실험 3 PLD/FPGA를 이용한 조합논리회로 설계 I

실험 목표

FPGA(Field Programmable Gate Array)는 임의의 논리회로를 프로그램 할 수 있는 내부 구조를 가진 범용 IC이다. FPGA 소자의 동작을 이해하고 'Quartus Ⅱ' software와 FPGA 소자를 사용하여 반가산기와 반감산기 및 패리티 발생기 등의 구현을 통해 이들의 동작을 이해하고, 회로를 직접 설계해 본다.

실험 장치 및 부품

FPGA 실험 보드(EP4CE6)
USB Blaster Ⅱ
Quartus Ⅱ

관련 이론

1. PLD 소자의 분류

가. Mask ROM(read only memory)

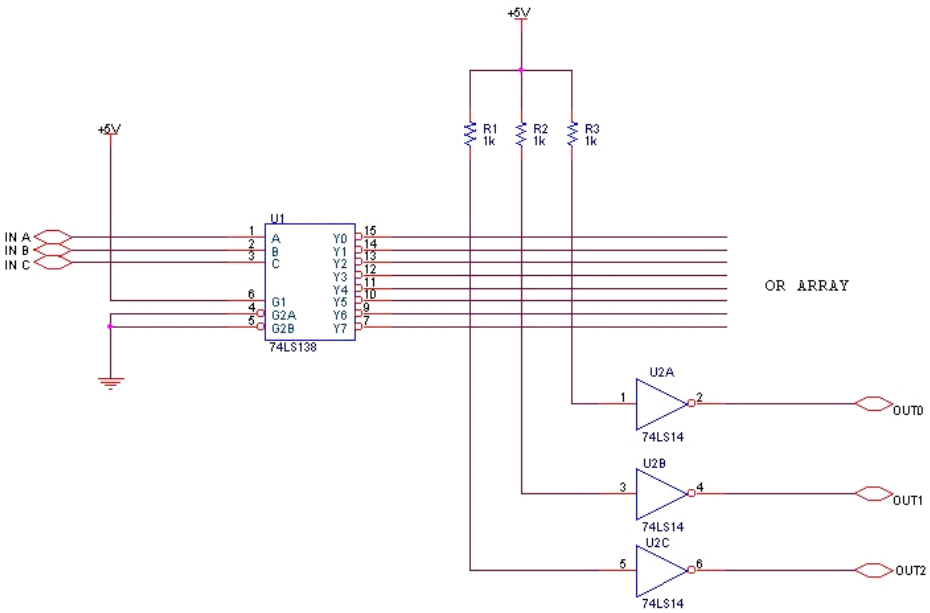


그림 3-1. Mask ROM의 일반적인 구조

Mask ROM의 일반적인 구조는 그림 4-1과 같으며, 이는 AND-OR 어레이(array) 구조에서 고정된 AND 어레이 및 고정된 OR 어레이로 구성되어 있다. 또한 Mask ROM은 조합논리회로를 대량으로 생산할 경우에 사용된다.

나. PLA(programmable logic array)

PLA의 일반적인 구조는 그림 3-2와 같으며 이는 가변의 AND 어레이 및 가변의 OR 어레이로 구성되어 있다.

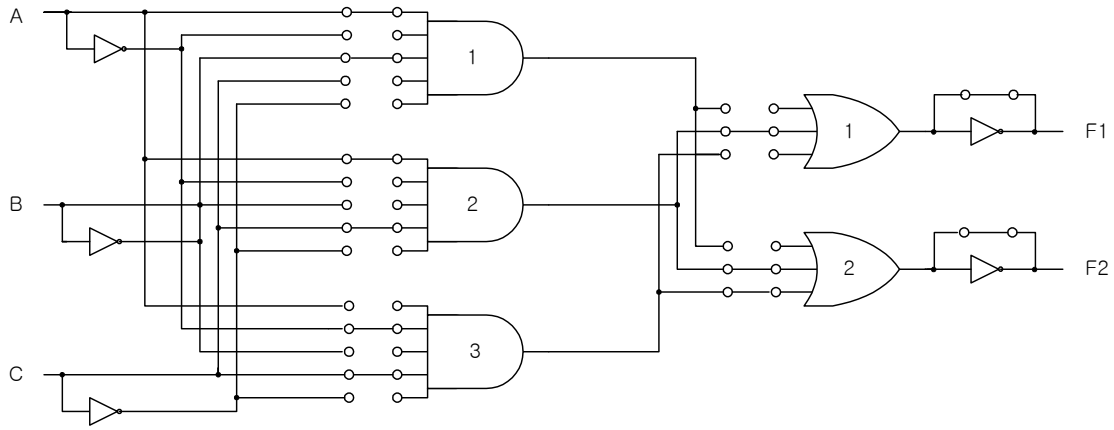


그림 3-2. PLA의 일반적인 구조

다. PAL(programmable array logic)

PAL의 일반적인 구조는 그림 3-3과 같으며 이는 가변의 AND 어레이 및 고정된 OR 어레이로 구성되어 있다.

