

DC 모터 제어용 SoC 설계

Design SoC for DC motor control

윤기돈, 오성남, 김갑일, 손영익
(Ki Don Yoon, Sung Nam Oh, Kab Il Kim, Young Ik Son)

명지대학교 전기공학과

요약문 : 본 논문에서는 ARM922T Core와 주변장치를 설계할 수 있는 100만 게이트의 FPGA를 내장한 알테라(Altera)사의 엑스칼리버(Excalibur)를 이용하여 DC모터 제어용 SoC를 설계하였다. SoC란 System on Chip의 약자로 하나의 칩 안에 프로세서와 다양한 목적의 주변장치들을 집적하는 것을 말한다. 모터를 구동하기 위한 PWM신호 생성기를 하드웨어 설계언어(Hardware Description Language)로 구현하고 시뮬레이션을 통해 설계모듈을 검증하였다. 이렇게 검증한 PWM 생성기 모듈과 ARM922T Core를 합성하여 SoC를 설계하였다. PWM 생성기 모듈을 구성하는 내부의 각 부분을 VerilogHDL로 코딩하여 심블로 만들어 통합하는 방식으로 설계를 하였으며 실제 모터를 구동하기 위해서 프로세서가 동작할 수 있도록 C언어로 프로그래밍하여 함께 칩에 다운로드하여 테스트를 하였다. SoC를 기반으로한 시스템 설계의 장점은 시스템이 간단해지고 고속의 동작이 가능하며 회로의 검증 및 다양한 시뮬레이션이 용이하다는데 있다.

Keywords : SoC, VerilogHDL, ARM, Excalibur

I. 서론

반도체 산업 및 회로의 집적화 기술의 발전으로 칩 집적도가 증가함에 따라 과거에 보드에 설계했던 시스템을 이제는 하나의 반도체에 집적화 할 수 있게 되었다. 즉 수억개의 트랜지스터(Transistor)로 구성된 하나의 칩에 시스템의 기능을 모두 포함할 수 있게 되었다. 과거에 각각 별개의 칩으로 존재하던 마이크로프로세서(Micro processor), 메모리(Memory) 등이 요즘에는 하나의 칩 안에 존재하게 되었다. 이와 같이 모든 모듈(Module)을 하나의 칩에 집적하는 것을 SoC(System on Chip : 시스템 온 칩)라고 하며 하나의 칩 안에 들어가는 여러 기능의 모듈을 IP(Intellectual Property)라고 부른다. IP의 종류가 다양해지고, 회로의 복잡성이 날로 증가하고 또한 기존에 존재하는 IP를 여러 다른 시스템에서 재사용 할 수 있음에 따라 SoC 설계를 위한 IP 검증이 중요하게 되었다. 이러한 IP들은 다양한 목적의 임베디드 시스템에 많이 사용되고 있다. 우리나라가 반도체 비메모리 분야에서 세계적인 우위를 차지하기 위해서는 SoC의 설계기술의 발전이 필수적이다. 뿐만 아니라 IP 설계 기술의 발전 또한 중요하다.

차세대 국가 성장 동력원으로 주목받고 있는 로봇산업 발전에 있어 로봇제어 기술은 매우 중요하다. 로봇의 움직임을 제어하기 위해서 관절의 모터제어는 핵심 기술이라 할 수 있다.

본 논문에서는 로봇 관절의 모터나 기타 여러 자동화 시스템에 이용되는 모터를 제어하기 위한 SoC를 설계하려 한다. 또한 모터제어용 SoC의 필수 모듈인 PWM(Pulse Width Modulation : 펄스 폭 변조) 생성기 모듈의 설계와 검증을 통해 재사용 가능한 IP를 만들려고 한다. DC모터 제어용 SoC를 모터 제어신호인 PWM 생성기 모듈과 프로세서 모듈인 ARM922T Core를 합성하여 하드웨어를 설계하였다. 이렇게 설계한 하드웨어를 동작시킬 수 있도록 C언어를 이용하여 작성한 구동 소프트웨어를 함께 컴파일(Compile)한 후 칩에 다운로드하여 SoC의 동작을 확인하였다.

