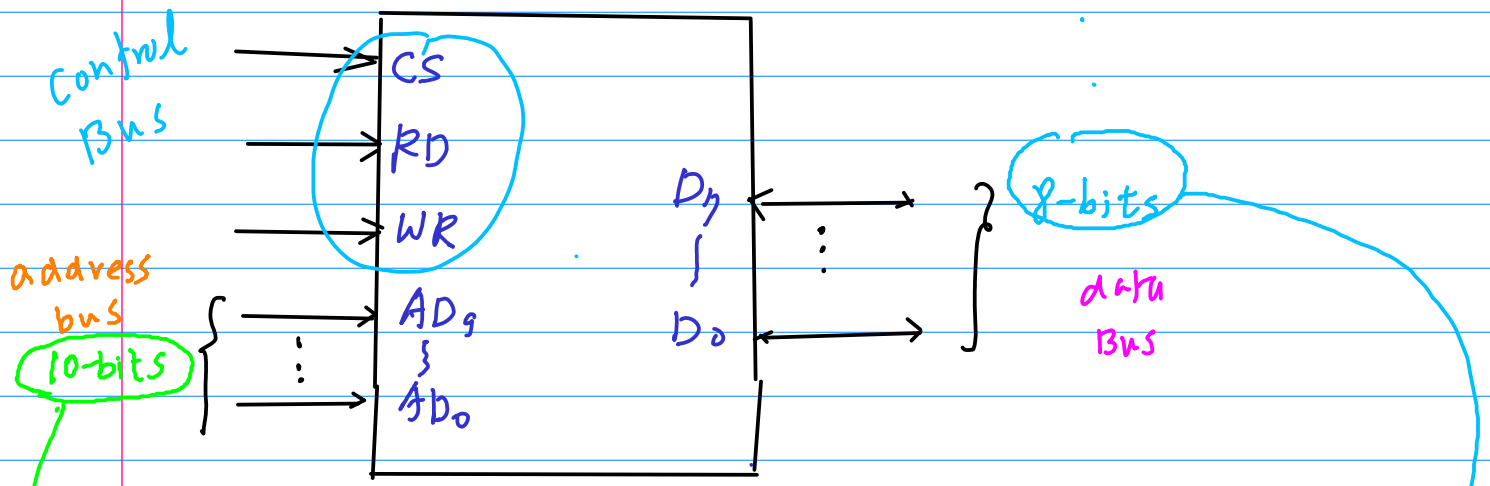
DRAM (H.1)

2015.12.11

www.wikiversity.org

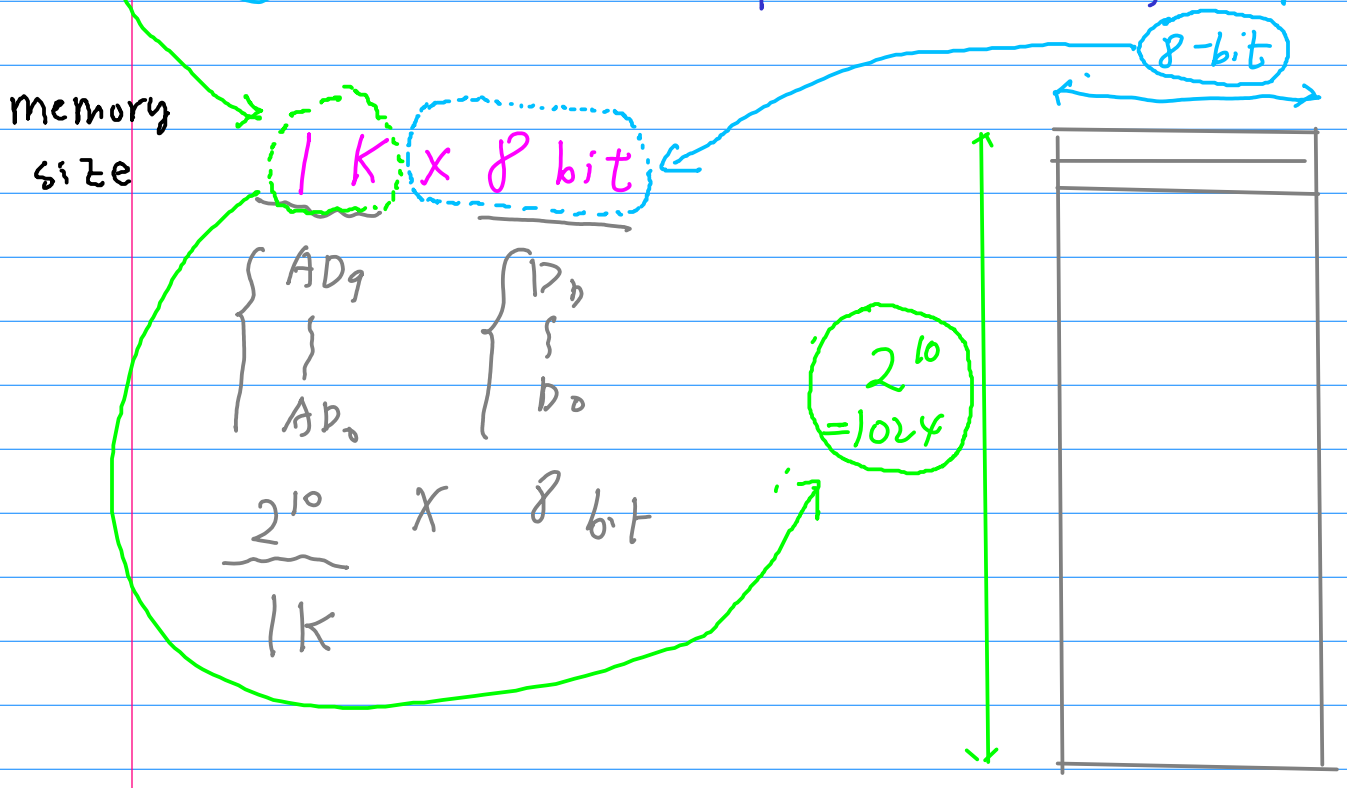
The necessities in Computer Organization

RAM (Random Access Memory)



Chip Select

CS	RD	WR	
0	X	X	선택 되지 않은
1	1	0	읽기 동작
1	0	1	쓰기 동작



DRAM --- capacitor ← 전하가 있으면 '1' 없으면 '0'

시간이 흐르면 전하가 사라질 수 있다 (leak)

→ refresh (주기적 재충전)

Dynamic

SRAM --- flip-flop

Static (refresh X)

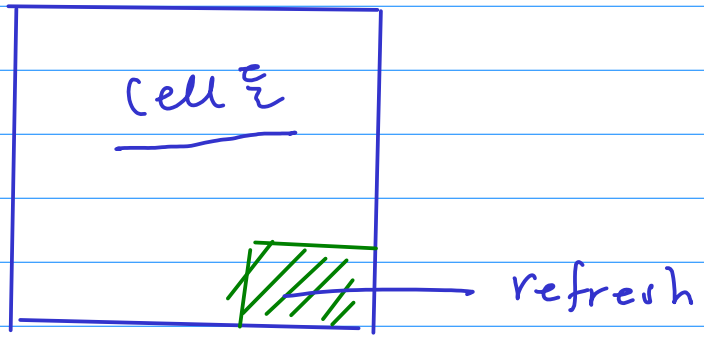
cell --- memory에서 1-bit를 저장하기 위한 기본구조

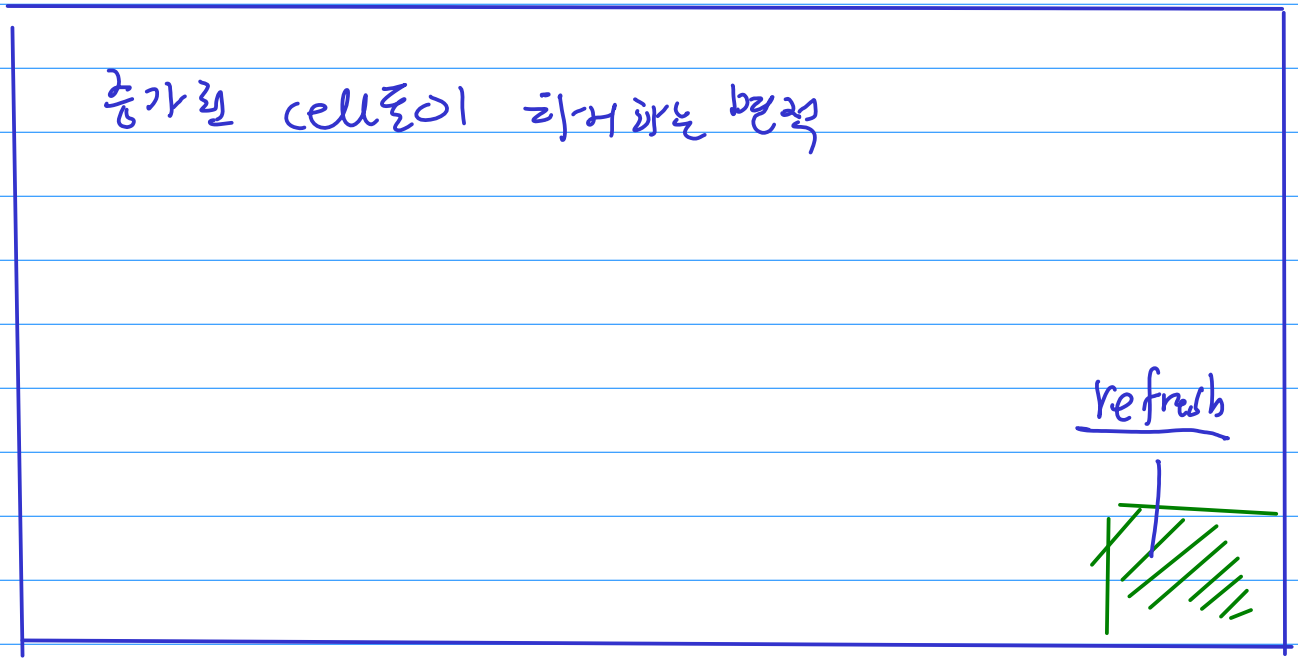
DRAM cell은 SRAM cell에 비하여 간단하므로
(적은 수의 transistor로 구현 가능)
반도체 chip 면적을 적게 차지한다.
단위 면적 당 많은 cell을 집적할 수 있어서
low cost integrate
↳ 대용량 main memory로 사용된다.

SRAM — fast - 비싸다 — Cache memory로 사용된다

→ refresh 회로도 chip에서 차지하는 면적이
DRAM의 용량이 커질수록
상대적으로 작아지기 때문에 가격에 영향이 없다

→ refresh 회로는 chip에서 차지하는 면적이
DRAM의 용량이 커질수록
상대적으로 작아지기 때문에 가격에 영향이 없다





64 bits

① 8 x 8 bit organization — address: 3-bits

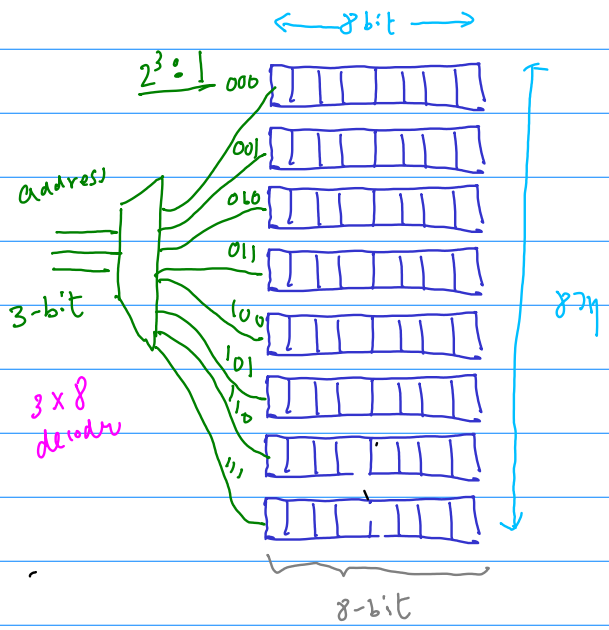
② 16 x 4 bit organization — address: 4-bits

③ 64 x 1-bit organization. — address: 6 bits
{ row addr: 3-bits
col addr: 3-bits

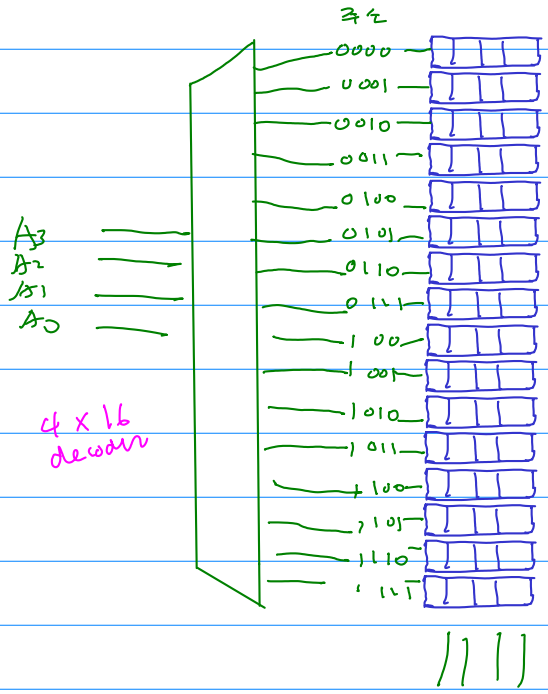
④ 16M x 4 bit

64-bit Memory Configuration

① 8 x 8 bit organization

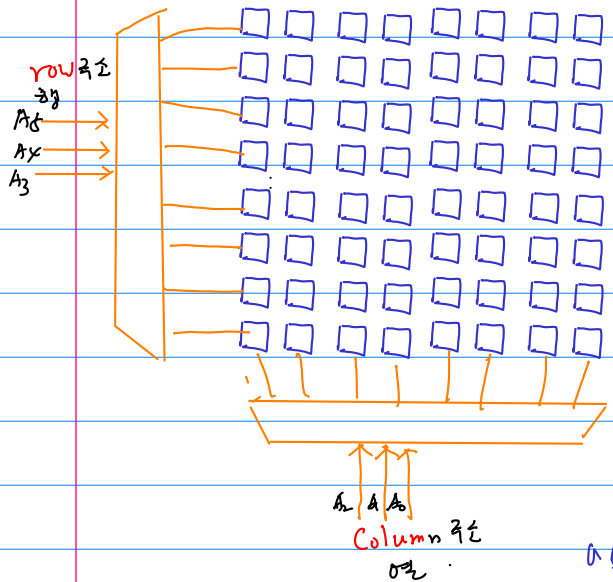


② 16 x 4 bit organization

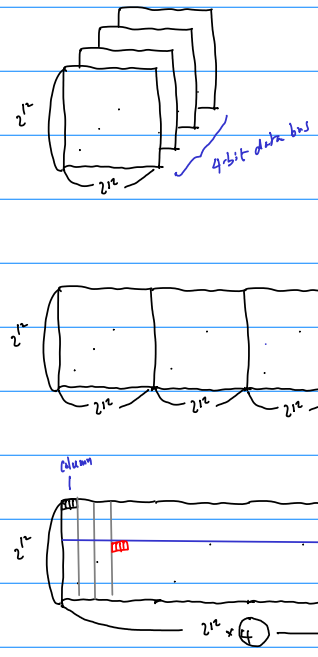


③ 64 x 1-bit organization.

주소 : 6-bit 데이터 : 1-bit

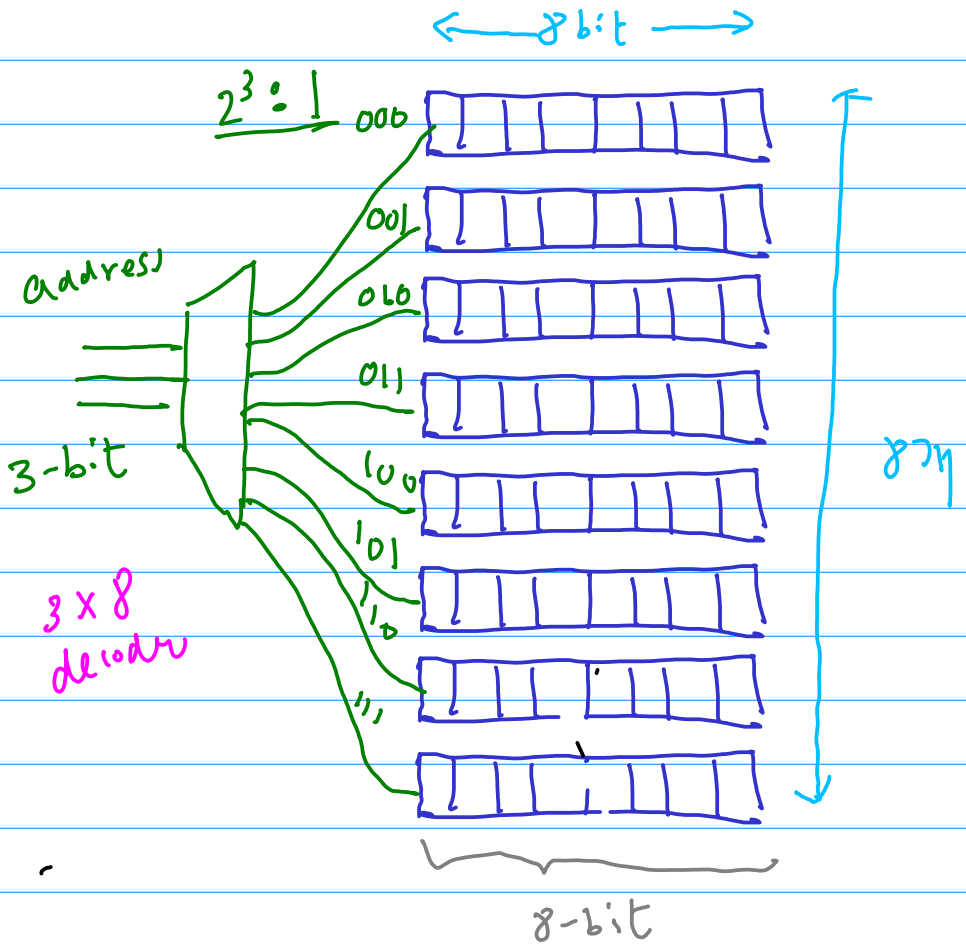


④ 16M x 4 bit



VLSI Layout

8 x 8 bit organization



만약 용량이 8Kbit → $1K \times 8\text{-bit}$ (2^{10})

address 10-bit, data 8-bit

$1K = 2^{10}$

$1M = 2^{20}$

$1Mbit \rightarrow 2^7 K \times 8\text{-bit}$

$2^{20} = 2^{17} \cdot 2^3$

$= 2^3 \cdot 2^{10} \cdot 8$

$= 2^7 K \cdot 8$

$128K \times 8\text{-bit}$

address 17-bit, data 8-bit.