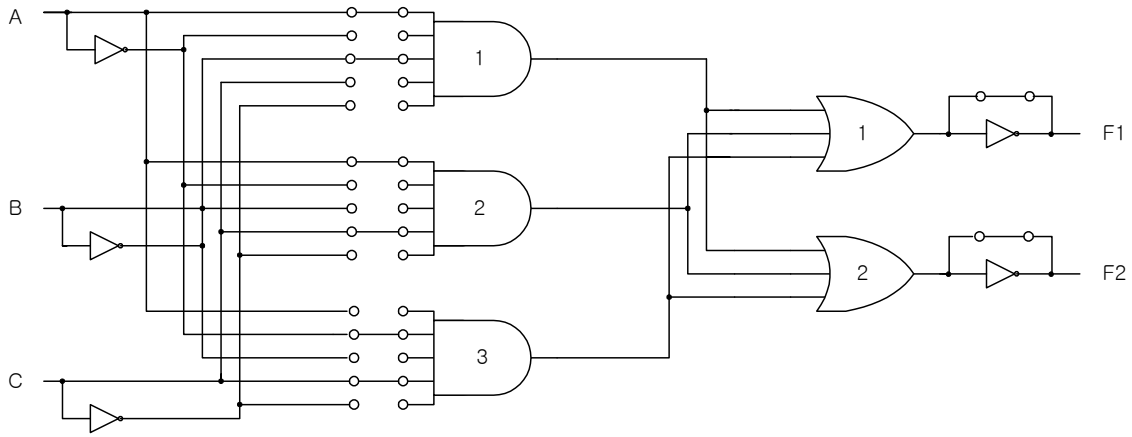


그림 3-3. PAL의 일반적인 구조

이러한 PAL은 1978년 MMI(Monolithic Memory Inc.)사에서 최초로 개발된 이래 현재까지 많이 사용되고 있다. PAL소자는 PLA소자에 비하여 논리회로 설계가 용이하며 가격이 저렴하고 고속이며 소비 전력이 적고 또한 PC상에서 논리회로의 Logic을 쉽게 프로그램할 수 있다는 장점이 있다. 반면에 OR 어레이가 고정이므로 설계 자유도가 낮다는 단점을 갖고 있으나, 여러 종류의 패밀리 소자를 준비하여 이러한 결점을 보충하고 있다.

2. 조합논리 회로 설계 순서

가. 주어진 문제에 대해 이용할 수 있는 입력 변수와 요구되는 출력 변수의 수를 결정하고, 변수명을 할당한다.
나. 입력과 출력 사이의 진리치표를 구한다.
다. Boole 대수와 Karnaugh maps, tabulation method 등을 이용하여 간략화 한다.
라. 회로도를 그린다.

3. 반가산기

기본적인 산술적 연산으로 2개의 2진수 숫자를 더하는 기능을 수행한다.

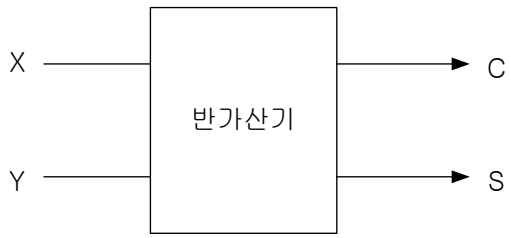


그림 3-4. 반가산기

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

*C : carry S : sum

표 3-1. 반가산기의 진리치표

4. 반감산기

두 비트를 빼서 그들의 차를 만드는 조합논리 회로로서, 두 수의 차를 나타내는 출력 D와 1을 빌어 왔는지를 나타내는 출력 B를 갖고 있다.

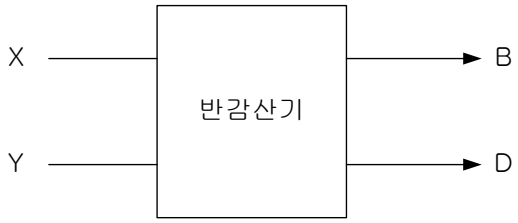


그림 3-2. 반감산기

X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

*B : borrow, D: difference

표 3-2. 반감산기의 진리치표

5. 패리티(parity)

패리티는 데이터 통신에서 데이터를 주고받을 때나 데이터 저장 장치에 저장된 데이터를 읽어낼 때 오류를 검출할 목적으로 사용되며, 홀수 패리티와 짝수 패리티가 있다.

