

제5장 제어 장치



제어장치의 개요

- **제어장치의 정의**

컴퓨터시스템의 모든 장치들을 유기적으로 제어하는 장치

- **제어장치의 역할**

- ▶ **주기억장치에 기억된 프로그램으로부터 명령어를 순차적으로 하나씩 읽어내어 해독한 후, 해당되는 여러 구성요소를 동작시키는 제어신호를 내보낸다.**



제어장치의 개요

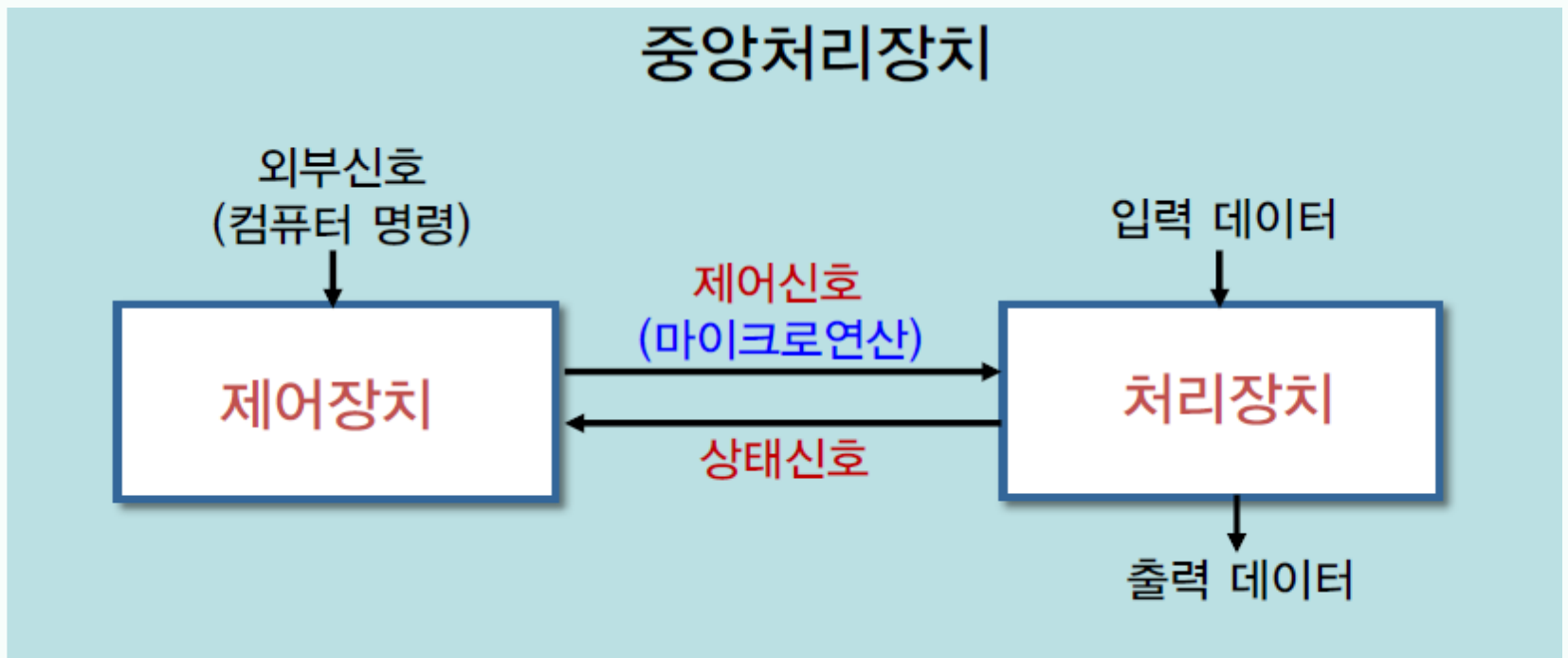
● 제어장치의 역할

- ▶ 하나의 명령어는 한 클록 펄스 기간 동안에 수행되는 마이크로 연산의 집합으로 구성
- ▶ 명령어는 마이크로 연산의 연속적인 반복에 의해 실행
- ▶ 제어장치는 요구되는 마이크로 연산을 연속적으로 수행하게 하는 신호를 보내어 명령어를 수행하는 역할



