컴퓨터 공학 기초 실험2 보고서

실험제목: Subtractor & Arithmetic Logic Unit(ALU)

실험일자: 2021년 10월 04일 (화)

제출일자: 2021년 10월 12일 (수)

학 과: 컴퓨터공학과

담당교수: 공영호 교수님

실습분반: 화요일 0, 1, 2

학 번: 2020202040

성 명: 박민형

1. 제목 및 목적
   1. 제목

Subtractor & Arithmetic Logic Unit(ALU)

* 1. 목적

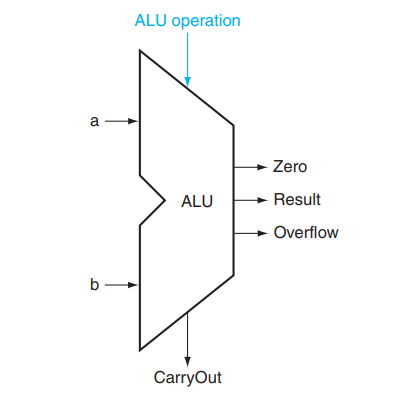
이번 시간에는 컴퓨터 구조에서 어찌 보면 가장 핵심적인 역할을 담당한다고 할 수 있는 산술 논리 연산 장치에 대해서 배우고 설계한다. 산술 논리 연산 장치는 산술 연산과 논리 연산을 수행하는데 우리가 쓰는 컴퓨터의 가장 낮은 레벨에서 연산을 수행하므로 기반이라고 할 수 있다. 컴퓨터의 하드웨어가 지금처럼 발전하기 이전에는 최대한 적은 비트로 최대의 아웃풋을 뽑아내야 했기 때문에 효율성이 중시되었는데 이런 배경 속에서 폰 노이만은 ALU의 가이드라인을 제시하였다. 이번 실험에서 우리는 이 발자취를 따라가 볼 것이다.

1. 원리(배경지식)
   1. Subtractor

Subtractor(감산기)는 Adder와는 달리 감산을 목적으로 사용하는 조합논리 회로이다. 조합 논리 회로에서 Adder와 마찬가지로 고유의 Carry를 수식적으로 처리해서 만들 수 있는 Half Subtractor(반감산기), Full Subtractor(전감산기)가 있으나, 본 실험에서는 Adder의 입출ㄹALU에서

* 1. ALU

Arithmetic Logic Unit(ALU, 산술 논리 연산 장치)는 산술 연산과 논리 연산을 처리하는 프로세서의 일부이다. ALU는 명령어에 따라 산술 연산(사칙연산, 증가, 감소, 보수)이나 논리 연산(AND, OR, NOT, XOR, 시프트) 중 하나를 실행한다. ALU를 그림으로 나타내면 다음과 같다.



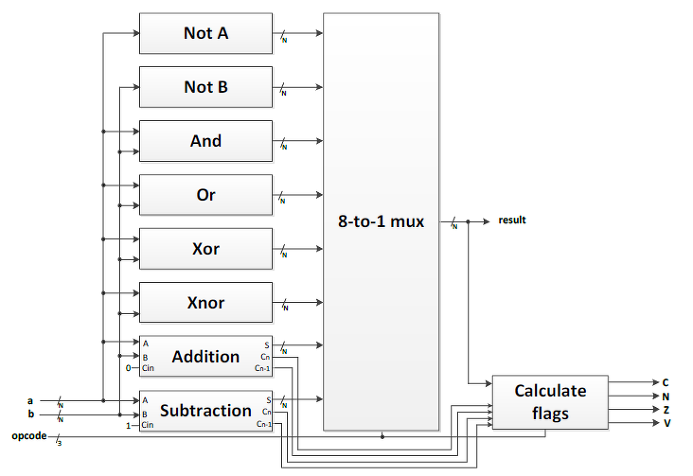
본 실험에서는 그림의 ALU operation에 대응하는 opcode라는 3bit code를 따라서 여러가지 연산을 할 것이다. Opcode는 다음과 같다.

|  |  |
| --- | --- |
| 3’b000 | NOT A |
| 3’b001 | NOT B |
| 3’b010 | AND |
| 3’b011 | OR |
| 3’b100 | XOR |
| 3’b101 | XNOR |
| 3’b110 | Addition |
| 3’b111 | Subtraction |

