

FPGA 기반 서보 모터 제어 IP 코어 개발에 관한 연구

A Study on the development of Servo Motor Control IP Core based on FPGA

문용선 · 노상현 · 조광훈* · 이영필** · 배영철***

Yongseon Moon, Sanghyun Roh, Kwanghun Jo*, Youngpil Lee** and Youngchul Bae***

순천대학교 공과대학 정보통신공학부

*메트로 시스템엔지니어링,** 레드원테크놀러지(주),

***전남대학교 공과대학 전기·전자통신·컴퓨터공학부

요 약

지금까지 산업용 시스템, 사무용 기기, 가전제품, 로봇틱스 분야 등 다양한 영역에 걸쳐 많이 적용이 되고 있는 서보모터 제어 장치의 구현은 MCU 및 DSC(또는 DSP)을 이용하였다. 그러나, MCU 및 DSC는 모터의 제어 효율을 극대화할 수 없으며 유연성이 부족하다는 문제점을 가지고 있다. 본 논문에서는 이러한 문제점을 해결하고자 모터 제어 효율과 제어의 유연성을 최대한 발휘할 수 있는 구조인 FPGA 기반의 서보 모터 제어용 IP 개발을 위한 설계 방법을 제시하고 이를 구현하였다.

Abstract

Until now, the implementation of servo motor control units which is being applied in various industrial ares such as industrial system, office equipment, home appliance and robotics, used the MCU and DSC(or DSP). However, MCU and DSC have limitations of not being able to maximize control efficiency of the motor and the flexibility. Thus, in this paper, we propose and implemented the designing method for development of servo motor control IP based on FPGA which have a structure to make the best motor control efficiency and flexibility of control,

Key Words : FPGA, 서보 모터 제어, IP 코어 개발

1. 서 론

모터는 전기에너지를 기계에너지로 바꾸는 기계를 말한다. 오늘날 모터는 산업용 시스템, 사무용 기기, 가전제품, 로봇틱스 분야 등 다양한 영역에 걸쳐 가장 많이 사용되고 있다. 모터는 모든 산업의 핵심이며 심장과도 같은 중요한 역할을 한다. 이러한 모터를 정밀하게 구동 및 제어하기 위해서는 별도의 드라이버 장치가 필요하며 드라이버 장치 구현을 위한 핵심 기술이 바로 서보제어 기술이다.

현재 국내에서 일부 기업을 제외한 대부분의 서보 모터 및 드라이버 제작 업체들은 서보 모터 구동 및 제어를 수행하는 핵심 제어 칩을 독자적으로 생산하는 것이 아니라 외산 모터 제어 칩을 사용하거나 범용의 MCU 및 DSC 등을 적용하여 독자적인 보드를 구현하는 형태를 가진다. 이러한 구조는 모터 제어의 효율을 극대화 할 수 없으며, 칩 내부에 고정적으로 임베디드화된 모터제어 블록들을 그대로

로 사용하므로 다양하고 유연한 모터제어가 어렵다는 문제를 가지고 있다. 또한 칩 변경과 같은 예기치 못한 사태가 발생하였을 경우 칩 변경 및 제어 로직의 포팅 과정에 상당한 시간이 소요되는 등 유연한 대처가 어렵다는 문제가 존재한다. 이에 최근에는 모터제어 전용 칩을 생산하는 전문 업체가 아니라도 FPGA(Field Programmable Gate Array) 및 ASIC(Application Specification Integrated Circuit) 기술을 이용하여 모터 제어를 위한 하드웨어 디자인을 직접 설계하여 원하는 모터 제어용 전용 칩을 만드는 방법들이 많이 적용이 되고 있다.

FPGA는 현장에서 빠른 시간 내에 직접 설계하여 구현 가능한 설계 식으로 저렴한 비용과 시간으로 원하는 칩을 만들 수 있다는 장점이 있으며, ASIC은 특정 용도로 설계되어 제작된 칩으로서 어느 정도 대량 생산을 고려한 형태를 말하는 것으로 샘플만 제작하는데도 많은 돈이 드는 단점과 불필요한 요소가 모두 제거된 형태로 설계를 해야 하기에 설계와 검증이 많은 노력이 필요한 설계 방식이다. 두 기술 모두 하나의 칩을 만드는 기술로서 처음에는 ASIC을 위한 과정으로 FPGA 를 사용하였으나 요즘은 소량으로 최단기간에 제품을 만들 때 FPGA 의 장점이 우수하기에 양산용으로도 많이 사용하고 있다[1].

