

## NTC FPGA 강좌 10. ADC 사용하기 (FSM 응용)

(주) 뉴티씨 (NewTC)

<http://www.NewTC.co.kr>

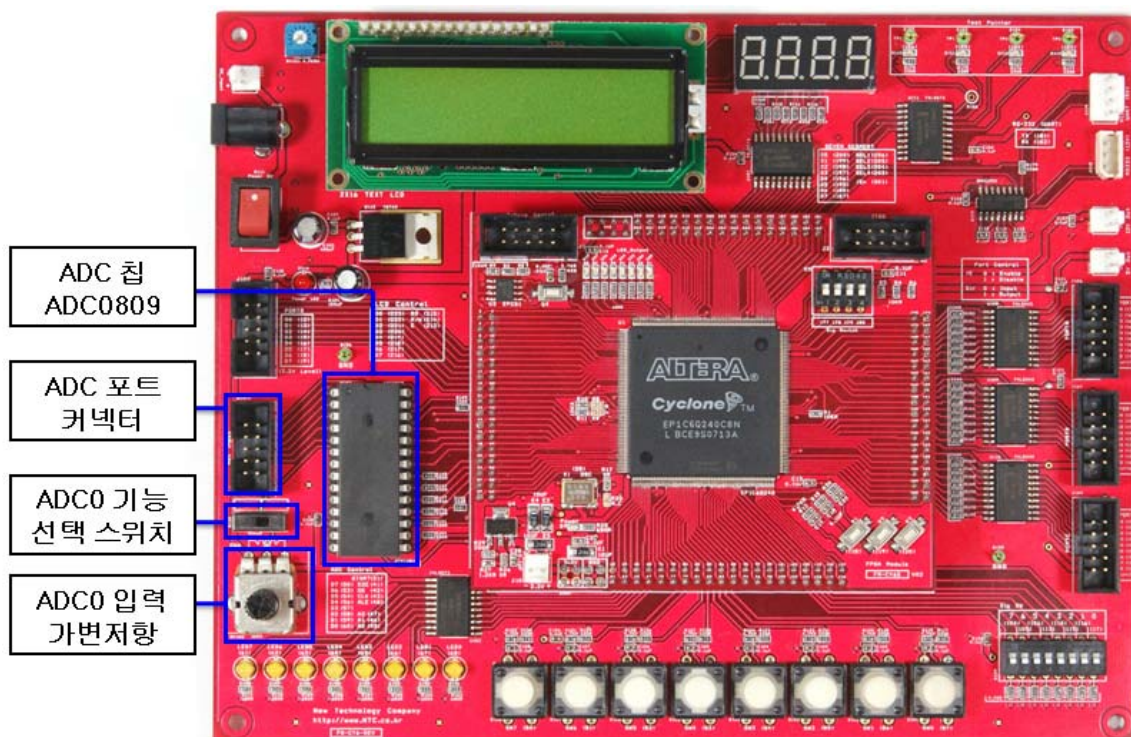
### 1 ADC 란?

ADC(Analog to Digital Converter)는 Analog 신호 입력을 받아서 Digital 값으로 변환해 주는 장치입니다. FB-CY6-DEV 보드에는 ADC0809 라는 저속의 ADC 칩이 내장되어 있습니다. 이 ADC를 이용하여 보드에 장착된 가변저항의 회전 정도를 감지하거나 외부 커넥터에 센서를 연결하여 아날로그 값을 디지털로 변화하여 읽을 수 있습니다.

이 ADC는 FPGA에 비하여 상당히 느리기 때문에 FSM을 이용하여 제어를 해야 합니다.

#### 1.1 ADC 칩과 주변 장치

FB-CY6-DEV 보드의 좌측에 Dip 타입 ADC 칩과 입력 포트, 그리고 가변저항이 있습니다. 가변저항은 ADC 컨트롤 시에 간단하게 입력을 변화시켜 볼 수 있습니다. 이 가변저항으로부터 입력을 받으려면 ADC0 기능 선택 스위치를 RV 쪽으로 위치시켜야 합니다.