용량이 $1Mbit = 2^{20}$ bits

$= 2^7 \cdot 2^{10} \cdot 2^3$ bits

$= 127 \cdot K \cdot 8$

$\frac{128K}{2^{17}} \times \frac{8 \text{ bits}}{1}$

address 17-bit, data 8-bit

용량이 $1Gbit = 2^{30}$ bit

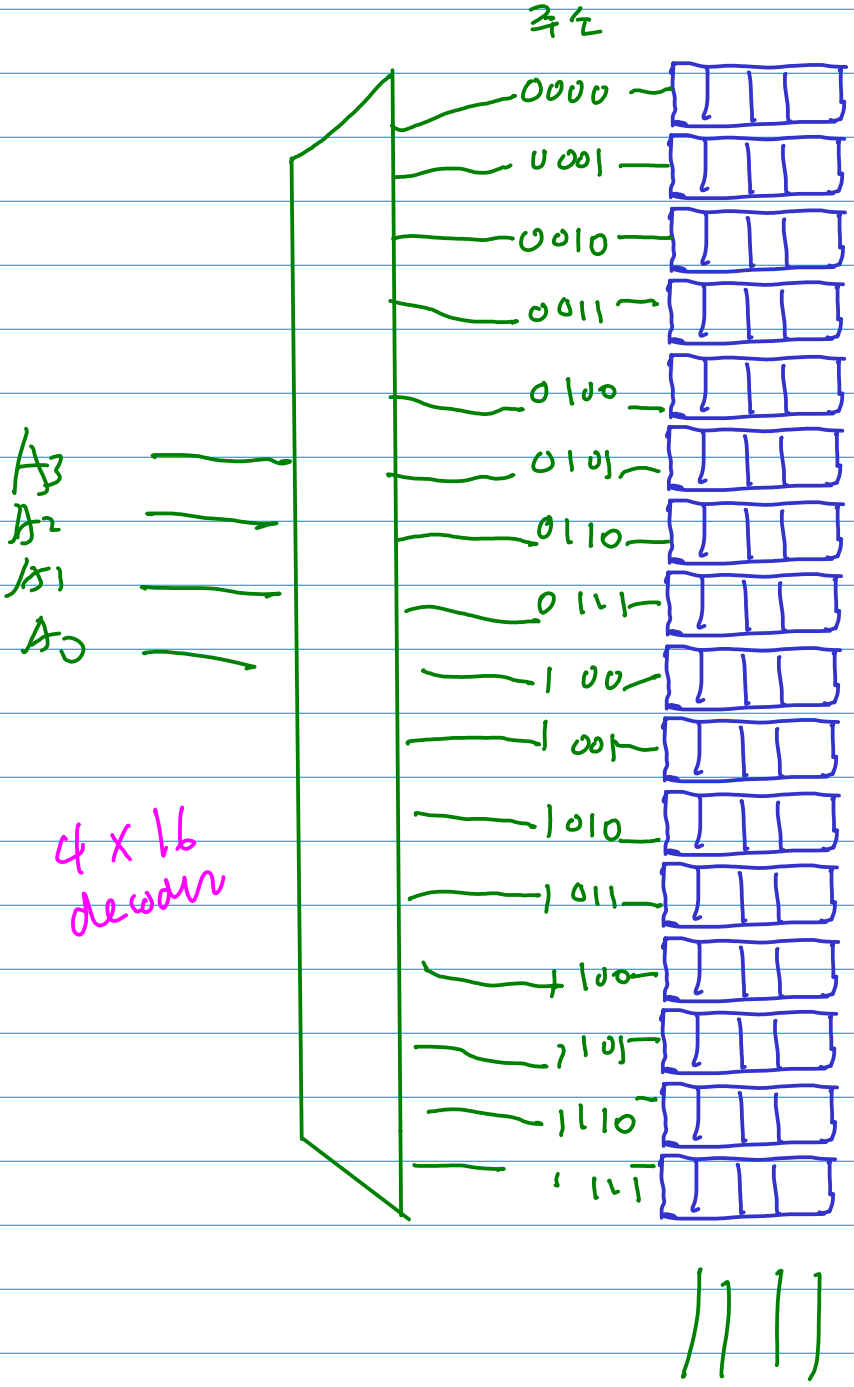
$= 2^7 \cdot 2^{20} \cdot 2^3$

$128M \times 8\text{-bits}$

address 27-bit, data 8-bit

64

② 16 x 4 bit organization



· 총용량이 8K bit

$2^3 \cdot 2^{10}$ — $2^{11} \cdot 2^2$

2K x 4bit

address : 11-bit data = 4bit

총용량이 1 Mbit

2^{20} bit $2^8 \cdot 2^{10} \cdot 2^2$

256K x 4bits

add: 18 data 4bits

총용량이 1 Gbit

2^{30} $2^8 \cdot 2^{20} \cdot 2^2$

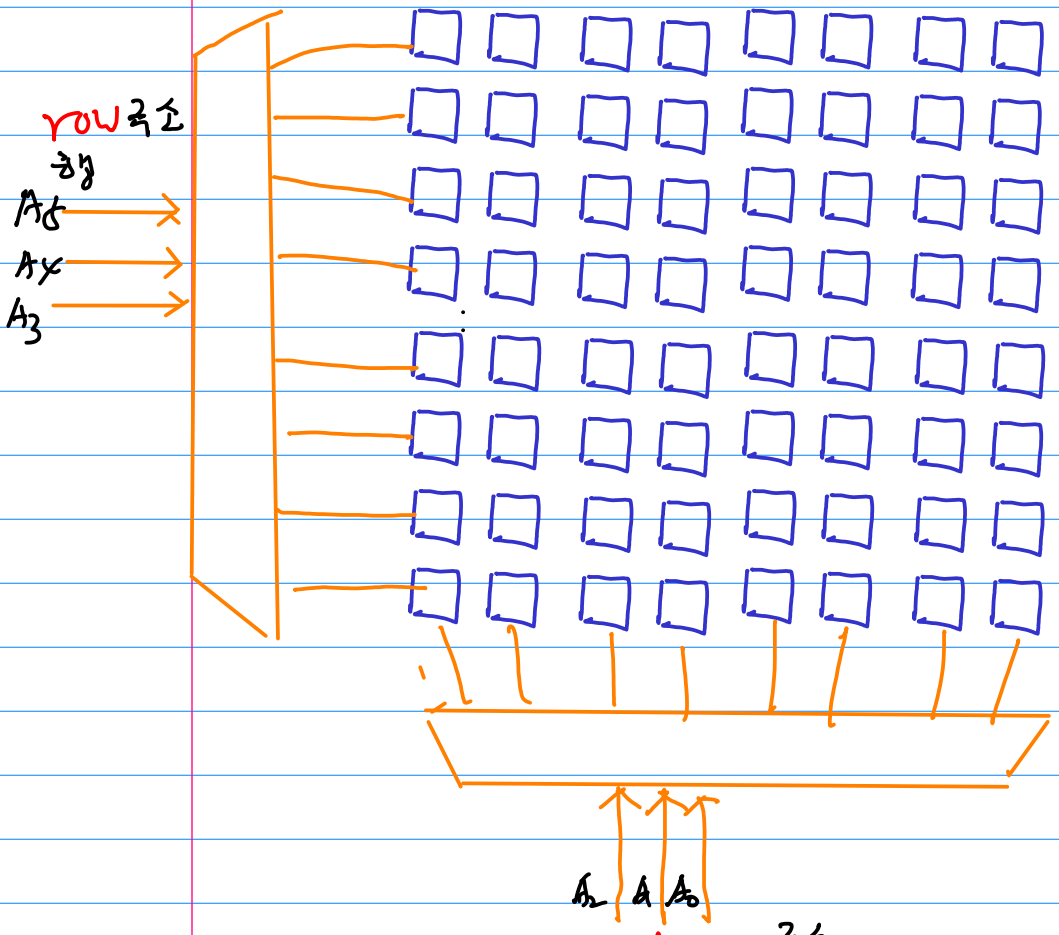
256 M x 4 bits

address 28 data 4bit

③ 64 x 1-bit organization.

2^6

address: 6-bit data: 1-bit

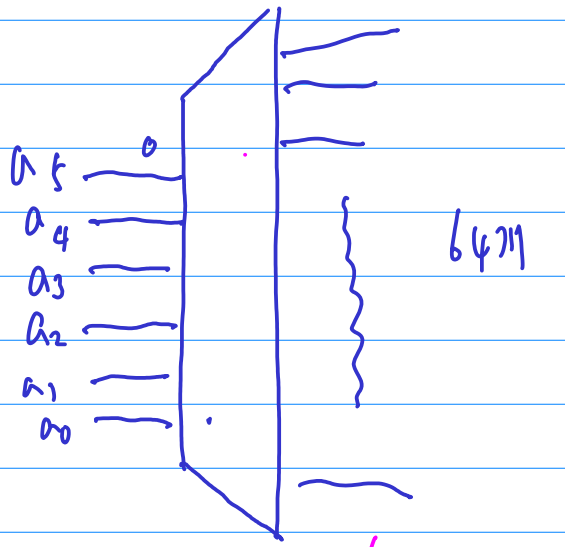


64 x 1-bit

2^6

data: 1-bit

address: 6-bit



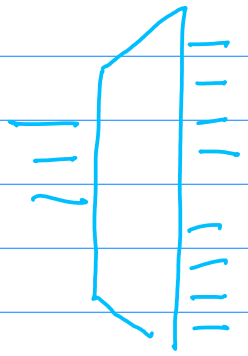
a_5 a_4 a_3 a_2 a_1 a_0

6x64 decoder

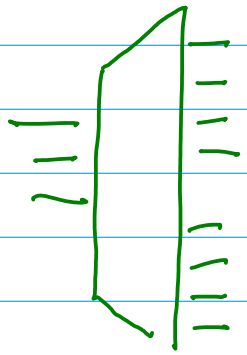
a_5 a_4 a_3 — row address

a_2 a_1 a_0 — colum address

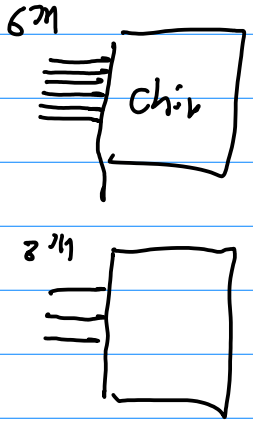
6x64 decoder



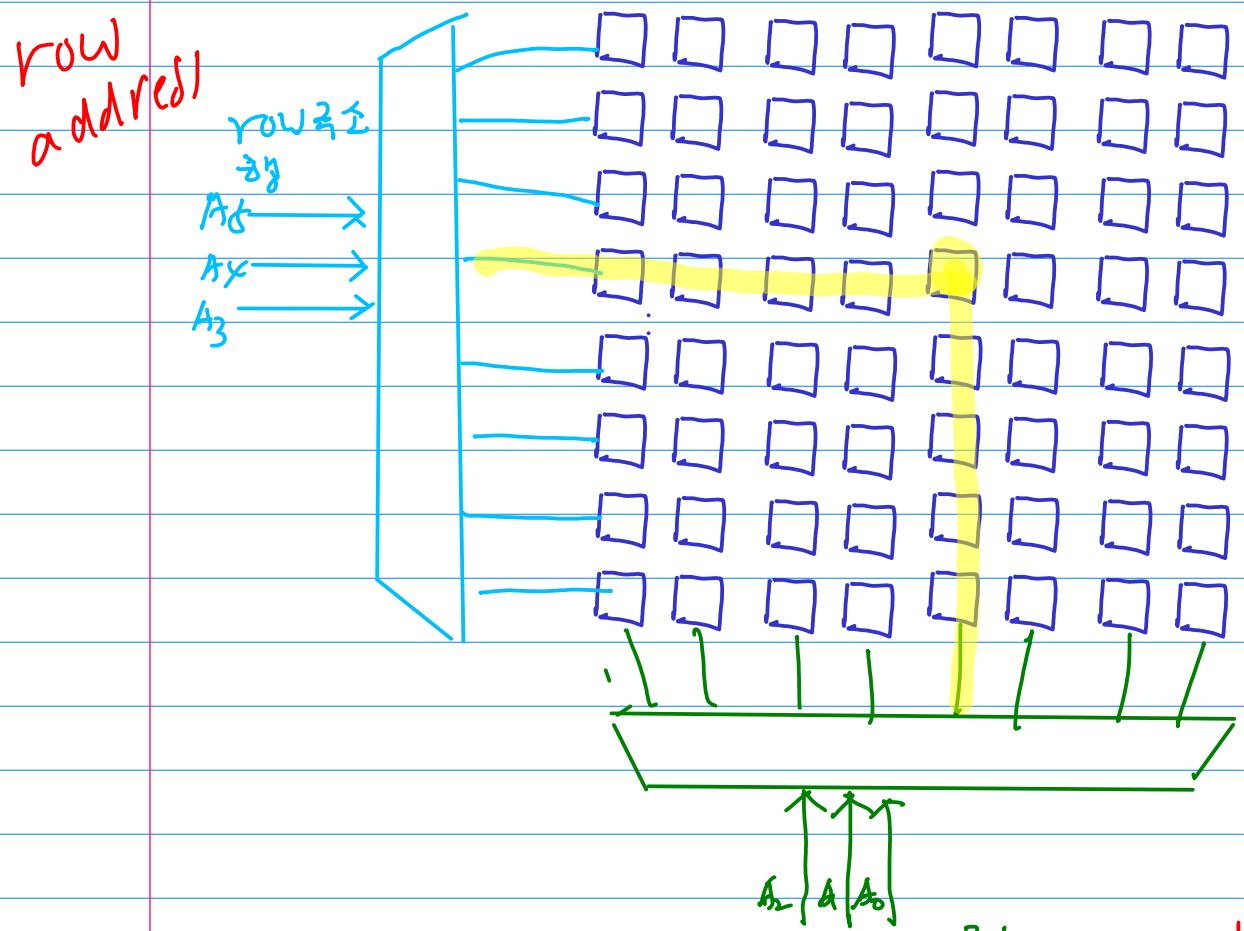
3x8 decoder



3x8 decoder



Package



총 용량이 8K bit 8K x 1-bit

$2^3 \cdot 2^{10}$

2^{13}

address: 13-bit data = 1-bit

용량이 1 Mbit 1M x 1-bit

2^{20}

address: 20 data : 1-bit

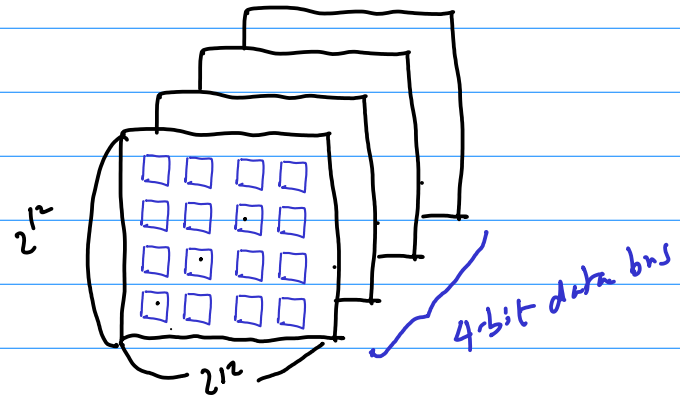
용량이 1 Gbit 1 G x 1-bit

2^{30}

address: 30-bit data: 1-bit

④ 16M x 4 bit

$16M = 2^4 \cdot 2^{20} = 2^{24} = (2^{12})^2$



conceptual

전체 용량 64Mbit $2^6 \cdot 2^{20} = 2^{26}$

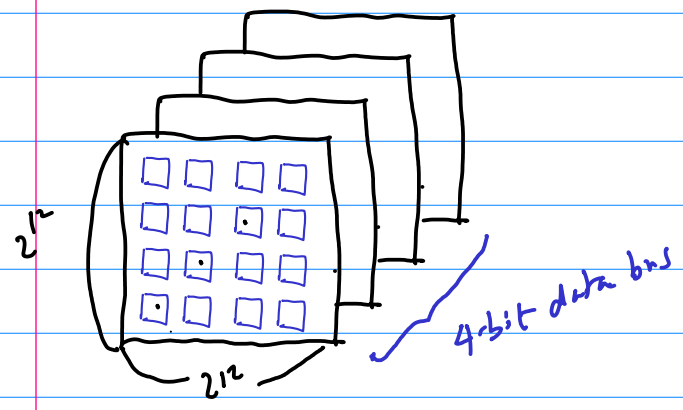
$2^{12} = 2^2 \cdot 2^{10} = 4 \times 1024 = 4096$