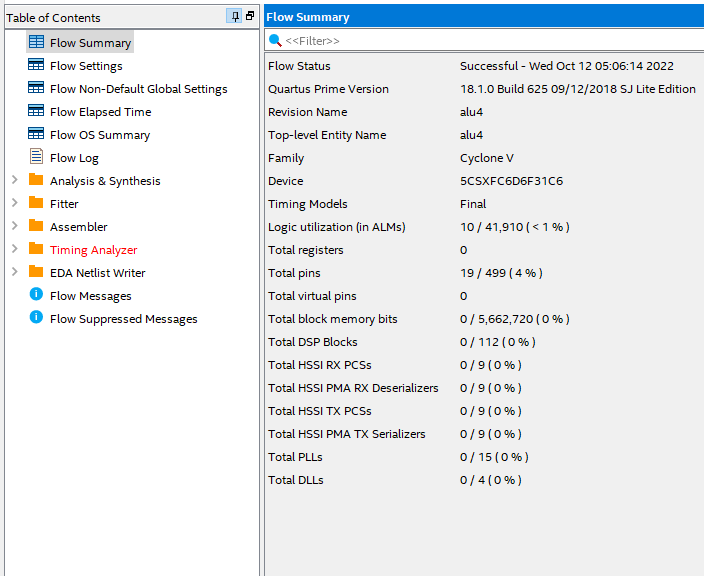
* 1. Carry와 Overflow
  2. Blocking과 non-blocking

1. 설계 세부사항

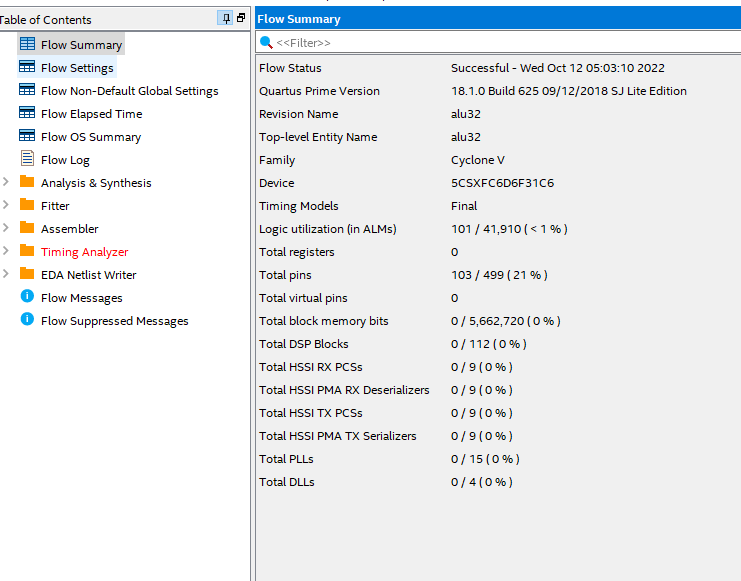
ALU의 Schematic은 다음과 같다.



