

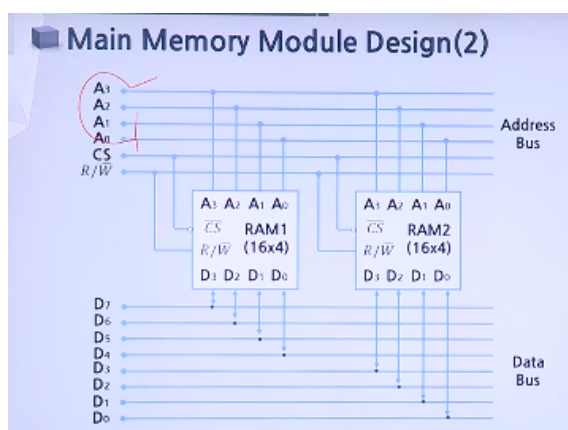
# 24강\_Main Memory Module Design

🕒 Created	@Aug 13, 2020 12:04 PM
🏷 Tags	RE

## Main Memory Module Design(1)

기억장치 칩의 데이터 비트 수가 단어 길이보다 짧은 경우 여러 개의 칩들을 병렬로 접속한 기억장치 모듈로 구성

- 단어의 길이 =  $N$ , 기억장치 칩의 데이터 비트 수 =  $B$  라면,  $N/B$  개의 칩들을 병렬접속
- [예]  $N = 8$ 일 때,  $16 \times 4$ 비트 RAM 칩들을 이용한 기억장치 모듈의 설계
  - 방법 : 2개의 RAM 칩들을 병렬 접속
  - 모듈의 용량 :  $(16 \times 4) \times 2 = 16 \times 8$ 비트 = 16단어
  - 주소 비트(4개:  $A_3 \sim A_0$ ) : 두 칩들에 공통으로 접속
  - 칩 선택 신호(CS)를 두 칩들에 공통으로 접속
  - 주소 영역 :  $0000 \sim 1111_2$



Active Low : Low 일 때 활성화가 된다

### Main Memory Module Design(3)

- 기억장치 칩의 데이터 비트 수가 단어 길이보다 짧은 경우 여러 개의 칩들을 병렬로 접속한 기억장치 모듈로 구성
  - [예2] 1K×8Bits RAM Chip들을 이용한 1K×32Bits Main Memory Module Design
    - 4개의 1K×8Bits RAM Chip들을 병렬로 접속
    - Module의 Capacity :  $(1K \times 8\text{Bits}) \times 4\text{개} = 1K \times 32\text{Bits} = 1K \text{ Words}$
    - Address Bits(10개:  $A_9 \sim A_0$ ) : 모든 Chip들에 Common으로 접속
    - Address 영역 : 000H ~ 3FFH(단, H 는 16 진수 표시)
    - Data 저장 : 동일한 Main Memory Address에 대하여 Chip 당 (32Bits들 중의) 8Bits씩 분산 저장

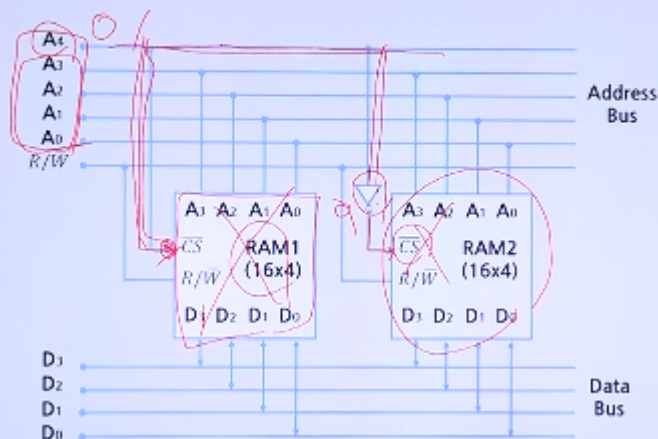
### Main Memory Module Design(4)



### Main Memory Module Design(5)

- 필요한 기억장치 용량이 각 기억장치 칩의 용량보다 큰 경우 여러 개의 칩들을 직렬로 접속하여 기억장치 모듈을 구성
  - [예] 두 개의 16×4비트 RAM 칩들을 이용한 32×4비트 기억장치 모듈의 설계 (기억장치 용량 확장)
    - 방법 : 2개의 RAM 칩들을 직렬 접속
    - 모듈의 용량 :  $2\text{개} \times (16 \times 4) = 32 \times 4\text{비트}$
    - 주소 비트 수 : 5개 ( $A_4 \sim A_0$ )
    - $A_4$  : 칩 선택 신호(CS)로 사용
    - $A_3 \sim A_0$  : 두 칩들에 공통으로 접속
    - 주소 영역 : RAM1: 00000 ~ 01111<sub>2</sub>, RAM2: 10000 ~ 11111<sub>2</sub>

