

7주차 3차시 기억장치의 확장

【학습목표】

1. 워드 용량의 확장을 예를 들어 설명할 수 있다.
2. 기억장치의 설계를 8비트 마이크로 컴퓨터를 위한 기억장치의 설계를 예시로 함께 설명할 수 있다.

학습내용1 : 워드 길이의 확장

1. 기억장치 확장의 방법

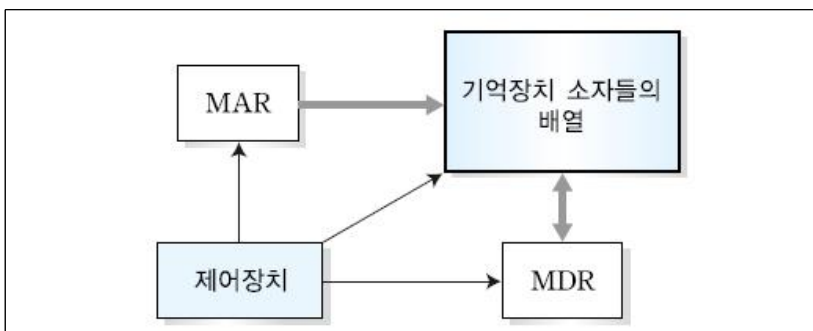
<여러 개의 기억장치 칩을 가지고 더 큰 용량을 기억장치를 설계>

- 워드의 수는 그대로 유지하고 워드의 길이를 확장하는 방법과 워드의 길이는 그대로 유지하면서 워드의 수를 증가시켜서 워드 용량을 확장하는 방법이 있음

① 기억장치의 워드길이(word length)를 확장하는 경우

- 늘어난 길이만큼 데이터 버스의 비트 수를 늘려야 함
- 원하는 워드의 용량 및 워드의 길이를 얻기 위해서 몇 개의 기억장치 칩이 조합할 때, MAR(memory address register), MDR(memory data register), 제어장치 등의 주변장치들이 기억장치를 적절하게 활용함

② 기억장치의 주변장치



2. 워드 길이 확장

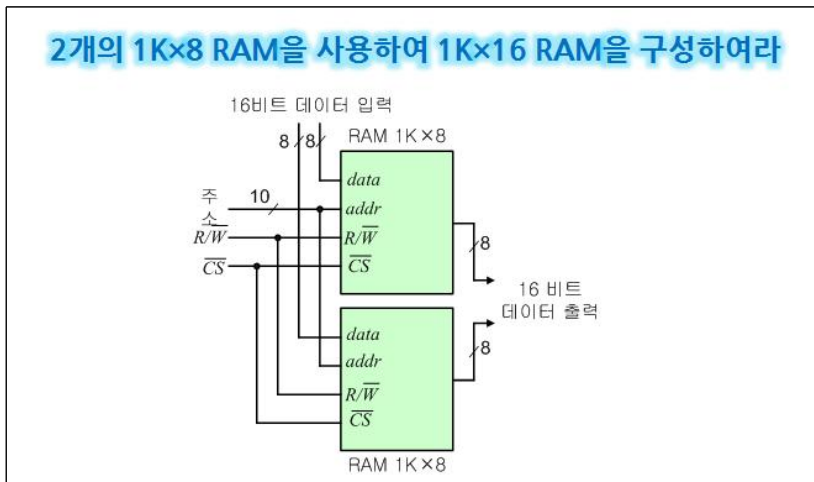
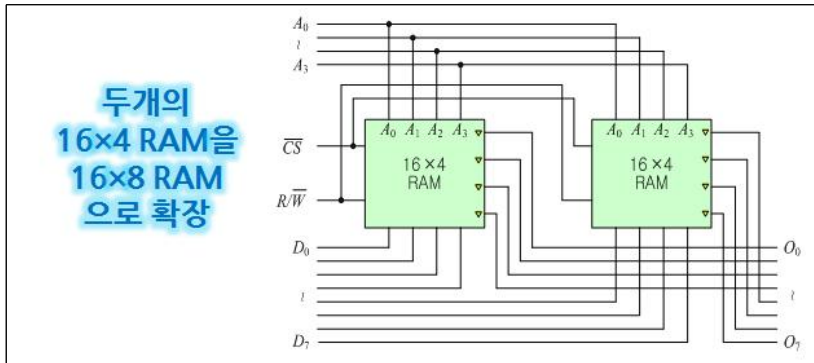
<기억장치 칩들의 주소버스와 제어버스는 공통신호를 사용하고 여러 워드를 순차적으로 연결하여 워드의 용량을 유지하면서 길이를 확장할 수 있음>

