

Slide 0**7장. 마이크로 프로그램된 제어****제어 메모리**

- 제어장치 : 마이크로 연산들을 수행하게 하는 신호를 생성
- 제어 장치의 종류
 - 하드와이어드 (hard wired) : 조합회로로 구현 (5장에서 설명)
 - 마이크로 프로그램 방식 (microprogrammed control)
 - * 명령어 수행에 대한 제어워드를 기억장치속에 저장
 - * 기계어 수준의 명령에 대하여 마이크로 명령어라고 부름
 - * 보통 CPU 내부 ROM에 마이크로 명령어들이 저장됨

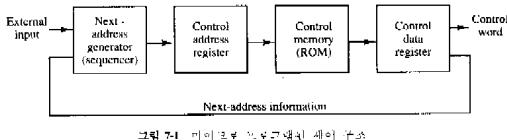
Slide 1

- 하드와이어드 (hard wired) : 조합회로로 구현 (5장에서 설명)
- 마이크로 프로그램 방식 (microprogrammed control)
 - * 명령어 수행에 대한 제어워드를 기억장치속에 저장
 - * 기계어 수준의 명령에 대하여 마이크로 명령어라고 부름
 - * 보통 CPU 내부 ROM에 마이크로 명령어들이 저장됨

제어 메모리 (계속)

* 마이크로 프로그램된 제어구조 (그림 7-1)

Slide 2



- * 장점 : ROM 프로그램의 변경으로 새로운 제어순서 생성 가능
- * 단점 : hard wired 방식에 비해 속도가 느리다

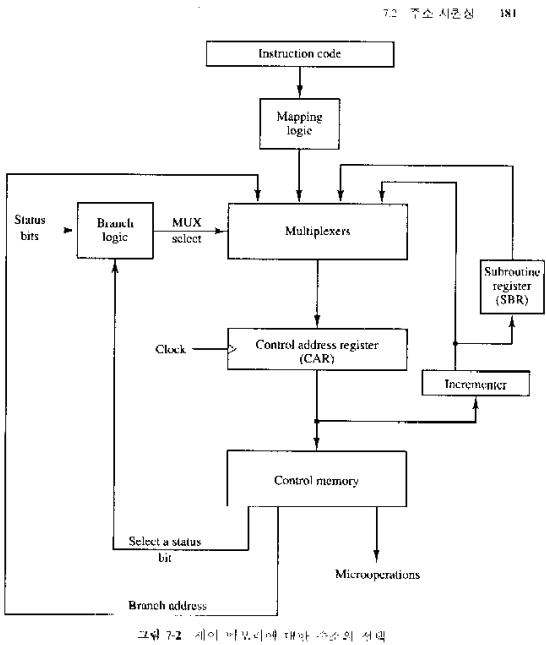
주소 시퀀싱

Slide 3

- 동작
 - 초기주소 : 보통 fetch 를 수행하는 마이크로 루틴을 가짐
 - 유효주소 계산 : fetch 후에 주소 모드별로 계산
 - macro operation 수행 : 명령어의 OP-code 부분을 해석 (mapping 이라함) 동작
 - 다음 명령어 fetch
- 제어메모리에서 주소를 결정하는 방법들
 1. 제어 주소 레지스터를 하나 증가시킴
 2. 무조건 분기와 상태 비트 조건에 따른 조건부 분기
 3. 명령어의 비트들로부터 제어 메모리의 주소로 맵핑 처리
 4. 서브루틴을 호출하고 (call), 복귀하는 (return) 기능
- 제어 메모리의 주소 선택 (그림 7-2)

주소 시퀀싱 (계속)

Slide 4



주소 시퀀싱 (계속)

Slide 5

- 조건 부 분기
 - 조건에 따라서 분기가 일어남
 - 조건에 따라서 주소 선택 MUX의 입력을 바꿈
 - 무조건 분기는 상태 비트를 1로 고정하고 이 비트를 선택하여 구현 가능
- 명령어의 매핑
 - macro 명령어에 대한 해당 micro 명령어를 찾는 방법
 - macro 명령어의 OP-code 부분을 이용
 - 매핑 방법의 한 예 (그림 7-3)

