

## 포물선 가감속 패턴을 가지는 정밀 펄스 모터 콘트롤러 칩의 설계 및 제작

원종백, 최성혁, 김종은, 박종식  
경북대학교 전자공학과

### Design and Implementation of Parabolic Speed Pattern Generation Pulse Motor Control Chip

Jong-Baek Won, Sung-Hyuk Choi, Jong-Eun Kim, Jong-Sik Park  
Department of Electronics Engineering, Kyungpook National University

**Abstract** - In this paper, we designed and implemented a precise pulse motor control chip that generates the parabolic speed pattern. This chip can control step motor[1], DC servo[2] and AC servo motors at high speed and precisely. It can reduce the mechanical vibration to the minimum at the change point of a degree of acceleration. Because the parabolic speed pattern has the continuous acceleration change.

In this paper, we present the pulse generation algorithm and the parabolic pattern speed generation. We verify these algorithm using visual C++.

We designed this chip with VHDL(Verilog Very High Speed Integrated Circuit Hardware Description Language) and executed a logic simulation and synthesis using Synopsys synthesis tool. We executed the pre-layout simulation and post-layout simulation with Verilog-XL simulation tool. This chip was produced with 100 pins, PQFP package by 0.35 um CMOS process and implemented by completely digital logic.

We developed the hardware test board and test program using visual C++. We verify the performance of this chip by driving the servo motor and the function by GUI(Graphic User Interface) environment.

### 1. 서 론

현대 고속 정밀 메카트로닉스 장비에서는 고속·정밀한 제어를 요구하고 있으며, 산업용 로봇, CNC 선반 및 반도체 장비 등을 고속으로 정밀하게 제어하기 위해서는 전용 콘트롤러를 이용한 실시간 제어가 필요하다.

본 논문에서는 메카트로닉스 장비에 사용되는 고속 펄스 모터를 제어하기 위한 정밀 펄스 모터 콘트롤러를 설계 및 제작하였다. 기본적으로 모터 콘트롤러 칩은 아주 빠른 속도로 특정한 주파수를 갖는 펄스열을 생성시키는 기능을 가진다. 모터 콘트롤러 칩은 펄스열을 만들기 위한 하드웨어와 이 회로에 설정치를 기억시키기 위한 기억장소를 가지고 있으며, 마이크로프로세서와 연결시키기 위한 포트들로 구성되어 있다. 정지한 상태의 모터를 원하는 속도로 구동하기 위해서는 일정한 가속도로 구동 속도까지 가속하여 구동하여야 하며, 구동 중인 모터를

정지하기 위해서는 일정한 가속도로 정지속도까지 감속시켜야 한다. 이러한 구동시작/정지 시에는 가속도 값을 작게 하여 모터에서 발생하는 기계적인 진동을 최소화하지 않으면 정밀한 제어를 요구하는 반도체 장비 등을 제어하기가 어렵다.

본 논문에서는 먼저 구동 속도에 해당하는 펄스를 출력하는 알고리즘과 포물선 패턴 가감속 알고리즘을 제시한다. 그리고, 이에 따라 설계한 모터 콘트롤러 칩은 구동시작/정지 시에 최소한의 가속도 값을 가질 수 있도록 포물선(S-curve) 패턴의 속도 프로파일로 가/감속 가능하도록 하였다. 그리고, 모터 구동 시 필요한 센서 신호에 의한 실시간 동작이 가능하며, 엔코더 2상 신호를 카운트하는 32-bit 카운터를 내장하고 있어서, 기계적인 위치를 저장할 수 있도록 하였다.

### 2. 본 론

#### 2.1 펄스 발생 알고리즘

모터 제어에 필요한 펄스를 프로그램된 속도로 출력하기 위해서 다음과 같은 펄스 발생 알고리즘을 사용하였다.