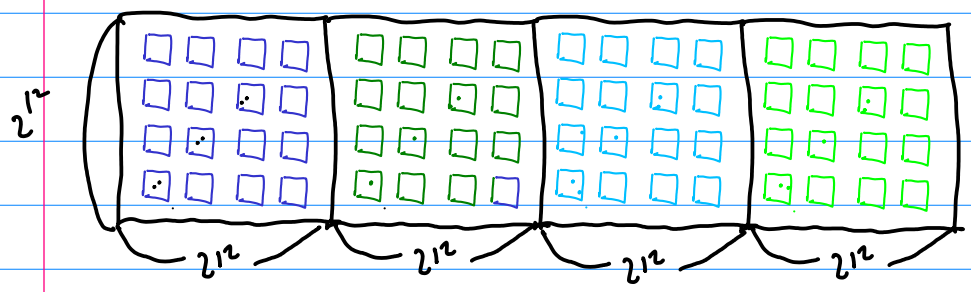
$2^{12} \cdot 2^{12} \cdot 2^2$

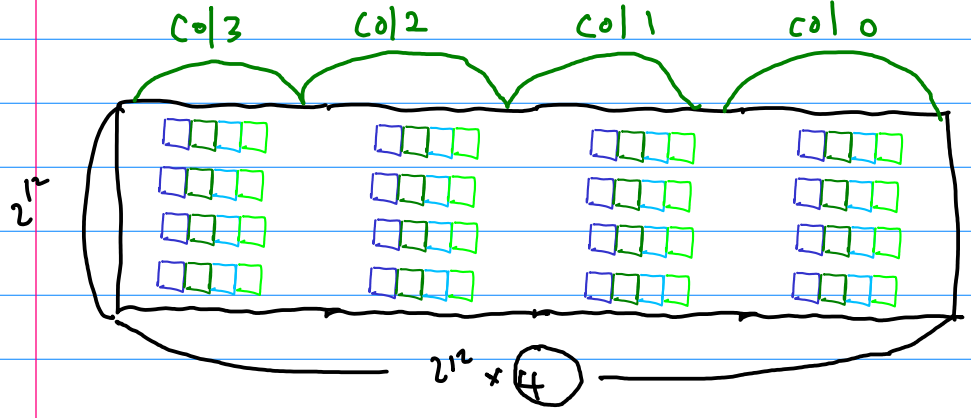
4096 x 4096 x 4-bit

12-bit row address | 12-bit col address

24-bit address

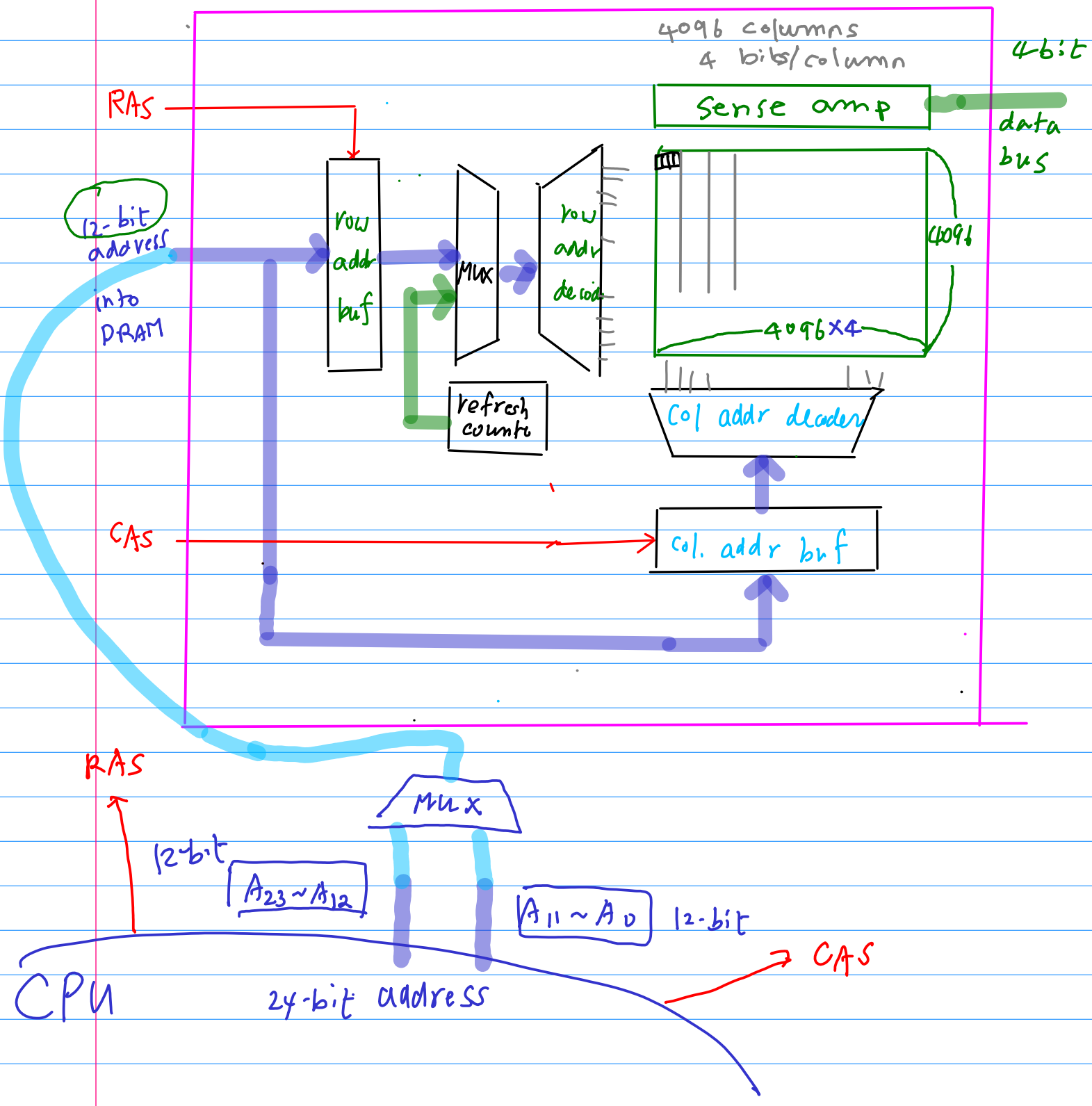

2^{12}
2^{12}
4-bit data bus

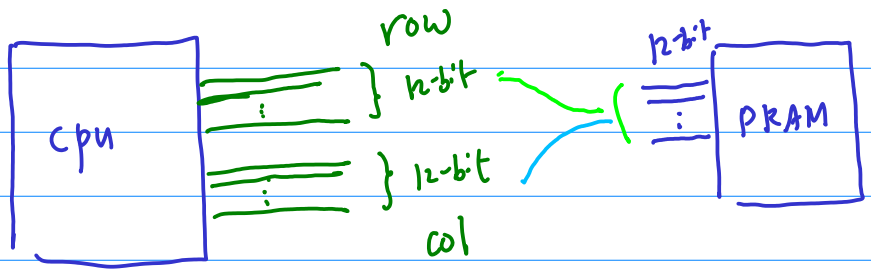

2^{12}
2^{12}
2^{12}
2^{12}
2^{12}


col 3
col 2
col 1
col 0
2^{12}
$2^{12} \times 4$

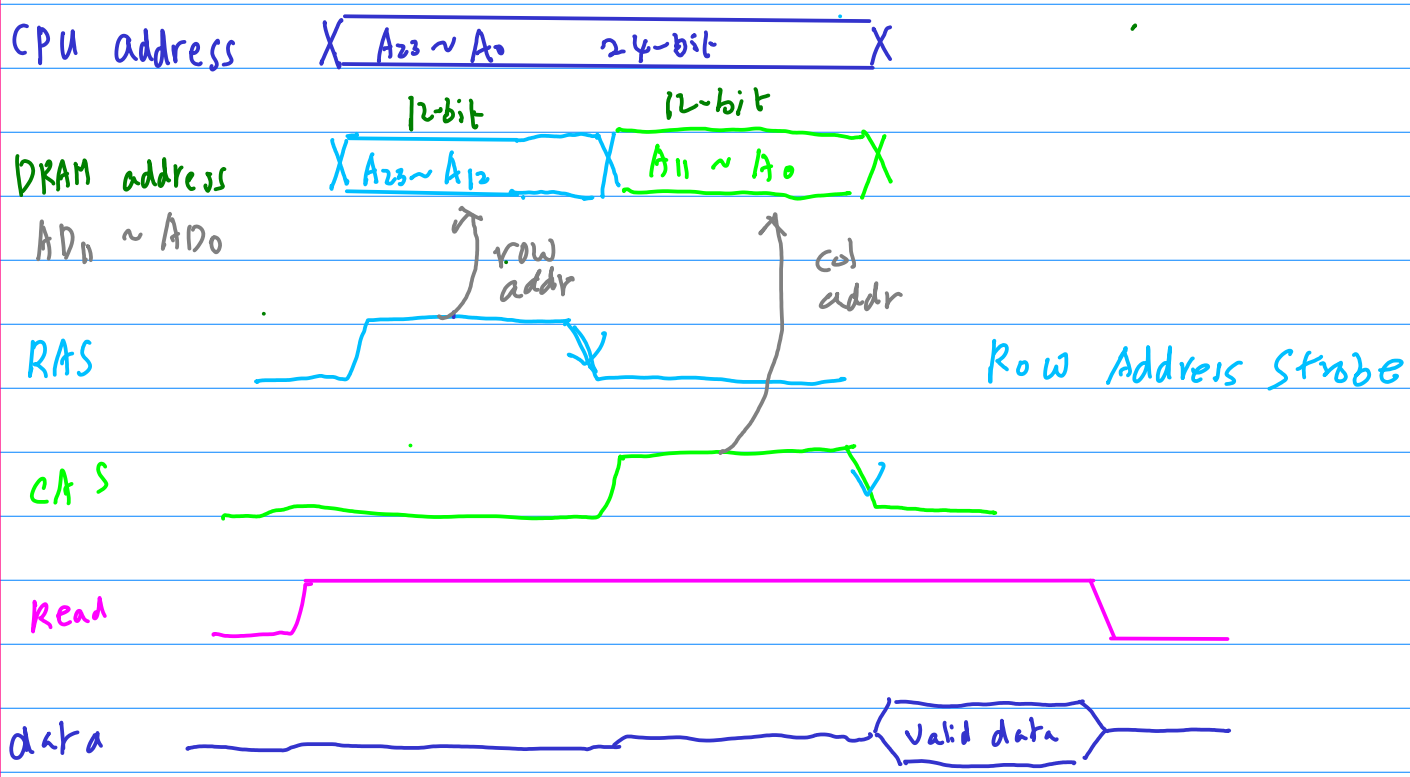
VLSI Layout

64Mbit (16M x 4bit) DRAM 칩





timing diagram



CPU address는 CPU가 내보내는 (출력)하는
총 24-bit address를 의미한다

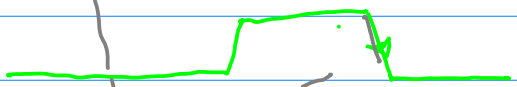
A_{23} ..... A_{12} A_{11} ..... A_0

Row Address Col address

시간적으로 먼저 선택이 되어서
DRAM chip의 address를 결정한다

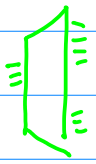
RAS (Row Address strobe)

다음에 하위 12bit의 선택이 되어서
DRAM chip의 address를 결정한다



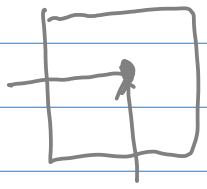
DRAM의 내부에 Address Buffer Register에
latch
Add Buffer Reg에 latch된 주소들은 각각

row address decoder

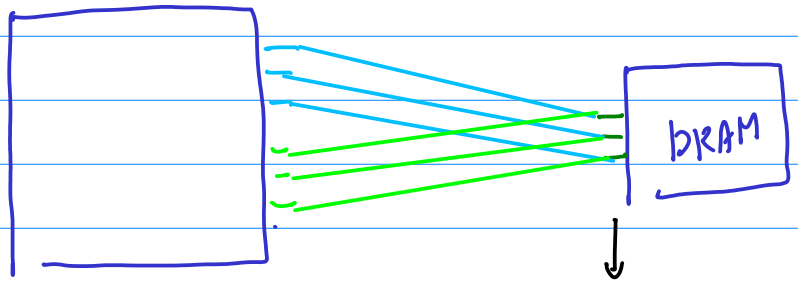


col address decoder

에 의해서 해당 기억장소를 선택한다



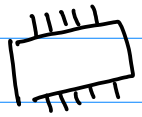
시간적으로 multiplexing

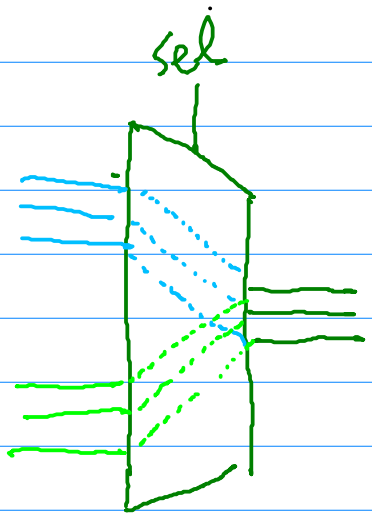


24 bit → 12-bit

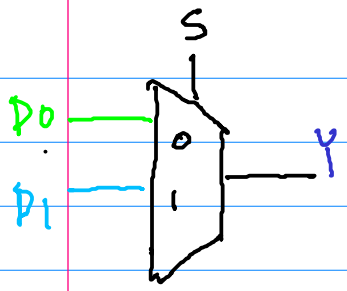
RAS & CAS

chipel pin 수를 줄임





Multiplexor



S	D1	D0	Y	
0	0	0	0	D0
0	0	1	1	D0
0	1	0	0	D0
0	1	1	1	D0
1	0	0	0	D1
1	0	1	0	D1
1	1	0	1	D1
1	1	1	1	D1

S	Y
0	D0
1	D1

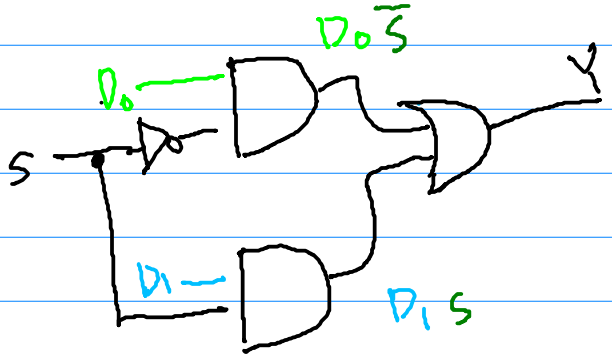
$Y = D_0\bar{S} + D_1 S$

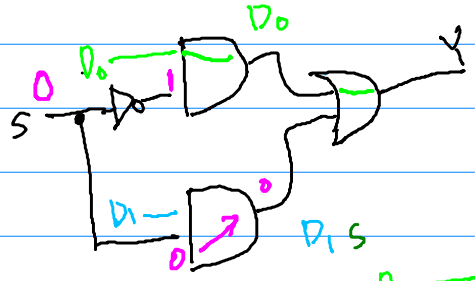
$\bar{S}=1$ 일때 D_0 이기나

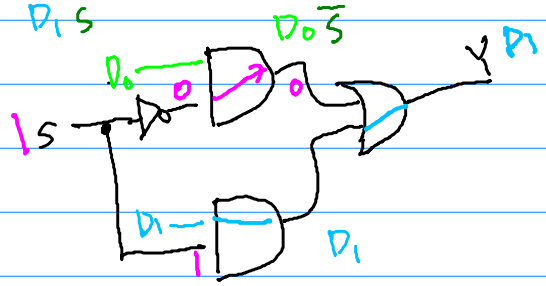
$S=1$ 일때 D_1

$S=0$ 일때 D_0 이기나

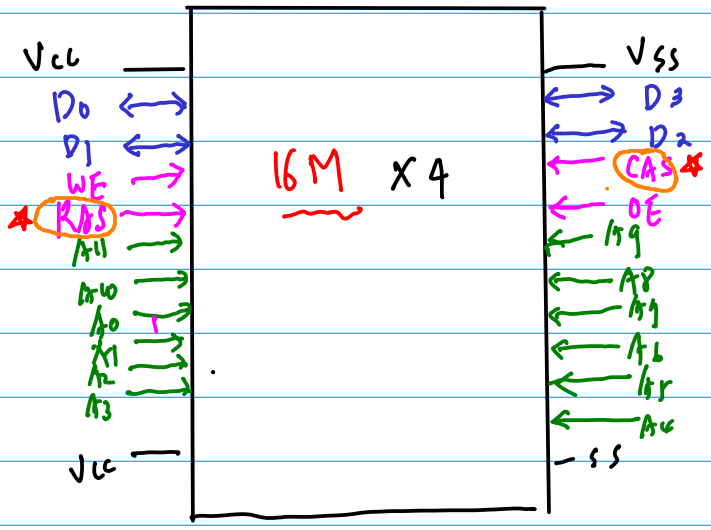
$S=1$ 일때 D_1







DRAM Chip



$(2^{12})(2^{12})$

$= 2^{24}$

$= 2^4 \cdot 2^{20}$

$= 16 \cdot M$

data bus	D3 ~ D0	4-bit
addr bus	A11 ~ A0	12-bit
control bus	WE, OE, CAS, RAS	

DRAM

|

* clock X

SDRAM

- D: Dynamic
- S: Synchronous 동기식
  - Clock 신호에 맞추어서 data를 전송한다

원래 DRAM은
필요한 data와 control signal 들어오는 즉시
access 동작 (RD/WR) 수행하여 응답한다

(쓰기동작 : 전송된 data를 즉시 해당 기억장소에 저장
읽기동작 : 인출동작을 수행한후 즉시 data 버스로 전송

DRAM에서는 이런 동작들이 연속적으로 발생
→ CPU가 wait
→ Bus를 다른 장치 가 사용 X
→ 성능이 안좋다.

SDRAM 액세스동작이 clock에 동기되어서 수행된다.

① CPU가 1 clock cycle 동안
address와 읽기 제어 신호를 Bus로 보낸다.
CPU는 결과를 기다리지 않고 다른 연산을 수행하게 된다

:

② SDRAM이 주소와 읽기 신호를 받은 즉시
access 동작을 수행한다
SDRAM System Bus 사용권리 획득한후
(clock cycle 동안 data를 Bus로 전송한다

③ CPU가 Bus에 있는 data를 읽는다.