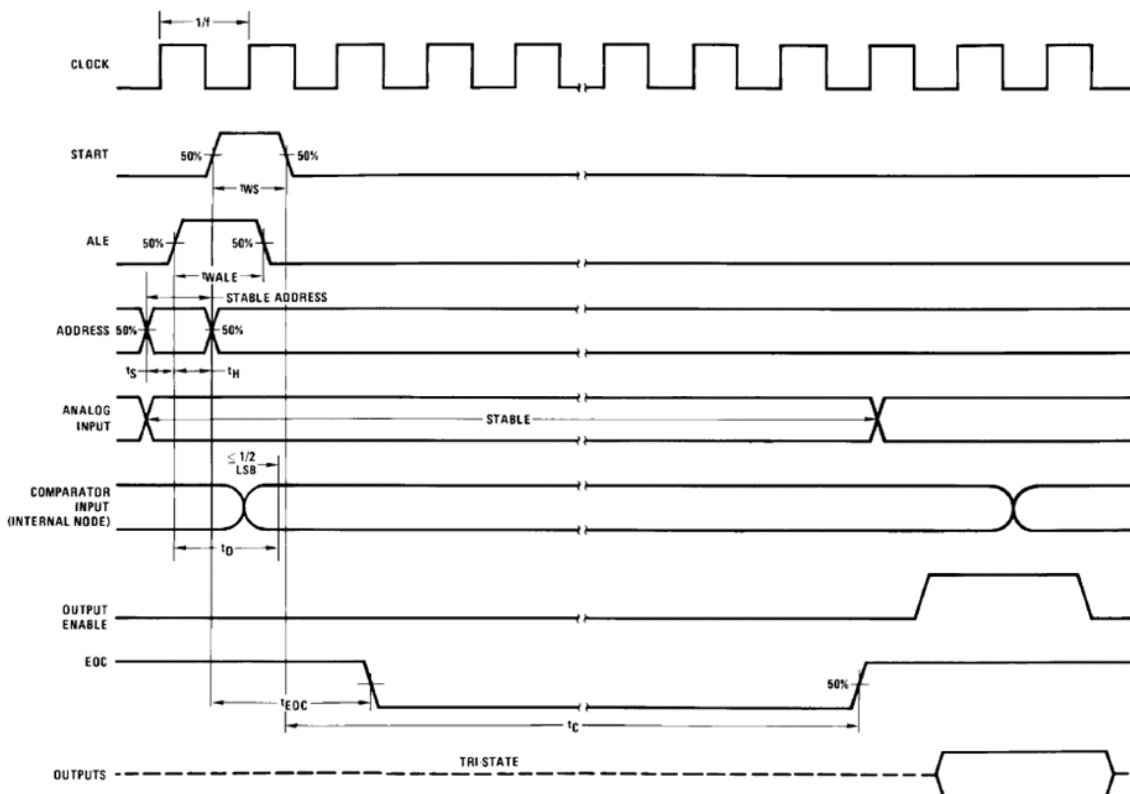


## 1.2 ADC0809 동작

보드에 장착된 ADC0809는 아래와 같은 특징을 가지고 있습니다.

분해능 (Resolution)	8Bit
채널 수	8Ch
AD 변환 속도	100us
AD 변환 범위	0V~5V

보드에 장착된 ADC0809는 저속의 ADC로 먼저 ADC를 START 시킨 후 변환이 완료된 후 데이터를 읽어 오는 구조입니다. ADC를 START 시키기 위해서는 어드레스 신호, ALE, START를 아래 파형과 같이 인가해야 합니다. ADC는 동작 속도가 느리기 때문에 변환하는데 시간이 걸립니다. 변환이 완료된 후에 EOC (End Of Conversion) 신호가 발생하는데 이 신호가 발생한 후에 ADC에 OE 신호를 인가하면 AD 변환된 데이터가 나오게 됩니다. 아래와 같은 파형을 발생시키고 ADC 데이터를 입력 받을 수 있도록 FSM을 설계해야 합니다.

