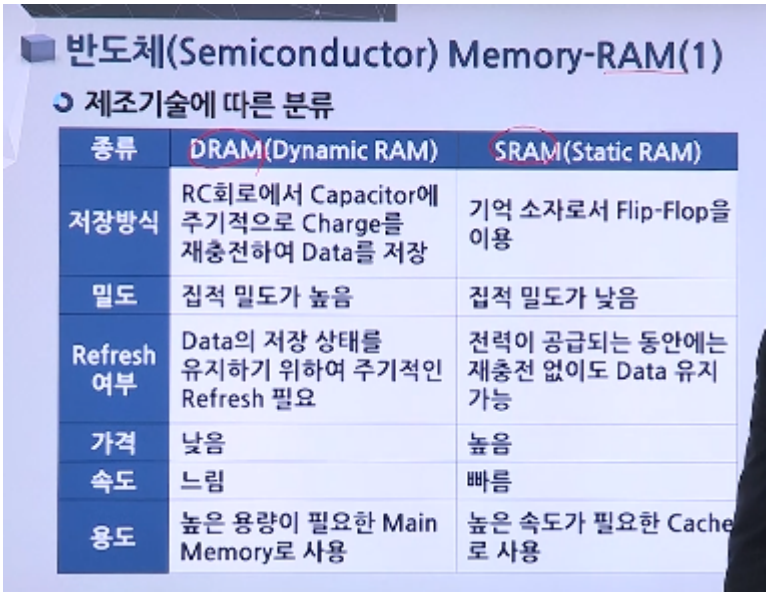
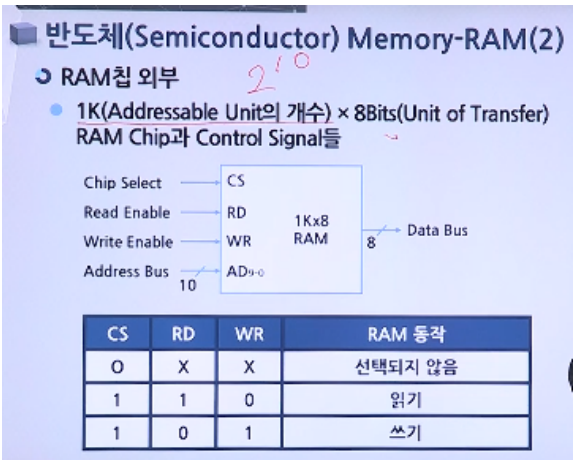
23강_Semiconductor Memory

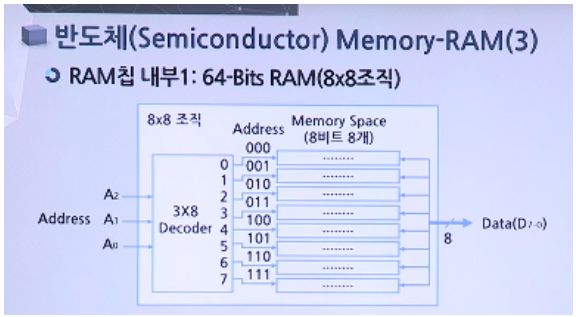
Created	@Aug 13, 2020 12:04 PM
Tags	RE

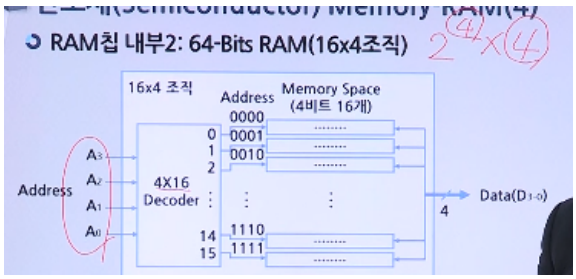


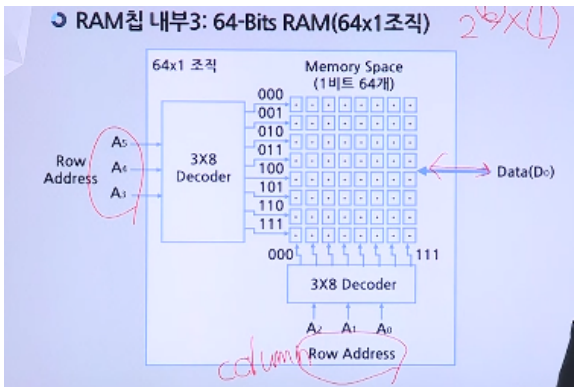


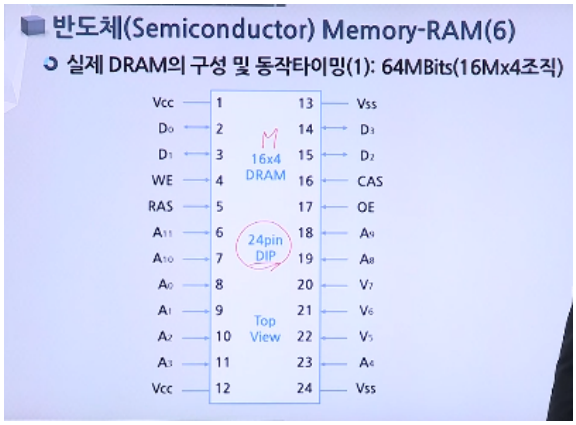
컴퓨터의 K는 1024~ : Address의 Bit 개수가 10비트. Address Line이 10개란 얘기

