

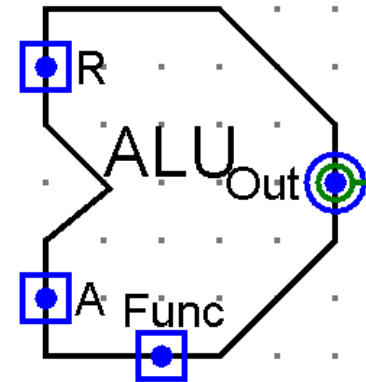
# HW1 연산 및 레지스터 제어

---

- 8bit 레지스터 4개로 구성된 회로에서 다음과 같은 기능이 가능한 회로 설계
  - 제어 입력과 데이터 입력을 통하여 레지스터 초기화
  - 데이터가 저장된 레지스터 간에 연산을 하여 레지스터에 저장
  - 연산 unit을 통하여 레지스터 간에 데이터 전송

# 연산 unit

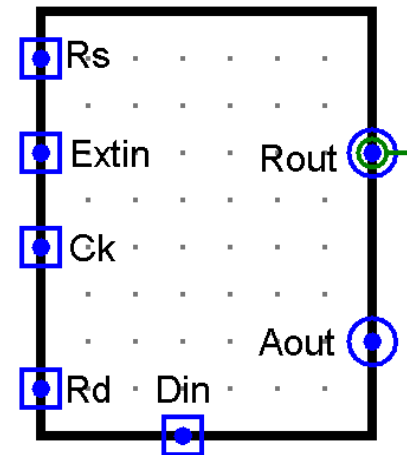
- Input
  - A, R : 8 bit
  - function code : 2 bit
- Output
  - Out : 8 bit 연산결과
  - ADD 연산 결과의 carry 출력 무시
- 동작 표



Func Code	Out
00	ADD : $\text{Out} = A + R$
01	Pass : $\text{Out} = R$
10	CIR : $\text{Out} = \text{Circular shift right } A$
11	CIL : $\text{Out} = \text{Circular shift left } A$

# Register Bank 기능

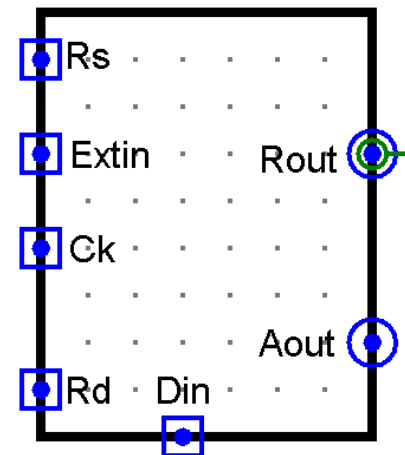
- 4 개의 8bit Register
  - A, B, C, D
- 2개의 8bit 입력선
  - Extin : 외부 입력선으로 Rout 으로 연결 가능
  - Din : 연산 결과를 레지스터에 저장할 입력 선
- 2개의 8bit 출력선
  - Rout : Extin, B, C, D 중에 Rs(2bit) 에 의해 선택된 것이 연결됨.
  - Aout : 레지스터 A 출력이 직접 연결됨.
- 2개의 2bit 선택 입력
  - Rs : Rout 으로 연결될 레지스터 선택
  - Rd : Din 데이터 입력을 저장할 레지스터 선택



