**디지털회로실험 보고서**

-최종 프로젝트-

전자공학과

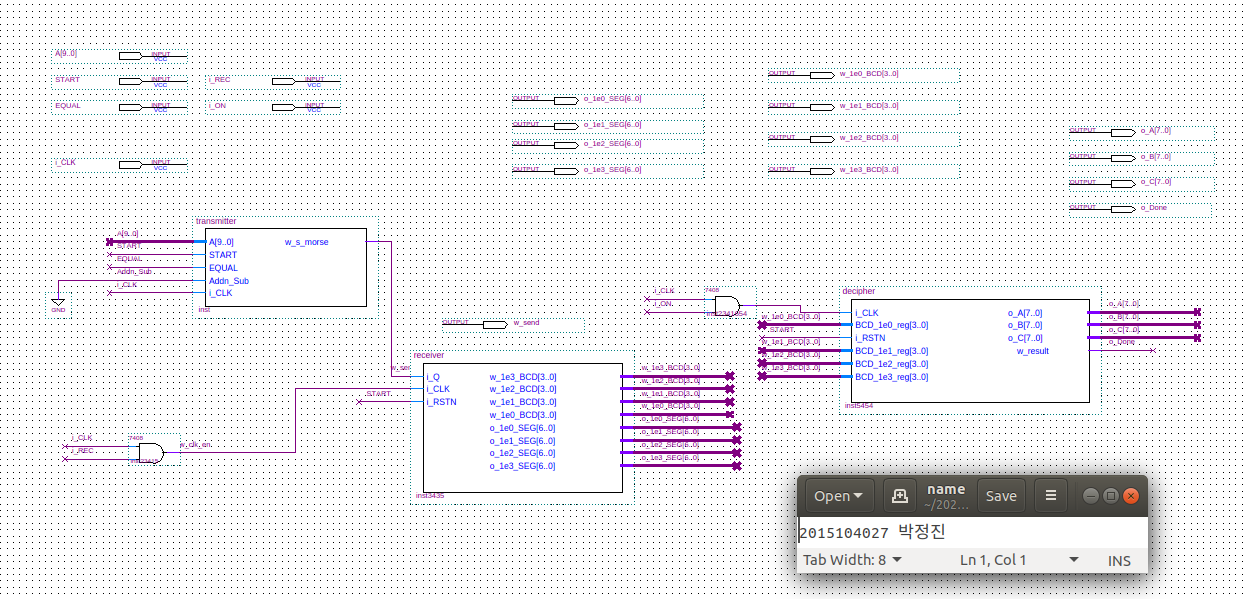
2015104027

박정진

**실험 결과**

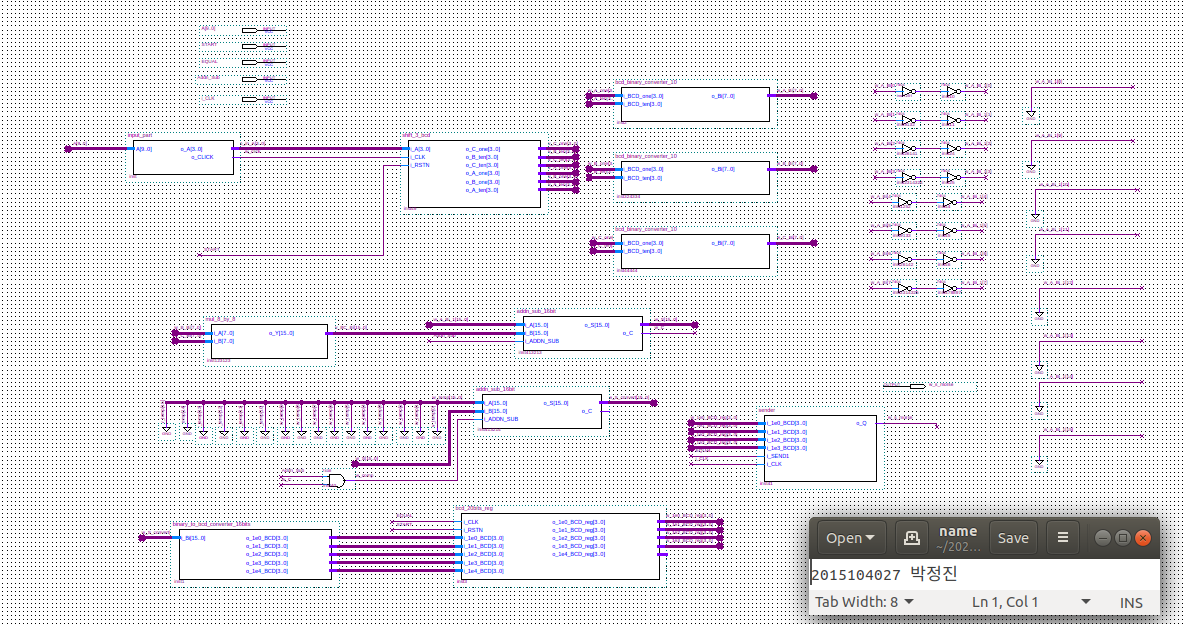
Quartus Simulation (top.bdf)