제어장치의 개요

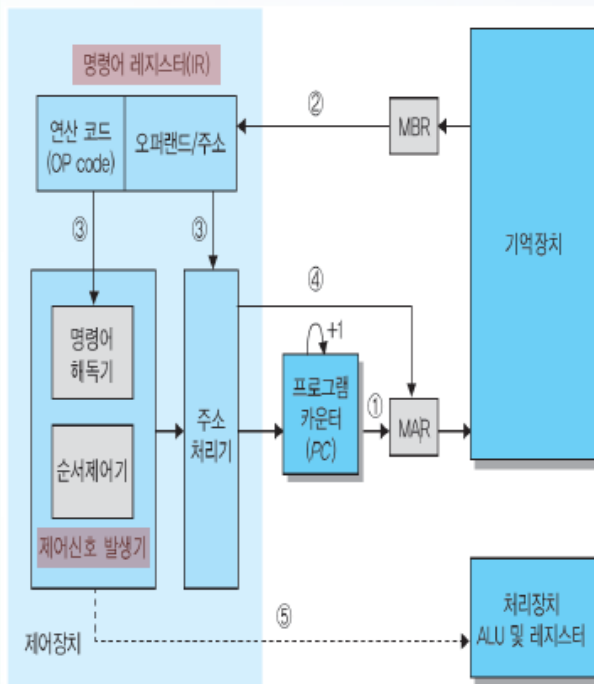
- 제어장치와 처리장치와의 관계





제어장치의 구성

■ 제어장치의 명령어 수행과정



〈제어장치의 세부 구성〉

✓ 먼저 수행할 명령이 들어 있는 기억장치의 주소가 프로그램 카운터(PC)에 기억되어 있다고 가정

- ① PC에 저장된 주소를 **메모리 주소 레지스터(MAR)**로 보낸다.
- ② MAR에 저장된 주소에 있는 기억장치의 명령어를 **메모리 버퍼 레지스터(MBR)**로 읽어 오고, 프로그램 카운터를 1 증가시킨다. 또한 명령어의 실행을 위해 읽어 온 명령어를 **명령어 레지스터(R)**에 저장한다.
- ③ 명령어는 연산코드와 오퍼랜드로 구성되어 있으며, 연산코드는 **명령어 해독기**로 보내고, 오퍼랜드는 **주소처리기**로 보낸다.
- ④ 주소처리기는 **메모리 주소 레지스터(MAR)**를 통해 명령어 수행에 필요한 오퍼랜드의 주소 또는 다음 명령어의 주소를 계산하며, 이를 위해 기억장치에 접근한다.
- ⑤ 제어신호 발생기에서는 위의 ①~④까지의 과정에서 필요한 제어신호와 연산코드를 해독하여 명령수행을 위한 **제어신호**를 발생시킨다.
- ⑥ 현재 명령어 레지스터에 있는 명령어의 수행이 끝나면 증가된 PC의 내용은 다음에 수행할 명령어의 주소를 지정하거나 점프 명령어와 같이 분기가 필요한 경우에는 해당 명령어의 주소를 지정하는데 이것은 **다시 ①부터의 과정을 반복**하며 실행된다.



제어장치의 구현

- 마이크로프로그램에 의한 제어장치(microprogrammed control device)
 - 제어단어와 같은 제어정보를 특별한 기억장치에 기억시킨 구조
 - 프로그램에 의한 제어방식
 - 장점: 제어신호를 수정하고자 할 때 융통성이 뛰어나
 - 단점: 기억장치에서 명령을 인출하는 별도의 시간 때문에 속도가 느림
 - 고밀도 집적회로(VLSI)가 가능하여 대부분의 컴퓨터에서 사용
- 하드웨어로 구현한 제어장치(hardwired control device)
 - 순서회로와 같은 게이트와 플립플롭(flip-flop)으로 이루어짐
 - 장점: 순서회로로 구성되기 때문에 고속 동작이 가능
 - 단점: 회로의 동작방법이 바뀌면 다시 설계해야 함
 - 특수한 컴퓨터에서 사용

마이크로프로그램에 의한 제어



마이크로프로그램에 의한 제어

● 제어방식

- 제어변수 값들을 여러 개의 단어로 결합하여 표현하고, 그것을 연속적으로 읽어서 제어논리회로에 공급하여 동작시키는 방식

● 제어장치의 역할

- 일련의 마이크로 연산을 순차적으로 수행하도록 제어.
- 이러한 제어를 위해 제어변수로서 제어단어를 사용



마이크로프로그램에 의한 제어장치

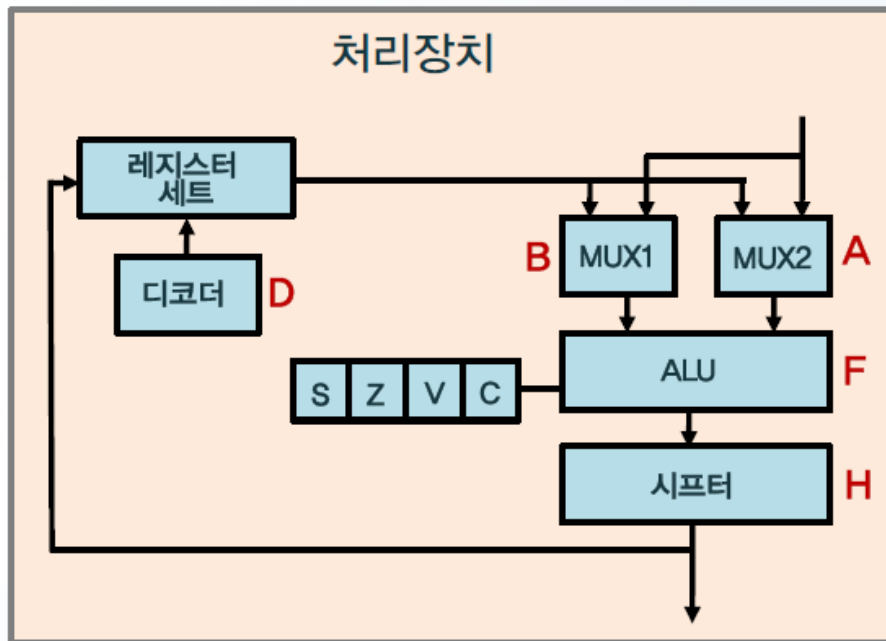
- 2진 제어변수를 기억장치에 저장하여 구현한 제어장치
 - 마이크로프로그램으로 구현
- 마이크로프로그램
 - ▶ 마이크로 명령 (microinstruction)
 - 제어기억장치에 있는 각 제어단어에는 마이크로명령이 포함.
 - 마이크로명령은 하나 혹은 다수의 마이크로연산을 나타냄.
 - ▶ 마이크로 프로그램(microprogram)
 - 순차적인 마이크로명령들의 집합
 - 마이크로프로그램은 일반적으로 ROM에 저장
 - ROM의 각 단어내용은 처리장치에서 수행할 마이크로 연산을 나타냄