본 연구에서는 기존의 고정적으로 적용되던 MCU 및 DSC(또는 DSP) 기반의 서보 모터 드라이버 시스템의 구조에서 모터 제어 효율과 제어의 유연성을 최대한 발휘할 수 있는 구조인 FPGA 기반의 서보 모터 제어용 IP를 개발에 대한 설계 및 구현 방법을 제시하고, 이를 모터 제어에 적

접수일자 : 2010년 4월 3일

완료일자 : 2010년 7월 2일

교신저자 : 배영철

본 논문은 본 학회 2010년도 춘계학술대회에서 선정된 우수논문입니다

감사의 글 : 본 논문은 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임

용한 결과를 기술하였다.

2. FPGA 기반 서보 모터 제어

2.1. 서보모터 제어 구현 기법

현재 서보 모터 구동 및 제어를 위한 드라이버를 구현하는 기술은 다양한 방법들이 존재하며 크게 마이크로 컨트롤러(MCU/DSC) + 모터 제어 칩을 이용하는 일반적인 방법과 FPGA를 이용하는 방법으로 구분된다[1].

이들 방법들은 각각 장단점을 가지고 있으나 성능 및 제어의 관점에서 보면 FPGA 기반 서보 제어 구조가 우수하다. 그 사례로, 필터를 구현할 때 필터의 차수가 증가할수록 MCU 및 DSP 상에서의 처리 사이클은 수십~수백 사이클로 증가할 수 있으나 FPGA 상에서는 차수에 따른 블록을 차수만큼 증가시키므로서 실제 처리 연산은 단 1사이클 만에 처리가 가능하게 된다.

서보 모터와 밀접한 PWM 출력 부분에 있어서도 그림 1과 같이 FPGA 기반 모터 PWM 방법이 모터의 성능에 크게 영향을 미치는 고조파 성분에 비교적 강인한 구조를 가지고 있다[1].

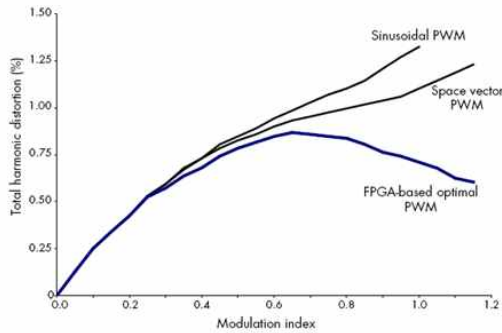


그림 1. 모터 PWM 출력 방식에 의한 고조파
Fig. 1. Harmonic by motor PWM output method

2.2. FPGA 기반 서보 모터 제어 IP 구조

FPGA를 이용한 서보 모터 제어 IP 설계의 구조는 FPGA 내부의 임베디드 프로세서를 이용한 소프트웨어 설계 기술과 HDL 구조를 이용한 하드웨어 설계 기술을 융합한 설계 방식과 단순 HDL 구조만을 이용하는 방법으로 크게 구분된다.

본 논문에서는 고속 동작이 가능하며 개발 시간이 비교적 빠른 방법인 HDL 구조만을 이용하여 서보모터 제어 IP 개발을 수행한다.

본 논문에서는 그림 2와 같은 FPGA 기반 서보 모터 제어 IP의 개념적인 구조를 설계하였으며[2] 이는 크게 인터페이스, 서보 제어, 메인 로직으로 구성한다.

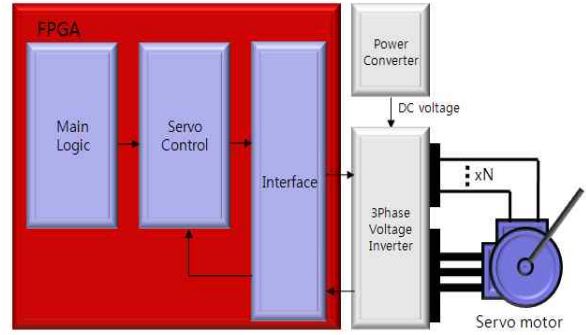


그림 2. FPGA 기반 서보 모터 제어 구조
Fig. 2. Structure of servo motor control based on FPGA

1) 인터페이스

인터페이스는 실제 하드웨어 인버터 드라이버 부분과 FPGA 칩 간의 인터페이스를 담당하는 부분으로서 모터 제어 IP 블록과 실제 하드웨어 부분은 분리해 주는 역할을 담당한다. 이는 앞으로 하드웨어 인터페이스가 변경이 되더라도 모터 제어 IP 블록을 변경하지 않고 Interface 블록만의 수정 및 변경만을 수행함으로써 손쉽게 하드웨어 인터페이스 구조를 변경할 수 있다는 장점을 가지고 있다.