**회로는 모두 74series로 구현하였다. (사용하는 Input 6개)(클락, 리셋, 숫자 입력 판, EQUAL, 녹음, 회로 전원)**

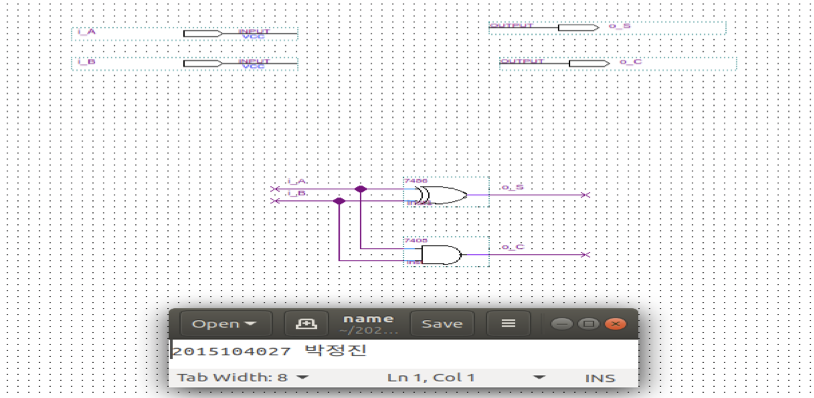


Serial Sending!