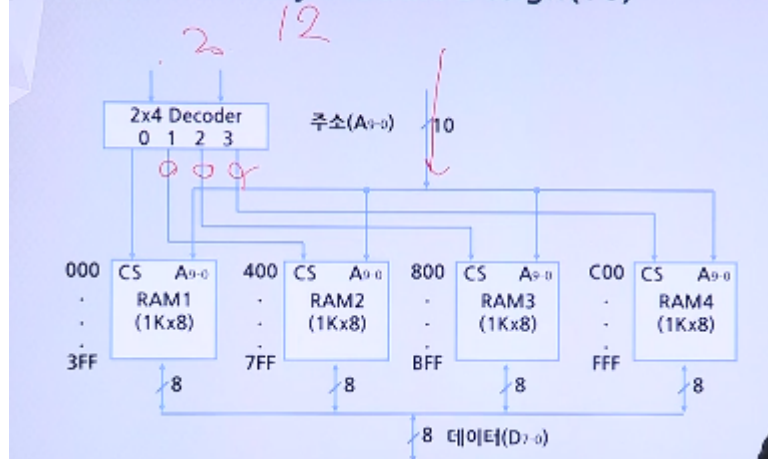
### Main Memory Module Design(6)



## Main Memory Module Design(7)

- 필요한 기억장치 용량이 각 기억장치 칩의 용량보다 큰 경우 여러 개의 칩들을 직렬로 접속하여 기억장치 모듈을 구성
  - 1K×8Bits RAM Chip들을 이용한 4K×8Bits Main Memory Module Design
    - 4개의 RAM Chip들을 직렬 접속
    - Module의 Capacity :  $(1K \times 8\text{Bits}) \times 4\text{개} = 4K \times 8\text{Bits} = 4K \text{ Byte}$
    - Address Bits(12개:  $A_{11} \sim A_0$ ) 접속 방법
      - 상위 2Bits: Address Decoder를 이용하여 4개의 Chip Select Signal 발생
      - 하위 10Bits: 모든 Chip들에 Common으로 접속
    - Address 영역 : 000H ~ FFFH
    - Data 저장 : 각 Main Memory Address당 8 Bits씩 저장
    - 각 RAM에 지정되는 Address 영역

## Main Memory Module Design(10)



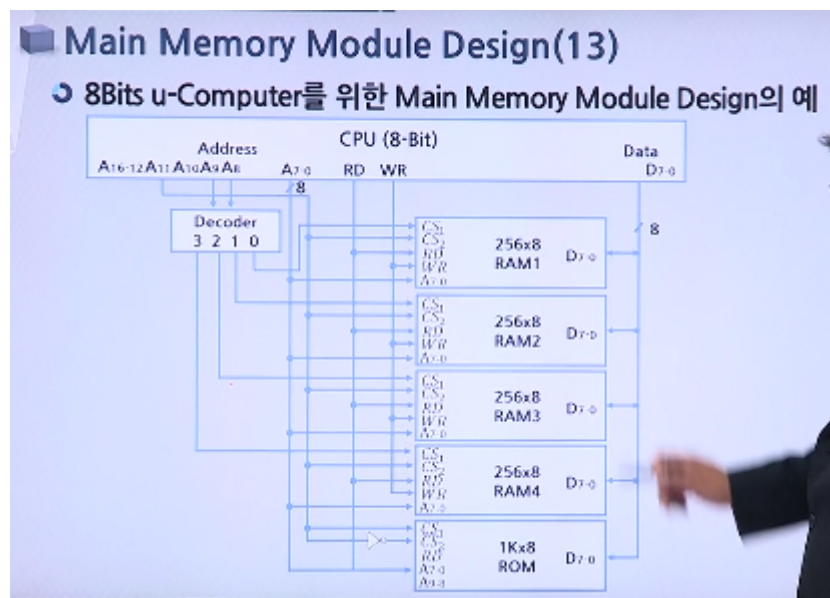
## Main Memory Module Design(11)

- 8Bits u-Computer를 위한 Main Memory Module Design의 예
  - 기억장치 모듈의 설계 순서
    - Computer System에 필요한 기억장치 용량 결정
    - 사용할 칩들을 결정하고, 주소 표를 작성
    - 세부 회로 설계
  - Capacity : 1K Bytes RAM, 1K Bytes ROM
  - 사용 가능한 Chip들 : 256×8 Bits RAM, 1K×8 Bits ROM

### Main Memory Module Design(12)

8Bits u-Computer를 위한 Main Memory Module Design의 예

칩	주소 (hexa)	Address Bit들											
		A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
RAM1	000H~0FFH	0	0	0	0	x	x	x	x	x	x	x	x
RAM2	100H~1FFH	0	0	0	1	x	x	x	x	x	x	x	x
RAM3	200H~2FFH	0	0	1	0	x	x	x	x	x	x	x	x
RAM4	300H~3FFH	0	0	1	1	x	x	x	x	x	x	x	x
ROM	800H~BFFH	1	0	x	x	x	x	x	x	x	x	x	x



