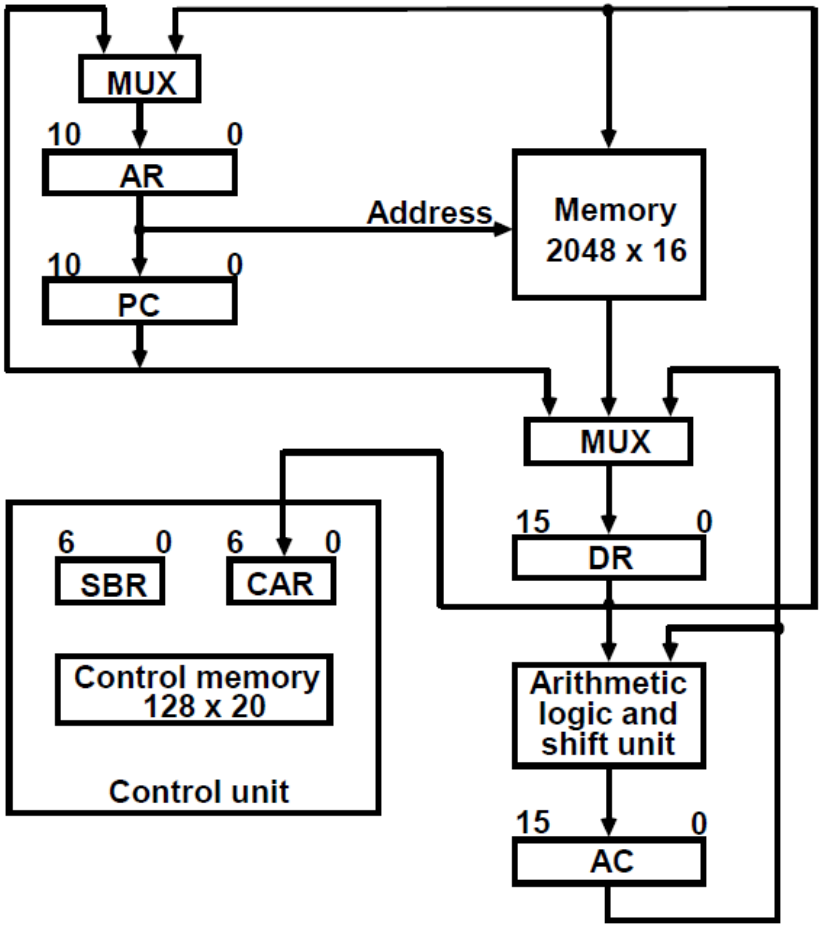
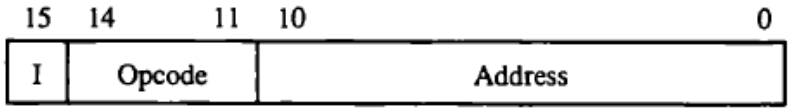
HW4 Microprogramming 실습

- 7장에 소개된 micropgrammed CPU를 구현
- 강의 자료에 설명된 Horizontal microinstruction 방식으로 구현



Instruction Format

- Instruction Format :



(a) Instruction format

  - I : 1 bit for indirect addressing
  - Opcode : 4 bit operation code
  - Address : 11 bit address for system memory