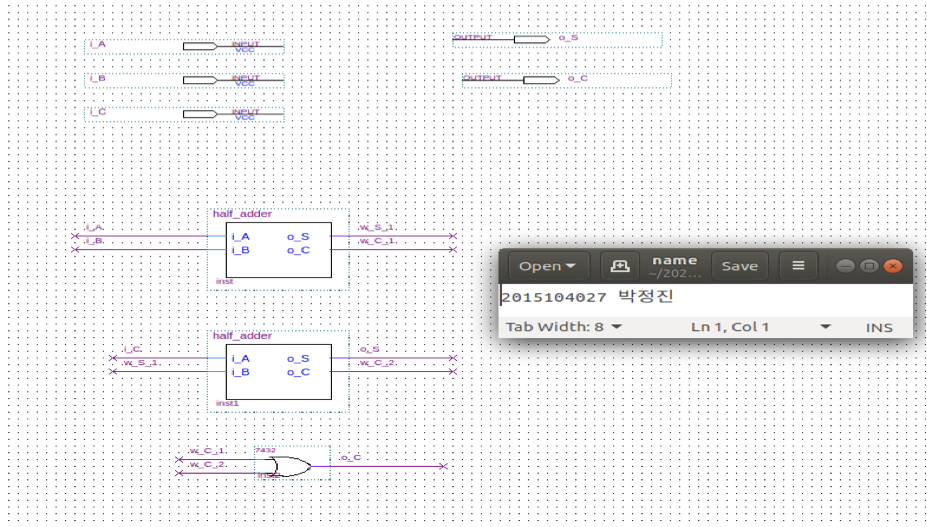
Transmitter



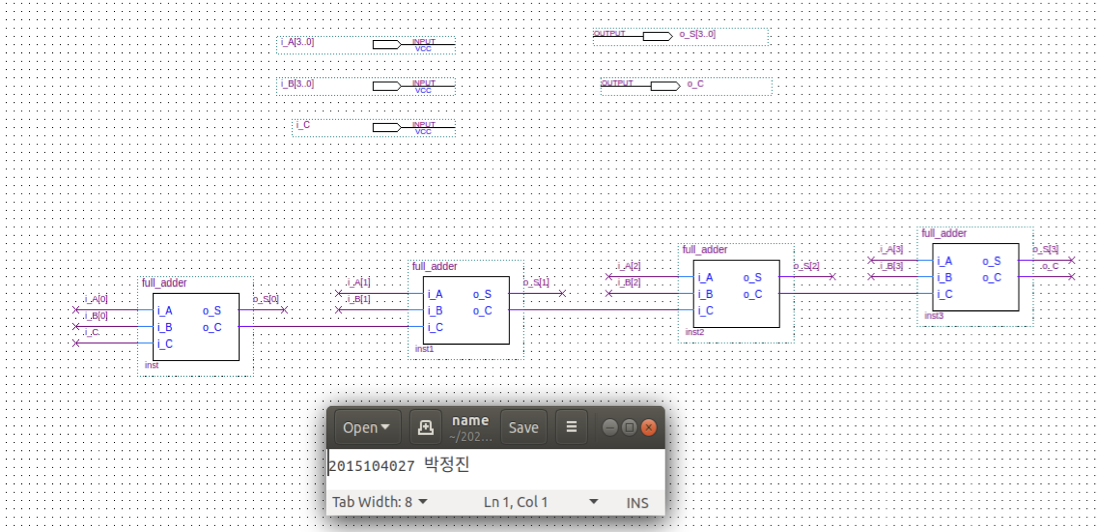
Sub-Module Schematic half adder



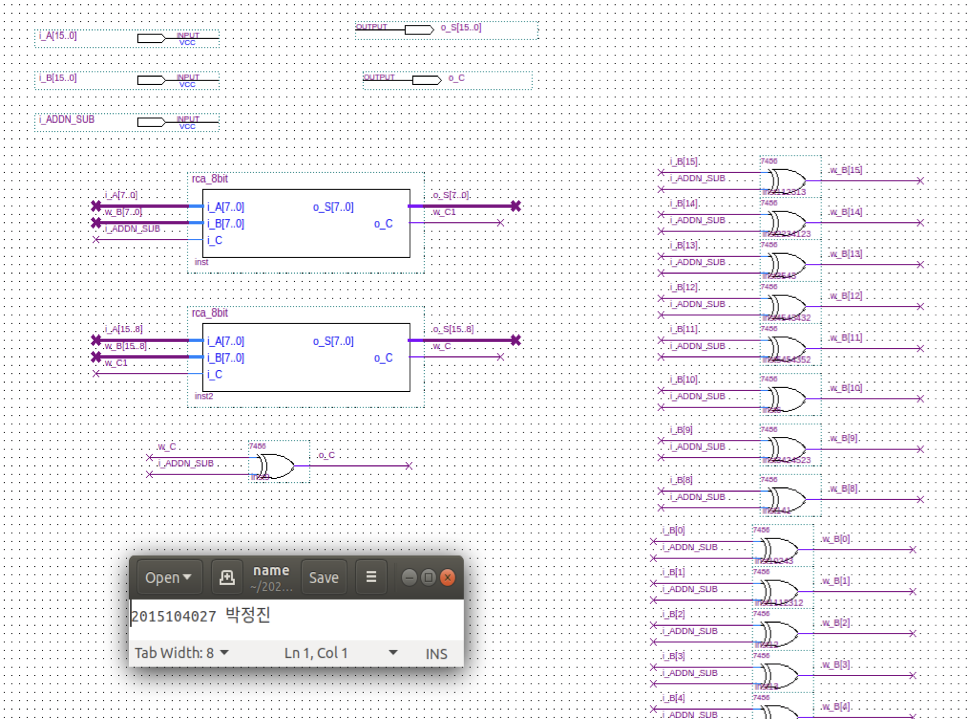
