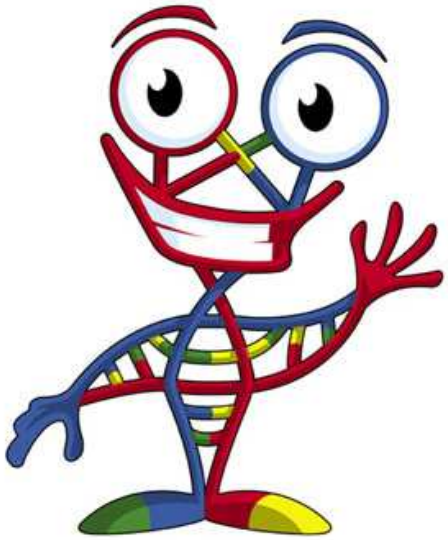
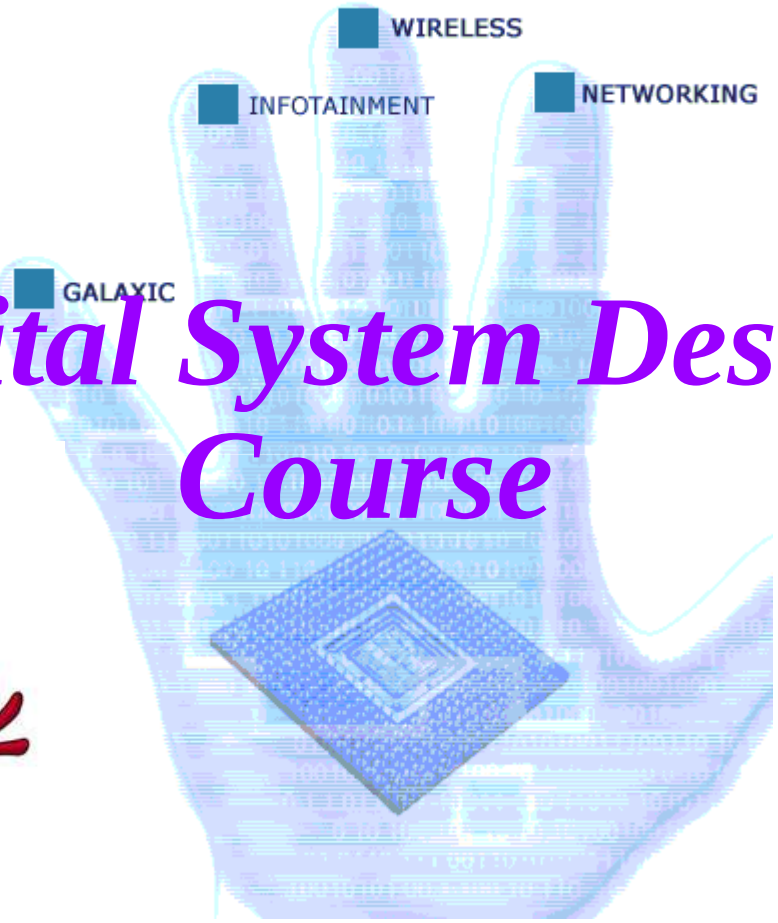
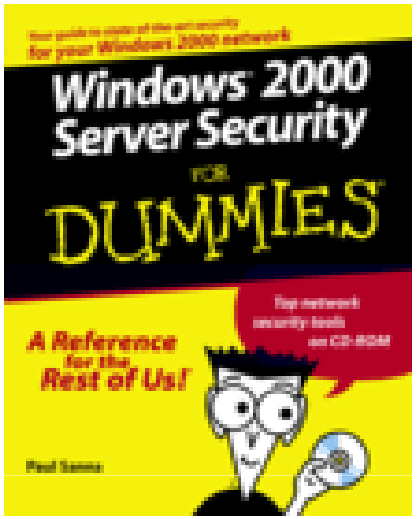
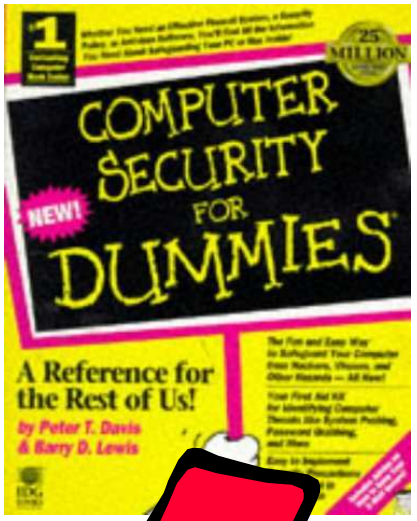


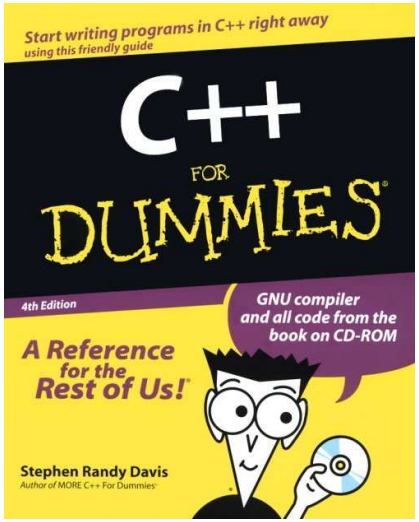


Digital System Design Course

Digital system design for Dummies

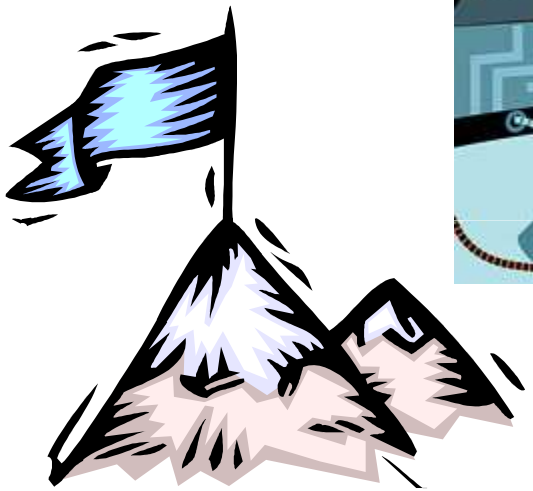


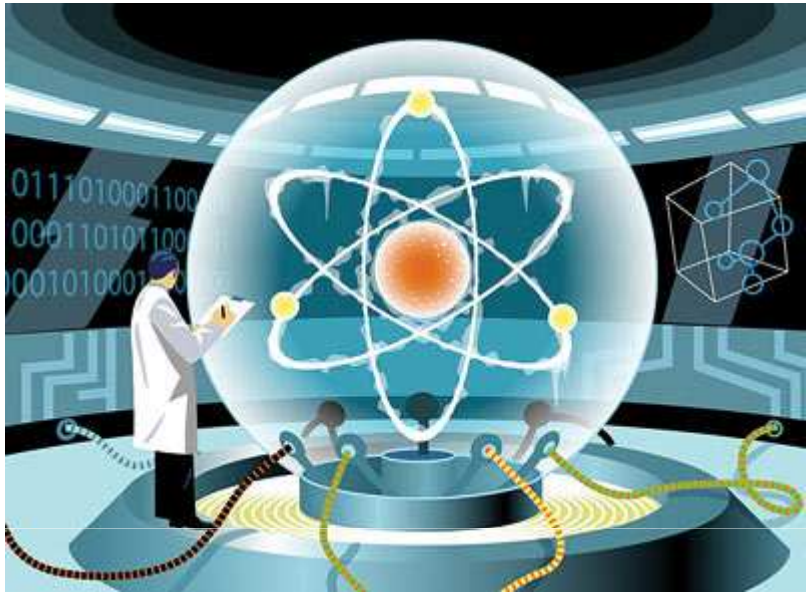




Do not read this….!!!