#### ▼ 스크립트

1. 네. 우리가 이제 지금 현재 Memory를 공부하고 있습니다. 가장 처음에 계층적 Memory 구조에 대해서 공부를 했고요. 그 계층적 Memory의 구조에 대표적인 걸로 Main Memory에 대해서 공부를 했습니다. 주로 이제 거기 들어가는 구체적인 칩은 RAM과 ROM이 있었죠.
2. 자 이제 마지막으로 이번 주차에 해야 될 게 그런 Main Memory RAM과 ROM을 어떻게 디자인을 하게 되냐 디자인 한다는 게 하드웨어를 속을 디자인하는 게 아니고요. 그런 제품들이 나와 있어요. 어떻게 CPU와 연결할 수 있게끔 Data Bus, Address Bus, Control Bus, 이런 것을 어떤 방식으로 연결해서 CPU와

Data를 오갈 수 있게 하겠느냐를 지금부터 공부할 겁니다. 그러니까 일종의 하나의 설계라고 볼 수 있겠죠.

3. 이제 설계하는 방법은 여러 가지가 있습니다. 첫 번째, 예부터 한 4가지 정도, 5가지 정도 공부를 할텐데요. 차근차근 공부를 해보도록 하죠. 기억장치 칩의 Data bit 수가 단어 길이보다 짧은 경우, 이제 단어 길이라는 게 뭐냐면 이런 애입니다. 예를 들어서 CPU가 있습니다. CPU.
4. CPU에서 필요한 단어 Data 버스는 이제 뭐 예를 들어서 16bit 짜리다, 그런데 Memory 내가 산 Memory 갖고 있는 Memory는, 예를 들어서 이런 거죠. 2의 6승 x 8이다 그럼 이런 Memory는요 한 순간에 8bit 밖에 전송을 해줄 수밖에 없습니다. 그런데 CPU가 받아야 되는 건 한순간 16bit를 받아야 되거든요. 이런 경우 어떻게 우리가 Main Memory 모듈을 설계를 할 건가의 애입니다.
5. 그게 바로 여기 보시면 기억장치 칩의 Data bit 수. 자, 이게 뭐냐, 8입니다. 그다음에 단어 길이 이거 CPU가 필요한 16 이거 얘기하는 거죠. 그러니까 16보다 8이 작다는 얘기죠. 작을 경우에 어떻게 설계 해야 되느냐, 여러 개의 칩들을 병렬로 접속한 기억장치 모듈로 구성을 하겠다. 무슨 얘기냐, CPU가 1개 있으면 Main Memory 첫 번째 1, 그 다음에 Main Memory 2 이것 두 개를 CPU에 같이 연결하겠다는 거죠. 이렇게 해야 되겠죠. 어쩔 수 없이. 이렇게 하는 게 기본적인 설계 방법론입니다.
6. 일반적인 경우를 고민하기 위해서 단어의 길이가 N개다 단어의 길이 여기서는 16이라고 아까 얘기했던 것 N개고 기억장치 칩의 Data bit 수가 B다. 8이라는 거죠. 지금 예에서는 두 개를 병렬한다고 했거든요. 그러니까 16이고 이게 8이니깐 N을 B로 나눈 이 개수만큼의 칩들이 당연히 필요하겠고요. 그걸 병렬로 접속을 하면 한 번에 16bit를 Memory로부터 얻어낼 수 있는 상황을 만들어 낼 수가 있겠죠.
7. 예를 한번 보실게요. N이 8, 그리고 16 x 4 bit RAM칩이다. 이것 이용해서 기억장치 모듈을 한번 설계를 해봐라, 물론 기억장치 칩은 RAM 칩은 두 개 이상이 사용될 수도 있습니다. 자 N이 8입니다. 근데 16 x 4 이거 2의 4승 x 4입니다.
8. 자 그럼 이게 어떤 의미냐면요. Address 버스가 4bit라는 의미고요. 그리고 그때 출력은 4bit를 내준단 얘기에요. 근데 CPU는 한번에 8bit를 받아야 되거든요. 그럼 어떻게 하겠냐는 거예요. 이거는 2의 4승 x 4짜리 RAM칩을 두 개를 병렬연결하겠다는 얘기에요. 방법은 2개의 RAM칩들을 병렬접속을 하겠다. 그렇게 되면 모듈의 용량이 어떻게 되냐면 16 x 4가 아니라 2개를 사용하기 때문에 16 x 8을 만들어 낼 수 있다는 거죠. 16 x 8은 2의 4승 x 8이죠. 이런 칩을 만들어 낼 수 있을 거란 얘기죠. 이거 두 개를 이용하면 이거 하나를 만들어 낼 수 있을 거다.

