

## 서보모터 구동회로의 FPGA 설계

김기훈, 한광흠, 서동해,  
충북대학교

### Servo motor Driving Circuit Design on FPGA

Ki-hoon Kim, Gwang-heum Han, Dong-hae Seo  
Chungbuk National University

**Abstract** - 본 논문에서는 FPGA를 이용하여 SPWM 펄스파형을 구현했다. 이 파형을 구현하기 위해서 삼각파와 정현파의 비교는 MATLAB을 사용하였고, 비교로 인해서 구현된 파형의 값으로 MAX-PLUS II의 설계를 통해 SPWM 파형을 구현했다. FPGA는 Altera ACEX EPIK100QC208-3N 를 모터는 MITSUBISHI AC SERVO MOTOR HC-KFS053를 사용하였다.

#### 1. 서 론

구동하려는 모터는 AC서보모터로 구동시키기 위해서 3상의 인버터 회로를 사용한다. 3상의 인버터 회로의 경우 모터에 바로 입력 전원을 가하는 것 보다 모터 구동에 있어서 효율이 좋으며 속도를 제어 하는데 있어서도 용이 하다. 3상의 인버터를 제어하기 위해서는 PWM 방식이 필요하다. PWM 방식 중에서 SPWM 방식을 이용할 것이며 SPWM 방식은 고조파 저감방식, 진동 저감방식이라고도 불리며 고조파와 진동에 있어서 효과적인 방식이다.

서보모터를 구동하기 위한 SPWM 파형의 구현을 FPGA의 설계를 통해 보인다. 그동안 모터의 제어에 이용되었던 아날로그 회로가 아닌 디지털 회로를 사용한 모터 제어의 방식을 제안 한다.

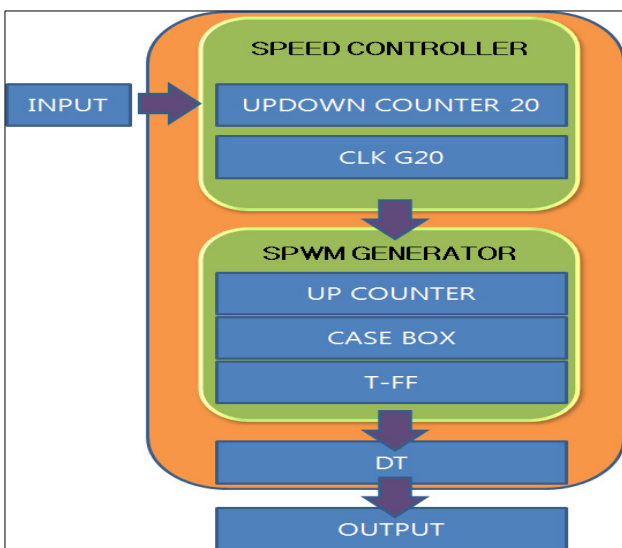
디지털 회로를 사용하게 될 경우 아날로그 회로에 사용되는 많은 소자를 프로그램을 이용한 설계를 통해서 칩 하나로 줄일 수 있으며, 결과의 출력도 정확해진다. 또한 제어가 더 편리해지게 된다.

VHDL 언어를 사용하여 알맞은 제어를 수행하도록 설계 하였으며, 시뮬레이션의 결과 파형을 통해서 결과를 제시한다.

#### 2. 본 론

##### 2.1 서보모터 드라이버

FPGA상에 설계되는 서보 모터 구동회로는 다음과 같은 블록도를 가진다.



〈그림 1〉 FPGA로 만든 서보모터 드라이브의 블록도

블록도에서 INPUT은 UP, DOWN 버튼을 사용해서 원하는 속도를 입력 한다.

SPEED CONTROLLER는 INPUT의 값을 받아서 해당되는 속도의 클

럭을 발생하는 회로로 이의 UPDOWN COUNTER 20 소자는 UP신호가 들어오게 되면 더하고 DOWN 신호가 들어오면 빼서 0~20 까지의 범위로 원하는 속도를 계산한다. CLK G20 소자는 UPCOUNTER의 숫자를 받아서 해당되는 주파수의 클럭을 발생한다. 0일 경우 0Hz 20일 경우 모터 최고속도의 주파수 값이 된다.