SDRAM

parallel access (pipeline access) ← 여러개의 **bank** 사용

bank 마다 다른 주소에 대한 access가 동시에 수행됨

512 Mbit 용량 SDRAM

4개의 bank 사용

512 M

$2^9\ 2^{20}$

$2^{29} \times 2^2$ (bank)

$2^{24} \times 2^3$

$2^4 \times 2^{20} \times 2^3$

16M x 8-bit 1 bank의 구성.

data bus의 width

16M x 8-bit x 4 bank

= 16M x 4 bank x 8-bit = 64M x 8-bit

= $2^4 \cdot 2^{20} \cdot 2^2$ x 8-bit = $2^6 \cdot 2^{20}$ x 8-bit

2^{26} x 8-bit

즉 address 26-bit

64M x4-bit $2^6 \cdot 2^{20} \cdot 2^2 = 2^{28}$

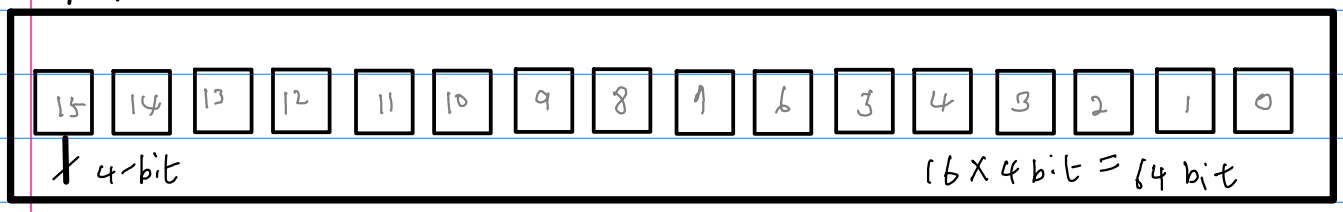
32M x8-bit $2^5 \cdot 2^{20} \cdot 2^3 = 2^{28}$

16M x16-bit $2^4 \cdot 2^{20} \cdot 2^3 = 2^{28}$

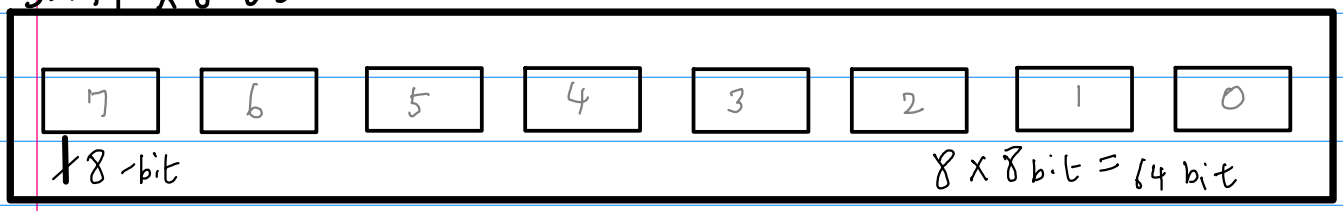
2^8 M bit = 256 M bit memory

	64M x4-bit	32M x8-bit	16M x16-bit
configuration	16M x 4 x 4 bnk	8M x8 x 4 bnk	4M x16x 4 bnk
	(4+20+2) bit	(3+20+2) bit	(2+20+2) bit
row addr	8K (13-bit)	8K (13-bit)	8K (13-bit)
bnk addr	4 (2-bit)	4 (2-bit)	4 (2-bit)
col addr	2K (11-bit)	1K (10-bit)	512 (9-bit)
	26-bit	25-bit	24-bit

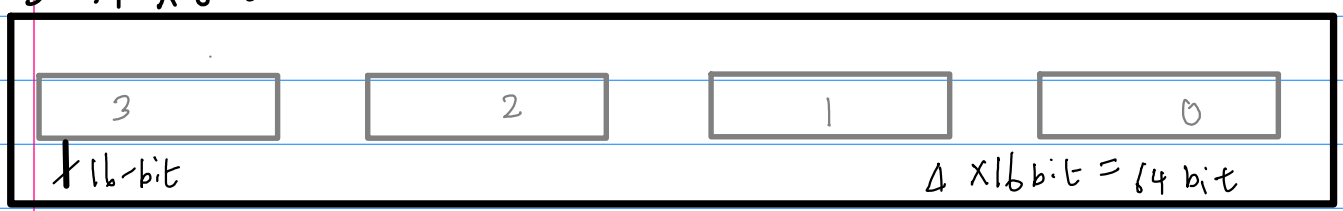
64M x4-bit



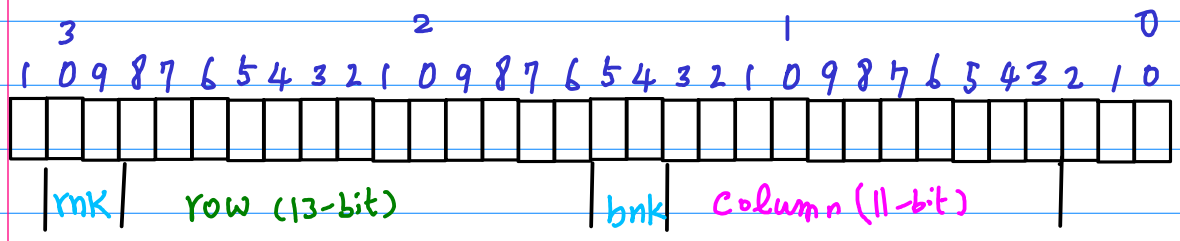
32M x8-bit



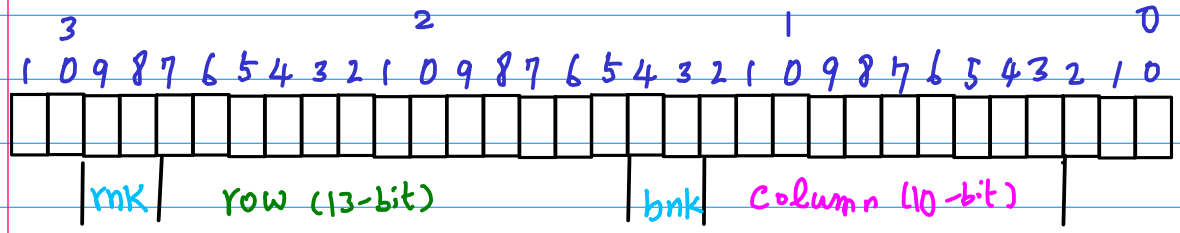
32M x8-bit



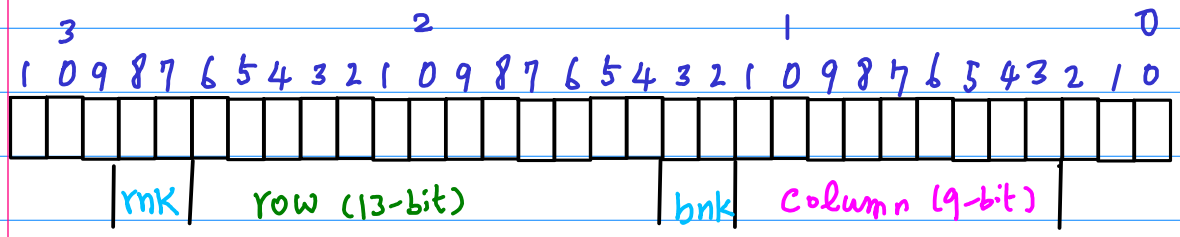
	64M x4-bit	32M x8-bit	16M x16-bit
configuration	16M x 4 x 4 bnk	8M x8 x 4 bnk	4M x16x 4 bnk
	(4+20+2) bit	(3+20+2) bit	(2+20+2) bit
row addr	8K (13-bit)	8K (13-bit)	8K (13-bit)
bnk addr	4 (2-bit)	4 (2-bit)	4 (2-bit)
col addr	2K (11-bit)	1K (10-bit)	512 (9-bit)
	26-bit	25-bit	24-bit



26-bit



25-bit



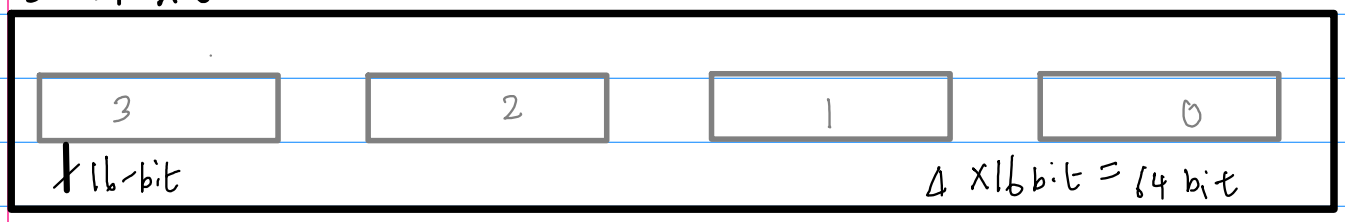
24-bit

64-bit = 8 bytes

3-bit address

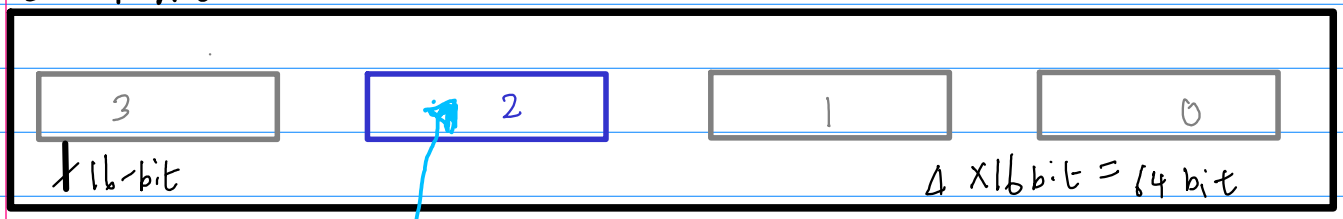
rank0

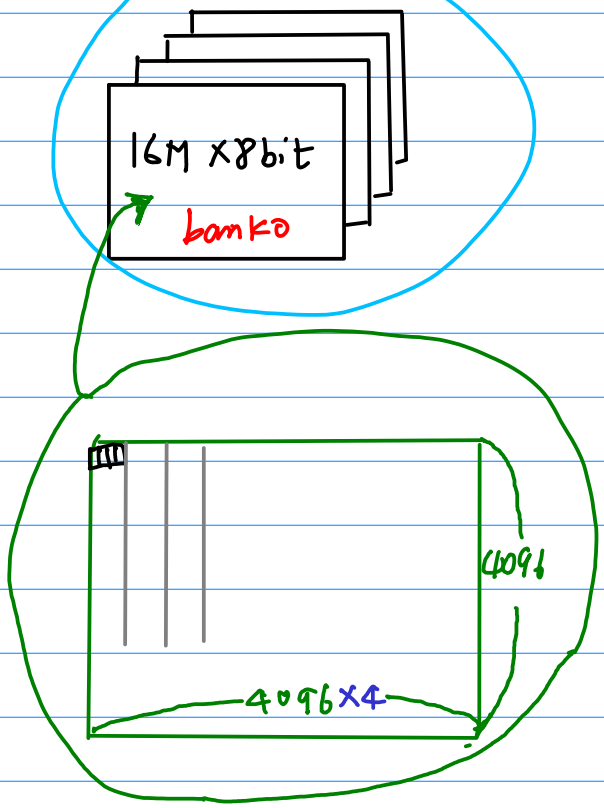
32 M x 8-bit



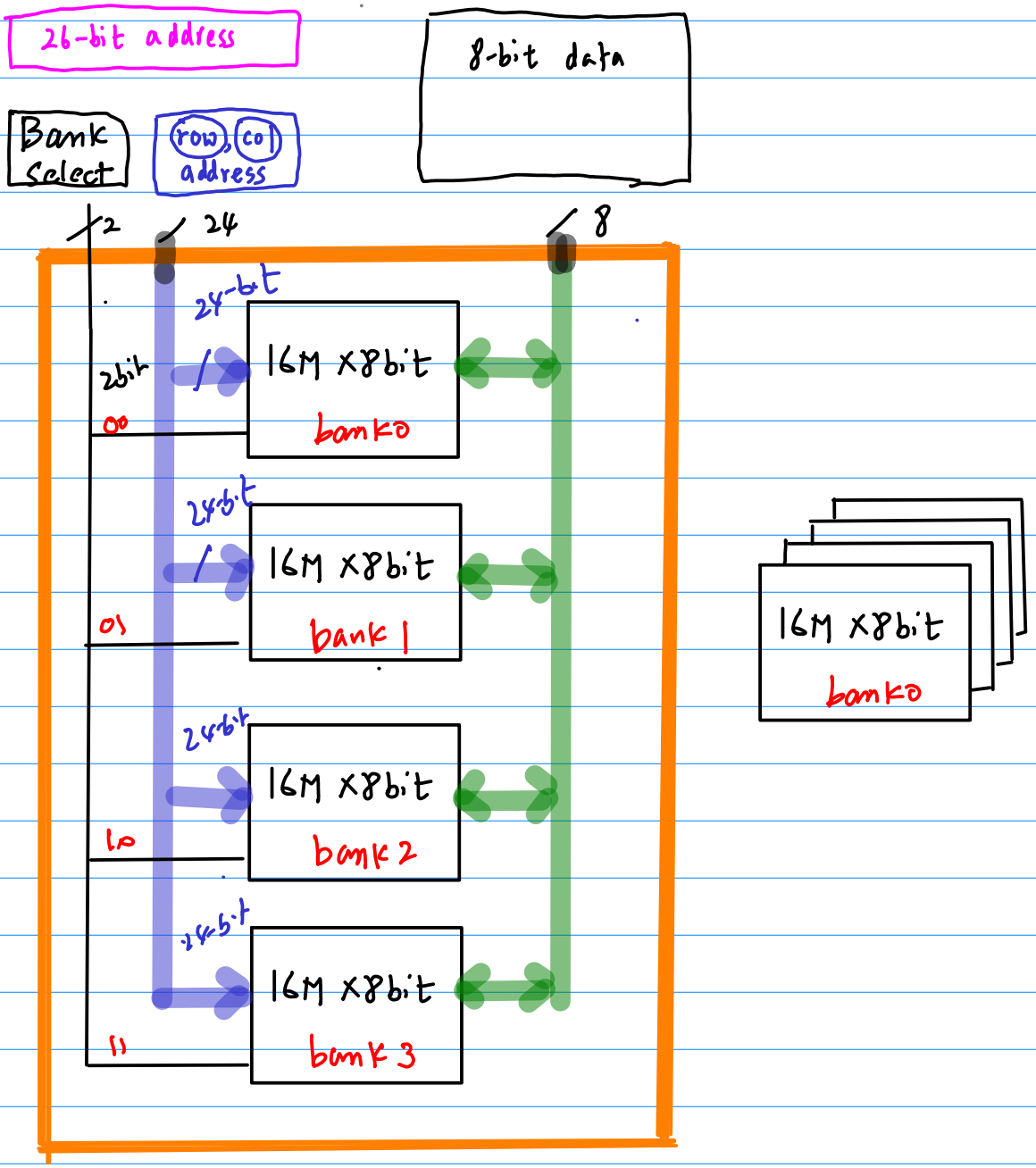
rank1

32 M x 8-bit





SDRAM



24bit address는
동시에 4개의
bank에
병렬로 입력된다

4개의 bank들이
1개의 data bus를
시간적으로 공유한다.

