

工學碩士學位論文

TMS320VC33을 이용한 고성능 서보모터
제어용 주제어기 설계

2002年 8月

昌	原	大	學	校	大	學	院
電		子	工	學	學	科	
尹			聖			恩	

工學碩士學位論文

**TMS320VC33을 이용한 고성능 서보모터
제어용 주제어기 설계**

**Main Controller Design applied TMS320VC33 for High
Performance Servo Motor Control**

指導教授 朴 勝 燁

이 論文을 工學碩士學位論文으로 提出함

2002年 8月

昌 原 大 學 校 大 學 院
電 子 工 學 科
尹 聖 恩

尹聖恩의 碩士學位 論文을 認准함.

審査委員長 朴 茂 薰(인)

審査委員 高 鳳 震(인)

審査委員 朴 塋 燁(인)

2002年 8月

昌原大學校 大學院

工學碩士學位論文

**TMS320VC33을 이용한 고성능 서보모터
제어용 주제어기 설계**

**Main Controller Design applied TMS320VC33 for High
Performance Servo Motor Control**

2002年 8月

昌 原 大 學 校 大 學 院
電 子 工 學 科
尹 聖 恩

TMS320VC33을 이용한 고성능 서보모터 제어용 주제어기 설계

요 약

본 논문은 TI 사의 DSP인 TMS320VC33를 이용하여 고성능 서보모터 제어용 주제어기를 설계하였다. 주제어기 회로는 전류를 센싱하기 위한 ADC 회로, 신호를 확인하기 위한 DAC회로, SVPWM을 위한 control signal transmitter와 receiver, 엔코더 종류별로 수신할 수 있게 한 encoder interface 회로, 실시간으로 소프트웨어의 변수들의 변화를 모니터링 하기 위한 serial and parallel communication회로, 제어기 내에 필요한 DC 전원을 regulator하는 전원회로, 초기 전원 인가시나 전원의 fail 될 때 DSP와 주변 장치를 초기화 시키는 reset와 power supervisor 회로들로 구성되어 있다. 또한 모터 구동하기 위해 필요한 주변회로들을 EP1K100QC-208이라는 ALTERA사의 ACEX1K계열의 칩을 이용하여 VHDL로 설계하였다.

VHDL로 설계한 주변회로들을 살펴보면 공간벡터기법을 이용하여 모터를 제어하기 위해 SVPWM signal generator, 엔코더 신호를 M/T method를 통해 위치와 속도를 계산하는 encoder signal generator, 마이크로프로세서의 supervisor인 watch dog timer, external interrupt를 더 많이 받을 수 있도록 처리해주는 expanded interrupt controller, 주변 장치와의 속도가 달라 타이밍을 맞추기 위한 wait signal generator, 외부 주변 장치인 ADC에 제어 신호를 보낼 수 있도록 한 ADC interface control signal generator, 외부 주변 장치인 DAC에 제어 신호를 보낼 수 있도록 한 DAC interface control signal generator, serial로 모니터링을 하기 위해 제어신호를 만드는 serial communication control signal generator, parallel로 모니터링을 하기 위한 제

어 신호를 만드는 parallel communication control signal generator , FPGA 의 프로그램을 초기화 시키고 DSP로 Reset 신호를 보내어주는 reset signal generator가 있다. 그리고 각각의 모듈별로 성능을 시뮬레이션을 하였고 또한 실험을 통해 입증하였다. 그리고 주제어기를 4 layer PCB 기판으로 설계 및 제작하였다.

목 차

한글 요약문	i
목 차	iii
그림 목차	vi
표 목차	ix
약 어	x
I. 서론	1
II. 서보 모터 구동을 위한 시스템	3
2.1 전체 구성도	3
2.2 주 제어기	4
2.2.1 마이크로프로세서	7
2.2.2 FPGA	7
2.2.3 ADC	8
2.2.4 Control signal transmitter/receive	8
2.2.5 엔코드 및 엔코드 수신부	8
2.3 전력 변환부	10
2.3.1 정류기	10
2.3.2 PFC	11
2.3.3 Inverter	12
2.3.4 전류센서	13
2.4 서보 모터	14
III. 주요 device 소개	16

3.1 TMS320VC33	16
3.1.1 TMS320VC33 의 내부	17
3.1.2 Boot Loader	17
3.1.2.1 마이크로프로세서 모드	17
3.1.2.2 마이크로컴퓨터 모드	20
3.1.3 내부 주변 장치	22
3.1.3.1 타이머	22
3.1.3.2 시리얼 통신	24
3.2 FPGA	28
3.2.1 종류	29
3.2.2 CPLD & FPGA 비교	31
3.3 ADC	33
IV. 제어보드 설계	37
4.1 마이크로프로세서	37
4.1.1 프로그램 다운로드	37
4.1.2 전기적 특성	38
4.1.3 PLL	39
4.2 FPGA	42
4.3 Analog to Digital Converter	47
4.4 Digital to Analog Converter	50
4.5 Control signal transmitter/receiver	53
4.6 Encoder Interface	57
4.7 전원	61
4.8 Reset & Power supervisor	64
V. VHDL에 의한 FPGA 설계	67

5.1 전체 설계부	67
5.2 SVPWM	68
5.3 Encoder Signal Processor	72
5.4 Watchdog Timer	75
5.5 Expanded Interrupt Controller	78
5.6 Wait Signal Generator	83
5.7 ADC Interface Control Signal Generator	85
5.8 DAC Interface Control Signal Generator	86
5.9 Serial Communication Interface Control Signal Generator	87
5.10 Parallel Communication Interface Control Signal Generator	88
5.11 Reset Signal Generator	88
VI. 실험 결과 및 시뮬레이션	89
6.1 PWM Generating Simulation	89
6.2 Encoder Signal Processing Simulation	90
6.3 Interrupt/External Interrupt Control Simulation	91
6.4 Watch dog Timer Simulation	92
6.5 Wait Signal Generating Simulation	93
6.6 주제어기 출력 파형	94
6.7 제작된 주제어기	96
VII. 결론	97
참고 문헌	100
초록(abstract)	102
감사의 글	104

그림 목차

그림 2-1 서보모터 구동을 위한 전체 구성도	3
그림 2-2 주 제어기부의 블록도	6
그림 2-3 정류회로 블록도	10
그림 2-4 부하에 따른 전압과 전류의 위상차	11
그림 2-5 Boost형 PFC회로	11
그림 2-6 Inverter 내부 구성도	13
그림 3-1 내부 기능 블록도	18
그림 3-2 메모리 맵	19
그림 3-3 타이머 내부 블록도	22
그림 3-4 시리얼 포트 블록도	23
그림 3-5 Handshake 모드	24
그림 3-6 고정 버스터 모드	25
그림 3-7 고정 Continuous 모드	25
그림 3-8 가변 Continuous 모드	26
그림 3-9 가변 버스트 모드	27
그림 3-10 가변 Continuos 모드	27
그림 3-11 PLD 블록도	31
그림 3-12 디지털 램프 A/D converter	33
그림 3-13 연속 근사 컨버터	34
그림 3-14 SAC 동작 순서	34
그림 3-15 플래쉬 A/D converter	35
그림 3-16 업/다운 A/D converter	35
그림 3-17 Dual slop A/D converter	36
그림 4-1 JTAG 포트 & 시리얼 포트 회로	37
그림 4-2 클릭 발생부	40

그림 4-3 Crystal를 이용한 모드	41
그림 4-4 EP1K100QC-208의 내부 구조	43
그림 4-5 Byteblaster 내부 회로	45
그림 4-6 PS mode와 JTAG mode 회로	46
그림 4-7 ADS7864의 내부 구조	47
그림 4-8 ADS7864 적용회로	48
그림 4-9 A/D 입력 신호 회로	49
그림 4-10 A/D 채널 입력 신호 회로	49
그림 4-11 DAC7864의 내부 구조	50
그림 4-12 DAC7625 적용 회로	51
그림 4-13 DAC 출력 회로	52
그림 4-14 DAC 레퍼런스 전압 발생 회로	52
그림 4-15 HFBR1521/HFBR2521 적용 회로	53
그림 4-16 표준 케이블 사용시 거리당 순방향 전류와의 관계	54
그림 4-17 HFBR1521/HFBR2521 내부 회로	54
그림 4-18 HFBR1521를 이용한 송신회로	56
그림 4-19 HFBR2521를 이용한 수신회로	56
그림 4-20 오픈 콜렉트형 엔코드	57
그림 4-21 전압 출력형 엔코드	57
그림 4-22 Line drive형 엔코드	58
그림 4-23 TTL형 엔코드	58
그림 4-24 Complementary형 엔코드	59
그림 4-25 토템폴형 엔코드	59
그림 4-26 엔코드 입력 회로	60
그림 4-27 +3.3V _{DC} , +1.8V _{DC} 발생 회로	61
그림 4-28 +5V _{DC} , -5V _{DC} 발생회로	62
그림 4-29 DSP 리셋입력 블록도	64

그림 4-30 전체 reset 타이밍도	64
그림 4-31 DSP 리셋 요구 시간	65
그림 4-32 MAX708 내부 블록도	66
그림 4-33 MAX708 리셋 타이밍도	66
그림 4-34 Power supervisor 회로	66
그림 5-1 전체 VHDL 설계 구성도	67
그림 5-2 SVPWM에서 인터럽트 발생 개념도,.....	68
그림 5-3 SVPWM generator 구성도	68
그림 5-4 Encoder signal processor 구성도	72
그림 5-5 Watchdog timer 구성도	75
그림 5-6 인터럽트 수행 순차도	78
그림 5-7 Expanded interrupt controller 구성도	79
그림 5-8 Wait signal generator 구성도	83
그림 5-9 ADC interface control signal generator 구성도,.....	85
그림 5-10 DAC interface control signal generator 구성도	86
그림 5-11 Serial communication interface control signal generator 구성도	87
그림 5-12 Parallel communication interface control signal generator 구성도	88
그림 5-13 Reset signal generator 구성도	88
그림 6-1 SVPWM generating simulation 파형	89
그림 6-2 Encoder signal processing simulation 파형	90
그림 6-3 Internal/external interrupt control simulation 파형	91
그림 6-4 Watchdog timer simulation 파형	92
그림 6-5 Wait signal generating simulation 파형	93
그림 6-6 U/V/W의 PWM 파형	94
그림 6-7 U상의 dead time 파형	95
그림 6-8 제작된 주 제어기	96

표 목차

표 2-1 주제어기 설계 조건	5
표 3-1 바이트폭으로 구성된 메모리	21
표 3-2 CPLD와 FPGA의 비교	32
표 4-1 PLL의 전기적 특성	39
표 4-2 클럭모드 선택 핀	40
표 4-3 일반적인 crystal을 이용한 회로 구성	40
표 4-4 바이트블라스트 25pin 연결 설정	44
표 4-5 바이트블라스트 10핀 연결 설정	45
표 4-6 DAC7625 제어로직 진리표	51
표 4-7 HFBR 종류별 송수신 거리	35
표 4-8 송신기의 전기적/광적 특성	54

약어

ADC	: Analog to Digital Converter
ACIM	: Alternating Current Induction Motor
ASIC	: Application Specific Integrated Circuit
BLDCM	: Brushless Direct Current Motor
CPLD	: Complex Programmable Logic Device
CT	: Current Transformer
CPU	: Central Processing Unit
DAC	: Digital to Analog Converter
DASRAM	: Dual Access Static Random Access Memory
DSP	: Digital Signal Processor
EDA	: Electric Design Automation
FPGA	: Field Programmable Gate Arrays
HCPLD	: High Capacity Programmable Logic Device
IGBT	: Insulated Gate Bipolar Transistor
JTAG	: Joint Test Action Group
LPF	: Low Pass Filter
MIPS	: Million Instructions Per Second
MFLOPS	: Million Floating-Point Operations Per Second
MOSFET	: Metal Oxide Semiconductor Field Effect Transistor
PFC	: Power Factor Correction
PLD	: Programmable Logic Device
PAL	: Programmable Array Logic
PLA	: Programmable Logic Array

PQFP	: Plastic Quad Flatpackage
PLL	: Phase-Locked Loop
QEP	: Quadrature Encoder Pulse
rpm	: revolution per minute
SMPS	: Switching Mode Power Supply
SPLD	: Simple Programmable Logic Device
SAC	: Successive Approximation Converter
SVPWM	: Space Vector Pulse Width Modulation
SPMSM	: Sinusoidal Permanent Magnet Synchronous Motor
TTL	: Transistor Transistor Logic
TR	: Transistor
VCO	: Voltage Controlled Oscillator
VHDL	: Very High Speed Integrated Circuit Hardware Description Language

I . 서 론

1950년 미국의 GE에서 사이리스터가 발표되어 사이리스터 정류기를 전원으로 한 직류기의 가변속 구동방식이 보급되기 시작한 후, 1970년을 전후로 Hasse 및 Blashk에 의해 'Field Orientation Control'의 개발과 반도체 기술에 의한 전력 디바이스들의 발전으로 인하여 모터 제어기술은 급격히 발전하기 시작하였고[1] 2000년대에 이르러 초고속의 6만rpm이상 10만rpm까지도 달성한 상태이다. 예를 들어, 6만rpm 정도로 회전하는 고속모터를 제어 하려면 일반적으로 4극 모터인 경우에 모터는 초당 1000회전을 하므로 전기적 주파수는 2kHz가 되고, 1cycle 당 걸리는 시간은 0.5ms 되므로 1cycle동안에 최소한 60°간격으로 6번 제어를 한다고 생각하면 약 83.5 μ s 동안 제어알고리즘을 연산시킨 후 제어신호를 생성하여야한다. 그런데 전류 ripple을 줄이기 위하여 제어 사이클을 증가시키면 더욱 짧은 시간동안에 이러한 동작이 이루어져야 한다. 더욱이 센서리스로 모터를 구동할 경우 고성능 위치 추정용 알고리즘 등을 구현하기 위한 프로그램에 의해 소프트웨어의 부담이 한층 증가하므로 더욱 고속의 연산이 필요하게 된다. 이러한 상황에서 마이크로프로세서의 발전은 모터제어기술의 발전에 중요한 공헌을 하였음을 간과할 수 없을 것이다. 1970년대에 마이크로프로세서의 응용이 시작되어 모터제어기술의 가속화에 공헌한 이래, 1990년 말경부터는 Texas Instruments사와 Analog Devices사에 의해 모터제어 전용의 DSP(digital signal processor)까지 출시되었고, 저전압 기술과 더불어 디지털회로 설계분야에서의 ASIC(application specific IC)과 VHDL(very high speed integrated circuit hardware description language : VHSIC HDL)의 출현으로 마이크로프로세서의 고속화와 주변회로의 집적화가 더욱 용이하게 됨으로써 DSP의 모터제어에 적극 도입되고 있다. 이와 같이 모터 제어에 디지털 기술이 적용됨에 따라 대부분의 모터제어에 chopper방식이 적용되고 있는데, 이는 전력량 제어와 효율 면에

서 우수한 반면에 고속 스위칭에 의해 발생하는 전자파의 발생과 전력 파형의 왜곡에 의한 전력효율의 저하라는 단점도 있으므로 환경보호의 차원에서 전자파장애 및 전력 효율에 대한 규제도 가정용 기기로부터 산업용 기기에까지 점점 강화되어가고 있는 실정이다.

본 논문에서는 이러한 조류에 맞추어, 최신의 DSP인 Texas Instruments사의 TMS320VC33-150을 탑재하고 VHDL에 의해 복잡한 주변 디지털회로를 집적화 함으로써, 고성능 모터제어와 동시에 전력효율을 고려한 제어 이론의 구현이 가능한 범용의 모터제어용 주제어기를 설계 및 제작하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 서보모터 구동을 위한 시스템에 대해 전반적으로 알아보고, 제 3장에서는 서보모터 구동을 위해 본 논문에서 사용된 마이크로프로세서인 Texas Instruments사의 TMS320VC33에 대해 기능적, 전기적 특성, FPGA와 ADC의 종류에 대하여 기술하였다. 제 4장에서는 주제어기를 설계하는데 필요한 디지털회로설계 기술과 디지털신호와 물리량과의 접속에 필요한 ADC 및 DAC, 그리고 반도체 디바이스간 노이즈 마진이 상이한 경우의 인터페이스 방법에 관하여 서술하고, 실제적인 하드웨어 설계를 하여 디바이스 및 회로정수들을 결정하는 과정을 보였으며, 제 5장에서는 프로그램 메모리 이외에 모터제어에 필요한 주변 디바이스들을 FPGA(field-programmable gate arrays)로 집적화하고 기타의 주변 디바이스들을 제어하기 위하여 어드레스 디코더(address decoder), 모터제어에 필요한 초퍼신호발생을 위한 공간벡터 PWM 신호 발생기(space vector PWM signal generator)와 확장된 인터럽트 컨트롤러(expanded interrupt controller), 모터의 회전자축에 부착된 회전수 엔코더로부터의 신호를 처리하기 위한 엔코더 신호 처리기(encoder signal processor)등을 VHDL로 설계에 관하여 기술하였다. 제 6장에는 FPGA내부에 설계된 어드레스 디코더, 공간벡터 PWM 신호 발생기, 확장된 인터럽터 제어기, 엔코더 신호 처리기등의 실험 결과 및 시뮬레이션 파형을 나타내었고, 제 7장에는 본 논문의 결론을 기술 하였다.

Ⅱ. 서보모터 구동을 위한 시스템

2.1 전체 구성도

본 논문에서 최종 목표는 TMS320VC33을 이용한 고성능 서보모터 제어용의 주제어기를 설계 및 제작하는 것이다. 서보모터를 구동하기 위한 일반적인 시스템 구조는 그림 2-1과 같이 크게 세 부분으로 나눌 수 있다.

첫째, 주제어기는 마이크로프로세서가 있어서 내부 및 외부에 있는 메모리로부터 제어를 하기 위한 구동 프로그램을 가져와서 실행시킨다. 또한 외부의 센서들로부터 모터의 각 상전류 및 회전수등의 신호를 입력받아 위치제어에 필요한 연산을 하고 그 연산 결과를 모터 입력전력제어용 신호로 변환하여 전력 변환부로 출력을 하게된다. 둘째, 전력 변환부는 주제어기로부터의 제어신호를 받아 서보모터 구동을 위한 전력 변환을 하게된다. 셋째, 제어 대상인 서보모터이다. 그림 2-1은 서보모터 구동을 위한 전체 시스템을 나타내고 있다.

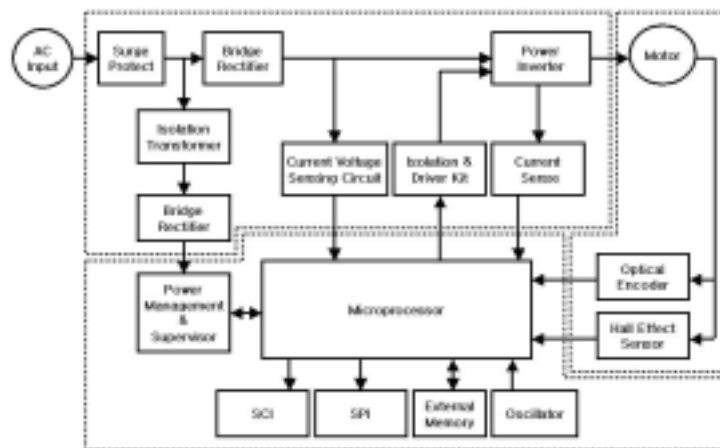


그림 2-1 서보모터 구동을 위한 전체 구성도

Fig. 2-1 Total block diagram for servo motor operating

2.2 주제어기

복잡한 제어 알고리즘을 실시간으로 제어를 하기 위해서는 많은 연산을 빠른 속도로 처리할 수 있어야 하고, 제어 대상인 시스템도 실시간으로 응답해야 한다. 반도체 설계기술이 발전하면서 마이크로프로세서의 성능이 더욱 좋아지면서 새로운 제어이론들을 구현하기 쉽게 되었다. 본 논문에서 설계하고자 하는 주제어기의 마이크로프로세서는 Texas Instruments사에서 최근에 출시된 저전압 기술이 채택되고 75MIPS(150MFLOPS)의 성능을 가진 TMS320VC33-150을 이용하여 고성능 서보모터 제어와 동시에 전력효율을 고려한 제어이론의 구현이 가능하도록 하였다. 설계하고자 하는 주제어기의 설계조건은 표 2-1과 같다. 또한, 표 2-1의 조건을 만족하기 위한 주제어기의 구성은 그림 2-2와 같은데 블록도를 간략히 설명하면, 주제어기용의 여러 가지 DC전원을 공급하기 위한 전원 장치인 SMPS와 voltage regulator, 전원 초기 인가시 및 전원 불안정시 마이크로프로세서를 reset시키기 위한 power supervisor, 제어용 프로그램을 수행하는 마이크로프로세서(main CPU), 제어용 프로그램이 적재될 external memory, 마이크로프로세서와 개발자와의 인터페이스를 위한 직렬 또는 병렬 통신을 제공하는 serial/parallel interface 및 프로그램 개발용 JTAG, 마이크로프로세서 내에 갖추지 못한 서보모터제어를 위한 각종 주변 device들을 집적화시킨 FPGA 및 configuration 프로그램 저장을 위한 인터페이스(byteblaster), 전력 변환부와 각종 제어신호 및 응답신호를 주고받기 위한 디바이스들로 구성되어 있다.

표 2-1 주제어기 설계 조건
Table 2-1 Design requirements of main controller

항 목	조건
Program Memory	• 16bit 1Mbyte 이상일 것
DSP Programming	• JTAG Mode와 Serial Boot Loader Mode를 지원할 것
FPGA Programming	• PS Mode와 JTAG Mode를 지원할 것
A/D Converter	• Sampling 주파수 : 100KHz 이상 • Resolution : 12bit (parallel 출력) • Differential 입력 가능하고, Amp. Gain Control가능할 것 • 6 Ch.(U/V/W상전류,PFC용,aux1,aux2) • Analog신호 전송 : 동축케이블 적용
D/A Converter	• Settling time : 10usec이하일 것 • Resolution : 12bit (parallel, Buffered Input) • Bipolar 출력일 것 • 4 Ch.(U,V,W,aux1)
Motor속도입력 기능	• Encoder : 인크리멘탈형 Encoder를 적용하고, Z신호 유/무에 무관할 것 • 전압형,전류형,Open Collector형 Encoder의 접속을 Option으로 선택 가능할 것 • 신호처리방법 : M/T Method 처리기를 가질 것
PWM신호발생기능	• SVPWM을 적용하고, Deadtime 발생기를 갖춘 대칭형 삼각파발생일 것 • PFC용 PWM 발생기 1채널을 갖출것 • PWM발생기는 Underflow, Overflow, Compare 시 인터럽트 발생이 가능한 구조일 것 • PWM 신호전송은 인버터부와 절연시킬 것
외부인터럽터 확장	• PWM발생기로부터의 인터럽터 - Period Counter Overflow,Underflow시 각 1event(2event) - Compare Counter U,V,W 각 2event (6event) - PFC용 PWM Compare Counter 완료시 1event • ADC의 완료시, Encoder의 M/T Method Count완료시, Serial 통신용, Parallel통신용 각 1event (4event) • 보조 외부인터럽터 2event • 총 15event를 우선순위 제어 가능하고, CPU의 인터럽터와 연결할 것
이상감지기능	• 전원전압강하, 프로세서 폭주 감지시 CPU는 자동으로 리셋되며, 외부 리셋키에 의해서 강제 리셋 가능할 것
외부모니터 기능	• PC-컨트롤러 직/병렬 통신 각1, LED (7 Segment 2Digit)

※ 전원전압 및 주파수범위는 170~265V 60Hz±2Hz로 한다

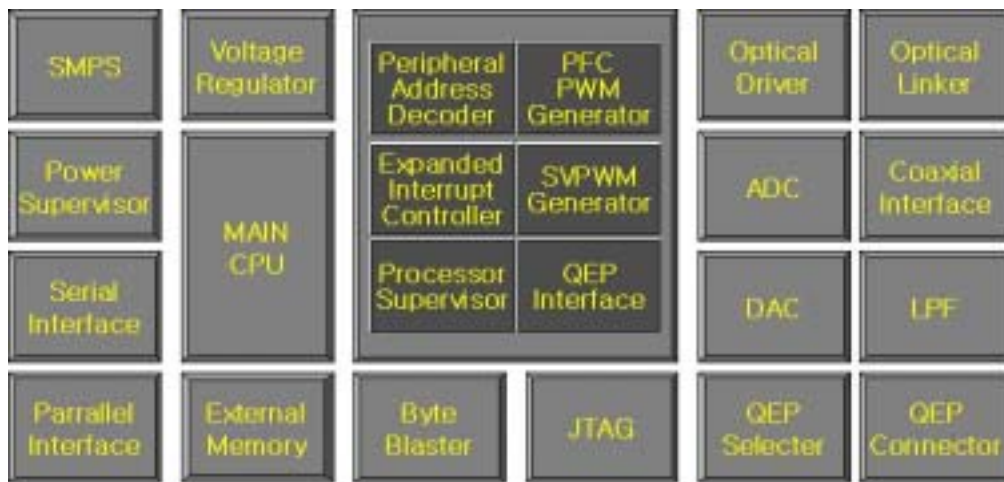


그림 2-2 주제어기의 블록도

Fig. 2-2 Block diagram of main controller

2.2.1 마이크로프로세서(main CPU)

전체 시스템의 두뇌에 해당하는 부분으로서 속도제어, 전류제어, 입출력제어 등을 담당하며, 이를 위해 각종 입력 데이터 및 측정 데이터를 바탕으로 많은 연산을 수행하고 내부적인 시간을 계산하며 제어신호나 시스템의 상태를 외부로 출력한다. 이와 같은 내부적인 연산처리는 대부분 부동소수점 연산이며 또한 매우 고속의 처리를 요구한다. 더구나 벡터제어를 위해서는 속도 및 전류제어기에서 제어 알고리즘 연산을 수행하고 전류와 전압의 좌표변환 등을 실시간으로 처리해야 하며 여기에는 삼각함수 등 많은 부동소수점 연산을 필요로 한다. 종래에는 범용 마이크로프로세서를 사용하여 부동소수점 연산 프로그램을 작성하였으므로 처리시간이 길어졌고, 따라서 전류나 속도측정의 샘플링 주기를 짧게하기 어려웠으며, 삼각함수 연산은 주로 데이터 테이블을 사용하였으므로 정밀도가 떨어지거나 메모리 용량이 커지는 문제를 감수 할 수밖에 없었다. 그러나 최근에는 매우 성능이 우수한 DSP소자가 개발되고 그 응용 분야가 급격히 넓어짐에 따라 가격도 저렴해지고 있어서 이러한 문제들을 용이하게 해결할 수 있다.[2]

2.2.2 FPGA

FPGA는 프로그램을 통해 회로를 구성할 수 있는 장점 때문에 복잡하고 넓은 공간을 사용하는 디지털회로를 설계할 수 있다. 반도체 집적화와 생산 기술이 발전하면서 프로그램을 통해 사용할 수 있는 공간이 넓어지고 여러 디지털 회로간의 지연들이 없이 내부지연 뿐이기 때문에 빠른 처리를 할 수 있는 능력을 가지게 되었다. 모터에 인가되는 전압을 가변제어하기 위해서는 SVPWM 신호 발생기능이 필요하며 모터제어 전용 마이크로프로세서가 아닌 범용 마이크로프로세서는 이 기능을 하드웨어적으로 구현을 해야하는데 이러한 경우에 논리회로 설계 툴인 VHDL를 이용하면 이러한 하드웨어를 쉽게 설계하여 FPGA로 구현하기가 용이하다. 그리고 어드레스 디코더, 인터럽트 컨트롤러등의 추가적인 디바이스와 주변장치들과의 인터페이스를 하기위한

회로를 손쉽게 구현할 수 있고 다시 수정하거나 회로를 보완하기가 용이한 장점을 가지고 있다.