Digital system design for Wise Guys





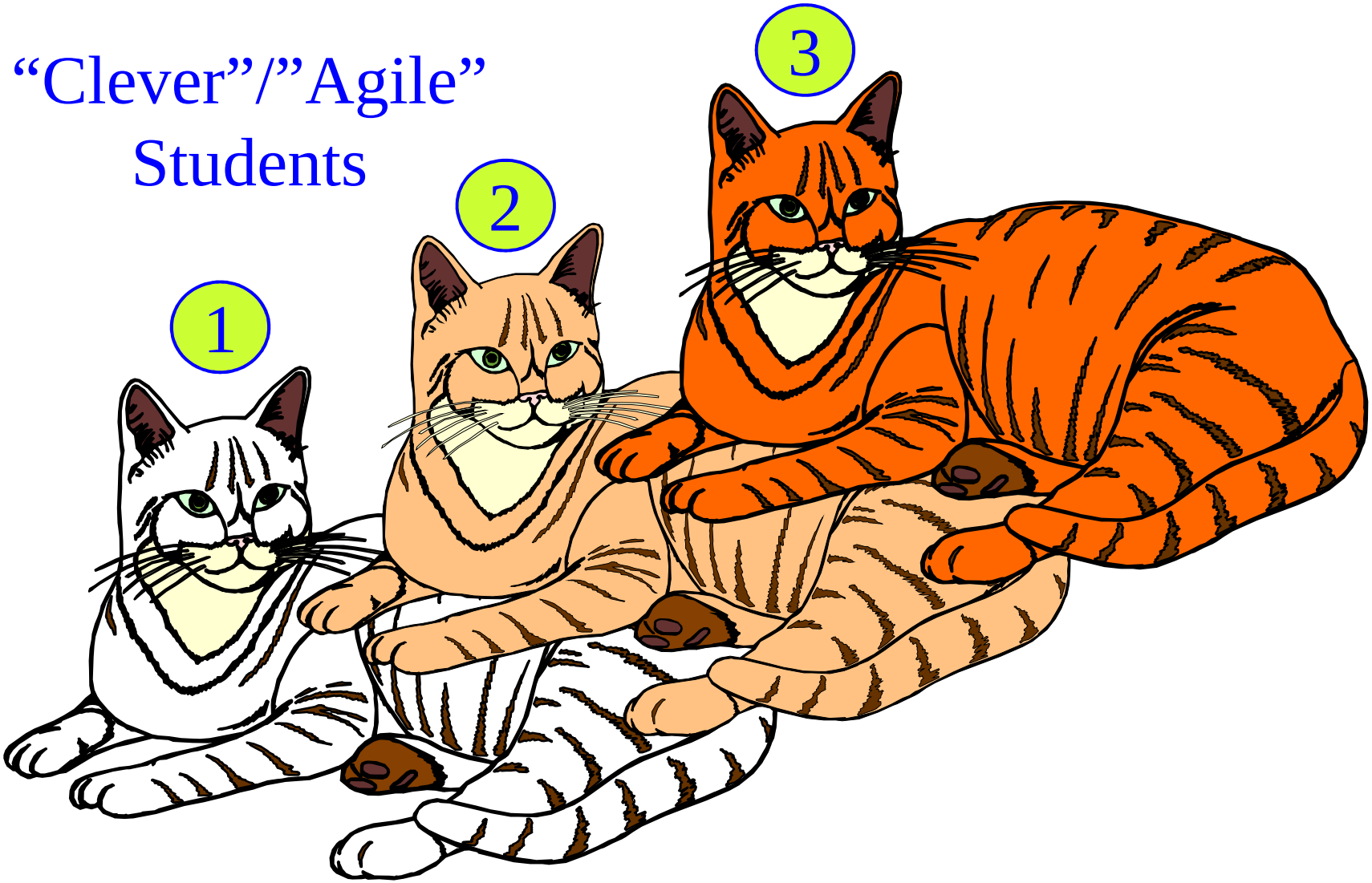
Wise Guys

Students of

Electronics Engineering

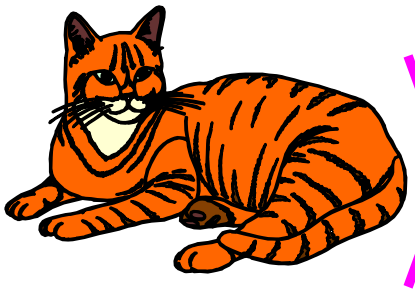
Digital system design for Wise Guys

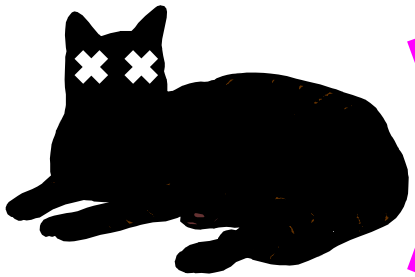
"Clever"/"Agile"
Students

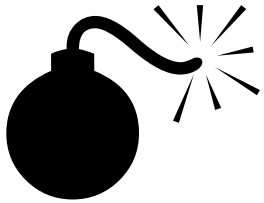


Digital system design for Wise Guys

$$\frac{1}{\sqrt{2}}\left|\ \right\rangle + \frac{1}{\sqrt{2}}\left|\ \right\rangle$$