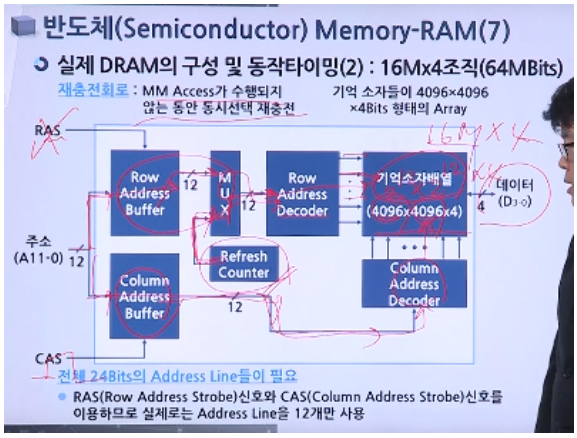
Address가 3 Bit 입력, 3×8 decoder를 통과시키게 됨

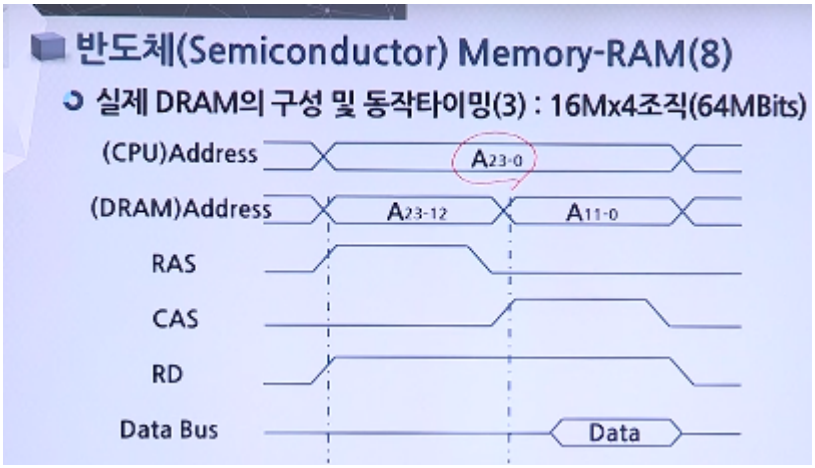

반도체(Semiconductor) Memory-RAM(3)
RAM칩 내부1: 64-Bits RAM(8x8조직)
8x8 조직
Address
Memory Space
(8비트 8개)
3X8
Decoder
Data(D7-0)
8

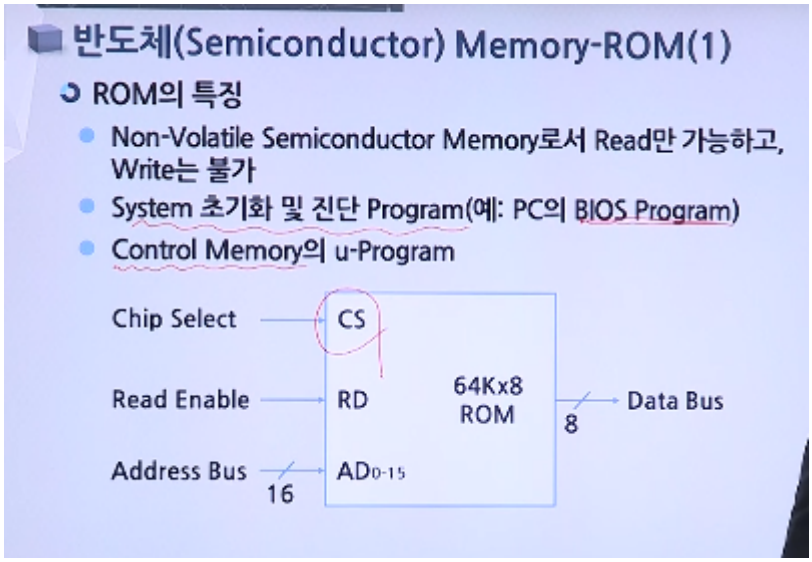

RAM칩 내부2: 64-Bits RAM(16x4조직)
16x4 조직
Address
Memory Space
(4비트 16개)
4X16
Decoder
Data(D3-0)
4

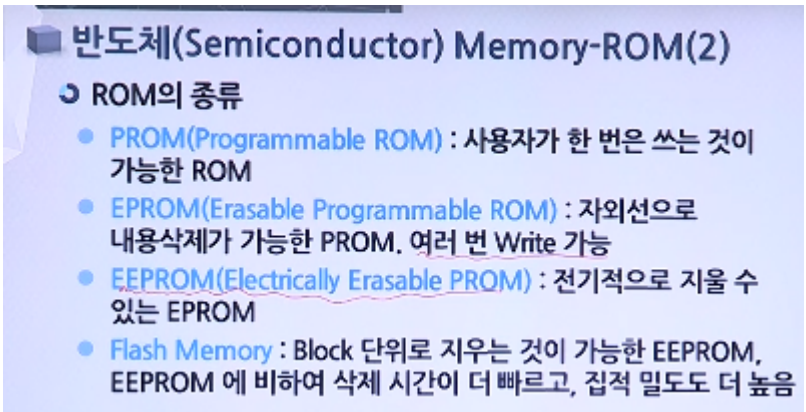

RAM칩 내부3: 64-Bits RAM(64x1조직)
64x1 조직
Memory Space
(1비트 64개)
Row
Address
3X8
Decoder
Data(D0)
3X8 Decoder
Row Address