2) 서보 제어

서보 제어 IP 부분에서는 모터의 전류, 속도, 위치제어 필요한 제어 로직과 모터 구동을 위한 정류 기능이 구현되는 부분으로서 모터의 특성에 맞는 다양한 제어기 및 정류 방식의 구현이 가능하다.

3) 메인 로직

설계된 모터 제어 IP 및 인터페이스 IP 블록에 대한 파라미터 설정 및 지령 값을 발생하는 부분으로서 모터 구동 명령 및 외부 인터페이스를 추가적으로 설계할 수 있는 부분이다.

3. FPGA 기반 서보 제어 IP 설계

3.1. 정류 모드(Commutation Mode) 설계

서보 모터의 기본적인 회전을 수행하는 정류 모드는 모든 모터 제어를 위해 가장 먼저 설계 되는 부분으로서 본 연구에서는 그림3과 같이 크게 4개의 IP 컴포넌트 블록 형태로 정류 모드를 설계하였다[2-5]

1) PWM 발생기 블록(generator block)

그림 4와 같은 PWM 발생기는 입력된 진폭(Pwm Duty)에 따라 PWM 신호를 출력하는 컴포넌트 블록으로서 진폭에 따라 출력된 PWM 신호는 정류 컴포넌트 블록을 통하여 최종적인 3상 HI, LOW PWM 신호로서 생성이 된다. 일반적으로 입력된 진폭에 따라 PWM 신호를 생성하는 방법에는 Edge Aligned PWM 방식과 Center Aligned PWM 방식 이렇게 2가지 나누어진다.

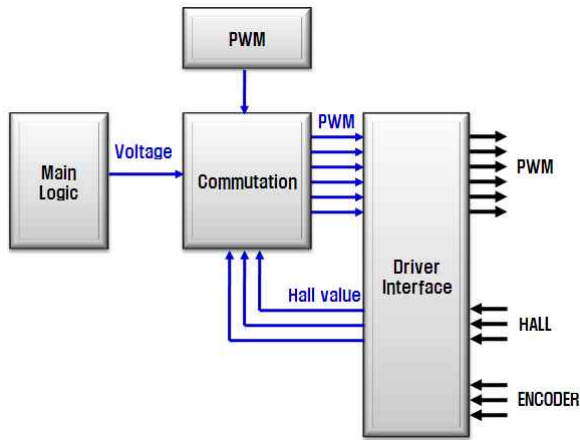


그림 3. Commutation mode IP 설계 구조
Fig. 3. Commutation mode IP design structure

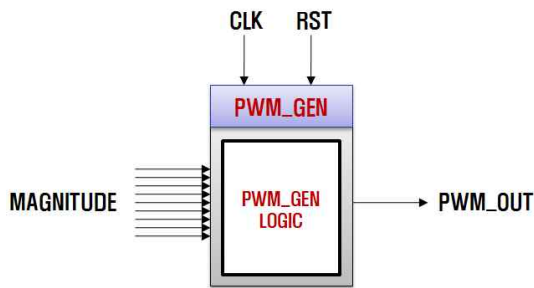


그림 4. PWM 발생기 블록
Fig. 4. PWM Generator block

본 논문에서는 이 중에서 대칭형 PWM 방식으로 서보 제어에 많이 사용이 되는 Center Aligned PWM 방식을 적용하였다. 실제 구현된 Center Aligned PWM 방식의 구조는 그림5와 같다.