2.2.3 ADC(Analog to Digital Converter)

모터 구동을 위해 인가되는 전류의 크기와 위상을 알기 위해 전류센서로 센싱된 신호를 ADC가 디지털 데이터로 변환하여 마이크로프로세서에 보내어 준다. 이때 resolution을 몇 비트로 할 것인지, 디지털로 변환된 데이터를 serial로 처리 할 것인지 parallel로 처리 할 것인지를 결정해야한다.

2.2.4 Control signal transmitter/receiver

고성능 서보모터중 3상 서보모터를 구동하기 위해서는 3상인 모터에 3상의 전류를 서로 120도 위상차가 생기게 인가해 주어야 한다. 여기서, 3상의 전압을 만들기 위한 방법으로 일반적으로 PWM 기법을 사용하며 이것을 만들기 위해 스위칭 소자들(예를 들면 TR, IGBT, MOSFET등)을 사용하게된다. 마이크로프로세서에서 제어 프로그램을 통해 연산한 시간동안 각 스위칭 소자를 on/off를 할 수 있는 펄스신호를 스위칭 소자로 보내게 된다. 이 신호를 보낼 때 노이즈나 외란에 영향을 받지 않게 하고, 전력변환부의 대 전원의 변화에 전기적으로 절연 시키기 위하여 포토 커플러나 광 케이블등을 이용하여 신호를 보낸다.

2.2.5 엔코더 및 엔코더 수신부

전동기의 회전속도를 측정하는 방법에는 여러 가지가 있으나 크게 아날로그 방식과 디지털 방식으로 나눌 수 있다. 아날로그 방식의 대표적인 것으로는 타코 제너레이터(taco-generator)를 사용하는 방식인데, 이것은 회전속도와 측정신호의 직선성이 우수하나 전동기의 기계적 관성은 크게하고 측정된 신호를 디지털 신호로 변환하는 과정이 필요하다. 또한 리졸버(resolve)는 아날

로그 방식으로 회전자의 위치를 검출하고자 할 때 많이 사용되나, 속도를 측정할 경우에는 이를 디지털 값으로 변환하여 사용하므로 디지털 방식으로 간주할 수 있다. 디지털 방식의 대표적인 것으로는 로터리 엔코더(rotary encoder)가 있는데 이것은 센서의 종류에 따라 광학식과 기계식이 있고, 출력값의 형태에 따라 증분형(incremental encoder)과 절대치형(absolute encoder)이 있다. 일반적으로 디지털 서보제어에 널리 사용되는 것은 분해능, 안정성, 경제성 등에서 우수한 특성을 갖는 광학식 증분형 로터리 엔코더이다. 이것은 회전 속도 뿐만 아니라 회전 방향을 검출할 수 있도록 90°의 위상차를 갖는 A,B상 2개의 출력을 가지고 있으며, 대부분의 경우 위치검출에도 사용할 수 있도록 1회전당 1개의 펄스를 발생하는 Z상의 출력도 가지고 있다.

2.3 전력 변환부

전력 변환부는 3상 또는 단상 교류 입력전원을 직류로 변환하고 이를 다시 마이크로프로세서의 지시를 받아 전동기의 구동에 적합한 가변 주파수 및 가변 전압(또는 가변전류)의 교류전원으로 변환하는 부분이다. 전력변환부는 컨버터(converter) 및 인버터(inverter)를 중심으로 하여 직류평활용 필터회로, 돌입전류 제한회로, 스너버 회로(snubber circuit), 서지 흡수회로 등으로 구성되며, 여기에 발전제동(dynamic braking)을 위한 트랜지스터와 제동저항이 추가된다. 인버터는 빠른 응답특성과 출력전압이나 출력전류에 고조파가 적을 것 등이 요구되며, 특히 서보제어용으로 사용되는 것을 전제로 하면 전동기의 매우 빈번한 기동, 정지 및 속도변화가 이루어지므로 충분한 과부하 내량을 갖도록 설계되어야 한다. 또한 전체 구동장치 외형의 크기는 대부분 이 전력변환부에 의하여 결정되므로 가급적 소형 및 박형화를 지향하는 컴팩트한 설계가 필요하게 된다.

2.3.1 정류기(rectifier)

AC-DC 컨버터를 보통 정류(rectifier)라 하는데 출력단 DC전압의 ripple을 가능한 줄이기 위하여 큰 capacitor를 DC 출력단에 적용하게된다. 이 capacitor는 입력 전압의 peak크기에 가깝게 충전을 하게되고, 여기서 생성된 DC 전압을 이용하여 inverter의 입력 전원으로 사용되어진다. [3]

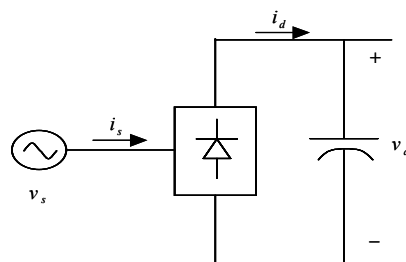


그림 2-3 정류회로 블록도

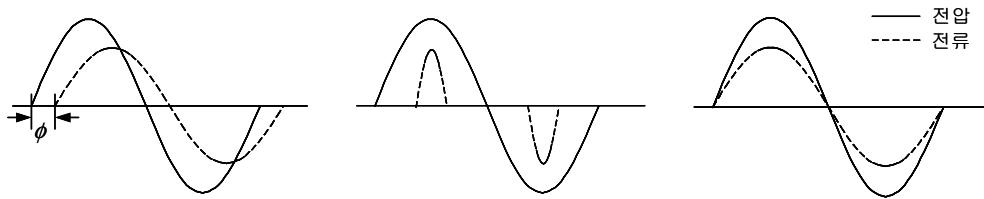
Fig. 2-3 Block diagram of rectifier

이때 capacitor의 용량이 커지면 부하전류의 변화에도 일정한 전압을 유지할 수 있다는 장점이 있으나, 입력 전원에서부터의 충전전류는 더욱 피크성이 되어 고조파 성분을 많이 포함하게 되므로 전력 효율이 나빠지게 된다. 이러한 문제를 해결하기 위하여 PFC가 적용이 되기도 한다.

2.3.2 PFC(power factor correction)

PFC를 구하는 식은 $Power\ Factor = \frac{P(\text{the real average power})}{S(\text{the apparent power})} = \cos \phi$ 로

써 전압과 전류의 위상이 같아서 $\cos \phi = 1$ 인 unity power factor 되게 하는 것이 좋다.[3]



a) 유도성 부하일 때 b) 인버터용 dc 전원단에서 c) $\cos \phi = 1$ 인 경우

그림 2-4 부하에 따른 전압과 전류의 위상차

Fig 2-4 Phase gap between voltage with current by inductive load

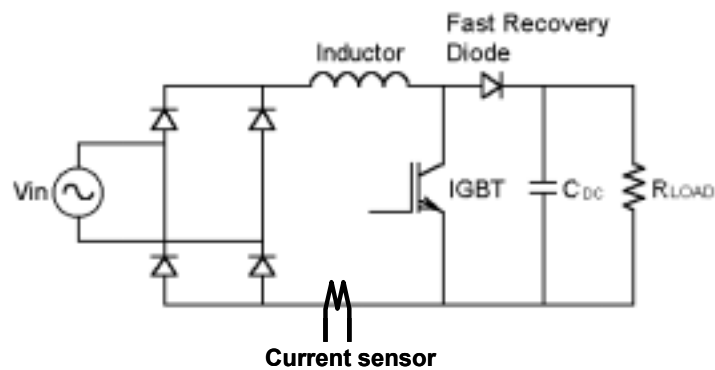


그림 2-5 Boost 형 PFC회로

Fig. 2-5 PFC circuit of boost type

그림 2-5는 본 논문에서 제어 대상으로 하는 boost형 PFC 회로이다. PFC를 사용하므로 입력전압과 입력전류의 위상을 동상으로 유지시켜 주어 입력전류의 최대치가 줄어 들게하며, 전원 입력단에서의 고조파를 감쇄 시키는 active-filter의 기능을 한다. 출력단 DC-link 전압을 항상 일정하게 유지시키므로서 inverter의 안정적인 운전이 가능하게 하고 전동기의 효율 향상 및 수명이 연장된다. 그리고 입력단의 V_{peak} 는 DC출력전압보다 작아야 한다.

2.3.3 Inverter

switch mode inverter중에 입력 소스에 따라 전압소스 인버터(voltage source inverter)와 전류소스 인버터(current source inverter)로 크게 나눌 수 있는데, 전류소스 인버터는 매우 큰 전력의 AC모터 드라이브에 주로 사용된다. 본 논문에서 적용 대상으로 하는 인버터는 전압소스 인버터로써 다음의 3가지로 분류할 수 있다.[3]

- 1) square wave inverter : 출력 AC전압의 주파수만 가변이 가능하다.
- 2) single phase inverter with voltage cancellation : 출력 AC 전압의 주파수와 크기를 가변 시킬 수 있다.
- 3) pulse width modulated inverter : 출력 AC전압의 주파수와 크기를 가변시킬 수 있다. 그림 2-6는 큰 DC입력인 V_{dc} 를 스위칭 gate인 $S_a, S_b, S_c, \underline{S}_a, \underline{S}_b, \underline{S}_c$ 로 on/off함으로써 주파수와 크기를 가변하는 3상의 V_a, V_b, V_c 출력전압을 만들어낸다.

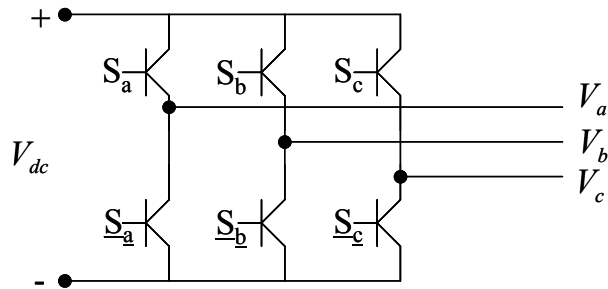


그림 2-6 Inverter 내부 구성도

Fig. 2-6 Internal diagram of inverter

2.3.4 전류 센서

계측 및 제어에 이용되는 전류센서는 크게 교류전류센서와 직류전류센서로 분류할 수 있다. 교류 전류센서는 일반적으로 전류트랜스방식에 의한 것이어서 CT(current transformer)라고 불리우는 것이고, 직류 전류센서는 로렌츠의 힘에 기인한 반도체의 홀효과(hall effect)를 이용한 것이다. 교류 전류센서는 순시값이 아닌 실효치(rms)의 측정에 주로 이용되며 직류전류센서는 직류전류에서부터 수kHz까지의 순시전류의 측정에 이용되고 있다. 교류전류센서는 측정하는 원래의 파형에 대하여 위상차를 가진 출력을 얻게되며 원칙적으로 상용주파수(50/60Hz)에서의 사용을 전제로 한 센서이다. 이에 비하여 직류전류센서는 직류에서 고주파 교류까지의 전류를 연속적으로 측정할 수 있는 특징이 있다. 그러므로, 서보 모터 제어분야에서는 직류전류센서를 일반적으로 적용하고 있으며, 본 논문에서도 직류전류센서의 적용을 기준으로 설계되었다.

2.4 서보 모터

1960년대 말부터 1970년대 초에 독일에서 개발되고, 1980년대 초기에 이르러 완벽하게 그 제어 알고리즘이 확립된 벡터제어 이론은 교류전동기의 가변속 운전의 성능을 직류전동기 수준 혹은 그 이상으로 가능하게 해 주었다. 이로 인하여 종래의 직류 전동기의 가변속 구동 응용분야를 점차 교류전동기가 대체해 나갔는데 그 결과, 소형 서보 시스템에서는 영구자석형 동기전동기가 많이 적용되고, 압연기, 전철용 차량등 대형 시스템에는 유도 전동기가 많이 적용되게 되었다. 서보모터의 종류를 크게 세 부분으로 나누면 다음과 같다.[6]

첫째, SPMSM(sinusoidal permanent magnet synchronous motor)이다. 회전자의 표면에 영구자석을 부착한 표면 부착형으로 영구자석의 저항이 크고 비투자율이 공기와 거의 같은 1정도이므로 회전자의 표면에 부착된 영구자석 부분이 공극과 같게 되어 돌극성이 적고, 따라서 사실상 공극의 길이가 매우 크게 되어 고정자의 슬롯등에 의한 돌극성도 거의 상쇄되므로 비돌극 권선형 동기 전동기와 유사한 특성을 갖게된다. 구조적 특성을 보면 해석이 간단하고 인덕턴스가 작아 서보제어용으로 사용하기가 쉬운 장점이 있다. 이는 회전자가 회전하면 고정자 코일에 유기되는 기전력의 파형이 사인파형으로 출력력이 되는 모터이다.

둘째, BLDCM(brushless DC motor)이다. 회전자의 철심 내부에 영구자석을 설치한 매입형으로 회전자 내부에서 영구자석이 있는 부분은 공극이 길어지고 영구자석 사이의 부분에서는 공극의 길이에 변화가 없으므로 돌극성이 뚜렷하여 돌극 권선형 동기전동기와 유사한 특성을 가지게 된다. 회전자의 구조가 기계적으로 강하여 고속용으로 적합한 장점을 갖는다. 이는 유기되는 기전력의 파형이 구형파(trapezoidal)로서 출력되는 모터이다.

셋째, ACIM(AC induction motor)이다. 유도 전동기의 고정자 권선과 회전자 권선의 전기적 결합은 변압기와 같이 생각할 수 있으며 등가회로는 기본적으로는 변압기와 동일하게 취급할 수 있다. 고정자권선에 의한 회전자속이 회

전자 권선과 쇠교하여 유기전압을 발생시킨다. 회전자 권선은 폐루프를 이루고 있기 때문에 회전자 유기전압에 의하여 2차 전류가 흐르게 된다. 회전자계의 회전속도와 회전자회전속도가 같으면 유도 기전력이 생기지 않아 모터가 회전을 하지 않지만 서로가 다른 속도로 회전할 때 슬립(slip)이 생겨나 회전을 하게 된다.[5]

Ⅲ. 주요 device 소개

3.1 TMS320VC33

마이크로프로세서의 출현은 1969년 일본의 sharp사가 전자식 탁상계산기의 CPU를 하나의 칩으로 만들고자 intel사와 계약을 맺음으로서 시작되었다. 그로부터 2년 뒤인 1971년에 intel사에서는 MOS기술을 이용한 고집적 회로로 요구된 제품을 만들어 내었으나, 요구된 처리속도에 미치지 못하였다. 결국 계약은 취소되었지만 intel사는 이것을 이용하여 intel 4004라는 이름으로 시판에 들어 가면서 프로세서의 개발은 시작되었다. 다음해에 8bit 마이크로 프로세서 8008를 개발해 내므로써 프로세서의 시대가 시작 되었다. 이것을 계기로 미국의 motorola사에서는 1974년에 8bit 마이크로프로세서인 MC6800을 개발하여 내놓았는데, 이 무렵부터 세상은 온통 마이크로프로세서 시대라고 말할수 있을 정도로 모든 분야에서 마이크로프로세서에 대한 관심이 고조 되었다. 1970년대 중반으로부터 후반까지는 8bit 마이크로프로세서의 전성 시대라고 말할수 있다. 1978년도에 이르러 16비트 마이크로프로세서로서 intel사의 intel 8086, MOTOROLA사의 MC 68000등이 발표 되었고 그 이듬해에는 zilog사의 Z8000이 발표 되었다. 32bit 마이크로 프로세로서는 1981년에 intel의 80386, MOTOROLA의 68020 및 68030이 발표되었다.[4]

본 논문에서 사용된 DSP(digital signal process)는 실시간 디지털 신호처리 분야에서 종래의 범용 마이크로프로세서나 마이크로 콘트롤러로는 부족한 기능이나 성능을 보완하기 위하여 만들어졌으므로 이들과 근본적으로는 다를 것이 없다. DSP는 RISC 마이크로 프로세서와 같이 1970년대 말부터 개발되기 시작하여 1980년대 초반에 본격적으로 상용화 되었다. 최초의 DSP는 1979년 AT&T의 벨연구소에서 자체적으로 사용하기 위하여 만든 DSP1 이었으며, 이후로 1980년에 NEC 에서는 μ PD7720을 개발하였고 1982년에 TI사에서 TMS32010을 발표 하였다. TI 사의 제품이 상업적으로 성공하므로써

DSP의 발전은 급속히 성장하여 현재는 TMS320C64X 같은 경우 16bit fixed point로서 3200~4800MIPS(million instruction per second) 정도이고 본 논문
에 사용한 TMS320VC33인 경우는 32bit floating point로써 75MIPS/150MFLOPS이다.[11]

3.1.1 TMS320VC33의 내부

32bit로써 높은 성능을 가진 프로세서로 주 전원은 3.3V이고 코어 전원은 1.8V 이다. 13ns/17ns 명령 사이클 타임을 가지고 16/32bit integer와 32/40bit floating으로 동작을 한다. 그리고 곱셈기를 가지고 있어 한 사이클 안에 곱셈연산을 실현할 수 있으며 곱셈 연산을 할 때 정밀도를 높이기 위해 40bit로 연산되어진다. 32bit 명령어로 동작을 하고 24bit 어드레스의 메모리 공간을 가진다. 내부 메모리는 Dual Access SRAM으로 1.1Mbit를 가지고 있으며 두 개의 1K x 32bit 메모리 블록과 두 개의 16K x 32bit 메모리 블록 구조로 되어 있다. 이러한 구조를 수정된 하바드 구조라 하고 내부에 프로그램 메모리와 데이터 메모리에 접근하기 위한 데이터 버스와 어드레스 버스를 별도로 가지고 있어 내부 동작을 빠르게 한 것이다. 그러나, 외부로는 하나의 데이터 버스와 어드레스 버스를 만들어 핀의 수를 줄였다. 내부 주변 장치로는 하나의 시리얼 포트와 두 개의 32bit 타이머, 하나의 DMA(direct memory access)를 가지고 있다. 그림 3-1은 내부 기능 블록도를 나타내고 있다.

3.1.2 Boot Loader

3.1.2.1 마이크로프로세서 모드

리셋, 인터럽트, 트랩 벡터가 0000H~003fH번지에 위치 하여 DSP 가 리셋 되고 나면 0000H번지에서 리셋 벡터가 읽혀져서 프로그램 카운터가 로드됨으로 프로그램의 실행이 시작되는 번지가 결정된다.

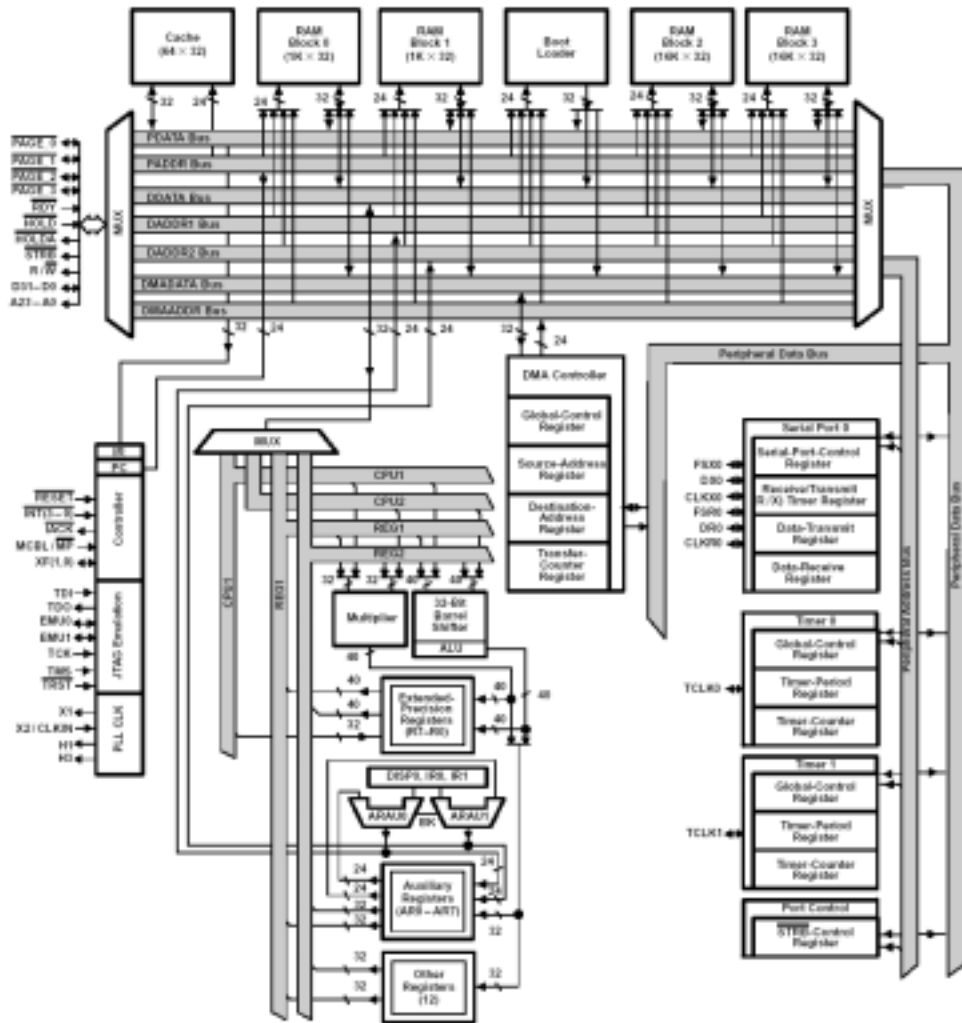


그림 3-1 내부 기능 블록도

Fig.3-1 Internal functional block diagram

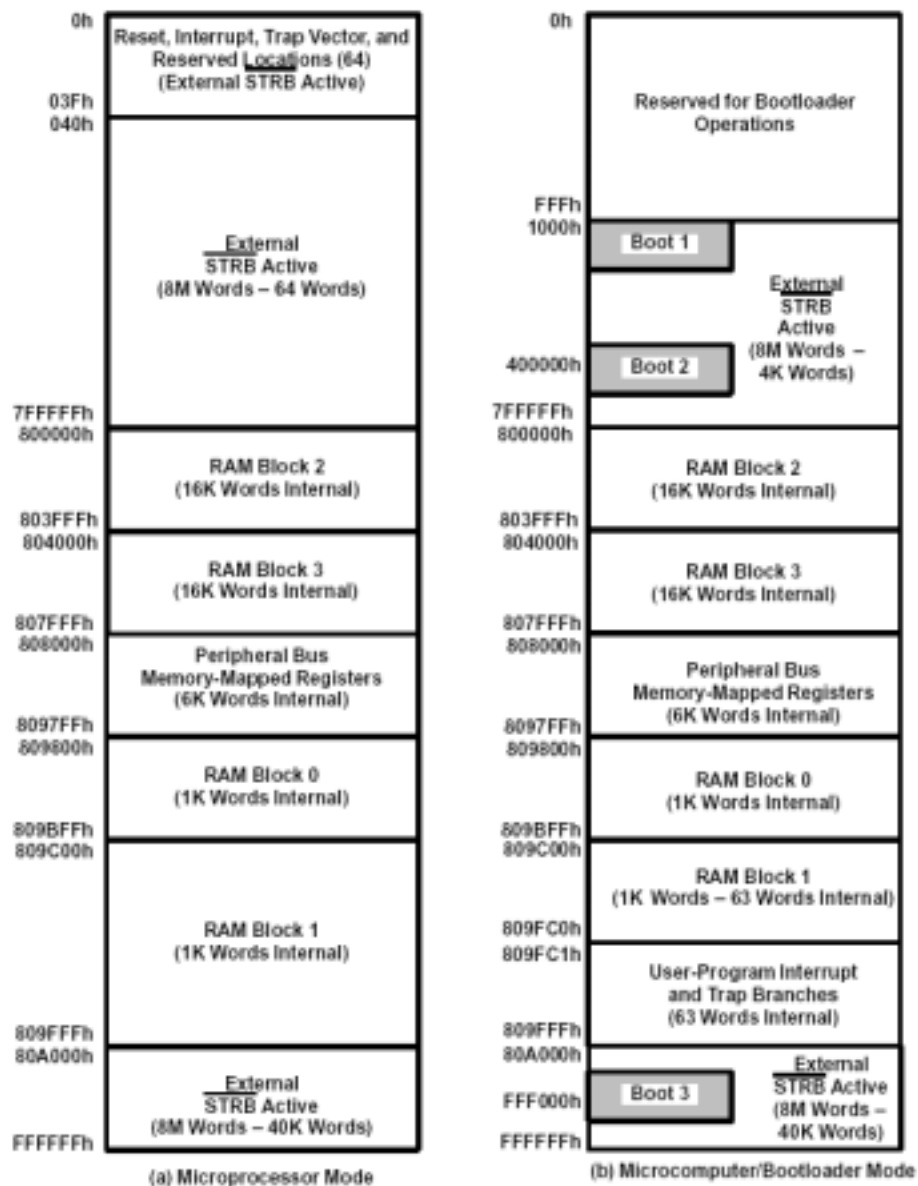


그림 3-2 메모리 맵
Fig. 3-2 Memory map

3.1.2.2 마이크로 컴퓨터 모드

마이크로 컴퓨터 모드로 사용할 때 boot-program을 load할 수 있도록 제조사에서 부트로드 프로그램을 0H - fffH 영역에 넣어두어 전원이 인가되면 이 프로그램이 동작하여 외부 인터럽터 INT0~3중 하나의 입력을 받게 된다. 이때 외부 인터럽트에서 어떤 입력이 들어오면 그 입력에 맞게 부트로드 프로그램에 의해 동작하게 되는데 4가지 모드로 부트로드 할 수 있게 되어 있다. 입력 신호가 INT3을 통해 입력이 들어오면 시리얼 통신으로 부트로드를 하고 INT0~2중 하나의 입력이 들어오면 외부의 메모리(예를 들어 EPROM)에 적재되어 있는 프로그램을 메모리 영역 BOOT1~3으로 부트용 프로그램이 로드된다. 이때 부트로드 되는 프로그램은 프로그램의 데이터의 크기, 전체 용량, 적재되는 시작번지등 부트로드 프로그램에 대한 정보를 알 수 있게 되어 있다. BOOT1~3을 이용한 부트로드 방법은[10]

- ① 부팅 메모리워드의 길이(8/16/32 bit)
- ② 속도가 느린 외부메모리로부터 로딩을 위한 wait 수를 결정
- ③ 부트로드 프로그램의 사이즈
- ④ 프로그램 로드 시작 번지

⑤ 유효한 프로그램 순서로 작성을 해야한다. 예를 들면, 부팅 메모리 워드의 길이가 8bit 이고 primary bus control register의 WCNT = 2, SWW = 11를 하고 부팅용 프로그램의 사이즈가 0x1FF이고 프로그램 로딩할 시작 번지가 0x809C00인 경우 표 3-1과 같이 작성을 하게 된다. 시리얼 통신을 이용한 부트 로드 방법은 INT3를 입력을 받아 부트하는 방법으로 주로 고정 버스트 모드를 사용한다. 송수신하려는 데이터의 길이가 고정되어 있고 FSX/FSR(frame synchronous transmitter/ receive) 펄스가 발생하게 되면 송수신 클럭의 다음 클럭에서 일정한 크기의 송수신 데이터워드를 전송하게 된다. 예를 들어, 데이터를 송신할 경우 송신할 데이터 워드가 DXR에 로드를 시키고 내부적으로 프레임 동기 신호(FSX)가 발생하면 다음 전송 클럭에서 전송할 데이터 첫 비트를 전송하면서 동시에 전송 인터럽트(XINT)을 받

생시키게 된다. 전송할 워드의 마지막 비트를 보내고 나면 수신측에서 받았다는 수신 인터럽트(RINT)가 발생한다. 이때 FSX신호는 내부적으로 전송 클럭의 한 주기 동안 펄스를 발생한 후 다음 클럭에서 전송하려는 워드의 데이터가 전송하게 되는데 마지막 비트가 전송이 완료되기까지 계속 low 상태를 유지해야한다. 또다시 다른 데이터를 전송 할 때는 먼저 프레임 동기펄스가 발생한 후 데이터를 전송하게 된다. 이렇게 반복적으로 전송할 데이터 길이가 고정된 크기의 데이터 전송을 하는 것이 고정 버스터 모드(fixed burst mode)이다.