① R/\overline{W} : 선택된 RAM 칩의 읽기(read)와 쓰기(write) 동작을 제어

- 선택된 RAM 칩의 읽기(read)와 쓰기(write) 동작을 제어

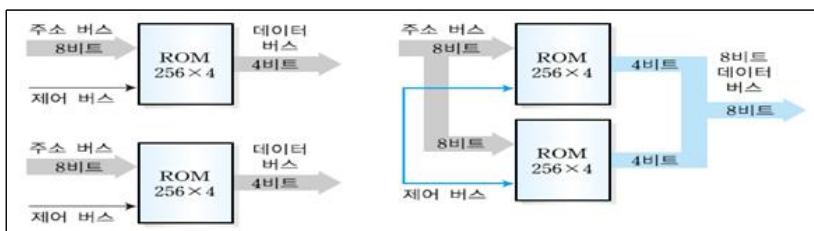
② 출력의 ∇ 표시는 3 상태(tri-state) 출력을 표시

- ③ $\overline{CS} = 1$ 이면 RAM 칩은 선택되지 않고 출력은 Hi-Z 상태가 된다.
 ④ $\overline{CS} = 0$ 이고 $R/\overline{W} = 1$ 이면 주소에 의해 선택된 8비트의 데이터가 출력선을 통하여 출력



<256×4bit ROM 두 개를 이용하여 256×8bit의 기억장치를 만드는 과정>

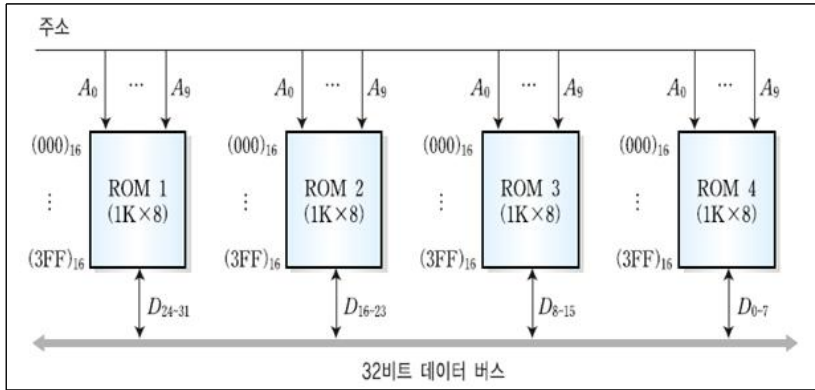
- 두 기억장치의 주소버스와 제어버스는 공통 병렬 신호로 사용이 되고 4비트의 길이를 갖는 각 워드들은 연결되어서 8비트의 워드길이가 됨



3. 1K×8bit RAM 칩들을 이용한 1K×32bit 기억장치 모듈 설계

<4개의 1K×8bit RAM 칩들을 적절하게 설계하면 워드의 길이가 확장되는 (1K×8bit)×4 = 1K×32bit의 기억장치 모듈을 설계할 수 있음>

- 각각의 1K × 8bit RAM은 A₀ ~ A₉까지 10개의 주소선을 입력선으로 갖게 됨 (2¹⁰ = 1024 = 1K)
- 000(16) ~ 3FF(16)까지 주소를 갖게 됨
- 데이터를 저장할 때는 동일한 기억장치의 주소에 대하여 32비트들 중에서 칩 당 8비트씩 분산 저장하게 됨