SPWM GENERATOR 소자는 클럭 속도에 따른 3상의 SPWM파형을 발생시키는 소자이다. UP COUNTER소자는 클럭을 받을 때마다 카운트를 하고 카운트된 수를 CASE BOX에 전달한다. CASE BOX 는 받은 수에 따라 펄스열을 발생시킨다. T-FF은 발생한 펄스열을 다음 신호가 들어올 때 까지 값을 유지시켜주는 역할로 SPWM 파형을 완성한다.

DT소자는 데드타임 보상으로 회로의 단락상태를 막아주는 역할을 한다.

구현된 SPWM 파형은 아래 식(1)에 따라 모터의 속도제어에 이용된다.

$$F = \frac{P}{2} \frac{n}{60} = \frac{Pn}{120} [\text{hertz}] \quad (1)$$

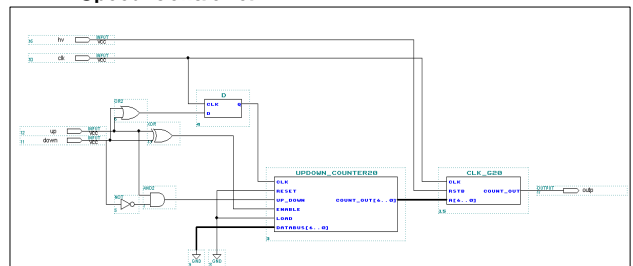
F : 공급전원의 주파수, hertz (Hz)

n : 회전자의 속도, rev/min (RPM)

P : 모터의 극수

만약에 4,500rpm의 속도를 원한다면 사용하는 모터는 8극이므로 SPWM=300Hz, 클럭 주파수는 300Hz \* 2048 = 0.614MHz가 필요하다.

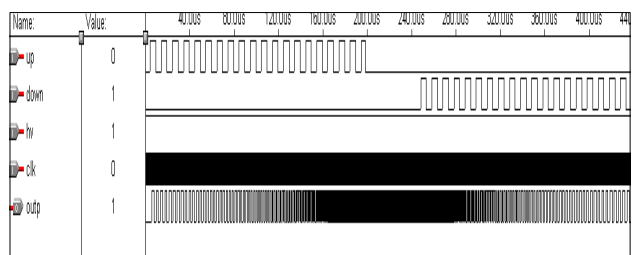
##### 2.1.1 Speed Controller



〈그림 2〉 Speed Controller의 회로

그림 2는 SPWM에 인가하는 펄스의 속도를 결정짓는 부분으로, 원하는 속도를 입력받아 클럭을 발생시킨다.

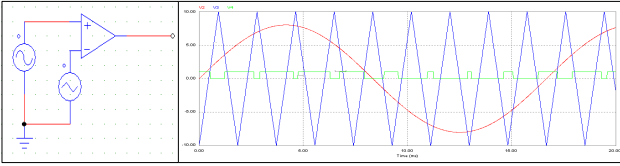
UP과 DOWN을 누른 수를 UPDOWN COUNTER 20에서 계수하여 CLK G20에서 그에 상응하는 클럭을 발생시키도록 만들었다. UPDOWN COUNTER20은 0~20까지 계수하고 0보다 작거나 20보다 클 경우 더 이상 계수 되지 않도록 설계했다. 또한 UP과 DOWN을 동시에 누르면 계수 되지 않도록 되었다. D-FF의 역할은 UP과 DOWN을 누를 때 UPDOWN COUNTER20을 동작시키기 위해 지연을 주기 위한 역할이다.



〈그림 3〉 Speed Controller의 출력 파형

그림 3은 UP을 입력했을 때 OUTPUT의 주파수가 빨라지고 DOWN을 입력했을 때 OUTPUT의 주파수가 느려지게 되는 것을 볼 수 있다.