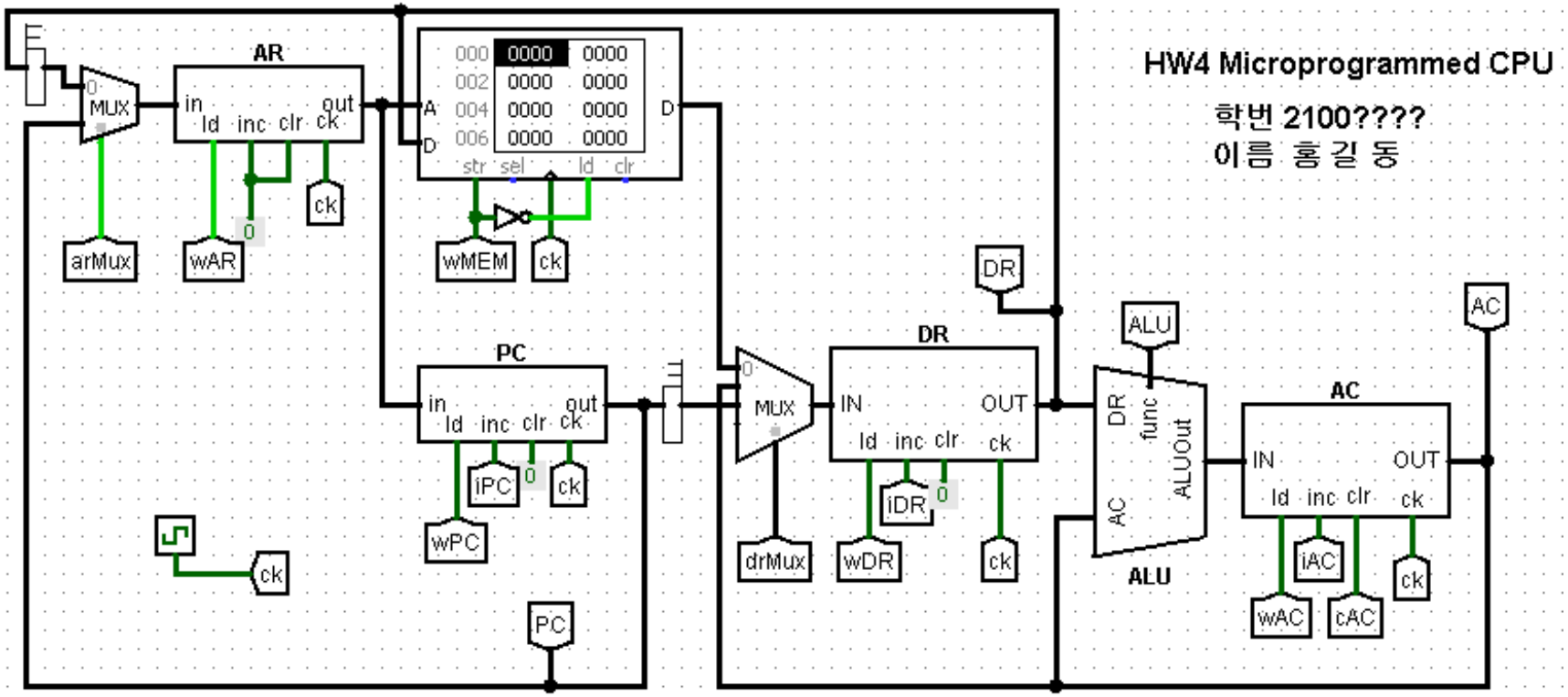
- Computer Instruction :
  - 16 명령어가 가능하며 4 개만 표시

Symbol	Opcode	Description
ADD	0000	$AC \leftarrow AC + M\,[EA]$
BRANCH	0001	If $(AC < 0)$ then $(PC \leftarrow EA)$
STORE	0010	$M\,[EA] \leftarrow AC$
EXCHANGE	0011	$AC \leftarrow M[EA], M[EA] \leftarrow AC$