표 3-1 바이트 폭으로 구성된 메모리

Table 3-1 Byte-wide configured memory

Address	Value	Comments
0x1000	0x08	Memory width = 8 bits
0x1001	0x00	
0x1002	0x00	
0x1003	0x00	
0x1004	0x58	Memory type = BWW = 11, WCNT = 2
0x1005	0x10	
0x1006	0x00	
0x1007	0x00	
0x1008	0xFF	Program code size = 0x1FF
0x1009	0x01	
0x100A	0x00	
0x100B	0x00	
0x100C	0x00	Program load starting address = 0x809C00
0x100D	0x9C	
0x100E	0x80	
0x100F	0x00	

3.1.3 내부 주변 장치

3.1.3.1 타이머

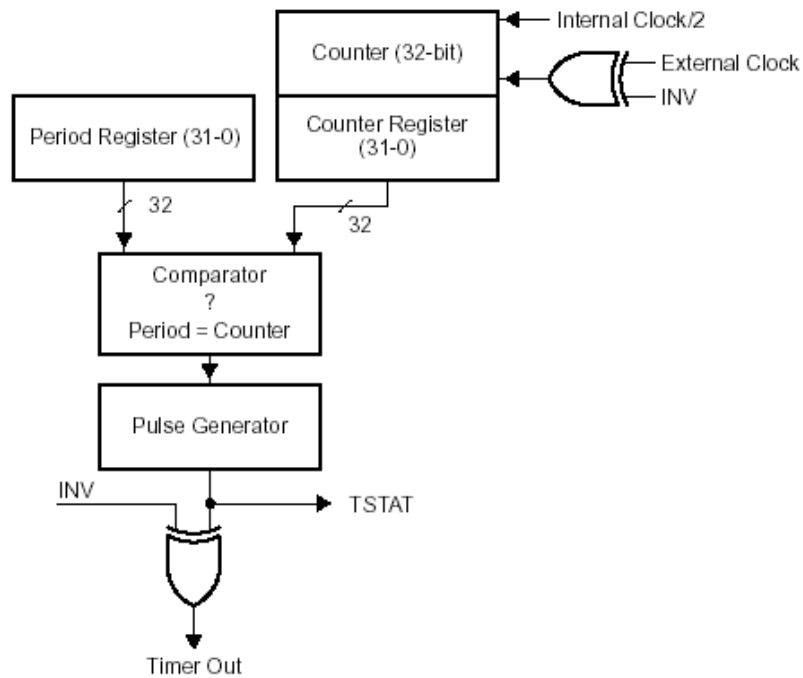


그림 3-3 타이머 내부 블록도

Fig. 3-3 Internal block diagram of timer

그림 3-3을 보면 내부에 두 개의 타이머가 있는데 내부 및 외부 클럭 중 하나를 타이머 클럭으로 사용할 수 있다. 타이머 클럭은 DSP의 H1/H3의 2분주된 것 보다 더 빠른 클럭을 사용할 수 없다. 타이머 사용을 위한 레지스터 중 global control register에서 타이머 동작을 시키면 카운터가 영으로 reset이 된 후에 하나씩 증가하는 카운터를 시작하게 되고 카운터의 값은 counter register에 계속 기록하게 된다. period register에 들어있는 값과 카운터의 값이 같은지를 계속 비교하다가 값이 같을 때 pulse generator를 통

해 펄스가 low이면 high로 high 이면 low로 바뀌게 된다. 이때 모드를 클럭 모드와 펄스 모드중에 선택하여 사용할 수 있다. 클럭 모드의 클럭 주파수는 $f = \text{타이머 클럭} / (2 \times \text{period register})$ 로 만들어지고, 펄스 모드의 클럭 주파수는 $f = \text{타이머 클럭} / \text{period register}$ 로 만들어진다

3.1.3.2 시리얼 통신

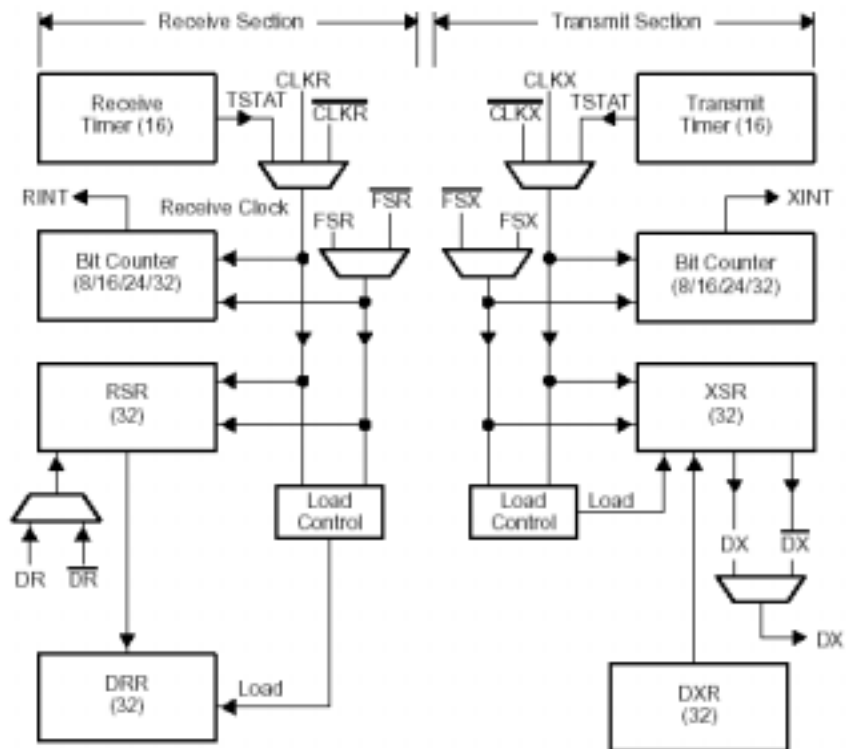


그림 3-4 시리얼 포트 블록도

Fig. 3-4 Block diagram of serial port

그림 3-4에서 시리얼 포트는 하나이며 8/16/24/32 bit로 양방향 통신을 할 수 있다. 시리얼 통신 방법으로는 크게 3가지로 나눌 수 있는데 fixed data rate mode 와 variable data rate mode, handshake mode로 크게 나눌 수 있다.

1) Handshake Mode

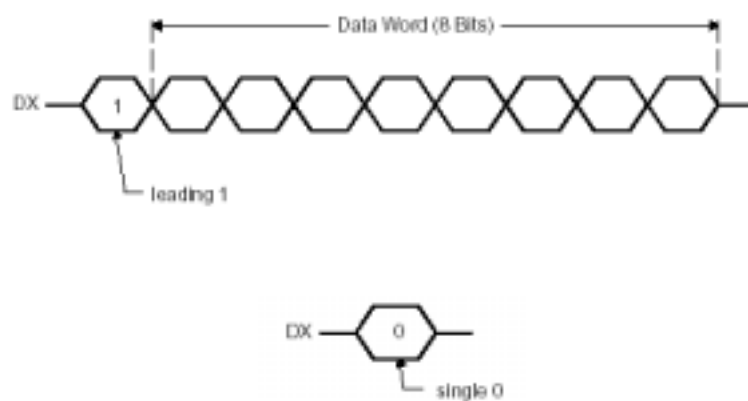


그림 3-5 Handshake 모드

Fig. 3-5 Handshake mode

그림 3-5에서 보면 8bit 데이터 워드를 송신할 경우 먼저 “1”이라는 값을 보낸 후, 데이터 8bit를 보내게 된다. 수신 시리얼 포트가 하나의 워드를 받고 그 워드를 DRR(data receiver register)로부터 읽은 후에 수신 시리얼 포트는 “0”이라는 데이터를 송신 시리얼 포트에 보낸다. 송신 시리얼 포트는 “0”을 받고 난 후에 새로운 데이터를 위와 동일하게 송신을 하게 된다.

2) Fixed data rate Mode

데이터의 길이가 8/16/24/32 bit중 하나로 고정하여 송수신을 하게 된다.

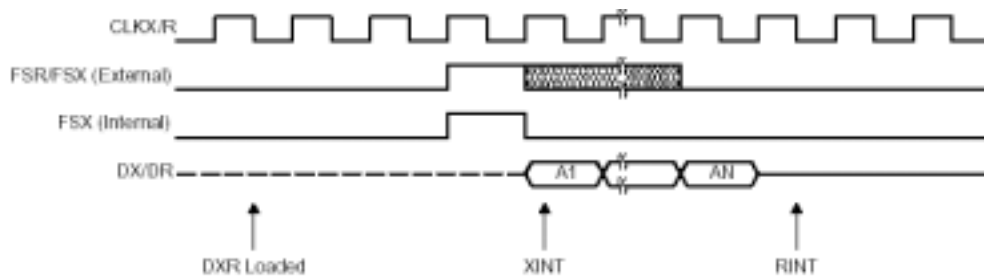


그림 3-6 고정 버스터 모드

Fig. 3-6 Fixed burst mode

① Burst mode

그림 3-6에서 송신을 하는 경우, 먼저 N bit로 정해진 데이터를 DXR(data transmit register)에 데이터를 로드한다, 그후 약 2.5 송신 사이클이 지나면 내부적으로 프레임 송신 동기신호(FSX) 펄스가 송신 클럭의 한 사이클동안 발생되어 송신 데이터는 송신 클럭에 맞추어 전송되게 된다. 송신 데이터 워드의 첫 비트에 송신 인터럽트(XINT)가 발생하게되고 마지막 비트가 송신된 다음 송신 클럭에서 수신 인터럽트(RINT)가 발생하게 된다.

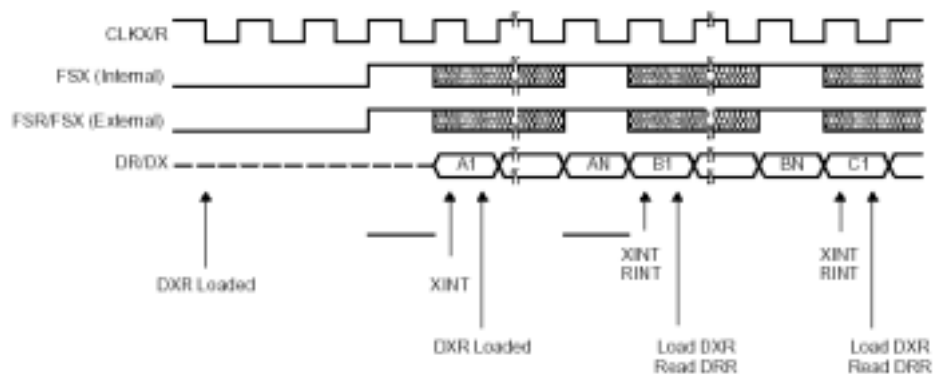


그림 3-7 고정 continuous 모드

Fig. 3-7 Fixed continuous mode

② Continuous mode

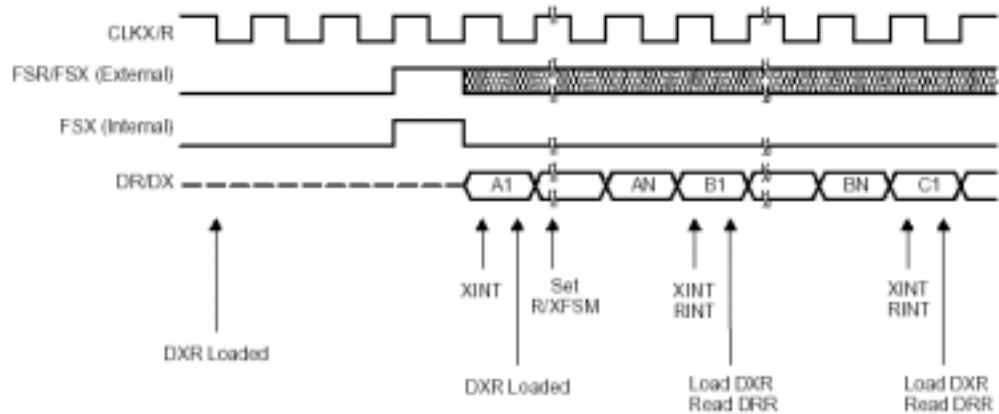


그림 3-8 가변 continuous 모드

Fig. 3-8 Variable continuous mode

데이터 송신시 프레임 송신 동기 신호가 첫 송신 데이터워드 일 때만 동기를 맞추고 그 이후에 전송되는 데이터 워드들은 연속해서 송신 클럭에 맞추어 전송하게 된다. 그림 3-7과 그림 3-8에서보면 연속적으로 데이터 전송이 되어지지만 내부적으로 프레임 송신 동기 신호가 각 데이터 워드가 보내질 때마다 동기 신호를 발생시키게 할 수도 있고 발생 시키지 않게 할 수도 있다.

3) Variable data rate Mode

fixed data rate mode와 비슷하게 동작 하지만 아래와 같은 차이점이 있다.

① fixed는 송신시 FSX 펄스의 크기가 송신 클럭의 한 주기 동안 발생하지만 variable 은 그 이상의 펄스 크기를 가진다. ② fixed는 FSX 펄스가 발생한 후 송신 데이터가 전송되지만 variable은 FSX 발생하자마자 송신 클럭 한 주기가 지나지 않은 상태에서 데이터 전송을 할 수 있다. ③ variable은 마지막 데이터 비트가 전송되기 전에 생겨난 다른 프레임 동기 신호(FSX)는 무시되지만 fixed는 전송워드 마지막 비트에 FSX신호가 발생하면 그 다음에 새로운 데이터 워드를 전송할 것을 인식하게 된다. 그림 3-9에서는 가변 버

스트 모드와 그림 3-10에서는 가변 continuous 모드를 나타내고 있다.

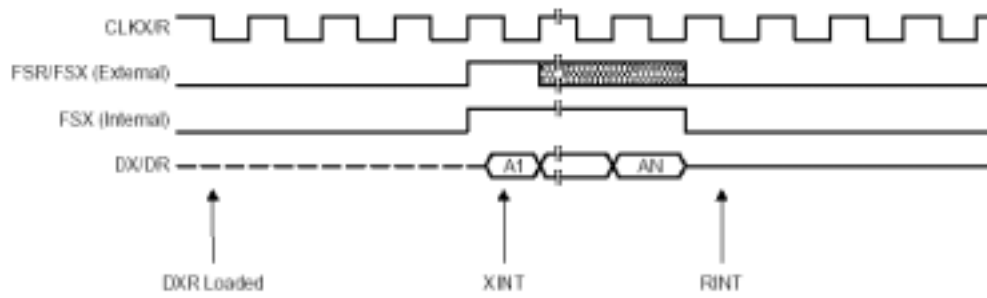


그림 3-9 가변 버스트 모드

Fig. 3-9 Variable burst mode

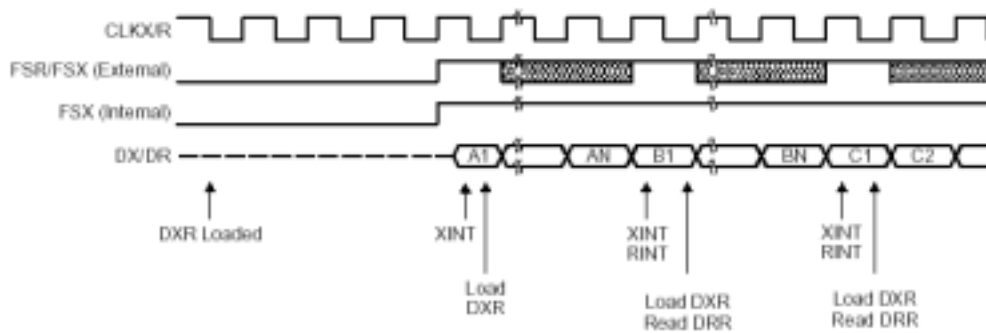


그림 3-10 가변 continuous 모드

Fig. 3-10 Variable continuous mode

3.2 FPGA(field-programmable gate arrays)

VHDL은 VHSIC HDL(very high speed integrated circuit hardware description language)이라는 말로써 80년대 초 미국 국방성의 VHSIC 프로그램의 산물이다. 또한 디지털 집적회로의 설계 규모가 방대해 지고 각 회사마다 서로 다른 설계툴과 설계방법을 가지고 있는 상황에서 집적회로의 표현에 표준화된 기술언어가 필요해졌기 때문에 이것이 탄생하게 된 배경이다. 현재 미 국방성 뿐만아니라 87년에 IEEE가 공식적으로 표준화시킨 것이 IEEE1076-87로 규정되고 5년후 92년에 수정되어 93년에 표준화된 것이 IEEE1076-93이다. VHDL의 장점은 매우 넓은 범위의 디자인을 가능하게 해 주고 미국 정부나 IEEE, 대부분의 미국 EDA(electric design automation)업체의 지지를 받고 있으므로 VHDL을 사용한 디자인은 많은 다른 회사들간의 호환성이 보장된다. 설계시 top-down방식의 디자인이 가능하여 설계기간이 훨씬 단축되고 설계자가 놓치기 쉬운 많은 단계에서의 error 요인들을 철저히 시뮬레이션을 통해 검증할 수 있다. 이는 제품 설계비용이 줄어들 수 있다는 것을 뜻한다. 그러나 VHDL은 언어 자체가 복잡하고 VHDL code를 회로 합성(logic synthesis)해보면 생성된 회로는 상당히 난해하고 gate 들간의 연결이 상당히 복잡하다. 이러한 장단점이 있지만 하드웨어 설계자들에게는 아주 유용한 언어임에 틀림없다.[8]

3.2.1 종류

PLD는 programmable logic device의 약자로서 IC내부의 배선을 사용자가 프로그램 해서 사용할 수 있는 IC이다.

이런 기능을 하는 PLD의 종류를 크게 6가지로 나눌 수 있다.[8]

첫째, PROM(programmable read only memory)

한 개의 디코더(AND 배열 부분)와 메모리셀(memory array)의 이차원 배열(OR 배열)로 이루어져 있다.

둘째, PLA(programmable logic array)

AND 배열(array)과 OR 배열(array)의 두 구조로 나누어지는데 PROM과 달리 AND 배열은 입력에 관한 어떤 논리곱을 표현 할 수 있고, OR배열은 앞에서 구현된 가장 맞는 이상적인 구조이다. 그러나, 이 구조는 두 개의 배열을 통해서 신호가 전달되므로 시간 지연을 가지는 단점을 가지고 있다.

세째, PAL(programmable array logic)

PLA의 구조와 비슷하게 이루어져 있다. 그러나 PAL의 구조는 AND 배열 부분을 사용자가 프로그램을 할 수가 있고 OR배열은 고정되어 있다는 차이점을 가지고 있는데 이 구조는 생산의 비용이 비교적 적게 들고 속도의 성능이 빠른 편이어서 많은 반도체 회사들이 이와 같은 기본 구조와 유사한 프로그램형 소자들을 만들어 내고 있다.

넷째, SPLD(simple programmable logic device)

이것은 PAL과 거의 유사한 구조를 가지고 있는데 사용자가 사용할 수 있는 게이트의 용량은 약 200개 정도가 된다. 이것은 CPLD의 기본 구조를 이룬다.

다섯째, CPLD(complex programmable logic device)

CPLD는 altera사에서 상업용으로 제품화되어 쓰이고 있으며 이러한 구조를 보면 메모리와 특별한 로직 함수를 구현하기 위해 embedded array가 있고 이 내부에는 여러개의 EAB(embedded array block)이 있다. 각각의 embedded array block은 2,048 bit 있으며 이것은 RAM, ROM, FIFO,

dual-port RAM등을 구현할수 있다. 또한 하나의 embedded array block 내부에는 8개의 logic element(LE)와 local interconnection(LI)으로 구성된다. 이러한 logic element에는 8bit counter, address decoder, state machine과 같은 중규모(medium size)의 logic block을 만들 수 있다. 외부와 인터페이스 되는 pin으로는 I/O element가 있고 이는 양방향의 I/O buffer와 전용(dedicated)의 핀들로 구성된다. 아울러 CPLD의 구조는 빠른 성능이나 정확한 타이밍의 예측이 필요로 하는 곳에 적합한 구조이며 FPGA(field-programmable gate array)의 구조는 많은 플립플롭(flip-flop)을 사용하는 순차 회로나 대용량이 필요한 곳에 적합한 것이기 때문에 서로 응용 분야에 따라 선택하여 적용하는 것이 좋다.

여섯째, FPGA(field programmable gate array)

CPLD와 달리 PLD의 블록간의 연결을 array 구조와 행(row)구조의 형태를 이용하여 연결을 하는 2가지의 형태로 나누어진다. 각각의 SPLD 블록간에는 나누어진 segmented interconnection를 쓰기 때문에 연결 상태에 따라 내부의 연결 길이가 변경 될 수가 있다. 그러므로 연결 상태에 따라 라우팅 딜레이(routing delay)가 다르게 되어 타이밍의 예측이 어렵고 논리 소자의 성능 또한 CPLD의 구조에 비해 떨어진다. 그러나 사용자가 사용할 수 있는 게이트의 용량은 CPLD보다는 많다는 장점을 가지고 있는데 이것은 각각의 구조가 나쁘다는 것보다는 사용자의 응용 부분에 따라서 특성화 할 수 있다는 방향으로 보는 것이 정확할 것이다.

3.2.2 CPLD 와 FPGA의 비교

그림 3-11에서 보는 것과 같이 PLD를 분류하고 있다. HCPLD(high-capacity programmable logic device)은 최근에 새로운 디바이스와 구조를 소개함으로써 급속히 확장되어 가고 있다. 그 중의 대표적인 것이 FPGA(field-programmable gate array)와 CPLD(complex programmable logic device)인데 서로간의 비교를 표 3-2로 나타내었다.[9]

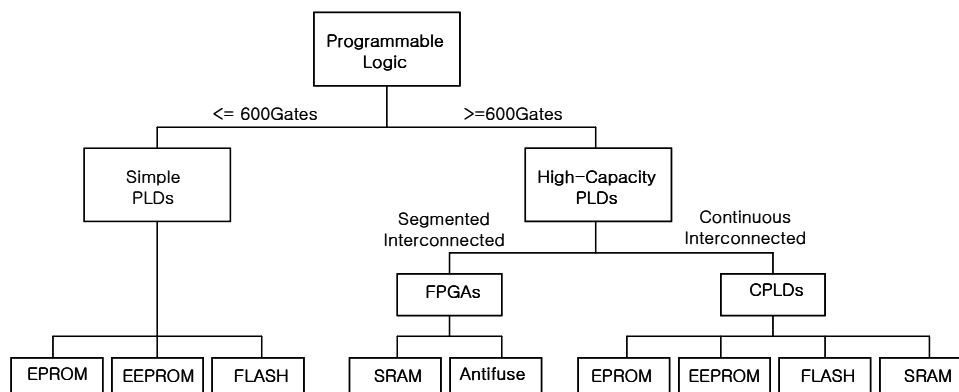


그림 3-11 PLD 분류

Fig. 3-11 Classifications of PLD

표 3-2 CPLD와 FPGA의 비교

Table 3-2 Compare of CPLD with FPGA

	FPGA	CPLD
Leading Vendor	Xilinx	Altera
Density	Medium to high	Low to high
Interconnect Structure	Segmented	Continuous
Timing	Variable/Unpredictable	Fixed/Predictable
CMOS option	SRAM, antifuse	EPROM, EEPROM, FLASH
Device performance	Moderate	High
Device utilization	Moderate	High
Hand-routing required	Yes	No
Reprogrammability	Yes(SRAM only)	Yes
In-circuit reconfigurability	Yes(SRAM only)	Yes(SRAM only)
In-system programmability	No	Yes(FLASH, EEPROM only)
Compilation times	Slow	Fast
Logic synthesis	Yes(third-party only)	Yes

3.3 ADC(analog to digital converter)

종류를 나누어 보면 크게 6가지로 나누는데 아래와 같이 하였다.

1) Digital ramp A/D converter : 변환 시간이 입력전압에 의존적이며 일정하지 않고 입력전압이 높을수록 변환 시간이 길어진다. 그림 3-12는 디지털 램프 A/D converter의 변환 원리를 나타내고 있다.

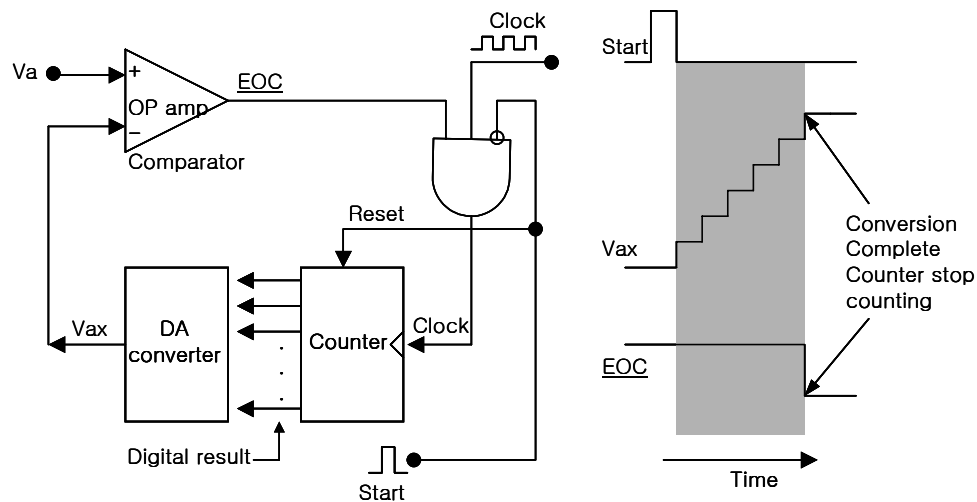


그림 3-12 디지털 램프 A/D converter

Fig. 3-12 Digital ramp A/D converter

2) SAC(sucessive approximation converter) : 입력전압에 관계없이 변환 시간이 일정하고 가장 일반적인 A/D converter이다. 그림 3-13은 연속 근사 컨버터의 블록도를 나타내고 있고 그림 3-13은 SAC 동작 순서를 나타내고 있다.

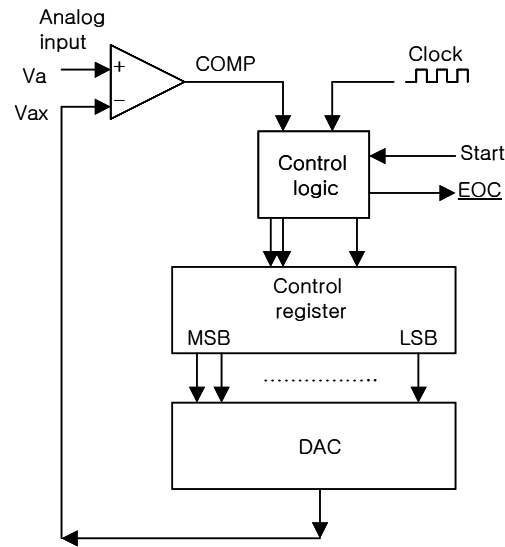


그림 3-13 연속 근사 컨버터

Fig. 3-13 SAC(successive approximation converter)

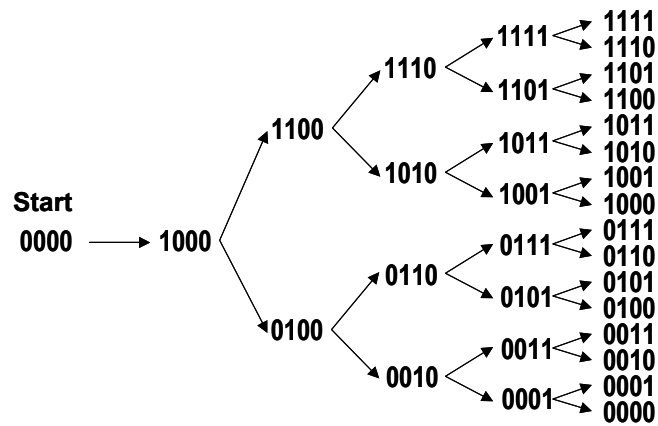


그림 3-14 SAC 동작 순서

Fig. 3-14 SAC operating process

3) Flash A/D converter : 가장 빠른 ADC이고 $2^N - 1$ 개(N : digital output bit 수)의 comparator가 필요하다. timing이나 process가 필요 없으므로 클럭 신호를 필요하지 않지만 A/D 변환은 연속적으로 이루어 진다. 그림 3-15에

서 블록도를 나타내고 있다.