# 선택 입력에 사용되는 Register 번호

- 2개의 선택 입력
  - Rs : Rout 으로 연결될 데이터 선택

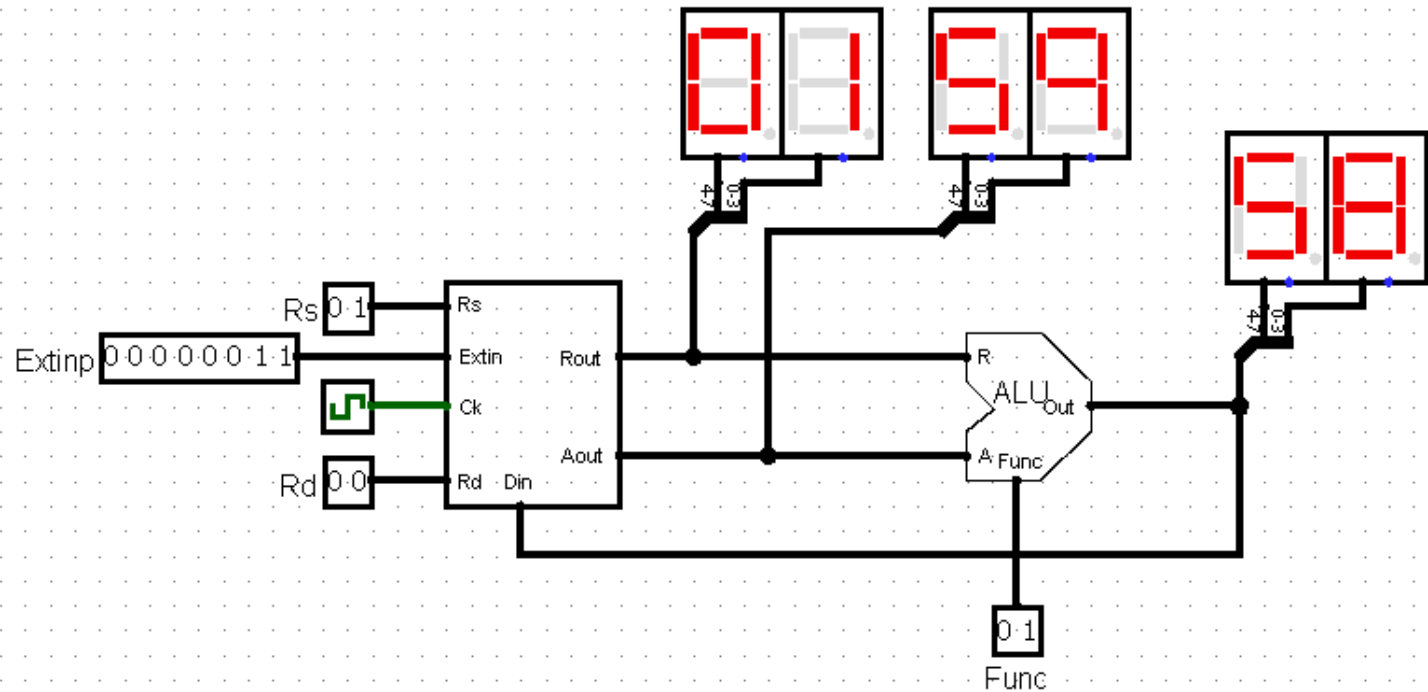
Rs	Rout
00	Extin
01	B
10	C
11	D



- Rd : Din 데이터 입력을 저장할 레지스터 선택

Rd	Register
00	A
01	B
10	C
11	D

# 연산 unit과 Register 제어 unit 을 연결한 그림



# 데이터 설정 및 동작 예

---

## ■ 초기화

- 외부 입력 Extin, Rs, Func, Rd 를 다음과 같이 설정하여 clock 입력
  - $\text{reg A} \leftarrow 50$  : Extin = 50, Rs = 0 (Extin), Func = 01 (Pass R), Rd = 0 (A)
  - $\text{reg B} \leftarrow 01$  : Extin = 01, Rs = 0 (Extin), Func = 01 (Pass R), Rd = 1 (B)

## ■ A를 1씩 증가

- $A \leftarrow A + B$  : Extin = xx, Rs = 1 (B), Func = 00 (A+B), Rd = 0 (A)
- clock 이 입력될 때마다 증가 됨