그림 1에서 보는 것과 같이 일정한 속도( $OBJ\_Vp$ )를 시간에 대하여 적분하면 그 값이 발생된 펄스 수와 같다.

$$n(0:t) = \int_0^t v_p(t') dt' \quad (1)$$

여기서,  $n(0:t)$ 은 펄스 출력 시작 후 현재시간  $t$ 까지 발생한 펄스의 합이 되며,  $v_p(t')$ 은 시간  $t'$ 에서 발생하는 펄스의 속도를 나타낸다.

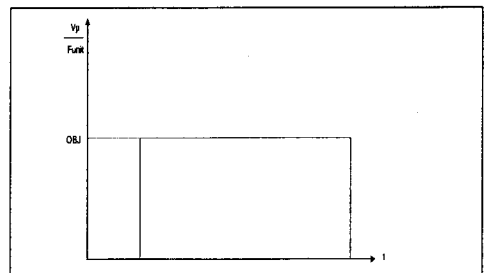


그림 1. 일정 속도 펄스 출력

계산의 편의를 위해서 시간( $t$ )은 칩의 동작 클럭 주기 카운트값을 사용한다. vp를 펄스 출력 속도(OBJ\_Vp)와 펄스 출력 주파수 단위(Funit(Hz))의 곱을 나타내면 다음과 같다.

$$v_p = OBJ\_Vp \times F_{UNIT} [pps] \quad (2)$$

식 (2)에서 사용하는 속도 단위는 단위시간당 출력된 펄스 수(Pulse Per Second)이며, 펄스 출력 속도를 1에서 65,535까지 지정 가능할 때  $F_{UNIT}$ 을 나타내면 다음과 같이 표시된다.

$$F_{UNIT} = \frac{f_{clk}}{RANGE \times 262,144} \approx \frac{f_{clk}}{RANGE \times 256K} \quad (3)$$

여기서,  $f_{clk}$ 은 동작 클럭 주파수이며  $RANGE$ 값은 펄스 출력의 배수값이다.

(1), (2), (3)을 정리하면 다음과 같이 표현된다.

$$n(0:t) = \int_0^t \frac{OBJ}{RANGE \times 256K} f_{clk} dt \approx \frac{OBJ}{RANGE \times 256K} \frac{t}{t_{clk}} \quad (4)$$

(4)는 모터의 동작 속도와 이에 해당하는 펄스 발생 수와의 관계를 나타낸다.

## 2.2 포물선(S-curve) 패턴 가감속 알고리즘

매카트로닉스 장비의 모터 제어는 고속 동작 제어도 중요하지만 이동 방향을 변경하기 위한 가감속도 제어가 작동하는 기계의 관성 및 마찰 등을 최소화하기 위해 절대적으로 필요하다. 속도 제어는 간단히 마이크로프로세서를 이용한 온/오프 제어, 직선 가감속 제어, 포물선 가감속 제어가 있다.

가감속이 없는 단순한 온/오프 제어는 고속 복잡한 제어에는 사용할 수 없으며 직선 가감속 제어는 고속 정밀한 제어에 사용할 경우 가속/감속의 시작과 끝에 가속도 변화가 크므로 진동이 발생할 확률이 크다.

본 논문에서는 고속 정밀 제어에 사용 가능한 포물선(S-curve)패턴 가감속 알고리즘을 연구하여 이를 응용한 정밀 모터 제어칩을 설계, 제작하였다.

포물선 패턴 가감속 그래프는 그림 2와 같다.

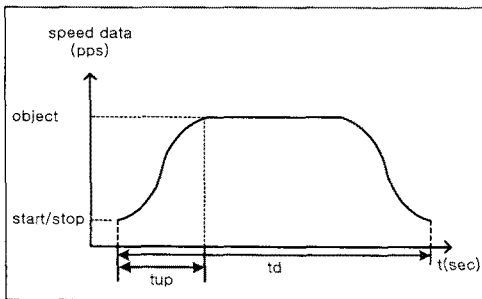


그림 2. 포물선 패턴 가감속 그래프

$t_d$ 는 펄스 출력 드라이브 구동 시간을 나타내고,  $t_{up}$ 은 가속 시간을 표시한다. start/stop은 시작 속도 데이터를 object는 구동 속도 데이터이다. 그림 2에서 가속 시간( $t_{up}$ )은 다음과 같다.

$$t_{up} = 8(object - start/stop) \times rate \times t_{clk} \quad (5)$$

rate 데이터는 가감속 시간을 결정하는 데이터이다.

