

VHDL과 Verilog를 이용한 FPGA 로봇설계

진현수*, 채규수*
 *백석대학교 정보통신학부
 e-mail : jhs1020@bu.ac.kr

Design of Robort using VHDL and Verilog

Hyun-Soo Jin*, Gyu-Soo Chae*
 *Information & Communication Devision, BaekSeok University

요 약

본 논문에서는 SoC 키트에 해당하는 iRoV-Lab 3000의 장착된 로봇 모듈인 FPA 모듈, Stepper Motor 모듈, 적외선 송수신 센서 모듈, 카메라 모듈, RF 모듈 LED, TEXT LCD, 7-segment를 제어하기 위한 FPGA를 사용하며, FPGA설계를 위해 Schematic Design 또는HDL에 대해 연구한다.FPGA의 내부구조를 이해하고 개발환경을 구축할수 있다. 로봇의 구성요소와 각각의 구성요소(Sensor 모듈, display 모듈, Stepper Motor 모듈, RF 모듈)의 동작 원리를 개발한다.

1. 서론

1.1. FPGA의 내부구조 및 개발환경구축

FPGA(Field Programmable Gate Array)소자를 디지털 시스템 설계에 활용하기 위해 SoC와 PLD의 개념과 구조에 대해 살펴보고 시스템을 설계하기 위한 FPGA 개발 환경 구성에 대한 내용을 다룬다. 즉, PLD 소자의 종류와 특성을 이해한다.FPGA를 사용한 개발 환경을 구성할 수 있다.

1.1.1 Schematic Diagram을 사용한 Counter 설

계 및 LED 표시

iRov-Lab 3000의 Main module에 설치된 50MHz의 Clock과 Counter 회로를 사용하여 일정 시간 간격에 따라 LED가 순차적으로 켜지는 동작을 실습한다. 실습예제에서는 0.2초 마다 LED가 2진수 표기로 1씩 증가하며 켜지는 동작을 구현한다.

1.1.1.1시행 방법

부록의 내용을 참조하여 Project navigator를 실행하고 새로운 프로젝트를 생성한다.예제에서 프로그램의 이름은 counter_led로 지정한다. 프로젝트의 생성, 디바이스의 설정등은 참고로 남는다.

설계에 사용할 소자는 cd4ce라는 이름을 가진 10진 카운터를 사용한다. 앞의 예제에서와 같이 Schematic editor에서 소자를 선택하여 도면에 배치한다.cd4ce소자는 CE핀이 High의 상태일

때 clock입력의 상승 에지에서 카운트 동작을 수행하여 그 결과 값을 Q0,Q1,Q2,Q3에 출력하며 10진수로 환산하

여 출력 Q0~Q3의 값이 9가 될 때 TC와 CEO의 출력이 High가 된다.CLR 신호가 High일때 출력은 모두 Low의 값으로 초기화되며, Clock 신호의 입력에 따라 정상적인 Count 동작을 하기 위해서는 CE는 High, CLR은 Low의 값을 가져야 한다.

2. 실습 키트

iRob-Lab 3000의 실습 장치에는 50MHz의 클럭을 사용하고 있으므로 이 값을 10,000,000으로 나누면 5Hz가 된다. 즉 0.2초마다 한번씩 변하는 파형을 만들 수 있다. cd4ce소자는 하나를 거칠 때 마다 입력 클럭을 10으로 분주한 값을 CEO에 출력하므로 7개의 cd4ce를 거치도록 연결하면 5Hz의 신호를 만들 수 있고 이 값을 또 다른 cd4ce에 입력하여 그 출력을 LED에 표시되도록 하면 0.2초마다 하나씩 증가되는 값을 LED에 표시할 수 있다. 또한 이 카운터의 CEO출력을 또 다른 cd4ce에 연결하여 사용할 경우 LED를 각각 두 개의 그룹으로 나누어 10의 자리와 1의 자리로 나누어 2진수 형태로 값을 표시할 수 있다. 실습 예제에서는 모두 9개의 cd4ce를 사용하여 LED 8개를 두 개의 그룹으로 나누어 10의 자리, 1의 자리를 4개씩의 LED에 표시한다.

3. 구조 설계

도면에 배치된 소자들 사이의 연결은 Add ->

Wire를 사용하여 이 메뉴의 선택은 앞의 I/O Makerdml 배치와 같은 방식으로 툴바의 메뉴, 마우스의 오른쪽 버튼을 사용한 팝업 메뉴등을 선택하여 지정할 수 있다. 소자들 사이의 결선은 Add -> Wire모드에서 핀의 끝 부분에 마우스 커서를 위치할 경우 커서의 모양이 일반 + 모양에서 아래와 같은 모양으로 변경되며 이때 마우스를 클릭하면 해당 위치에서부터 배선을 시작할 수 있다. Project navigator에서는 연결할 두 지점을 클릭하면 자동으로 그사이에 배선을 한다.