반도체(Semiconductor) Memory-RAM(6)
실제 DRAM의 구성 및 동작타이밍(1): 64MBits(16Mx4조직)
16x4
DRAM
24pin
DIP
Top
View
Vcc
D0
D1
WE
RAS
A11
A10
A9
A8
A7
A6
A5
A4
Vss
D3
D2
CAS
OE
A9
A8
V7
V6
V5
A4
Vss


반도체(Semiconductor) Memory-RAM(7)
실제 DRAM의 구성 및 동작타이밍(2) : 16Mx4조직(64MBits)
재충전회로 : MM Access가 수행되지
않는 동안 동시선택 재충전
기억 소자들이 4096×4096
×4Bits 형태의 Array
RAS
Row
Address
Buffer
12
M
U
X
12
Row
Address
Decoder
기억소자배열
(4096x4096x4)
4
데이터
(D3-0)
주소
(A11-0)
12
Refresh
Counter
Column
Address
Buffer
12
Column
Address
Decoder
CAS
전체 24Bits의 Address Line들이 필요
RAS(Row Address Strobe)신호와 CAS(Column Address Strobe)신호를
이용하므로 실제로는 Address Line을 12개만 사용


반도체(Semiconductor) Memory-RAM(8)
실제 DRAM의 구성 및 동작타이밍(3) : 16Mx4조직(64MBits)
(CPU)Address
A23-0
(DRAM)Address
A23-12
A11-0
RAS
CAS
RD
Data Bus
Data


반도체(Semiconductor) Memory-ROM(1)
ROM의 특징
Non-Volatile Semiconductor Memory로서 Read만 가능하고, Write는 불가
System 초기화 및 진단 Program(예: PC의 BIOS Program)
Control Memory의 u-Program
Chip Select
CS
Read Enable
RD
64Kx8 ROM
8
Data Bus
Address Bus
16
AD0-15


반도체(Semiconductor) Memory-ROM(2)
ROM의 종류
PROM(Programmable ROM) : 사용자가 한 번은 쓰는 것이 가능한 ROM
EPROM(Erasable Programmable ROM) : 자외선으로 내용삭제가 가능한 PROM. 여러 번 Write 가능
EEPROM(Electrically Erasable PROM) : 전기적으로 지울 수 있는 EPROM
Flash Memory : Block 단위로 지우는 것이 가능한 EEPROM, EEPROM 에 비하여 삭제 시간이 더 빠르고, 집적 밀도도 더 높음

▼ 스크립트

1. 네. 이전 까지는 이제 우리가 뭘 공부했냐면요. 계층적 Memory에 대해서 공부를 했습니다. 계층적 Memory 시스템. 그래서 가격이 비싼 거는 CPU와 가깝게 붙이고 그다음에 가격이 싼 거는 CPU에서 좀 멀게 한다. 그래서 지역성의 원리에 의해서 SSD, 먼 쪽에 있는 그 데이터들을 Main Memory에 올린다. 미리 올린다는 의미죠. 그런 식으로 동작을 하게 되는 건데요.
2. 그 중에서 이제 그 다음에 우리가 공부를 해야 될 건 바로 이제 Main Memory에 대한 이야기입니다. 그 중에서 가장 중요한 RAM에 대해서 먼저 말씀드리겠습니다.
3. 이제 RAM을 만드는 기술은 물론 반도체를 이용해서 만들겠죠? 그러니까 크게 두 가지가 있습니다. D RAM이 있고요. 그 다음에 S RAM이 있고요. D RAM은 Dynamic RAM이라고 하고요. 그 다음 S RAM은 Static RAM이라고 합니다. 이름을 보면요 Dynamic이 더 좋은 것 같은데요. 실제로 그렇지 않고요. Static이 훨씬 더 좋습니다.
4. 저게 Dynamic이라고 하는 건 뭐냐면 주기적으로 저 Dynamic은 방전이 되요. 방전이 된다는 이야기는 원래 전하가 차있으면 1, 전하가 비게 되면 0. 뭐 이런 식으로 판별하는데 차있는 상태에서 계속 유지가 되면 좋은데 조금 조금씩 방전이 됩니다.
5. 그러니까 주기적으로 리프래시를 시켜줘야 되요. Refresh를 시켜준다는 이야기가 충전을 반복해줘야 된다는 의미죠. 그런 것이 D RAM입니다. 그거가 Dynamic한다는 의미로 Dynamic RAM이라고 하고요. Static은 그런 게 필요 없습니다. 1을 딱 저장해 놓으면 전원이 계속 들어가는 한은 그 1을 계속 저장을 하고 있습니다. 그래서 당연히 Static RAM이 더 비싸고 훨씬 좋은 겁니다.
6. RC회로라는 것은 저걸 회로이론을 공부해야 되는 거기 때문에 별도로 RC회로에 얘기는 안하겠는데요. 어쨌든 R이라고 하는 건 레지스터를 얘기하는 거고요 저항이고요. 그 다음에 캐패시터는 이제 충전기를 얘기하는 겁니다, 그래서 저항과 충전기 회로를 반도체를 이용해서 모델링을 해낼 수가 있어요. 해 낸 다음에 주기적으로 전하를 재충전을 해 주는거죠. 그죠. 그런 식으로 데이터를 계속 저장값을 유지하는 그런 형태의 반도체가 반도체 Memory가 RAM입니다. D RAM.
7. 그 다음에 Static RAM같은 경우에는 여러분 Flip-Flop 말 들어 보셨을 거에요 레지스터를 구성하는 한 Bit의 요소가 Flip-Flop이라고 말씀드렸는데요. 한 Bit를 저장할 수 있다고 그랬죠? 그걸 이용하는 게 S RAM입니다. 따라서 구현을 하는 게 S RAM이 더 어렵고요 돈도 더 많이 들어가고요. 그리고 비싸기도 훨씬 비쌉니다.
8. 밀도는 집적 밀도가 매우 높아요. D RAM은 그러니까 삼성에서 메모리를 개발하는 Main Memory RAM으로 개발하는 삼성에서 많이 파는 그러한 Memory들은