그림 2와 같은 속도 패턴으로 가감속 할 경우 가속도는 그림 3과 같이 표시된다.

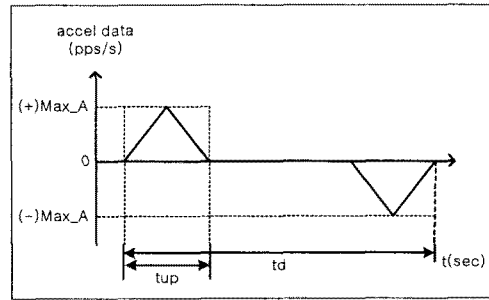


그림 3. 포물선 가감속의 가속도 그래프

Max\_A를 가감속 시 최대 가속도 값을 표시하며 그 값은 다음과 같다.

$$MAX\_A = \frac{1}{64rate^2 \times (object - \star t/stop)} \quad (6)$$

그림 2와 그림 3의 관계는 서로 미분과 적분의 관계이다. 즉, 그림 2의 포물선 패턴의 속도 가감속을 구현하기 위해서는 가속도 그래프를 적분함으로써 가능하다.

가감속 중 가속도는 직선으로 증감하고 등속 구간에서는 가속도 데이터는 0이 된다. 이와 같이 가속/등속/감속 시 속도 기울기인 가속도가 연속적으로 변화함으로써 속도 변경 부분에서 발생할 수 있는 기계적인 진동을 최소화 할 수 있다.

이와 같은 포물선 가감속 알고리즘의 구현을 위한 시뮬레이터 구동 화면은 그림 4와 같다.

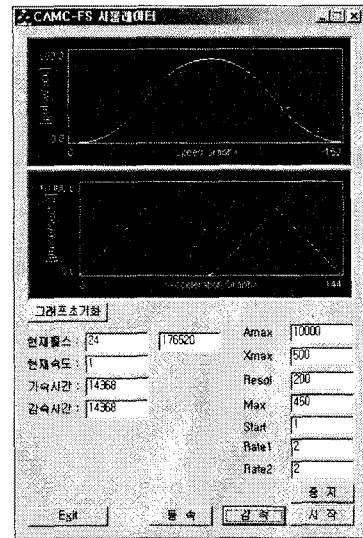


그림 4. 포물선 가감속 시뮬레이터

그림 4에서는 시작 속도 데이터(Start)는 1, 구동 속도 데이터(Max)는 450으로 설정하였으며 가감속 시간을 설정하는 데이터(Rate1, Rate2)는 2로 설정한 후 가감속 제어를 시뮬레이션 하였다. 시뮬레이션 결과 가속시간과 감속 시간은 식(5)의 계산 결과와 같이 14368(단위 = 구동 클럭 수)이 되었으며 전체 출력

펄스 수는 속도 그래프를 적분한 값인 24가 출력되었다. 가속도 그래프는 절대값으로 출력하였다.

## 2.4 모터 컨트롤러 칩의 설계

본 논문에서는 펄스 발생 알고리즘과 포물선 패턴 가감속 알고리즘 제시하였으며 이를 이용하여 고속 정밀 제어 가능한 모터 제어칩을 설계 하였다. 본 논문에서 설계한 칩의 기능 블록도는 그림 5와 같다.

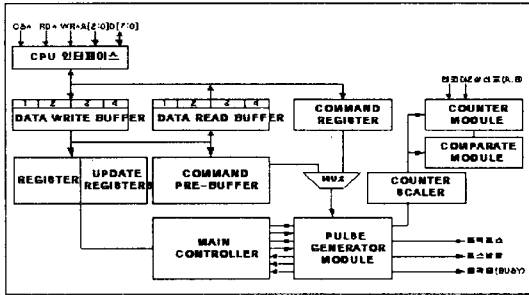


그림 5. 모터 제어칩 전체 기능 블록도

CPU 인터페이스 블록은 외부 마이크로프로세서와 데이터 전송 기능을 담당한다. 마이크로프로세서에서 입출력하는 데이터는 DATA WRITE BUFFER와 DATA READ BUFFER를 통해서 액세스하게 된다. 입력된 데이터는 내부 명령어에 의해서 REGISTER 블록에 저장된다.