주소 시くん싱 (계속)

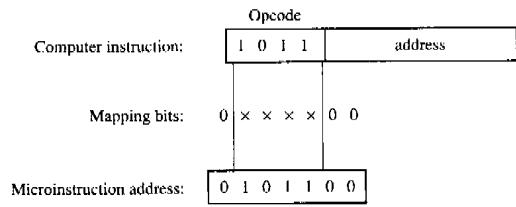


그림 7-3 명령어 코드로부터 제이 메모리 주소로의 매펑

Slide 6

- * 첨가된 하위 두 비트를 이용 매크로 명령어당 4개의 마이크로 명령어 구현 가능
 - * 첨가된 최 상위 한비트를 이용 4개 이상의 마이크로 명령어도 구현 가능
 - * 매펑 테이블을 ROM에 저장하여 임의의 위치에 마이크로 명령어를 둘 수 있음

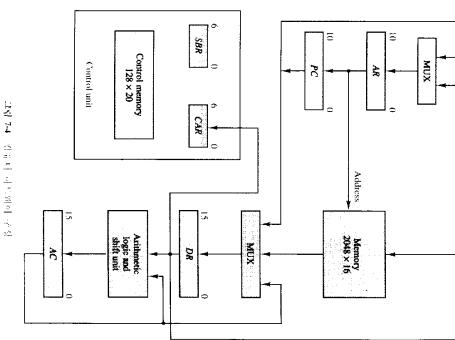
- 서브루틴

- 동일한 연산들을 서브루틴으로 만듦 → 제어 메모리 절약됨

마이크로 프로그램의 예

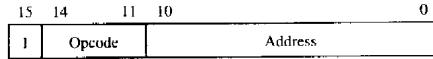
- 컴퓨터 하드웨어 구성 (그림 7-4)

Slide 7



- 컴퓨터 명령어 형식 및 4개의 메모리 참조 명령어 (그림 7-5)

마이크로 프로그램의 예 (계속)



(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

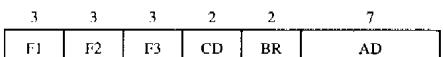
그림 7-5 마이크로 프로그램 예

Slide 8

- 마이크로 명령어 형식

마이크로 프로그램의 예 (계속)

- 형식 (그림 7-6)



F1, F2, F3: Microoperation fields

CD: Condition for branching

BR: Branch field

AD: Address field

그림 7-6 마이크로 명령어 구조 형식(20비트)

Slide 9

- 주소 비트가 7개이므로 제어 메모리 크기는 $2^7 = 128$
- 마이크로 명령어의 각 필드에 대한 기호와 이진 코드 (표 7-1)

마이크로 프로그램의 예 (계속)

표 7-1 마이크로 명령어의 기호화된 형식과 미시오 명령		
F1	Microoperation	Symbol
000	$AC \leftarrow AC + DR$	ADD
001	$AC \leftarrow 0$	CLRAC
010	$AC \leftarrow AC + 1$	INCAC
011	$AC \leftarrow AC - DR$	DRTAC
100	$AC \leftarrow M[AR]$	LOAD
101	$DR \leftarrow M[AR]$	PCTAD
110	$AC \leftarrow PC$	WRTE
111	$M[AR] \leftarrow DR$	WRTE

F2	Microoperation	Symbol
000	Noop	NOP
001	$AC \leftarrow AC \vee DR$	SUB
010	$AC \leftarrow AC \wedge DR$	OR
011	$AC \leftarrow AC \wedge DR$	AND
100	$DR \leftarrow M[AR]$	RDADR
101	$DR \leftarrow DR + 1$	INCDB
110	$DR \leftarrow DR + 1$	INCTDB
111	$DR(0\sim10) \leftarrow PC$	PCTDB

F3	Microoperation	Symbol	Comment
000	Noop	NOP	
001	$AC \leftarrow AC \wedge DR$	XOR	
010	$AC \leftarrow AC$	COM	
011	$AC \leftarrow \text{Not } AC$	NOT	
100	$AC \leftarrow DR$	SWTCH	
101	$PC \leftarrow DR + 1$	INSTR	
110	$PC \leftarrow DR$	ARTR	
111	Reserved		

