



7. 마이크로프로그램된 제어 Microprogrammed Control

목표

- 마이크로 프로그램을 이용한 제어 방법을 이해하고 설계할 수 있음.



マイクロ 프로그램된 제어 구조

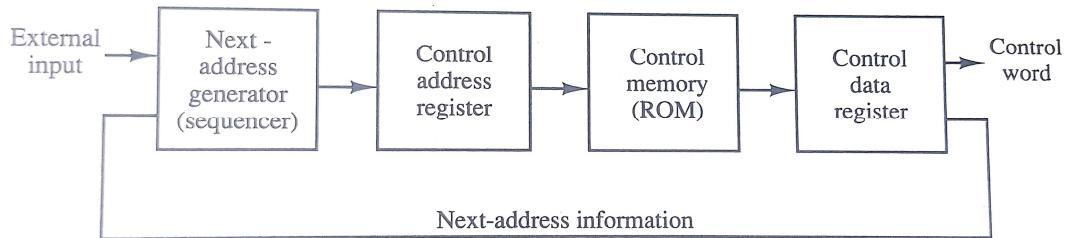


제어 장치

- 제어 신호 생성
 - 마이크로연산 수행을 위한 조건
- 하드와이어드 제어 장치
 - hardwired CU
 - 논리 회로 설계 방식에 따라 제어 신호 생성
- 마이크로 프로그램된 제어 장치
 - microprogrammed CU
 - 제어 기억 장치에 있는 제어워드를 이용 제어 신호 생성



마이크로프로그램된 제어장치



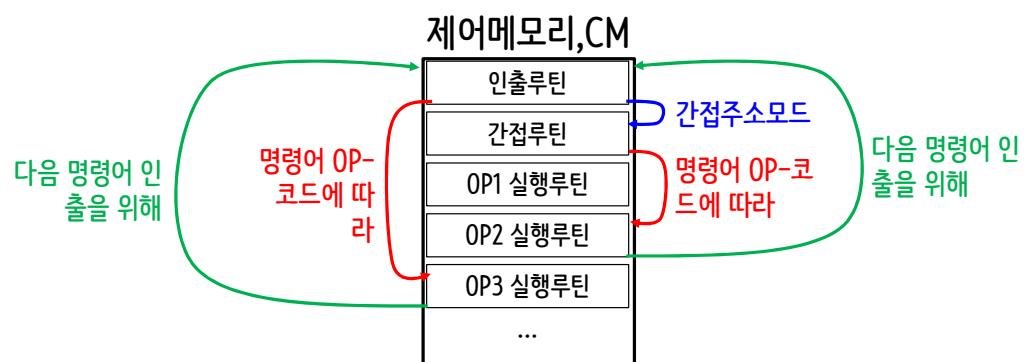
제어 메모리

- ROM으로 구성
- 마이크로 명령어 그룹 들로 구성
 - 명령어 인출 루틴
 - 유효 주소 결정 루틴
 - 명령어 연산 코드 실행 루틴, 등…