EA is the effective address

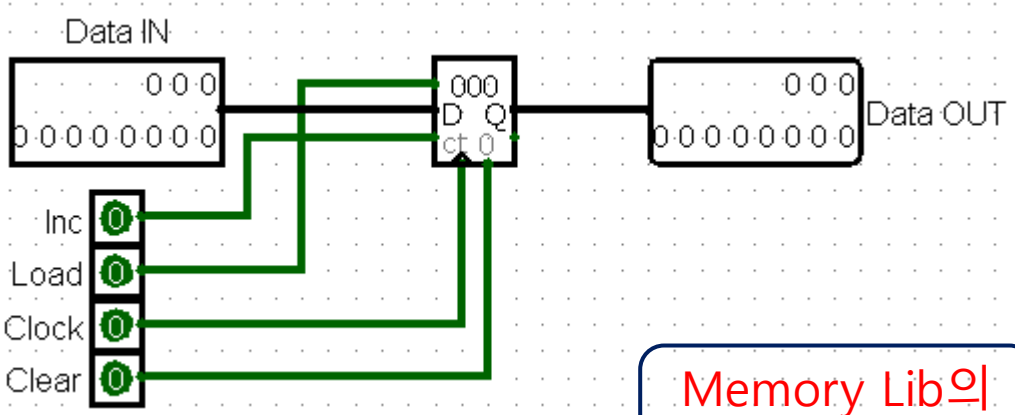
(b) Four computer instructions

Registers and Data Path in CPU

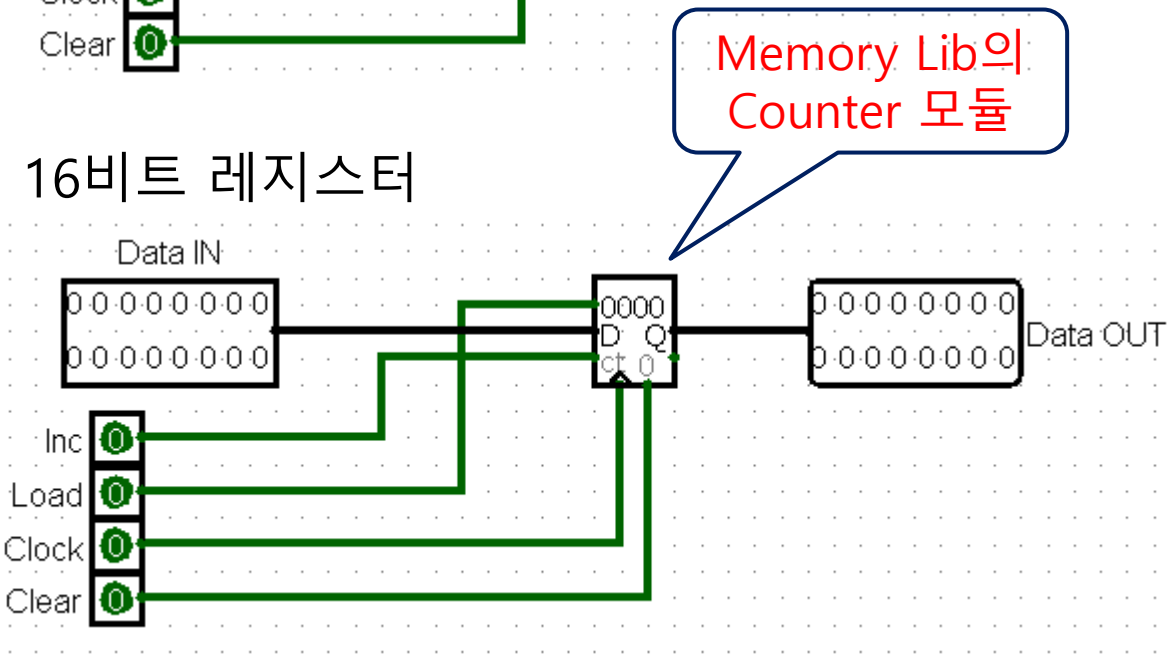


11비트 레지스터와 16비트 레지스터 subcircuit

- 11 비트 레지스터

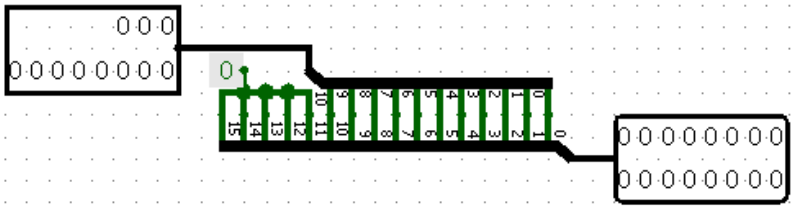


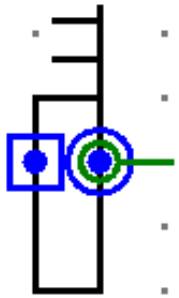
- 16비트 레지스터



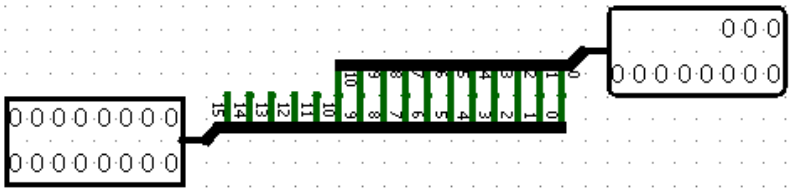
Bit 확장 및 축소 subcircuit