PULSE GENERATOR MODULE은 2.1과 2.2에서 기술한 알고리즘을 이용하여 가감속도 데이터를 출력하고 그 속도에 해당하는 모터 구동 펄스를 출력하는 블록이다.

본 논문에서 설계한 모터 제어칩에는 32 비트 내부 위치 카운터와 32 비트 외부 위치 카운터가 있어 이론적인 모터 위치와 실제 모터가 구동한 위치를 서로 비교할 수 있다.

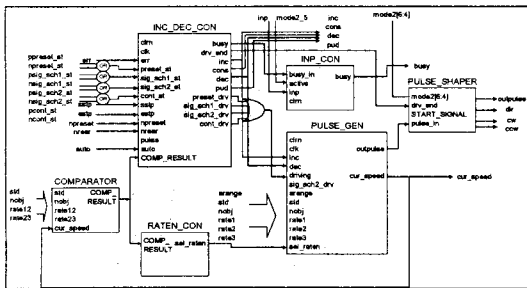


그림 6. 펄스 발생 기능 블록도

펄스 발생 기능 블록은 포물선 가감속 및 펄스 출력 알고리즘을 구현한 PULSE\_GEN 블록과 가감속도 제어 신호 발생 블록(INC\_DEC\_CON)과 기타 기능 블록들로 구성되어 있다. PULSE\_GEN 블록은 속도 레지스터로부터의 속도 데이터를 입력 받아 펄스 출력 구동 신호가 입력되면 2.1과 2.2에서 기술한 알고리즘으로 펄스를 출력한다. INC\_DEC\_CON 블록은 여러 가지 드라이브 구동 시작 신호를 입력 받아 가감속 제어 신호를 출력하고 출력 펄스의 방향을 결정하는 제어 신호를 출력하는 블록이다. COMPARE 블록은 현재 속도 데이터와 설정한 속도 데이터와의 비교 결과를 출력한다. INP\_CON 블록은 서보 모터 신호 In-Position(position 종료)신호에 의한 드라이브 종료 기능을 수행하는 블록이다. PULSE\_SHAPER블록은

여러 가지 사양의 모터 드라이버를 외부 회로 첨가 없이 프로그램적으로 수정 가능하도록 출력 펄스와 방향 신호를 변경하는 기능 블록이다.

## 2.5 설계된 칩의 시뮬레이션

본 논문에서 설계한 모터 제어칩은 VHDL로 설계하여 로직 시뮬레이션 및 합성 후 타이밍 시뮬레이션 결과를 확인하여 기능 동작 검증을 수행하였다.

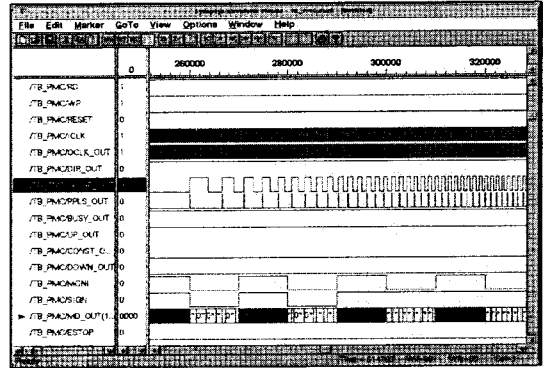


그림 7. 출력 펄스 가속 기능 시뮬레이션

속도 데이터가 가속하면서 해당 출력 펄스의 속도도 증가하는 기능을 시뮬레이션한 결과를 그림 7에서 표시하였다.