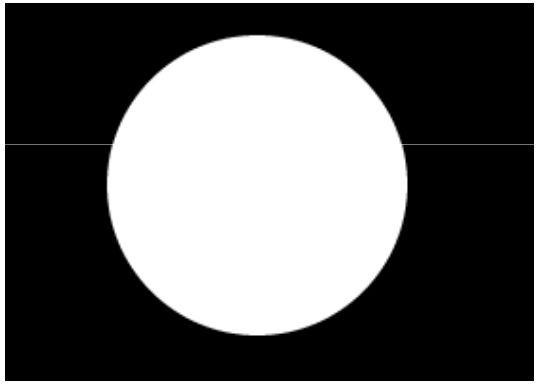


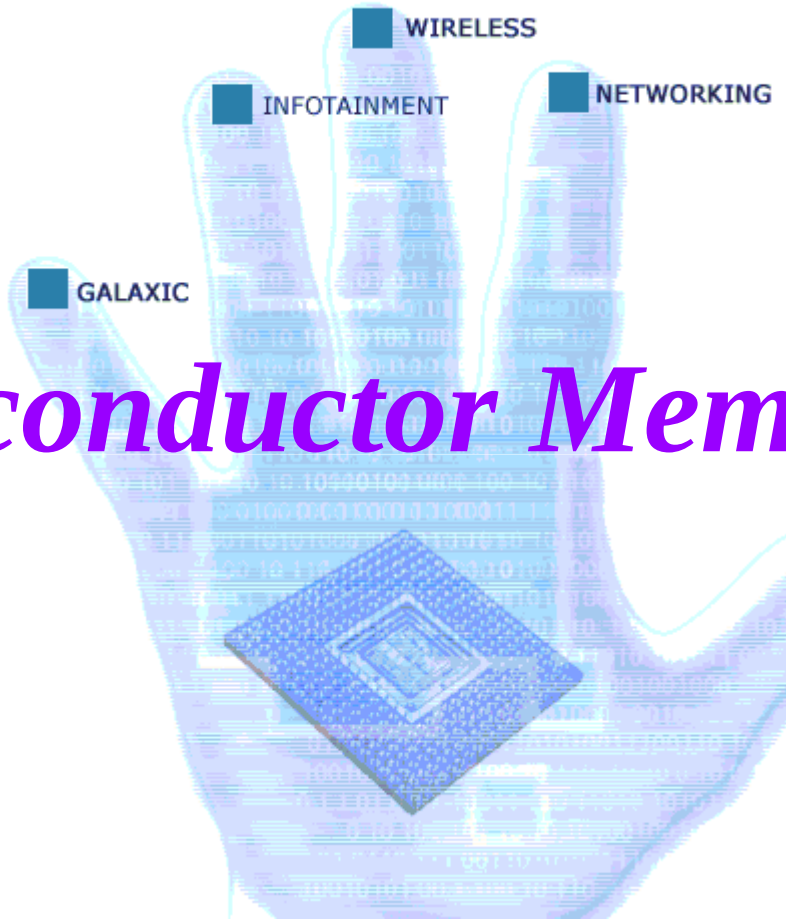




Laboratory 1  EEPROM

Digital system design for Wise Guys

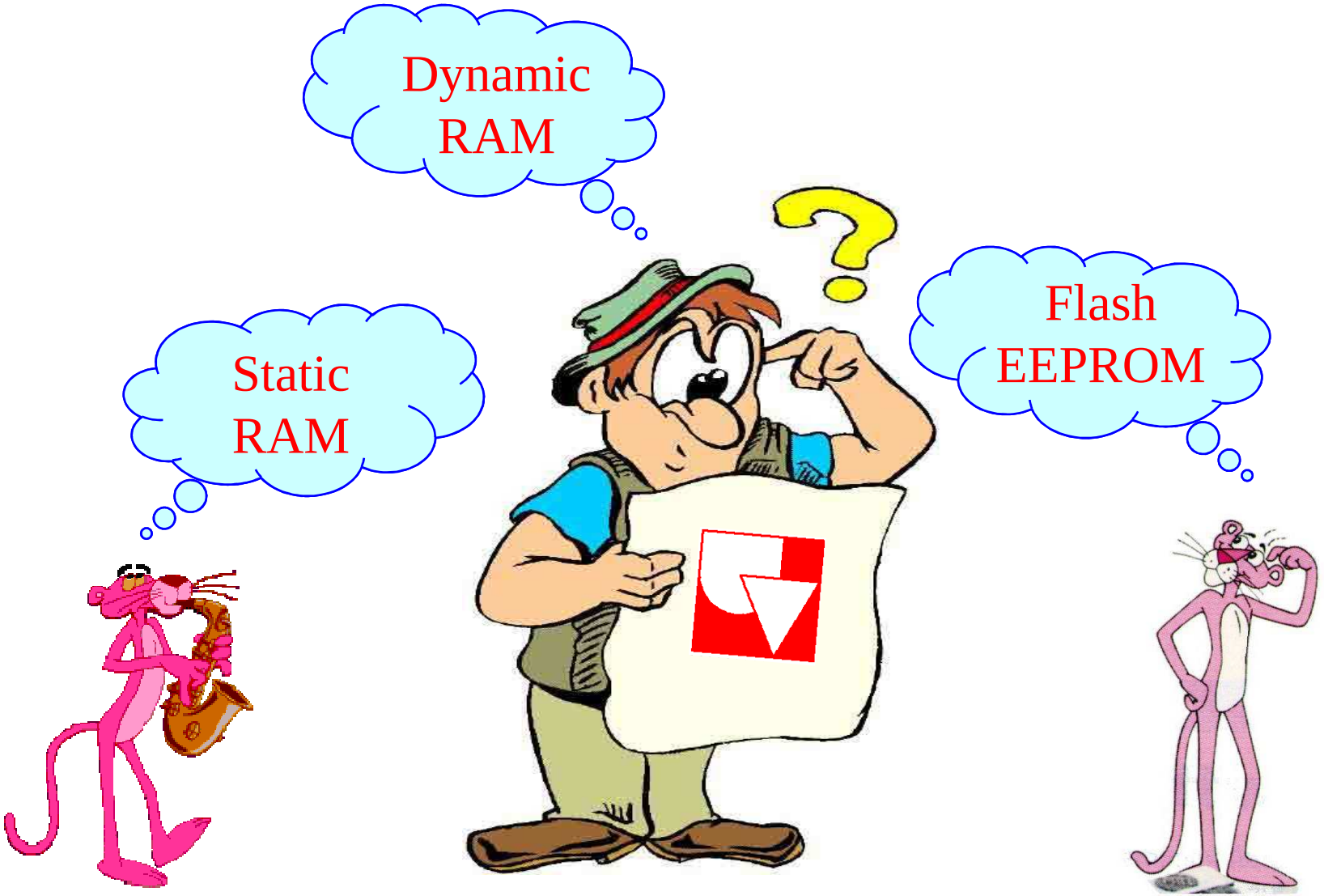




Semiconductor Memories

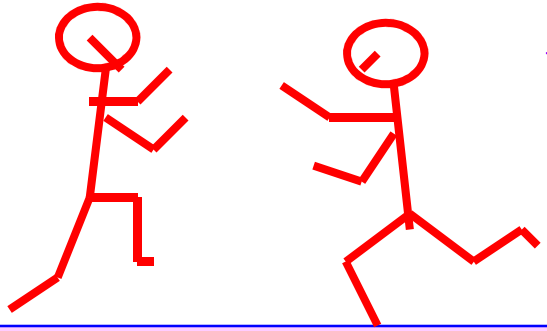
Memory Arrays

Memory Arrays


Dynamic
RAM
Static
RAM
Flash
EEPROM

Memory Arrays

Latches