full adder



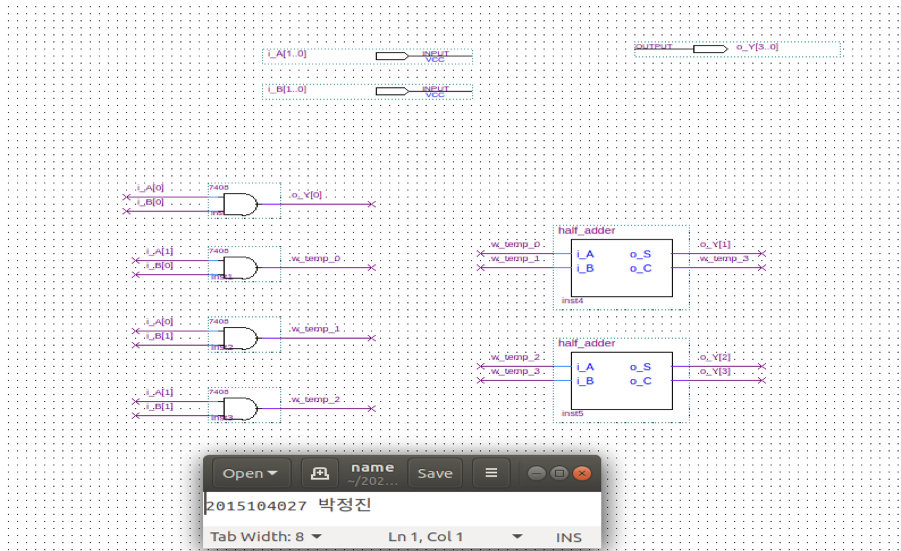
RCA 4bits



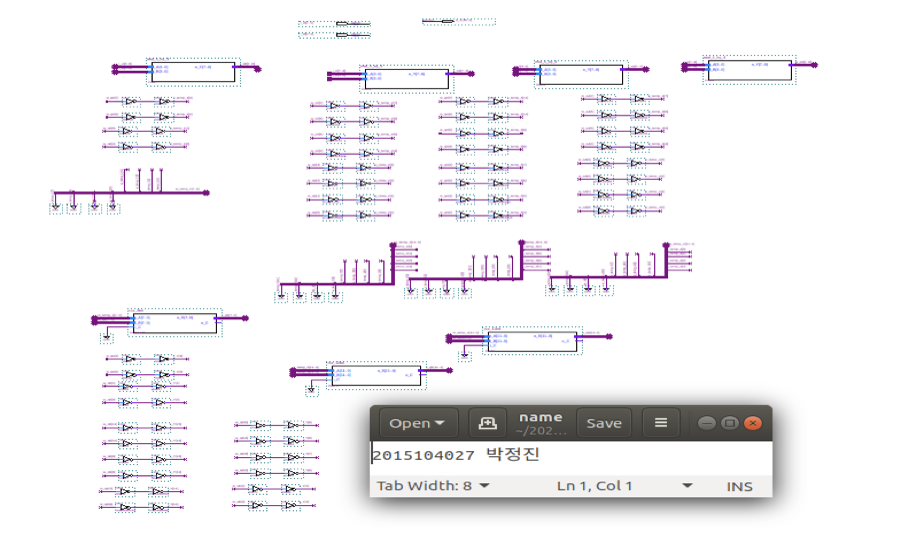
Addn\_Sub\_16bits



Vedic\_Multiplier\_2\_by\_2