마이크로연산 \subseteq 마이크로명령어 \subseteq 마이크로프로그램



마이크로프로그램에 의한 제어장치

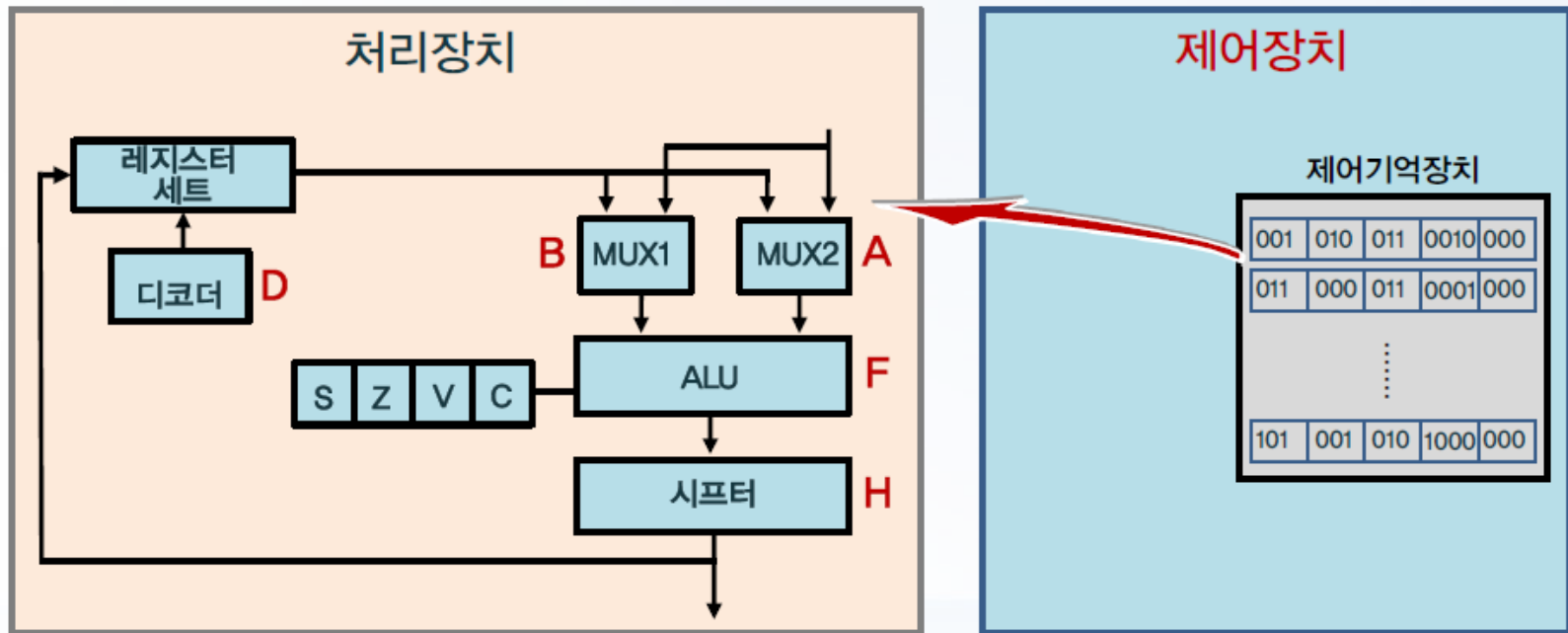
❖ 제어단어와 처리장치



〈제어단어〉

마이크로프로그램에 의한 제어장치

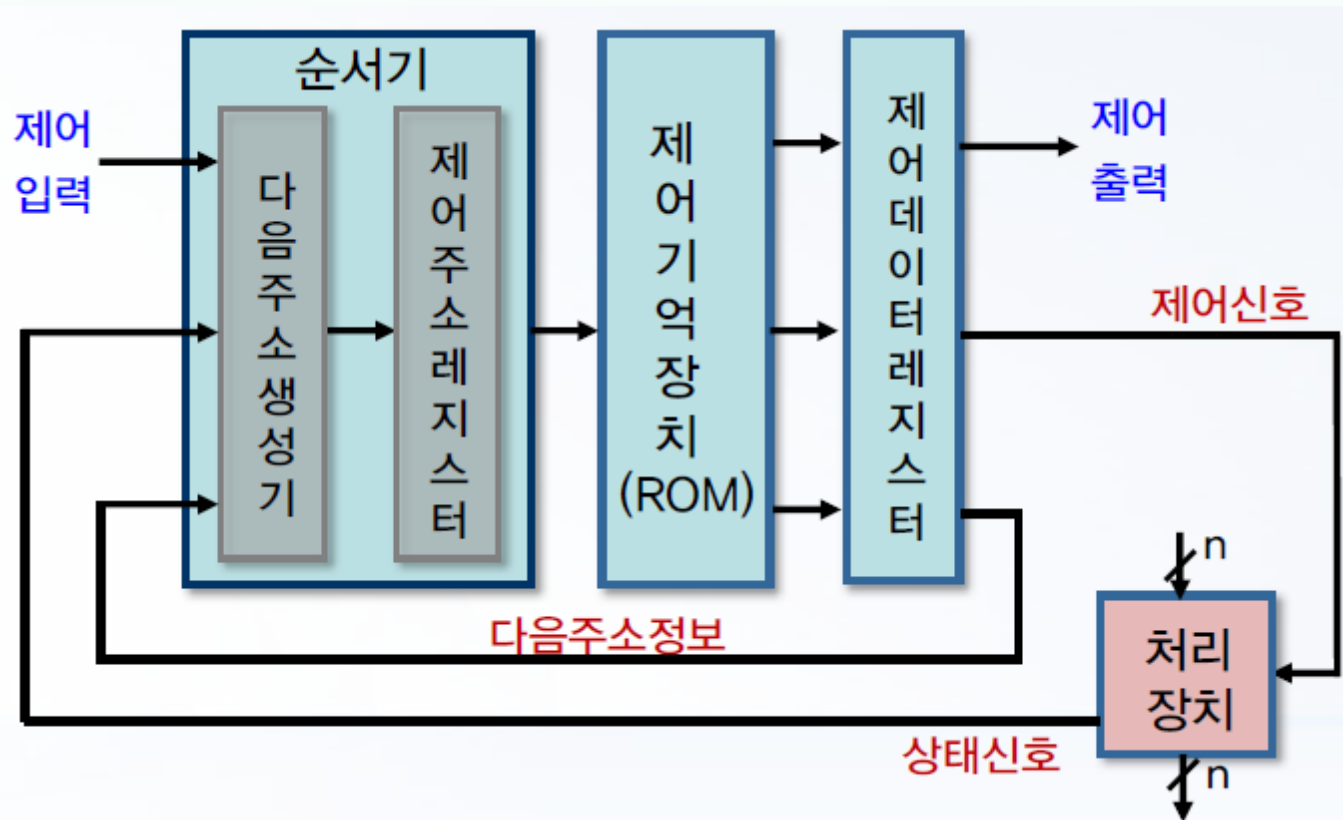
❖ 제어장치와 처리장치





마이크로프로그램에 의한 제어장치 구조

구성도(1)





마이크로프로그램에 의한 제어장치의 구조

구성요소

▶ 제어기억장치

- 마이크로명령들이 저장되어 있다.
- 저장된 마이크로명령은 하나 또는 다수의 마이크로연산을 나타내는 제어단어를 포함하고 있다.
- 마이크로명령이 모두 수행되고 나면, 다음에 수행될 마이크로 명령의 주소를 결정해야 한다.
- 다음 마이크로명령의 주소를 결정하기 위해서는 주소 정보가 현재의 마이크로명령 내에 한 비트 혹은 여러 비트로 있어야 한다.



마이크로프로그램에 의한 제어장치의 구조

구성요소

▶ 다음주소생성기(순서기)

- 다음에 수행될 마이크로명령의 주소를 만들어 낸다.
- 이 주소는 다음 클럭 펄스가 입력될 때 CAR로 이동되어 ROM으로부터 다음 마이크로명령을 읽어내는데 사용된다.
- 마이크로명령내에는 처리장치의 마이크로연산을 실행시키는 제어 비트와 실행될 마이크로명령의 순서를 정하는 비트가 포함되게 된다.

● 마이크로프로그램에 의한 제어장치의 구조

● 구성요소

▶ 다음주소생성기(순서기)

– 다음 마이크로명령의 주소는 순서기의 입력에 따라 다양한 방법으로 결정 될 수 있다.

- ① CAR의 1 증가
- ② 제어기억장치의 주소를 CAR로 적재
- ③ 외부로부터 온 주소를 CAR에 적재
- ④ 제어동작을 시작하는 초기주소 적재



마이크로프로그램에 의한 제어장치의 구조

● 구성요소

▶ 제어주소레지스터(CAR)