CD	Condition	Symbol	Function
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

DR	Symbol	Function
00	JMP	$CAR \leftarrow DR$ if condition = 0 $CAR \leftarrow CAR + 1$ if condition = 1
01	CALL	$CAR \leftarrow DR$, $M[AR] \leftarrow AC$ if condition = 1 $CAR \leftarrow CAR + 1$ if condition = 0
10	RTT	$CAR \leftarrow SBR$ (Return from subroutine)
11	RET	$CAR(2\sim5) \leftarrow DR(11\sim14)$, $CAR(0\sim6) \leftarrow 0$

Slide 10

마이크로 프로그램의 예 (계속)

- 총 $3 \times 7 = 21$ 개의 마이크로 연산 표시 가능 (000은 사용안함)
- F1, F2, F3의 세개의 마이크로 연산을 하나의 마이크로 명령어로 지정 가능
- 예) F1 = 000, F2 = 100, F3 = 101 일때

$DR \leftarrow M[AR]$ (F2 = 100)
 $PC \leftarrow PC + 1$ (F3 = 101)

\rightarrow F1, F2, F3 가 010 001 000 은 동시에 지정 불가능

Slide 11

- 기호로 표시된 마이크로 명령어
 - 마이크로 명령어를 기호로 표시
 - 어셈블러 이용하여 이진코드로 바꿀수 있음
 - 기호로 표시된 명령어의 형식
 1. 라벨필드는 공백 혹은 기호주소 가능, 라벨뒤에는 콜론(:)이 있어야 함
 2. 연산필드는 표 7-1에 정의된 한개나 두개 또는 세개의 기호로 구성, 기호는 콤마(,)로 분리
 3. CD 필드는 U, I, S, Z 중 하나의 문자

마이크로 프로그램의 예 (계속)

4. BR 필드는 JMP, CALL, RET, MAP 중 하나
5. AD 필드는 다음 세가지중 하나
 - * 기호 주소
 - * NEXT (다음 주소)
 - * BR 이 RET이나 MAP이면 공란
6. ORG 제어주소의 시작번지 알림

Slide 12

- Fetch 루틴

* fetch 에 필요한 micro 명령어

```
AR ← PC
DR ← M[AR], PC ← PC + 1
AR ← DR(0-10), CAR(2-5) ← DR(11-14),
CAR(0,1,6) ← 0
```

* fetch 를 어셈블리어로 구성

마이크로 프로그램의 예 (계속)

```
ORG 64
FETCH: PCTAR U JMP NEXT
        READ, INCPC U JMP NEXT
        DRTAR U MAP
```

* 어셈블러로 만든 이진 코드

Slide 13

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

- 기호로 표시된 마이크로 프로그램
 - indirect 주소일 경우는 유효 주소를 계산해야함
 - 공통적으로 사용됨으로서 브루틴으로 구성
 - 마이크로 명령어

```
INDRCT: READ U JMP NEXT
        DRTAR U RET
```

마이크로 프로그램의 예 (계속)

- 몇 가지 명령어의 마이크로 프로그램 (표 7-2)

Label	표 7-2 각 명령어의 마이크로 프로그램					
	Microoperations	CD	BR	AD		
ADD:	ORG 0 NOP READ ADD	I U U U	CALL INDRECT NEXT JMP			
BRANCH:	ORG 4 NOP NOP OVER: OVER: ARTIC	S U I U U	JMP CALL INDRECT FETCH			
STORE:	ORG 8 NOP ACTDR WRITE	I U U	CALL INDRECT NEXT FETCH			
EXCHANGE:	ORG 12 NOP READ, READ, DRTAC WHITE	I U U U	CALL INDRECT NEXT NEXT NEXT FETCH			
INDRCT:		RPT	JMP			

Slide 14

- 이진 마이크로 프로그램

- 어셈블러나 사람이 기호로 표시된 마이크로 프로그램을 이진 코드로 변환
- 표 7-2에 대한 이진 마이크로 프로그램 (표 7-3)

