10주차 결과보고서

전공 : 컴퓨터공학과 학년 : 3학년 학번 : 20211558 이름 : 윤준서

**1. 4-bit Binary Parallel Adder**

4-bit Binary Parallel Adder의 source code와 simulation code는 다음과 같다.

텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명텍스트, 영수증, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

두 코드로 Verilog를 통해 얻은 Schematic과 Simulation 결과는 다음과 같다.

텍스트, 도표이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

자동 생성된 설명

과정 : 4-bit Binary Parallel Adder는 1-bit Full Adder 4개를 직렬 연결한 형태다. 각 bit마다 덧셈 연산을 실행한 후 sum과 carry를 도출한다. 그리고 carry를 다음 bit 덧셈에 함께 계산을 해준다. 그럼 각 자리의 sum이 정확하게 계산된다. 마지막 carry는 overflow로 더 이상 계산에 이용하지는 않는다. Verilog에서 변수를 사용할 때 입력과 출력 변수 모두 배열을 사용하였다.

**2. 4-bit Binary Parallel Subtractor**

4-bit Binary Parallel Subtractor의 source code와 simulation code는 다음과 같다.

텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명텍스트, 영수증, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

두 코드로 Verilog를 통해 얻은 Schematic과 Simulation 결과는 다음과 같다.

텍스트, 도표이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 회로이(가) 표시된 사진

자동 생성된 설명

과정 : 4-bit Binary Parallel Subtractor는 1-bit Full Subtractor 4개를 직렬 연결한 형태다. 각 bit마다 뺄셈 연산을 실행한 후 difference와 borrow를 도출한다. 그리고 borrow를 다음 bit 뻴셈에 함께 계산을 해준다. 그럼 각 자리의 differencer가 정확하게 계산된다. 마지막 borrow는 overflow로 더 이상 계산에 이용하지는 않는다. Verilog에서 변수를 사용할 때 입력과 출력 변수 모두 배열을 사용하였다.

**3. BCD Adder**

BCD Adder의 source code와 simulation code는 다음과 같다.

텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명텍스트, 영수증, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

두 코드로 Verilog를 통해 얻은 Schematic과 Simulation 결과는 다음과 같다.

도표, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 라인이(가) 표시된 사진

자동 생성된 설명

과정 : BCD Adder는 4-bit Adder의 계산 결과를 BCD code의 형태로 나타내는 회로다. 더한 값이 10 이상일 경우 BCD code의 범위를 벗어나기 때문에 결과 값에 6(0110)을 더해준다. 이 과정은 10진수의 각 자리 수를 BCD code로 나타내는 것과 같다. 예를 들어 계산 결과 값이 12일 때, BCD Adder의 결과는 0001 0010인 것이다. 위의 BCD Adder는 4-bit만을 다루기에 십의 자리 수 부분의 값이 오직 0 또는 1이다. 따라서 십의 자리 수를 판단하는 변수 check를 이용한다. 계산 결과 값이 1010 - 1111 사이인 경우, check는 1의 값을 가진다. Verilog에서 변수를 사용할 때 입력과 출력 변수 모두 배열을 사용하였다.

**4. 결과 검토 및 논의 사항**

4-bit Binary Parallel Subtractor의 simulation 결과를 보면 bin의 값이 1이고 나머지 입력 값이 모두 0일 때, 모든 출력 값이 1임을 확인할 수 있다. 이는 2's complement와 관련이 있는데, bin만 값이 1이면 뺄셈 결과 값은 -1이다. 이 때 -1을 이진수로 표현할 때 2's complement를 사용하는데, -1를 4-bit 형태로 나타내면 1111이다. 따라서 difference의 모든 bit의 값이 1로 출력된다.

BCD Adder의 십의 자리 수를 판단하는 check 변수의 값은 다음의 두 가지로 판별이 가능하다.

1. 가장 마지막 carry 값이 1인 경우

2. 전체 sum의 값이 101X 또는 11XX 인 경우

**5. 추가 이론**

Verilog를 통해 여러 bit 값을 가지는 변수를 다룰 때, 배열을 사용하면 코드와 simulation 결과가 간결해진다.

source code에서 입력 변수의 이름이 X이고, 4-bit인 경우 **input [3:0] X**, 로 선언이 가능하다.

출력 변수를 assign할 때 X[index]의 형태로 사용할 수 있다.

simulation code에서 X의 모든 bit의 초기값을 0으로 정할 때, 다음과 같이 선언할 수 있다.

**initial begin**

**X = 4'D0;**

**end**

X의 내부 bit의 값을 하나씩 바꿀 때, 시간에 따라 하나씩 값을 바꾸는 과정은 다음과 같다.

**always@(X) begin**

**X <= #(time) X + 4'D1;**

**end**

simulation 결과는 0과 1을 구분한 pulse의 형태가 아닌, 구간마다 4-bit 전체를 묶어서 하나의 값으로 표시한다. 즉 임의의 구간에서 X의 모든 bit 값이 1010일 때, simulation에선 X의 값을 16진수인 a로 출력한다.