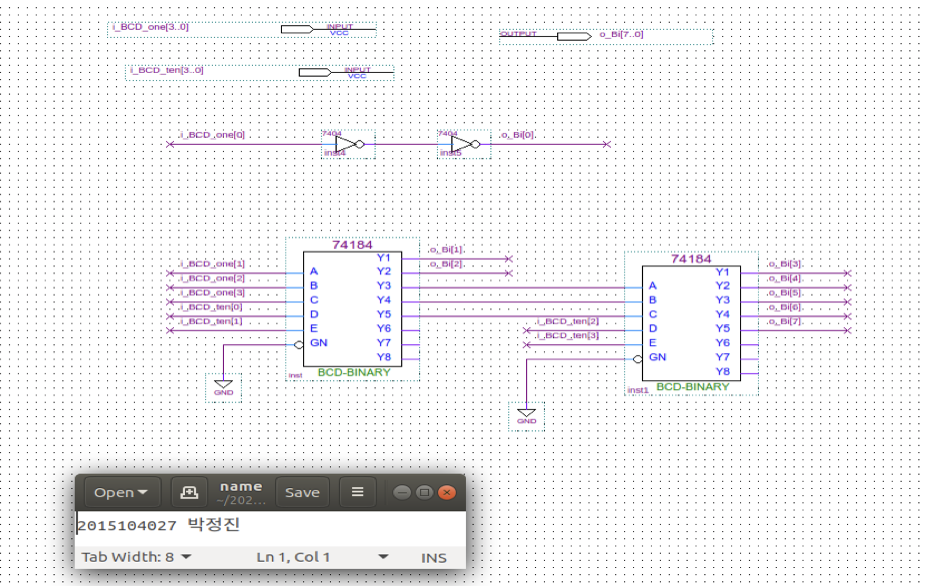


Vedic\_Multiplier\_8\_by\_8

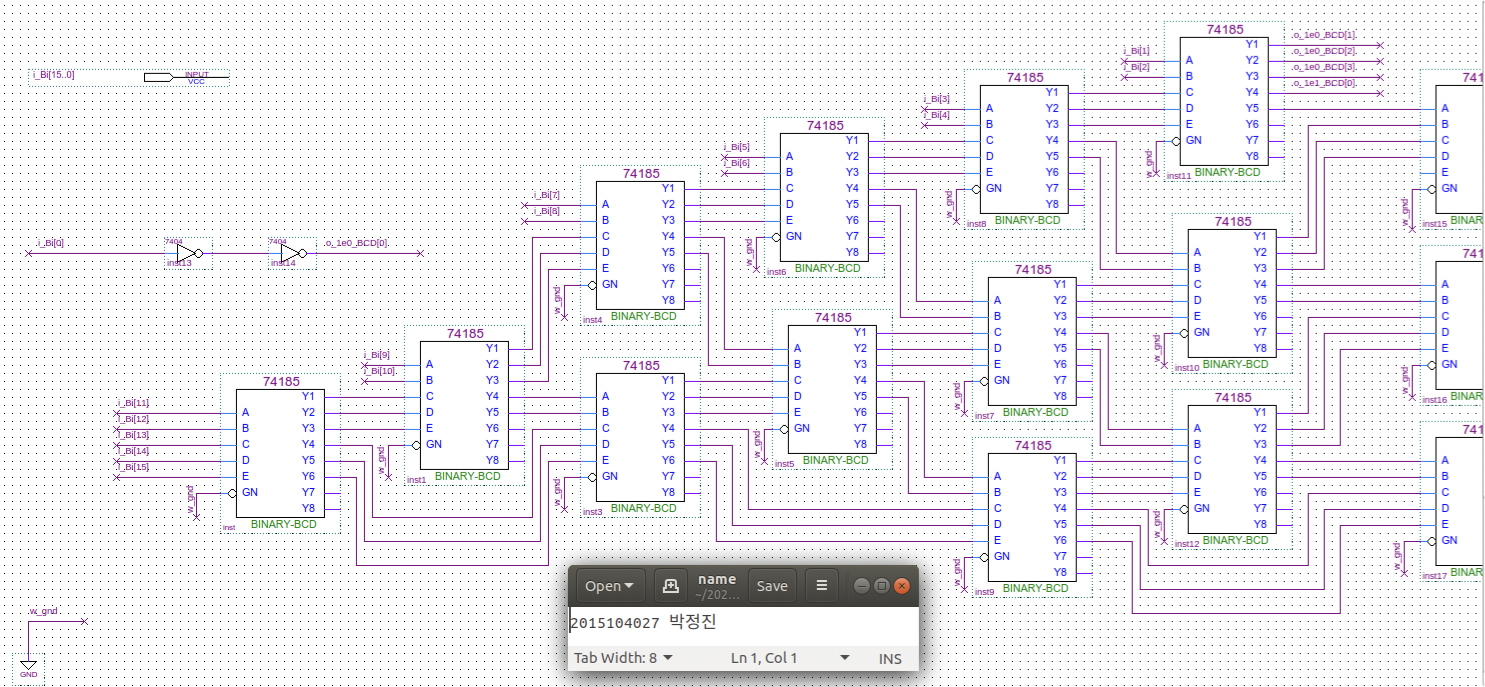
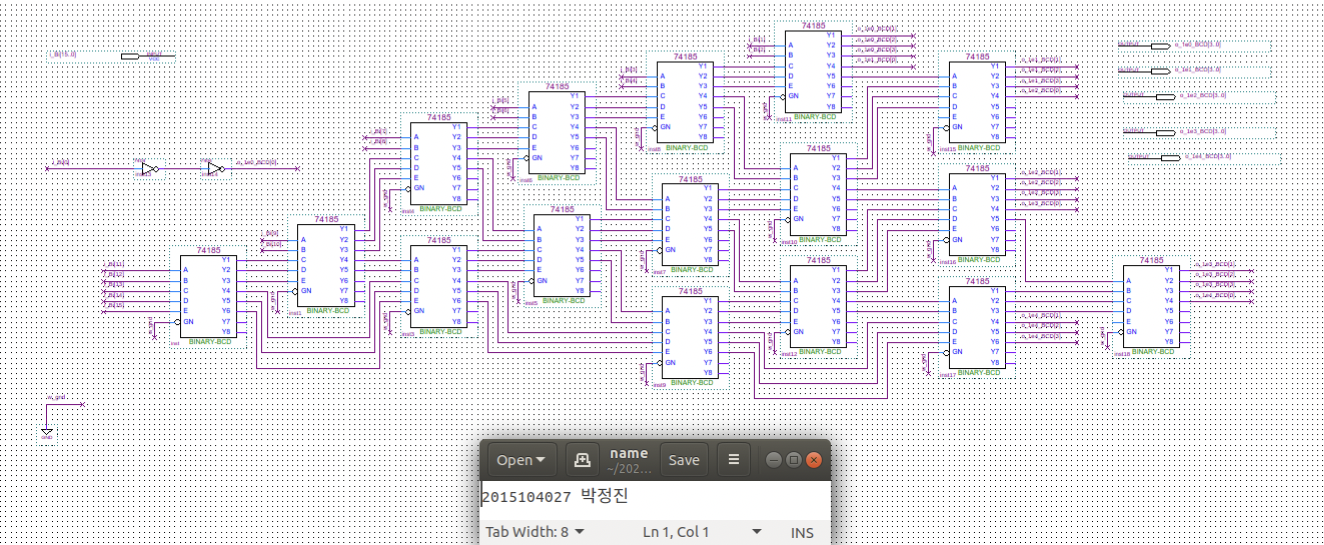