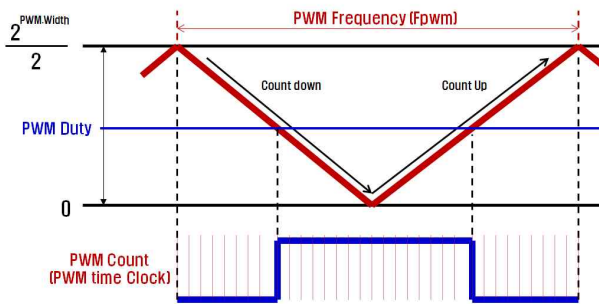


그림 5. Center Aligned PWM 구조
Fig. 5. Center Aligned PWM structure

Center Aligned PWM 방식의 카운팅 구조는 Down Up 카운트를 적용하였다. PWM 유닛 카운트는 최대 진폭 카운트에서 0으로 감소하였다가 다시 최대 진폭으로 증가하는 구조를 가지며, 지령 진폭 값인 PWM Duty 이 입력 될 경우 해당 PWM 유닛 카운트가 지령 진폭 보다 작은 구간에서는 PWM 신호가 하이 상태를 유지하고 지령 진폭 보다 큰 구간에서는 Low 상태를 유지하는 구조를 가지도록 설계하였다.

2) 정류 블록(commutation block)

그림 6과 같은 정류 블록에서는 서보 모터의 구동을 위해 자극의 위치 상태에 따른 정류를 수행하는 부분으로서 입력된 서보 모터의 홀 신호와 PWM Generator에서 생성된 PWM 신호를 융합하여 모터의 구동에 필요한 3상 PWM HI, LOW 신호를 생성하여 출력한다.

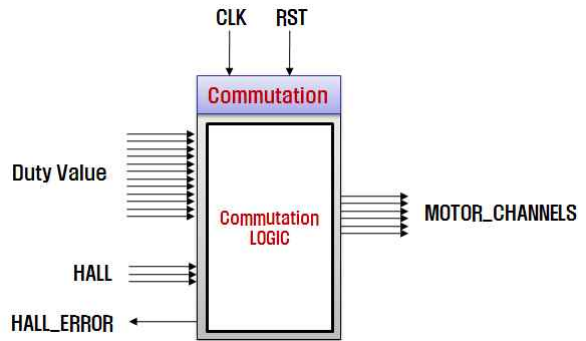


그림 6. Commutation 블록
Fig. 6. Commutation block

서보 모터의 정류를 위해서는 모터의 통전 방식 및 이에 따른 홀 섹터 간의 관계에 대한 기본적인 이해가 필요하다. 그림 7는 서보 모터의 구동을 위해 모터 내부에 연결된 3상 권선의 통전 순서 및 이에 따른 홀섹터 위치 및 선간 출력 파형의 관계를 나타낸다.

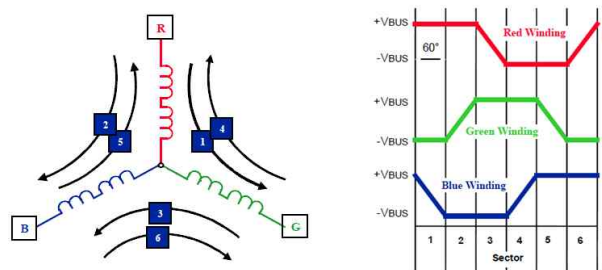


그림 7. 서보 모터 구동에 따른 홀 섹터 선간 출력 파형 (사다리꼴 구조)

Fig. 7. Output wave between hall sector line by servo motor driving(trapezoid)

그림 8은 그림 7의 통전 관계를 기반으로 실제 측정된 홀 센서의 값에 따른 홀 섹터의 위치와 이에 따른 PWM HI, LOW 신호의 출력 관계를 나타내고 있다.

본 논문에서 설계된 정류 블록은 그림 8과 같은 정류 규칙을 기반으로 설계가 되었다.

State	Hall Code	High Select	Low Select
0	001	0	2
1	011	1	2
2	010	1	0
3	110	2	0
4	100	2	1
5	101	0	1
6	others	3	3

그림 8. 홀 섹터에 따른 3상 PWM 출력 결과

Fig. 8. Result of 3-phase PWM output by hall sector

3) 구동기 인터페이스 블록(driver Interface block)

구동기 인터페이스 블록은 실제 설계된 모터 제어 IP 블록들과 하드웨어 핀과의 인터페이스를 위한 블록으로서 설계 목적은 모터 제어 IP 블록들을 하드웨어와 분리함으로써 FPGA 칩 또는 서보 모터 드라이버의 하드웨어 인터페이스가 변경이 되더라도 모터 제어 IP 블록들은 변경하지 않고 구동기 인터페이스 블록만 변경함으로써 설계된 모터 제어 IP 블록들의 재사용성을 높이기 위함이다. 실제 설계된 구동기 인터페이스 블록의 입출력 인터페이스를 그림 9와 같다.