주소 시퀀싱



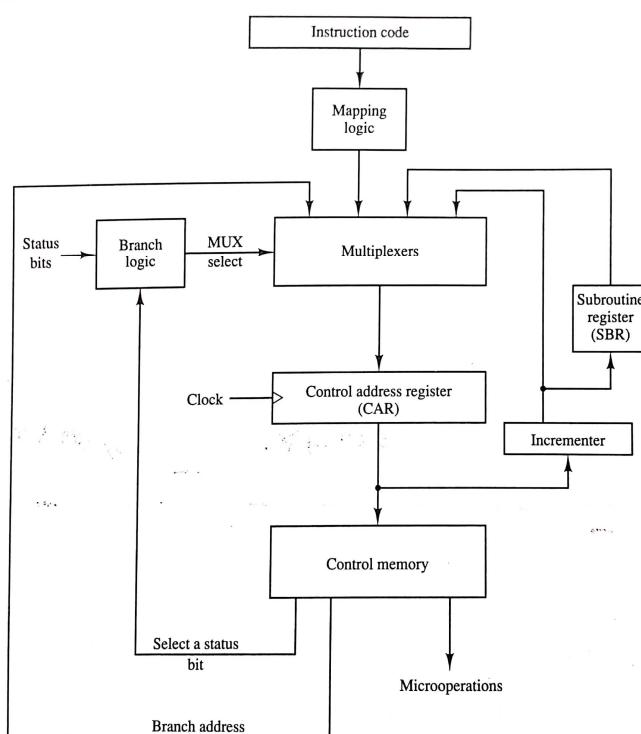
주소 시퀀싱



주소 시퀀싱 방법

- 제어 주소 레지스터 증가
- 무조건 분기/조건에 따른 조건부 분기
- 명령어 비트로부터 제어 메모리 주소 매핑
- 서브루틴 호출/복귀

제어 메모리 주소 선택



조건부 분기

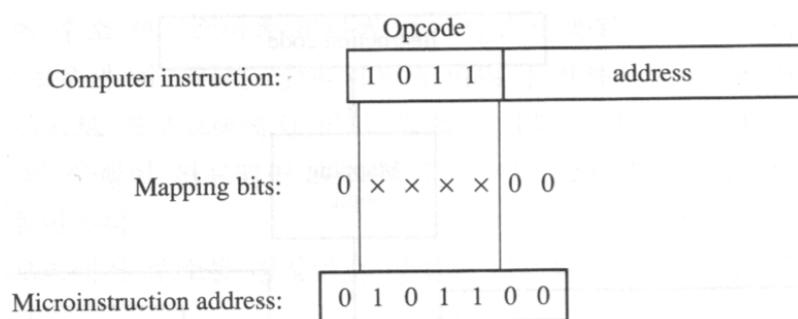
- 상태 비트에 따라 분기 논리(branch logic)에서 결정
- 조건에 따라 다음 마이크로 명령어 주소 결정

[프로그램에서의 분기와 마이크로프로그램에서의 분기는 구별되어야 함.](#)



매핑, mapping

- 명령어 OP-코드수행의 [실행 루틴의 주소](#) 결정
- 매핑 예 :



서브루틴

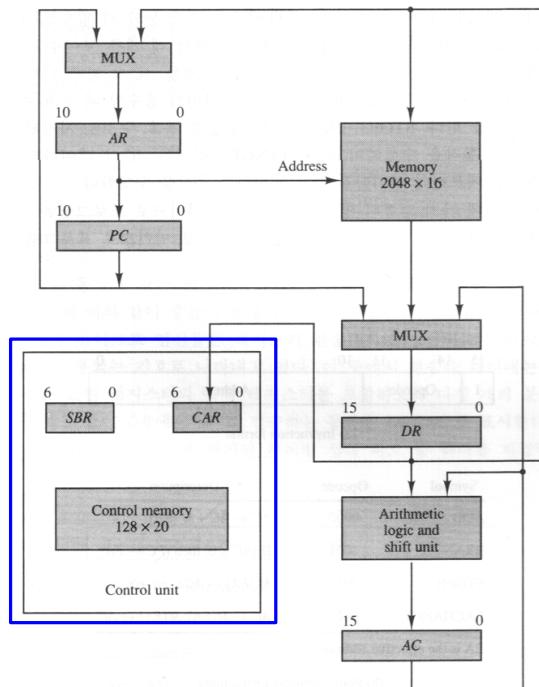
- 서브루틴에서 반환할 주소는 SBR
- ✓ 서브루틴 호출은 다른 마이크로프로그램의 루틴
 - 제어 메모리의 루틴 O
 - 주기억장치의 루틴 X