II. 본론

1. 하드웨어(Hardware) 설계

DC모터 제어용 SoC를 설계하기 위해 알테라(Altera)사의 엑스칼리버(Excalibur)를 사용하였다. 엑스칼리버는 ARM922T 코어(Core)와 메모리, 주변장치 등을 내장하고 있으며 개발자가 원하는 장치들을 설계할 수 있도록 100만게이트(Gate)의 FPGA를 포함하고 있어 SoC설계 및 임베디드(Embedded) 시스템 설계에 매우 유용한 칩이다. 설계를 위한 소프트웨어 툴로는 알테라사의 퀴터스(Quartus) II를 사용하였다. 퀴터스 II로 하드웨어를 합성하고 시뮬레이션을 할 수 있다.

DC 모터 제어용 SoC는 프로세서인 ARM922T 코어와 PWM 펄스(Pulse)를 출력하는 PWM 생성기 모듈

그리고 일반적인 목적의 8비트(Bit) 출력 포트(Port)로 구성된다. <그림 1>

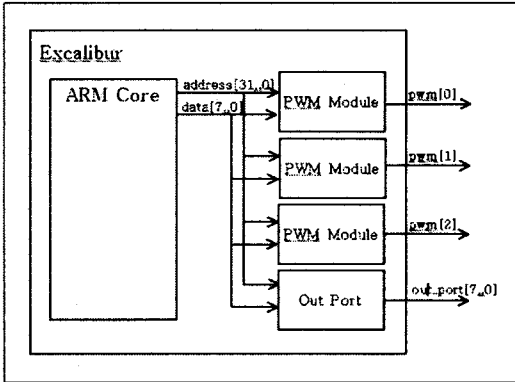


그림 1. 전체 구성도

프로세서로 모듈로는 ARM922T 코어를 사용하였는데 이는 ARM9 계열의 프로세서 코어로 32비트 RISC 프로세서이다. 하바드 구조이며 Fetch, Decode, Execute, Memory, Write의 5단계 파이프라인 방식으로 단일 클럭 사이클(Clock Cycle)에 명령어를 수행한다.[1]

ARM922T는 엑스칼리버 칩내에 장착된 것으로 설계에 사용된 소프트웨어 툴인 퀴터스 II에서 제공하는 MegaWizard 프로그램으로 필요로 하는 기능의 사양을 갖도록 생성하여 사용할 수 있다. 이 프로세서 모듈과 FPGA영역에 설계한 장치들과의 연결은 동작 클럭이 다른 두 영역 간 접속을 위한 브리지(Bridge)를 통해서 이루어진다.

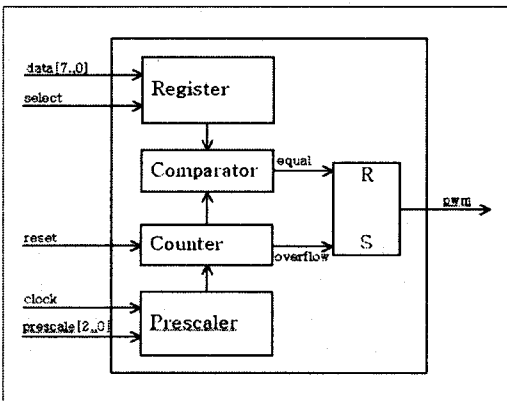


그림 2. PWM 모듈 구성도

PWM 생성기 모듈을 DC 모터 구동을 위해 구성하였다. PWM은 디지털(Digital) 값을 아날로그(Analog) 값으로 변환하는 방법 중 하나로서 출력 펄스의 듀티비

(Duty rate)를 조절하여 DC 모터를 구동하는 제어 값에 변화를 주어 원하는 속도로 DC 모터를 구동하는 방식이다. <그림 2>에서 보는 것과 같이 PWM모듈은 8비트의 데이터를 입력받도록 설계하였으며 이는 256단계의 정밀도를 갖는다.