Flip-flops
Registers
Register bank

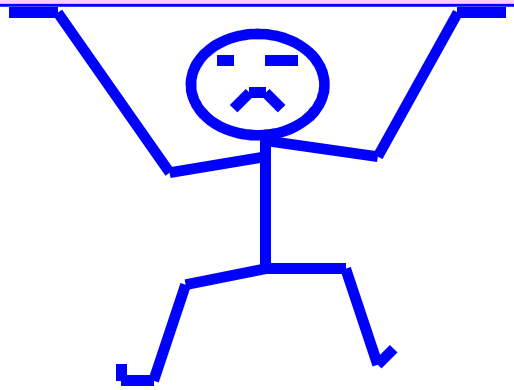


Very small capacity

Memory Arrays

RAM
- SRAM
- DRAM

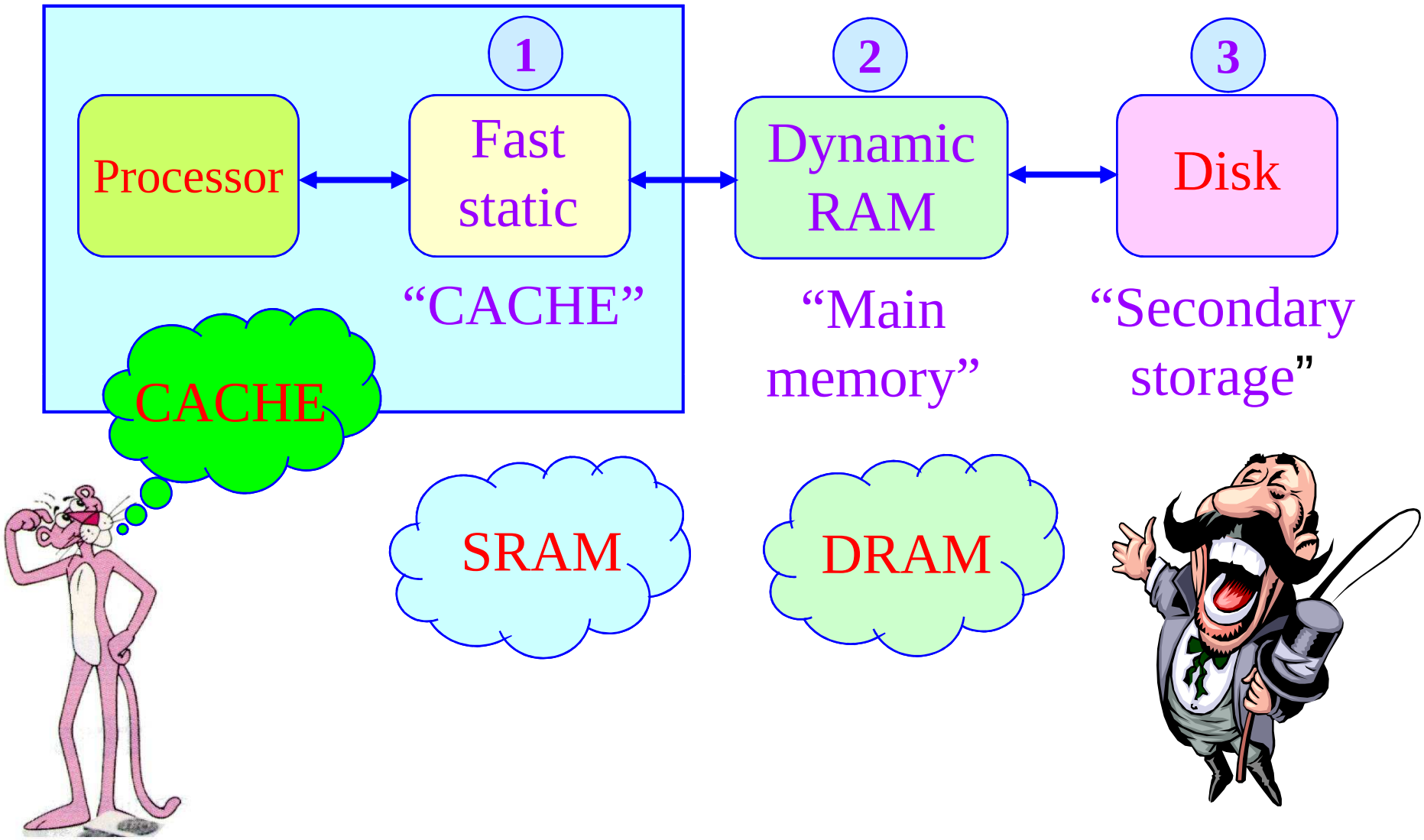
ROM
- EPROM
- EEPROM flash



Very high capacity

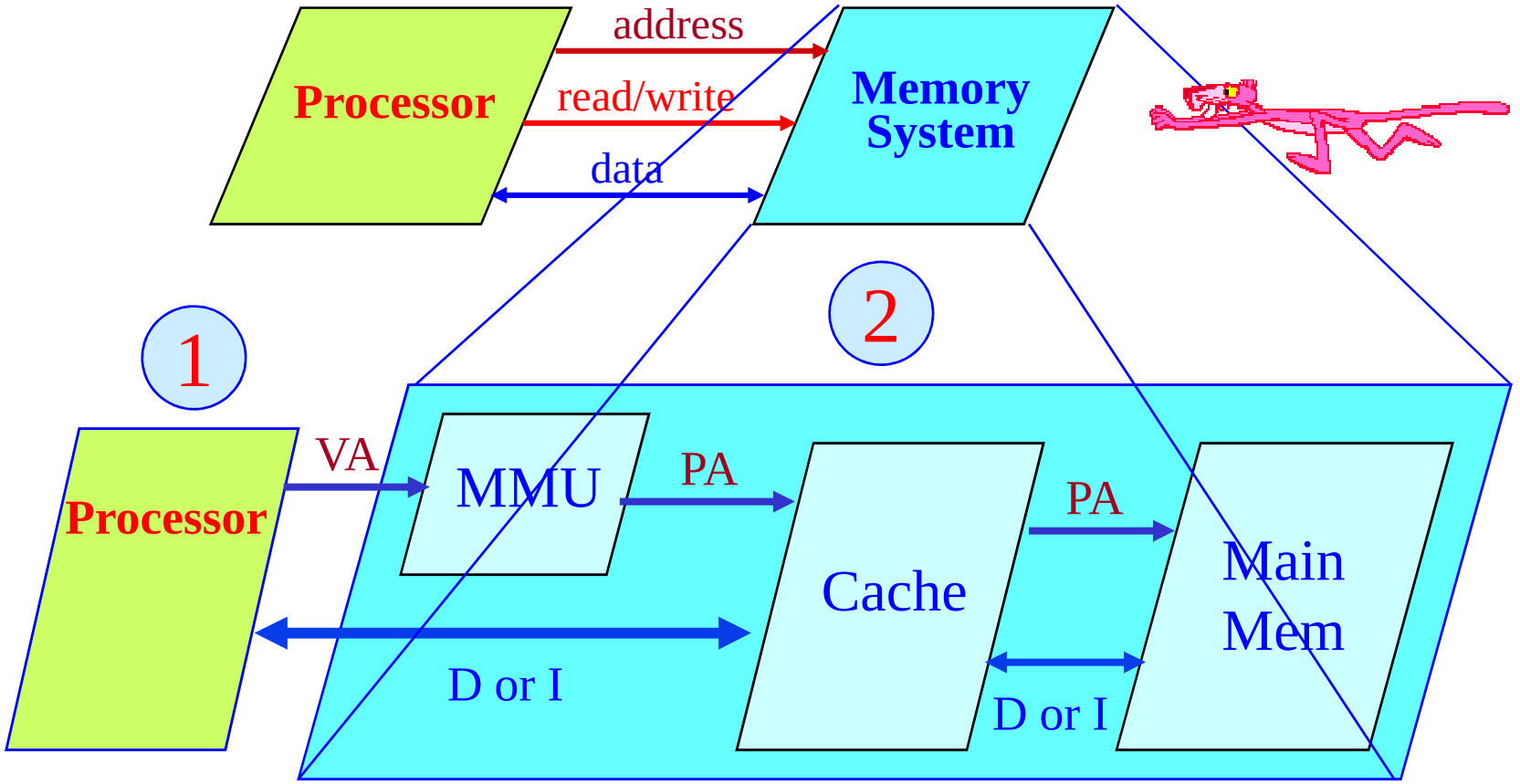

DRAM
EEPROM
Flash
TOOT

1. Memory Devices for Computers



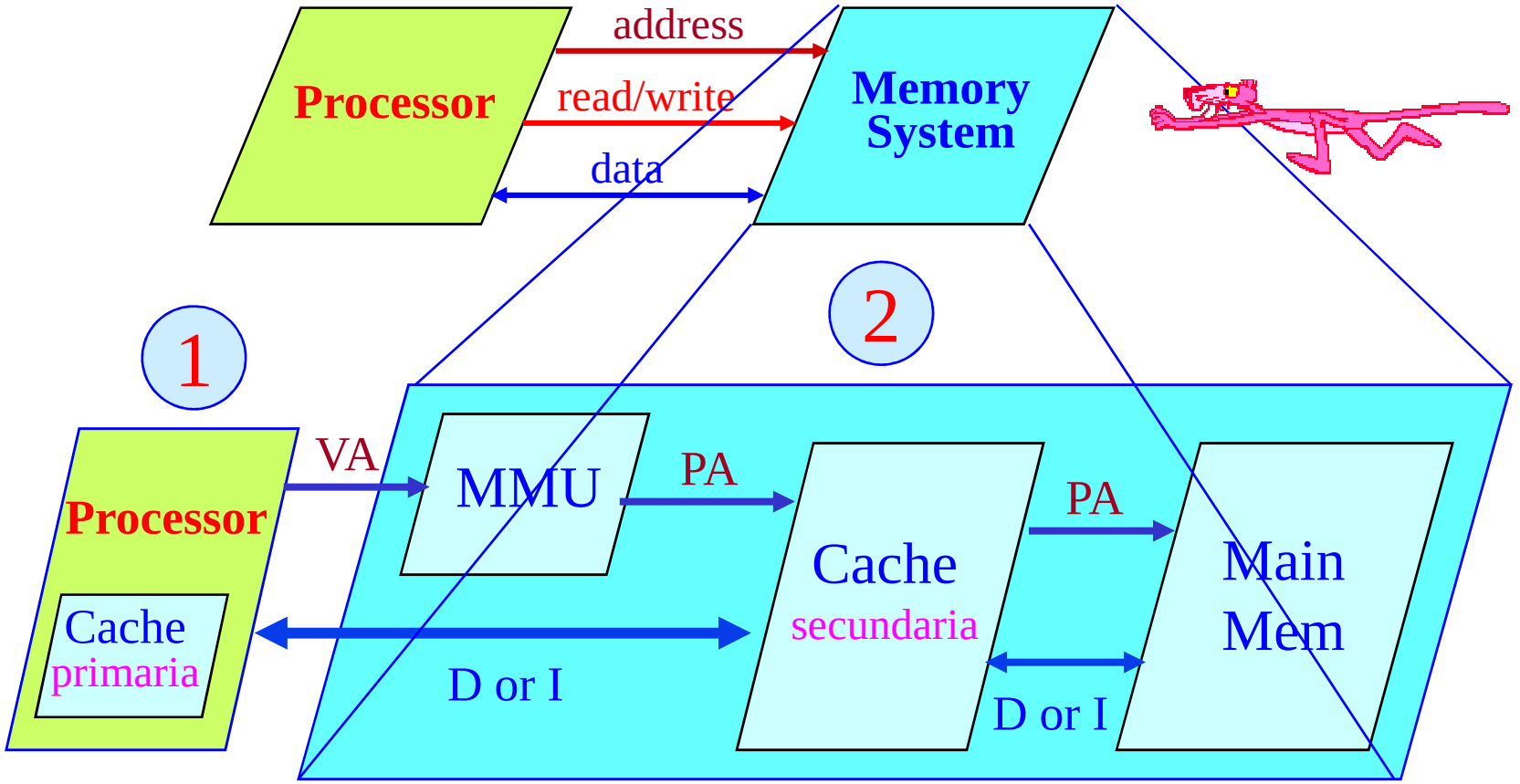