9. 자 주소 bit는 어떻게 몇 개를 써야 되느냐, 4승이니까 4개를 쓰면 되고요. 그 다음에 이 4개는 두 칩, RAM 칩 16 x 4bit RAM 칩 두 개의 공통으로 접속이 되어야 되고 칩 선택 신호를 두 칩에 역시 공통으로 접속을 해야 되고요, 주소 영역은 지금 주소 영역이라는 게 4bit의 주소로 만들어낼 수 있는 주소의 범위를 얘기해요. 0000부터 여기가 제 2가 생략됐다고 봐야 되겠네요. 이거 써야 되요. 2진수 0000부터 2진수 1111까지 총 16가지 주소 영역을 만들어 낼 수가 있겠죠.
10. 자 이걸 그림으로 그대로 표현을 해야 되겠습니다. 지금 보시면 16 x 4 bit RAM 칩이 2개가 병렬로 딱 이렇게 있습니다. 자 그런 다음에 Address bit는 여기서부터 여기까지입니다. 이게 Common으로, Common이라는 게 두 개의 공통으로 연결되었다는 얘기죠. 여기도 연결되고 여기도 연결되고.
11. 그럼 Address가 하나 주어진다면요. 여기서 bit가 4bit가 툭 튀어 나오고 여기서부터도 4bit가 툭 튀어나옵니다. 그러니까 Data 버스에 Data를 실을 때 상위 4개 bit에는 이 RAM1에서 튀어 나온 Data를 실게 되고요. 하위 4개bit에는 이 RAM2에서 나온 Data 4개를 실게 되죠. 그럼 total 한번 Address를 주게 되면 Address를 4bit 짜리를 한번 주게 되면 총 8개의 Data bit를 얻을 수 있는 그런 효과를 가져올 수가 있겠죠. 그런 식으로 구성하자는 게 이 첫 번째 예제가 되겠습니다.
12. Chip Select는 어떻게 연결 되어야 되냐면요. 두 개 다 동시에 연결이 되어 있어야 되고요. 그 다음에 Read, Write 이거는 읽을거냐, 쓸거냐를 결정하는 제어 신호가 되겠고요. Chip Select는 역시 제어신호, Chip Select도 이제 컨트롤 버스 안에 들어가는거죠. 여기서 재밌는 게 여기 보시면 이게 이렇게 바가 되있는 게 있어요. 모자 쓰듯이 써있는 거 요거, 이걸 뭐냐면 Active Low라고 합니다. Active Low.
13. Active Low.이게 무슨 의미냐면요. Low 일 때 활성화가 된다는 의미예요. 보통 우리가 이제 Signal 그림을 그리면 이렇게 되거든요. 이게 Low고요. 0인 상태 0Volt인 상태, 이거, 올라오는 상태가 5Volt 이고요. 이게 High입니다. 이 y축은요 전압이에요. 전압. 그림을 이렇게 그리면 y축은 전압을 얘기하는 겁니다. 그래서 전압이 0Volt면 Low, 5Volt면 High죠.
14. 보통 우리가 어떤 칩이 Enable된다는 것은 입력으로 High가 들어와야 되는 Low이면 Disable로 되어 있다가 High가 되면 이제 Enable이 되는 거죠. 그래서 칩이 동작을 하게 되는건데 만약에 이게 바가 붙어있으면 거꾸로 된다는 의미입니다.
15. 무슨 얘기냐면 High일 때 동작을 안하고요. Low로 내려와만 동작을 한다는 얘깁니다. 저렇게 구성하는 칩들도 굉장히 많습니다. 그래서 이렇게 동작시키는 것을 Active High 이렇게 동작시키는 걸 Active Low라고 부릅니다. 여기 바가 붙어있으면 Active Low에 동작을 하는 칩이 되겠습니다.



