

## 실험 4. 이진수 연산

2022. 04. 22.

디지털 시스템 설계 (CSED273)

### 1. 개요

이진수 덧셈에 사용하는 반가산기와 전가산기의 기능을 이해하고 회로를 구성한 후, 이를 사용해 덧셈, 뺄셈, 곱셈 회로를 구성한다. 자세한 학습 목표는 다음과 같다.

- 반가산기와 전가산기 구현
- 5-bit 리플 가산기/감산기 구현
- 5x3 이진 곱셈기 구현
- 테스트 벤치를 이용한 회로 검증

### 2. 이론적 배경

#### 1) 반가산기 (Half adder)

반가산기는 1-bit 이진수 두 개를 입력받아 합과 Carry (올림) out을 출력한다.

#### 2) 전가산기 (Full adder)

전가산기는 1-bit 이진수 두 개와 이전 가산기의 Carry in을 입력받아 합과 Carry out을 출력한다. 두 개의 반가산기를 사용해 구현할 수 있다.

#### 3) N-bit 리플 가산기 / 감산기 (N-bit Ripple Adder / Subtractor)

N-bit 가산기는 N-bit 이진수 두 개를 더하여 N-bit 덧셈 결과와 Carry out을 출력한다. N-bit 가산기는 다양한 방법으로 구현할 수 있는데, 그중 하나로 N-bit 리플 가산기가 있다.

N-bit 리플 가산기는 N개의 전가산기를 순차적으로 이어 구현한다. N개의 전가산기는 각각 N-bit 중 한 자릿수의 연산을 맡는다. 이때 k번째 자릿수를 계산하는 전가산기의 Carry out이 k+1번째 자릿수를 담당하는 전가산기의 Carry in에 연결되어 가장 낮은 자릿수부터 가장 높은 자릿수까지 Carry가 순차적으로 전파된다. 이러한 순차적인 연결 구조로 인해 낮은 자릿수의 연산이 끝나 Carry out이 결정되어야 높은 자릿수의 연산을 시작할 수 있다. 즉, 연산코자 하는 자릿수가 늘어남에 따라 연산에 걸리는 시간도 비례해서 늘어난다.

한편,  $A-B$  꼴의 뺄셈은  $A+(-B)$  꼴의 덧셈으로 나타낼 수 있다. 이러한 성질과 2의 보수(2's complement)를 활용하면 N-bit 가산기를 사용해 감산기를 간단히 구현할 수 있다.

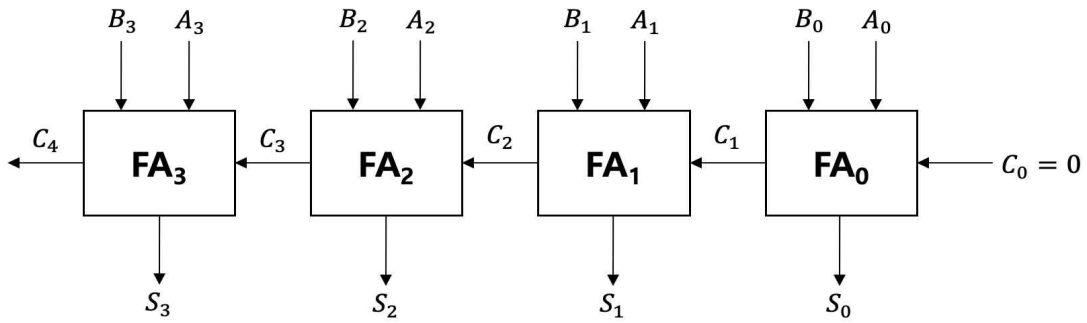


그림 1. 4-bit 리플 가산기.  $FA_{k-1}$ 의 Carry out이  $FA_k$ 의 Carry in에 연결되어있다.

#### 4) MxN 이진 곱셈기 (MxN Binary Multiplier)

MxN 이진 곱셈은 M-bit Multiplicand와 N-bit Multiplier의 각 자릿수의 부분 곱을 통해 얻은 이진수 N개를 합하여 계산할 수 있다. 부분 곱은 bitwise AND 연산을 통해 구현할 수 있으며, 가산기로 부분 곱을 모두 더하여 답을 구한다.

이때, 다음과 같이 각 부분 곱을 더하는 과정을 M-bit 가산기 N-1개로 구할 수 있다.

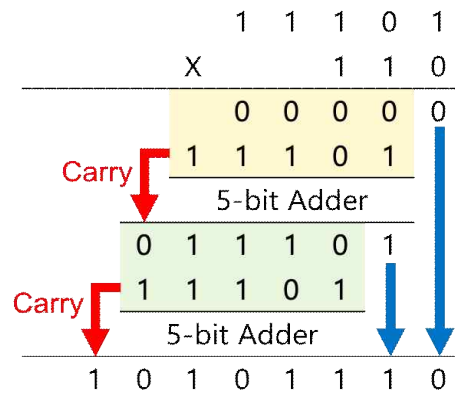


그림 2. 5x3 이진 곱셈 예시

### 3. 실험 준비

- 1) 반가산기의 진리표와 식을 구하고 회로를 그린다.
- 2) 전가산기의 진리표와 식을 구하고 반가산기를 사용해 전가산기 회로를 그린다.
- 3) 전가산기를 사용해 5비트 리플 가산기와 감산기 회로를 그린다.
  - 음수는 2의 보수로 표현한다.
- 4) 5비트 리플 가산기를 사용해 5x3 이진 곱셈기 회로를 그린다.

\* 보고서 필수 내용

- 실험 준비 과정

#### 4. 실험

##### 0) 공통 유의사항

- 주어진 코드에서 */\* Add your code here \*/* 로 주석 처리된 부분만 수정하여 구현한다.
- 모든 입출력은 Little-endian 형식으로 표현한다.

##### \* 보고서 필수 내용

- Schematic 기능으로 생성한 회로도 캡처
- 시뮬레이션 파형 캡처

##### 1) 반가산기, 전가산기 - lab4\_1.v

- ㄱ. 반가산기를 구현하고, 이를 사용해 전가산기를 구현한다.
- ㄴ. Schematic 기능으로 회로를 확인한다.

##### 2) 5비트 리플 가산기 - lab4\_2.v

- ㄱ. 실험 1의 전가산기 모듈을 사용해 5비트 리플 가산기를 구현한다.
- ㄴ. Schematic 기능으로 회로를 확인한다.
- ㄷ. 주어진 테스트벤치로 시뮬레이션을 실행해 정상 작동을 확인한다.

##### 3) 5비트 리플 감산기 - lab4\_3.v

- ㄱ. 실험 2의 5비트 리플 가산기 모듈을 사용해 5비트 리플 감산기를 구현한다.
  - 음수는 2의 보수로 표현한다.
- ㄴ. Schematic 기능으로 회로를 확인한다.
- ㄷ. 주어진 테스트벤치로 시뮬레이션을 실행해 정상 작동을 확인한다.

##### 4) 5x3 이진 곱셈기 - lab4\_4.v

- ㄱ. 실험 2의 5비트 리플 가산기 모듈을 사용해 5x3 이진 곱셈기를 구현한다.
- ㄴ. Schematic 기능으로 회로를 확인한다.
- ㄷ. 주어진 테스트벤치로 시뮬레이션을 실행해 정상 작동을 확인한다.

## 5. 제출

{학번}\_lab4.zip으로 다음 파일을 압축하여 PLMS로 제출한다.

- lab4\_1.v // 코드
- lab4\_2.v
- lab4\_3.v
- lab4\_4.v
- lab4\_report.pdf // 보고서