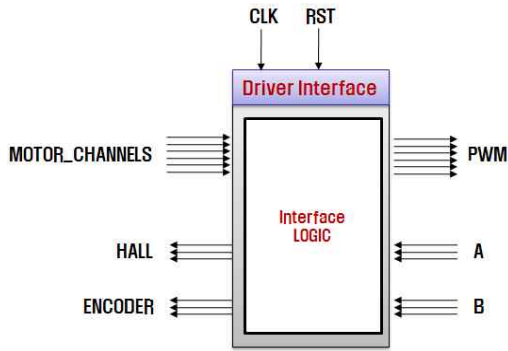


그림 9. 구동기 인터페이스 블록
Fig. 9. Driver Interface block

4) 메인 로직 블록(main logic block)

메인 로직 블록은 서보 모터 구동 및 파라미터를 설정하는 블록으로서 모터 구동을 위한 듀티 명령(duty command) 및 속도(speed command) 등 같은 설정 및 지령 인터페이스가 구현이 된다.

3.2 속도 제어 모드(speed control mode)

속도제어 모드는 서보모터의 기본적인 정류 모두 앞단에 속도제어기를 추가함으로써 서보모터가 단순 입력 진폭이 비례한 구동을 하기보다는 원하는 지령 속도를 추종하여 구동하도록 하는 동작 제어 모드이다. 속도 모드 경우 기존의 정류 모드 앞단에 속도 제어를 위한 속도 PI 제어기와 속도 측정을 위한 블록이 추가된다. 세부적인 구조는 그림 11과 같다[2-5]

1) 속도 측정 블록(speed measurement block)

속도 측정 블록은 입력된 홀 센서 또는 엔코더 신호를 이용하여 모터 구동 간 실제 모터의 회전 속도를 계산하여 속도 제어기로 전달하는 기능을 담당한다.

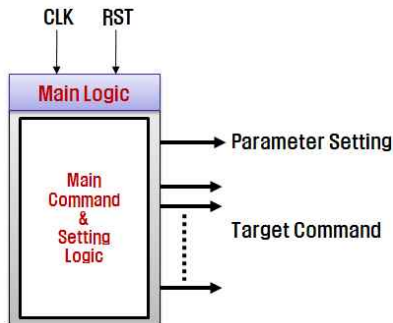


그림 10. 메인 로직 블록
Fig. 10. Main Logic block

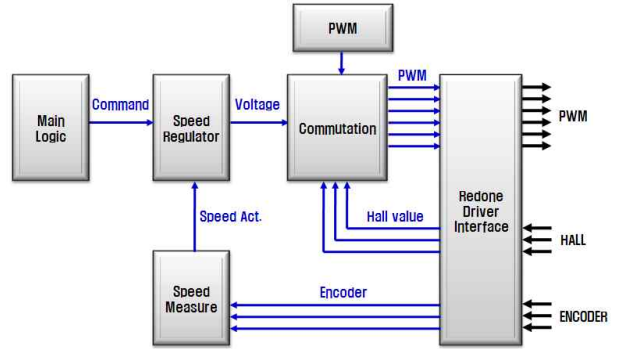


그림 11. 속도제어 모드 IP 설계 구조

Fig. 11. Speed control mode IP design structure

실제 설계된 속도제어기 블록은 그림 12와 같으며, 본 논문에서는 속도 측정 입력으로서 고정도 속도 측정이 가능한 엔코더를 적용하였다.