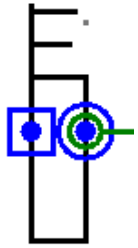
- 11 to 16





- 16 to 11

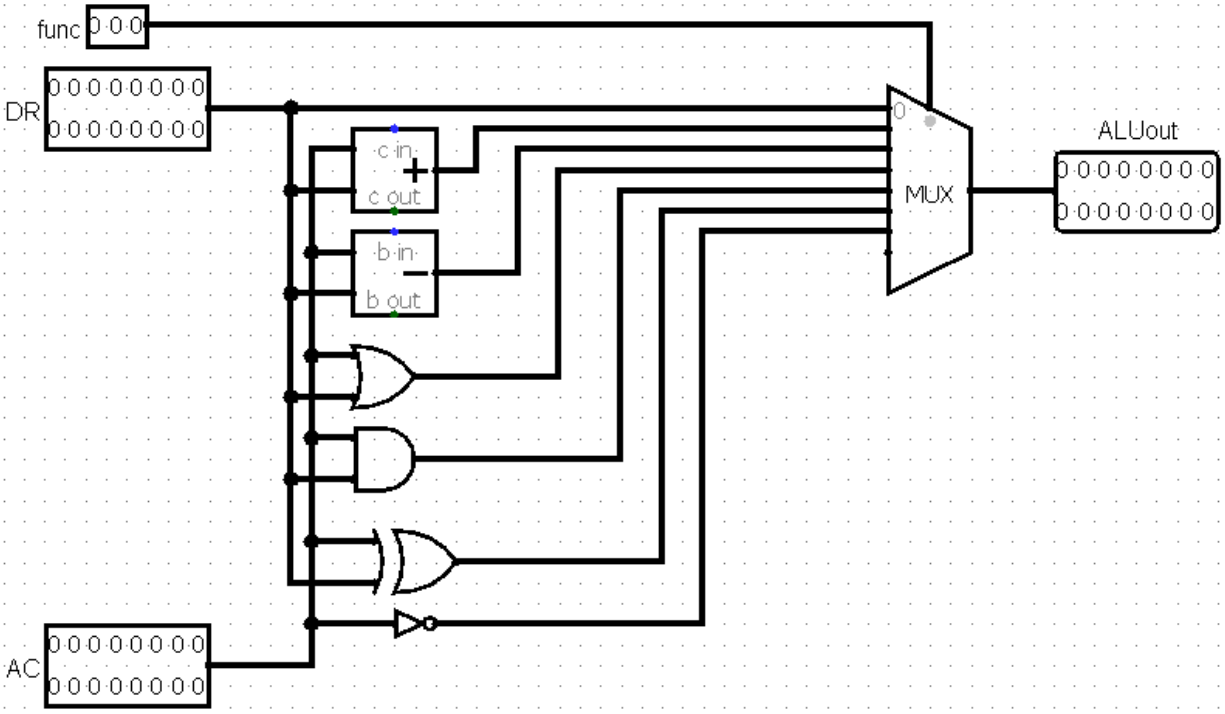




ALU subcircuit

- ALU function 을 7가지로 줄임 -> func 입력 : 3bit

func	ALU
0	DR
1	AC + DR
2	AC - DR
3	AC v DR
4	AC ^ DR
5	AC xor DR
6	AC'



Microprogram Sequencer

- CAR : 7 bit
- Input Logic : control Mux1

Input :

I_0, I_1 from Branch bit (BR)
T from MUX 2 (T)

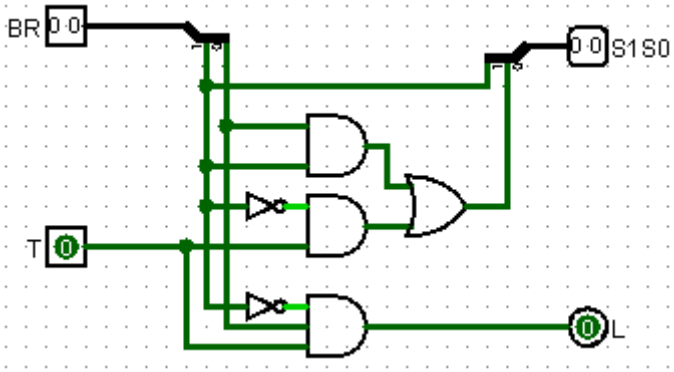
Output :