한번에 1개의 bank만
data bus를 사용가능

bank address positions

① 26-bit address

24-bit | 2-bit

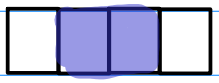
not used

② 26-bit address

row | 2-bit | col

OK

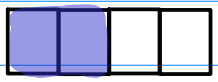
bank interleaving



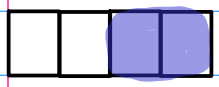
③ 26-bit address

2-bit | 24-bit

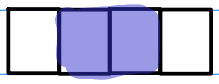
OK



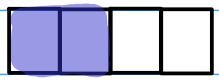
X



OK



OK



address	bank
0	b0
1	b1
2	b2
3	b3
4	b0
5	b1
6	b2
7	b3
8	b0
9	b1
10	b2
11	b3
12	b0
13	b1
14	b2
15	b3

address	bank
0	b0
1	b0
2	b1
3	b1
4	b2
5	b2
6	b3
7	b3
8	b0
9	b0
10	b1
11	b1
12	b2
13	b2
14	b3
15	b3

address	bank
0	b0
1	b0
2	b0
3	b0
4	b1
5	b1
6	b1
7	b1
8	b2
9	b2
10	b2
11	b2
12	b3
13	b3
14	b3
15	b3

b0	b1	b2	b3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

fragmentation

b0	b1	b2	b3
0	2	4	6
1	3	5	7
8	10	12	14
9	11	13	15

b0	b1	b2	b3
0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

bank interleaving

Bank Architecture

24-bit address

16M x8bit

bank

$16M = 2^4 \cdot 2^{20} = 2^{24}$

$2^{24} = 2^{13} \cdot 2^{11}$

row: $2^3 \cdot 2^{10} = 8K$ (13-bit)

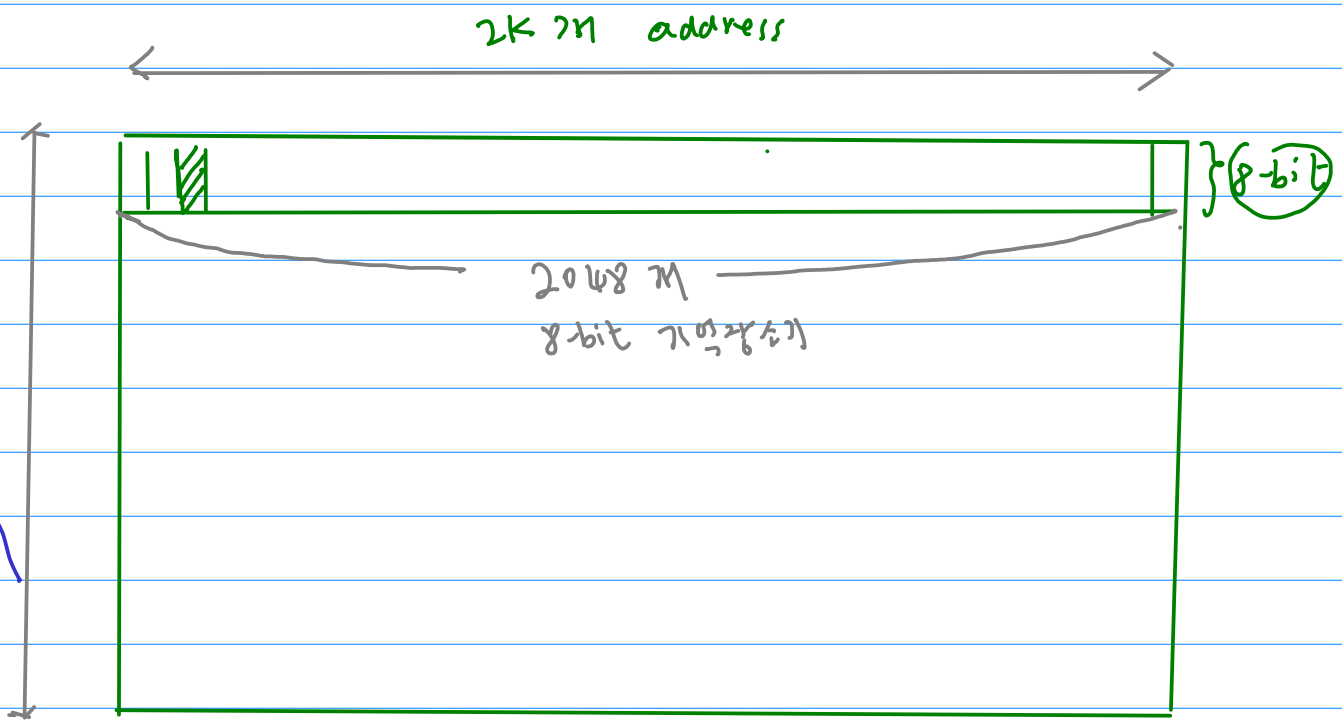
col: $2^1 \cdot 2^{10} = 2K$ (11-bit)

13-bit + 11-bit + 2 = 24-bit + 2 = 26-bit

16Kbit = 2K x8-bit linear memory array

low 8

$8 = 2^3$



row 13-bit

8K

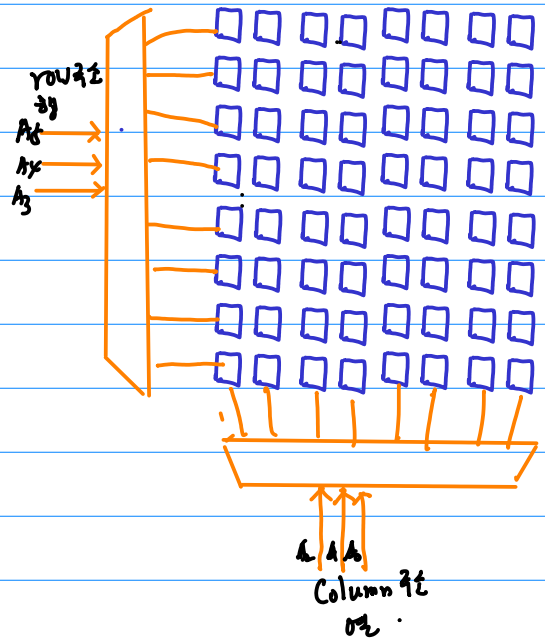
$= 2^{13}$

$= 8192$개

SDRAM

DRAM과 같이 (Row) & (Col) Address Decoder 사용

Row & Col address 사용



→ 1-bit data access 가능

8-bit data access를 위해서는 8개 Copy가 필요

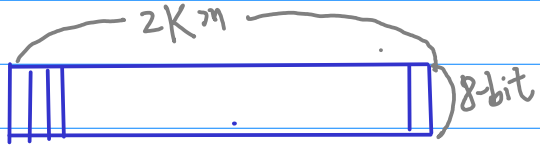
$2^{24} = 2^{13} \cdot 2^{11}$

row (13-bit) $2^3 \cdot 2^{10} = 8K$

col (11-bit) $2^1 \cdot 2^{10} = 2K$

- 총 8K개의 row가 있다.
- 각 row에는 16K bit가 저장된다.

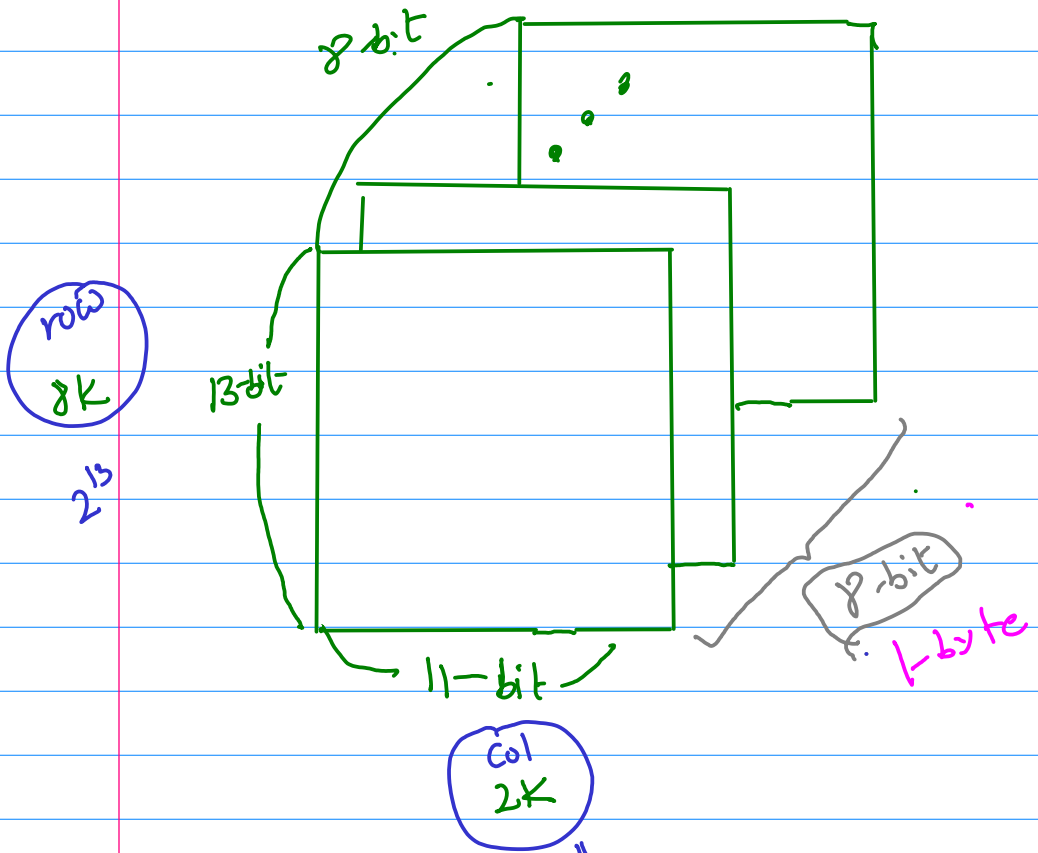
16Kbit = 2K x 8-bit



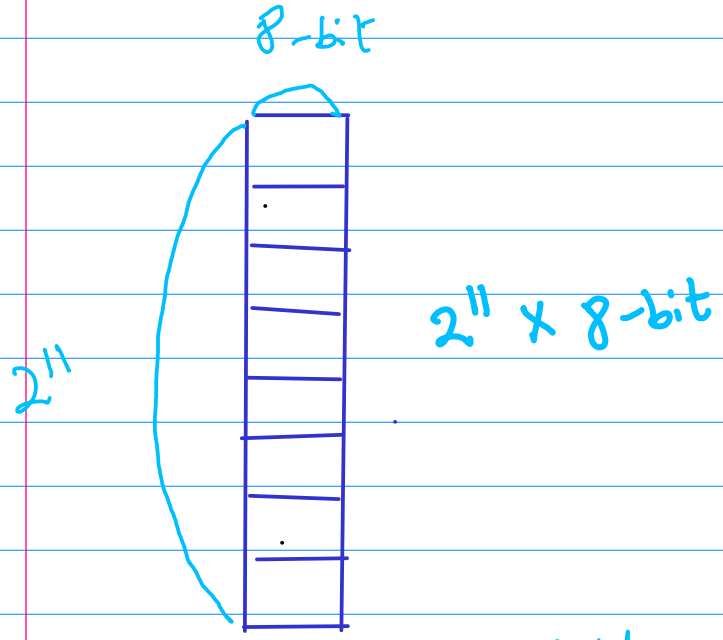
data bus가 8-bit 이므로

16K bit / 8-bit = 2K개 가 있다.

개념적인 그림

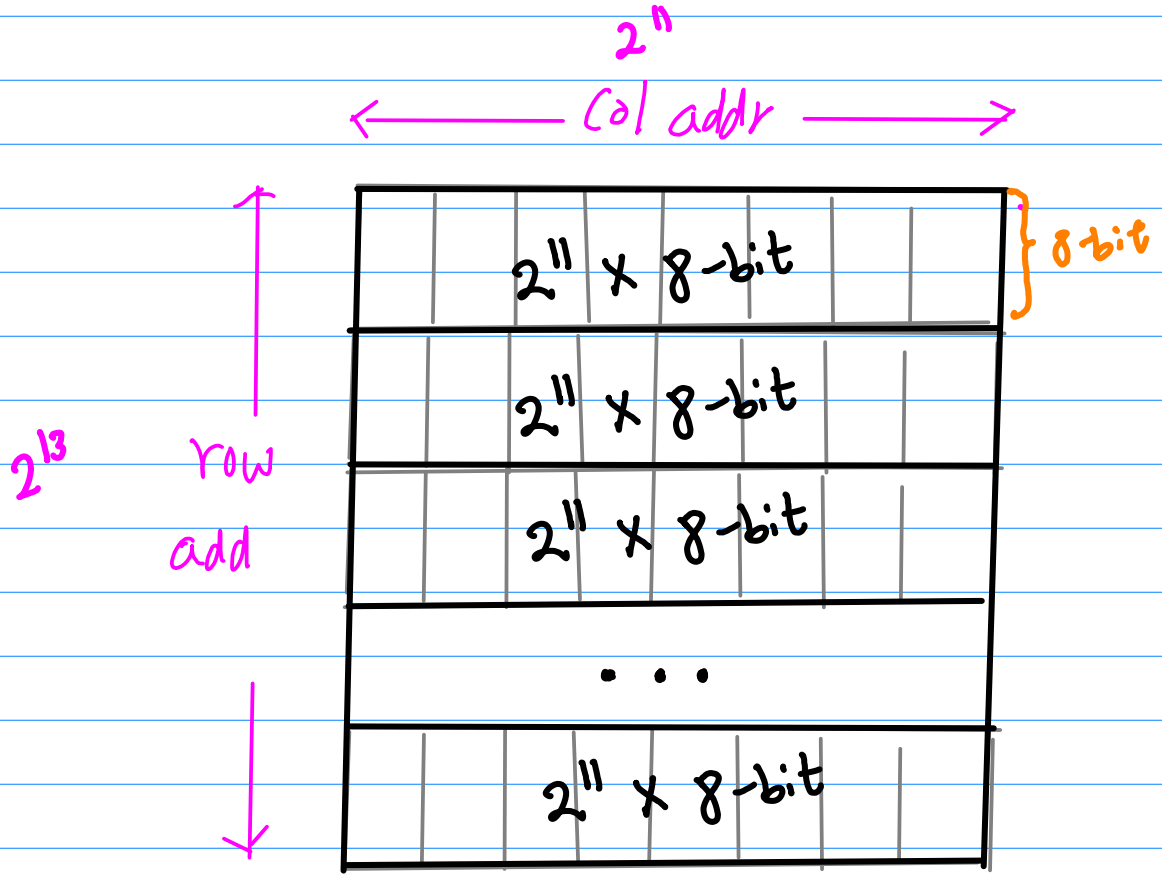


linear array



$2^{24} = 13+11$

address bus 24-bit
data bus 8-bit



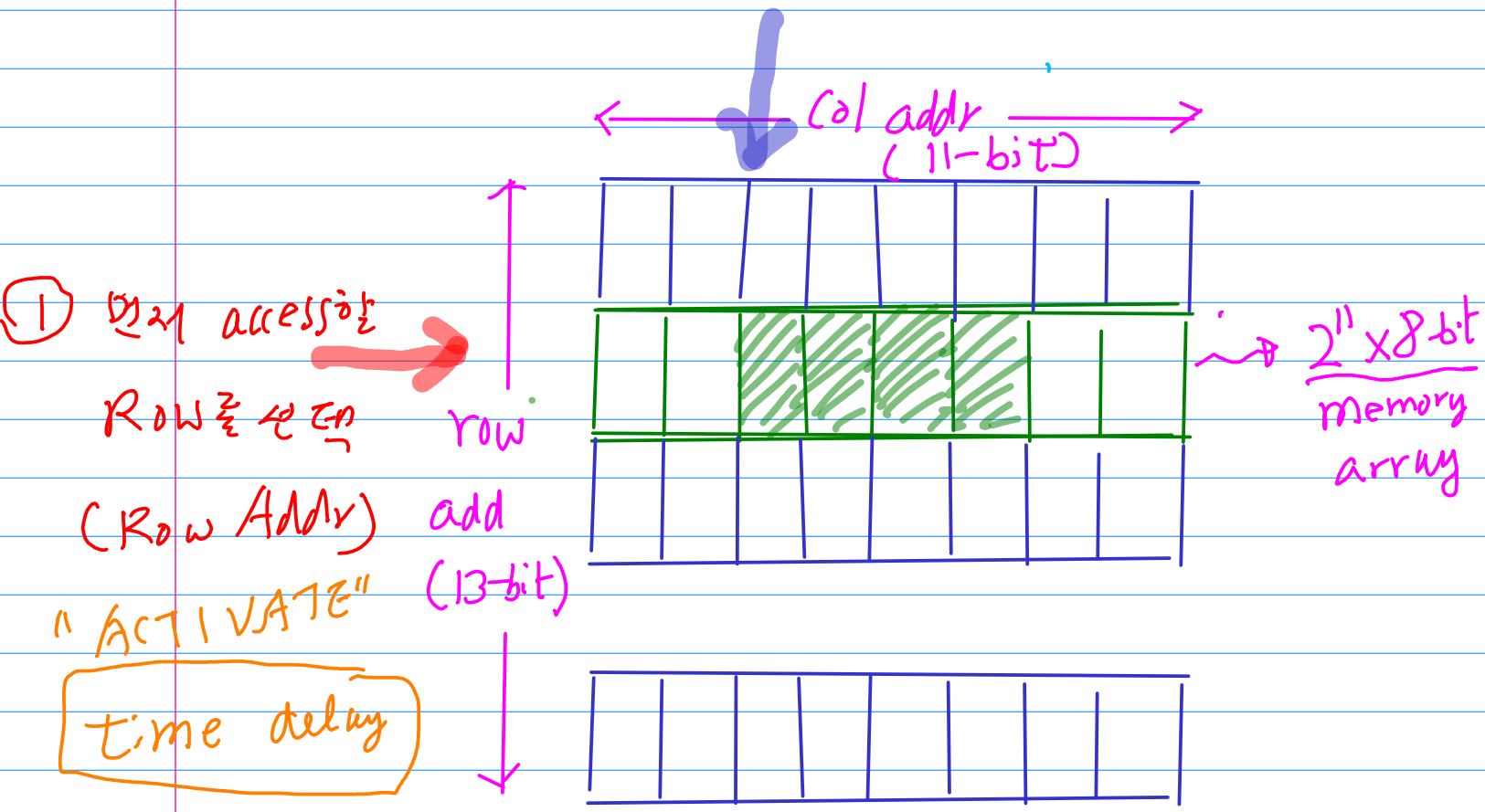
16M x 8-bit

$= 2^4 \cdot 2^{20}$

$= 2^{24} = 2^{13} \cdot 2^{11}$

② Col Address 를 선택한다.