전부 다 D RAM입니다. 전부 다. 그래서 집적 밀도가 매우 높고요. 집적 밀도가 낮습니다. S RAM은 Flip-Flop을 구성해야 되기 때문에.

9. 그래서 집적 밀도가 낮은 S RAM은 어디 많이 쓰냐면요. 캐시에 많이 쓰입니다. 대신 속도가 훨씬 빠르죠 D RAM보다. Refresh 여부 데이터의 저장 상태를 유지하기 위하여 주기적으로 충전을 계속 해줘야 된다. 그런 문제가 있고요, 그 다음에 S RAM 같은 경우에는 전력이 공급되는 동안에는 재 충전없이 데이터 계속 유지할 수 있고요. 그런 장점이 있고 가격은 당연히 이게 높고 속도도 이게 빠르고요. 그 다음에 용도는 이게 주로 Main Memory에 주로 많이 쓰이게 됩니다. D RAM 같은 경우에는 그리고 S RAM같은 경우에는 캐시에 많이 쓰이게 됩니다.
10. 자 그러면 이제 RAM칩 외부가 어떻게 되있는지 한번 우리가 구성을 보도록 할까요? 지금 보시면 1K라는 표현을 썼습니다. 이거 우리가 어떻게 1K가 나오게 되는지 고민을 해볼께요.
11. 1K는요 컴퓨터에서 K는 1000이 아니라 1024라고 했어요. 그러면 그것은 2의 10승입니다. 2의 10승이 1024이고요. 그러니까 저건 무슨 이야기냐. Address의 Bit 개수가 10Bit라는 얘깁니다. 그러니까 Address Line이 10개란 얘기에요. 그러면 Address Line이 00000부터 11111까지 하면 총 2의 10승이니까 1024가지의 서로 다른 조합이 나올 수 있겠죠. 그러니까 주소 지점을 몇 군데 할 수 있냐면 1024할 수 있다는 이야기에요. 그러니까 Memory 공간 저장 할 수 있는 데이터 공간이 1024까지 가능하다는 얘깁니다. 1K라는 얘기가.
12. 그 다음에 8Bit는 뭐냐 그 하나 하나의 공간에 몇 Bit를 저장할 수 있냐는 얘기에요. 거기에 8Bit를 저장할 수 있다는 얘기죠. 그래서 여기 보시면 Addressable Unit의 개수. 쉽게 얘기하면 어느 마을이 있으면 그 마을에서 가정 집에 개수 그걸 얘기하는 거고 8Bit는 그 안의 사람들의 명수를 얘기하는 겁니다. 그걸 이제 우리가 Unit of Transfer라고 한다고 전에 얘기를 했었죠.
13. 자, 그럼 이건 어떻게 되냐. 이건 8K Bit RAM이 되겠죠. 계산을 하면 이건 1K가 되고 이건 8이 되니까 곱하면 실제 8K RAM이 되는 거죠. 그렇게 얘기를 하기도 합니다. 정확하게 Address Line의 개수하고 Data Bus Unit of Transfer의 폭이 결국 Data Bus의 폭이 되니까요. 그거까지 고려를 할 것 같으면 1K x 8 이렇게 표현을 하기도 하고요. 어쨌든 이런 RAM이 있습니다.
14. 그 RAM에는 어떤 입력신호가 있는지 하나 하나 살펴볼께요. 먼저 Chip Select. Chip Select는 뭐냐면요. 저게 Enable이 되어야만 Enable이 된다는 건 0으로 있다가 1로 떠야만 이 RAM칩이 동작을 하게 됩니다. 전체 전원이라고 생각하시면 됩니다. 실제 전원은 아니고요 스위치, 동작스위치, 저게 올라가 있어야만 RAM이 동작을 하게 되요. 저게 내려가 있으면 이 아무런 동작도 안합니다.

15. 그 다음에 이제 Read Enable 내가 만약에 RAM에서 데이터를 일고 싶다, 아니면 RAM에 있는 데이터를 RAM의 데이터를 쓰고 싶다 하면 Read Enable과 Write Enable을 Enable을 시켜야 되겠죠. 그 다음에 Address Bus는 10Bit가 날아옵니다. 10Bit는 Address Bus의 폭이 10개란 얘기고 RAM은 그 10개의 Address를 다 한꺼번에 받아드리게 되죠.

16. 10개가 아니라 10개의 Bit를. 그럼 총 이건 0번지서부터 1023번지까지 가능하겠죠? 10Bit니까. 총 이 RAM안의 쭉 이렇게 칸이 있어서 뭔가 저장 되있을텐데. 그거의 전체 개수가 방의 개수가 1024개 라는 의미입니다. 그리고 방안에는 몇 Bit가 들어가 있냐? 8Bit가 들어가 있다는 거죠. 그래서 출력으로 8Bit를 낼 수가 있는 거죠. 그걸 우리가 1K x 8RAM이라고 부릅니다.