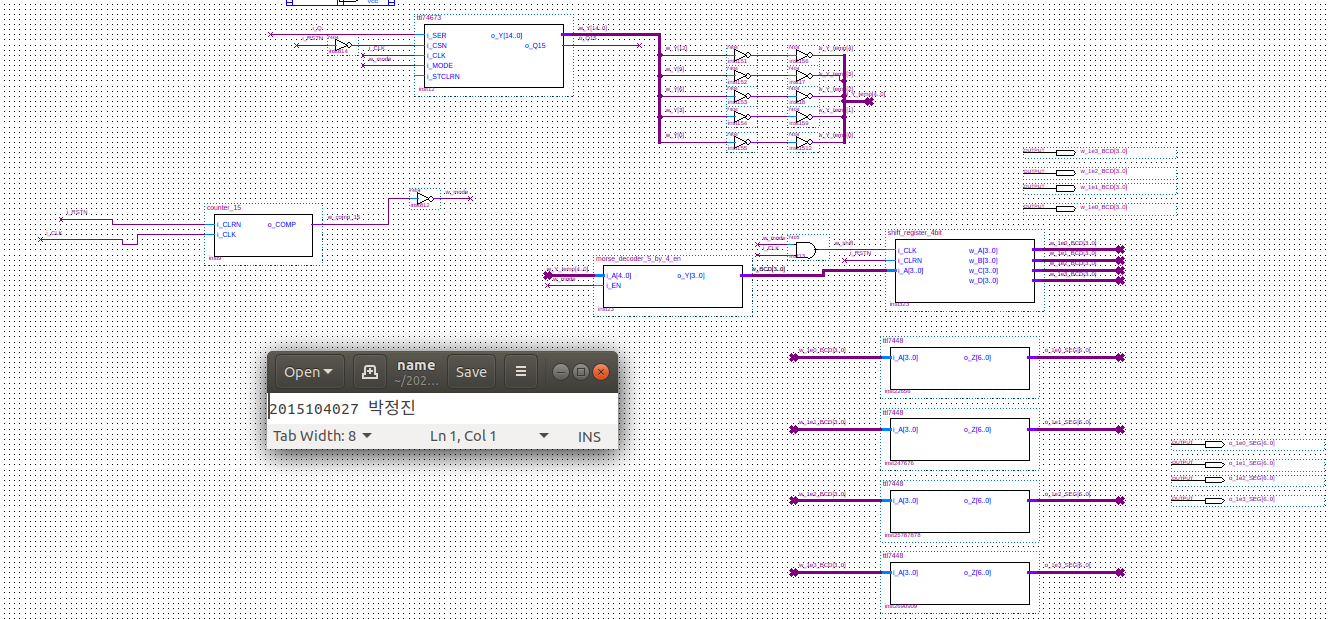
BCD\_to\_Binary\_Converter



Binary\_to\_BCD\_Converter



Receiver

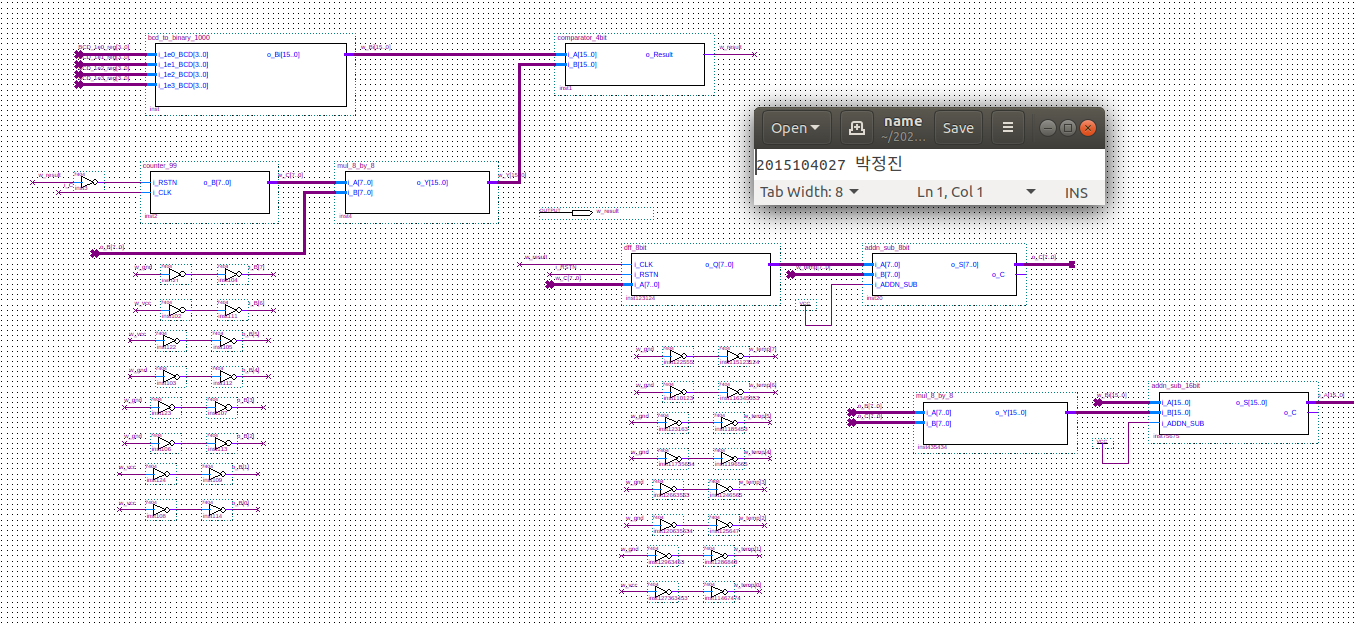


morse\_decoder\_5\_by\_4\_en.bdf

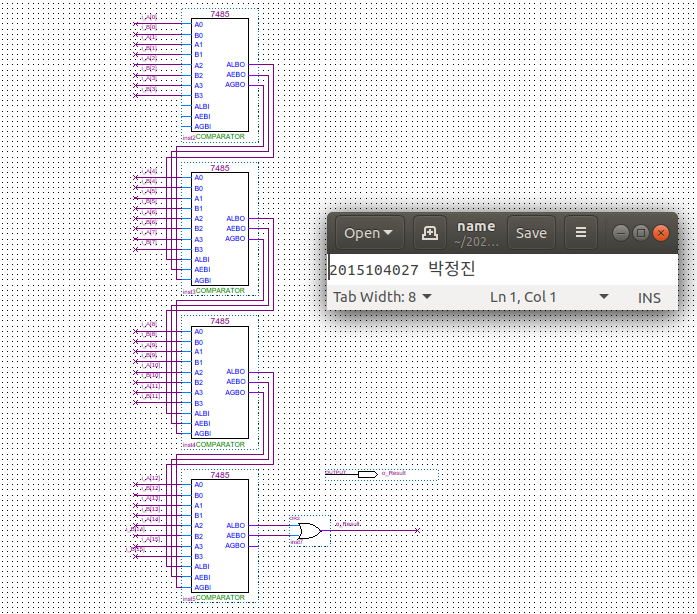
A screenshot of text

Description automatically generated

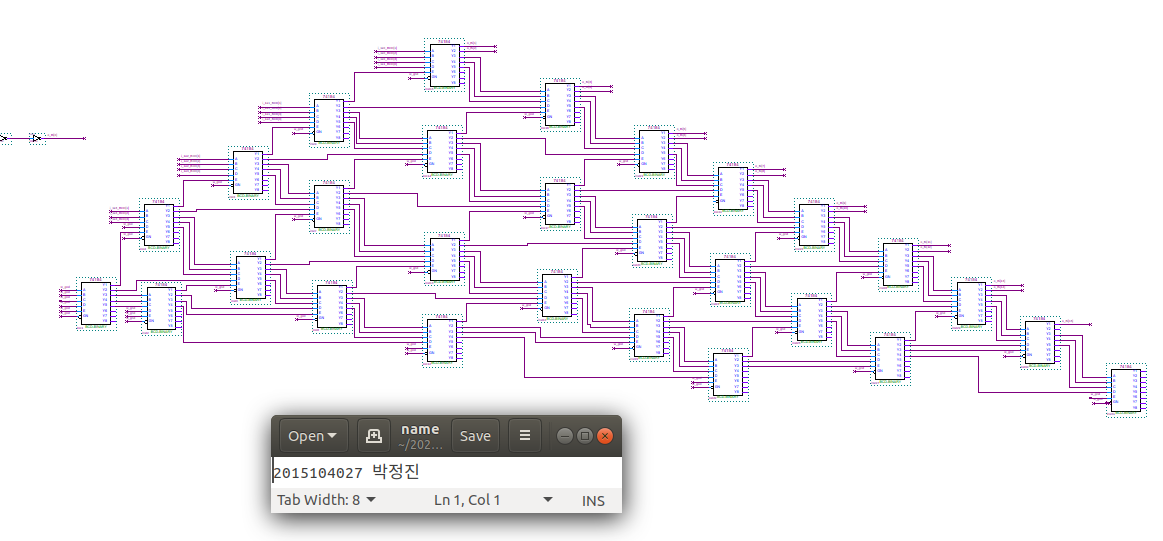
Decipher



Comparator\_16bit



BCD\_to\_Binary\_1000



회로의 구성

Transmitter

Input Part

Shift

Register

BCD

to

Binary\_Converter

Addn\_Sub

&

Multiply

20bits

Parallel

Register

Binary

to

BCD\_Converter

Sender

Morse

ROM

ttl74674

4bit Shift

4 Register

Receiver

Serial Sending!

