서 확인할 수 있듯이 각각의 명령을 수행하는데 있어서 비트 명령에 최적화 되어있지 않은 DSP는 필요한 비트 데이터를 획득하기 위한 오버헤드로 많은 시간을 소비했으며, 그 결과 본 논문에서 설계된 비트 프로세서에 비해 10~20배 정도 수행 시간이 더 필요하게 됨을 확인할 수 있었다.

## 3. 결 론

본 논문에서는 FPGA를 이용하여 PLC의 시퀀스 제어 위한 비트 프로세서를 구현하였다. 이를 위해 퀵로직(Quick Logic)<sup>[3]</sup>사에서 제공하는 VHDL 설계틀인 pASIC 2 SpDE<sup>[5]</sup>를 이용하여 Verilog HDL<sup>[6]</sup> 환경에서 시뮬레이션이 성공적으로 수행되었으며, 84핀 PLCC 형태의 FPGA로 프로그래밍 하여 동일한 동작을 수행하도록 프로그램 되어진 DSP(TMS320C32-40MHz)와 수행시간을 비교하였다. 본 논문의 실험 결과에서와 같이 산업 현장에서의 시퀀스 제어 및 공장 자동화 등과 같은 특정 응용 분야에서 쓰이는 비트 단위의 데이터를 처리하는데 있어서 일반 프로세서 및 DSP는 많은 오버헤드를 가지게 된다. 그러므로 본 논문에서 구현된 비트 프로세서를 빠른 정수연산 및 부동 소수점 연산이 가능한 DSP와 병렬 처리를 수행하도록 구성한다면 PLC의 시퀀스 로직 컨트롤을 위한 비트 연산과 수치 연산에 있어서 상당한 처리 속도의 향상이 있을 것으로 기대된다.

### [참 고 문 헌]

- [1] R. Lipsett, C. Schaefer, "VHDL : Hardware Description and Design," KALA, 1991.
- [2] Quick Logic, "Quick Works User's Guide with SpDE Reference," 1996.
- [3] Quick Logic, "Synplify-Light Verilog and VHDL Synthesis User's Guide for Quick Works version 5.1," 1995.
- [4] Quick Logic, "The VERILOG Golden REFERENCE Guide," 1997.
- [5] Texas Instruments, "TMS320C3X User's Guide," 1997.
- [6] 양 오, "프로그램어블 로직 컨트롤러(PLC)의 이론과 실습," 북두출판사, 1999.
- [7] N. Aramaki, Y. Shimokawa, S. Kuno, T. Saitoh, H. Hashimoto, "A New Architecture for High-Performance Programmable Logic Controller," in Proceeding of the 23th International Conference on Industrial Electronics, Control, and Instrumentation, Vol. 1, pp.187-190, 1997.
- [8] K. Koo, W. H. Kown, "Predicting Execution Time of Relay Ladder Logic for Programmable Logic Controllers," in Proceeding of the 1996 IEEE Conference on Emerging Technologies and Factory Automation, Vol.2, pp.670-676, 1996.
- [9] 안재봉, "PLC응용기술핸드북," 도서출판 기술, 1993.
- [10] Texas Instruments, "TMS320C3x/C4x Assembly Language Tools," 1997.