17. 자 이 표를 한번 보실께요. 이 표는 어떻게 되냐면 CS가 0일 때 CS가 0 이라는 것은 Chip Select가 되지 않았다는 의미거든요. 그럴 경우에는 동작하지 않아요. 그러니까 Read가 됐든, Write Enable이 됐든, 뭐가 됐든, 그것이 이 x라고 대신 대있는데요. 저건 뭐냐면요. Don’t Care라고 합니다. Don’t Care. 중요하지 않다. 아무 문제 없다. 그러니까 영향을 전혀 주지 않는다는 거죠. 0이 됐든 1이 됐든 관계없이 라는 뜻이에요. 관계없이 Chip Select가 0이면 선택되지 않는다. 동작을 절대 안 한다는 얘기죠,

18. 그 다음에 Chip Select가 1이되면 그러면 Read Enable이 1로 뜨고 Write가 0이면 읽기 동작을 수행해야 되겠죠. Read가 떴으니까, 그 다음에 Write가 1로 뜨게 되면 쓰기 동작을 수행하게 됩니다. 혹 어떤 학생이 이거 두 개 다 1이면 어떡하냐, 이런 얘기할 수 있어요. 그러면 오동작이죠. 어떻게 동작할지 모르죠. 그죠. 그런 경우를 만들어내면 안 되겠죠 당연히.

19. 자 64Bit RAM의 예를 갖고요 그 내부는 어떻게 되있을까 고민을 좀 해볼께요. 먼저 64Bit RAM을 갖고 8 x 8조직을 만들어 보겠습니다. 8 x 8 조직. 그러니까 64Bit RAM은요. 여러 가지 방법으로 만들어 낼 수도 있습니다. 8×8로 만들어 낼 수도 있고요. 아니면 16 x 뭡니까, 그 다음은 4가 되겠죠. 4로 만들 수도 있고요. 32 x 2 로 만들 수도 있고요. 어쨌든 64가 나오면 되니까. 그 다음에 64 x 1도 만들 수도 있습니다. 여러 가지 방법에 대해서 하나하나 얘기를 해 볼께요.

20. 먼저 8 x 8를 얘기 해볼께요. 8 x 8 이렇 게 있죠. 그러면 2의 3승 x 8이 되죠. 자 그럼 Address Bus의 폭은 얼마일까요? 3Bit죠. 3Bit. 이 식 많이 이용한다 그랬죠? 그 다음에 Data Bus의 폭은 어떻게 될까요? 8Bit죠. 8Bit.그죠. 그러니까 Address Line은 3개가 붙어 있어야 되고요. Data Bus Line은 8개부터 있어야 됩니다.

21. 자 그런 조직으로 이제 구성을 하게 되면 여기 보시면은 Address가 3Bit가 들어오고 있습니다. 저 3Bit의 입력을 이용해서요 3 x 8 Decoder를 통과시키기 되는데요. 저건 어떤 3 x 8 Decoder라는 건 어떤 동작을 하게 되냐면 예를 들어서 이

3개가 전부 다 0이었다 000이었다. 그걸 10진수로 환산하면 0입니다. 그렇게 되면 여기만 1이 전송되고요. 나머지는 다 Zero입니다. 다 Zero가 출력됩니다.

22. 예를 들어서 이게 011이었다. 라고 한다면 그러면 10진수로 바꾸면 3이 됩니다. 그러면 3의 해당하는 것만 1로 뜨고요, 이게 이고요. 나머지는 전부 다 0이 되버립니다. 그죠. 이거를 10진수로 환산을 해서 10진수의 어떤 하나 이 8개 중의 하나만 1을 출력하고 나머지는 Zero로 출력하는 거죠.

23. 이제 그렇게 해야 되는 이유가 뭘까요? 그거는 이 Addressable Unit이 8개가 있는데 그 중에 하나를 선택해야 되거든요. 그래서 거기에 있는 데이터를 Data Bus로 빼내야 되거든요. 그렇기 때문에 8개중 하나를 선택하기 위해서 그중에 하나만 1로 출력하고 나머지는 다 Zero로 출력하게 됩니다.

24. 이 Memory Space는 8Bit 짜리가 8개 있다고 했어요. 8 x 8이니까 그래서 8개 중에서 하나 선택이 되면 선택된 것이 Data Bus로 빠져 나오게 되죠. 근데 Data Bus는 당연히 8Bit가 되겠죠. 그래서 우리가 이걸 갖다가 8 x 8 조직이라고 표현할 수 있고요. 숫자로 하면 2의 3승 x 8 여러 번 얘기합니다. 2의 3승 x 8 이런식으로 우리가 얘기를 할 수가 있겠습니다.

25. 자 똑같은 형태를 이제 16 x 4 조직으로 만들어볼까요. 저것도 역시 마찬가지로 2의 승수로 표현을 하게 되면 이렇게 됩니다. 16은 2의 4승이 됩니다. 그리고 거기다 곱하기 4가 되죠. 이것도 64인데요. 이렇게 따지면 Address Bus는 4이어야 겠네요. 그 다음에 Data Bus 개수는 역시 4개여야 됩니다. 그러면 여기 보시면 Address Bus가 4개가 들어오고 있고요. Decoder는 구성이 아까는 3 x 8 이었는데 지금은 4 x 16이 되어야 겠죠. 왜냐면 이 4Bit로 가질 수 있는 조합의 개수가 총 16가지 이니까 여기 출력은 반드시 16개가 되게 끔 만들어야 됩니다.