4. 핀설정 실행

설계 도면의 핀과 FPGA의 출력 핀과의 설정을 위해 이번 예제에서는 Text editor를 기반으로 지정하는 방법을 설명한다. Implementation Constraints File을 counter_led라는 이름으로 추가하고 source창에서 추가한 counter_led.ucf파일을 선택한 후 process창에서 User constraints 항목을 확장하여 “Edit constraint(Text)”를 더블클릭하여 실행한다 텍스트 형식의 핀 설정을 편집하는 화면이 나타나면, Pin list를 참조하여 다음과 같이 입력한다

```
NET "CLOCK" LOC = "T9";
NET "LED_0" LOC = "N5";
NET "LED_1" LOC = "P7";
NET "LED_2" LOC = "T5";
NET "LED_3" LOC = "T8";
NET "LED_4" LOC = "T3";
NET "LED_5" LOC = "R3";
NET "LED_6" LOC = "T4";
NET "LED_7" LOC = "R4";
```

[그림 1] 핀 설정

5. HDL의 기본 문법 및 설계실습

이번 장에서는 HDL의 구문을 학습하여 이를 활용하여 설계한다. HDL에는 여러 종류가 있으나 대표적으로 많이 사용되는 VHDL과 Verilog HDL에 대하여 설명하고 HDL 구문을 사용하여 설계한 회로의 동작을 iRoV_LAB 3000 실습 장치의 FPGA에 구현하여 그 동작을 확인하는 내용을 다룬다. 아래 페이지의 VHDL 코드들은 HALF-Adder를 표기한 것으로서 세 코드 모두 동일한 동작을 한다.

Structural description은 회로도나 동일한 관점에서, Behavioral description은 알고리즘 측면에서, Dataflow description은 논리식을 사용하여 신호의 흐름을 나타내는 관점에서 기술한다.

VHDL을 이용한 모델링 기법은 다음과 같은 4가지 유형으로 정리되며 아래의 그림의 코드는 모두 동일한 기능(1-bit 반가산기)을 수행하나 모델링한 방법에 따라 각각 다르게 표현된다. 그림 1-2는 모델링 예제에 공통적으로 사용되는 Entity 구문이다. Entity구문은 회로의 입/출력 신호를 표현하는 부분이다.

```
library ieee;
use ieee_std_logic_1164.all;
entity half_adder1 is
    port( X,Y           : in std_logic;
          SUM,C_OUT     : out std_logic);
end half_adder1;
architecture behaviral of half_adder is
    signal sel :std_logic_vector( 1 downto 0);
begin
    SEL <= X&Y;
    process (SEL)
```

[그림 2] Modeling 예제에 사용되는 entity 구문

6. VHDL문법구조

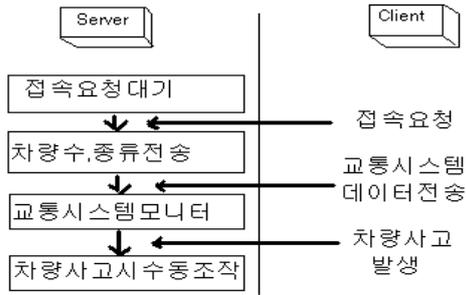
이번VHDL의 기본 문법 구조은 앞에서 언급한 것처럼 크게 두 부분으로 구성되며 VHDL은 entity ... isport end 로 구성되는 entity 구문과 Architecture ... of ... is ...begin ...end 로 구성되는 architecture 구문으로 나뉘며 주석문은 “ ... ”로 시작하여 enter 이전까지 한 문장에 걸쳐 유효하다.

시스템	IP주소
서버시스템	205.256.72.84
클라이언트시스템	205.256.72.84

포트번호	사용용도
0~2048	상용화된 인터넷 애플리케이션에서 사용
8000	서버, 클라이언트 프로그램

[그림 3] Entity 구문의 구성

VHDL의 선언문은 alias, attribute, constant, disconnection 등을 사용할 수 있으며 다음 그림은 교통설계 프로젝트에 적용한 결과이다.



[그림 4] 교통망 키트에 적용한 VHDL

vhdl .아키텍처 구문은 회로의 실질적인 동작이나 연결 상태등을 표현하는 것으로 기본적으로는 하나의 entity에 복수개의 Architecture를 사용할 수 있으나 모든 합성 툴에서 이를 지원하지는 않는다. Entity에서 IC의 외형을 표현한다면 Architecture에서는 IC의 내부 동작을 표현한다.

아키텍처 구문에서 Architecture ... of ... is 와 "begin:사이에 존재할 수 있는 선언문은 아키텍처 안에서 사용할 각 종 typerhk 변수 종류를 지정할 석으로 선언문으로 가능한 것은 type,use문이다.

참고문헌

[1] 도철용, 교통공학원론, 청문가, p448-598
 [2] 진현수, 외"퍼지이론을 이용한 도시교통신호 등의 제어에 관한 연구" 대한전자공학회 제어계측연구회 합동학술발표회 논문집, pp93-96, 1991
 [3] 진현수, 외"퍼지로직을 이용한 교통신호등의 최적주기 및 현시제어" 대한전자공학회, 대한전기학회, 한국 통신 학회, 인공지능, 신경망 및 퍼지 시스템 종합합동 학술회의 논문집, 1991
 [4] Borestein, J.and Feng. L., "A New Method for Combinining Data from Gyro and Odometry in Mobile Robots," IEEE Conference on Robotics and Automation, wp6.pp63-68, 1996
 [5] Dirk Schulz, Wolfram Burgard,Dieter Fox, Sebastian Thrun,Armin B.Cremers, "Web Interfaces for Mobile Robots in Public Places", IEEE Robotics and Automation Magazine,

pp48-56, 2000.

[6] 김종환, 한국현, 김용재, 김신, 박귀홍, 이강희, 정준수, 김용덕, "인터넷 기반 퍼스널로봇", pp39-68.
 [7] 진현수, 이상훈, 송진호, 김성환, "AHP를 이용한 퍼지교통신호기 설계", 한국퍼지및 지능시스템 학회 논문지, 제10권1호, pp79-83,2000.4