1. 설계 검증 및 실험 결과

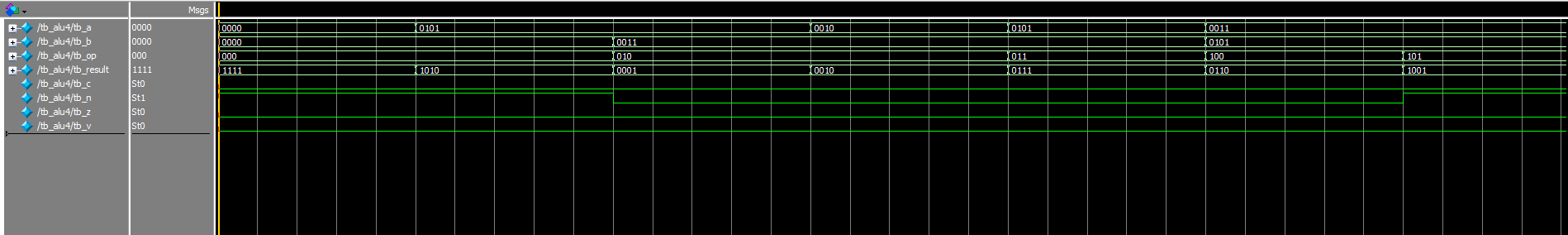


ALU 4bits의 Flow Summary

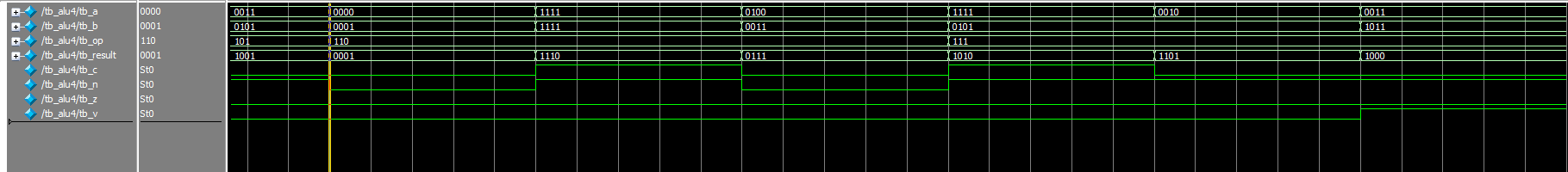


ALU 32bits의 Flow Summary

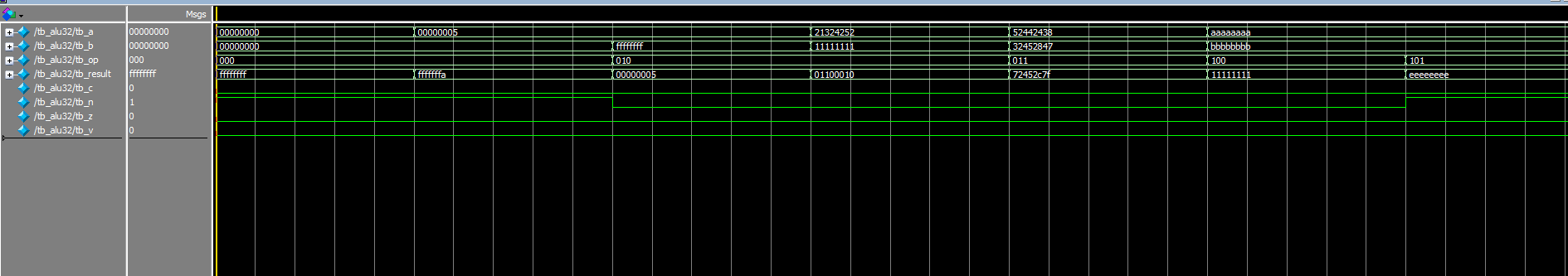
* 1. 시뮬레이션 결과



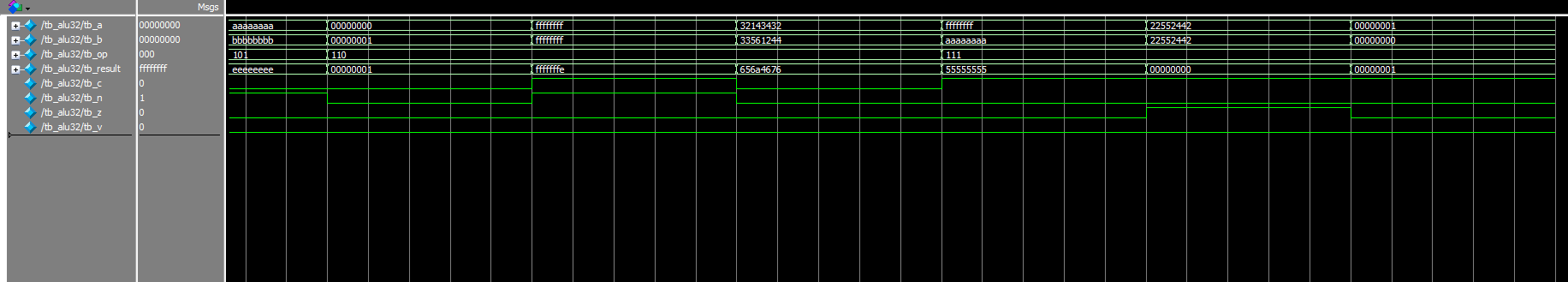
ALU 4bits의 Logic Module이다.



ALU 4bits의 Arithmetic Logic Module이다.