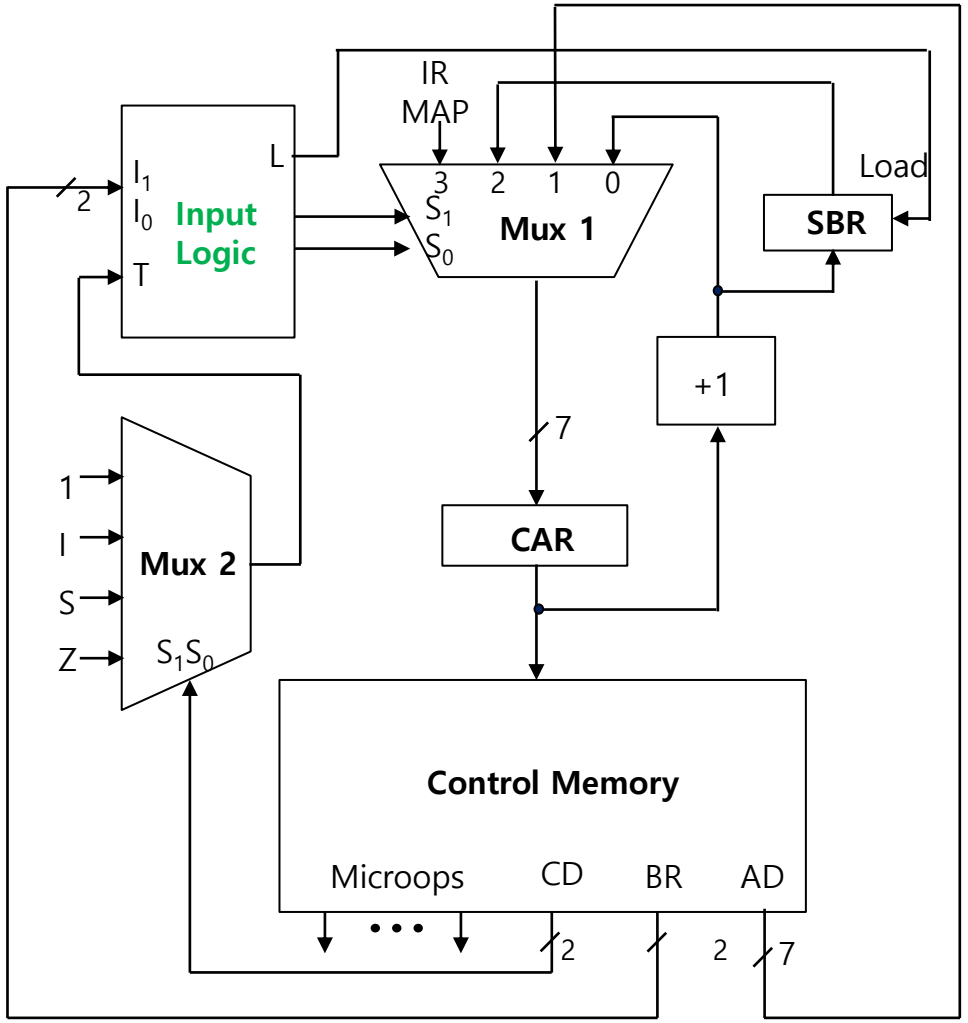
MUX 1 Select signal (S_1, S_0)

$S_1 = I_1$
$S_0 = I_1' T + I_1 I_0$

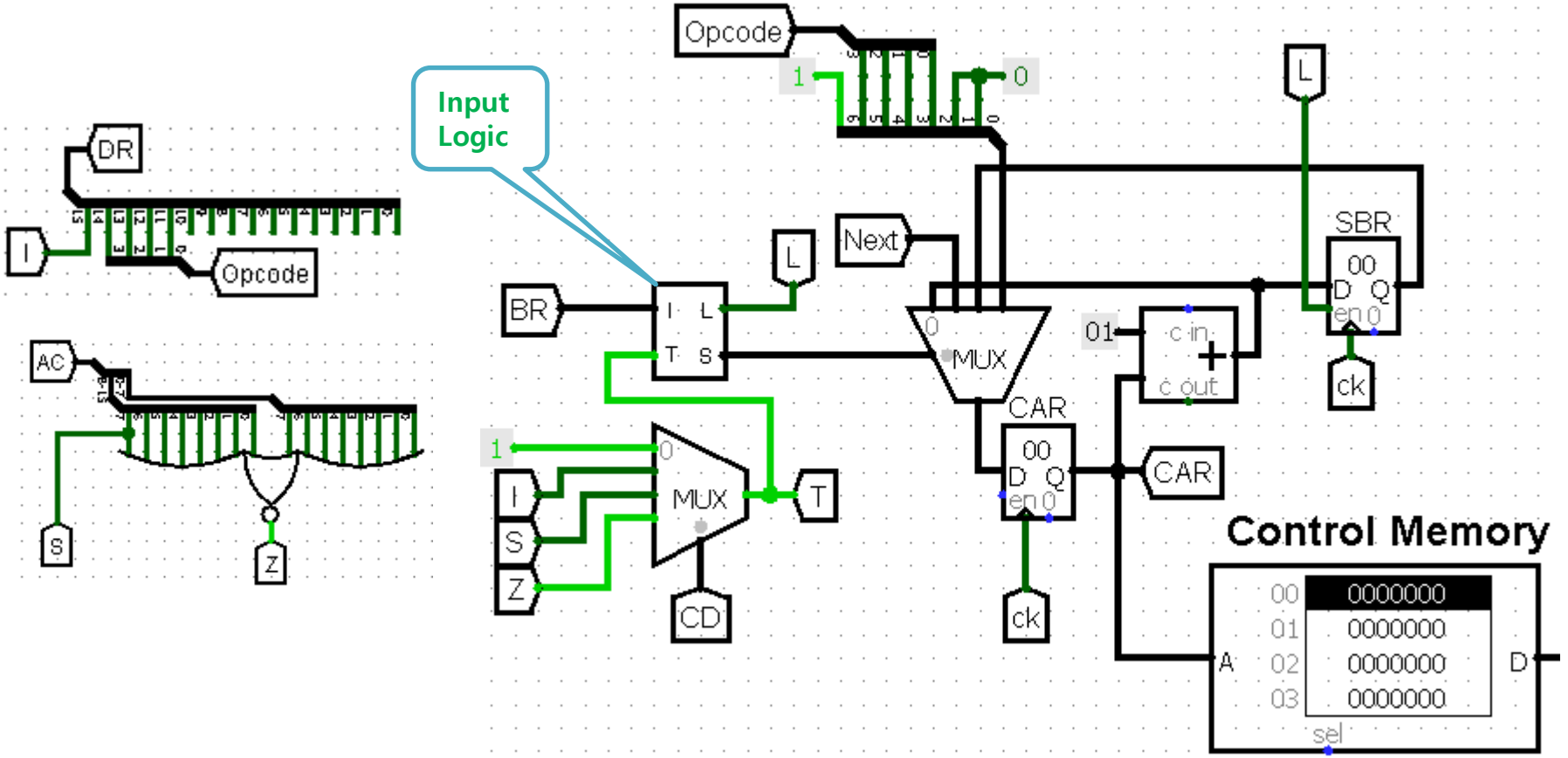
SBR Load signal (L)