- A를 1씩 증가하여 54까지
  - Func =00 선택 -> 클릭 4번



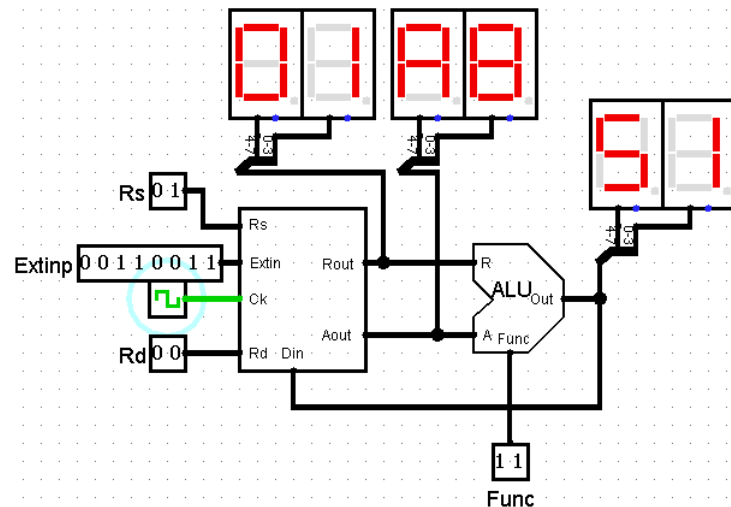
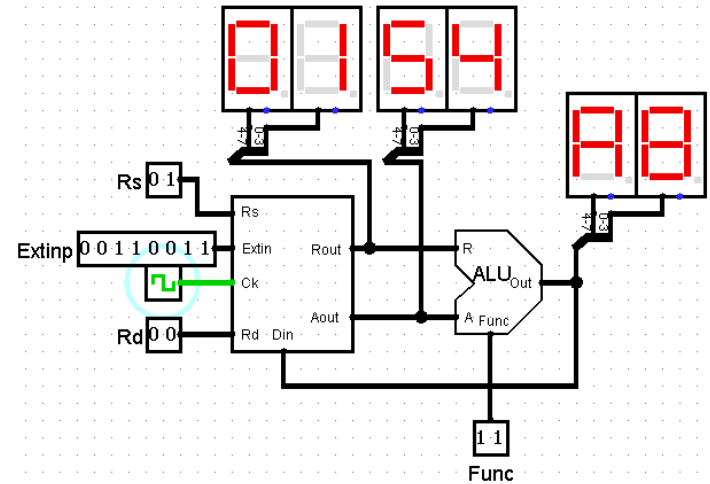
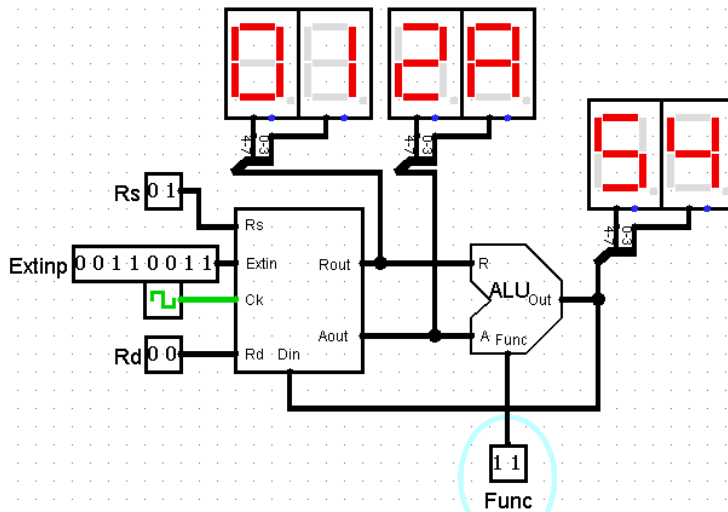
- A를 circular shift right 1번
  - Func = 10 선택 -> 클럭 1번





# 동작 그림

- A를 circular shift left 2번
  - Func = 11 선택 -> 클럭 2번



# 로지심 모듈만을 사용해서 구현 가능

- 디코더 모듈과 attribute 설정



Selection: Decoder	
Facing	North
Select Location	Bottom/Left
Select Bits	2
Three-state?	No
Disabled Output	Zero
Include Enable?	No

- 멀티플렉서 모듈과 attribute 설정

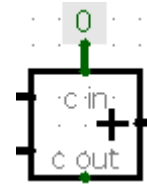


Selection: Multiplexer	
Facing	East
Select Location	Bottom/Left
Select Bits	2
Data Bits	8
Disabled Output	Zero
Include Enable?	No