### 2.1.2 SPWM Generator

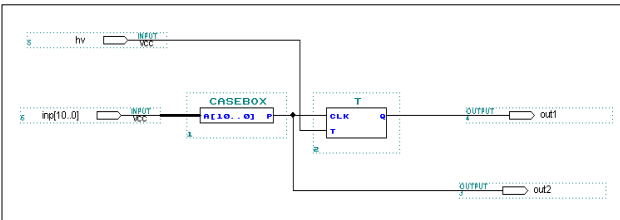


〈그림 4〉 PSIM을 통해 구현한 삼각파와 정현파 SPWM 파형

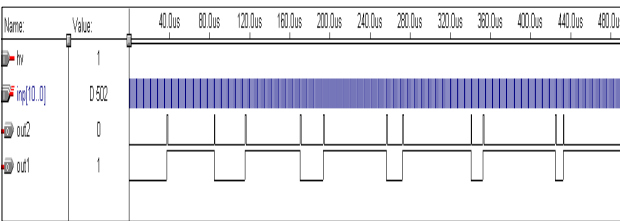
SPWM은 삼각파와 정현파를 비교하여 생성되는 펄스 파형이다. 따라서 위의 파형을 FPGA로 구현하기 위해 MATLAB으로 수치를 얻어 그에 상응하는 값을 FPGA에서 구현하여 파형을 만드는 방법을 사용했다. 정현파와 삼각파의 주파수 비율을 1:25로 만들었다.

MATLAB으로 삼각파와 정현파의 교차점을 구하였고 디지털로 구현하기 위해서 교차점의 값을 정수에서 2진수로 바꾸어 주었다.

위의 데이터를 통해서 VHDL로 SPWM을 나타내도록 만든다. 해당되는 X축의 좌표에서 펄스열을 나오게 한다. 그리고 그 펄스열이 T-FF를 지나가게 해서 SPWM을 만든다.

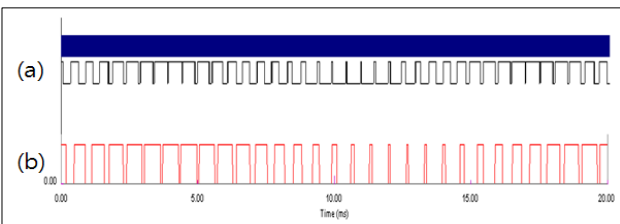


〈그림 5〉 SPWM GENERATOR 회로



〈그림 6〉 SPWM GENERATOR 출력 파형

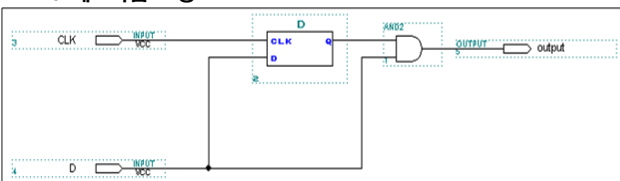
OUT2는 CASE BOX에서 나오는 펄스열이 출력되는 것을 볼 수 있다. CASE BOX는 입력되는 숫자에 따라 저장된 수치에서 펄스열을 만드는 역할을 한다. OUT1은 그 펄스열을 T-FF를 지나가게 하여 펄스로 만들어 주는 역할을 한다.



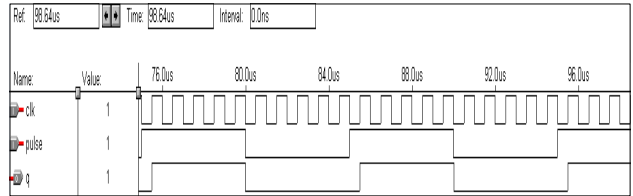
〈그림 7〉 (a) FPGA에서 구현한 SPWM파형 (MAXPLUS), (b) 아날로그 비교기를 이용한 시뮬레이션 결과 파형 (PSIM)

결과 파형은 위와 같이 SPWM이 만들어 지는 것을 확인할 수 있고, 아날로그 비교기를 이용한 파형과 같은 것을 볼 수 있다.