$L = I_1' I_0 T$



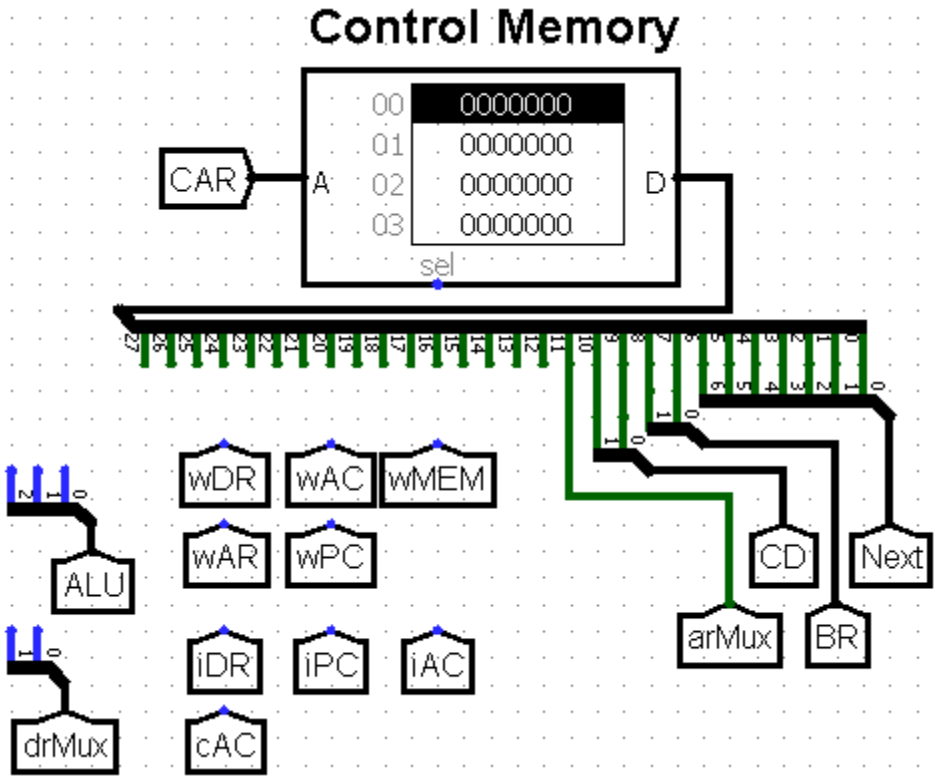


Sequencer and Condition Flag



Control Signals

- Control Memory Output: Excel Sheet에 배치된 순서로 제어신호 배치



제어신호에서 16진수 마이크로 코드 변환

- Fetch routine(address = 0)

u-operation		control singnals
PCTAR	U JMP NEXT	wAR, arMUX=1
READ, INCPC	U JMP NEXT	wDR , drMux = 0 (MEM), incpc
DRTAR	U MAP	wAR, arMux= 0 (DR), BR=3

B	D	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
		1	1	1	1	1	1	1	1	1	3	2	1	2	2	7	micro-code			
	dec	wMeM	cAC	iAC	iDR	iPC	wAC	wDR	wPC	wAR	ALU	drMux	arMux	CD	BR	Next	addr	hex		
fetch	0									1			1	0	0	1	00	00	20	801
	1					1		1				0	0	0	0	2	01	02	80	002
	2									1			0		3		02	00	20	180

