6주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211533 이름: 남정연

**1.**

1. 가산기 및 감산기의 개념을 이해한다.

2. 부호 변환기의 개념을 이해한다.

3. Verilog를 이용하여 가산기 및 감산기와 부호 변환기를 구현해보고 FPGA를 이용하여 회로의 정확성을 확인한다.

**2.**

반 가산기는 입력 a, b에 대해 합 s와 올라가는 수 c를 출력하며, s = a xor b의 관계가, c = ab의 관계가 있다. 따라서 아래와 같은 시뮬레이션 결과 및 진리표가 나타난다.

Timeline

Description automatically generated with low confidence

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | s | c |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

전 가산기는 반 가산기에서 이전 단계의 carry인 c0가 추가된 형태로, s = a xor b xor c0의 관계를, c = ab + (a xor b)c0의 관계를 가진다. 따라서 아래와 같은 시뮬레이션 결과 및 진리표가 나타난다.

Graphical user interface

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c0 | s | c |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**3.**

반 감산기는 입력 값 x와 y에 대하여 차 d와 빌려와야 하는 수 b를 출력하며 d = x xor y, b= x’y의 관계가 있다. 따라서 아래와 같은 시뮬레이션 결과 및 진리표가 나타난다.

Graphical user interface

Description automatically generated with medium confidence

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | d | b |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |

전 감산기는 반 감산기에서 이전 단계의 빌려와야 하는 수 b0가 추가된 형태로 d = x xor y xor b0, b = x’y + (x xor y)’b0의 관계를 가진다. 따라서 아래와 같은 시뮬레이션 결과 및 진리표가 나타난다.

Graphical user interface, chart

Description automatically generated with medium confidence

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | B\_i-1 | D | B |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**4.**

8421(BCD)-2421 부호 변환기의 진리표는 BCD 부호를 abcd, 2421부호를 wxyz로 나타낸다고 했을 때 아래와 같다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | w | x | y | Z |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

각 입력 a, b, c, d에 대하여 출력 w, x, y, z에 대한 카르노맵을 각각 그려보면 아래와 같다(같은 색상으로 둘러싸인 영역은 같은 그룹이다).

w)

minfied form = ((bd)’(bc)’a’)’ (SOP의 이중 부정)

SOP = = bd+bc+a

Table

Description automatically generated with medium confidence

POS = (a+b)(b’+c+d)

Table

Description automatically generated with medium confidence

x.

minfied form = ((bd’)’(bc)’a’)’ (SOP의 이중 부정)

SOP = bd’+bc+a

A picture containing table

Description automatically generated

POS = (a+b)(b’+c+d’)

A picture containing chart

Description automatically generated

y.

minfied form = ((b’c)’(bc’d)’a’)’ (SOP의 이중 부정)

SOP = b’c+bc’d+a

Table

Description automatically generated

POS = (a+b+c)(b’+c’)(a+c+d)

A picture containing chart

Description automatically generated

z.

minfied form = d = SOP

SOP = d

Table

Description automatically generated with medium confidence

POS = d’

Table

Description automatically generated

**5.**

반 가산기, 전 가산기, 반 감산기, 전 감산기의 모든 입력 값에 대하여 올바른 출력 값이 나타났으며 BCD-8421 코드 변환기 또한 제대로 작동하였다.

**6.**

Carry Select Adder는 이전 단계의 Carry 입력에 의존하는 Adder의 단점을 보완하기 위하여 이전 단계의 Carry를 각각 0과 1로 가정한 계산 결과를 미리 저장해 둔 뒤 실제 입력 받은 Carry에 따라 계산하는 회로이다. 이 또한 여러 Carry Select Adder를 연결하여 Linear Carry Select Adder를 구성할 수 있다.