16. 자 그다음, 2번째 예를 한번 볼까요. 이것도 역시 똑같습니다. 다만 추가적인 예제를 하나 더 보여드리는 거예요. 별로 어려운건 없습니다. 기억장치 칩의 Data bit 수가 단어 길이보다 짧은 경우 앞 경우와 똑같죠.
17. 여러 개의 칩들을 병렬로 접속한 기억장치 모듈로 구성한다. 똑같습니다. 똑같은 건데 다만 이걸 하나 더 드리는 이유는 여기 보시면 아까하고 달리 1K x 8bit고요 그걸 이용해서 1K x 32bit Main Memory 모듈을 만들으라는 얘기에요. 저건 어떻게 해야 될까요? 4개를 병렬연결하면 되죠. 4개. 여기는 8이고요. 여기는 32거든요. 그러면 4개를 병렬연결 하면 되요. 그죠.
18. 여기 있는 것들은 여러분들이 한번 읽어보시면 될 것 같고요. 제가 별도로 여기서 또 얘기하는 게 큰 의미는 없을 것 같아요. 어쨌든 연결해서 그림을 그려보겠습니다. 그러면 이렇게 그림을 그려질 수가 있죠. RAM1 첫 번째 RAM, 그 다음에 2번째 RAM, 3번째, 4번째가 있고요. 각각의 RAM은 1K x 8bit짜리 RAM이 됩니다.
19. 그럼 여기 주소공간을 한번 볼게요. 주소가 10bit가 있는데요. 주소가 10bit면 어떻게 되냐면 이렇게 됩니다. All Zero부터 이렇게 돼서 총 이렇게 되죠. 그러면 총 다 써보겠습니다. 헛갈릴 수 있으니까, 이렇게 되면 하위bit 이건 16진수로 바꾸면 이게 0입니다. 이걸 16진수로 바꿀게요. 이거 0입니다. 이거 16진수로 바꿀게요. 이거 0이죠. 그러니까 000부터 똑같은 방식으로 11 다 1이라고 쳐볼게요. 그럼 1111은 16진수로 뭐죠? F죠. F. 그다음에 이거, 이것도 역시 16진수로 f가 되죠. 이거는 16진수로 3이 되죠. 3FF까지.
20. 여기 주소공간은 000부터 3FF 있을 수가 있습니다. 그 주소 입력된 것에 따라서 이 여러 개 중에 2의 10승개니까 총 1024개 공간 중에 어느 한 공간을 선택을 딱 하게 되겠죠. 선택을 하는 순간 8개의 bit가 여기로 쭉 빠져나옵니다. 여기도 8개가 쭉 빠져 나오죠. 동일한 주소를 선택하니까, 쭉 8bit 짜리가 4개가 나오게 되요. 근데 실제 거기에 저장된 Data들은 다 다르죠. 칩들이 다 다르니까, 그래서 total 32bit를 얻어낼 수가 있게 됩니다.
21. 그래서 출력으로 32bit를 내줄 수 있게 됩니다. 그럼 이거는 식으로 표현하면 이렇게 되겠습니다. 식으로 간단히 표현해 볼게요. 1K x 8bit를 갖고 이것을 4개를 이용해서 곱하기가 여기 들어가 있죠. 이게 1K x 32 bit 짜리 RAM을 만들어 낸 거죠. 이렇게 이제 구성을 하게 되죠.
22. 자. 다음은 필요한 기억장치 용량이 각 기억장치 칩의 용량보다 큰 경우. 자 이건 무슨 얘기냐면.
23. 지금 여태까지는요, 예를 들어 이런 겁니다. 2의 6승 x 8 이렇게 칩시다. 지금 여태까지 어쨌냐면요. 이 8이 실제 필요한 게 8이 아니라 16이 되버리는 경우입니다. 그러니까 칩을 두 개를 병렬로 연결해서 두 개를 사용해서 2의 6승 x 16을 만

들거나 2의 6승 x 32를 만들거나 이런 식으로 Data 버스의 폭을 넓히는 그런 작업을 했었죠.

24. 이번에는 그게 아니라고요. 필요한 기억장치 용량이 각 기억장치 칩의 용량보다 큰 경우입니다. 이거는 이게 늘어나야 됩니다. 이게, Address bit의 개수가 늘어나는 케이스. 이 경우는 우리가 어떻게 Main Memory를 설계할건가 그걸 이제 지금부터 보겠습니다. 이런 경우에는 여러 개의 칩들을 직렬로 접속하여 기억장치 모듈을 구성하면 우리가 쉽게 구성을 할 수가 있습니다.
25. 한번 볼까요. 2개의 16 x 4bit RAM 칩들. 이걸 해보면 이렇게 되죠. 2의 4승 x 4가 되고요. 이걸 이용해 32 x 4bit 이걸 2의 5승 x 4가 됩니다. 기억 장치 모듈을 어떻게 설계를 할 거냐, 그러니까 원래 저장할 수 있는 공간의 개수가 16개에서 32개로 늘어나야 되기 때문에 늘어나야 된다는 얘기는 결국은 칩들을 두 개를 연속해서 사용해야 된다는 얘기에요. 그래서 상위 16개는 1번째 칩 하위 16개 공간은 2번째 칩을 사용해서 총 32개의 공간을 만들어 내야 되는거죠.
26. 2개의 RAM칩들을 직렬로 접속을 하겠다는 거고요. 그렇다면 모듈의 용량은 어떻게 되냐면 16 x 원래 4 RAM칩이었는데 2개를 사용하니까 32 x 4가 되죠. 이거 앞으로 곱했습니다. 뒤로 곱한 게 아니라. 우리가 앞선 예에서는 뒤에다 곱했는데요. 지금 이 예에서는 앞에다 곱했죠. Address Line이니까. 그래서 32bit x 4가 됩니다.
27. 총 이게 2의 5승 x 4가 되는 거거든요. 그렇게 되면 어떻게 되냐면 Address Line이 원래 칩에 들어가는 건 4bit밖에 안되죠. 2의 4승이니까 4bit밖에 안되지만 이게 이렇게 늘어났기 때문에 5개가 필요하겠죠. 그래서 주소 bit 수는 5개. 그리고 가장 그중에 MSB 상위 bit는 칩 선택신호로 사용을 하겠다.
28. 그러니까 두 칩 중 이 얘기죠. 위에 공간이 쪽 해서 16개 공간이 있고요. 아래 칩이 또 공간이 쪽 이렇게 있는데 16개가 있는데 처음에 Chip Select신호를 이용해서 이거냐 이거냐고 선택을 하겠다는 거죠. 그럼 이게 사실은 16개 16개 공간이지만 합치면 32개 공간이거든요. 그래서 가장 상위bit가 0이 되버리면 애를 선택하게 될꺼고, 가장 상위bit, MSB가 1이면 애를 선택하게 될꺼예요.
29. 그럼 역시 마찬가지로 주소가 1bit 확장 된 효과가 있습니다. 그거를 Chip Select신호로 입력을 시키겠다는 거죠. 주소 영역은 총 5bit가 필요하기 때문에 이 위에 거, 위에 거는 가장 상위가 00000부터 01111까지죠. 이게 Chip Select신호죠. 이게.
30. 그 다음에 두 번째 RAM칩은 역시 0000부터 1111까진데 Chip Select가 1이죠.
  1. 그죠. 여기가 이렇게 되는 거죠. 따지면 가장 상위가 00000 이거고요. 01111이렇게 되는 거고요. 여기가 10000, 여기가 11111 이렇게 되는 거죠. total 32개의