- 다음에 읽을 마이크로명령의 주소를 제어하는 레지스터

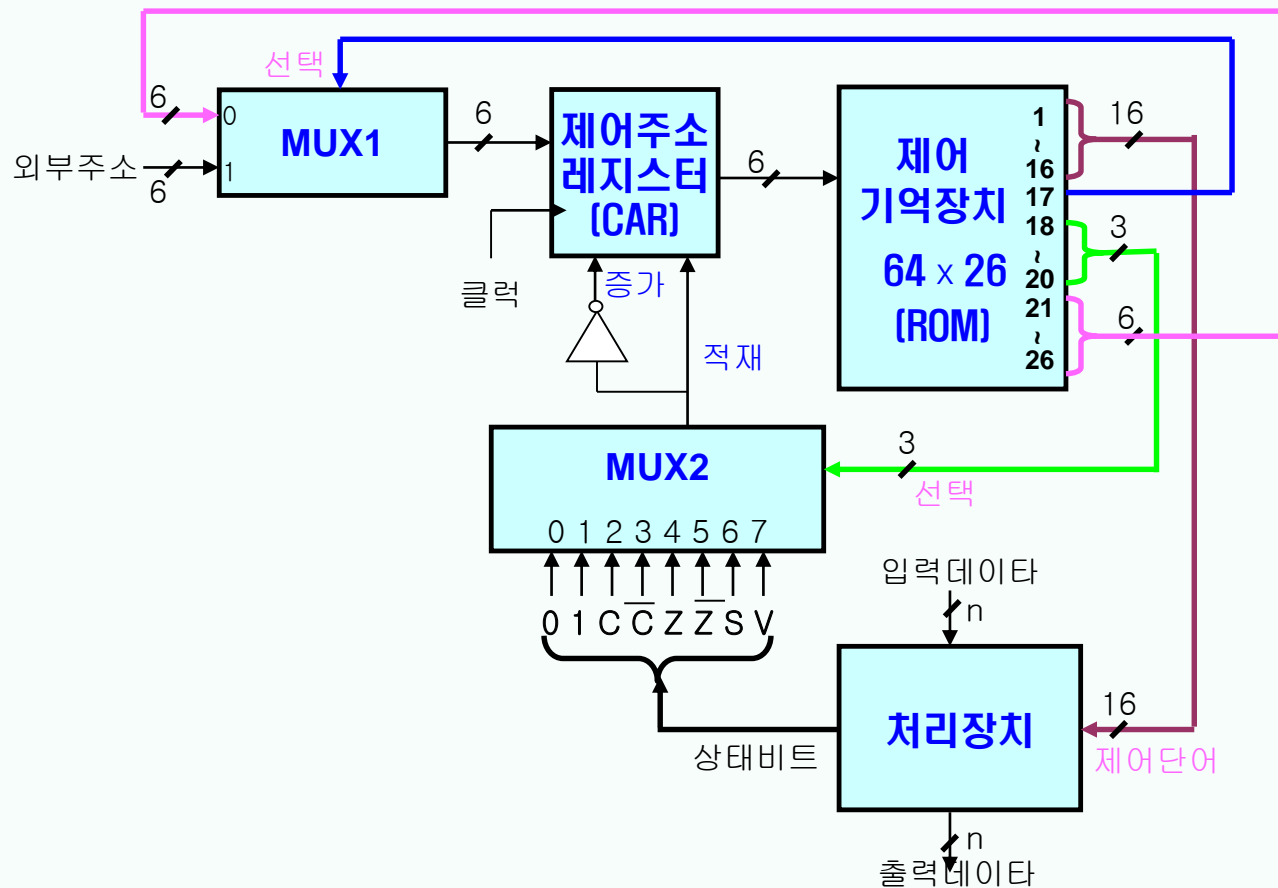
▶ 제어데이터레지스터(CDR)

- 다음 주소가 계산되어 다음 마이크로명령이 제어기억장치로부터 읽혀질 때까지 현재의 마이크로명령을 저장한다.



마이크로프로그램에 의한 제어장치 구조

구성도[2]





제어장치의 구성요소

● 처리장치

- ▶ 레지스터 세트[R1~R7]
- ▶ 산술논리연산장치
- ▶ 쉬프트
- ▶ 상태 레지스터[C, Z, S, V]