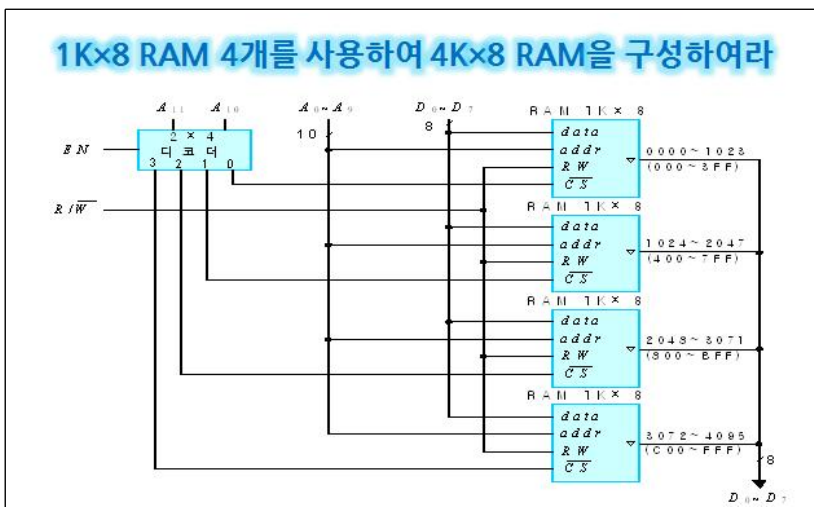
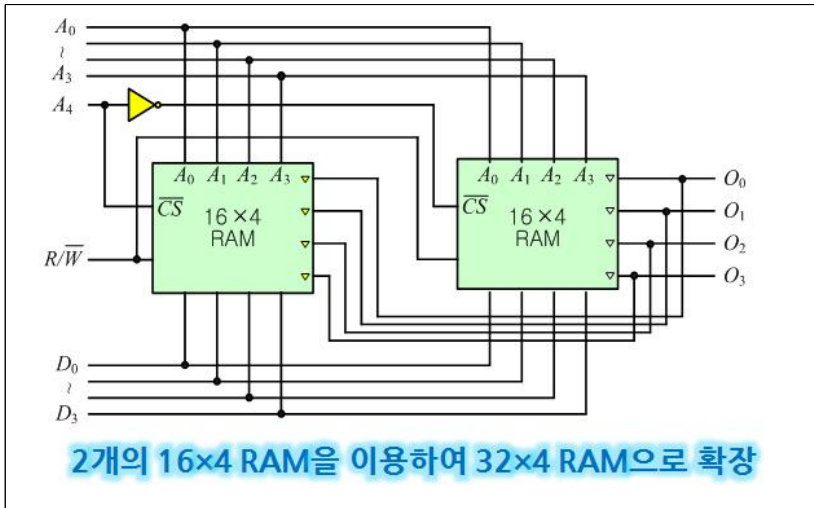
학습내용2 : 워드 용량의 확장

1. 워드 용량의 확장 방법

<기억장치 칩들의 칩 선택 신호는 제어 버스를 통해 공통으로 연결되고, 각 기억장치 칩의 선택을 통해서 주소를 확장하고 워드의 용량을 확장을 할 수 있음>

