

Practice 9 - OS + FPGA System

2017-19428 서준원

1. Overview

이번 실습은 Zedboard에 실제로 OS와 프로그램을 올려서 프로젝트를 실행시키는 작업을 했다. 첫째로 Vivado 프로그램을 이용하여 IP를 작성했다. 처음에는 IP를 활용하지는 않았기 때문에 Processing System과 BRAM만을 만들었고 그것을 연결해주는 Connectivity를 설정했다.

이 프로젝트를 완성하면 bitstream 파일이 생기는데 이를 sd카드에 담아 zedboard가 실행되면 이것을 실행하게끔 하였다.

완성된 비트스트림 파일을 리눅스 OS를 활용해 FPGA를 사용하여 실행했다. 리눅스 운영체제 상에서 비트스트림으로 만들어 놓은 BRAM을 접근하여 c 프로그램을 실행할 수 있었다. 운영체제가 Processing system을 사용하여 c 프로그램을 실행할 수 있도록 했다. Zedboard에 랜선을 연결하여 원격 깃 저장소로부터 코드를 다운받았다.

그리고 마이크로 5핀 케이블을 이용하여 USB 포트에 Serial 통신을 통해 컴퓨터에서 Zedboard를 제어했다. 즉, Zedboard에 Sd 카드에 있는 OS를 내 컴퓨터에서 사용할 수 있게 된 것이다.

이 과정에서 많은 시행착오를 거쳤다. 처음에는 리눅스 운영체제를 사용하여 minicom을 사용하여 파일 시스템으로 zedboard를 접근하려 했지만 알 수 없는 이유로 접근할 수 없었다. 지금 생각해보면 usb를 serial로 접근하기 위한 적절한 드라이버가 설치되지 않아서 인 것 같다.

그 후 윈도우에서 putty를 사용하여 시도해봤다. Putty는 프로그램 자체에서 Connection method를 지정해줄 수 있기 때문에 연결에 성공했다. 이를 통해 예시 프로그램을 실행하고, zedboard 상에서 메모리를 사용하는 방법에 대해 연습해볼 수 있었다.

2. Explanation

실행한 예시 프로그램은 아래와 같다. Mmap을 이용하여 fpga 상의 bram과 ip를 매핑한다. Bram은 4개의 ineeger만큼을 매핑하며 시작 주소는 bram의 주소인 0x40000000이다. 그리고 튜토리얼의 코드와는 다르게 fpga의 ip또한 mmap으로 메모리 매핑한다.

여기서 bram에는 2 * SIZE 만큼을 접근하는데 mmap에서는 SIZE만큼만 매핑을 한다. 이는 mmap 구현 상 페이지 단위로 매핑을 하기 때문에 큰 문제가 생기지 않는다.

또한 코드를 통해 ip의 주소가 0x43c00000임을 알 수 있다.

```
int foo = open("/dev/mem", O_RDWR);
int *fpga_bram = mmap(NULL, SIZE * sizeof(int), PROT_READ|PROT_WRITE, MAP_SHARED, foo, 0x40000000);
int *fpga_ip = mmap(NULL, sizeof(int), PROT_READ|PROT_WRITE, MAP_SHARED, foo, 0x43C00000);

// initialize memory
for (i = 0; i < SIZE; i++)
    *(fpga_bram + i) = (i * 2);
for (i = SIZE; i < SIZE * 2; i++)
    *(fpga_bram + i) = 0.0f;

printf("%-10s%-10s\n", "addr", "FPGA(hex)");
for (i = 0; i < SIZE * 2; i++)
    printf("%-10d%-10X\n", i, *(fpga_bram + i));

// run ip
*(fpga_ip) = 0x5555;
while (*fpga_ip == 0x5555);

printf("%-10s%-10s\n", "addr", "FPGA(hex)");
for (i = 0; i < SIZE * 2; i++)
    printf("%-10d%-10X\n", i, *(fpga_bram + i));
```

처음에는 SIZE개 (4개)만큼은 2i로 지정해 놓은 후, ip를 실행하기 전에 bram의 값을 출력해본다. 당연히 0으로 값을 넣어 놓은 부분은 0으로 나오게 된다.

그 후 fpga_ip의 첫 값에 0x5555를 넣어주며 ip를 run하게 된다. Bitstream 내부를 알 수 없지만 그 값을 통해 ip를 시작시킬 수 있는 것으로 추정된다. IDLE 상태일 때 0x5555이고, Done이면 0x5555가 아닌 다른 값으로 해당 메모리 값을 바꿔준다. 또한 ip가 끝나게 되면 그 값이 0x5555가 아닌 값으로 바뀌게 되어 ip의 실행이 끝났다는 것을 알 수 있다.

ip가 실행된 후 출력된 값을 보면 값이 2배가 되어 들어간 것을 볼 수 있다. 정확히 얘기하면 bram의 2 * i번째 주소의 값이 i주소 값의 2배가 되어 저장되었다.

C 코드를 수정하여 다른 값을 처음에 넣어줘도 2배가 되는 것을 확인할 수 있었다.

이를 통해 귀납적이긴 하지만 ip는 left shift 연산을 SIZE번 실행하는 것을 알 수 있었다.