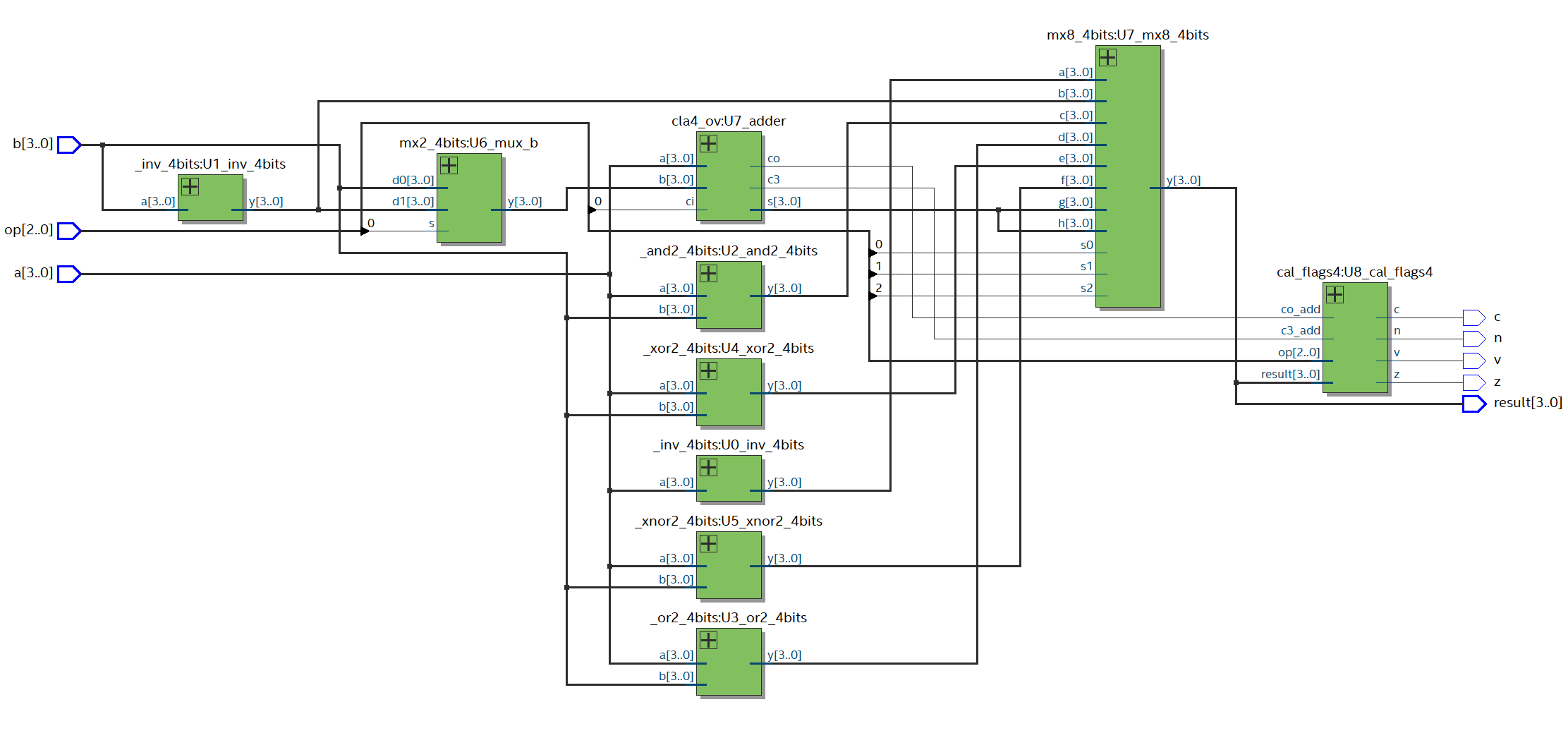


ALU 32bits의 Logic Module이다.