③ burst (sequential address)



① 원하는 access할 Row를 선택 (Row Addr) "ACTIVATE"

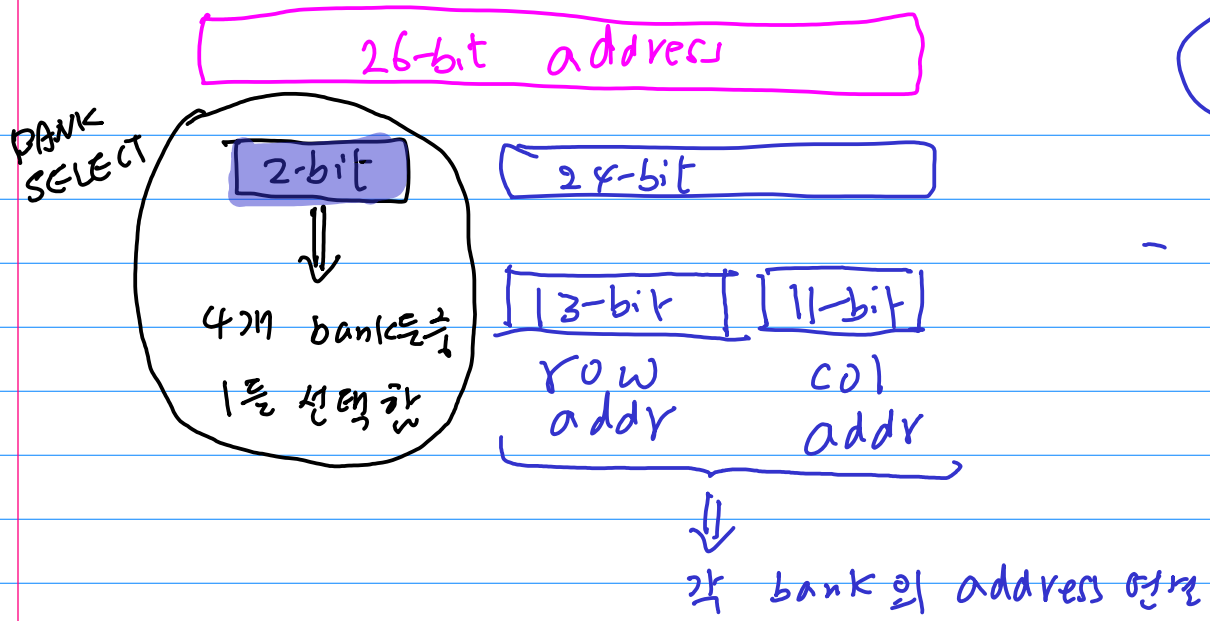
time delay

Bank Interleaving

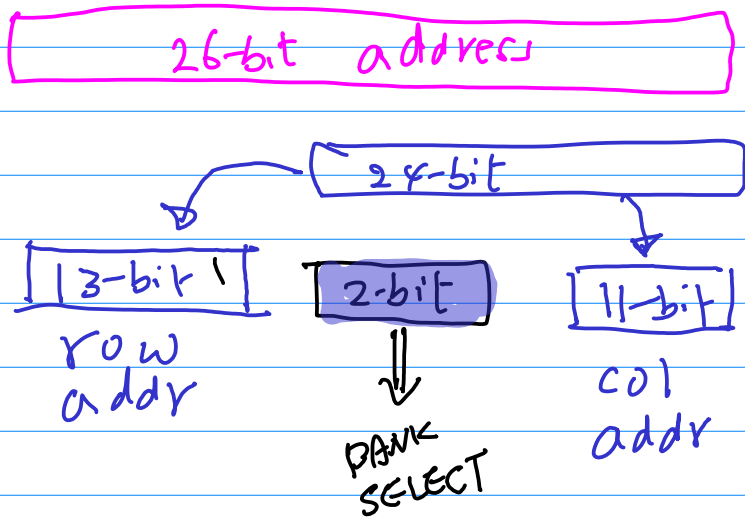
allows operations to 2 ~ 4 banks simultaneously

independent memory arrays
inside of a DRAM chip

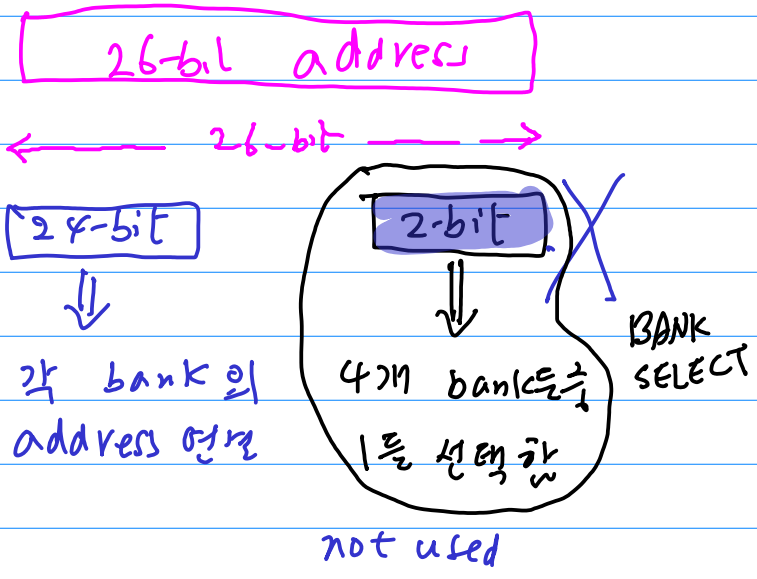
① OK



② OK



③ X not used



①

26-bit address

BANK SELECT

2-bit

⇓

4개 bank들 중 1를 선택함

24-bit

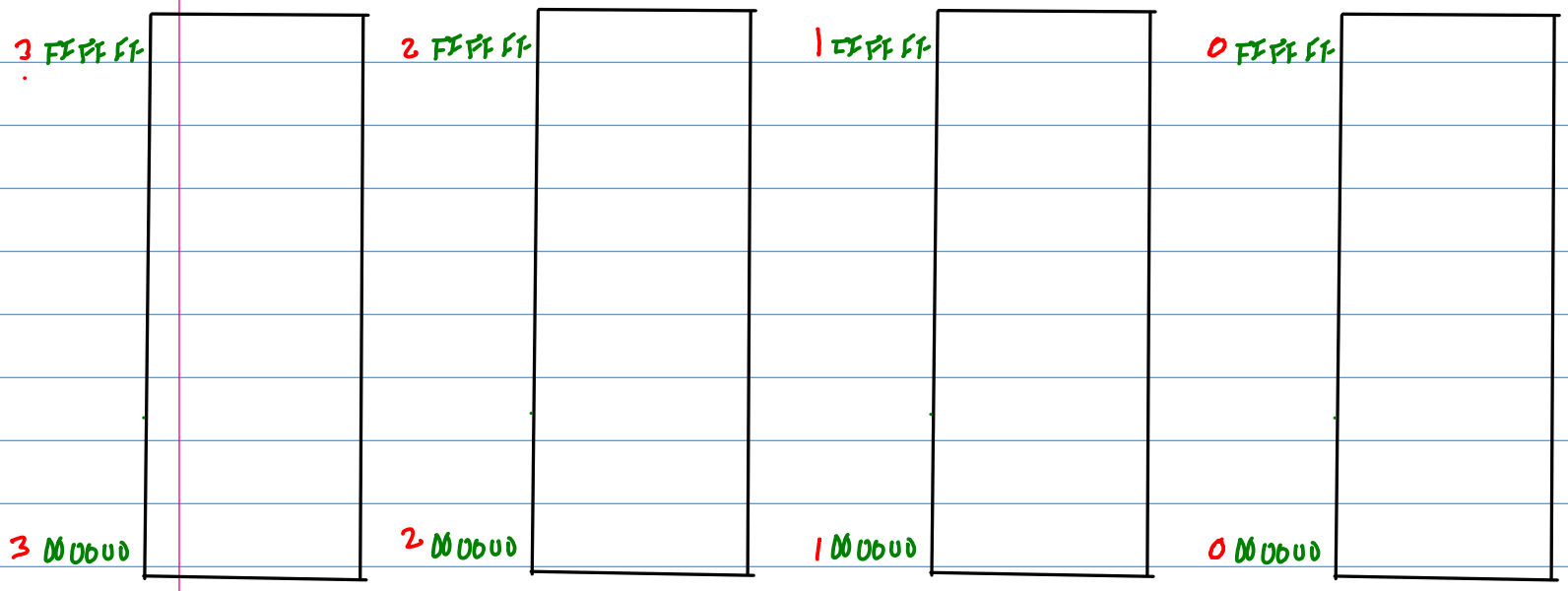
13-bit | 11-bit

row addr | col addr

⇓

각 bank의 address 연결

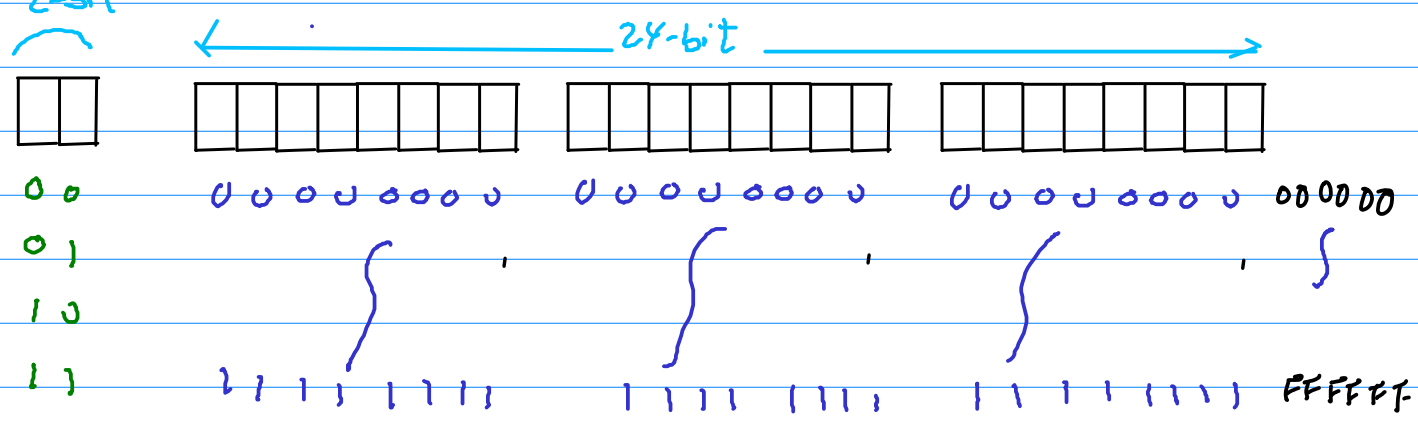
16M x8bit bank3 | 16M x8bit bank2 | 16M x8bit bank1 | 16M x8bit bank0



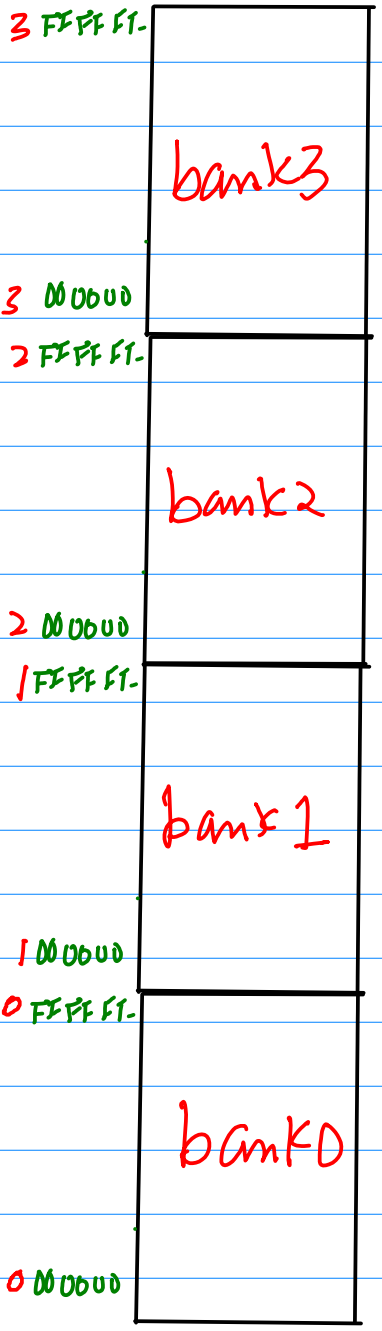
bank 수 2-bit

26-bit address bus

= 3x8 + 2



$2^{26} = 2^6 \cdot 2^{20} = 16M$

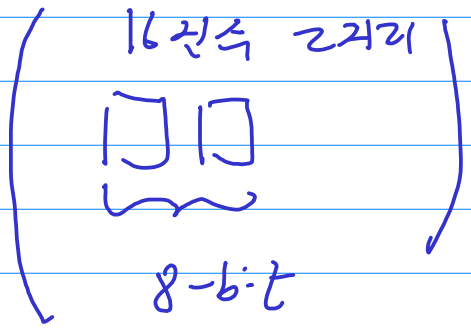


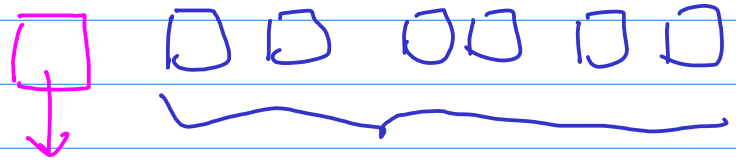
26-bit

2 + 24

bank | row + col

(13) · (11)