마이크로 프로그램의 예



타겟 컴퓨터 하드웨어 구성



명령어 형식



(a) Instruction format

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

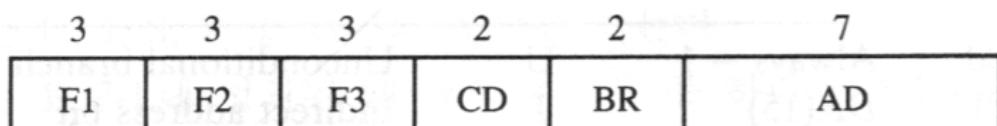
(b) Four computer instructions

マイクロ 명령어 형식

- 수평적 마이크로 명령어
 - 각 비트가 하나의 마이크로 연산을 제어
 - 고속 제어 신호 생성/마이크로 명령어 길이 증가
- 수직적 마이크로 명령어
 - 비트 그룹이 다수의 마이크로 연산 제어
 - 디코딩 회로 필요/마이크로 명령어 길이 감소



마이크로 명령어 형식



F1, F2, F3: Microoperation fields

CD: Condition for branching

BR: Branch field

AD: Address field

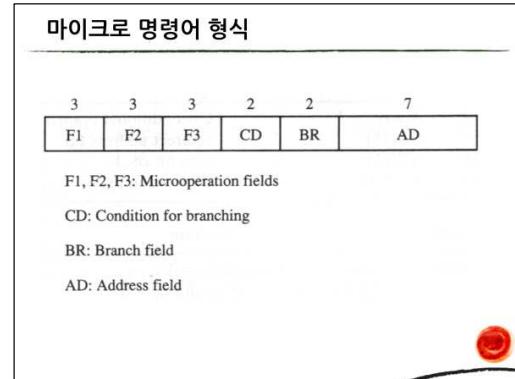


기호 마이크로 명령어 형식

F1	Microoperation	Symbol	F2	Microoperation	Symbol	F3	Microoperation	Symbol
000	None	NOP	000	None	NOP	000	None	NOP
001	$AC \leftarrow AC + DR$	ADD	001	$AC \leftarrow AC - DR$	SUB	001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow 0$	CLRAC	010	$AC \leftarrow AC \vee DR$	OR	010	$AC \leftarrow \overline{AC}$	COM
011	$AC \leftarrow AC + 1$	INCAC	011	$AC \leftarrow AC \wedge DR$	AND	011	$AC \leftarrow \text{shl } AC$	SHL
100	$AC \leftarrow DR$	DRTAC	100	$DR \leftarrow M[AR]$	READ	100	$AC \leftarrow \text{shr } AC$	SHR
101	$AR \leftarrow DR(0-10)$	DRTAR	101	$DR \leftarrow AC$	ACTDR	101	$PC \leftarrow PC + 1$	INCPC
110	$AR \leftarrow PC$	PCTAR	110	$DR \leftarrow DR + 1$	INCDR	110	$PC \leftarrow AR$	ARTPC
111	$M[AR] \leftarrow DR$	WRITE	111	$DR(0-10) \leftarrow PC$	PCTDR	111	Reserved	

CD	Condition	Symbol	Comments
00	Always = 1	U	Unconditional branch
01	$DR(15)$	I	Indirect address bit
10	$AC(15)$	S	Sign bit of AC
11	$AC = 0$	Z	Zero value in AC

BR	Symbol	Function
00	JMP	$CAR \leftarrow AD$ if condition = 1 $CAR \leftarrow CAR + 1$ if condition = 0
01	CALL	$CAR \leftarrow AD$, $SBR \leftarrow CAR + 1$ if condition = 1 $CAR \leftarrow CAR + 1$ if condition = 0
10	RET	$CAR \leftarrow SBR$ (Return from subroutine)
11	MAP	$CAR(2-5) \leftarrow DR(11-14)$, $CAR(0,1,6) \leftarrow 0$



명령어 인출 루틴

- Fetch 루틴
- 메모리에 저장된 명령어 인출
- 제어 메모리 64번지 부터 적재



명령어 인출 루틴

$AR \leftarrow PC$

$DR \leftarrow M[AR], PC \leftarrow PC + 1$

$AR \leftarrow DR(0-10), CAR(2-5) \leftarrow DR(11-14), CAR(0, 1, 6) \leftarrow 0$