패리티 비트는 1의 개수를 홀수 또는 짝수로 만들기 위해 2진식 정보에 포함시키는 특별한 비트로서, 패리티 비트를 포함한 정보가 수신측에 전송되어 오류 검출을 위해 사용된다. 검사된 패리티가 전송된 것과 일치하지 않으면 오류가 검출된다. 송신측에서 패리티 비트를 만들어내는 회로를 패리티 발생기, 수신측에서 검사하는 회로를 패리티 검사기라 한다.

짝수 패리티는 패리티 비트를 제외하고 1의 개수가 짝수이면 패리티 비트가 0이 되고, 홀수이면 1이 된다. 홀수 패리티는 짝수 패리티와는 반대로 동작한다.

Three-bit message			Parity bit
X	Y	Z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

표 3-3. 짝수 패리티 발생기에 대한 진리치표

6. 기본적인 VHDL 단위

VHDL 코드는 최소한 다음 세 가지 단위로 구성된다.

· LIBRARY 선언 : 설계에 사용되는 모든 라이브러리 목록을 포함 시킨다.

· ENTITY : 회로의 입출력 핀을 명시한다.

· ARCHITECTURE : 회로가 어떻게 동작(기능)해야 하는지를 기술한 VHDL 코드 부분.

1) 라이브러리 선언

LIBRARY를 선언하기 위해서는 두 줄의 코드가 필요하다. 하나는 라이브러리의 이름을 포함하고, 또 다른 하나는 use 절이다.

예)

```
LIBRARY library_name;
USE library_name.package_name.package_parts;
```

일반적으로, 설계할 때는 서로 다른 세 개의 라이브러리를 출처로 하는 최소한 3개의 패키지가 필요하다.

예)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY std;
USE std.standard.all;

LIBRARY work;
USE work.all;
```

위의 std와 work 라이브러리는 Default로 볼 수 있어 굳이 선언할 필요하가 없지만, ieee 라이브러리는 명확하게 기술해주어야 한다. ieee 라이브러리는 STD_LOGIC (또는 STD_ULOGIC) 데이터 타입이 설계에 사용되는 경우에만 필요하다.

3. ENTITY

ENTITY는 회로의 모든 입력 및 출력 핀(PORT)을 명시해 놓은 리스트이다.

예)

```
ENTITY entity_name IS
    PORT (
        port_name : signal_mode signal_type;
        port_name : signal_mode signal_type;
        ....);
END entity_name;
```

신호의 모드는 IN, OUT, INOUT 또는 BUFFER가 될 수 있다. IN, OUT은 단방향 핀이며,

INOUT은 양방향 핀이다. BUFFER는 출력 신호가 내부적으로 사용되어야 할 경우에 사용된다.
신호 타입에는 BIT, STD_LOGIC, INTERGER 등이 있다.
ENTITY의 이름은 기본적으로 어떤 이름으로 지정할 수 있는데, 다만 VHDL 예약어는 제외된다.

예)

```
ENTITY nand_gate IS
        PORT ( a,b : IN BIT;
                x   : OUT BIT);
END nand_gate;
```

4. ARCHITECTURE

ARCHITECTURE는 회로가 어떻게 동작해야 하는지를 기술한다. 이에 대한 구문은 다음과 같다.
예)