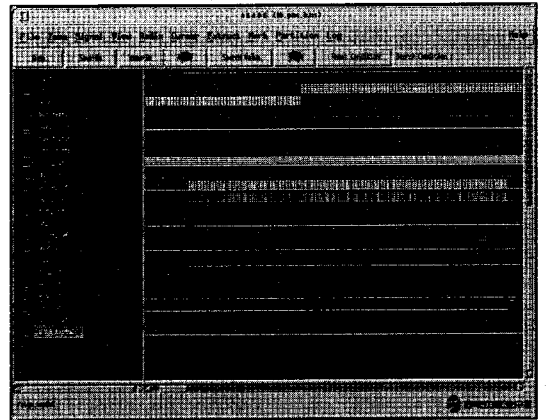


그림 8. 지정 펄스 수 드라이브 시뮬레이션

본 논문에서 설계한 모터 제어칩은 구동 펄스 수를 지정 한 후 지정한 펄스 수 만큼 자동으로 가감속 하면서 모터 구동 펄스를 출력하는 지정 펄스 수 드라이브 명령이 있다. 지정 펄스 수 드라이브 명령에 의한 20,000 펄스 구동 실행 시뮬레이션 결과를 그림 8에서 표시하였다.

## 2.6 칩의 성능 실험 및 결과

본 논문에서 설계한 정밀 모터 제어칩은 VHDL (Very High Speed Integrated Circuit Hardware Description Language)을 이용하여 설계하였고, 0.35 um CMOS 공정을 이용하여 제작하였다. 제작하기 위하여 Synopsys 합성기를 이용하여 회로를 합성하

였으며, 구동 주파수 33MHz, I/O 전압 4.75 V, 코어 전압 3.25 V, 동작 온도가 70°C인 worst case 환경까지 최종 시뮬레이션을 수행하여 원하는 제어 성능을 검증하였다. 제작된 칩은 230,000 게이트를 가지며, 100핀 PQFP 패키지를 사용하여 제작하였다.

제작된 칩의 성능을 확인하기 위하여 칩 성능 측정 보드를 개발하였다. 구동 주파수는 10MHz, 16.384MHz, 20MHz 그리고 32.768MHz로 변경하여 사용하였으며, IBM PC의 ISA 슬롯에 장착하여 본 논문에서 설계한 칩의 기능을 확인하였다.

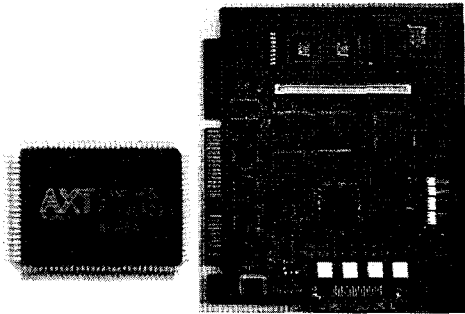


그림 9. 제작된 칩과 테스트 보드

제작된 칩과 칩 성능 검증을 위한 테스트 보드를 그림 9에서 표시하였다.

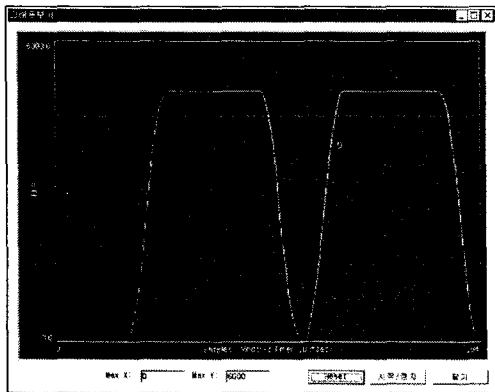


그림 10. 지정 펄스수 드라이브 구동 결과

그림 10에서는 테스트 보드를 이용하여 5,000,000개 지정 펄스 수 드라이브 명령을 실행하여 자동으로 시작 속도에서 구동 속도로 가감속 동작을 수행하는 결과를 표시한다.