26. 저거 다 만들 수 있어요 굉장히 쉽게. 그중에 하나 선택이 됐습니다. 이게 선택이 됐어요. 그럼 이게 출력이 되야 되는데 중요한 건 뭐냐, 요 안에 들어가있는 방 Bit를 저장할 수 있는 Bit는 4개라는 얘깁니다. 아까 여기 4라고 했습니다. 그래서 Data Bus도 역시 4개로 이루어졌죠. 그래서 4Bit의 데이터를 선택된 하나를 출력을 하게 되게 되죠. 이걸 우리가 16 x 4조직이라고 하고 그 조직 내부에는 여기 보시는 것처럼 4Bit 짜리의 Memory 공간 저장 할 수 있는 공간이 16개가 존재하게 됩니다. 그런 형태가 되는 겁니다.

27. 자 마지막으로 이제 64 x 1을 고민해볼까요. 이건 좀 복잡해 지겠네요. 64 x 1은 2의 6승 x 1입니다. 이게 재밌는 게 이거 Address Bit가 6개어야 된다는 얘기고요. Data Bus는 1Bit 1개여야 되는 거에요. 지금 여기보시면 데이터가 1개만 출력되고 있어요. 1Bit만 그 다음에 이제 6개의 Bit가 입력이 되어야 되니까 여기 보시면 Row Address 그다음에 Column인데요. 여기가 Row가 아니고요. Column입니다. Column이죠. Column Address. Column.

28. Column Address로 3Bit가 입력이 되고 Row Address가 3Bit가 입력이 되어야 되요 그러면 여기 Decoder가 3 x 8 Decoder를 사용해야 되고 여기도 3 x 8 Decoder를 사용해야 되겠죠. 그러면 여기서는 특이한 게 이 8개 중에 어느 하나가 1이 되고 나머지가 0이 되어야 되고요. 여기도 동시에 8개 중에 1개만 1이되고 나머지는 0이되어야 됩니다. 그러니까 행에서 1개 선택되고 열에서 1개 선택되고 그러면 이 64개중에 어느 하나가 선택이 되겠죠. 그게 Data Bus로 출력이 되는 거죠. 이렇게 되면 이건 64 x 1 조직이 되는 겁니다.

29. 지금 여태까지 한 게 굉장히 복잡해 보일 수도 있는데요. 곰곰이 생각하면요. 굉장히 쉬워요. 굉장히 쉽게 우리가 Memory 내부 조직을 어떻게 구성해야 되는지를 이해를 할 수가 있을겁니다.

30. 실제 이제 D RAM의 구성을 한번 볼게요. 여기 보시면 16M x 4 D RAM이고요. 그 다음에 24핀 dip 라고 했어요. 24핀이라는게 핀 수가 24개라는 이야기고요. dip는 딥이라고 얘기합니다. dual in-line package.

31. 듀얼이라는 게 양쪽으로 쌍으로 있다는 거고요 in-line이라는 게 정렬이 되있다는 얘깁니다. 즉 여기 핀들이 쭉 맞게 정렬이 되있죠. 쌍으로 양편에 딱 정렬이 되있어요. 그래서 쌍으로 핀으로 정렬되있어서 나와 있으면 딥이라고 얘기합니다. dual in-line package, 패키지표현을 왜 쓰게 되냐면 실제로 반도체는 이만큼 밖에 안 들어가 있습니다. 실제 반도체에는. 여기에 뚜껑을 덮어논 형태에요.

32. 그렇게 되고요. vcc는 뭘 얘기하냐면 전원을 얘기합니다 +5volt. 이 vss 는 sink 얘기하죠, 0 volt 짜리 음극. 그러니까 전원이 이 양단에 연결이 되있어야 되겠죠. 보조 전원이 여기 아래에도 있네요. vcc 하고 vss 둘 중에 어느 쪽으로든 연결을 해주면 되겠고요. 그 다음에 이거든 16 M x 4 니까요. M는 얼마일까요. M는 K x K입니다. K는 얼마입니까? 2의 10승이죠. 그럼 이건 어떻게 되냐 계산하면 2의 24승이 되겠네요. 다시요. 2의 24승이 되겠네요. 그죠.

33. 그러니까 Address Line이 24개가 있어야 되고요. Data Line이 4개가 있어야 한다는 의미에요. Data Line이 4개라는 건 여기 D0부터 D3까지 4개가 나와 있고요. 그 다음에 Address Line이 24개인지 한번 살펴봅시다. Address Line은 여기서부터 여기까지 그리고 또 여기서부터 여기까지네요. 그럼 이게 24개인가요? 아니에요. 12개에요, 12개. 12개밖에 안되요. 근데 24개 여야 되거든요. 이게 16M x 4가 될려면.

34. 그렇게 하는 방법이 있습니다. 그걸 어떻게 하냐면 여기 보시면 이제 Column Address Strobe, CAS 그다음에 Row Address Strobe라는 게 있어요. 저걸 이용해서 Address를 12Bit 짜리를 연속으로 2번을 주게 됩니다. 2번을 줘서 한번은 Row Address를 Enable을 시켜서 Row 입력으로 넣고 한번은 Column Address를 Enable을 시켜서 Column Address로 넣고 그러니까 두 번

Address를 넣어서 그걸 합치는 거죠. 그러면 24개의 Address가 만들어지게되겠죠. 그런 방식으로 동작을 하게 되는데.