4bit Shift

4 Register

Morse

Decoder

ttl74673

BCD

to

Binary\_Converter

ttl7448

Display Result

Comparator

16bit

Count

&

Multiply 99

Find A+B\*C

(B=99)

Decipher

회로 설명

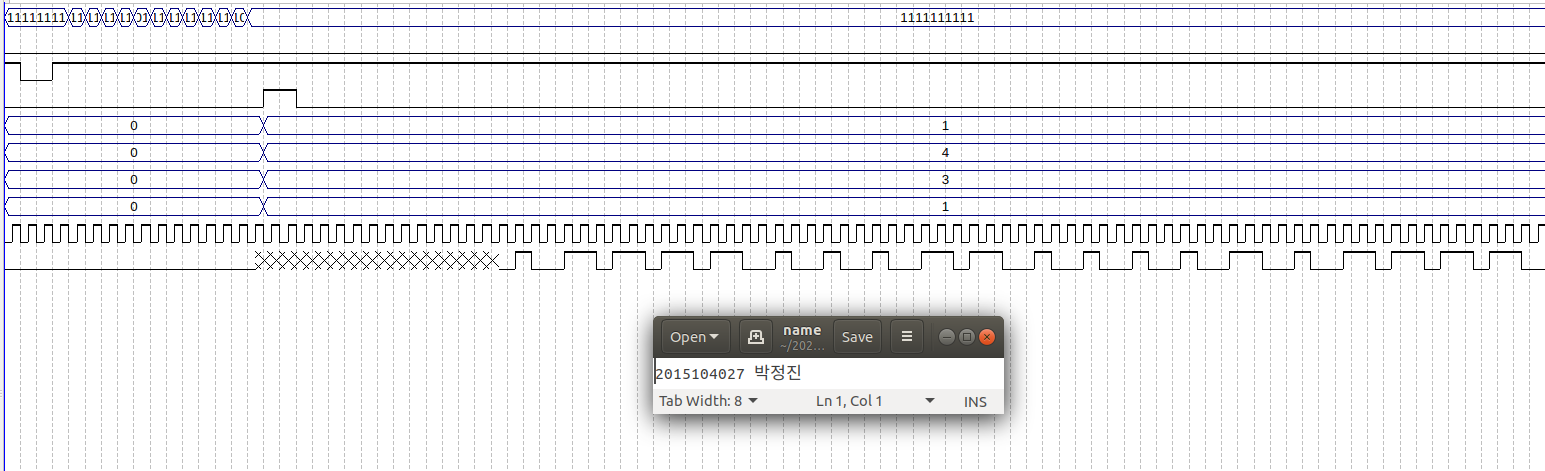
이번 프로젝트의 목표는 다음과 같았다. A+B\*C 의 결과를 정해진 모스 신호로 1bit씩 전송하고 그 것을 받아 다시 원래의 숫자로 해독하여, 최종적으로 7segment에 디스플레이하는 것이다. 거기에 더해 원래의 숫자를 보고 A+B\*C의 하나의 조합을 찾는 것 또한 포함 되었다. 본 프로젝트에서는 **하나의 클락만을 사용해 회로를 구성하였고,** 이를 통해 모든 회로(Transmitter, Receiver, Decipher)가 synchronous하게 동작하게 끔하였다. 특히 Receiver, Decipher는 클락에 Gate Clock을 넣어 Clock Enable 신호와 같은 기능으로 구현하였다. Receiver는 **모스 신호가 들리면 i\_REC이라는 신호를 통해 녹음을 한다는 설정이다.** 이렇게 모스 신호 4자리를 들으면 i\_REC을 꺼 녹음을 멈춘다. 이후 클락이 지나면 세그먼트에 최종 결과 값이 뜨는데 **이 때 Decipher 회로에 i\_ON이라는 신호를 통해 전원을 킨다는 설정도 추가하였다.** 이를 통해 세그먼트에 결과가 완료되는 것을 확인하면 바로 A+B\*C를 찾을 수 있고, 카운터를 이용한 회로이므로 각각의 결과에 따라 나오는 시간은 상이하다. **해독이 완료되면 Decipher 회로에서는 o\_DONE이라는 Flag bit를 통해 해독이 완료되었음을 알려준다. 이 신호를 LED 같은 것에 연결해 사용하면 해독이 완료된 것을 가시적으로 확인할 수 있을 뿐더러, i\_ON 을 꺼 회로의 전원을 끄는 것과 같이 사용할 수 있을 것이다.** 이후 다시 한번 사용하려면 START라는 마스터 리셋을 통해 모든 것을 초기화 하고 다시 같은 방법으로 동작시키면 된다.

A+B\*C 조합을 찾는 방법은 다음과 같다. 먼저 B=99로 고정을 한다. 이를 통해 결과 값을 99로 나눈 몫과 나머지를 찾는 효과인데, **99로 고정을 하면 A+B\*C에서 나올 수 있는 모든 결과를 A+B\*C의 형태로 표현할 수 있다.** 찾는 방법은 카운터 회로를 통해 99\*C를 계산하면서 결과 값과 계속 비교를 한다. 어느 순간 99\*C는 결과 값보다 커질 것이고 그 때는 Comparator가 1을 내보낸다. 1 신호를 받으면 카운팅을 멈추고 저장한다. 저장한 C에 1을 빼고(99\*C가 결과 값 보다 크므로) 다시 99\*C를 계산하고 그 결과를 모스신호로 해석된 결과값에 뺀다. (모스 신호 결과 값 - 99\*C = A) 이를통해 A, B, C의 조합을 쉽게 찾을 수 있다.