1. Anatomy of a Computer

- *Block diagram view: memory system*



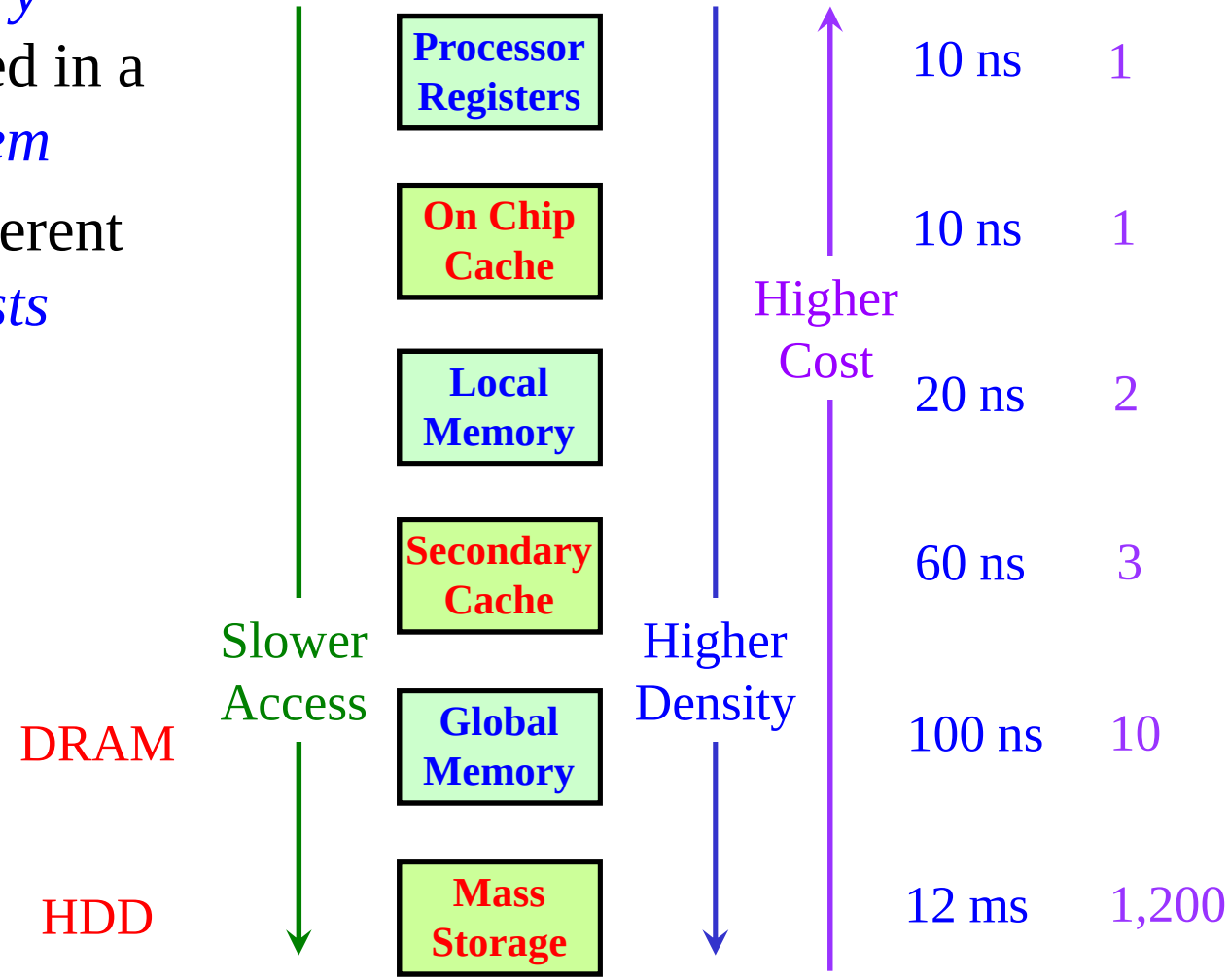
1. Anatomy of a Computer

- *Block diagram view: memory system*



1. Memory Hierarchy

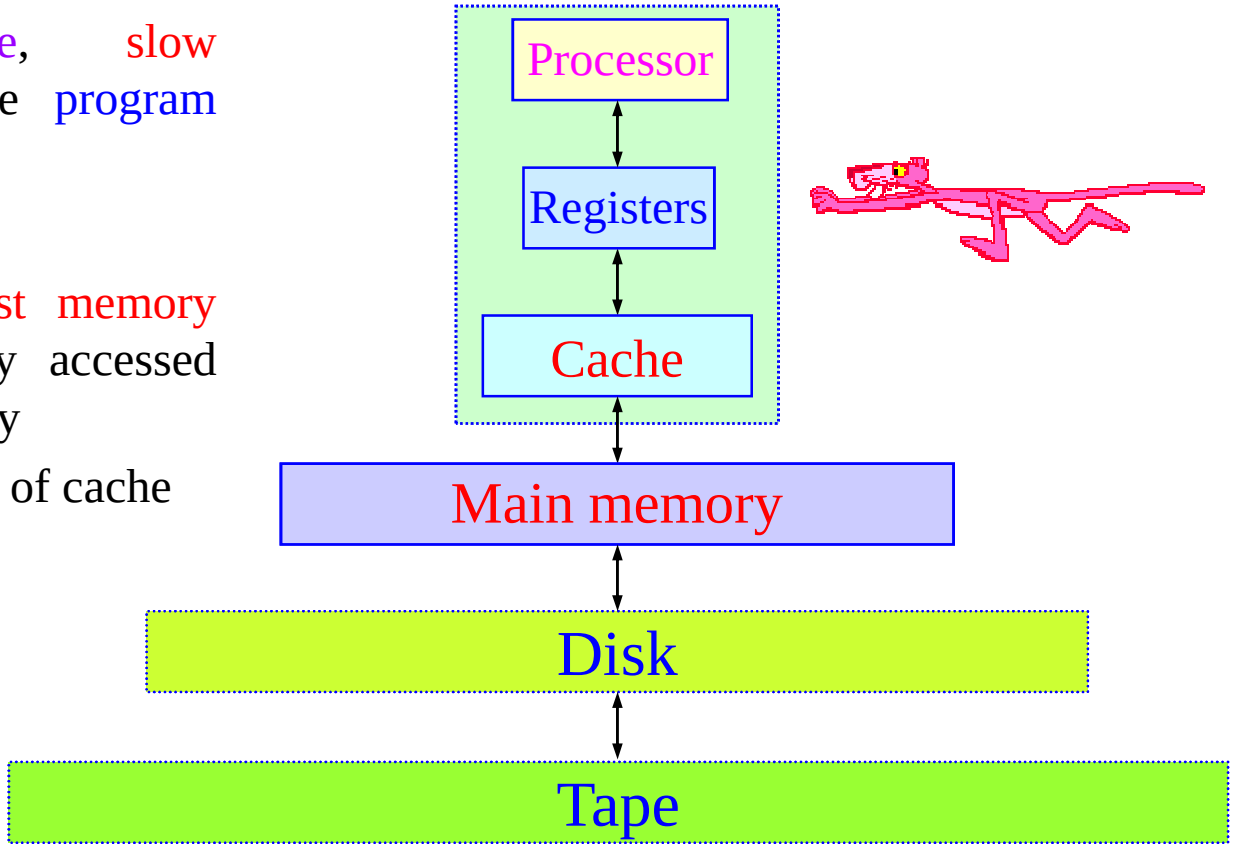
- Various *memory devices* are used in a *computer system*
- They have different *speeds and costs*