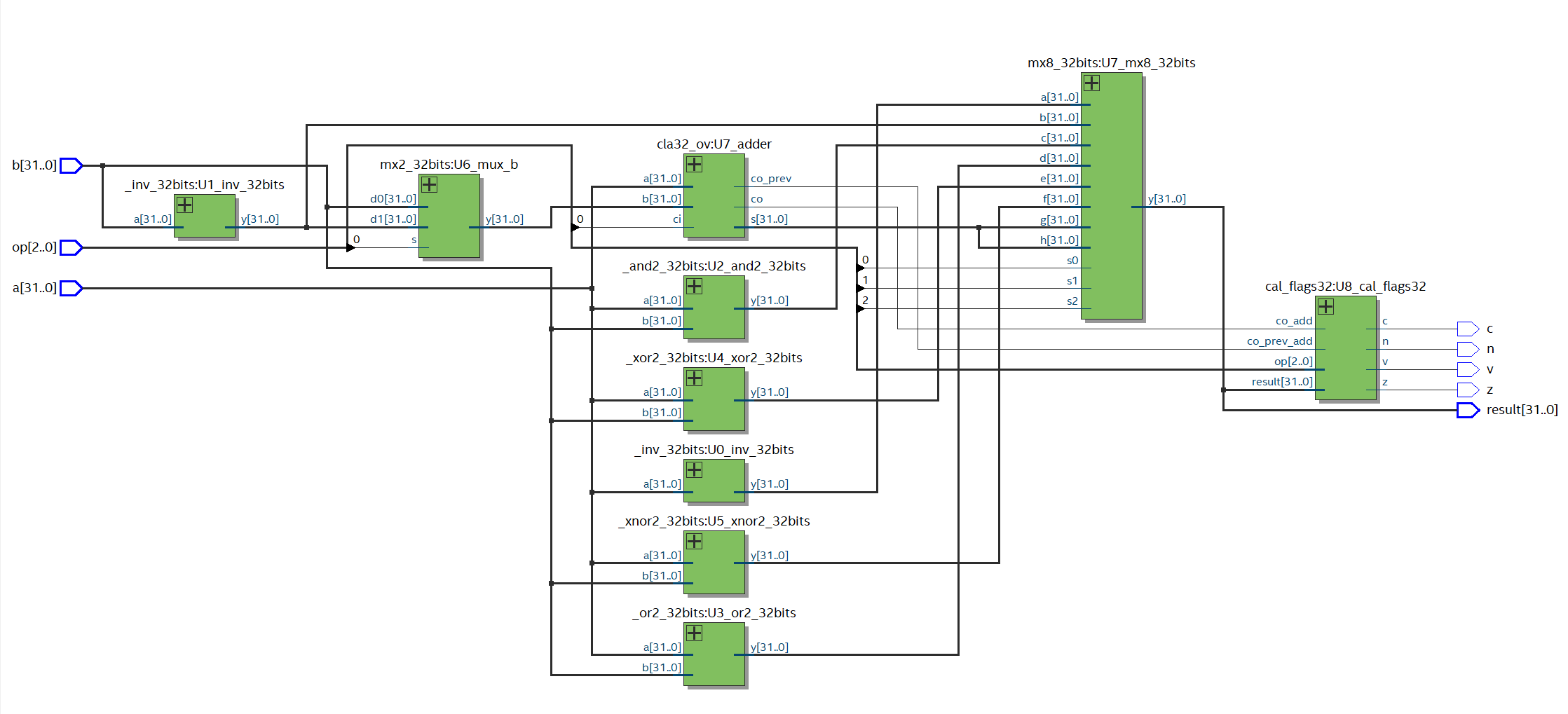


ALU 32bits의 Arithmetic Logic Module이다.

* 1. 합성(synthesis) 결과



ALU 4bits



ALU 32bits

1. 고찰 및 결론
   1. 고찰

Logic Module과 Arithmetic Module 두 모듈은 논리 연산과 산술 연산이라는 다른 연산과정을 거치기 때문에 어떻게 ALU를 설계해야 하나 의문이었다. 그런 와중에 주어진 opcode를 보고 이는 조건문을 사용해야하겠다고 생각했더니 과제 자료에서 조건문을 제시하였다. If문 비슷하게 짜긴 했지만, C언어에 있는 case문으로 짜도 훌륭한 결과물이 나올 것으로 예상된다.

* 1. 결론

사실 ALU를 직접 설계해보기 전까지는 MUX가 왜 필요한지에 대해서 잘 몰랐었다. 그런데 ALU를 설계해보고 나니 왜 MUX를 사용하는지 분명하게 알게 되었다. Verilog가 사용하는 벡터가 C언어의 배열과 유사하면서도 약간의 차이가 있는 것을 이번 실험을 통해서 알게 되었다.

1. 참고문헌

해당 과제를 수행하며 참고한 서적/강의자료/웹의 출처를 작성하여 줍니다.

서적의 경우에는 **저자/제목/출판사/출판년도**의 순으로 작성한다.

강의자료의 경우에는 **강사/강의제목/강의장소(학교명(과명))/강의년도**의 순으로 작성한다.

웹의 경우에는 **주제/URL**의 순으로 작성한다.

또한, 문헌이 아닌 다른 사람의 도움을 많이 받았을 경우에는 ‘어떠한 부분은 누구의 도움을 받아 수행하였다’는 내용을 명시한다.