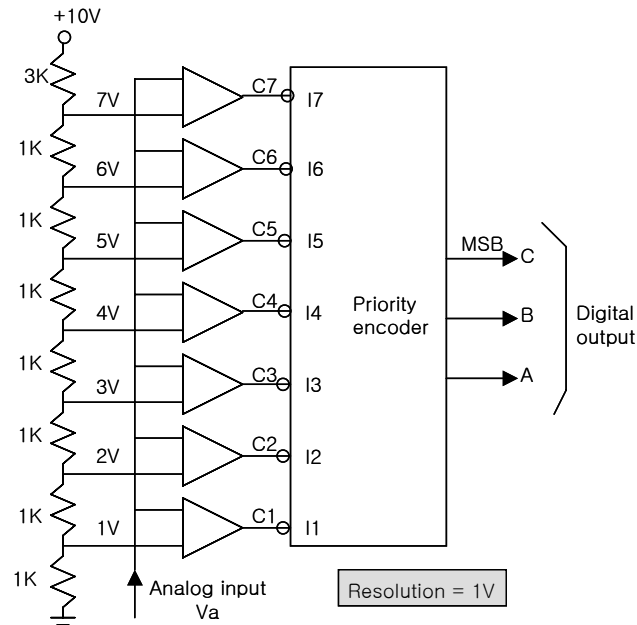


그림 3-15 플래쉬 A/D converter

Fig. 3-15 Flash A/D converter

4) Up/down A/D converter : 변환 시간이 아날로그 입력전압에 의존적이며 일정하지 않다. 그래서 이전의 입력 전압과 현재의 입력전압차가 클수록 변환 시간이 길다. 그림 3-16에서 변환 원리를 나타내고 있다.

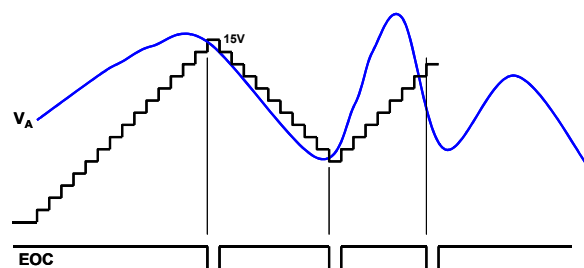


그림 3-16 업/다운 A/D converter

Fig. 3-16 Up/down A/D converter

5) Voltage to frequency A/D converter : DAC(digital to analog converter)를 대신하여 전압조정 발진기(VCO : voltage controlled oscillator)를 사용하며 아날로그 입력전압에 따라 선형적으로 변화하는 출력주파수를 카운터 함으로써 디지털로 변환한다. 0.1% 이상의 정밀도를 가지는 VCO를 설계하기가 어려우므로 높은 정밀도를 갖는 변환기를 만들기는 어렵지만 잡음이 심한 환경에서 작은 아날로그 신호를 변환기회로부터 제어 컴퓨터로 전송하고자 할 때 유용하게 사용된다.

6) Dual slop A/D converter : 변환시간이 매우 늦지만 DAC나 VCO를 필요로 하지 않으므로 경제적이고 잡음에 강하다. 주로 디지털 전압계 및 회로 분석기 등에 적용한다. 그림 3-17에서 기능 블록도를 나타내고 있다.

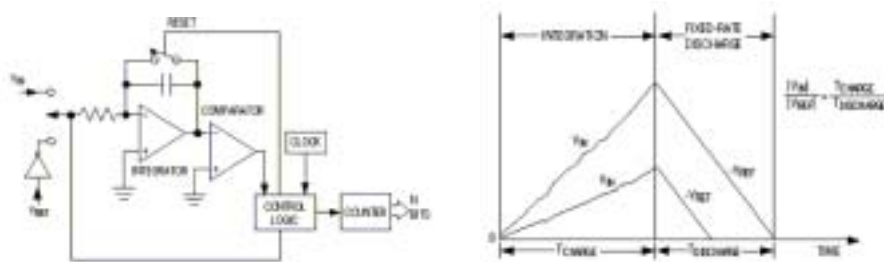


그림 3-17 Dual slop A/D converter
Fig. 3-17 Dual slop A/D converter

IV. 제어보드 설계

4.1 마이크로프로세서

4.1.1 프로그램 다운로드

본 논문에 사용된 TMS320VC33-150은 144핀 PQFP(plastic quad flatpack)타입을 사용하였고 Code Composer를 이용하여 프로그램을 다운로드 할 수 있도록 JTAG 포트를 이용하였다. 뿐만 아니라 마이크로 컴퓨터 모드에서 전원인가 시 외부 external 인터럽트인 $\overline{INT0}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{INT3}$ 에 의해 시리얼 부트로드, BOOT1, BOOT2, BOOT3 mode중 한 boot loader 방법이 결정되어 외부 메모리 영역에 저장되어 있는 부트로드 영역에서 프로그램을 로딩하여 프로그램을 수행하게 된다.[10] 하드웨어적으로 시리얼 부트로더를 할 수 있게 시리얼 포트를 부트로드 용으로 회로를 구성하였다. 그리고 외부 인터럽트 발생을 FPGA 내부에서 프로그램을 통하여 발생할 수 있게 설계하였다. 그림 4-1은 부트로드하기 위한 실제 구성 회로이다.

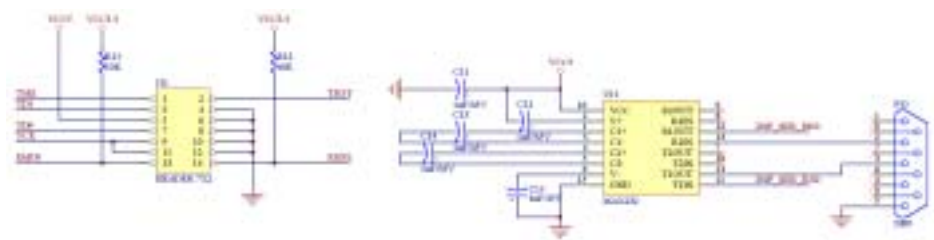


그림 4-1 JTAG 포트 & 시리얼 포트 회로

Fig. 4-1 JTAG port & serial port circuit

4.1.2 전기적 특성

먼저 메모리와 마이크로프로세서의 인터페이스에 있어서의 DC특성을 살펴 보면, 메모리의 전원이 5V이고 마이크로프로세서의 전원 중 입출력에 관련한 전압이 3.3V이므로 데이터버스의 경우는 마이크로프로세서의 전압입력요구량의 한계를 넘어서게 되지만, 마이크로프로세서로부터 메모리에의 제어신호들 (address, $\overline{CS}$, $R/\overline{W}$)은 $V_{OH} = 2.4 V_{MIN}$, $V_{OL} = 0.4 V_{MIN}$ 이어서 메모리의 입력전압요구량인 $V_{IH} = 2.0 V_{MIN}$, $V_{IL} = 0.8 V_{MIN}$ 을 만족시켜 $V_{NH} = V_{NL} = 0.4V$ 의 노이즈마진을 가지므로 데이터버스에만 단방향 버퍼를 설치하면 된다. 고전압다이오스와 저전압다이오스를 인터페이스 시키는데 사용되는 디바이스로는 저전압기술을 이용한 LVTH 시리즈가 있으므로 그 중 16bit용 단방향 버퍼인 29C1024를 선정하여 이들 디바이스간의 전기적 특성을 살펴보면,[11]

1) Memory와 LVTH16244 사이의 Noise Immunity

Data Sheet로부터

$$V_{NH} = V_{OH(memory)} - V_{IH(LVTH)} = 4.2 - 2 = 2.2 [V]$$

$$V_{NL} = V_{IL(LVTH)} - V_{OL(memory)} = 0.8 - 0.45 = 0.35 [V]$$

로 Noise Immunity는 충분하다.

그리고, Fan-Out을 고려함에 있어서 CMOS인 경우는 일반적으로 최대허용 전과지연시간에 의해 결정되므로 입출력전류량에 좌우되지는 않으나, Data Sheet를 비교해 보면,

$$I_{OH(memory)} = -400 \mu A @ V_{OH} = 2.4 V$$

$$I_{IH(LVTH)} = 1 \mu A (\max)$$

$$I_{OL(memory)} = 2.1 mA @ V_{OL} = 0.45 V$$

$$I_{IL(LVTH)} = -5 \mu A (\max)$$

이어서 Fan-Out은 400이 되므로 하나의 디바이스를 구동하는 상황에서는 고려할 정수가 아니며, 최대전과지연시간은 LVTH의 입력 Capacitance C_i 에 의하여 결정되는데,[12] Data Sheet를 참고하면 4.5ns(t_{PHZ} @Vcc=3.3V)인데 비하여, 메모리의 전과지연시간은 AT29C1024-12의 경우 120ns이므로 마이크로프로세서로의 프로그램 로드시 많은 wait를 사용하여야 함을 알 수 있다.

2) AT29C1024와 마이크로프로세서사이의 Noise Immunity

Data Sheet로부터

$$V_{NH} = V_{OH(LVTH)} - V_{IH(CPU)} = (V_{CC} - 0.2) - 0.7 * 3.3 = 0.79 [V]$$

$$V_{NL} = V_{IL(CPU)} - V_{OL(LVTH)} = 0.3 * 3.3 - 0.5 = 0.49 [V]$$

가 되므로 마이크로프로세서 - 메모리, 메모리 - 데이터버스 버퍼 - 마이크로프로세서의 인터페이스에 있어서의 DC 및 AC 특성이 만족함을 확인하였다.

4.1.3 PLL(phase-locked loop)

on-chip crystal oscillator 또는 EXTCLK(external clock)를 사용할 경우 PLL 특성을 표 4-1와 같이 나타낼 수 있다.

표 4-1 PLL의 전기적 특성

Table 4-1 PLL Characteristics

PARAMETER		MIN	MAX	UNIT
F_{PIN}	Frequency range, PLL input	5	15	MHz
F_{POUT}	Frequency range, PLL output	25	75	MHz
I_{PLL}	PLL current, C/VDD supply		2	mA
P_{PLL}	PLL power, C/VDD supply		5	mW
PLL_{dc}	PLL output duty cycle at H1	45	55	%
PLL_J	PLL output jitter, $F_{POUT} = 25$ MHz		400	ps
PLL_{LOCK}	PLL lock time in input cycles		1000	cycles

PLL의 입력의 주파수 범위는 5MHz ~ 15MHz안에서 사용을 해야하고 출력으로 사용할 경우에는 25MHz ~ 75MHz의 범위 내에서 출력으로 사용해야 한다. 그림 4-2는 클럭 발생부의 내부를 나타내고 있다.

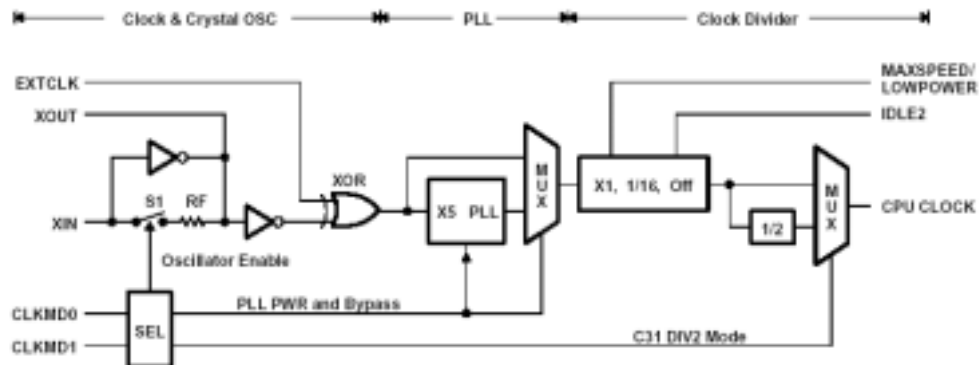


그림4-2 클럭 발생부

Fig. 4-2 Clock generation

표 4-2 클럭모드 선택 핀

Table 4-2 Clock mode selection pins

CLKMD0	CLKMD1	FEEDBACK	PLL PWR	RATIO	NOTES
0	0	Off	Off	1	Fully static, very low power
0	1	On	Off	1/2	Oscillator enabled
1	0	On	Off	1	Oscillator enabled
1	1	On	On	5	2 mA @ 60 MHz, 1.8 V PLL power, Oscillator enabled

본 논문에서는 표 4-2에서 보면 알수 있듯이 on-chip crystal oscillator 또는 EXTCLK를 이용하여 CPU clock으로 사용할 때 5배배를 하기 위해 CLKMD0, CLKMD1을 high로 만들었다.

그림 4-3 는 외부에 crystal을 이용하여 입력 클럭으로 사용하는 경우이다.

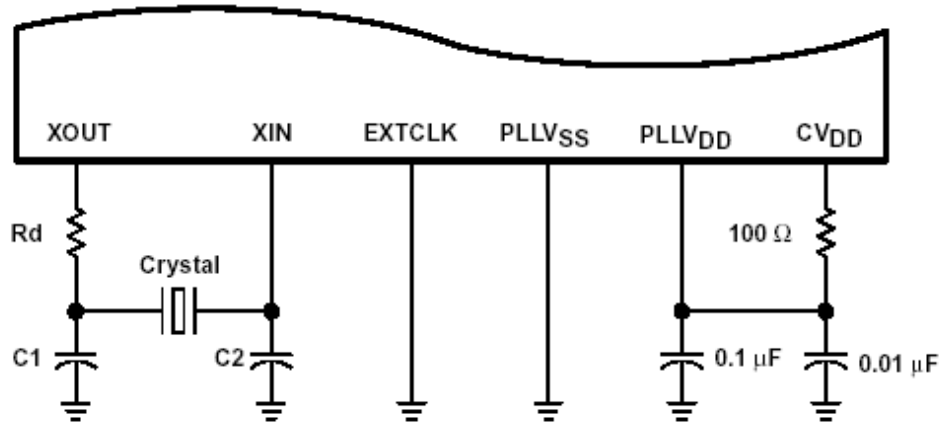


그림 4-3 Crystal를 이용한 모드

Fig. 4-3 Self oscillation mode

표 4-3 일반적인 crystal을 이용한 회로 구성

Table 4-3 Typical crystal circuit loading

FREQUENCY (MHz)	R_d (Ω)	C_1 (pF)	C_2 (pF)	$CL^{\dagger}$ (pF)	$RL^{\dagger}$ (Ω)
2	4.7k	18	18	12	200
5	2.2k	18	18	12	60
10	470	15	15	12	30
15	0	15	12	12	25
20	0	9	9	10	25

[†] CL and RL are typical internal series load capacitance and resistance of the crystal.

본 논문에서는 외부 crystal을 12MHz를 사용하기 때문에 표 4-3를 보면 R_d , C_1 , C_2 값을 10MHz와 15MHz 사이의 주파수이기 때문에 R_d 를 100 Ω , C_1 과 C_2 를 10pF를 사용하였다.

4.2 FPGA

ALTERA사의 ACEX1K 계열중에 EP1K100으로 10만 게이트, 208핀의 사양을 가진 EP1K100QC-208칩을 선택하여 사용하였다. 이 칩은 외부 핀으로 사용자가 쓸 수 있는 일반적인 I/O의 수가 147개이다.

- . address decoder를 위한 address line (input) : 24
- . data bus (input/output) : 16
- . DA Convertor Control signal : 6
- . AD Convertor Control signal : 11
- . Serial Communication Control signal : 15
- . Parallel Communication Control signal : 3
- . PWM Pulse Control signal : 12
- . Encode Pulse Control signal : 3
- . DSP Control signal : 25
- . etc : 8

총 122핀이 필요하므로 208핀을 선택하였고,

delay time이 1ns에서 5ns까지의 종류가 있는데 delay time이 1ns 줄어들수록 칩의 가격이 두배 정도로 비싸진다. 본 논문에서 적용한 것은 delay가 3ns이지만 외부의 입력 클럭을 DSP의 H1을 이용하기 때문에 한 사이클의 시간이 약 17ns인 것을 사용한다.

그림 4-4은 EP1K100QC-208의 내부 구조를 나타내고 있다.

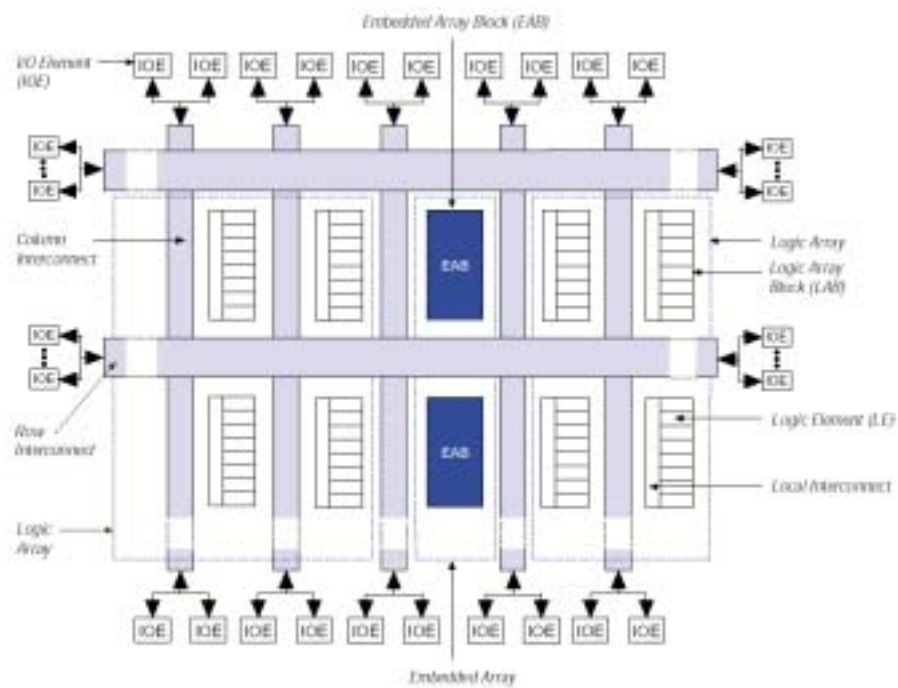


그림 4-4 EP1K100QC-208의 내부 구조

Fig. 4-4 Internal structure of EP1K100QC-208

EP1K100QC208에 프로그램을 할 수 있는 방법으로 여러 가지가 있지만 본 논문에서 사용한 방법은 byteblaster를 이용하여 프로그램을 다운로드 하였다. byteblaster를 이용하여 다운로드 하는 방법은 JTAG mode 와 PS mode 가 있는데 JTAG 모드는 EP1K100QC208에 로드할 수 있는 외부 메모리인 EPC2에 프로그램을 다운로드 하는 방법으로 다운로드를 한 후에 전원을 OFF하고 다시 ON하면 EPC2에 있는 프로그램이 EP1K100QC208에 로딩되는데 프로그램의 확장자가 *.pof이다. PS mode는 EP100QC208안에 SRAM에 직접 프로그램을 하는 방법으로 전원 OFF 시키면 내부의 프로그램은 지워져 버린다. 여기에서 다운로드 되는 프로그램의 확장자는 *.sof이다. 표 4-4에 각 모드 별로 PC의 병렬 포트(25P)와의 연결 상태를 나타내었다. 표 4-5은 주제어기 보드에 연결될 byteblaster의 포트 핀 연결번호를 나타낸 것이다. 그림 4-5은 byteblaster의 내부 회로도이다. 그림 4-6은 실제 회로를 나타내고 있다.[13][14][15]

표 4-4 바이트블라스트 25pin 연결 설정

Table 4-4 Byteblaster 25-pin header pin-out

Parallel Port PIN	JTAG mode	PS mode
2	TCK	DCLK
3	TMS	nCONFIG
8	TDI	DATA0
11	TDO	CONF_DONE
13	NC	nSTATUS
15	GND	GND
18 ~ 25	GND	GND

표 4-5 바이트블라스트 10핀 연결 설정

Table 4-5 Byteblaster 10-pin female pin-out

PIN	JTAG mode	PS mode
1	TCK	DCLK
2	GND	GND
3	TDO	CONFIG_DONE
4	VCC	VCC
5	TMS	nCONFIG
6	NC	NC
7	NC	nSTATUS
8	NC	NC
9	TDI	DATA0
10	GND	GND

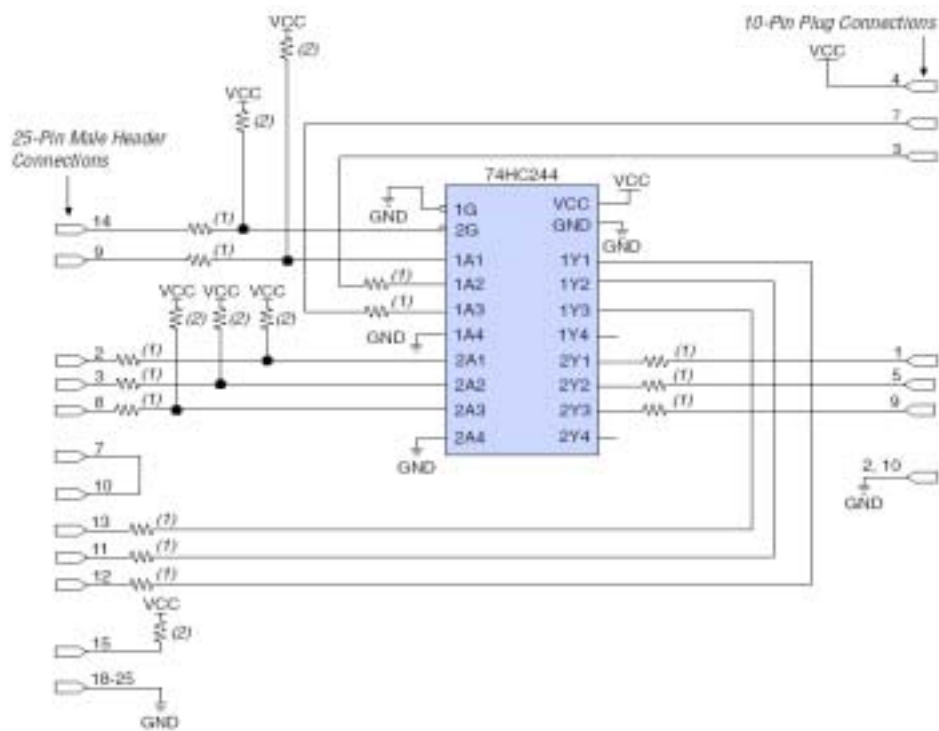


그림 4-5 Byteblaster 내부 회로

Fig. 4-5 Internal circuit of byteblaster

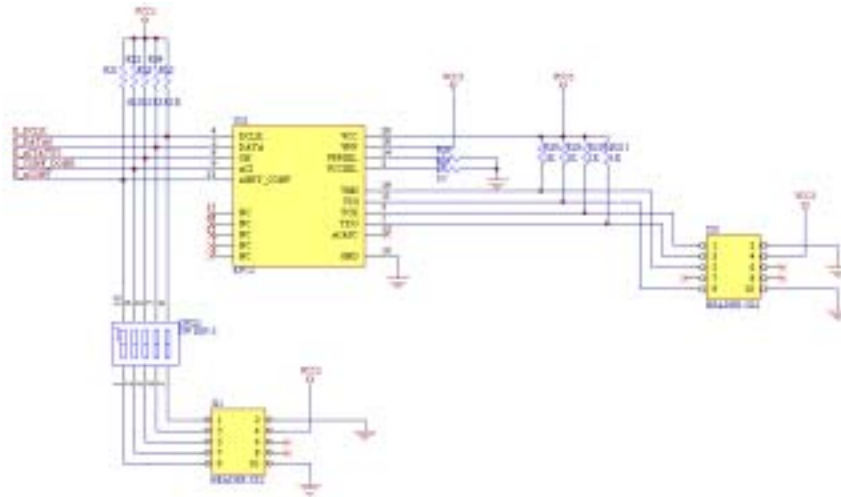


그림 4-6 PS mode 와 JTAG mode 회로

Fig. 4-6 Circuit of PS mode and JTAG mode

4.3 Analog to Digital Converter

본 논문에 사용한 ADS7864는 TI사의 제품으로 6개의 채널을 가지고 있으며 입력 클럭으로는 8MHz를 사용한다. 내부 구조를 보면 그림 4-7과 같으며 적용회로를 그림 4-8과 같이 하였고, conversion동작은 어떤 한 채널에 대해 클럭의 falling edge에서 $\overline{HOLDx}$ 신호를 low로 하면 그 채널이 선택되며 앞서 conversion중이던 것이 완료되지 않았으면 기다리고, 그렇지 않으면 conversion을 시작하게된다. 선택되지 않은 채널은 hold mode로 있게된다. 그리고 동일 채널에 대해서 conversion이 완료되지 않은 상태에서 다시 $\overline{HOLDx}$ 신호가 들어오면 conversion 명령을 무시하게되고 다른 채널에 대해서는 같은 클럭에서 $\overline{HOLDx}$ 신호가 들어오면 우선적으로 A 채널, B 채널, C 채널 순으로 conversion을 하게된다. 한 채널을 conversion 하는 시간은 $125\text{ns}(=1/8\text{MHz}) \cdot 12.5\text{cycle} = 1.56\mu\text{s}$ 이고 동일 채널에서 다음 conversion 명령을 주기 위해서는 최소한 $125\text{ns}(=1/8\text{MHz}) \cdot 16\text{cycle} = 2\mu\text{s}$ 의 시간이 필요하다. conversion 된 데이터는 A2~A1를 이용하여 읽을 채널을 선택하고 $\overline{RD}$ 와 $\overline{CS}$ 가 low로 최소 30ns동안 있으면 읽을 수 있는데 DATA15~12 bit는 유효한 데이터인지 어떤 채널의 데이터인지를 알려주고 DATA11~0 bit의 데이터가 실제 conversion한 데이터를 나타낸다.[16]

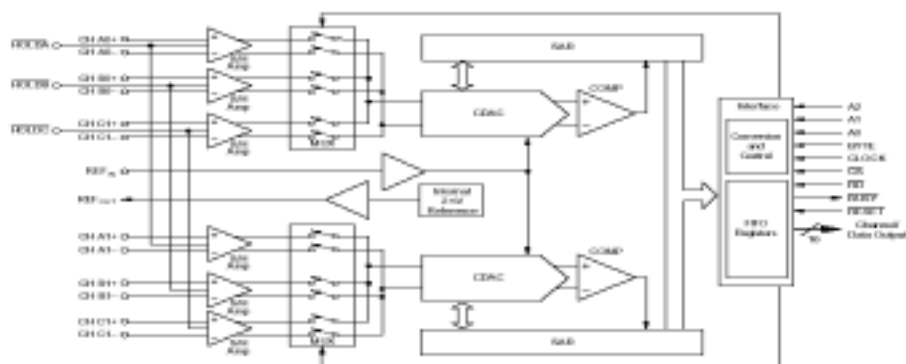


그림 4-7 ADS7864의 내부 구조

Fig. 4-7 Internal structure of ADS7864

그림 4-9은 ADS7864 각 채널의 신호 입력 회로도이다.

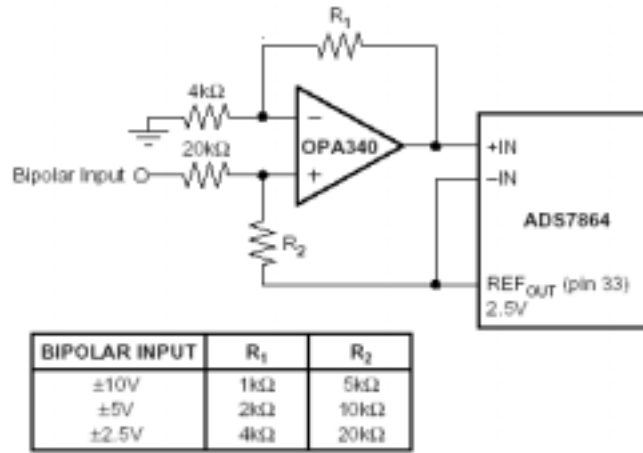


그림 4-9 A/D 입력 신호 회로

Fig. 4-9 A/D Input signal circuit

ADS7864의 각 채널은 바이폴라로 입력을 받으며 아날로그 입력신호의 크기를 저항 R_1, R_2 의 설정에 의해 결정된다. 그림 4-10에서 보면 채널로 입력을 받을 때 외부의 노이즈의 영향력을 줄이기 위해 BNC connector를 사용하였고 low pass filter를 사용하여 고속 인버터의 스위칭 주파수에 의한 영향을 제거 시켰다.