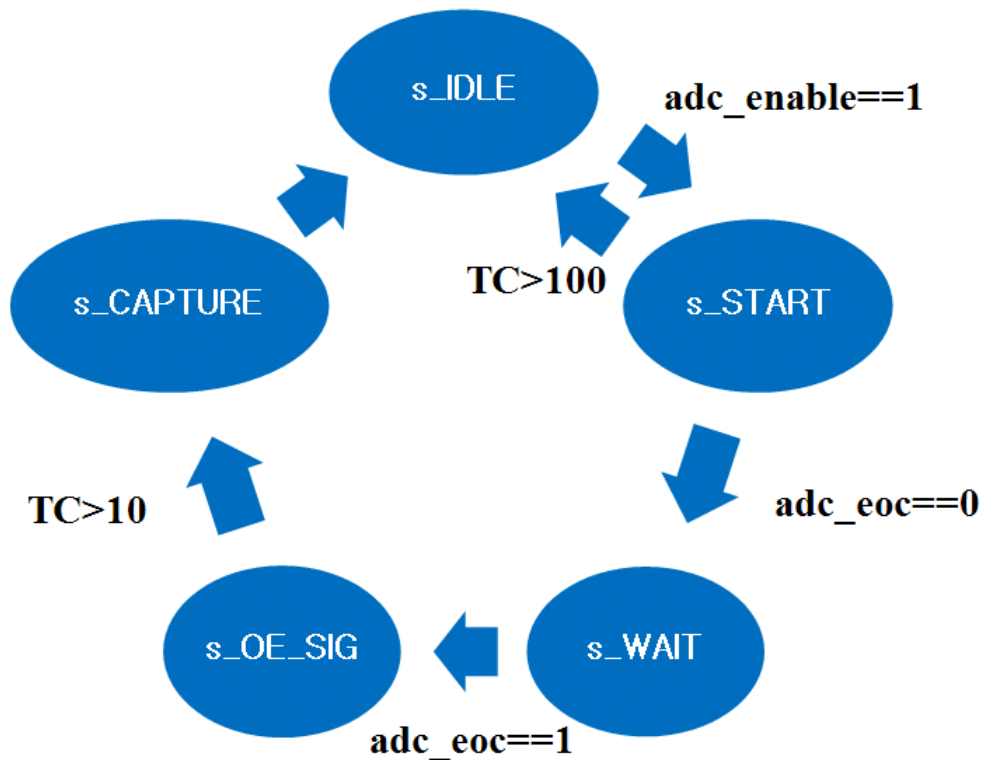


ADC0809 제어 신호 타이밍도

## 2 FSM 설계하기

### 2.1 ADC 제어 FSM 설계

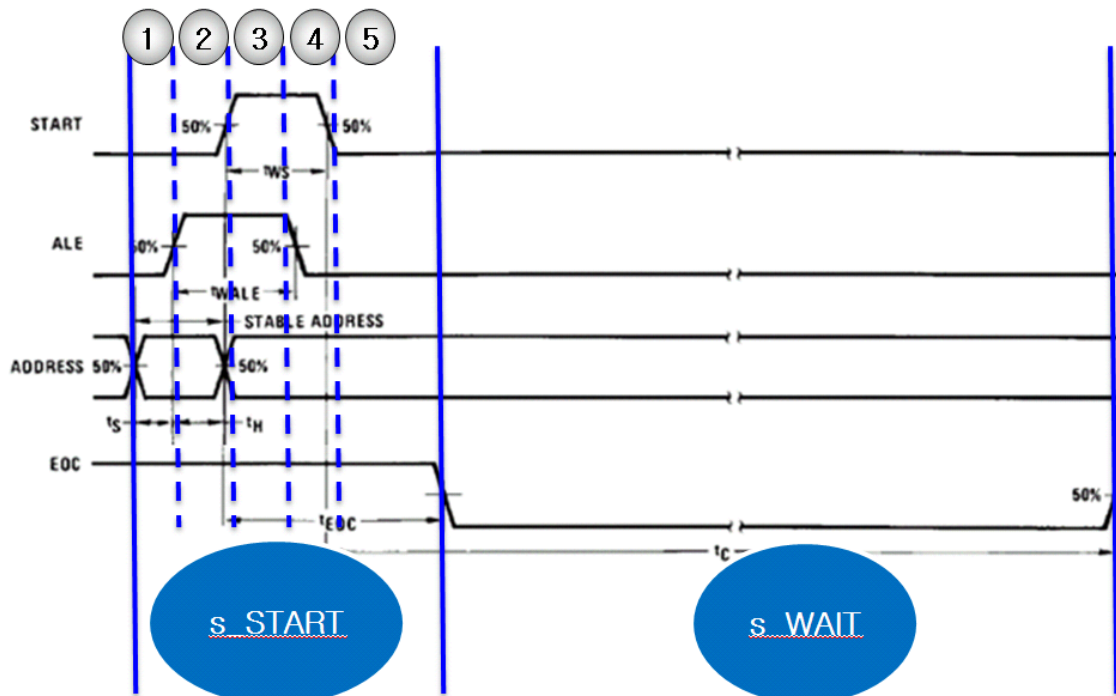
아래 그림과 같이 5개의 상태로 표현할 수 있습니다. s\_IDLE 상태에서 adc\_enable 신호가 입력되면 ADC를 동작 시키게 됩니다. s\_START 상태에서 어드레스 신호, ALE, START를 인가하고 adc\_eoc 신호가 0으로 바뀌는 것을 감지하여 s\_WAIT 상태로 가게 됩니다. adc\_eoc 신호가 100 클럭 동안 발생하지 않을 경우 타임아웃이 발생하도록 하였습니다. s\_WAIT 상태에서 adc\_eoc 신호가 1로 바뀔 때까지 기다리게 됩니다. adc\_eoc 신호가 발생한 것은 AD 변환이 완료된 것을 의미하기 때문에 adc\_eoc 신호가 발생한 다음에는 ADC 칩에서 데이터를 읽을 수 있도록 OE 신호를 발생시킵니다. 시간 지연 후에 s\_CAPTURE 상태에서 데이터를 읽어 들이고 s\_IDLE 상태로 가게 됩니다.