### 3. 결 론

본 논문에서는 메카트로닉스 장비에 사용되는 고속 펄스 모터를 제어하기 위한 정밀 펄스 모터 콘트롤러를 설계 및 제작하였다. 먼저, 고속 정밀한 모터 제어를 위한 포물선 가감속 프로파일을 출력하는 알고리즘을 제시하였으며, 현재 속도 데이터에 해당하는 펄스를 출력하는 알고리즘을 제시하였다. 이 알고리즘을 이용한 모터 구동 펄스 출력 기능 블록을 포함한 정밀 모터 제어칩을 VHDL로 설계한 후 Synopsys 합성기를 이용하여 회로 합성을 실시하였으며, 논리 시뮬레이션 및 타이밍 시뮬레이션으로 설계 기능 검증을 수행하였다. 0.35 um CMOS 공정을 이용하여 100핀 PQFP 패키지로 제작하였다.

제작한 칩은 테스트 보드를 제작하여 직접 서보 모터를 구동하여 칩의 구동 성능을 확인하였다.

메카트로닉스 장비에는 가속시간과 감속 시간을 서로 달리해야하는 경우도 있다. 이와 같은 장비에 적용하기 위해 가속/감속 시 서로 다른 가속도 기울기를 가지면서 속도 및 위치 제어가 가능한 알고리즘 및 회로에 대한 연구가 있어야 할 것이다.

### [참 고 문 헌]

- [1] A. C. Leenhouts, "Art an practice of stepmotor control", 1987.
- [2] Y. Dote, "Servo motor and motion control : using digital signal processors", Prentice Hall, 1990.
- [3] M. A. Bayoumi, VLSI Design Methodologies for Digital Signal Processing Architecture, Kluwer Academic Publishers, /Boston/Dordrecht/London, 1994.
- [4] "PCL-832 : 3-axis servo motor control card, command library and user's manual", Advantech, 1998.
- [5] "General purpose motion control ICs : HCTL-1100 series technical data", Hewlett Packadd, 1996.
- [6] "LM628/LM629 precision motion controller application note", National Semiconductor, 1997.
- [7] "New high torque 5 phase geared step motor with compact driver", Oriental Motor U.S.A Corp., 1998.
- [8] "stepping motor controller PC card M20", Spindler&Hoyer GmbH, 1998.
- [9] "2 axis, 2-D controller for stepper motor driven machines", TOMOMOTEK Product Information Sheet, 1998.
- [10] "TEKCEL motion controller", Tekcel, 1997.
- [11] "PC/104 module : AIM104-MOTION-1", Arcrom.
- [12] "PIC-SERVO motion control chipset", JR KERR
- [13] "EMC-XYZ-Parallel port CND stepper motor controller user's manual", Super-Tech, 1997.
- [14] "Synthesis shorcut user manual", Ver. 3.4b, Synopsys, 1997.
- [15] "(V)HDL compiler reference manual", Ver.3.4a, Synopsys, 1996.
- [16] "Design compiler tutorial", Ver. 3.4, Synopsys, 1996.
- [17] J. R. Armstron, F. Gail Gray, Structured Logic Design With VHDL, Prentice Hall, New Jersey, 1993.
- [18] N. H. E. Wste, Kamran Eshrhgian, Principles of CMOS VLSI Design, Addison-Wesley, 1993.
- [19] Wayne Wolf, Modern VLSI Design A Systems Approach, PTR Prentice Hall, 1994.
- [20] "XILINX foundation series V1.4", XLIMIX, 1998.
- [21] "CAMC-5M user manual", (주)아진엑스텍, 1999.
- [22] 이보희, 김진걸, "DSP를 이용한 로봇 제어시스템 개발", 제어·자동화·시스템공학 논문지, 제 1권, 제 1호, pp. 50-57, 1995년 9월.
- [23] 김종환, 전정열, "실험적 진화 프로그래밍을 이용한 초정밀 위치 제어", 전자공학회지, vol. 22, no. 11, pp. 85-95, 1995.
- [24] Robert J. Schilling, Fundamentals of Robotics Analysis & Control, Prentice-Hall, 1990.