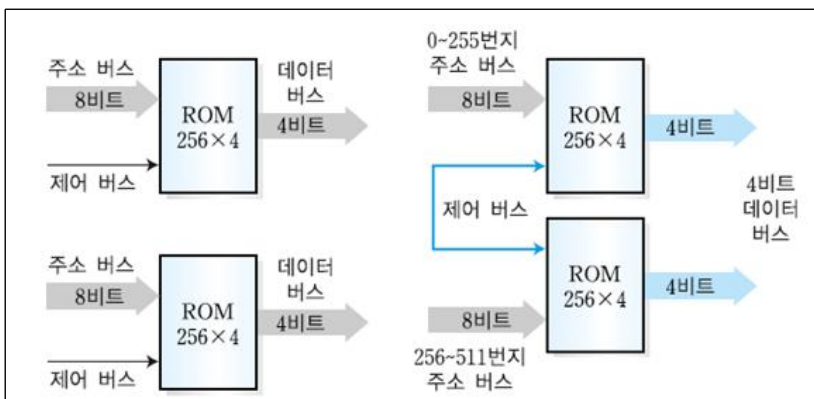
2. 워드 용량의 확장 예

- ① 16×4 RAM 2개를 사용하여 32×4 RAM을 구성하는 경우
- ② 32개의 서로 다른 주소가 존재하므로 주소버스의 길이는 5
- ③ A₄=0이면, A₄A₃A₂A₁A₀ = 00000~01111
- ④ A₄=1이면, A₄A₃A₂A₁A₀ = 10000~11111



<256×4bit ROM 두 개를 이용하여 512×4bit의 기억장치를 만드는 과정>

- 공통 제어버스의 칩 선택 신호가 첫 번째 ROM을 선택하면 주소 0~255번지 내에 저장된 데이터를 접근하게 됨
- 두 번째 ROM을 선택하면 주소 256~511번지 내에 저장된 데이터를 접근할 수 있게 됨

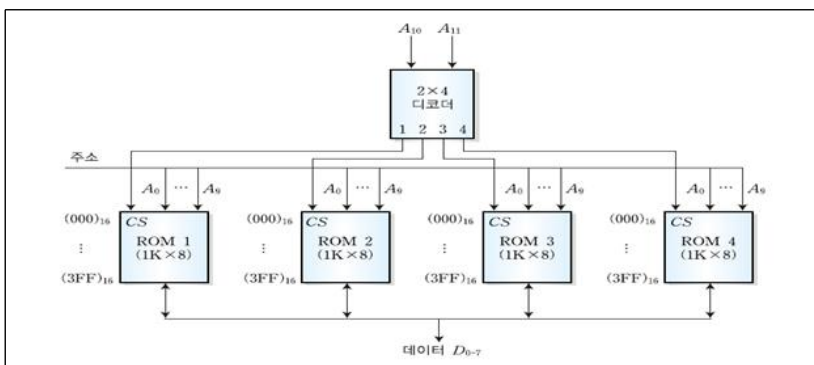


3. 1K×8bit RAM 칩들을 이용한 4K×8bit 기억장치 모듈 설계

- <4개의 1K×8bit RAM 칩들을 적절하게 설계하면 용량이 (1K×8bit)×4 = 4K×8bit인 기억장치 모듈을 만들 수 있음>
- 각각의 RAM은 A0 ~ A9까지 10개의 주소선을 입력선으로 갖게 됨(210 = 1024 = 1K)
 - RAM들의 주소값이 연결되기 때문에 RAM1은 000(16) ~ 3FF(16), RAM2는 400(16) ~ 7FF(16), RAM3은 800(16) ~ BFF(16), RAM4은 C00(16) ~ FFF(16), 까지 각각 다른 주소를 할당 받게 됨
 - 주소로 접속을 할 때는 상위 10개 비트 A0 ~ A9까지 각 RAM에 공통으로 할당되고 하위 두 비트 A10 ~ A11 은 2×4 decoder의 출력으로 4개의 칩 선택신호를 발생시켜 각 RAM을 선택하도록 함