● 제어장치

- ▶ 제어기억장치[64X26 ROM]
- ▶ 제어주소레지스터[CAR]
- ▶ MUX1
- ▶ MUX2



마이크로명령

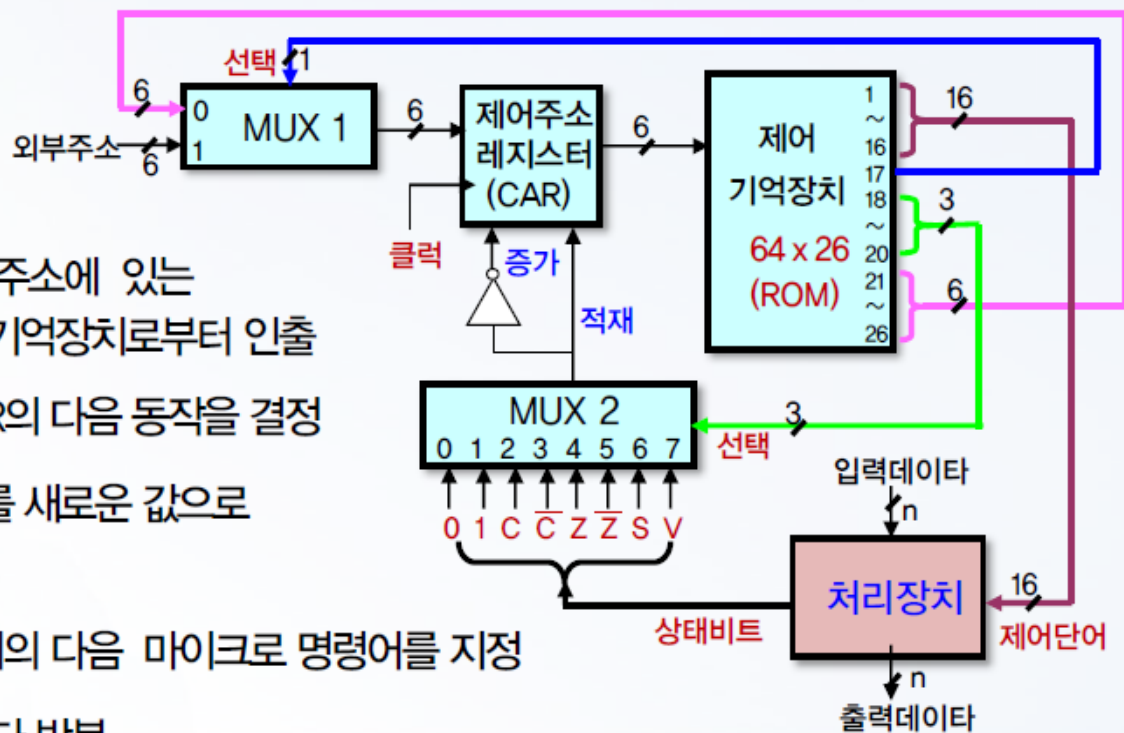
- 제어기억장치에 저장된 제어 명령
- 마이크로명령의 내역
 - ▶ 모두 26비트로 구성됨
 - 처리장치 제어를 위한 16비트
 - 다음주소 선택을 위한 10비트



제어장치의 동작

❖ 제어장치의 동작

- 1) CAR에 새로운 주소가 들어오면, 이 주소에 있는 26 비트의 마이크로 명령어를 제어기억장치로부터 인출
- 2) 마이크로 명령어의 제어단어는 CAR의 다음 동작을 결정
- 3) 처리장치에서 해당하는 상태 비트를 새로운 값으로 갱신하며, 다음 주소를 CAR로 전송
- 4) CAR의 새로운 주소는 제어기억장치의 다음 마이크로 명령어를 지정
- 5) 이러한 과정을 각 클럭의 천이 때마다 반복



마이크로 명령어 형식



마이크로명령 형식

- 구성 내역

- ▶ 8개의 필드로 구성

A	B	D	F	H	MUX1	MUX2	ADRS
3비트	3비트	3비트	4비트	3비트	1비트	3비트	6비트



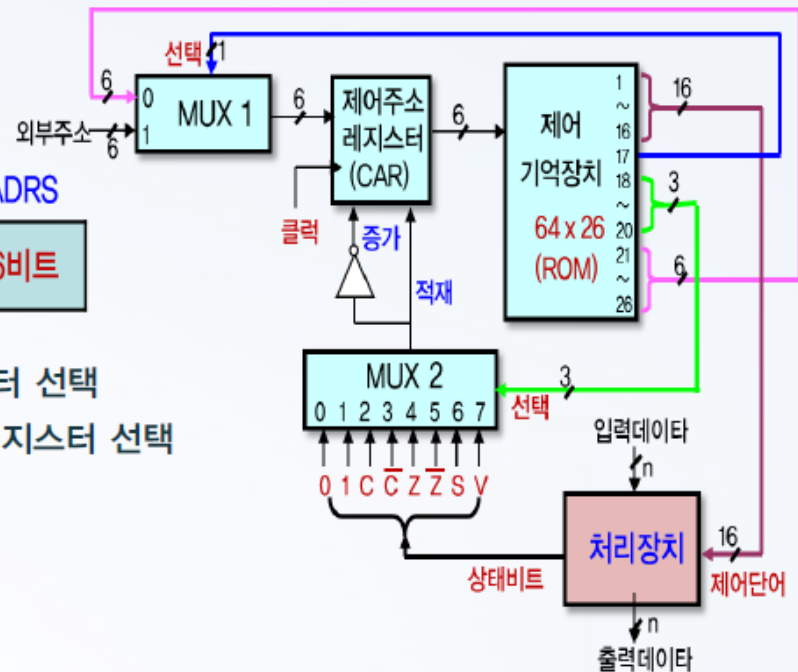
마이크로명령 형식

❖ 구성내역 예(26비트의 경우)

> 8개의 필드로 구성

A	B	D	F	H	MUX 1	MUX 2	ADRS
3비트	3비트	3비트	4비트	3비트	1비트	3비트	6비트

- ✓ **A 필드**: 처리장치의 레지스터 중 하나의 출발 레지스터 선택
- ✓ **B 필드**: 처리장치의 레지스터 중 다른 하나의 출발 레지스터 선택
- ✓ **D 필드**: 처리장치의 레지스터 중 도착 레지스터 선택
- ✓ **F 필드**: 처리장치의 ALU 연산 선택
- ✓ **H 필드**: 처리장치의 시프트 연산 선택
- ✓ **MUX 1 필드**: 0과 1로서 내부와 외부 주소를 선택
- ✓ **MUX 2 필드**: 상태비트의 값에 따라 CAR을 구동
- ✓ **ADRS 필드**: ROM의 2진 번지에 해당하는 10진 주소