24-bit

0
1
2
3
bank

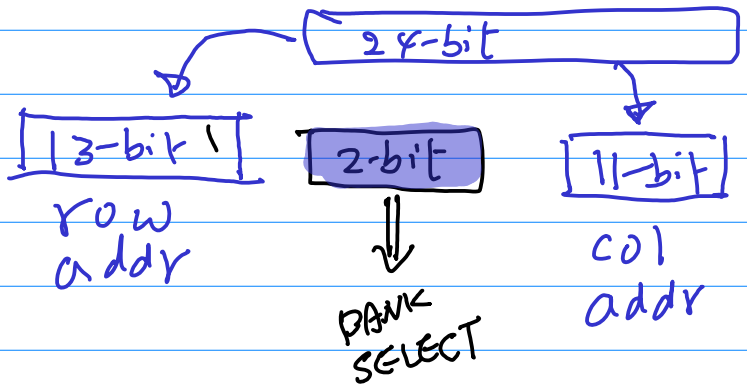
FF FF FF

00 00 00

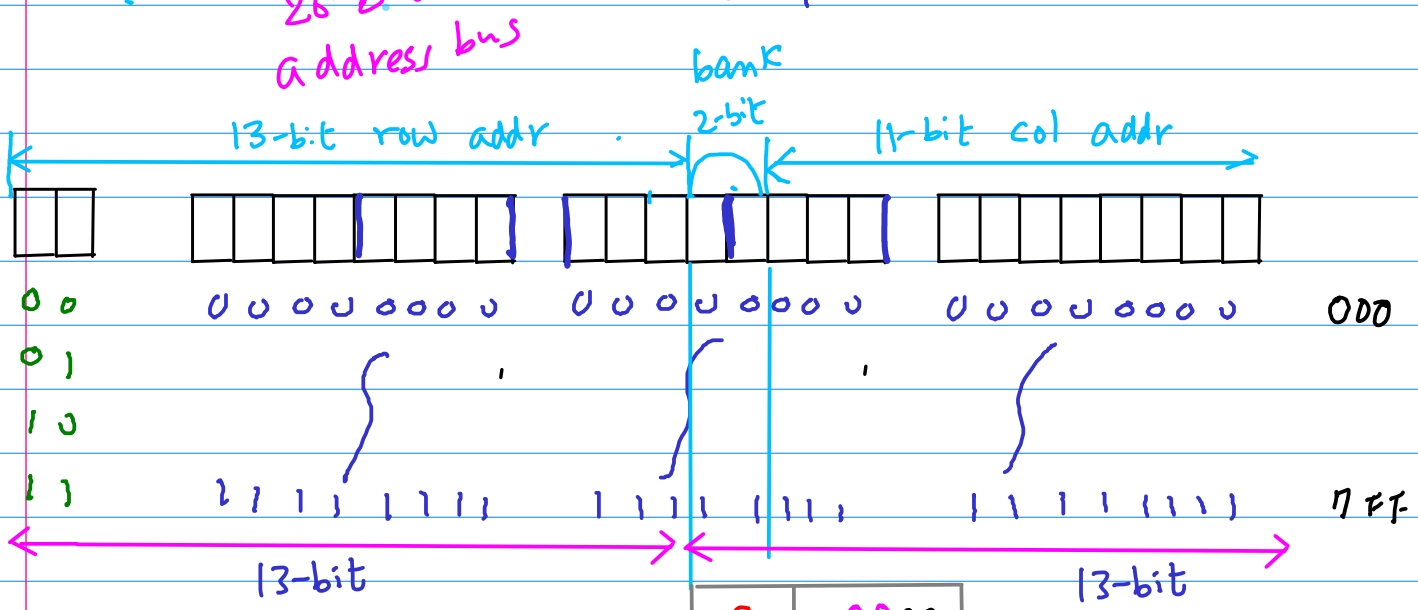
2

26-bit address

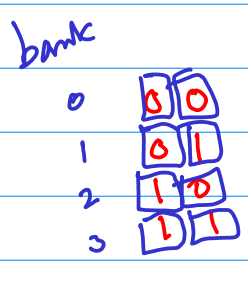
OK

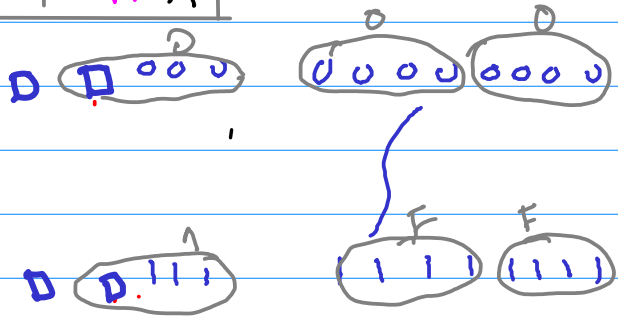


= 3 x 8 + 2



0	0000
	07FF
1	0800
	0FFF
2	1000
	17FF
3	1800
	1FFF

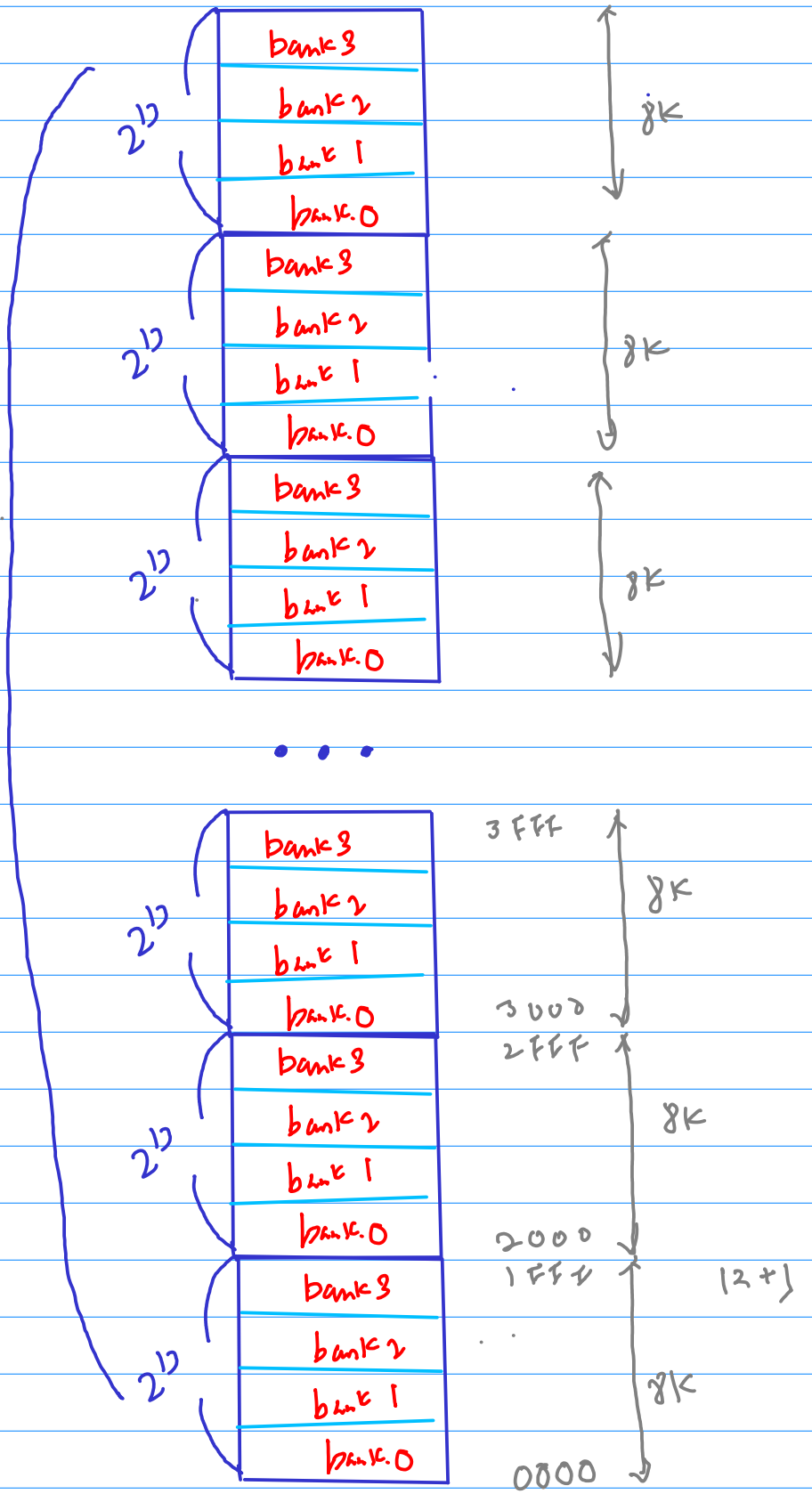




26-bit address

$2^{26} = 2^6 \cdot 2^{20} = 64\,M$

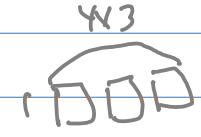
$2^{13} = 2^3 \cdot 2^{10} = 8\,K$



$\frac{2^{13} \cdot 2^{13}}{2^{13+13}} = 2^{26}$

$8K \cdot 8K = 64\,M = 2^6 2^{20}$

12+1



3

26-bit address

← 26-bit →

24-bit

⇓

각 bank의 address 연결

2-bit

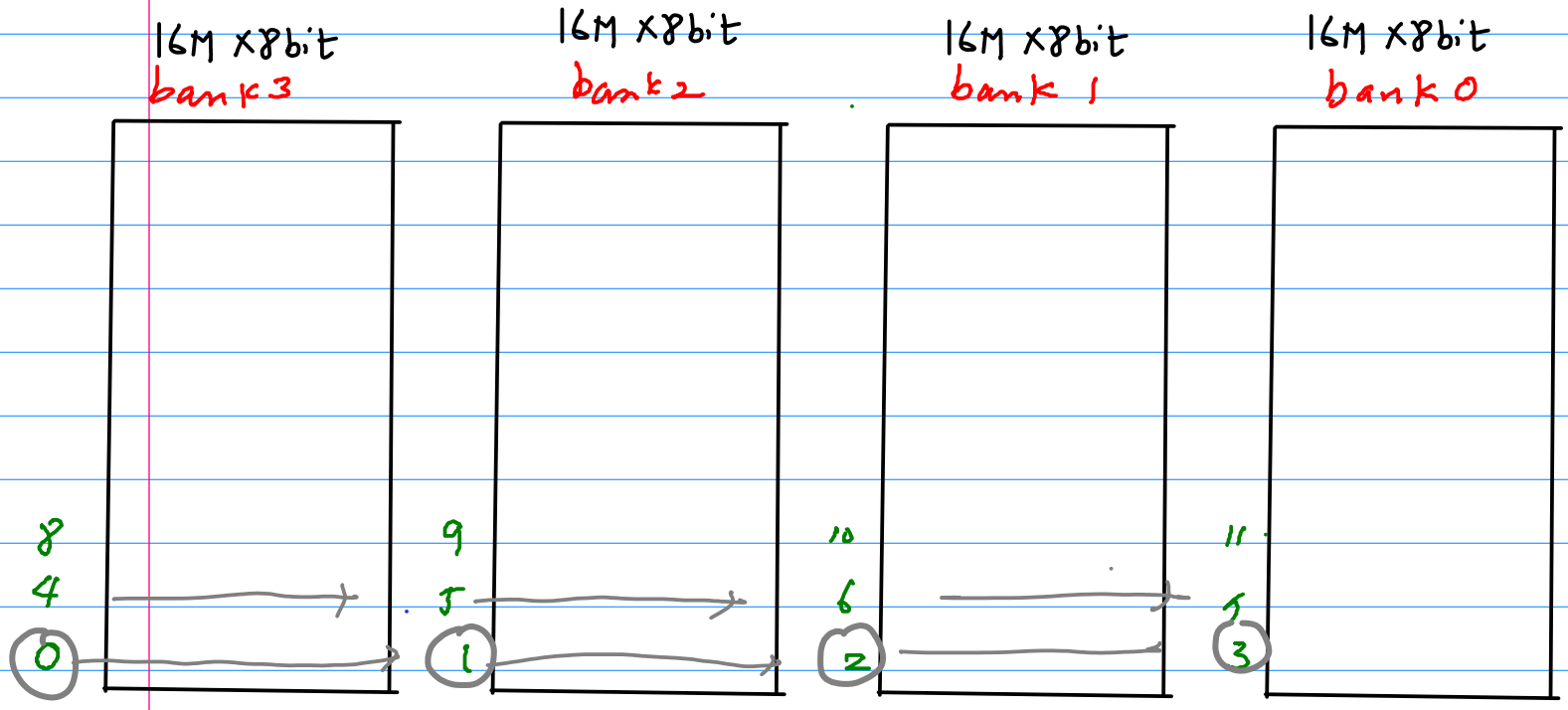
⇓

4개 bank들 중 1을 선택함

BANK SELECT

not used

X not used



26-bit address bus

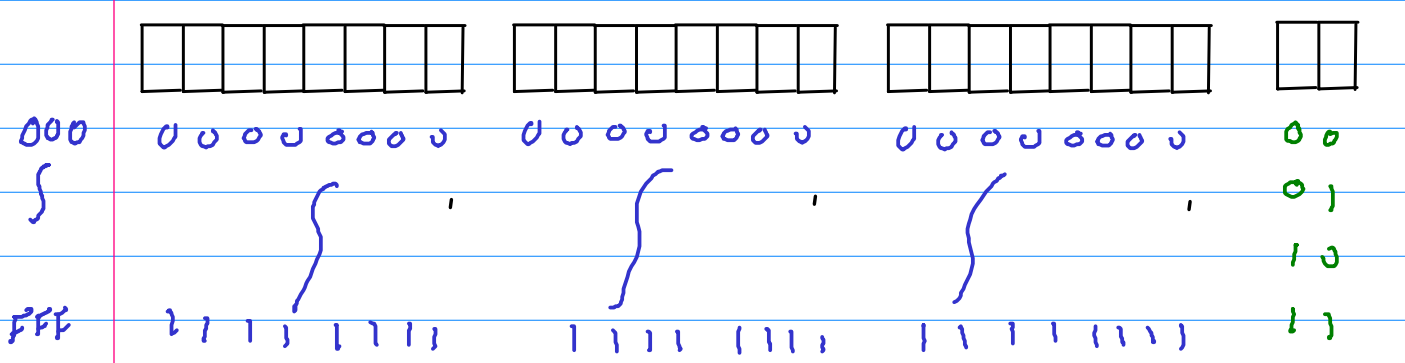
= 3 x 8 + 2

bank

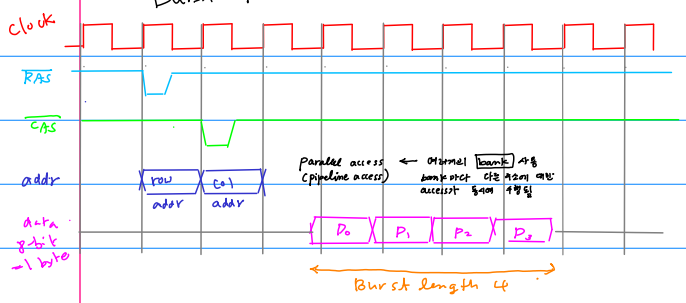
⇑

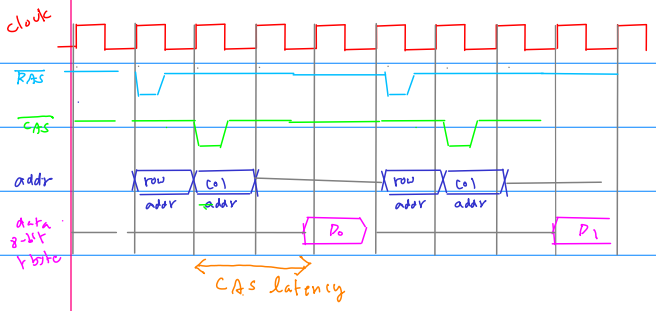
2-bit

← 24-bit →



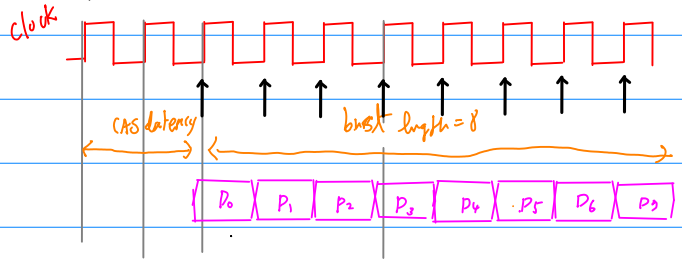
Burst Read Mode





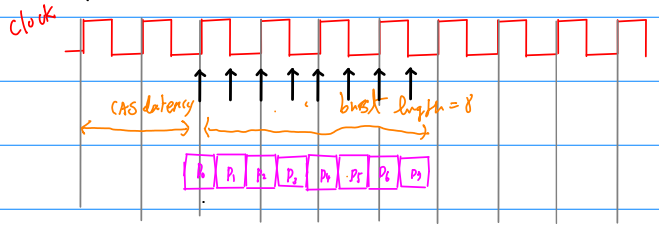
SDRAM

7.5ns. (133MHz)



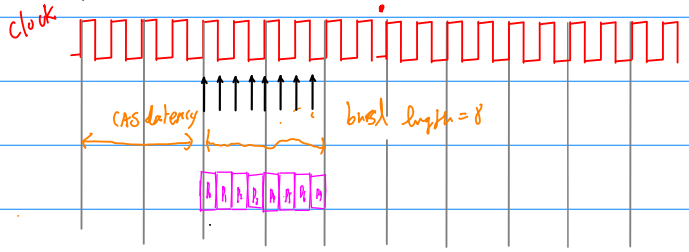
DDR

7.5ns. (133MHz)

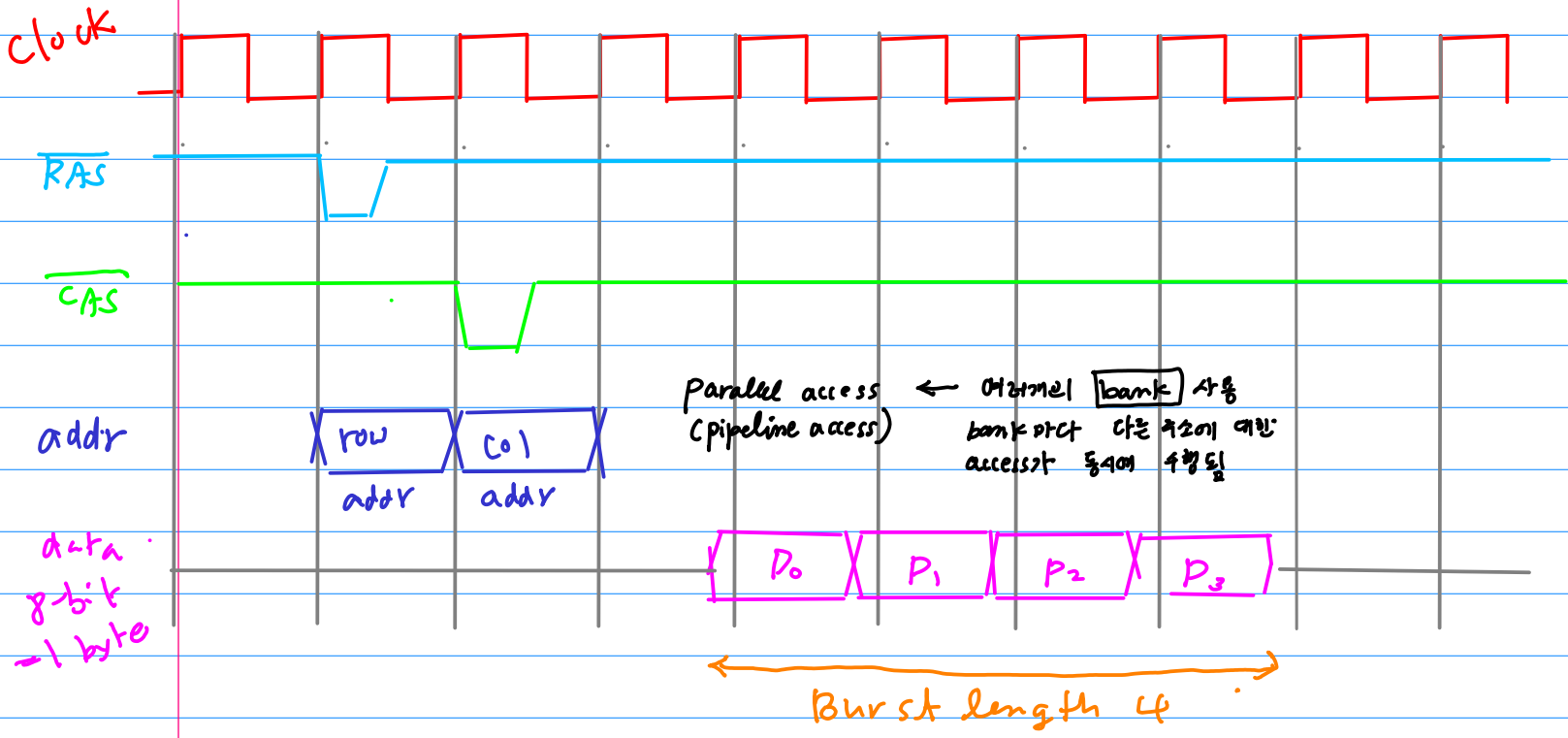


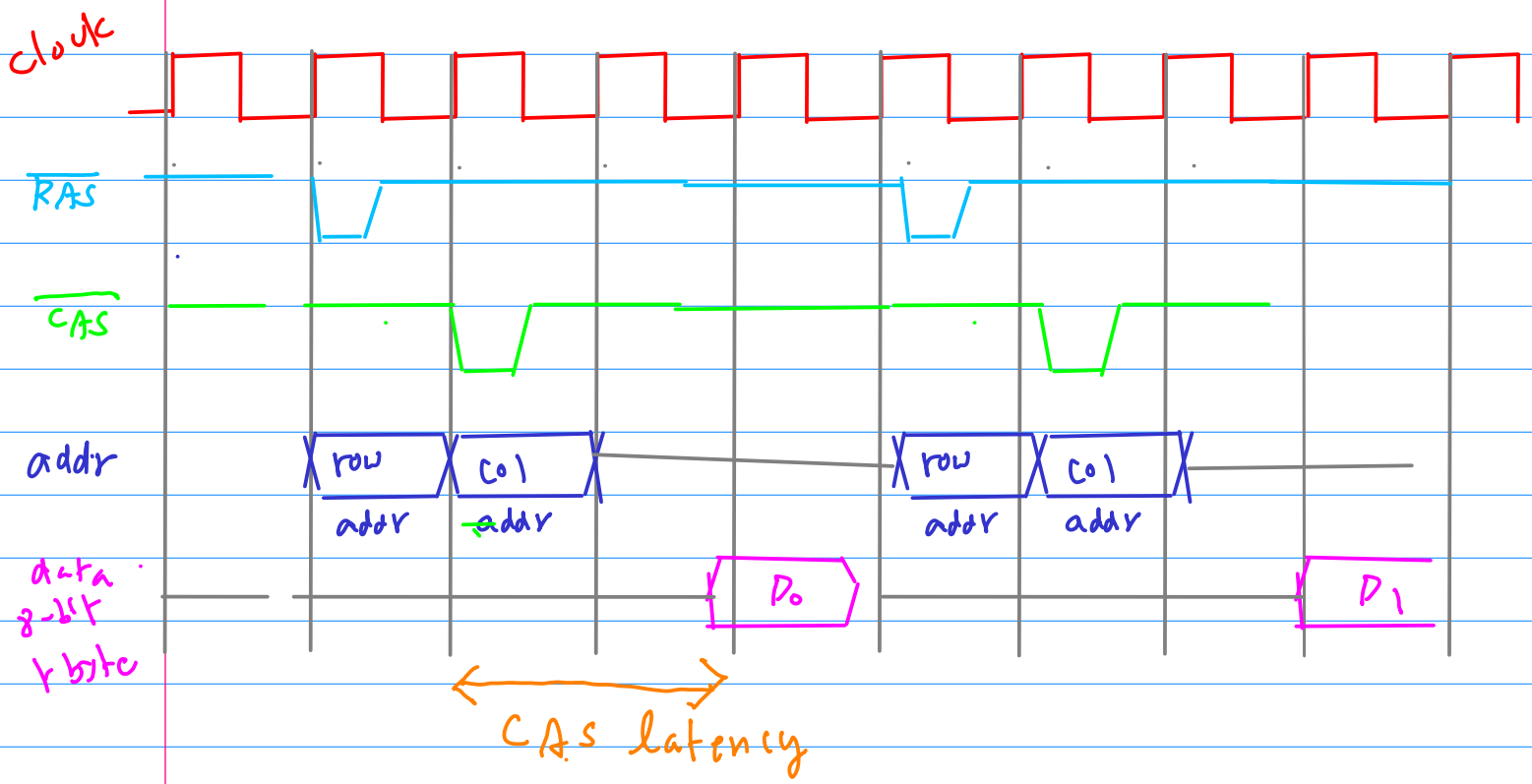
DDR2

7.5ns/2 = 3.75ns (266MHz)