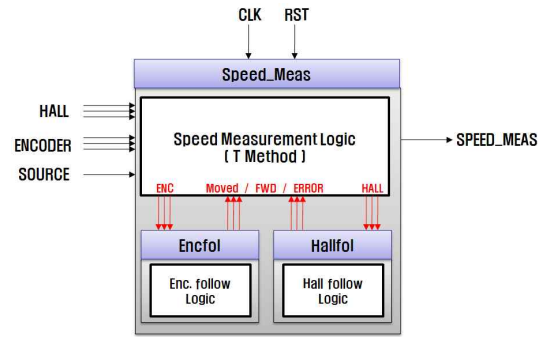


그림 12. 속도 측정 블록

Fig. 12. Speed Measurement block

일반적으로 모터 속도 측정을 위해 사용하는 방법은 모터 구동 간 측정되는 단위 펄스의 이동 시간을 계산하는 T 방식과 단위 시간당 측정되는 펄스의 수를 측정하는 M 방식 사용이 된다.

본 논문에서는 M 방식에 비해 비교적 정확한 속도 측정이 가능한 T 방식을 사용하여 서보 모터의 속도를 측정하였다. T 방식의 속도 측정 구조는 그림 13과 같다. T 방식의 속도 측정은 그림 13과 같이 모터 구동 간에 출력되는 펄스의 시간을 측정하고 해당 펄스에 비례한 모터의 이동거리 및 소요 시간을 계산하여 전체 모터의 RPS 및 RPM을 계산하는 방법이다.

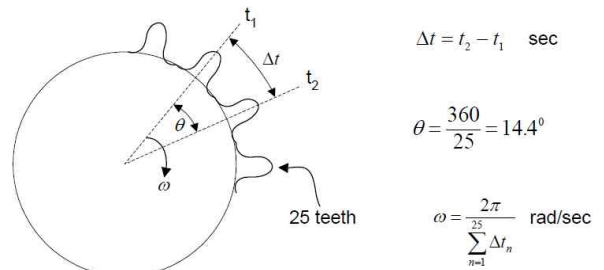


그림 13. T 방식의 모터 속도 측정 구조

Fig. 13. Motor speed measurement structure of T method

2) 속도 제어기 블록(speed controller block)

속도 제어기 블록은 지령된 속도와 측정된 속도의 차를 비교하여 오차에 대한 증분, 적분을 통한 제어 출력을 생성하는 부분으로서 기본적인 입출력 구조는 그림 14와 같다.

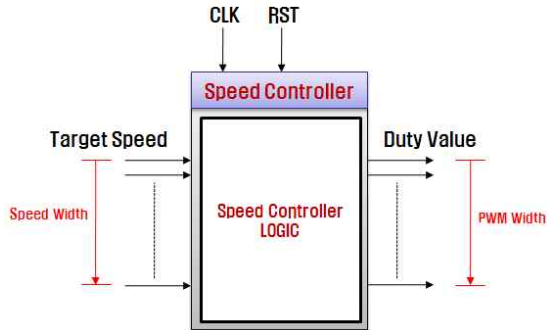


그림 14. 속도 제어기블록
Fig. 14. Speed controller block

서보모터의 속도 제어기는 모터 제어의 일반적인 형태의 PI 제어기에 가변 부하에 능동적인 대응을 위한 Speed Feed forward를 추가한 Speed PI 제어기를 설계하였다 [2-5].

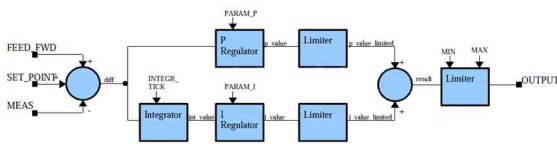


그림 15. 속도 제어기 블록 구조
Fig. 15. Structure of speed controller block

4. FPGA 기반 서보 제어 IP 구현 및 실험

본 논문에서는 서보 모터의 제어 모드 중 속도제어 모드에 대한 IP를 구현하였으며 FPGA 개발 환경은 ALTERA CycloneIII FPGA 칩인 3P3C40F484C6N을 적용하였으며, 개발 툴은 ALTERA QUARTUS II 9.1을 사용하였다.

4.1 하드웨어 구조 설계

하드웨어 구조는 정류기, 속도 제어기, PLL 컴포넌트 블록을 메인 블록으로 적용하며, 최상위 계층인 Top Level Entity 블록에서 해당 컴포넌트 블록 간의 연결 인터페이스를 구성하는 형태로 설계를 하였다. 또한 각각의 컴포넌트 블록들은 다시 그림 16, 17과 같이 하위의 세부적인 컴포넌트 블록을 포함하고 있다[6].