ORG 64

FETCH : PCTAR U JMP NEXT
READ,INCPC U JMP NEXT
DRTAR U MAP

기호로 표시된 마이크로프로그램



명령어 인출 루틴

ORG 64

FETCH : PCTAR U JMP NEXT
READ,INCPC U JMP NEXT
DRTAR U MAP

Binary

Address	F1	F2	F3	CD	BR	AD
1000000	110	000	000	00	00	1000001
1000001	000	100	101	00	00	1000010
1000010	101	000	000	00	11	0000000

명령어 인출 마이크로프로그램 루틴의 이진형식(제어메모리에 적재되는 형식)



간접주소 마이크로프로그램 루틴

INDRCT : READ U JMP NEXT
DRTAR U RET

마이크로 프로그램 -기호표기

	Label	Microoperations	CD	BR	AD
ADD:	ORG 0 NOP READ ADD		I U U	CALL JMP JMP	INDRCT NEXT FETCH
BRANCH:	ORG 4 NOP NOP		S U	JMP JMP	OVER FETCH
OVER:	NOP ARTPC		I U	CALL JMP	INDRCT FETCH
STORE:	ORG 8 NOP ACTDR WRITE		I U U	CALL JMP JMP	INDRCT NEXT FETCH
EXCHANGE:	ORG 12 NOP READ ACTDR, DRTAC WRITE		I U U U	CALL JMP JMP JMP	INDRCT NEXT NEXT FETCH
FETCH:	ORG 64 PCTAR READ, INCPC		U U	JMP JMP	NEXT NEXT
INDRCT:	DRTAR READ DRTAR		U U U	MAP JMP RET	

マイ クロ プロ グラム イジン 表記

Micro Routine	Address			Binary Microinstruction					
	Decimal	Binary		F1	F2	F3	CD	BR	AD
ADD	0	0000000	000	000	000	01	01	1000011	
	1	0000001	000	100	000	00	00	0000010	
	2	0000010	001	000	000	00	00	1000000	
	3	0000011	000	000	000	00	00	1000000	
BRANCH	4	0000100	000	000	000	10	00	0000110	
	5	0000101	000	000	000	00	00	1000000	
	6	0000110	000	000	000	01	01	1000011	
	7	0000111	000	000	110	00	00	1000000	
STORE	8	0001000	000	000	000	01	01	1000011	
	9	0001001	000	101	000	00	00	0001010	
	10	0001010	111	000	000	00	00	1000000	
	11	0001011	000	000	000	00	00	1000000	
EXCHANGE	12	0001100	000	000	000	01	01	1000011	
	13	0001101	001	000	000	00	00	0001110	
	14	0001110	100	101	000	00	00	0001111	
	15	0001111	111	000	000	00	00	1000000	
FETCH	64	1000000	110	000	000	00	00	1000001	
	65	1000001	000	100	101	00	00	1000010	
	66	1000010	101	000	000	00	11	0000000	
INDRCT	67	1000011	000	100	000	00	00	1000100	
	68	1000100	101	000	000	00	10	0000000	