동작 방법은 미니프로젝트 1, 2 와 유사하다. 첫 시작 때 START버튼으로 초기화를 시키고원하는 숫자를 6개를 누른다. 이 숫자를 누른 후 EQUAL 을 눌러야만 (0->1) 계산을 실행하고 완료되면 모스 신호를 전달한다. 모스 신호를 전달하면(소리가 나면) Record를 시작한다는 input인 i\_REC을 1로 만든다. 전송되는 모스 신호는 Receiver 회로를 통해 들어가는데 모스 신호를 4자리를 받으면 (20개) 레코딩을 멈춘다. 이후 클락이 지나면 Segment에 값이 고정되는데 이 때가 받은 모스 신호를 처리한 결과이다. 이 Segment가 고정이 된 것을 확인하면 Decipher의 i\_ON을 켜 (0->1) 해독을 시작한다. 카운팅을 몇 번 하냐에 따라 해독이 끝나는 시간이 상이하지만 o\_DONE 플래그를 통해 해독이 완료됨을 확인 할 수 있고, 이 플래그가 나왔을 때 결과 값은 고정적으로 나온다. 이후 해독이 끝나면 i\_ON을 꺼 회로가 파워를 소모하지 않도록 한다.

Simulation

계산한 결과를 모스 신호로 전달하는 모습



모스 신호 송신

Case 1)

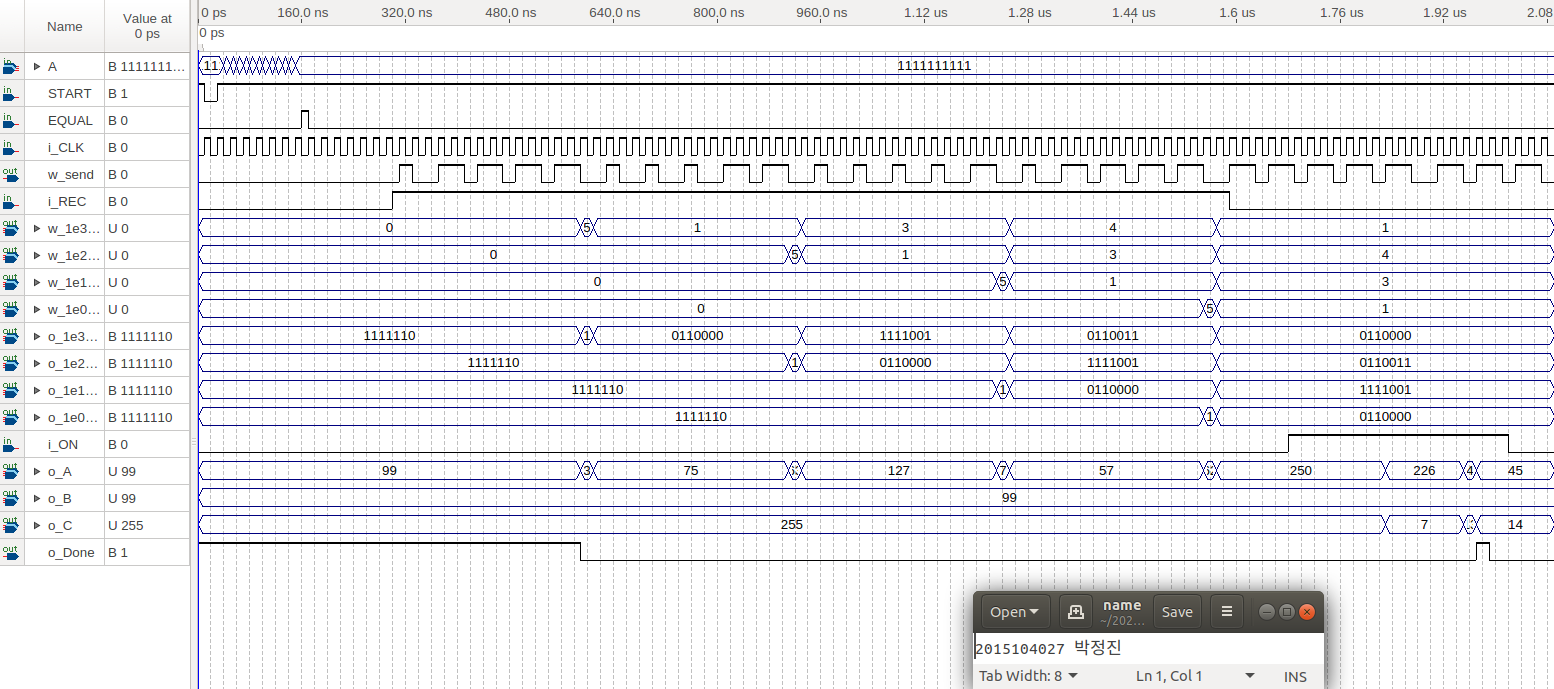
A = 8-> 7 -> 5-> 6 -> 2 -> 4 순으로 입력

결과 예상

87+56\*24 = 1431

조합 : 45 +99\*14 = 1431

고정된 7Segment도 결과 숫자에 맞게 제대로 출력되는 것을 확인 할 수 있다.



해독 끝

해독 완료

결과 고정

해독 시작

녹음 끝

모스 신호 송신

녹음 시작

Case 2)

Case 1 이후 START로 리셋한 후

A = 9-> 6 -> 4-> 6 -> 7 -> 2 순으로 입력

결과 예상

96+46\*72 = 3408

조합 : 42 +99\*34 = 3408

Case 1과 마찬가지로 고정된 7Segment도 결과 숫자에 맞게 제대로 출력되는 것을 확인 할 수 있다.