4. 설계된 4K×8bit 기억장치 모듈

- 1) 2×4 decoder의 입력 비트는 4개의 RAM 중에서 하나를 선택하며, 선택된 RAM의 주소범위를 알맞게 설정



- 2) 설계된 4K×8bit 기억장치 모듈에서 각 RAM의 주소 영역

- 2×4 decoder의 입력 비트 값이 주소 비트의 상위 두 비트가 됨

칩 번호	주소 비트												주소 영역
	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
RAM 1	0	0	0	0	0	0	0	0	0	0	0	0	(000)16번지부터
	0	0	1	1	1	1	1	1	1	1	1	1	(3FF)16번지까지
RAM 2	0	1	0	0	0	0	0	0	0	0	0	0	(400)16번지부터
	0	1	1	1	1	1	1	1	1	1	1	1	(7FF)16번지까지
RAM 3	1	0	0	0	0	0	0	0	0	0	0	0	(800)16번지부터
	1	0	1	1	1	1	1	1	1	1	1	1	(BFF)16번지까지
RAM 4	1	1	0	0	0	0	0	0	0	0	0	0	(C00)16번지부터
	1	1	1	1	1	1	1	1	1	1	1	1	(FFF)16번지까지

학습내용3 : 기억장치 설계의 예

1. 8비트 마이크로 컴퓨터를 위한 기억장치의 설계 예

1) 8비트 마이크로 컴퓨터의 스펙

① 용량

- RAM : 1K byte / ROM : 512byte

② 주소 영역

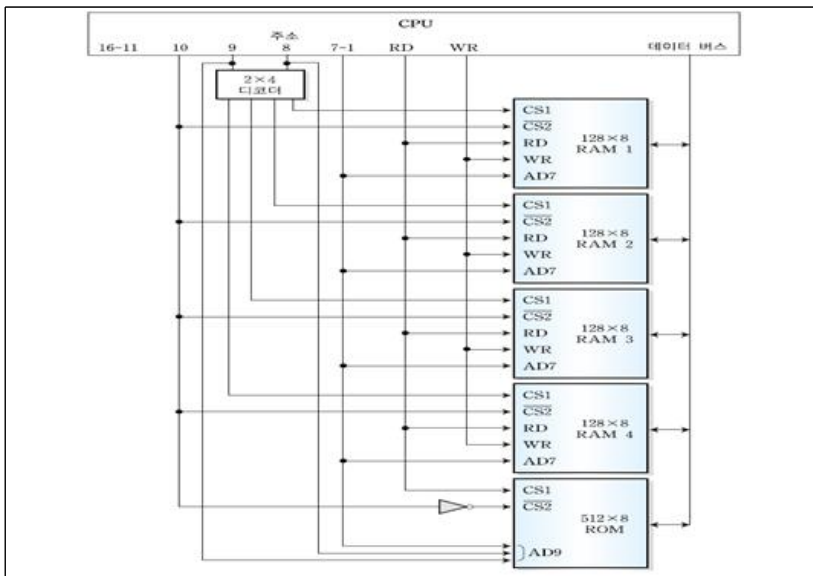
- RAM : $0_{(16)}$ 번지부터 / ROM : $800_{(16)}$ 번지부터

③ 사용 가능한 칩

- RAM : 256×8 비트 / ROM : 512×8 비트

2) 8비트 마이크로 컴퓨터를 위한 기억장치 주소표

칩 번호	주소 비트												주소 영역
	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
RAM 1	0	0	0	0	x	x	x	x	x	x	x	x	$(000)_{16} \sim (0FF)_{16}$
RAM 2	0	0	0	1	x	x	x	x	x	x	x	x	$(100)_{16} \sim (1FF)_{16}$
RAM 3	0	0	1	0	x	x	x	x	x	x	x	x	$(200)_{16} \sim (2FF)_{16}$
RAM 4	0	0	1	1	x	x	x	x	x	x	x	x	$(300)_{16} \sim (3FF)_{16}$
ROM	1	0	x	x	x	x	x	x	x	x	x	x	$(800)_{16} \sim (BFF)_{16}$



【학습정리】

1. 기억장치 칩들의 주고버스와 제어버그에 공통 신호를 사용하고 여러 워드를 순차적으로 연결하면 워드의 용량을 유지하면서 워드의 길이를 확장할 수 있다.
2. 기억 장치의 칩선택 신호는 제어버스를 통해 공통으로 연결되고, 각 기억 장치 칩의 선택을 통해서 주소를 확장하고 워드의 용량을 확장할 수 있다.