주소공간을 만들어 낼 수가 있게 되는 겁니다. 그거를 그림을 그려보겠습니다. 이와 같이 그릴 수 있네요.

31. 보시면 Address로 들어가는 총 5bit가 있습니다. 그 중에 MSB A4는 뭘로 갖느냐. 일로 이렇게 옵니다. Chip Select로 오게 되고요. 그다음에 쪽 가서요. 일로 또 오게 됩니다. Chip Select. 근데 여기 Active Low죠. 여기 반전이 일어나죠. 그러면 여기가 만약에 0이라고 하면 일로 0이 들어와서 동그라미가 있으니까 1이 됩니다. 동그라미가 있어서 1이 됩니다. Active Low이기 때문에 이 칩이 선택이 되고요. 여기가 0이 들어오게 된다고 했을 때 여기 반전이 일어나죠. 1이 들어갑니다. 이 칩 선택 안되요.
32. 그러니까 여기가 0이면 여기 안에 있는 16개의 Data 중에 하나를 꺼내는 거죠. 만약에 여기가 1이 됐다고 치자고요. 1이 됐다고 치면 그때는 이게 Active Low이기 때문에 이거 선택 안됩니다. 이게 선택됩니다. 그럼 여기에 있는 16개 중에 하나를 꺼내게 되겠죠.
33. 그 16개 중에 어느 걸 꺼내느냐 그거는 이 밑에 있는 4개의 주소에 의해서 결정이 되겠죠. 그러면 결론적으로 이거는 이 둘 중에 하나, 둘 중에 하나가 선택이 돼서 그중에 16개 중에 하나가 이 Data 버스로 출력이 되게 되는 그런 효과를 갖게 되죠.
34. 이렇게 되면 이거는 2의 5승이죠. 그 다음 곱하기 4가 됩니다. 그러니까 입력주소가 5개, 출력 Data 버스가 4bit 그래서 2의 5승 x 4 그죠. 이거는 2의 4승 x 4의 RAM칩 이거죠. 16 x 4 이거 두 개를 이용해서 2의 5승 x 4를 만들어 낼 수가 있는 거죠. 이제 이런 방식으로 두 번째 예가 되겠습니다.
35. 자 그 다음 예제를 하나 더 볼게요. 지금 똑같은 방식으로 1K x 8 RAM칩이 있는데요. 저거를 이용해서 4K x 8 Main Memory를 만들어야 되는거예요. 저것도 쉽죠. 역시 마찬가지로. 이걸 앞쪽의 4K가 됐으니까 원래는 이걸 계산하면 2의 10승 x 8인데요. 이거를 2의 12승 x 8로 만들어주겠다는 얘기에요. 그죠.
36. 이럴 경우에는 RAM칩이 몇 개가 필요할까요. 4개가 필요합니다. 4개. 4개가 필요하죠 당연히. 이게 더 많이 공간이 4배가 되어 되니까. 그래서 이걸 4개의 RAM 칩들을 직렬로 접속을 하면 됩니다. 그래서 모듈의 Capacity는 1K x 8 x 4. 그랬는데 이 4를 사실은 이 앞에 곱하는게 더 효율적인 거보다는 논리에 맞겠죠. 그래서 4 x 1K x 8 이렇게 되겠죠. 그러면 4K x 8이 나옵니다.
37. 이런 방식으로 설계를 하겠다라는 얘기고요. 결국은 합쳐지는 RAM 칩들, 설계되는 모듈은 주소가 12개, 주소bit가 12개 여야됩니다. 여기 보시는 것처럼 12개. 상위 2bit 같은 경우, 상위 2bit는 4개 중에 1개를 선택해야 되죠. 그래서 00이나 01이나 10이나 11이나 이 4가지 입력에 따라서 4개의 칩중에 하나를 선택하는 용도로 쓰이게 되고 그다음에 하위의 10bit는 모든 칩들의 Common으로 연결 되

있어야 되죠. 공통적으로 다 연결이 되었어서 선택된 칩의 주소를 알려주게 되는 거죠.

