

# Practice 8 - How to use FPGA board

2017-19428

컴퓨터공학부 서준원

## 1. Introduction



이번 실습에서는 zedboard를 사용하는 방법을 알아보았다. Digilent의 ZedBoard는 Xilinx Zynq-7000 전체 프로그래밍 가능 SoC(AP SoC)용 저가형 개발 보드다. 이 보드에는 Linux, Android, Windows 또는 기타 OS/RTOS 기반 설계에 필요한 모든 것이 포함되어 있다. 또한 여러 확장 커넥터가 제공되므로 사용자가 프로세싱 시스템과 프로그래밍 가능 논리 I/O에 쉽게 접근할 수 있다.

이를 사용해 Matrix multiplication을 구현할 것이다. 이를 사용하는 법을 알아보기 위해 User-guide를 포함한 다양한 다큐멘테이션을 읽는 방법을 익히고 간단한 프로그램을 실행시켜보았다. User guide에는 I/O port를 포함해 어떤 장치들이 있는지 나열되어 있다. 또한 이를 어떻게 사용해야 할 지를 친절하지는 않지만 설명해준다. SoC기 때문에 그냥 연결해주는 것뿐만 아니라 전압 등을 따로 설정해주어 Synthesis를 해주어야 한다.

프로그램을 모두 작성한 후에는 usb로 연결하여 보드에 코드를 옮기었다. 이것이 제대로 실행되는지 확인해보는 것이 이번 실습의 주된 내용이었다.

## 2. Implementation

이번 실습에서는 1-sec checker를 구현하였다. 이는 FPGA에서 초시계를 구현한 것으로, Zedboard에 내장된 clock을 이용하여 1초마다 올라가는 counter를 만든 것이다. Zedboard에 내장된 clock은 100MHz이므로, 100M번을 센 후에 한 번씩 카운터를 증가시켰다. 이 결과를 LED에 2진법의 형태로 visualize 했다.

카운터의 코드는 아래와 같이 간단하게 작성 가능하다.

```
always @(posedge GCLK or posedge BTNC) begin
```

```
    if(BTNC) begin
```

```
        down_counter = 100000000;
```

```
        LD = 0;
```

```
    end
```

```
    else begin
```

```
        down_counter = down_counter - 1;
```

```
        if(down_counter == 0) begin
```

```
            LD = LD + 1;
```

```
            down_counter = 100000000;
```

```
        end
```

```
    end
```

```
end
```

뿐만 아니라 asynchronous한 reset또한 구현하였다. Zedboard의 내장된 center\_button이 클릭되었을 때 카운터를 리셋하도록 구현하였다. 이렇게 베릴로그의 코드를 작성한 후, 우리 코드의 인풋과 아웃풋을 Zedboard의 구성요소와 연결해주는 작업을 해줘야 한다.

이는 Constraint로 지정해주며, set\_property를 이용해 input/output port와 package pin을 연결해주었다. 이거들은 User Guide에서 이름을 찾을 수 있으며 보드에도 작게 표시되어 있다.

```
set_property PACKAGE_PIN T22 [get_ports {LD[0]}}; # "LD0"
```

```
set_property PACKAGE_PIN T21 [get_ports {LD[1]}}; # "LD1"
```

```
set_property PACKAGE_PIN U22 [get_ports {LD[2]}}; # "LD2"
```

```

set_property PACKAGE_PIN U21 [get_ports {LD[3]}; # "LD3"

set_property PACKAGE_PIN V22 [get_ports {LD[4]}; # "LD4"

set_property PACKAGE_PIN W22 [get_ports {LD[5]}; # "LD5"

set_property PACKAGE_PIN U19 [get_ports {LD[6]}; # "LD6"

set_property PACKAGE_PIN U14 [get_ports {LD[7]}; # "LD7"

set_property IOSTANDARD LVCMOS33 [get_ports -of_objects [get_iobanks 33]];

```

또한 버튼이나 LED같은 것들을 사용한다면 보드에 전압을 걸어줘야 한다. 이또한 constraint에서 Set\_property로 지정해주었다. 이것을 설정하지 않으면 에러가 남을 확인할 수 있었다.

```

set_property PACKAGE_PIN Y9 [get_ports {GCLK}; # "GCLK"

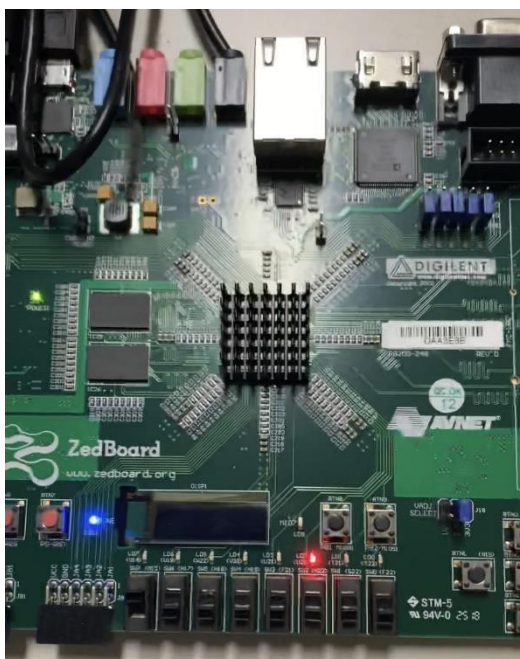
set_property PACKAGE_PIN P16 [get_ports {BTNC}; # "BTNC"

set_property IOSTANDARD LVCMOS33 [get_ports -of_objects [get_iobanks 13]];

set_property IOSTANDARD LVCMOS25 [get_ports -of_objects [get_iobanks 34]];

```

### 3. Result



이 사진과 같이 1초마다 카운트가 올라갔다. 또한 오른쪽 버튼들 중 가운데 버튼을 누르면 리셋이 가능하다. 동영상을 통해 결과를 확인할 수 있다.