ADC0809 제어 FSM

s\_START 상태에서는 카운터를 동작시키면서 아래와 같이 파형을 발생시킵니다. s\_START 상태에서 발생시키는 신호는 START, ALE, ADDRESS 3개의 신호입니다. 각 신호의 변화를 보면 아래의 ①,②,③,④로 구분할 수 있으며 각 단계마다 시간 지연은 약 50ns ~ 80ns 정도입니다. 보드에 장착되어 있는 클럭이 50Mhz로 한 클럭 사

이클 타임이 20ns 이기 때문에 3~4 클럭 정도 주기마다 신호를 변화 시키면 됩니다.  
①~④ 신호를 인가한 후 ⑤단계에서는 EOC 신호가 0이 될 때 까지 기다린 후 s\_WAIT 상태로 넘어가게 됩니다.



s\_START 상태에서 신호를 출력하는 부분은 아래와 같이 tc 값을 기준으로 바뀌도록 구현할 수 있습니다.

```
//output by state
always @ (posedge clk) begin
    if(!reset) begin
        adc_ale      <= 0;
        adc_start    <= 0;
        adc_address   <= 0;
    end else begin
        if(state == s_START) begin
            if(tc < 3) begin
                adc_ale      <= 0;
                adc_start    <= 0;
                adc_address   <= con_addr;
            end else if(tc < 6) begin
                adc_ale      <= 1;
                adc_start    <= 0;
            end
        end
    end
end
```

```

                                adc_address    <=  con_addr;
                                end else if(tc < 9) begin
                                    adc_ale      <= 1;
                                    adc_start    <= 1;
                                    adc_address  <= 0;
                                end else if(tc < 12) begin
                                    adc_ale      <= 0;
                                    adc_start    <= 1;
                                    adc_address  <= 0;
                                end else begin
                                    adc_ale      <= 0;
                                    adc_start    <= 0;
                                    adc_address  <= 0;
                                end
                                end
                                end
                                end
end

```

## 2.2 ADC 클럭 공급

ADC 동작을 위해서 클럭을 공급해야 합니다. 여기서 사용되는 ADC0809에는 약 1Mhz 클럭을 공급해야 합니다. 입출력 신호와 동기를 맞춰줄 필요 없이 별도로 공급합니다. 개발보드의 경우 내부에 50Mhz 클럭 소스를 사용하고 있기 때문에 50분주를 하여 1Mhz신호를 만들어 ADC에 공급합니다.

## 실습 과제

1. ADC 입력을 받아 LED 또는 7-Segment 로 출력합니다. adc\_enable 신호를 1로 셋팅하여 AD 변환이 계속 동작하도록 합니다. 가변저항을 돌리면서 ADC 출력을 확인합니다.
2. 과제 1에서 설계한 AD 변환을 1초에 1번씩 동작하도록 수정합니다. (s\_IDLE에서 1초를 기다렸다가 1초 후에 동작하도록 FSM 부분을 수정합니다.)