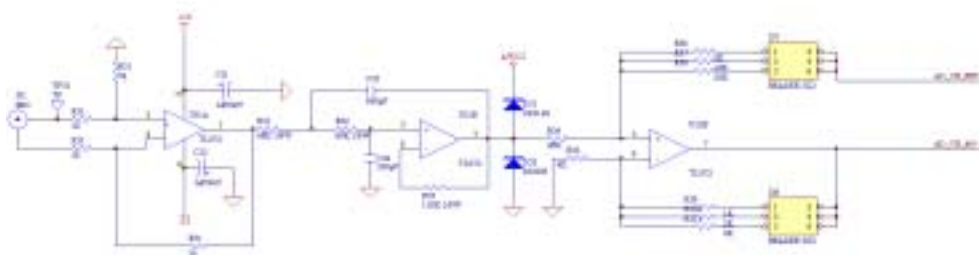


그림 4-10 A/D 채널 입력 회로

Fig. 4-10 A/D channel input circuit

4.4 Digital to Analog Converter

본 논문에 사용한 DAC는 DAC7864로써 TI사의 제품으로 4개의 채널을 가지고 있다. 내부 구조는 그림 4-11과 같고 외부적용 회로는 그림 4-12과 같이 하였다. 출력 동작은 표 4-6에서 12bit의 데이터를 $R/\overline{W}$, $\overline{CS}$, $\overline{LDAC}$ 가 low이고 $\overline{RESET}$ 은 high 상태이면 A0,A1에 의해서 채널을 선택하여 출력 하게된다. $\overline{CS}$ 가 low로 enable된 상태에서 약 100ns에서 최대 160ns이상 이 되어야 유효한 데이터가 채널을 통해서 12비트의 resolution을 가지는 신호로 아날로그 출력으로 보낼 수 있다. 이 DAC에 레퍼런스전압을 +/-2.5V로 주고 아날로그 출력 전압의 범위는 -2.5V ~ +2.5V이다.[17] 디지털 출력 전압의 크기를 계산하는 방법은

$$V_{OUT} = V_{REFL} + \frac{(V_{REFH} - V_{REFL}) * N}{4096} \text{ 와 같다.}$$

$V_{REFH} = +2.5V$, $V_{REFL} = -2.5V$ 이고 N은 입력 디지털 12bit 값이다.

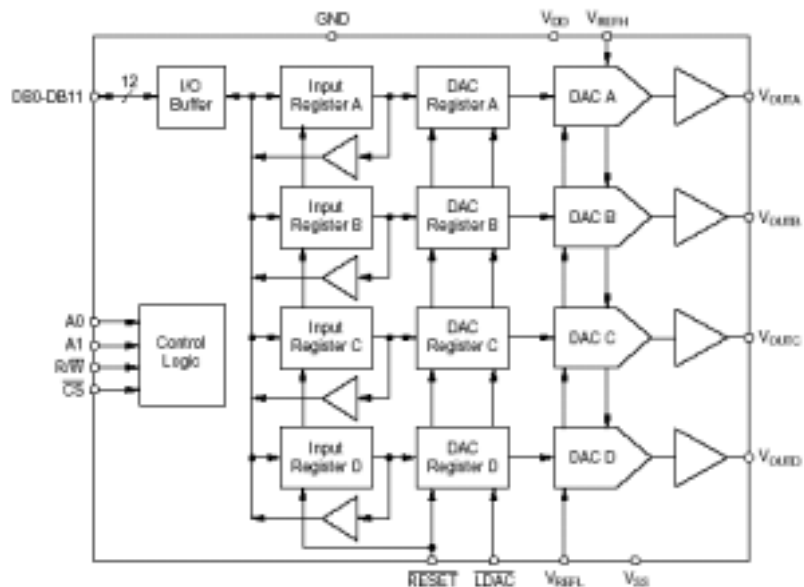


그림 4-11 DAC7864의 내부 구조

Fig. 4-11 Internal structure of DAC7864

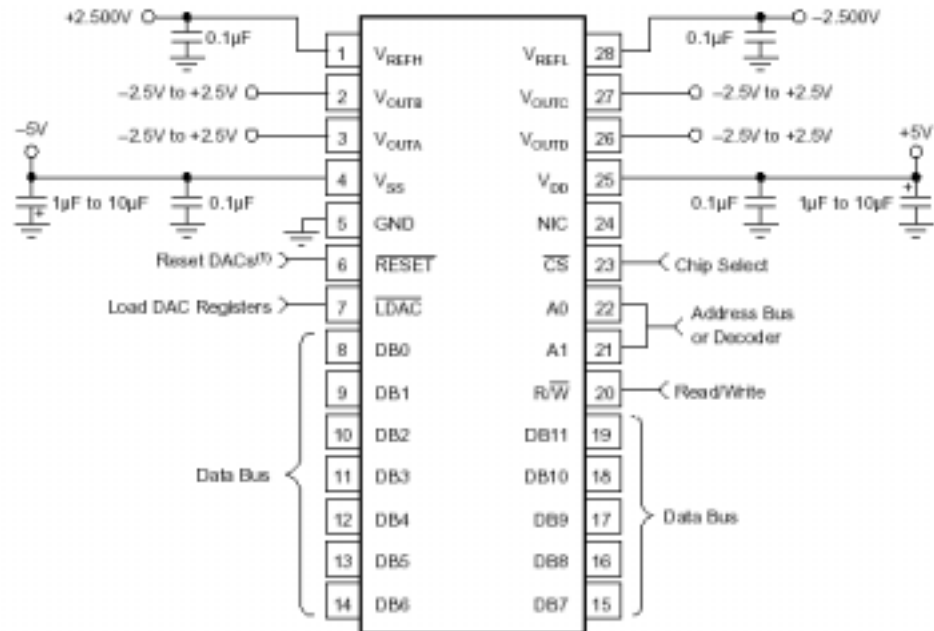


그림 4-12 DAC7625 적용 회로

Fig. 4-12 Application circuit of DAC7625

표 4-6 DAC7625 제어로직 진리표

Table 4-6 DAC7625 control logic truth table

A1	A0	R/W	CS	RESET	LDAC	SELECTED INPUT REGISTER	STATE OF SELECTED INPUT REGISTER	STATE OF ALL DAC REGISTERS
L ⁽¹⁾	L	L	L	H ⁽²⁾	L	A	Transparent	Transparent
L	H	L	L	H	L	B	Transparent	Transparent
H	L	L	L	H	L	C	Transparent	Transparent
H	H	L	L	H	L	D	Transparent	Transparent
L	L	L	L	H	H	A	Transparent	Latched
L	H	L	L	H	H	B	Transparent	Latched
H	L	L	L	H	H	C	Transparent	Latched
H	H	L	L	H	H	D	Transparent	Latched
L	L	H	L	H	H	A	Readback	Latched
L	H	H	L	H	H	B	Readback	Latched
H	L	H	L	H	H	C	Readback	Latched
H	H	H	L	H	H	D	Readback	Latched
X ⁽³⁾	X	X	H	H	L	NONE	(All Latched)	Transparent
X	X	X	H	H	H	NONE	(All Latched)	Latched
X	X	X	X	L	X	ALL	Reset ⁽⁴⁾	Reset ⁽⁴⁾

그림 4-13은 DAC의 출력측의 회로이고 그림 4-14는 레퍼런스 전압을 regulator하는 회로이다.

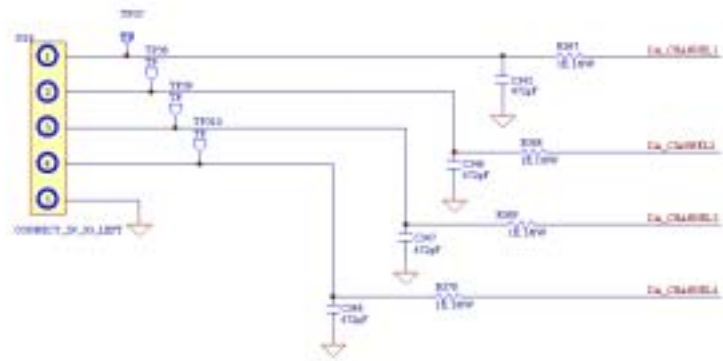


그림 4-13 DAC 출력 회로

Fig. 4-13 DAC output circuit

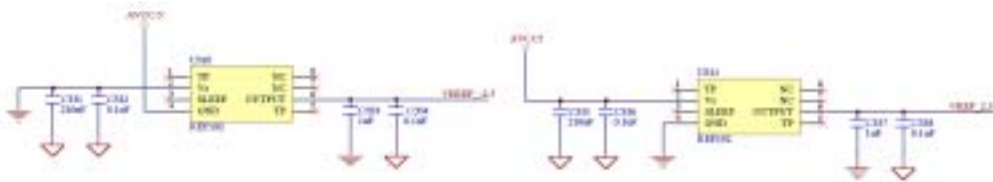


그림 4-14 DAC 레퍼런스 전압 발생 회로

Fig. 4-14 Reference voltage generate circuit of DAC

4.5 Control signal transmitter/receiver

본 논문에서는 광케이블을 이용하여 PWM 제어신호를 송수신 할려고 한다. 데이터 송수신 신호 전송비가 5MBd로서 높은 전송비를 가지고 주제어기와 전력변환부사이의 거리가 약 1m이내에서 주로 작업한다고 가정하여 표 4-7에서 송신부에 HFBR-1521, 수신부에는 HFBR-2521인 전송거리를 20m의 거리 내에서 유효한 동작을 할 수 있는 것을 선정하였다.[19]

표 4-7 HFBR 종류별 송수신 거리

Table 4-7 Data line distance of HFBRs

Signal Rate	Distance (m) 25 °C	Distance (m)	Transmitter	Receiver
40 kBd	120	110	HFBR-1523	HFBR-2523
1 MBd	20	10	HFBR-1524	HFBR-2524
1 MBd	55	45	HFBR-1522	HFBR-2522
5 Mbd	30	20	HFBR-1521	HFBR-2521

회로의 구성은 그림 4-15과 같이 하였다.

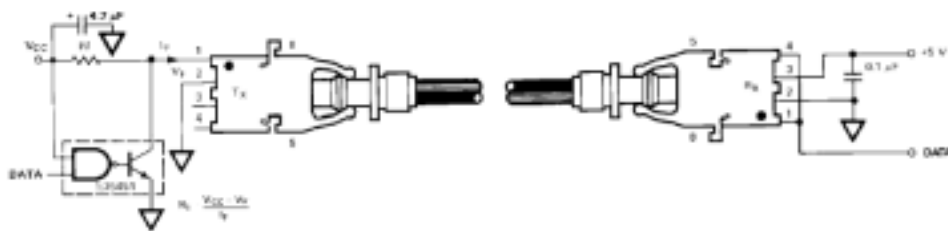


그림 4-15 HFBR1521/HFBR2521 적용 회로

Fig. 4-15 Application circuit of HFBR1521/HFBR2521

송신부의 R_1 의 값을 선정하기 위해 먼저 표 4-8를 보면 순방향전압강하 (V_F)는 1.67V이다. 그러나 전송거리가 20m가 아닌 5m로 두었을 때 그림 4-16에서 I_f 의 전류를 30mA 정도임을 알 수 있다. 따라서 V_{cc} 가 5V일 때

$$R_1 = \frac{V_{cc} - V_F}{I_F} = \frac{5V - 1.67V}{30mA} = 111\Omega$$

이 되므로 표준저항을 적용하여 100 Ω 으로 설정을 하였다.

표 4-8 송신기의 전기적/광적 특성

Table 4-8 Transmitter electrical/optical characteristics

Parameter	Symbol	Min.	Typ. ^[3]	Max.	Units	Conditions	Ref.
Transmitter Output Optical Power	P_T	-16.5		-7.5	dBm	$I_{Fdc} = 60 \text{ mA}$	Notes 1, 2
		-14.3		-8.0	dBm	$I_{Fdc} = 60 \text{ mA}, 25^\circ\text{C}$	
Output Optical Power Temperature Coefficient	$\Delta P_T/\Delta T$		-0.85		%/ $^\circ\text{C}$		
Peak Emission Wavelength	λ_{PK}		660		nm		
Forward Voltage	V_F	1.45	1.67	2.02	V	$I_{Fdc} = 60 \text{ mA}$	
Forward Voltage Temperature Coefficient	$\Delta V_F/\Delta T$		-1.37		mV/ $^\circ\text{C}$		Fig. 9
Effective Diameter	D		1		mm		
Numerical Aperture	NA		0.5				
Reverse Input Breakdown Voltage	V_{BR}	5.0	11.0		V	$I_{Fdc} = 10 \mu\text{A}$, $T_A = 25^\circ\text{C}$	
Diode Capacitance	C_O		80		pF	$V_F = 0$, $f = \text{MHz}$	
Rise Time	t_r		80		ns	10% to 90%, $I_F = 60 \text{ mA}$	Note 3
Fall Time	t_f		40		ns		

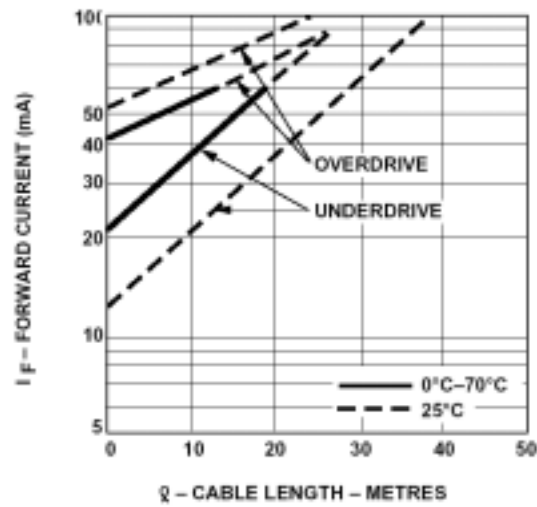


그림 4-16 표준 케이블 사용시 거리당 순방향 전류와의 관계

Fig. 4-16 Guaranteed system performance with standard cable

송신부와 수신부의 각각의 내부를 보면 그림 4-17로 나타난다.



(a) 송신부

(b) 수신부

그림 4-17 HFBR1521/HFBR2521 내부 회로

Fig. 4-17 Internal circuit of HFBR1521/HFBR2521

그림 4-18과 그림 4-19는 실제 송수신 회로를 나타내고 있다.

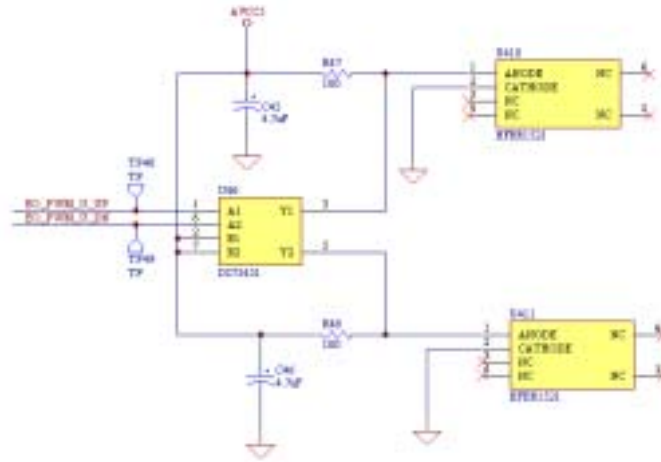


그림 4-18 HFBR1521를 이용한 송신 회로

Fig. 4-18 Transmitter circuit applied HFBR1521

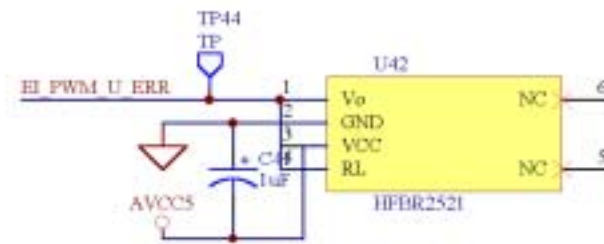


그림 4-19 HFBR2521를 이용한 수신 회로

Fig. 4-19 Receiver circuit applied HFBR2521

4.6 Encoder Interface

Encoder의 출력방식에 따라 6가지로 분류할 수 있다.

첫째, Open Collector 타입으로 그림 4-20과 같이 transistor의 collector를 부하에 연결하지 않고 그대로 출력하는 방식(사용자가 임의의 부하 연결 사용)으로 전송거리 50m이하 ($V_{cc}=5V$) 100m이하 ($V_{cc}=12V$)에서 주로 사용한다.

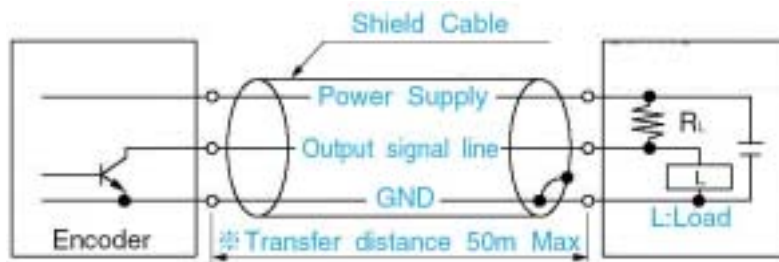


그림 4-20 오픈 콜렉트형 엔코더

Fig. 4-20 Open-collector type of encoder

둘째, 전압출력(voltage output) 타입으로 그림 4-21에서 보듯이 encoder의 전원과 출력 transistor collector를 사이에 외부부하(L)를 연결하여 출력하는 방식으로 전송거리 2m 이하에 주로 사용한다.

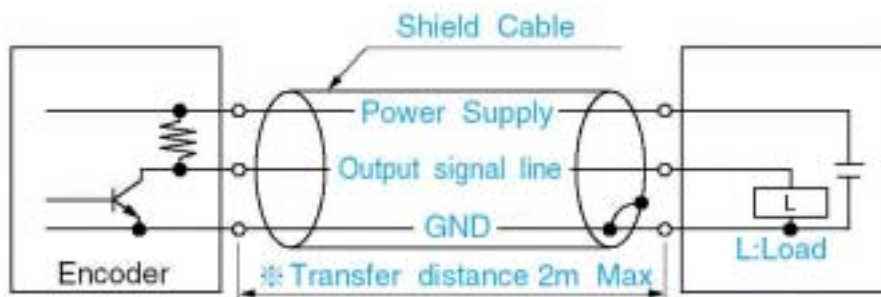


그림 4-21 전압 출력형 엔코더

Fig. 4-21 Voltage output type of encoder

세째, Line Drive 타입으로 그림 4-22에서 보듯이 data 전송용의 출력전용

IC를 이용하여 출력하는 방식으로 전송거리가 먼 것에 우수하여 장거리 전송
사용에 주로 사용한다.

noise에 강하고 전송거리 1km이하에 주로 사용한다.

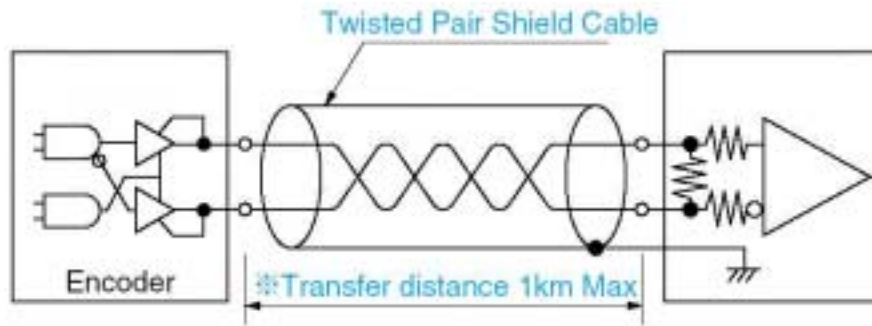


그림 4-22 Line drive형 엔코더

Fig. 4-22 Line drive type of encoder

넷째, TTL 타입으로 그림 4-23과 같은 형태이며 출력단에 TTL IC 채용한
것으로 전송거리가 2-3m 이하로써 전류가 작고 noise에 취약한 단점을 가지
고 있다.

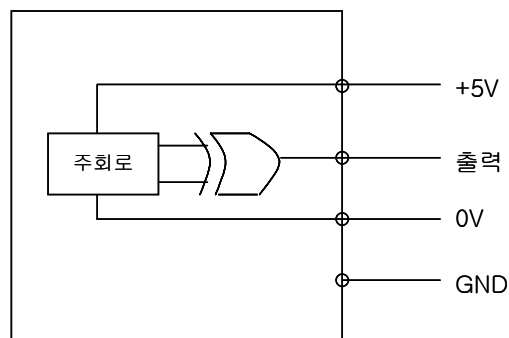


그림 4-23 TTL형 엔코더

Fig. 4-23 TTL type of encoder

다섯째, Complementary (Push-Pull) 타입으로 그림 4-24에서 보면 상보대칭
접속방법으로 공동 common이므로 전송전류가 상시 흘러 noise에 강하다.

500m 이하($i=+10\text{mA}$)에 주로 사용한다.

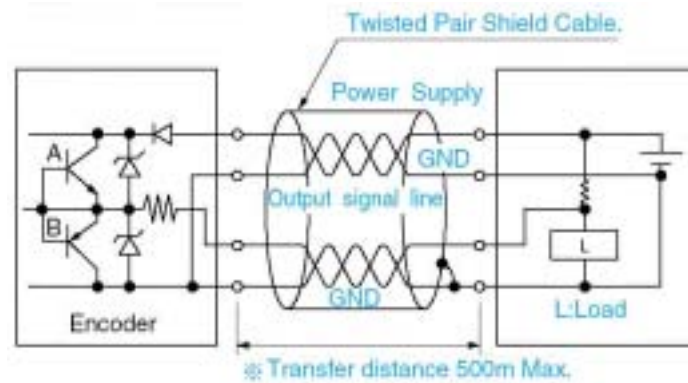


그림 4-24 Complementary형 엔코더

Fig. 4-24 Complementary type of encoder

여섯째, Totempole 타입으로 그림 4-25에서 보면 트랜지스터 쌍으로 이루어져 있으며 출력신호가 "1"일 경우는 상부 TR1은 ON, 하부 TR2은 OFF이며 출력신호가 "0"일 경우는 이와 반대이다. 특징으로는 전송 전류가 상시 흐르므로(유출, 유입전류) noise에 강하고 파형왜곡이 적다. 본 논문에 적용한 것은 line drive type을 사용하기 위해 line drive receiver인 26LS31을 구성하였고 open collector 형, totempole 형 및 전류구동형으로 encoder 신호를 수신 가능하게 회로를 구성하였다.

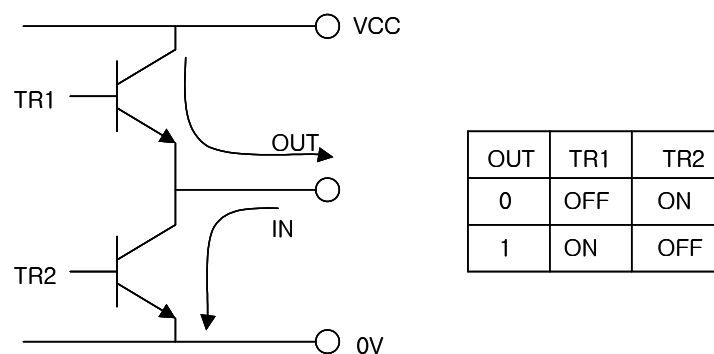


그림 4-25 토템폴형 엔코더

Fig. 4-25 Totempole type of encoder

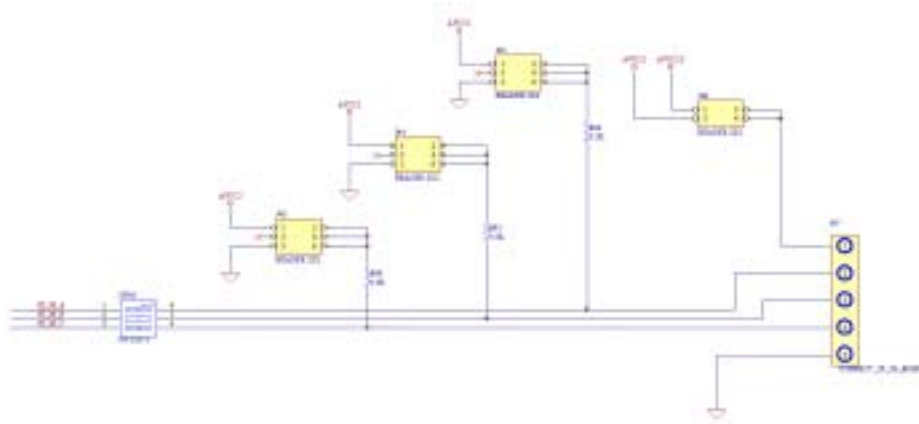


그림 4-26 엔코더 입력 회로

Fig. 4-26 Input circuit of encoder

4.7 전원

주제어기 보드에 필요한 입력 전원은 +15V/-15V와 +5V로 SMPS를 통해서 들어온다. 추가로 +12V와 +5V/-5V, +3.3V, +2.5V, +1.8V가 필요로 하여 그림 4-27과 같이 전원 회로를 구성 하였다. 먼저 FAN1537B를 통해서 기본적으로 V_1 에 +3.3V전원이 나오고 R_{610} 과 R_{611} 의 저항값에 따라 V_2 의 전압이 결정된다.[21]

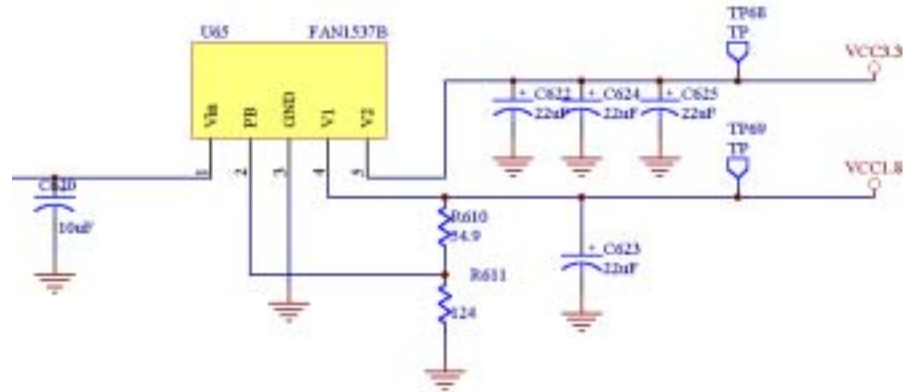


그림 4-27 +3.3V_{DC}, +1.8V_{DC} 발생 회로

Fig. 4-27 Regulation circuit of +3.3V_{DC} and +1.8V_{DC}

V_2 를 구하는 수식은 아래와 같다.

$$V_2 = V_{REF}(1 + R_{610}/R_{611})$$

여기서 V_{REF} 는 1.25V이고 V_2 를 2.5V전원이 필요 하다면

$$2.5V = 1.25V(1 + R_{610}/R_{611})$$

에서 $R_{610}/R_{611}=1$ 로 두면 된다. 여기서는 약 100Ω의 값을 R_{610} , R_{611} 의 값으로 설정한다. 그리고 1.8V는 $R_{610}/R_{611}=0.44$ 이라서 $R_{610}=56Ω$ 이고 $R_{611}=127Ω$ 으로 설정을 하였다

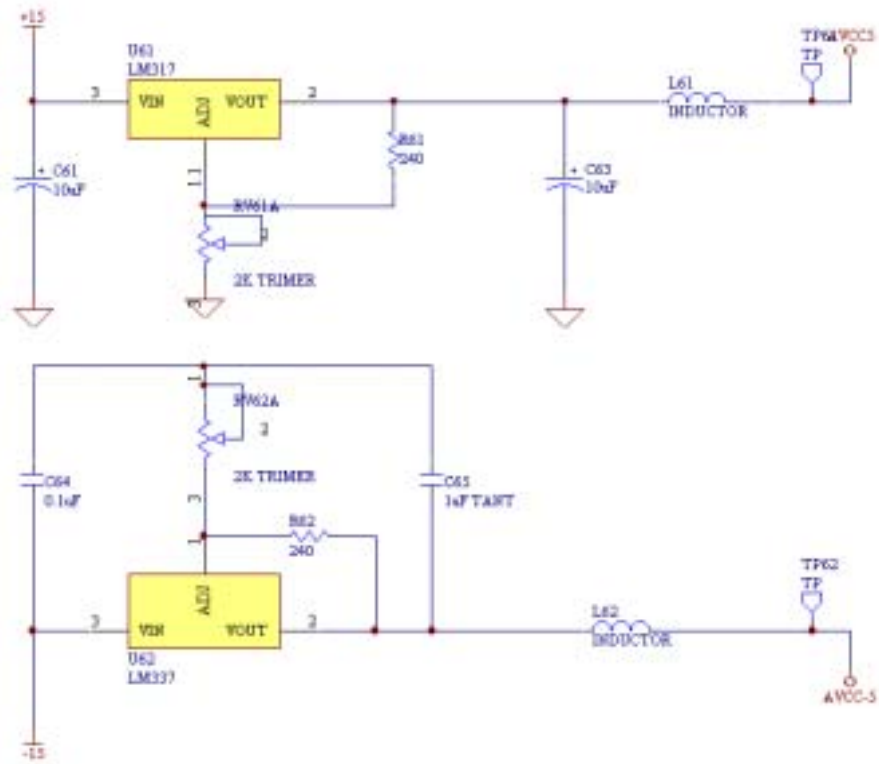


그림 4-28 +5V_{DC} , -5V_{DC} 발생회로

Fig. 4-28 Regulation circuit of +5V_{DC} and -5V_{DC}

그림 4-28회로에서 $V_o = 5V$ 를 만들려고 할 때 추천 조건에서 $I_{adj} = 100\mu A$ 로 주어진 값을 사용하면 $V_o = 1.25V(1 + RV_{61A}/R_{61}) + I_{adj}RV_{61A}$ 는 $5V = 1.25V(1 + RV_{61A}/R_{61}) + 100\mu A RV_{61A}$ 이다. $100\mu A RV_{61A}$ 의 값은 무시할 수 있어서 $3 = RV_{61A}/R_{61}$ 이 되는데 $R_{61} = 240\Omega$ 이면 $RV_{61A} = 720\Omega$ 이 된다. C_{63} 는 안정성을 위해서 필요하는 것이 아니라 과도상태(transient respose)를 개선시키기 위해서 필요하다. 레귤레이터가 power supply의 필터에서 4인치 이상의 거리에 위치하면 C_{61} 를 적용할 것을 추천하고 있다.