35. 있다가 그건 타이밍도를 보면서 하도록 하고요. 그 다음에 Write Enable이 있습니다. 또 Output Enable이라는 게 있죠. 그러니까 쓸거냐, 읽을거냐, 그거에 따라서 둘 중에 하나를 Enable을 시켜줘야 되겠죠.

36. 자 내부 구조를 한번 보실게요. 앞서 본 16M x 4조직에서 내부는 어떻게 되있는지 보겠습니다. 지금 여기 보시면 16M x 4이기때문에요. 이게 4096 x 4096이 되죠. 4096은 2의 12승 1024가 2의 10승이니까 거기에다 4를 곱한 게 4096이 되기 때문에요. 2의 12승이 4096이 되고요. 곱하기 이건 또 2의 12승이 되죠. 곱하기 4가 되죠. 이걸 2개 더하면 2의 24승이 되겠죠. 그래서 이건 total 16M가 되는 겁니다. 그런 계산에 의하면 16M x 4가 되죠. 이거는 저장하는 저장 공간이 16M 개가 있고 각 저장공간에 4Bit씩 저장 되있다 라는 의미입니다.

37. 이제 그러면 Address가 총 24Bit가 있어야 된다고 했죠? 자, 어떻게 하나 봅시다. 주소가 12Bit가 한꺼번에 들어옵니다. 들어오는데 한번은 Row Address Strobe를 Enable을 시켜요 그러면 그게 주소가 일로 빠져 나가게 되요. 그래서 일로 이렇게 연결되가지고 Row Address Decoder를 통과하게 되죠. 이 입력이 12개가 들어오니까 이 선이 몇 개여야 되냐면요, 2의 12승개가 되어야 되요. 그러니까 4096이 되어야죠.

38. 그 중에 하나를 선택해서 Row 행을 갖다가 선택을 하고요. 그다음에 이걸 죽여버리고 이걸 Enable을 시킵니다. 그런 다음에 이 주소를 넣어요. 그럼 12Bit가 이쪽으로 들어오게 되죠. 그래서 Column Address Decoder를 통해서 또 여기에 있는 4096 하나중에 선택을 하게 됩니다.

39. 그럼 그중에 4Bit를 선택하게 되겠네요. 행도 선택되고 열도 선택되어있으니까 어떤 위치에 저장된 데이터가 특정이 딱 됬어요. 그럼 4Bit를 내보내는 거죠.

40. 중간에 이게 Refresh Counter라는 게 있는데요. 아까 D RAM같은 경우에는 주기적으로 충전을 시켜줘야 된다고 했는데 여기에 카운터를 하나 만들죠. 카운터는 숫자 세는거거든요. 0,1,2,3,4,5,6,7.

41. 그걸 이용해서 주기적으로 Column Address 버퍼를 통해서 데이터가 이동할 때 이 때는 Row는 쉬고 있거든요. 아무 짓도 안하고 쉬고 있거든요. 그 경우에 여기에다 MUX를 통해서 MUX는 두 개의 입력 중에 하나를 선택하는 겁니다. 그러니까 Row Address Strobe Row Address가 들어올 때는 이걸 선택해서 이걸 내보내고, 그리고 여기가 이걸 선택해서 내보내는 순간 Column을 내보내는 순간 Column Address를 내보내는 순간 Refresh Counter를 선택해서 이게 여기에 있는 Decoder를 통해서 기억소자에 있는 RC회로를 재충전하게끔 그렇게 만들어 주는 거죠. 그런 식으로 이제 구성이 되겠습니다.

42. 이건 이제 재충전 회로로서 Main Mamory Access가 수행되지 않는 동안 동시 선택을 통해서 재충전을 하게 된다라는 의미가 방금 전에 말씀드린 겁니다. 자 그 다음은 전체의 24Bit의 Address Line들이 필요로 한데요. 실제로는 12Bit를 이용한다고 그랬죠? 그래서 Row Address Strobe 신호하고 Column Address Strobe를 이용하므로 실제로는 Address Line이 12개면 충분합니다.

43. 자, 그럼 이제 타이밍도를 보겠습니다. 타이밍도는 CPU에서는 분명히 24개의 Address를 Memory에 전달을 해주지만 D RAM이 입력받는 건 이런 식으로 입력받게 되는 거죠. 먼저 상위 12Bit를 입력시킬 때는 Row Address Strobe를 올리고 그 다음에 하위 12Bit를 받을 때는 Column Address Strobe를 Enable을 시키고, 그다음에 Read 신호는 Address를 줄 때부터 동시에 올라와 있어야 겠죠.

44. 그럼 데이터 버스는 계속 데이터가 없다가 Row Address 그러니까 Row Address가 다 입력되고 Column Address가 입력되고 약간의 시간이 지났을 때 이쯤되서 Memory가 이제 어떤 위치의 데이터인지를 측정을 딱 하게 되죠. 그럼 이제 실제 Data Bus에다가 실제 4Bit 짜리 데이터를 싣게 되는 거죠. 그럼 이제 여기에 데이터가 전송이 되게 되는거죠. 이런 방식으로 동작 타이밍을 갖게 됩니다.