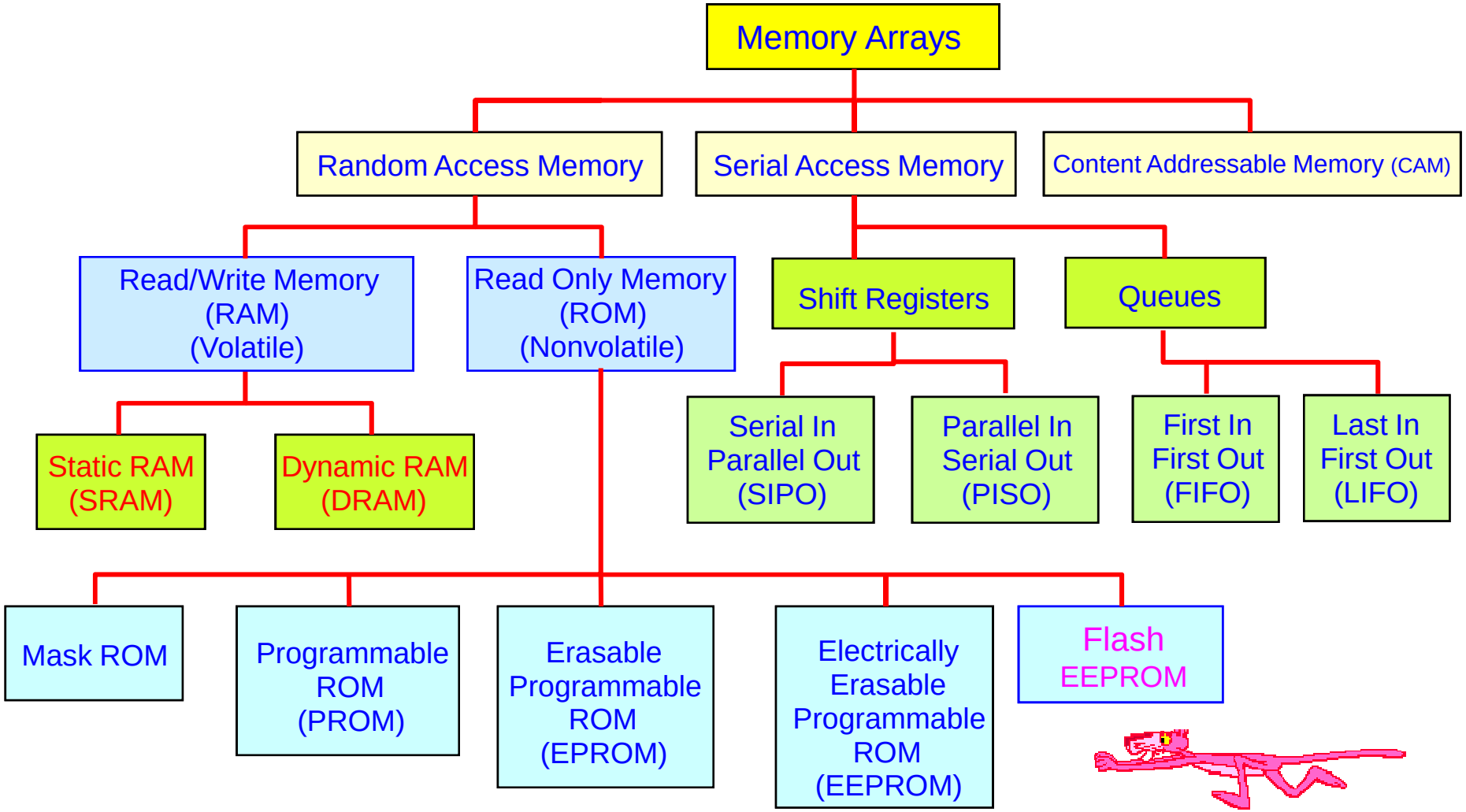
1. Memory Hierarchy

Want inexpensive, fast memory

- **Main memory**
  - Large, inexpensive, slow memory stores entire program and data
- **Cache**
  - Small, expensive, fast memory stores copy of likely accessed parts of larger memory
  - Can be multiple levels of cache



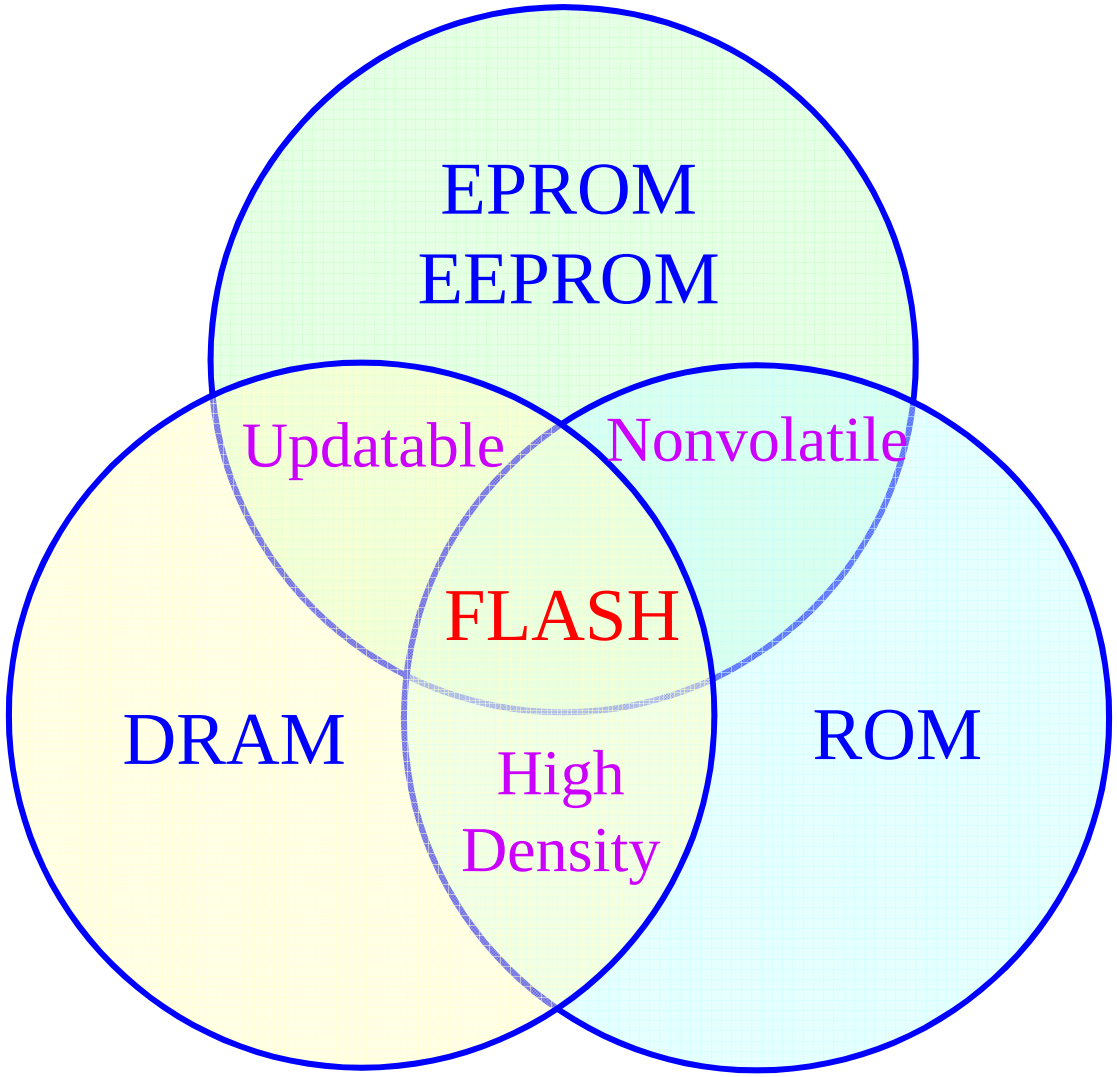
1. Memory Arrays

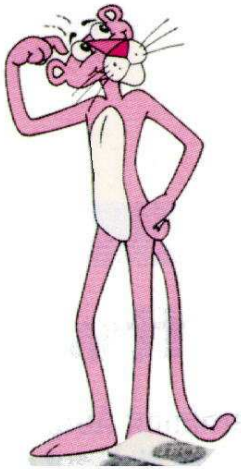


1. Memory types

Memory Types	Features
FLASH	Low-cost, high-density, high-speed architecture; low power; high reliability
ROM Read-Only Memory	Mature, high-density, reliable, low cost; time-consuming mask required, suitable for high production with stable code
SRAM Static Random-Access Memory	Highest speed, high-power, low-density memory; limited density drives up cost
EPROM Electrically Programmable Read-Only Memory	High-density memory; must be exposed to ultraviolet light for erasure
EEPROM or E^2 Electrically Erasable Programmable Read-Only Memory	Electrically byte-erasable; lower reliability, higher cost, lowest density
DRAM Dynamic Random Access Memory	High-density, low-cost, high-speed, high-power

❖ *FLASH Full-Featured Memory Solution*





1. Memory: basic concepts

- *Stores large number of bits*
  - *m* x *n*: *m* words of *n* bits each
  - k = $\text{Log}_2(m)$ address input signals
  - or $m = 2^k$ words
  - e.g., 4,096 x 8 memory:
    - 32,768 bits
    - 12 address input signals
    - 8 input/output data signals
- *Memory access*
  - r/w: selects read or write
  - enable: read or write only when asserted
  - multiport: multiple accesses to different locations simultaneously