그림 16, 17의 설계 구조를 기반으로 실제 ALTERA QUARTUSII 툴 상에 설계된 서보 모터의 속도 제어 하드웨어 디자인 파일 및 계층 구조는 그림 18과 같다.

그림 19와 같이 서보 모터 속도제어기 구현을 위한 디자인 파일의 설계가 완료되면 그림 20과 같이 블록 스키매틱 파일 (block schematic file) 형태로 작성된 최상위 디자인 파일인 Top Level Entity 파일 상에서 그림과 같이 하위 컴포넌트들 간의 인터페이스 및 외부 하드웨어 핀 간의 입출력 포트를 연결해 준다.

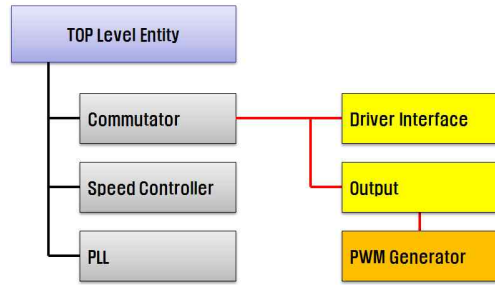


그림 16. 서보 모터 속도 제어 블록 구조 1 (정류기 블록 세부 구조)

Fig. 16. Speed control block structure 1 of servo motor (commutator block detail structure)

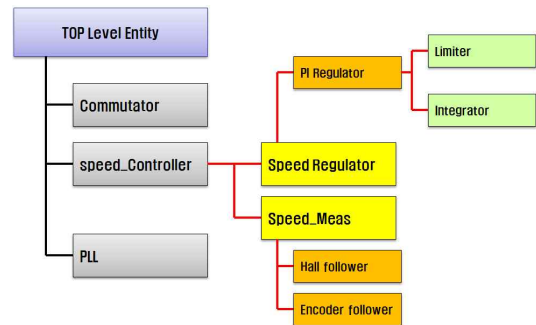


그림 17. 서보 모터 속도 제어 블록 구조 2 (속도 제어기 블록 세부 구조)

Fig. 17. Speed control block structure 2 of servo motor (speed controller block detail structure)

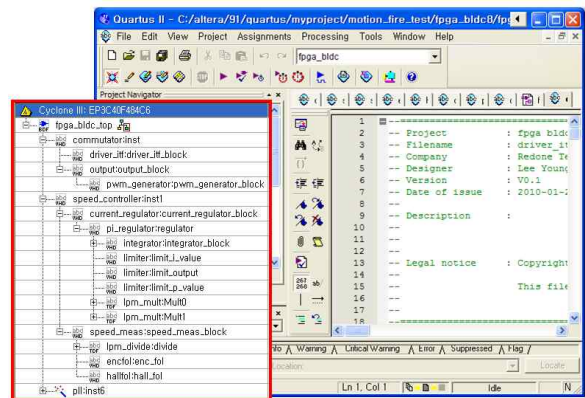


그림 18. 서보 모터 속도제어기 설계 파일
Fig. 18. Design file of servo motor speed controller

최종적으로 핀 플래닝 과정을 통하여 실제 FPGA 와 연결된 드라이버 보드의 입출력 핀을 설계한 입출력 포트 상에 연결한 후 최종 설계를 컴파일하면 서보 모터 제어 IP 구현을 위한 FPGA 하드웨어 설계가 완료가 된다[3].

최종적으로 FPGA 내에 설계된 서보 모터 제어 IP의 구조는 그림 19와 같다.



이영필

2006년 2월 : 순천대학교 전자공학과
(공학사)
2008년 2월 : 순천대학교 전자공학과
(공학석사)
2008년 ~ 현재 : 레드윈테크놀로지(주)
연구원

관심분야 : 로봇 및 모터 제어, 산업통신망



배영철

1984년 : 광운대 전기공학과 졸업.
1986년 : 동 대학원 석사
1997년 : 동 대학원 박사
1986년~1991년 : 한국전력공사
1991년~1997년 : 한국과학기술정보연구원
1997년~2006년 : 여수대학교 교수
2001년~2002년 : Brigham Young
University 방문교수
2006년~현재 : 전남대학교 공학대학 전기·전자통신·
컴퓨터 공학부 교수

관심분야 : 로봇 제어, 카오스 제어 및 동기화, 비선형 제어,
산업통신망 제어