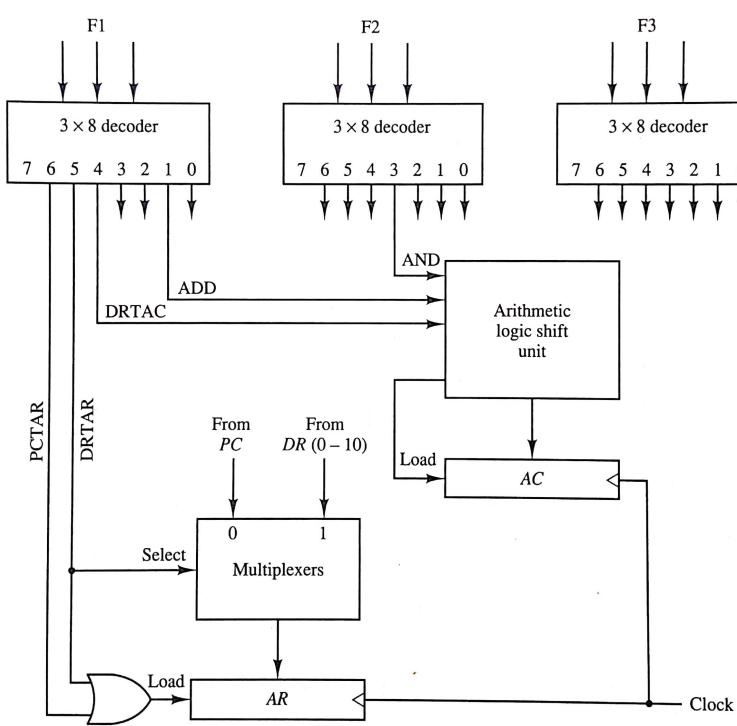
제어메모리에 적재되는 형식

제어메모리 영역

ADD	0~3
BRANCH	4~7
STORE	8~11
EXCHANGE	12~15
[]	
FETCH	64~66
INDRCT	67~68
[]	127

제어장치 설계

제어장치 설계-디코딩



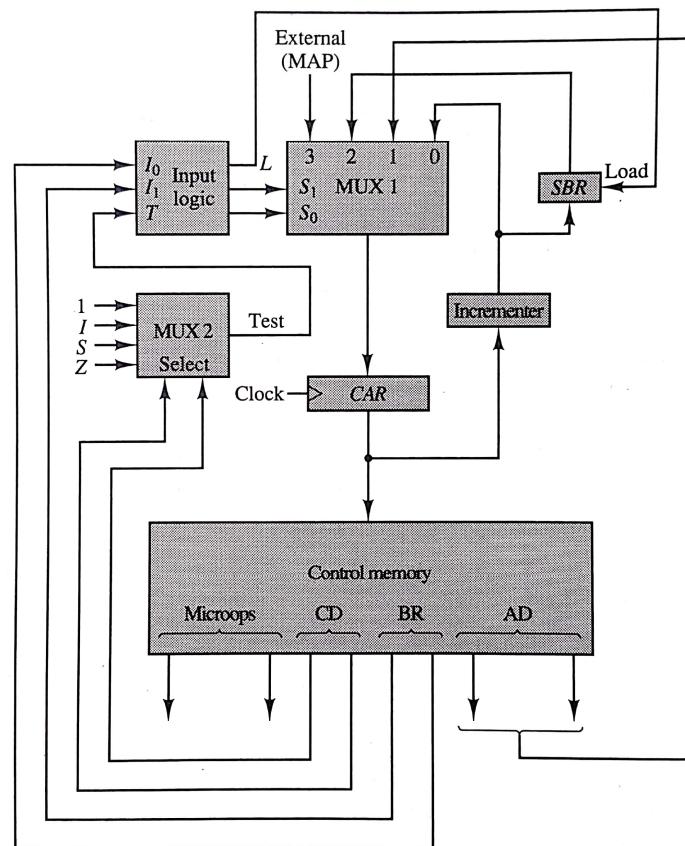
마이크로 프로그램 시퀀서

- 다음 마이크로명령어 주소 결정 로직

조합논리회로

- 입력 : BR 필드와 조건(I, S, Z, 1) 검사결과
- 출력 : MUX1의 선택 라인, SBR의 load

마이크로 프로그램 시퀀서



제어장치 설계-마이크로프로그램 시퀀스

BR Field	Input			MUX 1		Load	SBR
	I_1	I_0	T	S_1	S_0	L	/
0 0	0	0	0	0	0	0	
0 0	0	0	1	0	1	0	
0 1	0	1	0	0	0	0	
0 1	0	1	1	0	1	1	
1 0	1	0	x	1	0	0	
1 1	1	1	x	1	1	0	

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



Q&A