6 bit -> 2자리 hex

8 bit -> 2자리 hex

12 bit -> 3자리 hex

10진수 주소

=DEC2HEX(D5, 2)

=DEC2HEX(H5*32+I5*16+J5*8+K5*4+L5*2+M5 ,2)

=DEC2HEX(N5*128+O5*64+P5*32+Q5*4+R5 ,2)

=DEC2HEX(S5*2048+T5*512+U5*128+V5,3)

제어신호에서 16진수 마이크로 코드 변환

- Indirect subroutine (address = 3)

u-operation	control singnals
READ U JMP NEXT	wDR , drMux = 0 (MEM)
DRTAR U RET	wAR, arMUX=0, BR=2

		1	1	1	1	1	1	1	1	1	3	2	1	2	2	7	micro-code			
		wMeM	cAC	iAC	iDR	iPC	wAC	wDR	wPC	wAR	ALU	drMux	arMux	CD	BR	Next	addr	hex		
indiret								1				0		0	0	4	03	00	80	004
										1			0		2		04	00	20	100

제어신호에서 16진수 마이크로 코드 변환

- ADD routine (address = 40h)

u-operation	control signals
NOP I CALL INDRCT	BR=1(CALL), CD=1(I bit) , INDRCT
READ U JMP NEXT	wDR , drMux = 0 (MEM)
ADD U JMP FETCH	wAC, ALU = 1 (Add)

	A	B	D	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
3				1	1	1	1	1	1	1	1	1	3	2	1	2	2	7	micro-code			
4			dec	wMeM	cAC	iAC	iDR	iPC	wAC	wDR	wPC	wAR	ALU	drMux	arMux	CD	BR	Next	addr	hex		
69		ADD	64													1	1	3	40	00	00	283
70			65							1				0		0	0	66	41	00	80	042
71			66						1				1			0	0	0	42	01	04	000
72			67																43	00	00	000

제어 신호 채우기

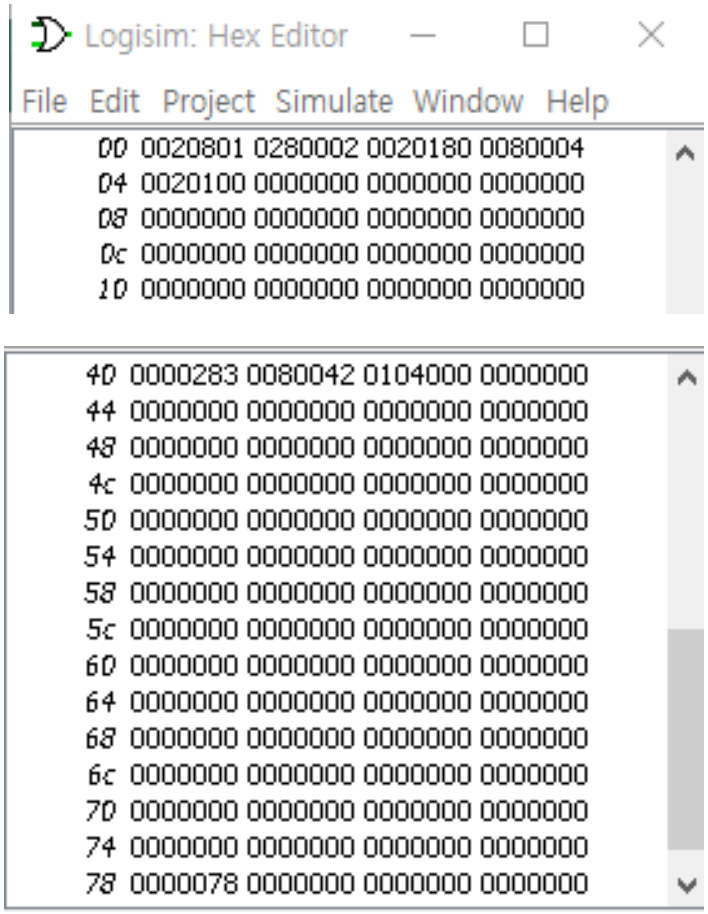
- 교재 표7.2 (p185) 참조하여 나머지 명령어 코드에 매핑되는 빈 공간에 필요한 신호를 채워서 16진수 micro-code를 완성한다.