38. 그 다음에 Address 영역 같은 경우에는 역시 마찬가지로 12bit이기 때문에 FFF죠. 왜냐면 F하나가 4bit 짜리거든요. 그래서 1111 1111 1111 이렇게 되는 거죠. H가 뒤에 붙으면 역시 마찬가지로 이것도 Hexadecimal이라는 뜻으로 16진수라는 뜻입니다. 뒤에 H가 붙는 것들 이런 것들은 16진수를 의미합니다.
39. 자 Data 저장은 각 Main Memory Address에 8bit씩 쪽 저장을 하면 된다. 자 이거를 말로 이렇게 보는 것 보다요. 그림으로 보는게 편합니다. 그림을 보실게요.
40. 지금 제가 앞서 Address Space는 넘어 갔는데요. 여러분들이 Address Space는 확인을 해보세요. 그림을 다 보신 다음에, 여기 지금 주소가요, 총 12bit가 들어오게 되는데 10bit는 모든 RAM칩의 Common으로 다 연결이 되어 있고요. 상위 2개bit는 Decoder를 통과하게 됩니다. 그죠. Decoder라는게 여러 번 얘기했지만 00이 들어오면 여기만 1, 01이 들어오면 여기만 1 나머지 다 Zero. 10이면 여기만 1, 11이면 여기만 1 이렇게 되는거죠. 그게 Decoder라고 그랬죠.
41. 만약에 00이 들어왔다. 그럼 이것만 선택이 됩니다. 이것만 1이고 나머지 다 0입니다. 이게 Chip Select로 들어가게 되죠. 그럼 이 Chip Select는 이렇게 이렇게 들어가는 3개는 전부 다 0이 되기 때문에 이거 선택 안됩니다. 다. 선택안되요. 이것만 선택이 되요.
42. 그런 다음에 Common으로 들어오는 10개의 주소를 통해서 10bit니까요. 여기 총 1024개의 Memory Space가 있게 되는데 그중에 하나를 선택해서 출력으로 내게 되죠. 8bit 짜리로 그런 다음에 Data를 쪽 뽑아냅니다. 그럼 8bit 짜리 Data를 얻을 수가 있게 되는 거죠.
43. 자 이런식으로 하다 보면 이 2bit의 입력이 뭐냐에 따라서 4개중에 선택이 되거든요. 그럼 이게 주소 공간을 기준으로 따져보면 여기가 000부터 여기가 3FF까지 그 다음에 400부터 7FF 이런 식으로 주소가 마치 연속적으로 연결된 것 같은 효과를 얻을 수 있게 되요. 그죠?
44. 그래서 이 주소 공간을 한번 써보는 연습을 반드시 해야 되고요. 이 써보는 연습이 그대로 우리 퀴즈나 연습문제 뒤쪽에서 여러분 스스로 하는 것에 나와 있으니까요. 반드시 연습을 꼭 해보시기 바랍니다.
45. 자. 이제 Main Memory 모듈을 디자인하는 마지막 예로써 8bit 컴퓨터를 위한 Main Memory 모듈을 총체적으로 한번 디자인을 해보겠습니다. 기억장치 모듈의 설계 순서라고 쪽 있는데요. 이게 중요한 건 아니고 개념적으로 이게 어떻게 설계해야 될 것인거 머릿속에 잘 들어와야 되겠죠.
46. 그래서 한번 읽어보겠습니다. 컴퓨터 시스템에 필요한 기억장치 용량을 결정을 하고요. 그 다음에 사용할 칩들을 결정해야 됩니다. 그리고 주소 표를 작성하라고

되어 있습니다. 그런 다음에 세부 회로를 설계하시면 되는데요. 여기는 말들이 중요한 건 아니고 실제로 설계를 잘 하시면 되죠. 보겠습니다.