1. Memory: basic concepts

k address signals → 2^k words

A_0	1 → 2	A_{10}	11 → 2K	A_{20}	21 → 2M
A_1	2 → 4	A_{11}	12 → 4K	A_{21}	22 → 4M
A_2	3 → 8	A_{12}	13 → 8K	A_{22}	23 → 8M
A_3	4 → 16	A_{13}	14 → 16K	A_{23}	24 → 16M
A_4	5 → 32	A_{14}	15 → 32K	A_{24}	25 → 32M
A_5	6 → 64	A_{15}	16 → 64K	A_{25}	26 → 64M
A_6	7 → 128	A_{16}	17 → 128K	A_{26}	27 → 128M
A_7	8 → 256	A_{17}	18 → 256K	A_{27}	28 → 256M
A_8	9 → 512	A_{18}	19 → 512K	A_{28}	29 → 512M
A_9	10 → 1024 = 1K	A_{19}	20 → 1024K = 1M	A_{29}	30 → 1024M = 1G

1. Memory: basic concepts

$2^{1} = 2$

$2^{2} = 4$

$2^{3} = 8$

$2^{4} = 16$

$2^{5} = 32$

$2^{6} = 64$

$2^{7} = 128$

$2^{8} = 256$

$2^{9} = 512$

$2^{10} = 1024$

$2^{11} = 2048$

$2^{12} = 4096$

$2^{13} = 8192$

$2^{14} = 16384$

$2^{15} = 32768$

$2^{16} = 65536$

$2^{17} = 131072$

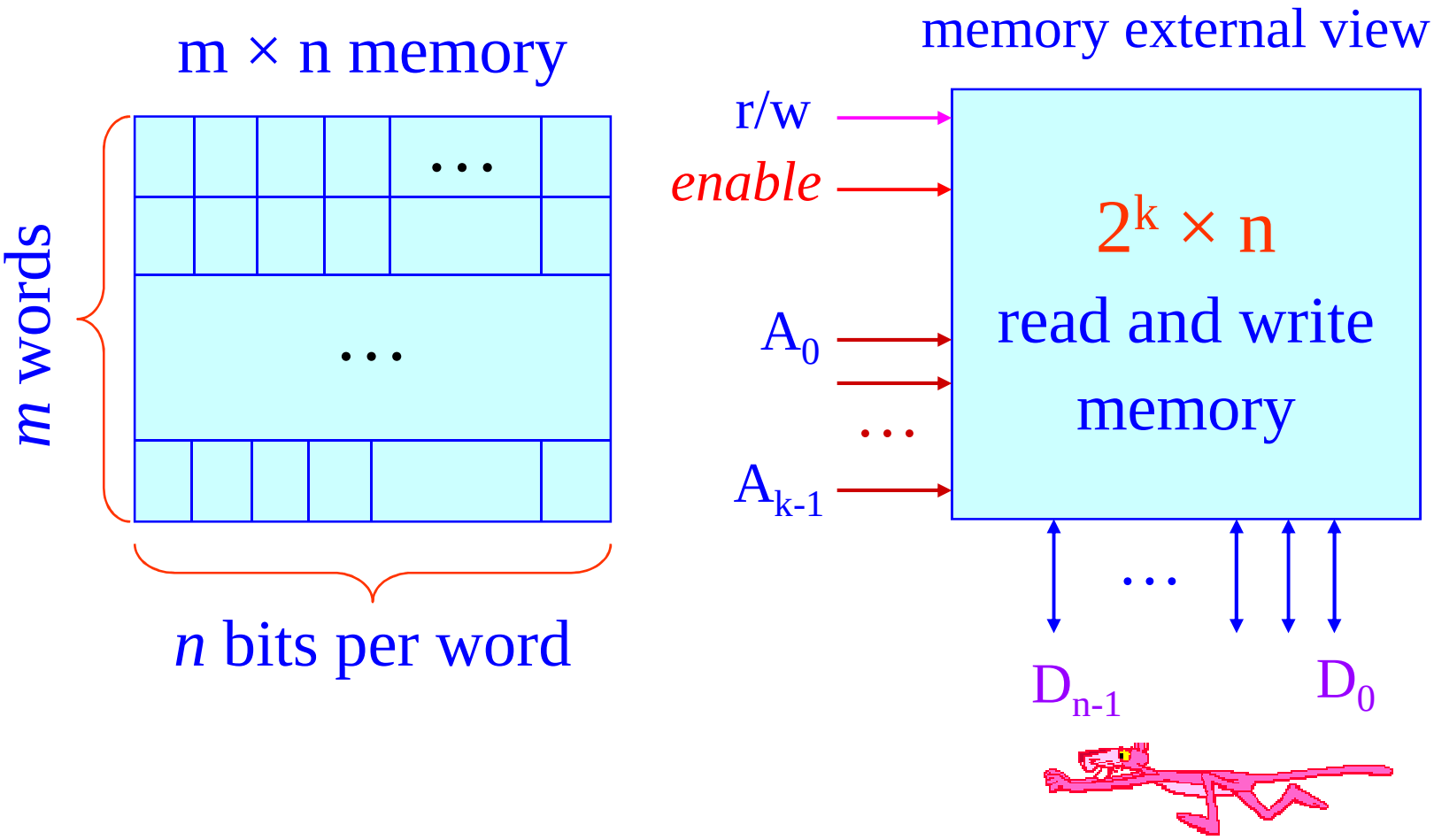
$2^{18} = 262144$

$2^{20} = 1\ X10^{6}$

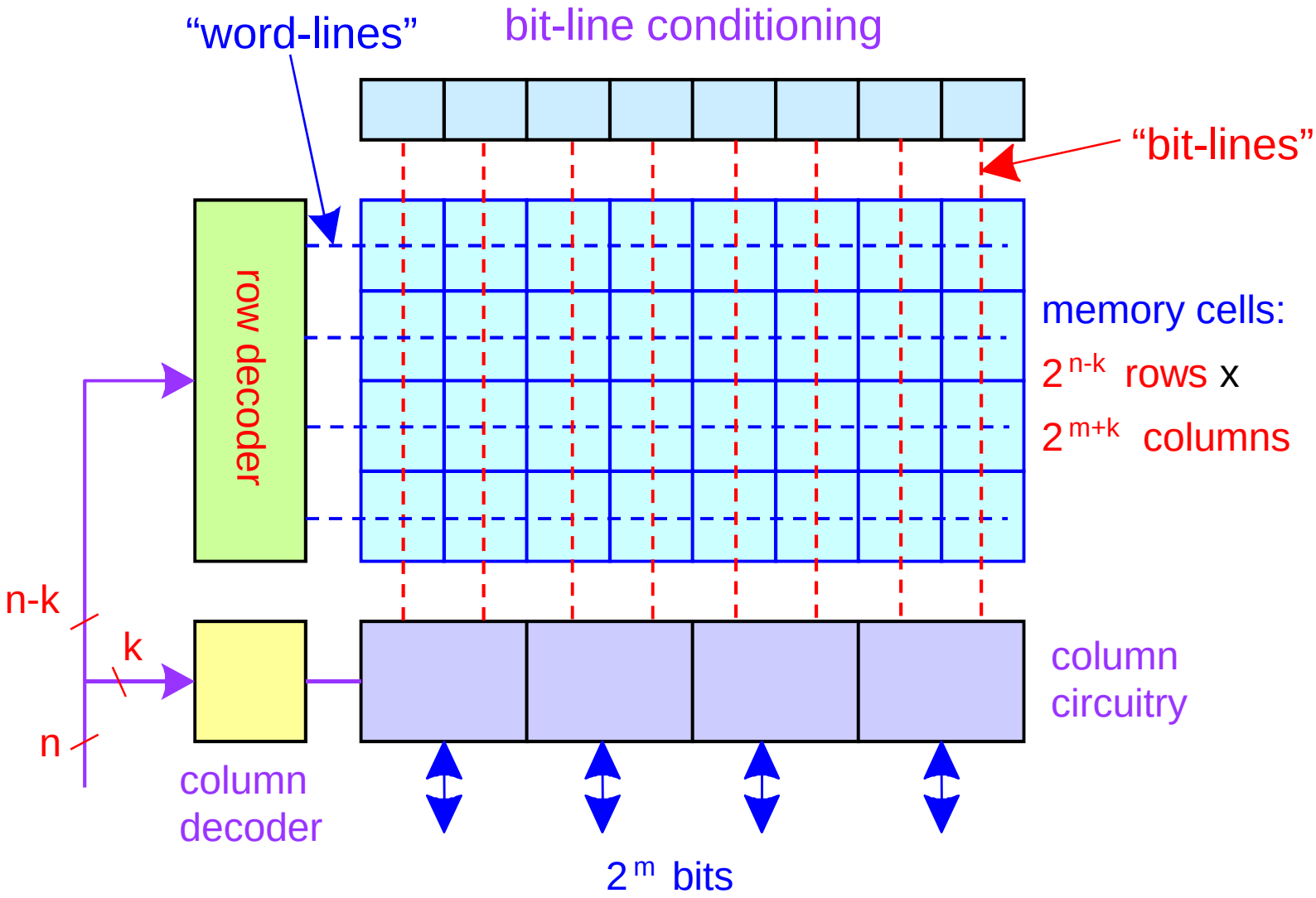
$2^{30} = 1\ X10^{9}$

$2^{40} = 1\ X10^{15}$

1. Memory: basic concepts



1. Memory Architecture



1. Array Architecture

- 2^n *words* of 2^m *bits* each
- If n >> m, fold by 2^k into fewer *rows* of more *columns*