$V_o = -5V$ 를 만들려고 할때

$$V_o = -1.25 V(1 + RV_{62A}/R_{62})$$

에서 $-5V = -1.25 V(1 + RV_{62A}/R_{62})$ 이다.

따라서, $3 = RV_{62A}/R_{62}$ 이 되는데 $R_{62} = 240\Omega$ 으로 두면 $RV_{62A} = 720\Omega$ 이 된다. C_{65} 는 안정성을 위해서 필요하며 $1\mu F$ 의 솔리드 탄탈 커패시터를 사용하던지 $10\mu F$ 의 알루미늄 전해 커패시터를 사용할 것을 메이커에서 추천하고 있다. 본 논문에서는 주파수 특성이 좋은 $1\mu F$ 의 솔리드 탄탈 커패시터를 선정하였다.[22][23]

4.8 Reset 및 Power Supervisor

그림 4-29과 같이 DSP의 입력 reset신호는 스위치 입력인 manual reset, 초기 전원 인가시 또는 전원의 변동 마진폭 보다 큰 입력일 때 MAX708에 의해 low를 유지 시켜주는 것 과 EP1K100QC208의 CONFIG_DONE신호 와 watchdog reset을 조합하여 DSP의 reset 입력이 된다. 그림 4-30은 타이밍도를 나타내었다.

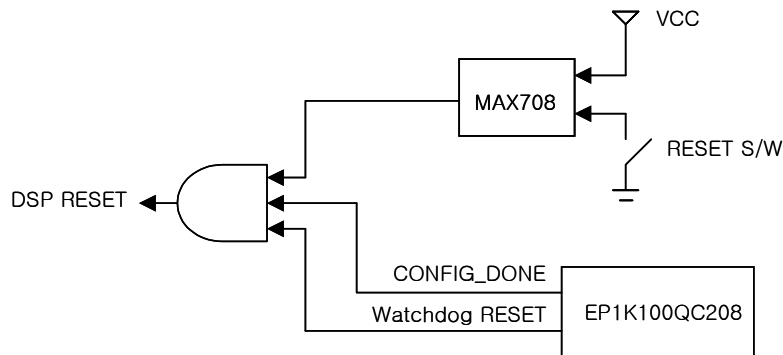


그림 4-29 DSP 리셋입력 블록도

Fig. 4-29 Reset input block diagram of DSP

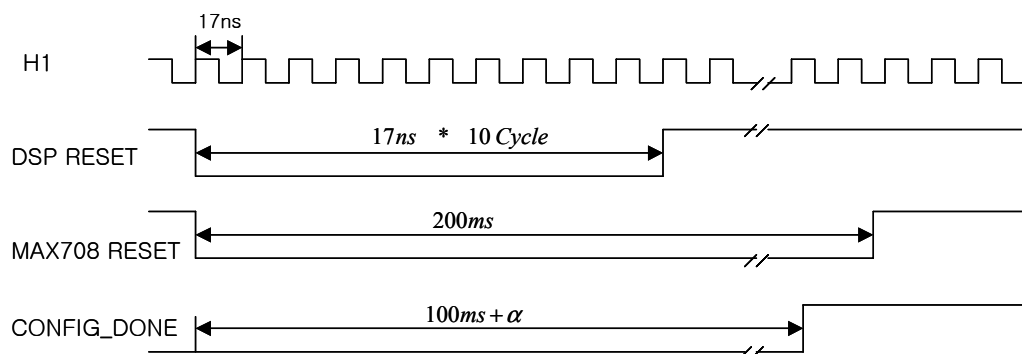


그림 4-30 전체 reset 타이밍

Fig. 4-30 Reset timing of total

DSP에서 reset을 인식하는데 요구되는 시간은 그림 4-31과 같이 external clock의 상승에지에서 H1의 상승에지 까지 $t_d(\text{EXTCLKH-H1H})$ 가 2ns~7ns 까지 걸린 후 reset신호가 low로 10개의 H1 클럭 사이클 동안 유지하면 reset을 인식하게된다.

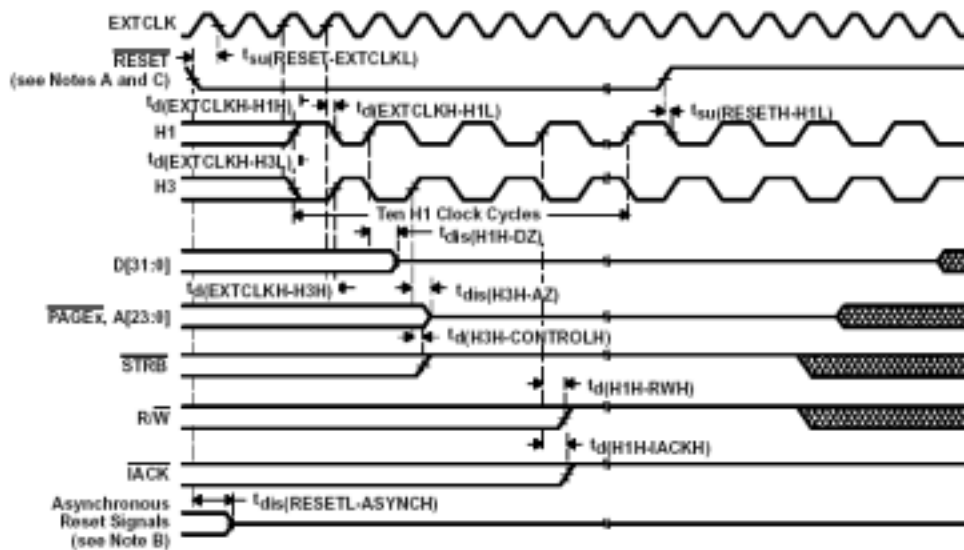


그림 4-31 리셋 시간

Fig. 4-31 Reset timing

power supervisor용 회로는 MAX708를 사용하여 구성하였다. 그림 4-32는 MAX708내부 블록도를 나타내었다. 그리고, 그림 4-33을 보면 MAX708은 전원 전압의 변화가 있어 threshold 전압으로 4.38V 이하가 되었을 때 reset으로 인식하고, manual reset 입력($\overline{MR}$)은 low로 되면 reset 신호로 인식하게 된다. 그리고 reset후 마이크로프로세서 초기화를 위해 200ms(t_{RP})동안 low로 유지시켜 주어야 된다. 그림 4-34은 실제 구성한 회로도이다.

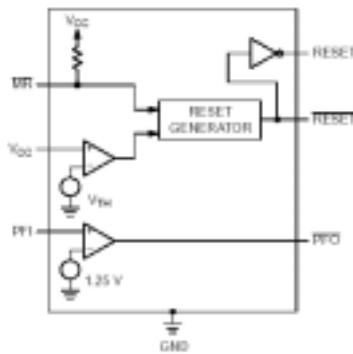


그림 4-32 MAX708 내부 블록도

Fig. 4-32 Internal block diagram of MAX708

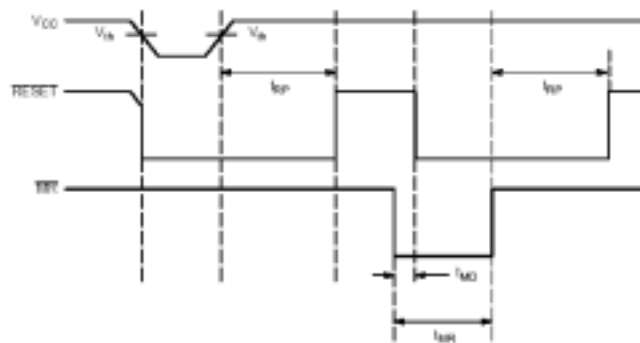


그림 4-33 MAX708 리셋 타이밍

Fig. 4-33 Reset timing of MAX708

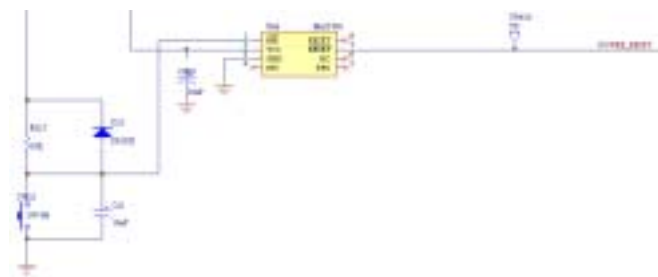


그림 4-34 Power supervisor 회로

Fig. 4-34 Power supervisor circuit

V. VHDL에 의한 FPGA 설계

5.1 전체 설계부

그림 5-1은 전체 구성도를 나타내고 있으며 각각의 기능별로 블록화 시켜서 설계하였다. 구성을 보면 SVPWM signal generator, encoder signal generator, watch dog timer, expanded interrupt controller, wait signal generator, ADC interface control signal generator, DAC interface control signal generator, serial communication control signal generator, parallel communication control signal generator 및 reset signal generator로 구성되어 있다.

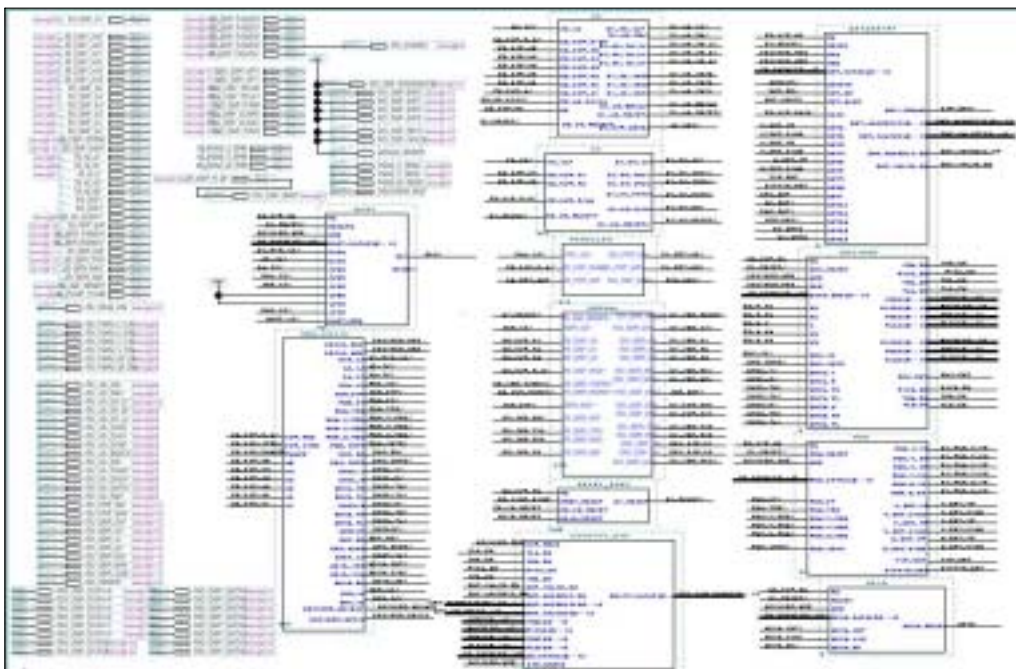


그림 5-1 전체 VHDL 설계 구성도

Fig. 5-1 Total VHDL block diagram

5.2 SVPWM(Space Vector Pulse Width Modulation)

그림 5-2는 SVPWM의 인터럽터가 발생하는 시점을 나타내었다.

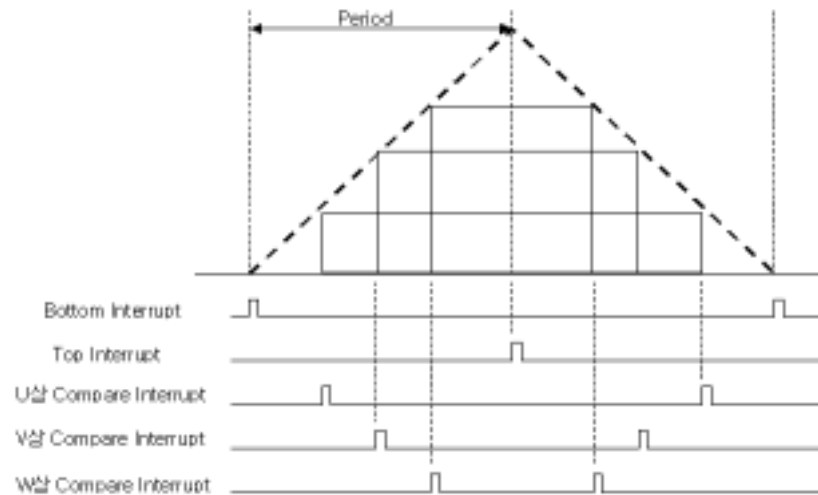


그림 5-2 SVPWM에서 인터럽트 발생 개념도

Fig. 5-2 Conceptual diagram of interrupt generation in SVPWM

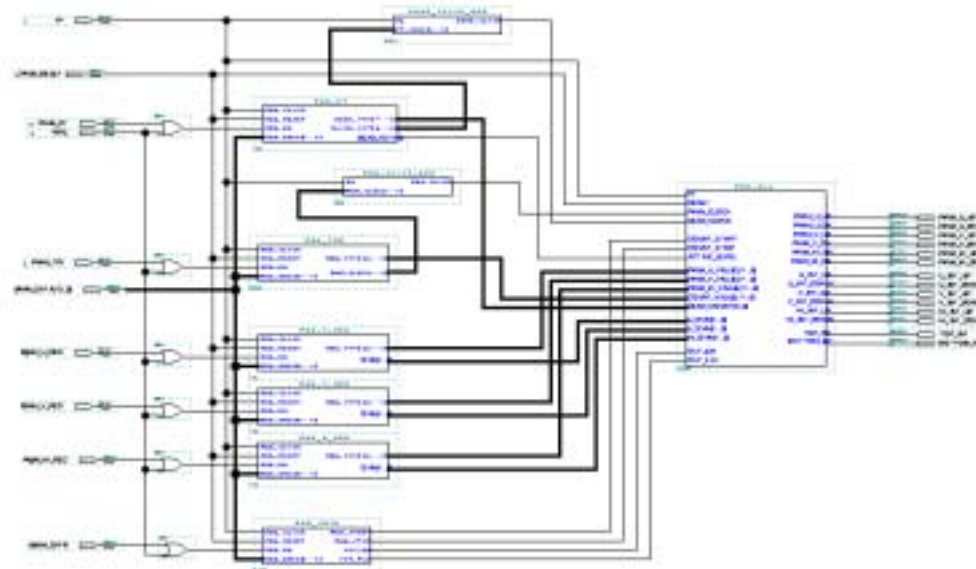


그림 5-3 SVPWM generator 구성도

Fig. 5-3 Symbolic diagram of SVPWM generator

그림 5-3에는 실제 설계 구성도로써 PWM를 만들기 위한 제어를 PWM control register를 통해 PWM 발생을 시작하게 하고 정지시킬 수 있고 출력 active level에 맞게 설정을 할 수 있으며 출력을 강제로 low/high되게 할 수 있다.

PWM dead time register를 통해 dead time의 클럭과 dead time 시간을 가변할 수 있으며, PWM active level을 low 또는 high로 설정할 수 있다.

PWM period register에 한 주기의 값을 넣을 수 있는데 한 주기를 카운터하는 클럭도 설정을 할 수 있게 하였다.

PWM U/V/W-phase value register에 PWM 한 주기 중에 각 상의 비교시점의 값을 적을 수 있게 하였다. 뿐만 아니라 한 상에 대해 윗상과 아래상중 한 상만 출력으로 보낼 수 있고 두 상 모두 출력으로 보낼 수 있으며 또한 두 상 모두 출력을 보내지 않게 할 수 있다.

* PWM_CNTR[= PWM Control Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	x	x	x	x	OEN	FLV	STOP	START

15 ~ 4 BIT : Don't care

3 BIT : OUT Enable Bit

'0' -> 2 BIT의 값에 따라 강제 출력에 의한 출력

(Force Mode Operation)

2 BIT : Force Level Output

강제로 출력 레벨을 결정함

'1' -> PWM 출력 파형을 강제로 High Level로 출력

'0' -> PWM 출력 파형을 강제로 Low Level로 출력

1 BIT : PWM Operation Stop

PWM Counter 동작을 중지 시키는 비트

‘1’ -> PWM Counter 동작을 중지 하지 않음

‘0’ -> PWM Counter 동작을 중지 함

‘1’ -> 일반적인 출력 (PWM Mode Operation)

0 BIT : PWM Operation Start

PWM Counter 동작을 시작하게 만드는 비트

‘1’ -> PWM Counter을 시작하게 함

‘0’ -> PWM Counter을 시작하지 않게 함

* PWM_DT[= PWM Dead Time Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x	x	PWMAL	DTCLK			DTC									

15 ~ 14 BIT --> Don't care

13 BIT --> PWM Active Level

‘1’ -> ACTIVE High Level

‘0’ -> ACTIVE Low Level

12 ~ 10 BIT --> Dead Time Clock 값 설정

입력 클럭 H1에 대해 분주하여 Dead Time 클럭을 발생
시킴

‘000’ -> H1 = 60Mhz

‘001’ -> (H1)/2

‘011’ -> (H1)/4

‘100’ -> (H1)/8

‘101’ -> (H1)/16

‘110’ -> (H1)/32

‘111’ -> (H1)/64

9 ~ 0 BIT --> Dead Time Count 값

Dead Time Counter 할 값을 적음

* PWM_TRI[= PWM Period Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x	PWMCLK			PWMPR											

15 BIT --> Don't care

14 ~ 12 BIT --> PWM Counter Clock 값 설정

'000' -> H1 = 60Mhz (DSP의 H1 출력 주파수)

'001' -> (H1)/2

'011' -> (H1)/4

'100' -> (H1)/8

'101' -> (H1)/16

'110' -> (H1)/32

'111' -> (H1)/64

11 ~ 0 BIT --> PWM Counting Period 값을 설정

$2^0(=0) \sim 2^{12}(=4096)$ 의 범위까지 설정 가능

*PWM_U/V/W_REG[=PWM U/V/W-Phase Value Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x	x	OVM		U/V/W											

15 ~ 14 BIT : Don't care

13 ~ 12 BIT : Switching Leg Selection Bits

'00' -> U/V/W 데이터에 따라 정상적으로 출력

'01' -> 윗 상만 ON 시킴

'10' -> 아래상만 ON 시킴

'11' -> 전부 OFF 시킴

11 ~ 0 BIT : U/V/W 의 비교시점의 각 값

5.3 Encoder Signal Processor

그림 5-4에서 보면 엔코더 입력 신호 중 A상,B상,Z상을 입력을 받거나 원점 신호가 없는 A상,B상만 입력신호를 받는 경우 모두 사용할 수 있게 각각의 모듈로 설계하였다. M/T method를 사용하였고 위치 값을 읽기 위해 stop을 하였을 때 그 값을 읽을 수 있는 시점에서 인터럽터가 발생되도록 하였다.

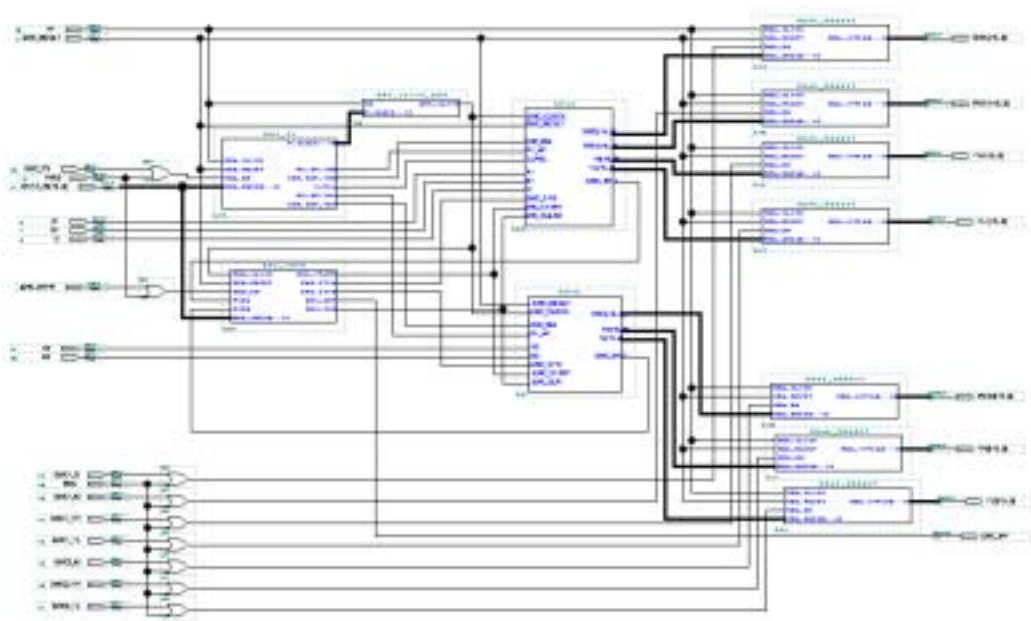


그림 5-4 Encoder signal processor 구성도

Fig. 5-4 Symbolic diagram of encoder signal processor

*ENC_CS[= Encoder Control Status Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F_CLK		x	x	PI_4 P _ONE	DIR_ DEF_ _ONE	x	x	x	C _POL	PI_4 P _TWO	DIR_ DEF_ _TWO	x	x	x	x

15 ~ 14 BIT : Encoder Clock을 결정하는 비트

‘00’ -> H1

‘01’ -> (H1) / 2

‘10’ -> (H1) / 4

‘11’ -> (H1) / 8

12 BIT : Mode

11 BIT : 4체배 펄스를 사용할 것인가 1체배 펄스를 사용할 것인가를 결정하는 비트

‘1’ -> 1체배 펄스를 사용

‘0’ -> 4체배 펄스를 사용

10 BIT : 사용하는 엔코더의 신호에 따라 아래와 같이 사용

‘1’ -> A 신호가 B 신호보다 90도 앞설 때

‘0’ -> B 신호가 A 신호보다 90도 앞설 때

6 BIT : 엔코더 C 신호의 극성을 설정

‘0’ -> C 신호의 극성이 ACTIVE LOW 인 경우

5 BIT : 4체배 펄스를 사용할 것인가 1체배 펄스를 사용할 것인가를 결정하는 비트

‘1’ -> 1체배 펄스를 사용

‘0’ -> 4체배 펄스를 사용

4 BIT : 사용하는 엔코더의 신호에 따라 아래와 같이 사용

‘1’ -> A 신호가 B 신호보다 90도 앞설 때

‘0’ -> B 신호가 A 신호보다 90도 앞설 때

* ENC_CNTR[= Encoder Control Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	x	x	CLR	INT	STO2	STO1	x	START

15 ~ 6 BIT : Don't care
5 BIT : Register Store Value Clear
4 BIT : Encoder Interrupt Occur Flag
3 BIT : Encoder 2 Sampling Time Out
2 BIT : Encoder 1 Sampling Time Out
0 BIT : '1' -> Encoder Start
 '1' -> Encoder Stop

5.4 Watchdog Timer

그림 5-5에서 보면 WDOG_ID값과 WDOG_CON1의 하위 8bit 값이 같은 값이 되지 않으면 watch dog controller register의 값을 변경하지 못한다. watchdog timer에 의해 일정 시간 이내에 timer counter값을 clear시키지 못하면 watchdog reset이 발생하게 된다. 일정한 watchdog timer내에 control register의 값을 계속 clear를 시켜 주어야한다.

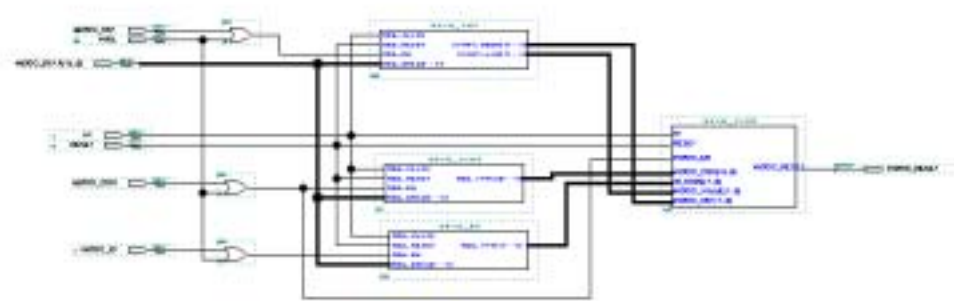


그림 5-5 Watchdog timer 구성도

Fig. 5-5 Symbolic diagram of watchdog timer

* WDOG_CNT[= Watchdog Counter Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
COUNT_HIGH								COUNT_LOW							

15 - 8 BIT : Watchdog Reset이 걸리면 내부의 카운터를 시작함

Watchdog Reaset이 걸린 후 Watchdog Reset으로 있는 시간이 내부의 카운터 값이 COUNT_HIGH 값이 될 때까지 카운터를 하게 된다. 내부의 카운터 값과 COUNT_HIGH 값이 같아지면 Watchdog Reset이 해제가 된다.

7 - 0 BIT : Watchdog를 실행을 시키면 COUNT_LOW값이 될 때까지
Watchdog Timer가 동작을 하면서 Watchdog Timer의 카운터를 증가시킨다.

이 Watchdog Timer 카운터 값이 COUNT_LOW의 값이 되기 전에 Watchdog Timer 카운터 값을 영으로 초기화를 시키지 않으면 COUNT_LOW의 값과 Watchdog Timer 카운터 값이 같아지면 Watchdog Reset이 발생하게 된다.