47. 자 이제 갖고 있는 용량이라는 게 뭐냐면 1K byte짜리, 1K byte면 1K x 8입니다. Byte는 bit로 변경하면 8이기 때문에 1K x 8 RAM칩이 있고요. 그 다음에 1K Byte짜리 ROM이 있습니다. 이제 이게 실제로 8bit 짜리 마이크로 컴퓨터의 용량이 이렇게 1K x 8짜리 두 개가 있어야 되는데 실제 내가 가지고 있는 칩은 그런 것들이 가지고 있는 것이 아니라 여기 있는 것처럼 256 x 8짜리를 가지고 있다는 의미예요. 그러니까 부족하죠. 256은 1K가 안되니까요. 그 다음에 1K x 8짜리 ROM을 하나 갖고 있고요.
48. 이런 것밖에 가지고 있는 칩이 없다고 한다면 도대체 어떻게 설계를 해야 되는 거냐. 당연히 이거 4개 사용해야 되죠. 이거 4개 사용해서 앞에다 4를 곱해야 되는 거죠. 앞에다 4를 곱해야 되니까 주소공간을 확장을 해야 되죠. 저건 이제 직렬로 연결한다고 표현하겠죠. 이 뒤에 이 걸 확장하는 것은 병렬로 연결한다는 표현을 썼고요. 이제 앞의 4 x 4를 해서 주소공간을 확장하는 것은 직렬로 연결한다고 표현을 했습니다.
49. 자 Address Table을 볼게요. Address Table을 어떻게 구성하라 라는 조건이 있는데요. Address 영역을 처음에 RAM같은 경우에는 000H번지 H는 Hexadecimal이라는 의미니까 000번지부터 시작을 하고 RAM은, ROM은 800번지부터 시작을 해라라고 되어 있습니다. 800번지. Hexadecimal이죠. 이걸 Binary로 바꾸면 8같은 경우에는 1000이 되죠. 그리고 00은 똑같이 0000 이 0도 0000.이렇게 되는거죠. 이런 식으로부터 이 번지부터 시작해서 채우라는 의미입니다.
50. 자 우리가 이제 그 사용하는 칩들의 주소공간을 일일이 하나하나 따져보겠습니다. 먼저 편한 ROM부터 볼까요. ROM같은 경우에는 아까 800부터 시작하라고 그랬으니까요. 주소공간이 10이 돼. 여기가 Don't Care죠. X X 여기는 여기 있는 값이 이제 다 0일수도 있고 최저가, 가장 빠른 주소가 다 0일 수도 있고 가장 아래 주소, 아래에 있는 주소 다 1이죠. 그러니까 10 000000부터 10 111111까지 주소공간을 ROM이 차지하고 있다는 얘깁니다. 그런 얘기고,
51. 그 다음에 RAM칩같은 경우에는 RAM이 앞서 말씀드렸지만은 256 x 8이었어요. 그러면 256은 2의 8승입니다. 그러면 Address Line이 총 8개가 있어야 된다는 의미거든요, 그래서 8개에 대해서는 Don't Care, 그리고 상위bit 2개에 대해서는 우리가 Decoder를 이용해서 2x 4의 Decoder를 만들어 내야 됩니다. 그래서 00일 때는 RAM 1이 선택되게 하고 01일때는 RAM 2가 선택되게 하고. 10일때는 RAM3, 11일때는 RAM4가 선택이 되게 되어 되겠죠.
52. 그걸 이제 헥사로 변형을 해야 될것같으면 여기서 All Zero부터 시작해서 0000부터 이거는 4개씩 끊어서 봐야되니까요. 이걸 이렇게, 아. 여기서부터 보죠. X라

고 되있는게 0일수도 있고 1일수도 있으니까요 00000000이고요. 11111111하면 FF입니다. 그래서 00부터 FF까지.

53. 이거는 이것도 역시 00000000일 수 있으니까 이거는 100이 되겠쥬, 그래서 100부터 시작해서 이게 전부 다 1이면 1FF까지. 이렇게 되겠쥬. 그래서 RAM2는 100부터 1FF까지 주소공간을 채우게 되고요.
54. 그 다음에 RAM3같은 경우에는 역시 마찬가지로 계산하면 200부터 2FF까지 RAM4도 마찬가지로입니다. 300부터 3FF까지 그렇게 되면 RAM과 ROM 사이의 약간 뜨쥬. 중간에 공간이 뜨긴 하는데 처음의 설계 조건이 그랬기 때문에 뜬 채로 놔 둥겁니다.
55. 보시겠쥬습니다. 여기 보시면 CPU 8bit짜리고요. Data는 8bit로 연결됩니다. 조건이 그랬고요. 그 다음에 여기 보시면 이제 A0부터 7까지는 모든 칩에 연결이 되쥬. 모든 칩에 연결이 되고, 그 다음에 Read, Write. 모든 칩에 연결이 되는데 ROM에 대해서는 Write는 연결되지 않게 되고요.
56. 그 다음에 A8하고 A9를 이용해서 Decoder를 이용해서 2 x 4 입력이 2개 출력이 4개, 선택을 내게 됩니다. 그래서 칩 4개 RAM1, RAM2, RAM3, RAM4중에 하나를 선택을 하게 만듭니다. 그쥬.
57. 그 다음에 이제 우리 Chip Select에 연결되는 게 A11입니다. 중간에 떴쥬. 800부터 시작해야 되기 때문에 ROM의 주소가 800부터 시작해야 되기 때문에 A10이 뜹니다. A10은 사용하지 않아요. 그래서 A11을 연결을 하는데 이거를 Chip Select에 연결을 하는거쥬. 그래서 저게 0이면 RAM중에 하나가 선택이 되게 하고 저게 1이면 ROM이 선택하게, 하게 되면 그러면 우리가 원하는 8bit 마이크로 컴퓨터를 위한 Main Memory 모듈 전체를 디자인 할 수 있게 됩니다.
58. 내용 자체가 제가 조금 너무 화살표가 복잡해서 이걸 하나 말씀을 안 드렸거든요. 근데 이게 그렇게 어렵지 않아요. 이렇게 한번 쭉 따라가면서 보시면 다 화살표가 연결된 데는 다 이유가 있거든요. 그래서 한번 이 화살표가 제대로 연결 되었는지 한번 여러분들이 검사를 해보시길 꼭 부탁드립니다.