마이크로 프로그램의 예 (계속)

- 이진 마이크로 프로그램은 제어메모리 (ROM)에 저장됨

표 7-3 세이 메모리를 위한 이진 마이크로 프로그램

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	0001000	000	000	000	10	00	0000110
	5	0001001	000	000	000	00	00	1000000
	6	0001010	000	000	000	01	01	1000011
	7	0001011	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	1000010
	68	1000100	101	000	000	00	10	0000000

Slide 15

제어 장치의 설계

Slide 16

- 마이크로 프로그램에 따라서 실행되게끔 제어 장치를 설계함
- F1,F2,F3 의 세연산자 필드의 연산을 수행하기 위한 decode 회로 (그림 7-7)

제어 장치의 설계 (계속)

Slide 17

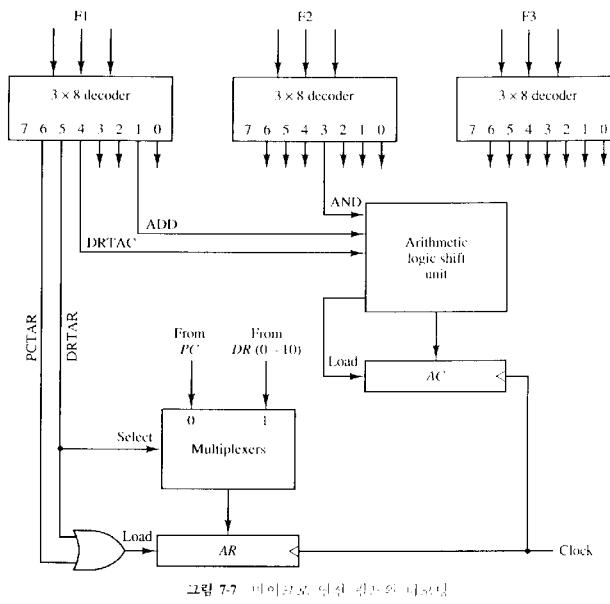


그림 7-7 마이크로 프로그램된 제어 장치의 설계

제어 장치의 설계 (계속)

- 예) F1 = 101 (DRTAR), 110 (PCTAR) 를 수행시키기 위한 회로
- 마이크로 프로그램 시퀀서

Slide 18

- 마이크로 프로그램 방식의 기본요소
 1. 제어메모리
 2. 마이크로 프로그램 시퀀서
- 마이크로 프로그램 시퀀서
 - * 마이크로 연산에서 다음 주소를 결정하는 회로
 - * 현재 진행중인 명령어의 주소 비트 상태 비트에 따라서 다음 주소가 결정됨
 - * 마이크로 프로그램 시퀀서 (그림 7-8)

제어 장치의 설계 (계속)

Slide 19

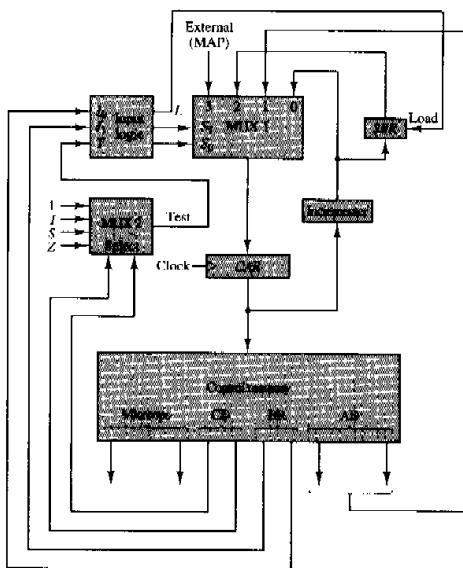


그림 7-8 차어 메모리 를 위한 마이크로 프로그램 시퀀서

제어 장치의 설계 (계속)

* 마이크로 프로그램 시퀀서를 위한 입력 논리 진리표 (표 7-4)

Slide 20

* 멀티플렉서 선택 신호를 위한 부울식

$$S_1 = I_1$$

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

$$L = I_1' I_0 T$$