# 연산과 register 모듈

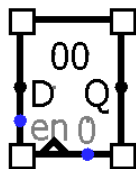
## ■ Adder 와 Subtractor

- c in 입력을 0 으로 연결
- c out은 open



## ■ register 모듈

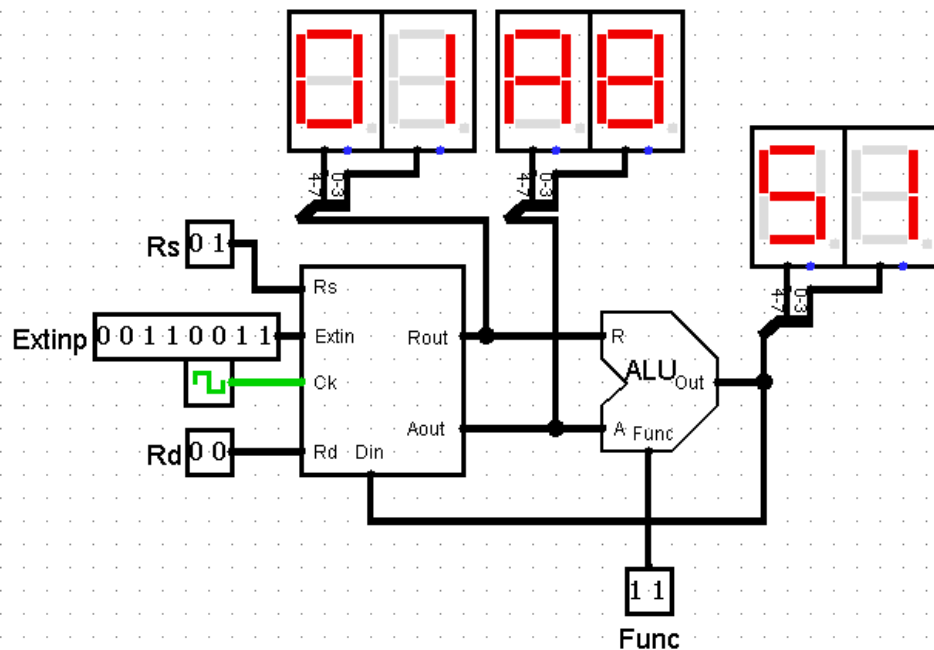
- 설정할 필요없이 디폴트 값으로 동작하도록 되어 있음
- 0 으로 표시된 비동기 clear 입력은 사용할 필요 없음



Selection: Register	
Data Bits	8
Trigger	Rising Edge
Label	
Label Font	SansSerif Plain 12

# 요구사항

- 아래와 같은 Main 회로를 구성한다.
  - register bank 제어와 ALU 회로를 subcircuit 으로 만든다.
  - subcircuit 모양과 핀 배치를 최대한 주어진 모양과 비슷하게 만든다.
  - subcircuit 핀 이름들도 그림과 같은 이름으로 나타낸다.
  - 레지스터 모듈과 연산 모듈 출력에 Hex Digit Display로 값을 나타낸다.
  - Document 이름과 저자 정보



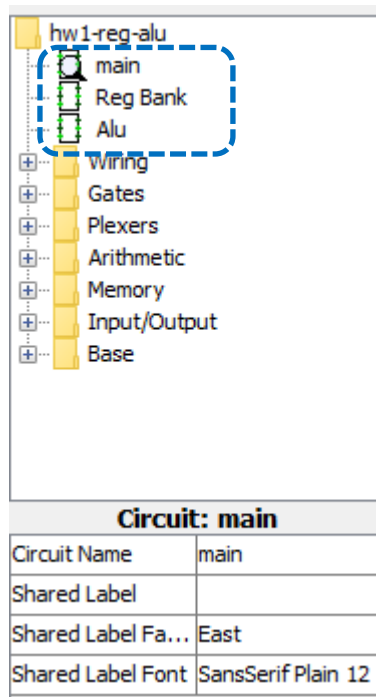
Hw1 연산과 레지스터 제어

학번 : 2111111

이름 : 홍길동

# 요구사항

- 앞에 동작 그림에서 보여준 4가지 Sample 출력과 같은 형식의 출력을 캡처하여 MS word 파일에 그림을 삽입하여 제출한다.
  - 4가지: 초기화한 후, A를 1씩 증가한 결과, Circular shift right 1번 후, Circular shift left 2번 후
- 로지심 회로에는 아래와 같이 입출력을 나타내는 main 회로와 2개의 subcircuit 로만 구성된 회로를 제출한다.



# 제출물

---

- logisim 회로 파일 (파일이름: 학번-hw1.circ)
- 샘플 출력들을 캡처한 MS word 파일 (파일이름: 학번-hw1-output.docx)
- 마감 일시:
  - 9월 28일 23시 59분