### 2.1.3 데드타임 보상



〈그림 8〉 데드타임 보상 소자

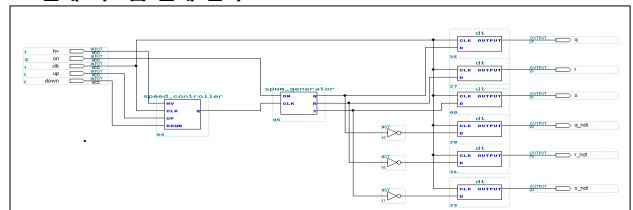


〈그림 9〉 데드타임 보상 결과 파형

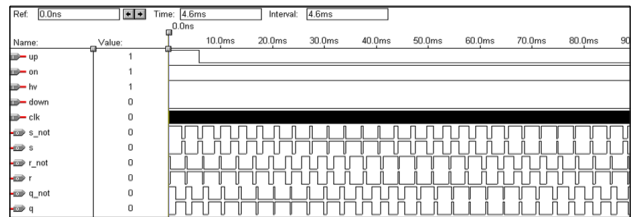
데드타임 보상은 3상 인버터 회로에서 쌍을 이루고 있는 소자가 상태의 전환이 이루어질 때 마다 짧은 시간 동안 두 개의 소자를 모두 OFF 시키는 것이다. 이것을 행하는 이유는 두 개의 소자가 전환 시에 동시에 ON상태가 되는 과도구간이 발생하면 인버터 회로는 단락 상태가 될 수 있기 때문이다.

회로도에서 데드타임 보상을 구현하는 소자는 DT 소자이다. 이 소자는 입력되는 클럭이 UP이 될 시에 한 클럭을 지연시켜 출력 클럭을 UP 시키는 소자로 이 소자를 이용하면 두 개의 소자가 동시에 ON 상태가 되는 것을 막을 수 있다.

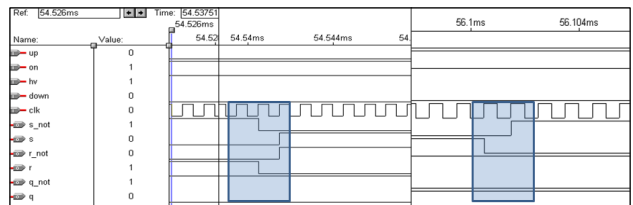
### 2.2 전체 시스템 설계 결과



〈그림 10〉 설계된 회로도



〈그림 11〉 최종 출력된 6개의 3상 파형



〈그림 12〉 확대된 최종 출력 파형

그림 11은 앞에서 설명한 각 과정을 모두 지나고 출력된 출력 파형이다. 스위치 6개에 들어갈 각 신호들이 보이며 이 신호로 인해 6개의 스위치는 ON, OFF 동작을 하게 된다. 그림 12의 확대된 파형을 보면 블록 안에 데드타임 보상을 보여주고 있다. 각각의 신호가 ON이 되는 것은 입력 클럭의 한 주기 뒤에서 이루어지는 것을 확인할 수 있다. 이것으로 3상 인버터에 입력이 될 SPWM펄스 파형이 만들어 졌다.

### 3. 결 론

본 연구에서는 아날로그가 아닌 디지털을 이용한 모터제어 신호를 구현했다. 결국 모터의 속도를 제어하기 위해서는 파형의 주파수를 제어해야 하고 주파수의 제어를 위해서 UP, DOWN 방식을 이용했다. 모터의 최대 주파수를 20으로 나누어 해당 범위만큼 주파수가 변경되면서 속도 제어는 이루어지게 된다. 20의 값을 더 크게 가져가게 되면 더욱 세밀한 모터의 속도제어가 이루어질 것이다.

### [참 고 문 헌]

[1] Afarulrazi Abu Bakar, Md Zarafi Ahmad and Farah Salwani Abdullah, "Design of FPGA Based SPWM Single Phase Inverter", MUCET2009, 2009  
 [2] 야경산업 자동화 연구소, "서보모터 제어이론과 실습", 1997