레지스터(Register) 부분은 어드레스 버스(address Bus)의 주소 값에 의해 select신호가 1이 될 때 데이터 버스의 값을 읽어 저장한다. 8비트 카운터(Counter)는 입력되는 매 클럭마다 증가되며 비교기(Comparator)가 레지스터와 카운터의 값을 비교하여 같으면 출력이 0으로 되고 카운터가 오버플로우(Overflow)되면 출력이 1로 되는 동작으로 PWM 펄스를 출력한다. 또한 PWM 생성기 모듈은 펄스의 주기를 변화시킬 수 있도록 입력 클럭 신호를 최대 8분주 할 수 있는 프리스케일러(Prescaler)를 가지고 있어 여러 종류의 모터를 효율적으로 제어할 수 있다.[2]

PWM 생성기 모듈을 구성하는 각 부분을 Verilog HDL을 이용하여 설계를 하였으며 각각의 디자인 심볼(Design Symbol)을 생성하였다. 이렇게 생성한 각각의 디자인 심볼들로 조합하여 PWM 생성기 모듈을 설계하였다. 이러한 설계 방법을 bottom-up 설계 방법이라 하는데 이는 먼저 필요한 부분들을 만들고 이것들을 이용하여 더 큰 부분들은 만드는 방법이다.[3]

출력 포트는 어드레스 버스의 주소 값에 의해 선택되었을 때 데이터 버스의 값을 읽어 외부로 출력하는 기능을 한다. 이 포트를 모터의 동작허용과 회전방향을 선택하는데 사용하였으며 SoC의 동작상태를 알릴 수 있는 LED를 제어하는 용도로도 사용하였다.

설계된 PWM 생성기 모듈을 칩에 다운로드하여 확인하기 전에 시뮬레이션을 통해 동작을 확인하는 것은 설계된 하드웨어의 오류 발견과 함께 개발 기간을 단축하는데 있어서 중요한 과정이다. <그림 3>은 PWM 생성기 모듈의 시뮬레이션(Simulation) 결과를 보여준다. <그림 3>에서 앞부분은 PWM 데이터가 50일 때의 출력 펄스를 보여주고 뒷부분은 PWM 데이터가 200일 때의 출력 펄스를 보여주고 있다.

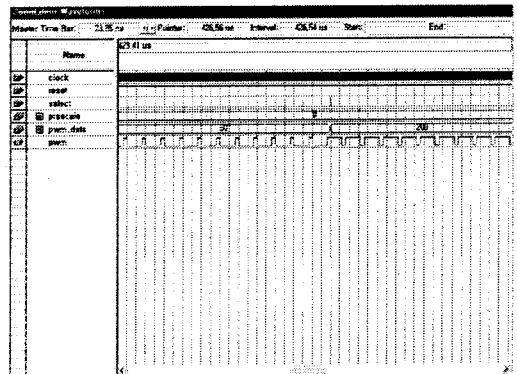


그림 3. PWM 모듈의 시뮬레이션 결과

시뮬레이션을 통하여 확인된 PWM 생성기 모듈과 출력에 대한 명령을 내리는 프로세서 모듈로 SoC를 설계하여 각각 입출력에 맞는 엑스칼리버 칩의 핀들을 배정한 후 컴파일하여 하드웨어를 완성하였다.

2. 소프트웨어(Software) 설계

설계된 SoC 하드웨어의 동작을 확인하기 위해 소프트웨어를 작성해야 한다. 하드웨어 설계시 PWM 생성기 모듈과 출력 포트 모듈에 다음과 같이 주소를 할당하였다.