* WDOG_CON1[= Watchdog Control Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x	x	CMD		HLD	CLK_SEL			PASS_ID							

13 - 12 BIT : Watchdog Mode를 선택함

‘00’ --> HLD 값과 CLK_SEL값을 설정

WDOG 가 발생하게 되면

HLD(= 11 BIT) 가 ‘0’ 이면 Watchdog Reset 상태로 계속 되고

HLD(= 11 BIT) 가 ‘1’ 이면 Watchdog Reset이

WDOG_CNT 레지스터의 COUNT_HIGH 값 동안 Reset이 걸려 있다가 자동으로 회복되게 함

‘01’ --> Watchdog Reset동작을 하게 함

‘10’ --> Watchdog Timer의 카운터 값을 Clear 시킴

‘11’ --> Watchdog Timer를 동작을 시킴

이것을 동작시키지 않으면 Watchdog는 동작을 하지 않음 한번 동작을 하고 나면 정지시킬 수 없다.

11 BIT : ‘0’ 이면 Watchdog Reset 상태로 계속 되고

‘1’ 이면 Watchdog Reset이 WDOG_CNT 레지스터의

COUNT_HIGH 값 동안 Watchdog Reset이 걸려 있다가 자동
으로 회복되게 함

10 - 8 BIT : Watchdog Timer의 클럭을 설정하는 비트

'000' --> (H1)
'001' --> (H1)/ 128
'010' --> (H1)/ 256
'011' --> (H1)/ 512
'101' --> (H1)/ 1024
'110' --> (H1)/ 2048
'111' --> (H1)/ 4096

7 - 0 BIT : Watchdog Timer를 Clear를 시키기 위한 명령이나 Watchdog를
동작시키기 위한 명령 등 Watchdog에게 명령을 내릴 때
WDOG_ID의 값과 같은 값을 기록하지 않으면 명령을 내릴 수
없게 되어 있다.

* WDOG_ID[Watch Dog Password Number]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
x	x	x	x	x	x	x	x	WDOG_ID							

7 - 0 BIT : Watchdog를 정지시킬 때 이 WDOG_ID 값과 WDOG_CON1
의 WDOG_PASS의 값이 같지 않으면 Watchdog를 정지시키기
위한 명령을 내릴 수 없다.

이 값은 사용자가 Watchdog를 동작을 시키기를 원할 때 이
Default 값을 사용하여도 되고 사용자가 다른 값을 설정하여
도 된다

5.5 Expanded Interrupt Controller

그림 5-7에서 보면 확장 인터럽트는 DSP의 외부 인터럽트 중 $\overline{INT0}$ 에 cascade된 구조로 설계되어 있다. 인터럽트를 14개까지 받을 수 있으며, Priority는 INT1에서 INT14순으로 하였다. 인터럽터가 동시에 발생되면 동시에 발생한 인터럽트 중 우선권이 앞선 인터럽트부터 처리하게 된다. 인터럽트 요구 신호가 DSP로 보내어질 때 DSP의 H1 클럭(17ns)을 기준으로 3클럭 동안 LOW신호를 보내어준다. 내부적으로 default된 일정한 시간동안 인터럽트 처리를 하지 않으면 다음 우선순위의 인터럽트에 대한 요구신호를 발생시킨다. 인터럽트 요구 신호를 보내는 사이에 발생한 신호는 무시된다. 인터럽트를 수행할 때 어떤 인터럽트가 발생되었는가를 알기 위해 각 확장 인터럽트에 대한 offset vector를 DSP로 읽어들이어 DSP의 인터럽트 벡터로부터 branch 할 어드레스를 가지게된다. 확장 인터럽트의 offset vector를 처리하는 방법은 그림 5-6 으로 나타낼 수 있다.

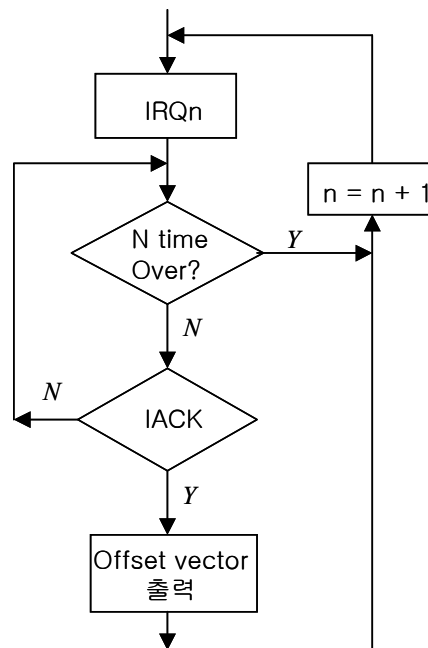


그림 5-6 인터럽트 수행 순차도

Fig 5-6 Flowchart of interrupt

DSP에서 확장 인터럽트를 처리하기 위해서는 외부 인터럽트 중 $\overline{INT0}$ 인터럽터 서비스 루틴 내에서 확장 인터럽트 controller의 offset vector를 읽어들이어 발생된 인터럽트 요구가 어떤 확장 인터럽트인가를 확인하여 그에 대한 인터럽트 서비스 루틴으로 분기하도록 프로그램 해야한다.

예제, INT0_ISR :

- ① IACK 신호 출력
- ② offset vector 읽음
- ③ INT0_vector_addr + offset vector로 하여 branch함

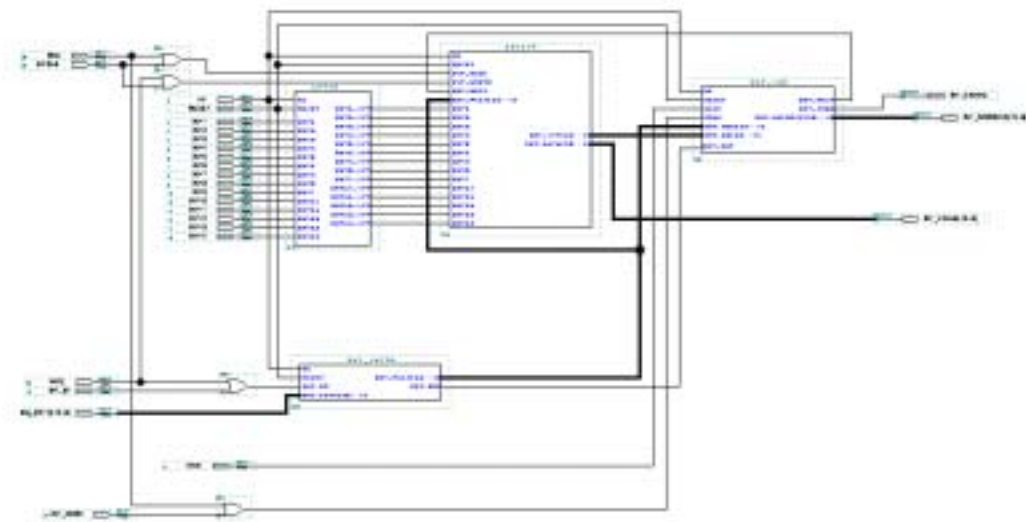


그림 5-7 Expanded interrupt controller 구성도

Fig. 5-7 Symbolic diagram of expanded interrupt controller

* INT_IR[= Interrupt Control Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	x	IMF
14	13	12	11	10	9	8	7	6	5	4	3	2	1		

15 BIT : EX_INT2

'1'이면 INT14 Enable
 '0'이면 Disable
 14 BIT : EX_INT1
 '1'이면 INT13 Enable
 '0'이면 Disable
 13 BIT : SER_INT
 '1'이면 INT12 Enable
 '0'이면 Disable
 12 BIT : PAR_INT
 '1'이면 INT11 Enable
 '0'이면 Disable
 11 BIT : AD_INT
 '1'이면 INT10 Enable
 '0'이면 Disable
 10 BIT : ENC_INT
 '1'이면 INT9 Enable
 '0'이면 Disable
 9 BIT : BOTTOM_INT
 '1'이면 INT8 Enable
 '0'이면 Disable
 8 BIT : TOP_INT
 '1'이면 INT7 Enable
 '0'이면 Disable
 7 BIT : W_INT_DOWN
 '1'이면 INT6 Enable
 '0'이면 Disable
 6 BIT : W_INT_UP

'1'이면 INT5 Enable

'0'이면 Disable

5 BIT : V_INT_DOWN

'1'이면 INT4 Enable

'0'이면 Disable

4 BIT : V_INT_UP

'1'이면 INT3 Enable

'0'이면 Disable

3 BIT : U_INT_DOWN

'1'이면 INT2 Enable

'0'이면 Disable

2 BIT : U_INT_UP

'1'이면 INT1 Enable

'0'이면 Disable

0 BIT : Interrupt Mask Flag

'1' 이면 전체 인터럽터 Disable

'0' 이면 전체 인터럽터 Enable

* INTBUF[= Interrupt Buffer Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	INT 14	INT 13	INT 12	INT 11	INT 10	INT 9	INT 8	INT 7	INT 6	INT 5	INT 4	INT 3	INT 2	INT 1

이 인터럽트는 read/write 가 가능하다.

인터럽트가 걸려서 순서적으로 인터럽트에 들어 있다가 순서적으로 인터럽트가 수행되는데 사용자가 인터럽트 버퍼의 값을 읽고 인터럽트를 기다리고 있는 인터럽트가 걸리지 않게 하거나 다른 인터럽트가 걸리게 이 인터럽트 레지스터에 적을 수 있다.

‘0’ --> 인터럽트를 발생

‘1’ --> 인터럽트를 발생시키지 않게 함

* INT_ADDR[= Interrupt Occur Address]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	x	x	INTERRUPT ADDRESS					

인터럽트 서비스 루틴으로 가기 위해 INT1~INT14중에 어떤 인터럽트가 발생했는지 알기 위해 Interrupt Occur Address의 값을 읽어서 하위 어드레스로 사용하여 branch 할 어드레스를 생성하게 된다. 이 인터럽트 어드레스를 이용하여 인터럽트 벡터에 정의된 인터럽트 서비스 루틴으로 branch 하게 된다.

INT1 --> 0X00h

INT2 --> 0X02h

INT3 --> 0X04h

INT4 --> 0X06h

INT5 --> 0X08h

INT6 --> 0X0Ah

INT7 --> 0X0Ch

INT8 --> 0X0Eh

INT9 --> 0X10h

INT10 --> 0X12h

INT11 --> 0X14h

INT12 --> 0X16h

INT13 --> 0X18h

INT14 --> 0X1Ah

NOT INTERRUPT --> 0XFFh

5.6 Wait Signal Generator

그림 5-8에서 보면접근하는 8개 주변회로들에 대해 각각 wait를 줄 수 있게 하였고, 각각 wait 크기도 다르게 줄 수 있게 하였다.

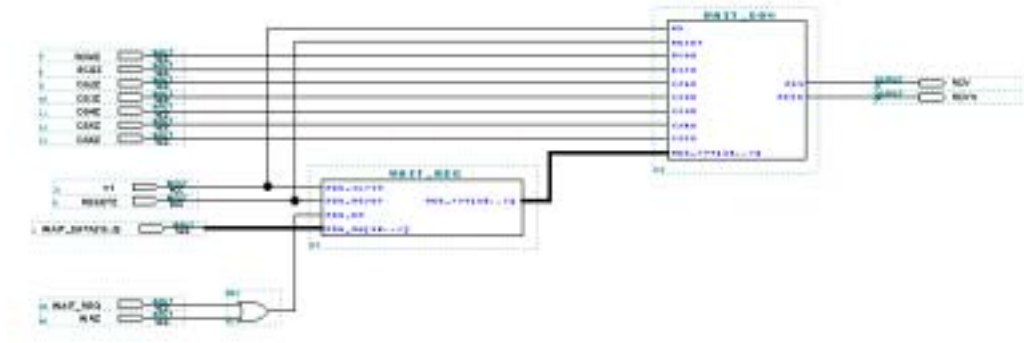


그림 5-8 Wait signal generator 구성도

Fig. 5-8 Symbolic diagram of wait signal generator

* WAIT_REG[= Wait Register]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ROMZ		ECSZ		CS2Z		CS3Z		CS4Z		CS5Z		CS6Z		CS7Z	

15 - 14 BIT : ROM을 Select 할 때 Wait를 주는 비트

13 - 12 BIT : AD 컨버터 인터페이스부를 Select 할 때 Wait를 주는 비트

11 - 10 BIT : DA 컨버터 인터페이스부를 Select 할 때 Wait를 주는 비트

9 - 8 BIT : Parallel 통신 인터페이스 부를 Select 할 때 Wait를 주는 비트

7 - 6 BIT : Serial 통신 인터페이스 부를 Select 할 때 Wait를 주는 비트

5 - 4 BIT : Wait 발생부를 Select 할 때 Wait를 주는 비트

3 - 2 BIT : PWM 발생부를 Select 할 때 Wait를 주는 비트

1 - 0 BIT : Encoder 검출부를 Select 할 때 Wait를 주는 비트

CS2Z,CS3Z,CS4Z,CS5Z,CS6Z,CS7Z는

"00" --> 1 Wait

"01" --> 2 Wait

"10" --> 4 Wait

"11" --> 7 Wait

ROMZ와 ECSZ는

"00" --> No Wait

"01" --> 1 Wait

"10" --> 2 Wait

"11" --> 7 Wait

로 H1클럭을 기준으로 Wait가 발생하면서 DSP에 있는 Ready 신호를 Low로 만들어준다.

5.7 ADC Interface Control Signal Generator

그림 5-9에서 보면하위 어드레스(A0~A5)를 이용하여 6채널 중 사용할 채널을 선택하게 하였고 ADC의 입력 클럭용으로 주파수 8MHz을 만들어서 ADC로 보내준다. AD converter가 완료된 것을 알리는 신호를 확장된 인터럽트의 입력이 되게 하였다.

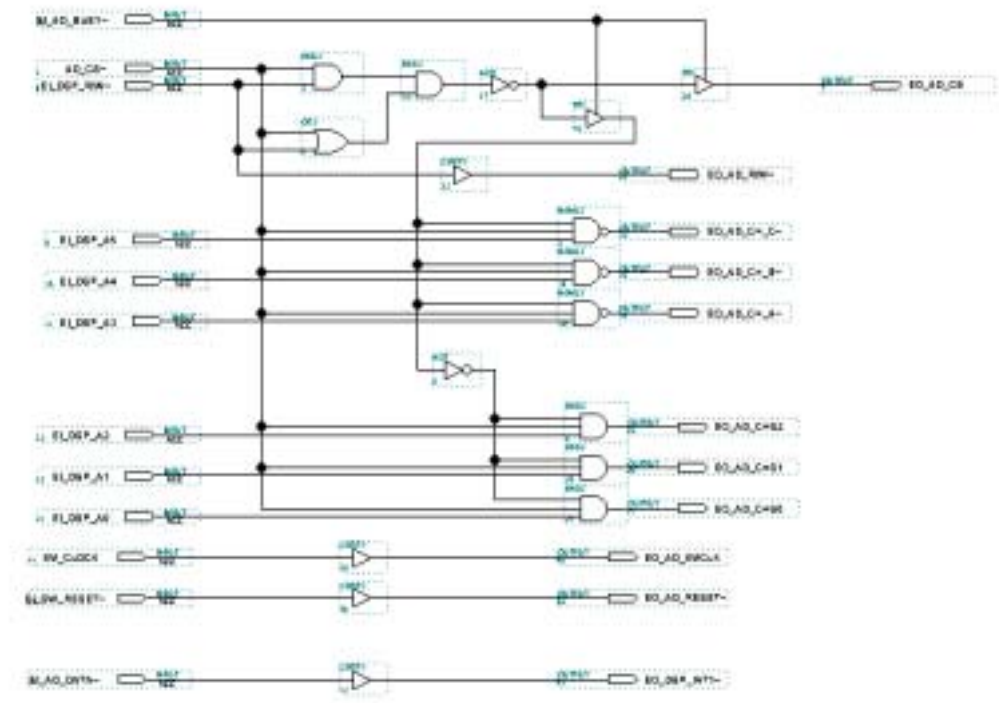


그림 5-9 ADC Interface control signal generator 구성도

Fig. 5-9 Symbolic diagram of ADC interface control signal generator

5.8 DAC Interface Control Signal Generator

그림 5-10에서 보면 하위 어드레스인 A0와 A1을 이용하여 4개의 채널중한 채널을 선택할 수 있게 하였다. 그리고 데이터를 DSP에서 DAC로 보낼 때 EO_DA_DATA~ 신호를 만들어내어 DAC의 $\overline{LDAC}$ 핀의 입력으로 준다.

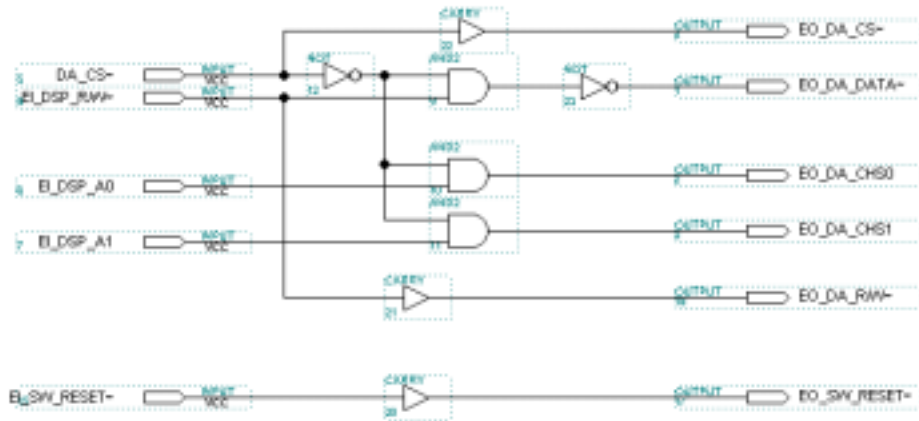


그림 5-10 DAC interface control signal generator 구성도

Fig. 5-10 Symbolic diagram of DAC interface control signal generator

5.9 Serial Communication Interface Control Signal Generator

사용자 PC(personal computer)에서 프로그램을 MAX232를 통해 모니터링을 하기 위해 serial로 통신을 할 때 serial 데이터를 8bit parallel로 바꾸어 DSP로 통신할 수 있게 하는 TL16C550 IC를 사용하였다. 그림5-11에서 보면 TL16C550의 레지스터를 선택하기 위해 하위 어드레스인 A2~A0를 사용하였고 EO_SER_ADS~를 통해 레지스터 선택 어드레스임을 알릴 수 있게 하였다. TL16C550에서 receiver error, FIFO mode에서 timed out, transmitter holding register가 비었거나, 모뎀 상태 인터럽트가 enable되었을 때 인터럽트가 발생하는 것을 확장 인터럽트의 입력으로 받게 하였다.

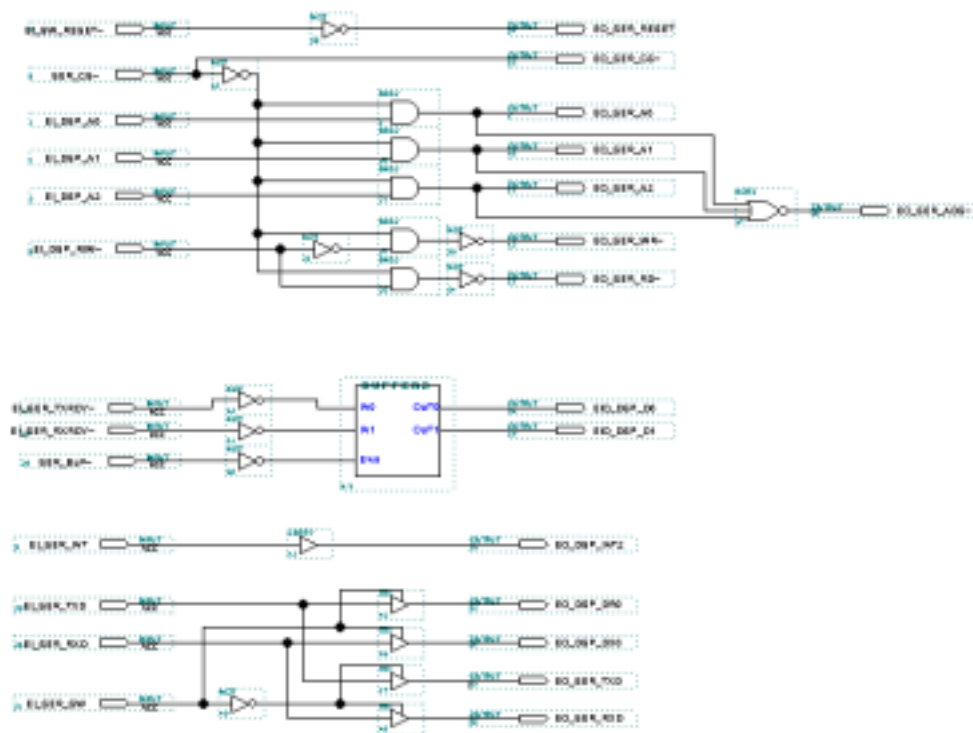


그림 5-11 Serial communication interface control signal generator 구성도

Fig. 5-11 Symbolic diagram of serial communication interface control
signal generator

5.10 Parallel Communication Interface Control Signal Generator

그림5-12에서 보면 병렬 25p 포트를 이용하여 통신을 할 때 read /write 신호를 보내게 하였고, 인터럽트를 확장 인터럽트의 입력으로 설계하였다.

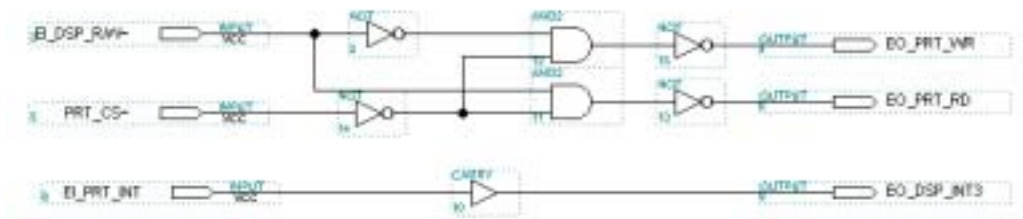


그림 5-12 Parallel communication interface control signal generator 구성도

Fig. 5-12 Symbolic diagram of parallel communication interface control signal generator

5.11 Reset Signal Generator

그림5-13에서 보면 DSP에 리셋을 보내기 위해 외부의 reset 스위치 신호와 내부의 watchdog timer에 의해 발생된 리셋 신호에 의해 DSP에 리셋 신호를 보내게 된다.



그림 5-13 Reset signal generator 구성도

Fig. 5-13 Symbolic diagram of reset signal generator

VI. 실험 결과 및 시뮬레이션

6.1 SVPWM Generator Simulation

그림 6-1에서 보면 SVPWM 출력 펄스 파형인 PWM_W/V/U_UP 과 PWM_W/V/U_DN이 각기 다른 주기의 파형으로 출력되는 것을 볼 수가 있다. PWM_U_UP과 PWM_U_DN에서 보면 delay시간만큼 지연되어서 출력되고 각 상의 compare 시점(W/U/V_INT_UP,W/U/V_INT_DN)에서, 그리고 up counter가 끝나는 시점인 top(TOP_INT)에서 인터럽트가 발생되고 있음을 알 수 있다.

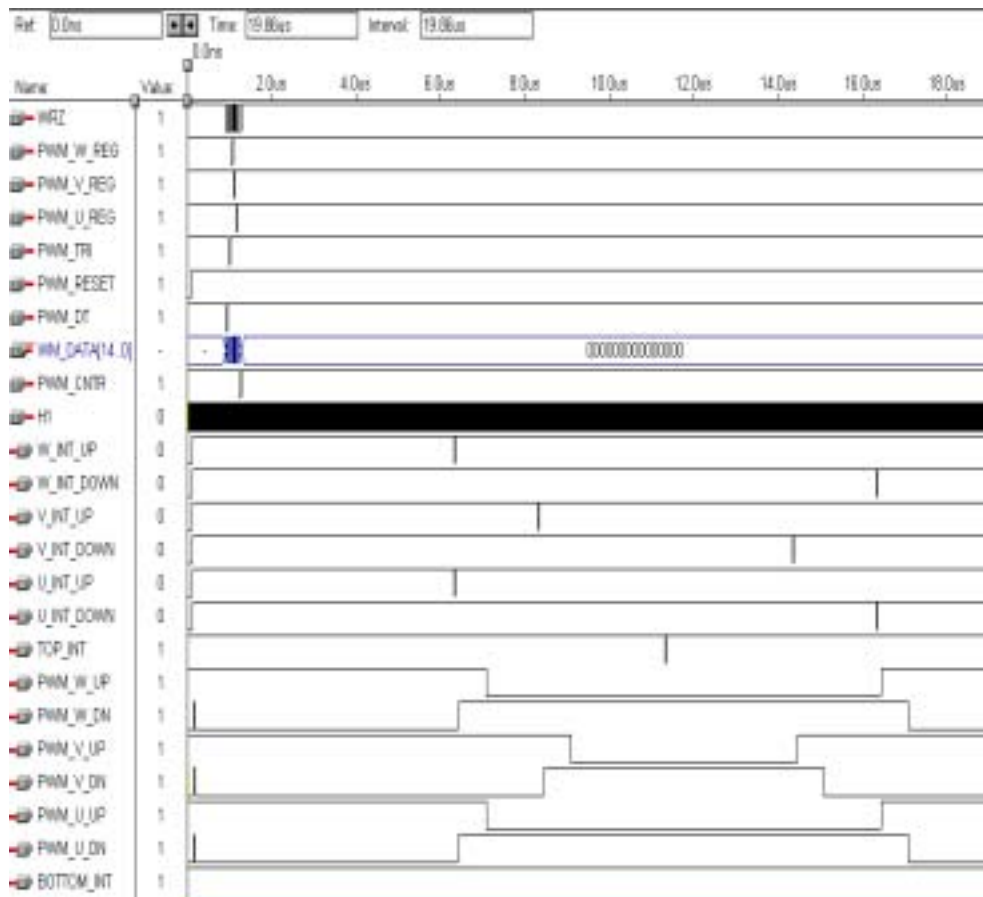


그림 6-1 SVPWM generator simulation 파형

Fig. 6-1 Wave form of SVPWM generator simulation

6.2 Encoder Signal Processing Simulation

그림 6-2를 보면 입력 엔코더 신호인 A1, B1과 C상으로 들어오는 경우와 A2, B2상만 들어오는 경우 모두 입력으로 받고 있는 가운데 각 값(ENC2_TL, ENC2_TH, ENC2_M, ENC1_TL, ENC1_TH, ENC1_O, ENC1_M)을 읽고자 하는 시점에서 각 레지스터(POS2, TL2, TL1, POS1 등)에 값을 저장시킨 후 다음에 요구할 때까지 레지스터에 값을 저장하고 있다.