		1	1	1	1	1	1	1	1	1	3	2	1	2	2	7	micro-code			
	dec	wMeM	cAC	iAC	iDR	iPC	wAC	wDR	wPC	wAR	ALU	drMux	arMux	CD	BR	Next	addr	hex		
ADD	64													1	1	3	40	00	00	283
	65							1				0		0	0	66	41	00	80	042
	66						1				1			0	0	0	42	01	04	000
	67																43	00	00	000
BR	68																44	00	00	000
	69																45	00	00	000
	70																46	00	00	000
	71																47	00	00	000
ST	72																48	00	00	000
	73																49	00	00	000
	74																4A	00	00	000
	75																4B	00	00	000
EX	76																4C	00	00	000
	77																4D	00	00	000
	78																4E	00	00	000
	79																4F	00	00	000
	119																77	00	00	000
HLT	120													0	0	120	78	00	00	078

(* HLT 실행은 다음 주소를 자신의 주소로 설정하여 제어 unit 진행을 멈춤. *)

Control 메모리 입력

- Control 메모리 입력하여 저장



Test CPU

1. image 파일 (ch7-Main-RAM-TEST1)을 Main memory에 load 하여 실행하여 결과 확인

   0A 번지: 0 -> FFFE(-2) -> 4
   0B 번지: 0 -> 000a

2. image 파일 (ch7-Main-RAM-TEST2)을 Main memory에 load 하여 실행하여 결과 확인

   0C 번지: 0010 -> 0011 -> 0012 -> 0013 -> 0014
   0E 번지: FFFC(-4) -> FFFD(-3) -> FFFE(-2) -> FFFF(-1) -> 0000
   10 번지: 0030 -> 0031
   11 번지: 0040 -> 0041
   12 번지: 0050 -> 0051
   13 번지: 0060 -> 0061

Test Program1

```
        ORG     0
        ADD     DATA1   / A = 0 + -8
LOOP,   ADD     DATA2   / A = A + 6
        ST      DATA3   / -2 -> +4
        BR      LOOP    / if A < 0, goto LOOP
        ADD     DATA2   / A = A + 6
        ST      DATA4   / A = 000A (10)
        HLT

        ORG     8
DATA1,  HEX     FFF8    / -8
DATA2,  HEX     6
DATA3,  HEX     0       / 0 -> FFFE(-2) -> 0004
DATA4,  HEX     0       / 0000 -> 000A
        END
```

Label	Address ▲	Instruction	Hex
	000	ADD DATA1	0008
LOOP	001	ADD DATA2	0009
	002	ST DATA3	100A
	003	BR LOOP	0801
	004	ADD DATA2	0009
	005	ST DATA4	100B
	006	HLT	7000
	007		0000
DATA1	008	HEX FFF8	FFF8
DATA2	009	HEX 6	0006
DATA3	00A	HEX 0	0000
DATA4	00B	HEX 0	0000

Test Program1 image file name : ch7-Main-RAM-TEST1

Test Program2

```
        ORG     0
LOOP,   EX      PTR I
        ADD     ONE
        EX      PTR I    / increment M[PTR]
        EX      PTR
        ADD     ONE
        EX      PTR      / increment PTR
        EX      CNT
        ADD     ONE
        ST      CNT      / increment CNT
        BR      LOOP      / if CNT < 0, goto LOOP
        HLT

        ORG     C
PTR,    HEX     10       / 10 -> 14
ONE,    HEX     1
CNT,    HEX     FFFC     / -4 -> 0

        ORG     10
        HEX     30
        HEX     40
        HEX     50
        HEX     60

        END
```

Label	Address ▲	Instruction	Hex
LOOP	000	EX PTR I	980C
	001	ADD ONE	000D
	002	EX PTR I	980C
	003	EX PTR	180C
	004	ADD ONE	000D
	005	EX PTR	180C
	006	EX CNT	180E
	007	ADD ONE	000D
	008	ST CNT	100E
	009	BR LOOP	0800
	00A	HLT	7000
	00B		0000
PTR	00C	HEX 10	0010
ONE	00D	HEX 1	0001
CNT	00E	HEX FFFC	FFFC
	00F		0000
	010	HEX 30	0030
	011	HEX 40	0040
	012	HEX 50	0050
	013	HEX 60	0060

Test Program2 image file name : ch7-Main-RAM-TEST2

제출물

- logisim 회로 파일 (파일이름: 학번-hw4.crc)
  - 마이크로 프로그램이 포함되어 실행 가능한 최종 logisim 파일
  - main 회로에 문서 정보 기록(HW4 .., 학번, 이름)

- Excel 파일
  - 마이크로 프로그램을 제작한 Excel 파일 (파일이름: 학번-hw4.xlsx)

- 마감일
  - 11월 15일(화) 23시59분