마이크로명령 형식의 예

❖ 마이크로명령어의 예(26비트의 경우)

✓ 레지스터 전송문 $R1 \leftarrow R1 \wedge R2$, $CAR \leftarrow 43$ 에 대한 마이크로명령어

	A	B	D	F	H	MUX 1	MUX 2	ADRS
	3비트	3비트	3비트	4비트	3비트	1비트	3비트	6비트
기호 마이크로명령어 :	R1	R2	R1	AND	NSH	INT	LAD	43
2진 마이크로명령어 :	001	010	001	1000	000	0	001	101011

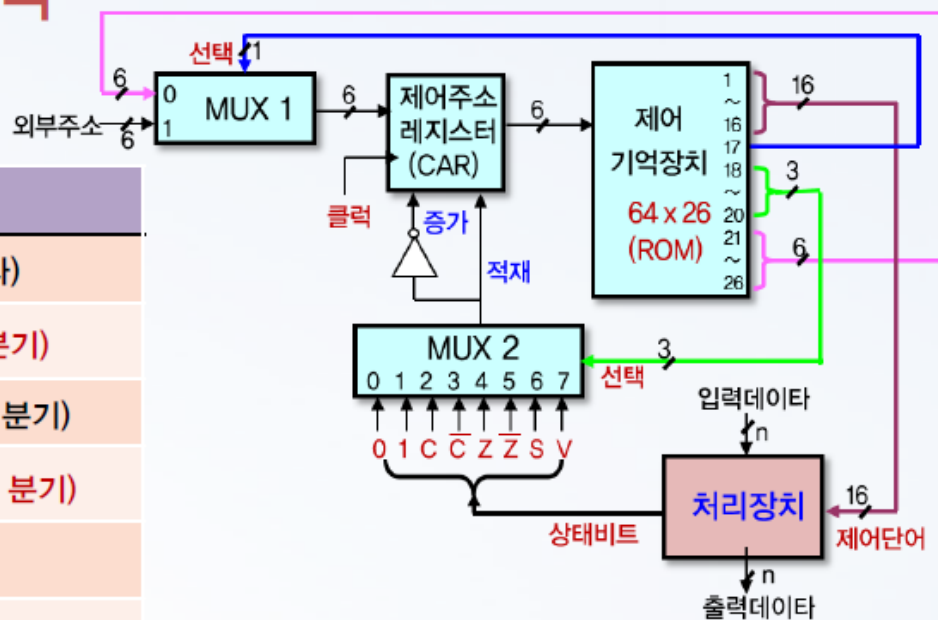


마이크로명령 형식

마이크로명령어 형식의 내역

❖ MUX 2의 선택기능

2진코드	기호	기 능
000	NEXT	CAR을 증가(다음 주소로 간다)
001	LAD	CAR에 주소를 적재(무조건 분기)
010	LC	캐리가 있으면 적재(C=1이면 분기)
011	LNC	캐리가 없으면 적재(C=0이면 분기)
100	LZ	제로면 적재(Z=1이면 분기)
101	LNZ	제로가 아니면 적재(Z=0이면 분기)
110	LS	부호가 바뀌었으면 적재(S=1이면 분기)
111	LV	범람이 발생했으면 적재(V=1이면 분기)





마이크로명령어의 예

❖ 마이크로프로그램이 아님

번호	CAR 주소	제 어 단 어					선 택		다 음 주소필드
		A	B	D	F	H	MUX1	MUX2	
1	36	R1 001	R2 010	R1 001	AND 1000	NSH 000	- 0	NEXT 000	- 000000
2	40	R3 011	- 000	R3 011	DEC 0110	NSH 000	INT 0	LAD 001	43 101011
3	52	- 000	- 000	R4 100	TSF 0000	ZERO 011	INT 0	LS 110	37 100101
4	59	R6 110	R7 111	R2 010	ADD 0010	NSH 000	EXT 1	LAD 001	- 000000



마이크로명령어의 예

예 1

- ▶ 주소 36에 저장되어 있는 마이크로명령
- ▶ $R1 \leftarrow R1 \wedge R2$, $CAR \leftarrow CAR + 1$

번호	CAR 주소	제 어 단 어					선 택		다 음 주소필드
		A	B	D	F	H	MUX1	MUX2	
1	36	R1 001	R2 010	R1 001	AND 1000	NSH 000	- 0	NEXT 000	- 000000



마이크로명령어의 예

예 2

- ▶ 주소 40에 저장되어 있는 마이크로명령
- ▶ $R3 \leftarrow R3-1$, $CAR \leftarrow 43$

번호	CAR 주소	제 어 단 어					선 택		다 음 주소필드
		A	B	D	F	H	MUX1	MUX2	
2	40	R3 011	- 000	R3 011	DEC 0110	NSH 000	INT 0	LAD 001	43 101011