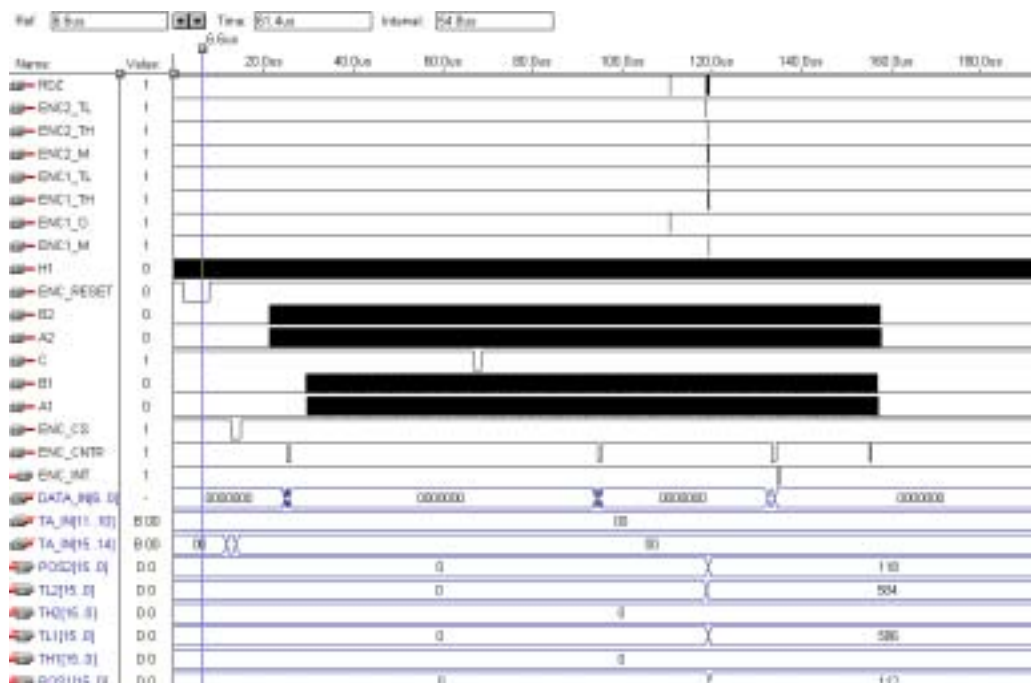


그림 6-2 Encoder signal processing simulation 파형

Fig. 6-2 Wave form of encoder signal processing simulation

6.3 Internal/External Interrupt Control Simulation

INT1~14 중에 인터럽트 입력을 받을 수 있으며 그림 6-3에서 보는 것과 같이 INT11 과 INT10이 동시에 low로 인터럽트가 발생했을 때, REG_TEMP에서 11bit, 10bit가 low가 됨으로 인터럽트가 발생된 것을 알게 된다. 동시에 두 개의 인터럽트가 들어왔기 때문에 INT10 인터럽트가 priority가 높기 때문에 OUT_TEMP에서 10bit가 high가 되어서 현재 인터럽트는 INT10임을 알려준다. 그리고 INT_SIGNAL이 high에서 low로 되면서 DSP에 인터럽트 신호를 보내게 된다.

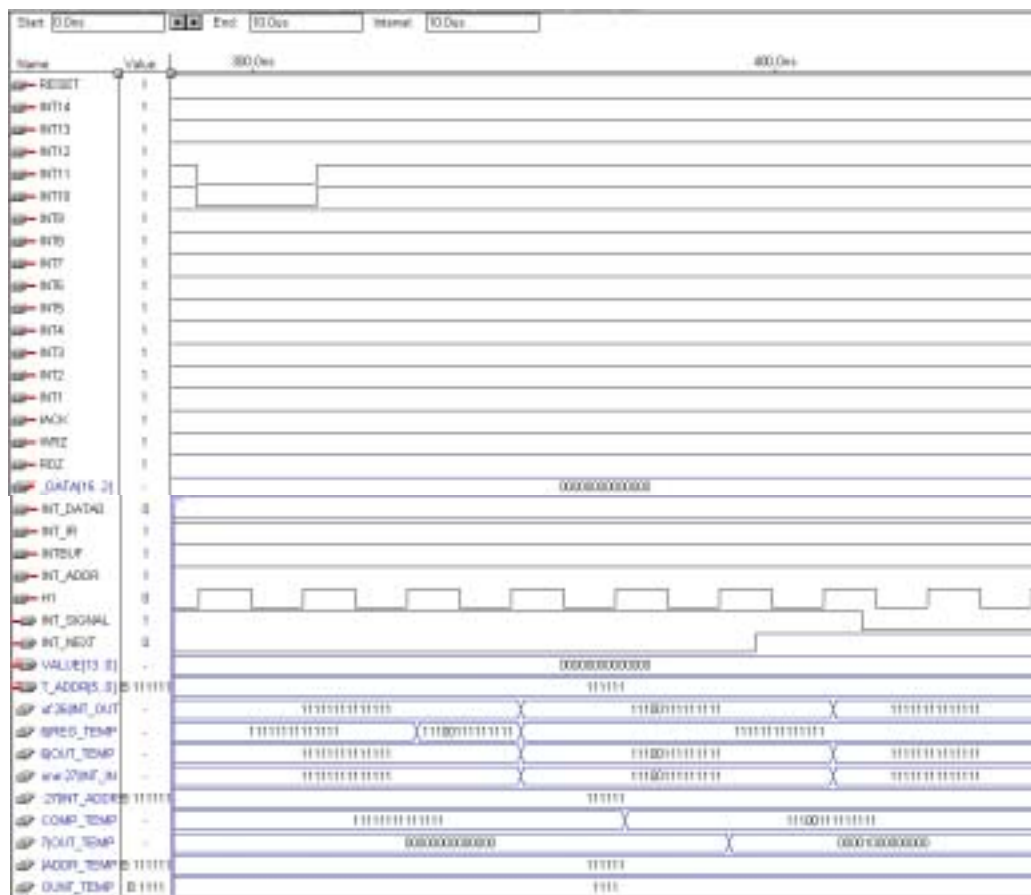


그림 6-3 Internal/external interrupt control simulation 파형

Fig. 6-3 Wave form of internal/external interrupt control simulation

6.4 Watchdog Timer Simulation

watchdog counter register에서 설정한 카운터 값 동안 카운터를 하다가 카운터를 clear를 시키지 않을 때 설정한 카운터 값이 같아지면 reset을 출력해 낸다. 그림 6-4에서 보면 COUNT_OUT이 카운터를 시작하다가 watchdog count register값과 같아지면 DOG_RESET이 high에서 low로 watchdog reset 신호를 보내게 된다.

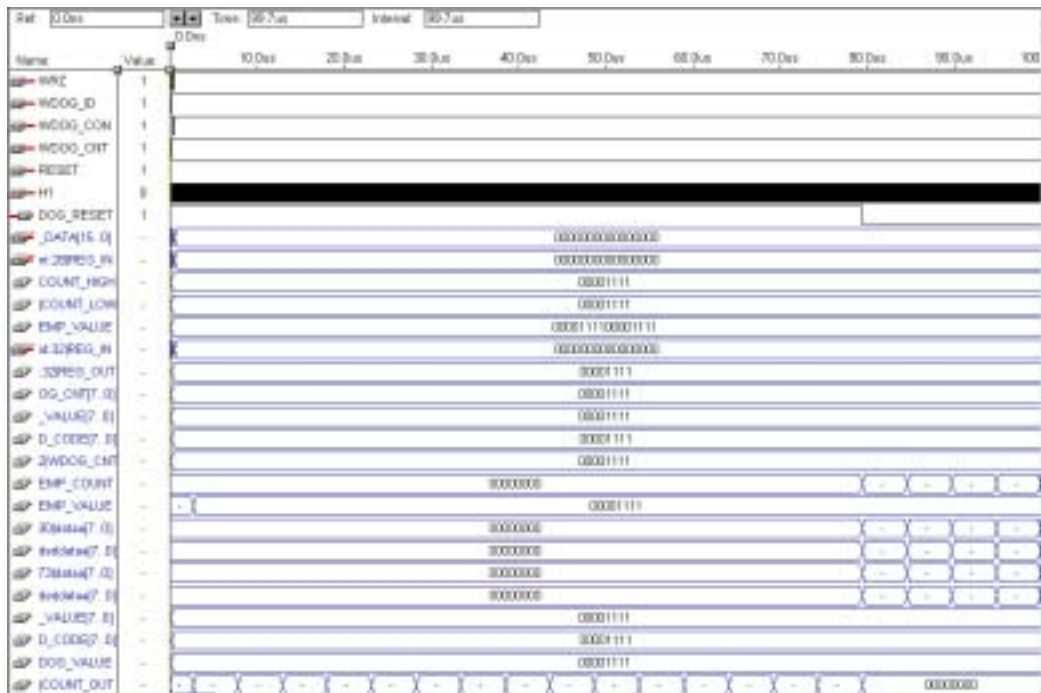


그림 6-4 Watchdog timer simulation 파형

Fig. 6-4 Wave form of watchdog timer simulation

6.5 Wait Signal Generating Simulation

그림 6-5를 보면 WAIT_REG를 통해 wait register에 값을 설정을 한 후 CS5Z로 chip select 신호를 보낼 때 RDY의 신호가 high에서 low로 2cycle동안 low active되어 DSP에 READY 신호로 보내게 되는 것을 알 수 있다.

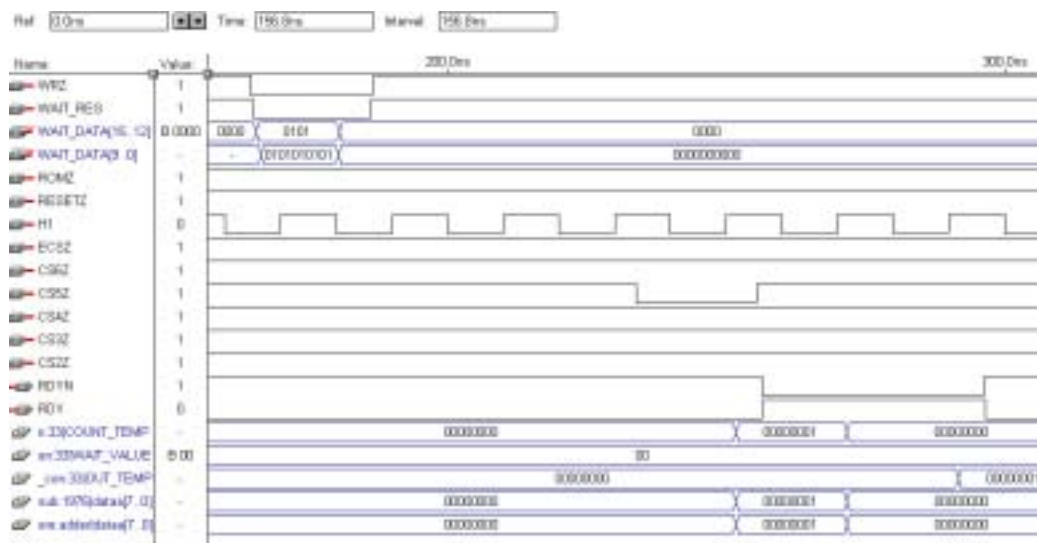


그림 6-5 Wait signal generating simulation 파형

Fig. 6-5 Wave form of wait signal generating simulation

6.6 주제어기 출력 파형

그림 6-6은 3상의 U/V/W의 PWM 위상의 출력 파형을 서로 다른 pulse 폭을 가지면서 출력 gate를 통해서 출력되는 것을 나타내고 있다. 그림 6-7은 U상의 PWM파형을 위 상과 아랫 상을 나타낸 것인데 dead time 때문에 서로 약간의 pulse폭의 차가 있는 것을 알 수 가있다.

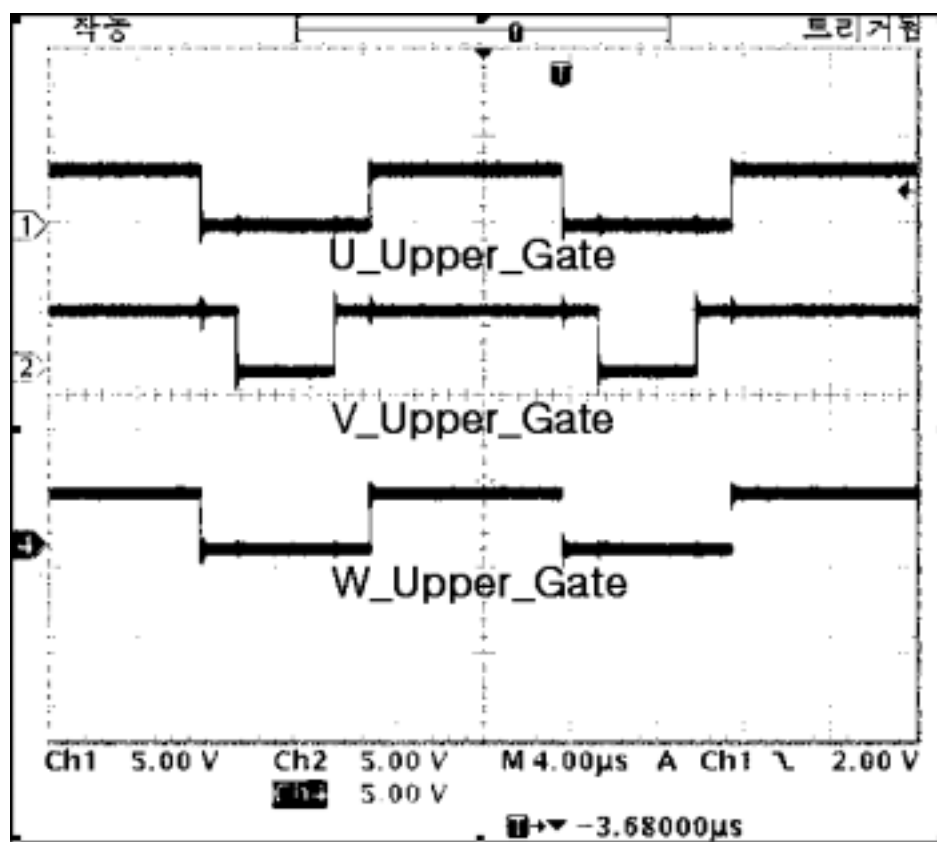


그림 6-6 U/V/W의 PWM 파형

Fig. 6-6 Wave form of U/V/W PWM

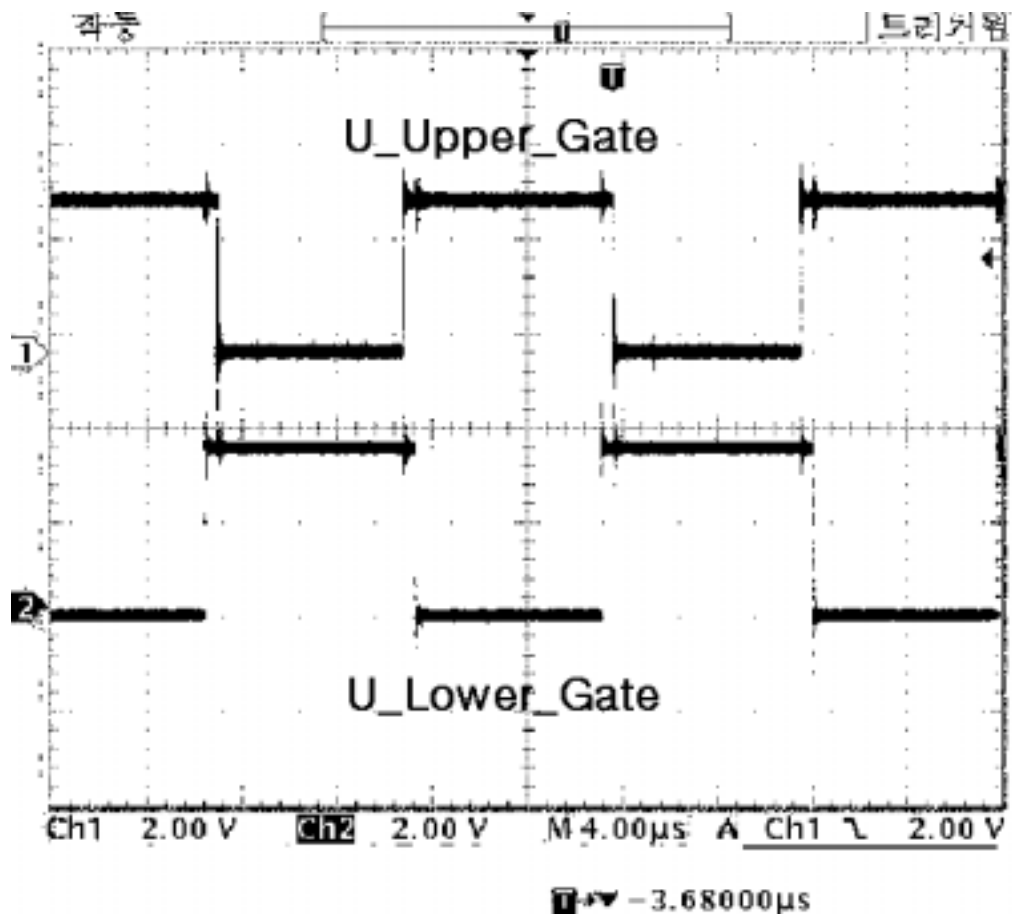


그림 6-7 U상의 dead time 파형

Fig. 6-7 Dead time wave form of U-Phase

6.7 제작된 주제어기

그림 6-7은 주제어기를 330mm x 247 mm(가로 x 세로)의 크기로 PCB 4layer로 제작된 것이다.



그림 6-7 제작된 주제어기

Fig. 6-7 Real hardware of main controller

VII. 결 론

본 논문에서는 고성능 서보 모터 제어용으로 TMS320VC33을 이용한 주제 어기를 설계 및 제작하였다. 또한 서보 모터를 제어하기 위한 주변 회로를 VHDL을 이용하여 FPGA내에 구현함으로써 단일 device화 하였다.

본 논문의 결론은 다음과 같이 요약할 수 있다.

1. 75MIPS/150MFLOPS의 고속, 저전압 기술이 적용된 3.3V device와 5V계 device의 혼합 탑재된 회로를 설계하고, 직접 PCB를 설계함으로써 noise immunity 및 회로도에 나타나지 않은 분포정수를 고려한 신뢰성있는 controller 설계에 대하여 연구하였다.
2. motor의 위치 및 속도 검출을 위한 encoder신호가 line drive형, open collector형, totempole형 및 전류구동형 중 어떤 신호로든 수신이 가능하게 하드웨어로 설계 및 구현함으로써 다양한 모터의 엔코더 신호에 대응할 수 있도록 하였다.
3. VHDL 프로그램 다운로드용 JTAG mode 포트와 PS mode 포트를 모두 사용 가능하도록 설계함으로써 개발시와 개발완료 후의 FPGA의 configuration를 효율적으로 사용 할 수 있도록 하였다.
4. DSP 프로그램을 다운로드용 JTAG 포트, serial boot loader mode 및 외부 인터럽트를 이용한 boot loader mode를 가능하게 함으로써 개발시의 on-line 테스트와 개발완료 후의 stand alone operation이 가능하도록 하였다.
5. ADC 채널 입력부에 3종류의 입력전압 범위 조정용 option selection 기능을 설치함으로써 모터의 용량에 따라 전류량의 크기가 달라질 경우에도 검지

신호의 resolution을 최적화 할 수 있도록 하였다.

6. serial port와 parallel port를 이용한 모니터링 채널을 갖추으로써 개발자 PC의 사양에 따라 실시간으로 다양한 프로그램 변수들의 변화를 모니터링 할 수 있도록 하였다.

6. VHDL을 이용하여 각 상 전압 발생을 위한 compare시 및 carrier 주파수의 주기마다 interrupt 신호 출력을 가지며 이를 enable 및 disable할 수 있는 SVPWM generator를 설계함으로써 제어 알고리즘의 시작 시점을 다양한 인터럽트 신호로 사용할 수 있으며 각 상 출력전압을 독립적으로 출력 제어 할 수 있게 하였다.

7. VHDL을 이용하여 motor 속도 검출을 위한 encoder 신호 처리기를 encoder의 신호출력 형태별 신호종류로 각각 처리 가능하도록 설계하여 제어하고자 하는 motor가 어떤 형태의 encoder 신호 출력이든 입력을 받을 수 있게 하였다.

8. VHDL을 이용하여 SVPWM 관련 interrupt, ADC의 EOC(end of conversion), 인버터로부터의 U/V/W상 fault신호, 스위치 과열 신호 및 PFC 용 switch fault 신호들을 인터럽트로 처리할 수 있도록 priority control이 가능한 expanded interrupt controller를 설계함으로써 인버터에서 발생하는 긴급한 상황들을 즉시 대응할 수 있도록 하였다.

9. 마이크로프로세서의 폭주 또는 무한루프 실행 등의 오동작 감시를 위한 watchdog timer를 VHDL을 이용하여 설계함으로써 일정한 시간 이내에 watchdog timer를 clear시키지 않으면 마이크로프로세서를 reset 시키는 supervisor기능이 가능하도록 하였다.

10. 속도가 빠른 마이크로프로세서와 상대적으로 느린 주변device들 사이의 원활한 interface를 위하여 마이크로프로세서에게 ready 신호를 줄 수 있는 가변 wait signal generator를 FPGA내에 설계함으로써 속도가 다른 device 간의 timing 문제에 대하여 효율적으로 대처할 수 있도록 하였다.

참고문헌

- [1] A narative of development for vector control, 일본전기 학회 논문지 D, 11권 1호, 94년
- [2] 윤덕용, “공간전압벡터 PWM 기법을 이용한 영구자석형 동기 전동기의 속도제어,” 단국대학교 대학원 박사학위논문,1995.
- [3] Wiley, Mohan undeland robbins, *Power Electronics*
- [4] Z80 중심의 마이크로프로세서와 그 응용, 청문각
- [5] **서보기기**, 창원대학교 국책사업단
- [6] 최종우, “D-Q 상호 간섭성을 이용한 속응 교류 제어기,”서울대학교 대학원 박사학위논문,1996.
- [7] 윤덕용, *TMS320C31 마스터*,ohm사
- [8] 박세현, *디지털 시스템 설계를 위한 VHDL 기본과 활용*, 그린
- [9] CPLD vs FPGA comparing high-capacity programmable logic, ALTERA, February 1995
- [10] TMS320C3X User's Guide, Texas Instrument, Jun 1991
- [11] TMS320VC33 Digital Signal Processor, Texas Instrument, July 2000
- [12] SN54LVTH16244A, SN74LVTH16244A 3.3-V ABT 16-BIT BUFFERS/DRIVERS WITH 3-STATE OUTPUT, Texas Instrument, October 2000
- [13] Configuration Devices for ACEX,APEX,FLEX & Mercury Devices, ALTERA, March 2001
- [14] ByteBlaster Parallel Port Download Cable, ALTERA, February 1998
- [15] ACEX Programmable Logic Device Family, ALTERA, August 2001
- [16] ADS7864, Burr-Brown, September 2000
- [17] DAC7624,DAC7625, Burr-Brown, April 2000
- [18] TL16C550,TL16C550CI Asynchronous Communication Element with

Autoflow Control, Texas Instrument, March 2001

[19] Versatile Link The Versatile Fiber Optic Connection, Agilent Technologies

[20] MAX707,MAX708 μ P Supervisory Circuit, Semiconductor, November 2001

[21] FAN1537 Dual 1A Adjustable/Fixed Low Dropout Linear Regulators, Fairchild, March 2002

[22] LM317 3-Terminal Adjustable Regulator, Texas Instrument, December 2001

[23] LM237,LM337 3-Terminal Adjustable Regulator, Texas Instrument, July 1999

Abstract

Main Controller Design applied TMS320VC33 for High Performance Servo Motor Control

Yoon Seong Eun

Dept. of Electronic Eng.

Graduate School of Changwon National University

(Advisor : Prof. Park, Seung Yub Ph.D.)

This paper is a main controller design applied TMS320VC33 of DSP in Texas Instruments to be used high performance servo motor control. Main controller circuits are composed of ADC circuits for current sensing, DAC circuits for signal checking, Control signal transmitter, receiver circuits for SVPWM signals, Encoder interface circuits for receiving each types signal of encoder, Serial and parallel communication circuits for monitoring various software variables in real time, power circuits for voltage regulation of each dc power sources, Reset and Power supervisor circuits for initializing CPU and peripherals at the first power input or power fails.

Also, design is implemented in FPGA by using VHDL. These are needed to drive motors; Watchdog timer for CPU supervisory, SVPWM signal generator for power control, signal processor applied M/T method for several signals from encoders, Expanded interrupt controller for expanded external interrupts to 14 events, Wait signal generator for interfacing a

fast CPU that is relatively slower peripherals, Control signals generator for ADC and DAC, Serial communication control signal generator for sending control signal in serial monitoring, Parallel communication control signal generator for sending control signal in parallel monitoring, Reset signal generator for sending reset signal to DSP and initializing FPGA program. These function block are simulated by MAX+plusII individually and tested totally in a EP1K100QC-208 of ACEX1K classified in ALTERA. User can choose download method by PS mode or JTAG mode.

Main controller are assembled on the 4 layer PCB.

감사의 글

30년이 지난 지금, 나의 삶의 흔적을 보면 두 발자국 아닌 누군가의 발자국이 항상 옆에 있었음을 발견합니다. 그리고 길게만 느껴졌던 것이 돌이켜 보면 너무나 짧은 시간임을 알게 하는 대학원 생활 속에서도 많은 사람들의 발자국이 있었습니다. 그런 많은 분들에게 고마움과 감사의 마음을 이 지면을 통해서 대신코자 합니다. 저에게 대학원을 시작할 수 있도록 기회를 주시고 많은 관심과 사랑으로 지도해주신 박승엽 교수님, 언제나 넉넉한 마음으로 대해주신 최평석 교수님, 학문적으로 지식의 기반을 쌓게 해주신 최중경 교수님, 하판봉 교수님, 은재정 교수님, 고봉진 교수님, 박무훈 교수님, 그리고 김영희 교수님께 감사의 마음을 전합니다.

언제나 한없이 높아만 보이는데도 인생의 선배로, 학문적 선배로 저의 곁에서 아낌없는 사랑을 주신 박성수 부장님께 미흡하나마 감사의 말씀을 드립니다. 자동제어 실험실로 한 식구가 되어 많은 도움을 주며 같이 고민해 주었던 박사 과정의 삼영이, 민식이, 황정원씨, 졸업한 재녕이 모두에게 감사하며 그리고 새내기이시지만 저에게는 큰형으로써 많은 관심과 도움을 주신 넷크레이브의 백승택 사장님께 감사 드립니다.

언제나 옆에서 나의 일같이 도와준 광전자 실험실의 남영이, 논문을 쓸 수 있도록 많은 도움을 준 주홍이, 한동안 같이 공부 했던 산업전자제어연구실의 현아씨, 효원이, 홍수, 진애등과 광전자 실험실의 영환이등 모든 분께 감사 드립니다.

SHINE으로 하나가 되어 많은 고생과 즐거움을 같이 하면서 큰 꿈을 이루기 위해 노력했던 날들이 지금은 아름다운 추억이 되게 해준 김해룡, 최일출, 박용하, 신용한에게 이 지면으로 감사 드립니다. 지금은 또 다른 곳에서 꿈을 펼치지만 언제나 건강하고 하는 일들이 잘 이루어지기를 바라며 기도합니다. 힘들 때 나의 옆에서 힘이 되어준 영우, 영원한 나의 조언자 응림이, 판식이,

성대등 모든 친구들에게 감사드리며, 나에게 기쁨이 되어주는 기교, 상호, 민음의 친구인 경민이에게 감사하며 언제나 나를 지켜보고 있는 대건이, 뒤에서 묵묵히 따라주는 종월이, 군에 있는 막내 정보, 그리고 차분한 영진이에게 감사의 말을 전합니다. 언제나 넉넉한 마음으로 베풀어주는 지웅이, 재수씨인 은진씨 그리고 보배인 아들 강혁이 모두의 건강을 기원하며 감사의 말을 전합니다. 나에게 많은 인생의 일깨움을 주는 명남이 누나에게도 감사의 말을 전합니다.

항시 저를 위해 밤낮으로 기도하시는 외할머니께 감사드리며 언제나 건강하시기를 기도합니다. 옆에서 지켜보고 도와주시는 정희, 정선 이모에게 감사하며 하시는 모든 일들이 잘 되시기를 기도합니다. 많은 것을 도와주지도 못하고 괴롭히기만 하는 동생을 언제나 든직한 기둥으로서 지켜보고 있는 형, 그리고 형수에게 감사 드립니다. 그리고 우리 집안의 보배들인 회원이, 신원이, 은원이 모두 건강하게 자라주어서 고맙고 더욱 건강하게 성장하기를 기도합니다. 언젠가는 웃으면서 뒤돌아 볼 일이 있을 동생에게 감사의 말을 전합니다. 늦게나마 공부하는 저를 지켜보며 마음 고생하시고 기도하신 부모님께 이 논문을 바칩니다. 많은 것을 해드리지 못한 것 죄송하며 언제나 건강하시기를 기도합니다.

새벽에 자동제어연구실 창밖을 내다 보며