Burst Read Mode

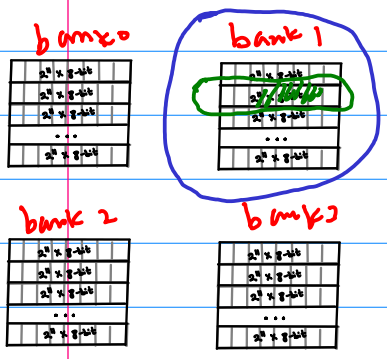




※ Bank에서 현재의 col addr, row addr
access하고 있을 CAS latency hide

같은 bank 내의
같은 row 에 있는

Burst Mode: 연속 전송 동작

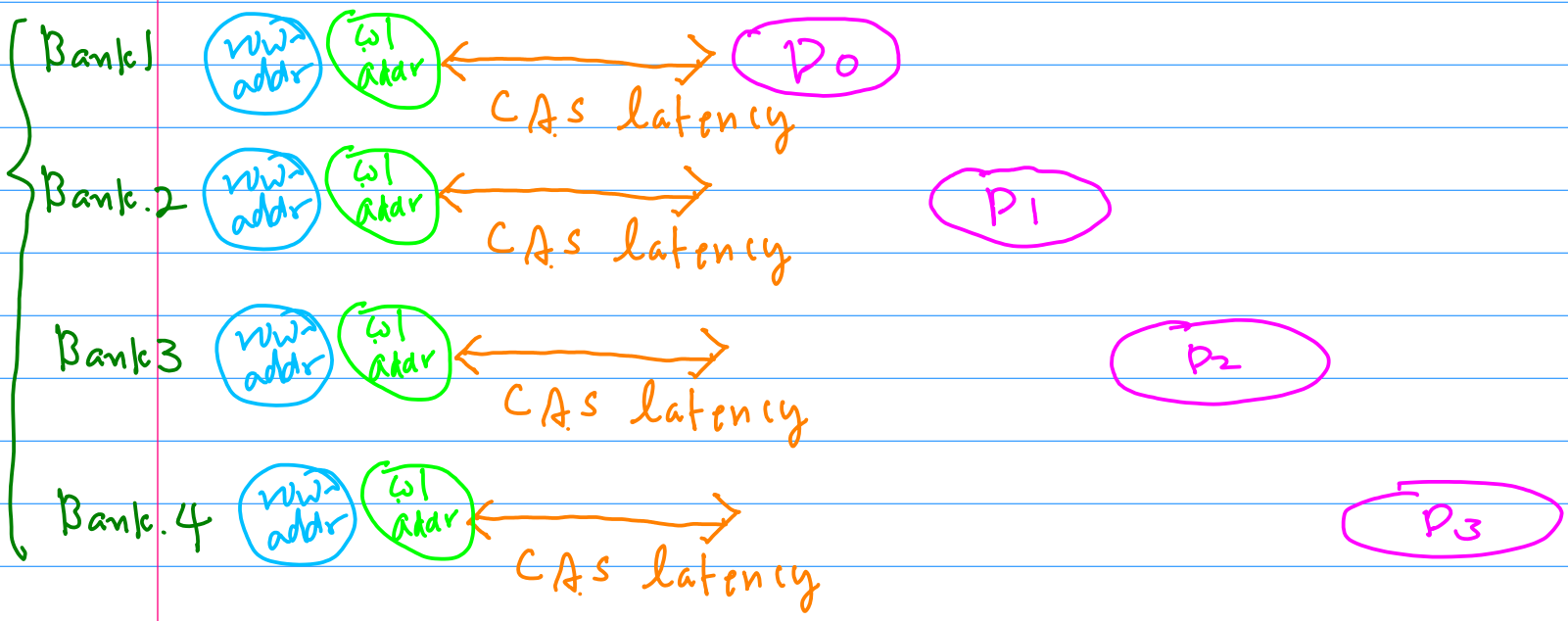


한번 읽을 때 여러개의 Byte 들을
연속적으로 전송할 수 있다

D_0, D_1, D_2, D_3

burst length = 4

각 Bank에서 한번씩 col addr, row addr
access 하므로 CAS latency hide

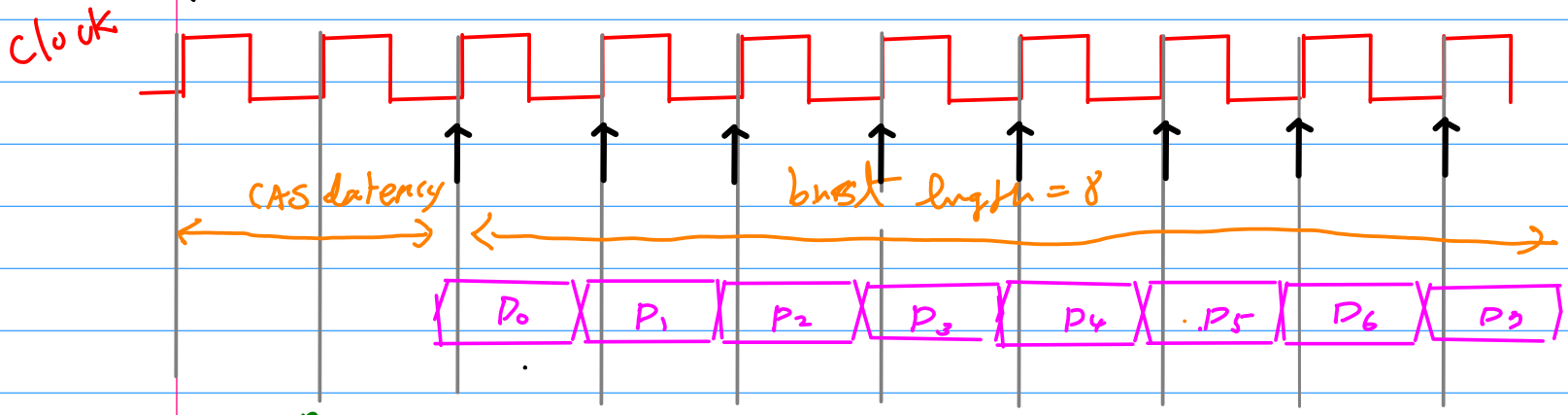


DDR (Double Data Rate)

SDRAM PC133 ~ 133 MHz

$\frac{1}{133 \times 10^6}$ sec $= \frac{1}{133}$ μsec $= 0.00752$ μsec $= 7.5$ nsec

7.5ns. (133 MHz)



8 bit X8 = 64bit

64-bit data / cycle

7.5 ns 간격 clock rising edge 마다 64-bit data가 전송

burst Length = 8 byte

8 byte 전송 끝 마지막 byte
8 × 7.5 = 60ns 필요

CAS latency 2 × 7.5 = 15 ns

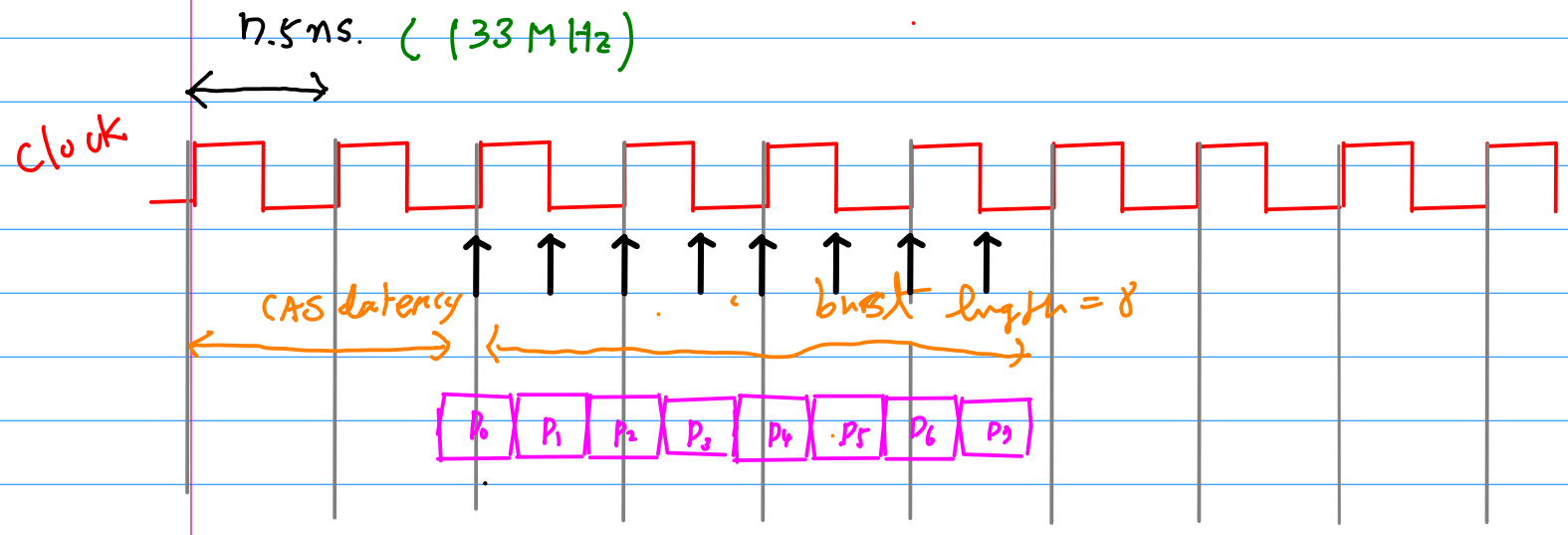
75 ns

SDRAM (Single Data Rate)

- Synchronous
- Single

DDR (Double Data Rate)

PDR-266 Clock은 133MHz x 2



{ rising edge
falling edge

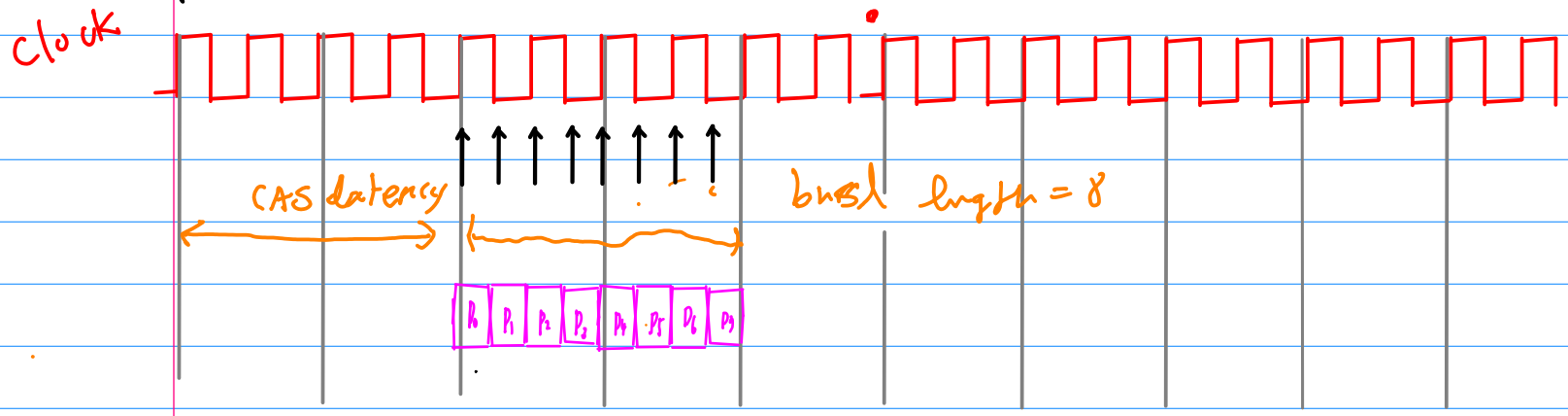
CAS latency 2 x 7.5
data transf 60 / 2 = 30
45ns

{ PDR-266 실제 Clock은 133MHz
PDR-333 실제 Clock은 166MHz
PDR-400 실제 Clock은 200MHz

DDR 2

2 배 빠른 클럭 freq를 사용하는 DDR SDRAM.

7.5ns/2 = 3.75ns (266MHz)



CAS latency 4 × 3.75 = 15

data transfer 30/2 = 15

30 ns

google

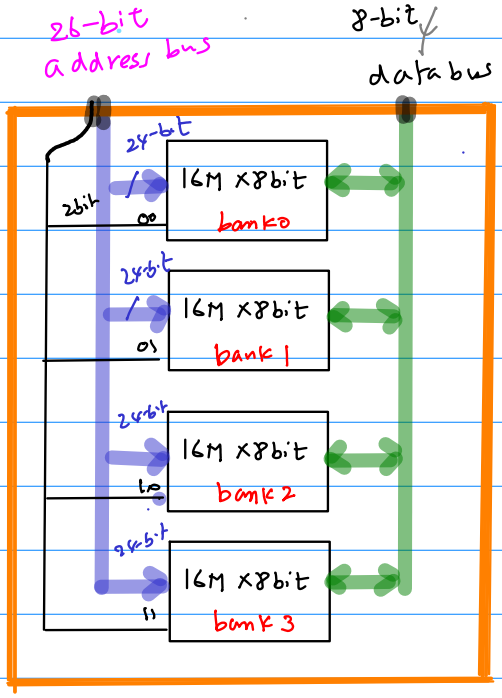
Computer Organization in plain view

wikiversity.org

(The necessities in Computer Organization

wikiversity.org)

SDRAM

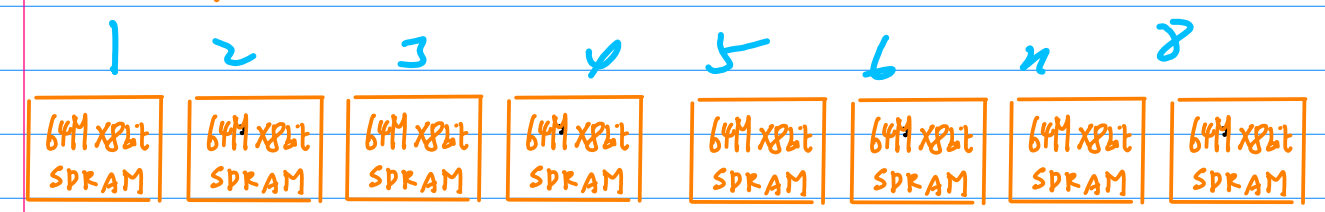


$2^{26} = 2^6 \cdot 2^{20}$

64M x8bit SDRAM

칩당 8-bit 입출력

64-bit PC



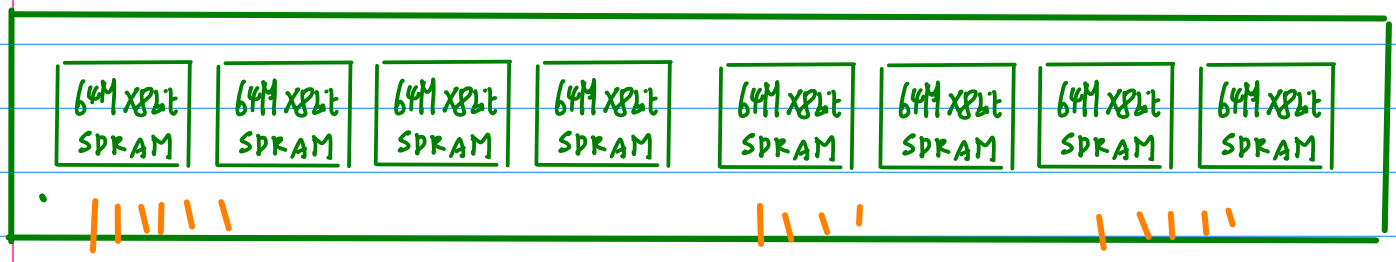
64-bit 입출력

$8 \times 64M \times 8bit$

$\frac{2^3 \cdot 2^{16} \cdot M \times 8bit}{2^{19}}$

$M \times 8bit$

512M x8-bit



SIMM (Single In-line Memory Module)

한면

DIMM (Dual In-line Memory Module)

양면

SDRAM : 2Gbit = 256M x 8bit

$2 \cdot 2^{30} = 2^8 \cdot 2^{20} \cdot 2^3$

SIMM = 8 x SDRAM = 8 x 256M x 8bit

= 8 x (2G Bit)

= 16G Bit

= 2G Byte

memory rank

일반적으로 CPU — Memory 64-bit data 전송

데이터 입출력 width가 64-bit 단위가 되도록
구성된 module을 memory rank

JEDEC 정의 데이터 입출력 폭이 64-bit가
여러개의 메모리칩을 사용하여
구성한 기억장치 module

각 rank에서 address 마다
64-bit씩 read/write 한다

SDRAM · (□ x 4bit) 16개

= □ x 64bit

SDRAM · (□ x 8bit) 8개

= □ x 64bit

Single rank module — 64-bit

rank가 한개

dual rank module — 64-bit X 2.

rank가 두개