마이크로명령어의 예

예 3

- ▶ 주소 52에 저장되어 있는 마이크로명령
- ▶ $R4 \leftarrow 0$, if($S=1$) then($CAR \leftarrow 37$) else ($CAR \leftarrow CAR+1$)

번호	CAR 주소	제 어 단 어					선 택		다 음 주소필드
		A	B	D	F	H	MUX1	MUX2	
3	52	-	-	R4	TSF	ZERO	INT	LS	37
		000	000	100	0000	011	0	110	100101



마이크로명령어의 예

예 4

- ▶ 주소 59에 저장되어 있는 마이크로명령
- ▶ $R2 \leftarrow R6 + R7$, $CAR \leftarrow$ 외부주소

번호	CAR 주소	제 어 단 어					선 택		다 음 주소필드
		A	B	D	F	H	MUX1	MUX2	
4	59	R6 110	R7 111	R2 010	ADD 0010	NSH 000	EXT 1	LAD 001	- 000000

마이크로프로그램의 작성



마이크로프로그램

- **마이크로프로그램이란?**

제어기억장치에 저장된 일련의 마이크로명령들의 집합

→ **마이크로프로그램 루틴(microprogram routine)**

❖ 이러한 마이크로프로그램을 이용하여 제어장치를 구현

→ **마이크로프로그램에 의한 제어장치**

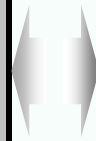


마이크로프로그램의 작성

작성 예

동작 예

- 1) 레지스터 R1의 내용에서 R2의 내용을 빼서 그 결과를 R3에 저장한다
- 2) 또한 R1과 R2의 내용 중에서 작은 수를 R4에 더하고, 만약 R1과 R2의 내용이 같으면 R4를 1 증가시킨다
- 3) 마지막으로 R4의 내용을 처리장치의 출력단자로 내보낸다.



순서적인 표현

- 1) $R3 \leftarrow R1 - R2$ (이때 C와 Z에 새로운 값이 들어간다.)
- 2) a. $R1 < R2$ 이면($C=0$)
 $R4 \leftarrow R4 + R1$
b. $R1 > R2$ 이면($C=1, Z=0$)
 $R4 \leftarrow R4 + R2$
c. $R1 = R2$ 이면($Z=1$)
 $R4 \leftarrow R4 + 1$
- 3) $Output \leftarrow R4$, 외부주소로 분기한다.



마이크로프로그램의 작성

- 레지스터 전송문 (마이크로프로그램 루틴은 20번지에서 시작하여 외부번지로 분기한다고 가정)

순서적인 표현
1) $R3 \leftarrow R1 - R2$ (이때 C와 Z에 새로운 값이 들어간다.)
2) a. $R1 < R2$ 이면($C=0$) $R4 \leftarrow R4 + R1$
b. $R1 > R2$ 이면($C=1, Z=0$) $R4 \leftarrow R4 + R2$
c. $R1 = R2$ 이면($Z=1$) $R4 \leftarrow R4 + 1$
3) $Output \leftarrow R4$, 외부주소로 분기한다.



주소	마이크로 연산과 분기조건
20	$R3 \leftarrow R1 - R2, CAR \leftarrow CAR + 1$
21	if($C=1$) then ($CAR \leftarrow 23$) else ($CAR \leftarrow CAR + 1$)
22	$R4 \leftarrow R4 + R1, CAR \leftarrow 26$
23	if($Z=0$) then ($CAR \leftarrow 25$) else ($CAR \leftarrow CAR + 1$)
24	$R4 \leftarrow R4 + 1, CAR \leftarrow 26$
25	$R4 \leftarrow R4 + R2, CAR \leftarrow CAR + 1$
26	출력 $\leftarrow R4, CAR \leftarrow$ 외부주소



마이크로프로그램의 작성

기호 마이크로프로그램

주소	A	B	D	F	H	MUX1	MUX2	다음주소
20	R1	R2	R3	SUB	NSH	—	NEXT	—
21	—	—	NONE	TSF	NSH	INT	LC	23
22	R4	R1	R4	ADD	NSH	INT	LAD	26
23	—	—	NONE	TSF	NSH	INT	LNZ	25
24	R4	—	R4	INC	NSH	INT	LAD	26
25	R4	R2	R4	ADD	NSH	—	NEXT	—
26	R4	—	NONE	TSF	NSH	EXT	LAD	—



마이크로프로그램의 작성

2진 마이크로프로그램

주소	A	B	D	F	H	MUX1	MUX2	다음주소
010100	001	010	011	0101	000	0	000	000000
010101	000	000	000	0000	000	0	010	010111
010110	100	001	100	0010	000	0	001	011010
010111	000	000	000	0000	000	0	101	011001
011000	100	000	100	0001	000	0	001	011010
011001	100	010	100	0010	000	0	000	000000
011010	100	000	000	0000	000	1	001	000000



마이크로프로그램의 작성

● 마이크로프로그램의 개념

- ▶ 제어장치 설계를 위한 시스템적인 접근과정
- ▶ 마이크로명령 형식을 설정하면 마이크로프로그램의 작성과정은 컴퓨터의 프로그램 작성과정과 유사
- ▶ 이러한 이유로 펌웨어(firmware)라고도 함