```
ARCHITECTURE architecture_name OF entity_name IS
        [declarations]
BEGIN
        (code)
END architecture_name;
```

위에서 보는 바와 같이, 아키텍처는 두 개의 부분, 즉 신호와 상수가 선언되는 선언부와 코드부로 구성된다. ENTITY의 경우처럼 ARCHITECTURE의 이름도 ENTITY와 같은 이름을 포함하여 기본적으로 어떤 이름이라도 될 수 있다. 다만 VHDL 예약어는 제외된다.
* VHDL은 대소문자를 구별하지 않는다.

실험 순서

실험 A 논리회로의 설계

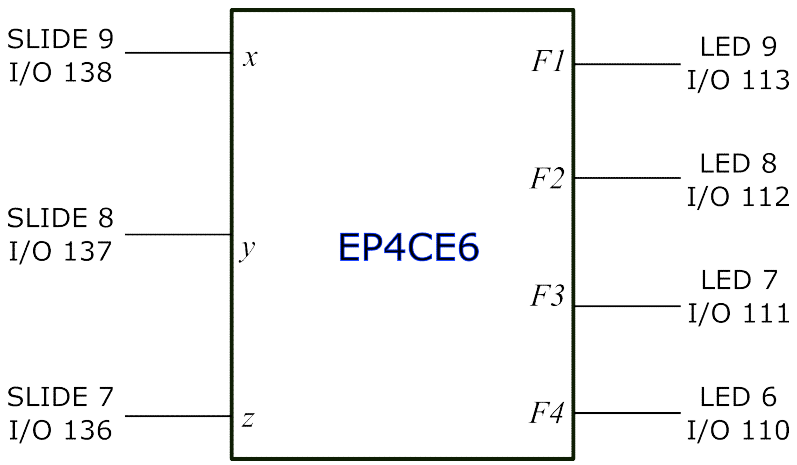
다음과 같은 네 개의 출력을 갖는 하나의 논리회로에 대해 아래에 지시한대로 설계, simulation, 구현 및 시험하시오.

1) $F1(x,y,z) = \sum m(0,1,5,7)$
2) $F2(x,y,z) = \sum m(1,2,3,6,7)$
3) $F3(x,y,z) = \sum m(3,5,6,7)$
4) $F4(x,y,z) = \sum m(0,2,3,4,6)$

(1) Logisim을 사용하여 회로도를 그리고 시뮬레이션 하시오. (간략화 하지 마시오.)

(2) 각각의 출력에 관한 논리식을 간략화한 후, Logisim을 사용하여 시뮬레이션 하시오.

(3) 'Quartus Ⅱ'를 이용하여
1) VHDL code를 작성하고
2) ModelSim을 통해 simulation을 한 후,
3) FPGA 소자에 프로그램하고 실제 동작을 테스트 하시오.



실험 B FPGA 소자를 이용한 전가산기

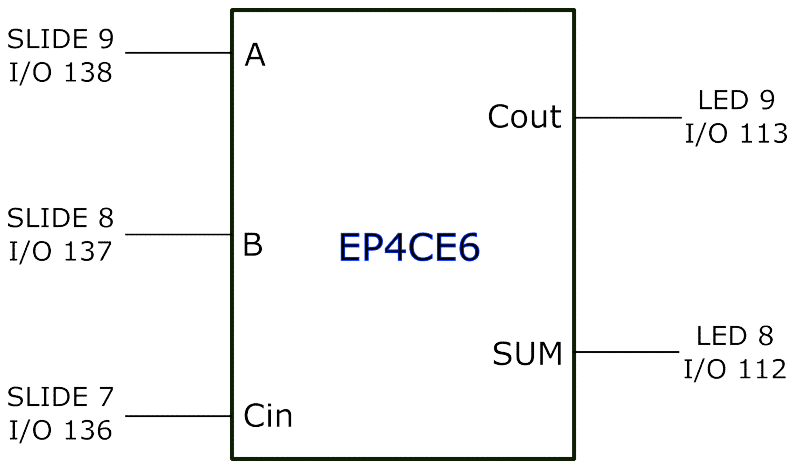
전가산기는 가산기의 일종으로 두 개의 2진수 이외에 하위 자리의 덧셈과정에서 발생한 자리올림수도 고려하여 덧셈을 하는 조합논리회로이다.

(1) 전가산기에 대한 입력 진리치표를 작성하시오.

입력			출력	
A	B	Cin	Cout	SUM

(2) 'Quartus Ⅱ'를 이용하여

1) VHDL code를 작성하고
2) ModelSim을 통해 simulation을 한 후,
3) FPGA 소자에 프로그램하고 실제 동작을 테스트 하시오.



실험 보고서

데이터 및 관찰 내용

실험 A 논리회로의 설계

(1) Logisim을 사용하여 작성한 회로도 및 8개의 입력 조합 각각에 대한 Simulation 결과

(2) 간략화 이전의 회로와 간략화한 회로의 비교
  1) 간략화한 논리식을 Logisim으로 작성한 논리회로
  2) 간략화한 논리 회로의 8개의 입력 조합 각각에 대한 Simulation 결과
  3) 이 결과를 위 (1)번 실험 결과와 표를 이용하여 비교.

실험 B FPGA 소자를 이용한 전가산기

(1) 진리치표

입력			출력	
A	B	Cin	Cout	SUM

(2) 'Quartus Ⅱ'를 이용하여 구현한 결과
  1) VHDL code
  2) ModelSim을 통해 simulation 한 결과
  3) FPGA 소자에 프로그램하고 실제 동작을 테스트 한 결과

결과 및 결론