- PWM0 : FPGA 영역의 8번지
- PWM1 : FPGA 영역의 16번지
- 출력포트 : FPGA 영역의 32번지

```

int main(void)
{
    :
    :
    PWM0_ADD = (volatile unsigned int*)
                (EXC_PLD_BLOCK0_BASE + 8);
    PWM1_ADD = (volatile unsigned int*)
                (EXC_PLD_BLOCK0_BASE + 16);
    OUTPORT_ADD = (volatile unsigned int*)
                  (EXC_PLD_BLOCK0_BASE + 32);

    scale = 4;
    :
    :
    while (1){
        *PWM0_ADD = (scale<<8)||;
        *PWM1_ADD = ((scale-2)<<8)|(255-i);
        *OUTPORT_ADD = 0x03;

        i++;
        delay(200000);
    }
    :
    :
}

```

그림 4. 동작 프로그램

프로그램은 <그림 4>와 같이 동작을 확인할 수 있도록 간단하게 작성하였다. 확인 프로그램의 동작을 보면 PWM0는 데이터가 0부터 255까지 증가하도록 출력하였고 PWM1은 데이터가 255부터 0까지 감소하도록 출력하였다. 출력 포트로는 0x03을 출력하여 PWM0과 PWM1에 연결된 모터의 동작회용과 방향을 결정하여 주었다. scale 변수를 통해 클럭을 4분주 하였다. <그림 4>와 같이 작성된 프로그램을 ADS를 이용하여 컴파일 하였다.

3. 실제 동작 확인

프로세서 모듈과 PWM 생성기 모듈, 출력 포트 모듈을 통합하여 디자인한 하드웨어를 시뮬레이션만으로 동작을 확인하였다. 하지만 확실하고 신뢰성 있는 IP를 검증하기 위해서 실제 동작을 확인하여야 한다. 그래

서 설계된 하드웨어와 이를 동작시킬 소프트웨어를 쿼터스 II에서 함께 컴파일한 후 바이트 블래스터(ByteBlaster) 케이블을 이용하여 엑스칼리버에 다운로드 하여 설계한 SoC에 의해 동작되는 모터를 확인하였다. <그림 5>에 실제 동작을 확인하기 위한 시스템을 보여 주고 있다.

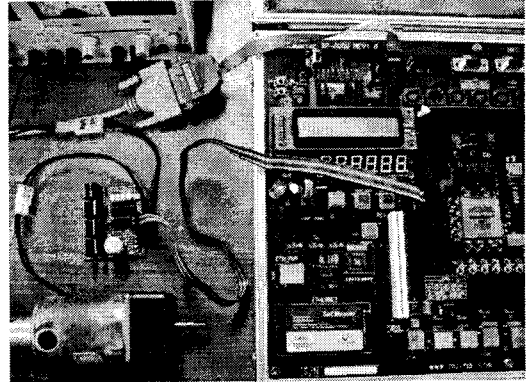


그림 5. 실제 동작 확인을 위한 시스템 구성

시스템은 엑스칼리버 칩을 사용한 실험 장비인 SoC Master Kit와 모터 드라이버 그리고 DC모터로 구성하였다.

III. 결론

본 논문에서 DC 모터 제어를 SoC로 구현하였다. 아울러 DC 모터 제어의 필수요소인 PWM 생성기 모듈을 설계하고 이를 검증함으로써 재사용이 가능한 IP를 구현하였다. 비록 간단한 DC 모터 제어용 SoC이지만 검증된 IP인 PWM 생성기 모듈과 함께 다른 기능의 IP들을 이용하여 로봇제어 시스템, 인터넷을 이용한 원격 제어 시스템 등 다양한 목적으로의 활용이 가능하겠다.

The authors would like to thank the Korea Ministry of Science and Technology and the Korea Science and Engineering Foundation for their support through the ERC program

참고문헌

- [1] Excalibur Device Hardware Reference Manual Version 3.0, Altera, pp.15, July 2002.
- [2] 이우선, 이상우, 인텔 80C196KC 구조·용용 : 모터 자동제어, Ohm사, pp.121-122, 1995.
- [3] Samir Palnitkar, VerilogHDL : A Guide to Digital Design and Synthesis (장훈 옮김), 영한출판사, pp.41-42, 2002.
- [4] Excalibur Hardware Design Tutorial Version 1.5, Altera, 2002