10주차 결과보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20211547 이름: 신지원

1.

4bit Binary Parallel Adder 는 4-Bit 이진수의 덧셈을 연산하도록 만들어진 가산기로 올림수를 포함하여 연산한다. 따라서 bit 를 a,b의 배열로 표현하였으며, 각각의 올림을 cn으로 표현하여 코드를 구현하였다.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

가장 처음 a가 2이고, b가 f 를 띄고 있다. (F는 15로 배열이 0~3인 이 배열에서는 3을 의미한다.) 2와 3을 더하면 5인데, sum 은 1을 가리키며 C4에 신호가 1인 것으로 보아 올림수가 나타난 것을 알 수 있다. 5는 3+2와 같기 때문에 올림수를 의미하는 C4에 신호가 가고, sum의 값은 0다음 1을 가리키는 것이다. 따라서 위에서 구현한 Verilog source 코드가 잘 작동하는 것을 알 수 있다.

**2.**

4bit Binary Parallel Substraction 는 4-Bit 이진수의 뺄셈을 연산하도록 만들어진 감산기로 내림수를 포함하여 연산한다. 따라서 bit 를 a,b의 배열로 표현하였으며, 각각의 내림을 dn으로 표현하여 코드를 구현하였다.

텍스트, 번호, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

가장 처음 a가 8이고 b가 4일 때, 둘의 차는 4여야 하며 시뮬레이션 상에서도 옳게 구현된 것을 알 수 있다. 그렇다면 처음 내림수가 발생하는 (C4에 신호가 가해지는) 곳을 살펴보면, a는 2, b는 f가 나타난다. 여기서 f는 3과 같으며 따라서 2에서 3을 빼기 때문에 내림수가 발생해야 한다. 따라서 d의 값은 3이 나타날 수밖에 없으며 simulation 상에서도 옳게 작동하는 것을 볼 수 있다.

**3.**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

4bit Binary Parallel Adder의 동작이 끝나면 [0:3]s와 C를 얻을 수 있고, 이 값들을 이용하여 위 연산을 수행한다. 위 연산을 통해 [0:3]sum 와 Cout을 얻을 수 있다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

시뮬레이션을 자세히 살펴보자, 맨 처음 a가 5, b가 0일 때, 올림이 발생하지 않았고 결과는 5로 도츨되었다. 그 다음 a가 6, b가 f(15)일 때, 결과는 b(11)이며 C4에 1이 도출되어 올림수가 발생한 것을 알 수 있다. 이처럼 위에서 구현한 코드를 돌려보았을 때 simulation 상에서 BCD연산이 원활하게 작동하는 것을 볼 수 있다.

**4.**

4bit Binary Parallel Adder, 4bit Binary Parallel Subtractor, BCD Adder 의 코드를 직접 짜보고 Simulation 을 돌려보며 결과를 확인하는 과정을 통해 bit 가 하나하나 연산되는 과정을 구체적으로 볼 수 있었다. 실습에서는 가산기와 감산기를 따로 구현하였지만, 예비보고서에서 작성하였듯이 둘을 하나에 구현할 수 있다는 사실을 늦게 떠올려 본 실습에서는 구현하지 못하였다.

또한 BCD는 카르노맵이 복잡하여 논리식을 구현하는데 어려움이 있었다.

**5.**

BCD Adder 에 대해 알아보았다면, BCD Substraction 에 대해 알아보고자 한다. BCD Substraction 은 1의 보수, 9의 보수, 10의 보수를 이용하는 등 다양한 방법이 있다. 알아본 방법은 1의 보수를 이용하는 방법으로 carry를 올려 계산하는 방법이다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명



텍스트, 폰트, 타이포그래피, 디자인이(가) 표시된 사진

자동 생성된 설명