45. 자 그 다음에 이제 ROM. RAM에 대해서는 전반적인 얘기를 다 했고요. ROM에 대해서 얘기를 해보겠습니다. ROM은 먼저 Non-Volatile Semiconductor Memory다. 그러니까 비휘발성이고, Read만 가능하고, Write를 할 수가 없습니다.

46. 이제 ROM은 어디에 많이 쓰냐면요. 우리가 처음에 부팅할 때, 부팅할 때는 규정된 어떤 방식의 프로그램이 돌아가는 거거든요. 부팅한다는 것의 의미가 주변장치를 찾아내고 그 주변장치에 해당하는 어떤 세팅값 세팅하고 뭐 이런것들이 있어야 되요. 그리고 제대로 지금 RAM이 동작하는지 진단해야되고 초기화해야 되고 이런 작업들을 하게 되거든요. 그런것들은 정해져 있어요. 그래서 그거 같은 경우에는 ROM에다가 프로그램을 잘 해놓은 다음에 우리가 파워를 딱 누르면 그 프로그램이 쭉 계속 반복해서 돌아가거든요.

47. 그러니까 파워를 누르는 순간 그 프로그램을 쭉 돌려서 초기에 시동을 하기 위한 프로그램들을 쭉 돌려 놓는 거죠. 그래서 시스템 초기화 및 진단 프로그램을 저장하는 Memory가 있는데요. 그걸 우리가 ROM의 프로그램을 저장을 하고 그 프로그램 이름을 바이오스 프로그램이라고 부릅니다. Basic Input Output System이라고 하죠. 그래서 바이오스 시스템을 저장하는데 ROM이 많이 쓰이겠죠.

48. 그 다음에 동시에 컨트롤 Memory라고 아마 제가 얘기한 적이 있을 거에요. 이건 뭘 얘기하냐면 컨트롤 유닛 안에 마이크로 프로그램이 저장되있는 Memory 기억나시나요? 거기에도 역시 고정된 어떤 프로그램이 있기 때문에 거기에도 역시

ROM을 쓰게 됩니다. ROM은 일반적으로 이런 형태를 갖죠. 역시 마찬가지로 Chip Select가 있어야 되고요. ROM은 WriteEnable은 없습니다. Read Enable만 당연히 있겠죠.

49. 그 다음 Address Bus는 지금 이 예에서는 16Bit가 입력이 되고요. 16Bit니까 2의 16승입니다. 16승은 2의 6승에 K입니다. 그래서 64K가 되는거죠. 그다음에 출력 Bit는 8Bit가 나가거든요. 그래서 64K x 8 ROM이 되겠죠. 이런 식의 핀 구성이 일반적인 ROM의 블록도가 되겠습니다.
50. 네. ROM의 종류에는 여러 가지가 있습니다. 지금 여기 나와 있는 것처럼 PROM, EPROM, EEPROM, Flash Memory 이런 것들이 있습니다.
51. 기본적으로 ROM이라는 거는 지웠다 다시 쓰는 건 안됩니다. 근데 PROM같은 경우에는 프로그램들 ROM이라고 해서 사용자가 한번은 쓰는 것이 가능합니다. 그런 ROM이 있고요.
52. 그 다음에 EPROM이라고 해서 쓰는 것 뿐만 아니라 지우는 것도 가능합니다. 그러니까 쓰고 지운 게 가능하니까 RAM하고 크게 다를 바가 없는데 이건 이제 Non-Volatile 타입이기 때문에 그런 장점이 있겠죠 그래서 여러 번 Write가 가능하다. 근데 우리가 내용 삭제를 할 때 자외선으로 하는 방식이 있고 전기적으로 하는 방식이 있습니다. 전기적으로 내용을 삭제할 때가 있는데요. 그게 가능한 ROM의 이름을 EEPROM이라고 하게 되요. Electrically ROM 전기적으로 지울 수 있는 ROM 그걸 우리가 EEPROM이라고 합니다
53. 마지막으로 중요한 게 이 Flash Memory인데요. Flash Memory는 최근에 여러분들도 많이 사용을 하고 있죠. 대표적인 게 USB Memory죠. USB Memory는 분명히 전원이 연결되있지 않는데도 불구하고 그 안의 데이터는 보존이 되죠. ROM이거든요. ROM인데 ROM은 원래 지우고 쓰는 게 안된다 그랬는데, 이 Flash Memory같은 경우에는 지우고 쓰는게 거의 무한대까지 가능합니다.
54. 그래서 블락 단위로 지우는 것이 가능한 EEPROM을 우리가 Flash Memory라고 부르고요. EEPROM에 비해서 특히 또 장점이 뭐냐면 삭제 시간이 어마어마하게 빨라요. 굉장히 빠르게 지워지죠. 동시에 집적 밀도도 훨씬 높습니다. 더 많은 용량을 가질 수 있다는 거죠.
55. Flash Memory같은 경우에는 여러분 쓰는 USB Memory 뿐만 아니라 SSD 라는 것 있죠 여기서 얘기하는 SSD는 Secondary Storage Device 보조저장장치가 아니라 Solid State Disk. 요즘 하드디스크를 대체하는 제품들이 많이 나오고 있잖아요. 속도 엄청 빠르게요. 그거를 의미하는 SSD 거기에도 대표적으로 Flash Memory가 많이 사용됩니다. 그래서 앞으로도 아마 Flash Memory는 각광을 받는 매체가 될 것 같습니다.
56.