- Good regularity – easy to design
- Very high density if good cells are used

2. ROM: Read-Only Memory

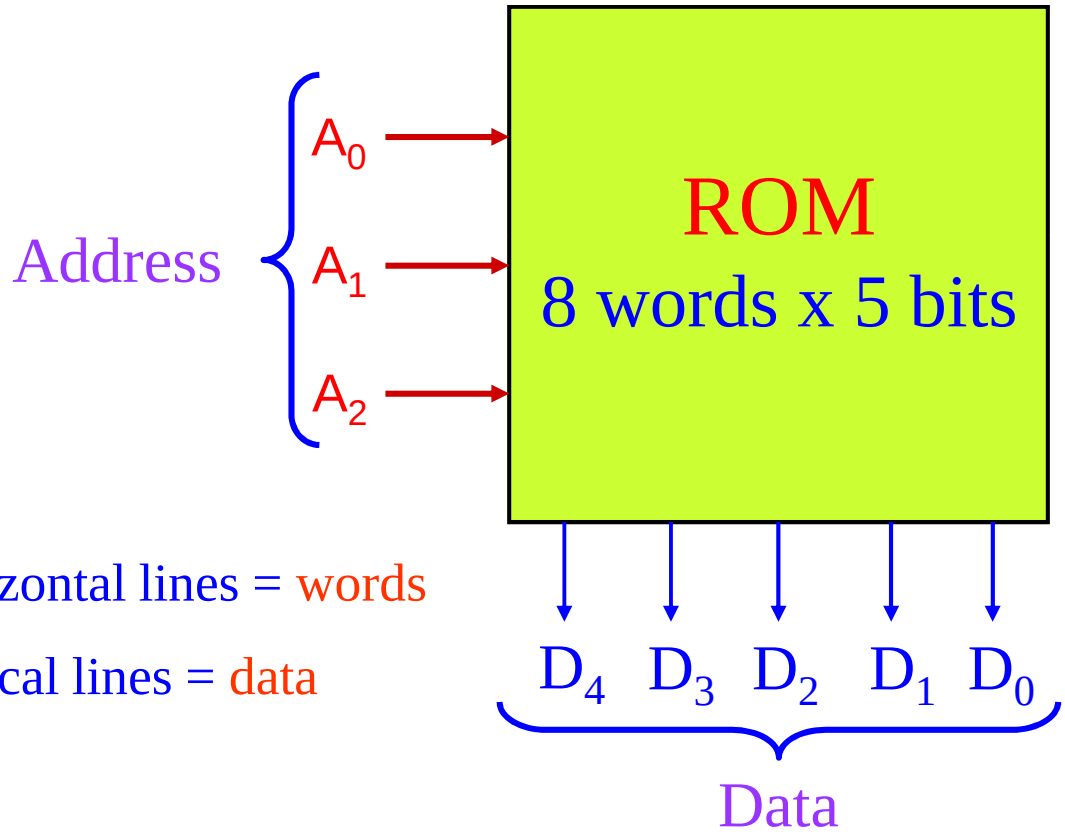
- ROM used for *storing programs* and *constant tables*
- Conflicting requirements
  - non-volatility
  - (re)-programability

2. ROM Types

- EEPROM: E^2PROM and FLASH E^2PROM
  - charge on a "floating gate"
  - erased by electrical signal
- (UV) EPROM
  - charge on a "floating gate"
  - erased by UV light
- OTP PROM
  - fusible nichrome/polysilicon links
  - programmed by blowing fuses
- ROM
  - mask-programmed
- Emulated ROM ≡ NVRAM ≡ RAM-with-a-battery

2.1 ROM: Read-Only Memory

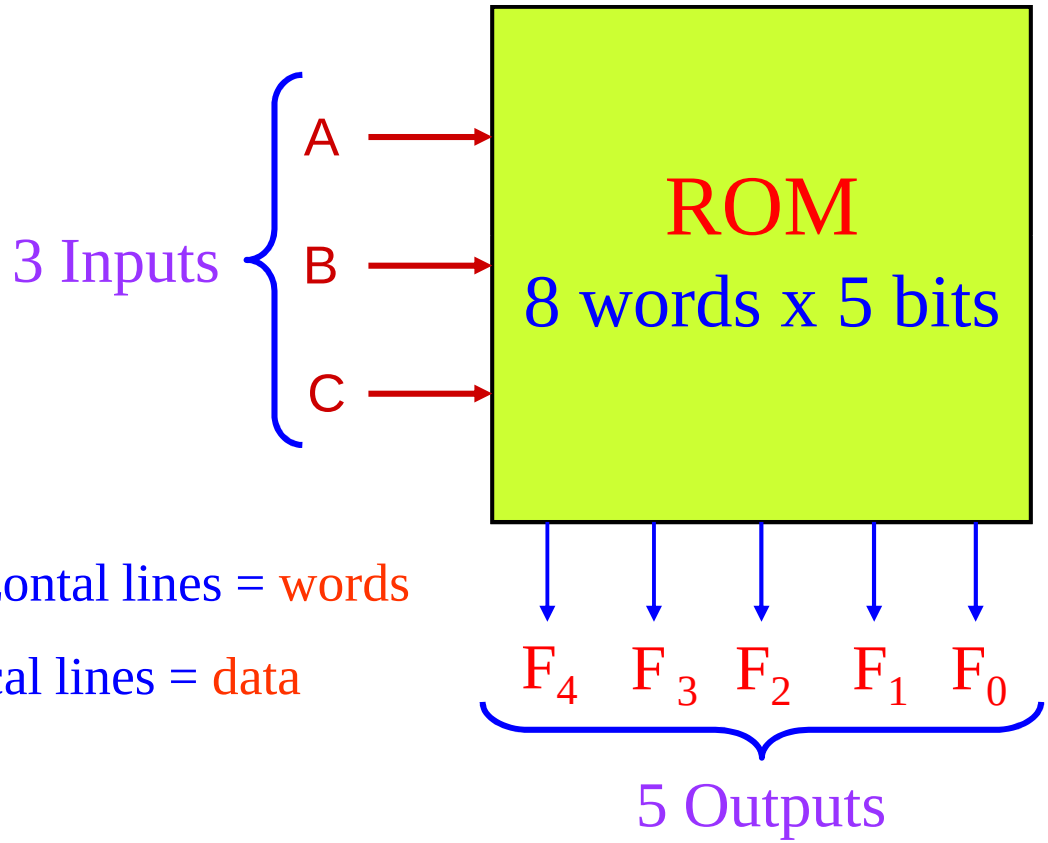
- *N input bits* (3) $\rightarrow 2^N$ words (8) by M bits (5)
- *M arbitrary functions* (5) of *N variables* (3): 8 words by 5 bits



- Horizontal lines = words
- Vertical lines = data

2.1 ROM: Read-Only Memory

- *N input bits* (3) $\rightarrow 2^N$ words (8) by M bits (5)
- *M arbitrary functions* (5) of *N variables* (3): 8 words by 5 bits



- Horizontal lines = words
- Vertical lines = data

2.1 ROM: Read-Only Memory

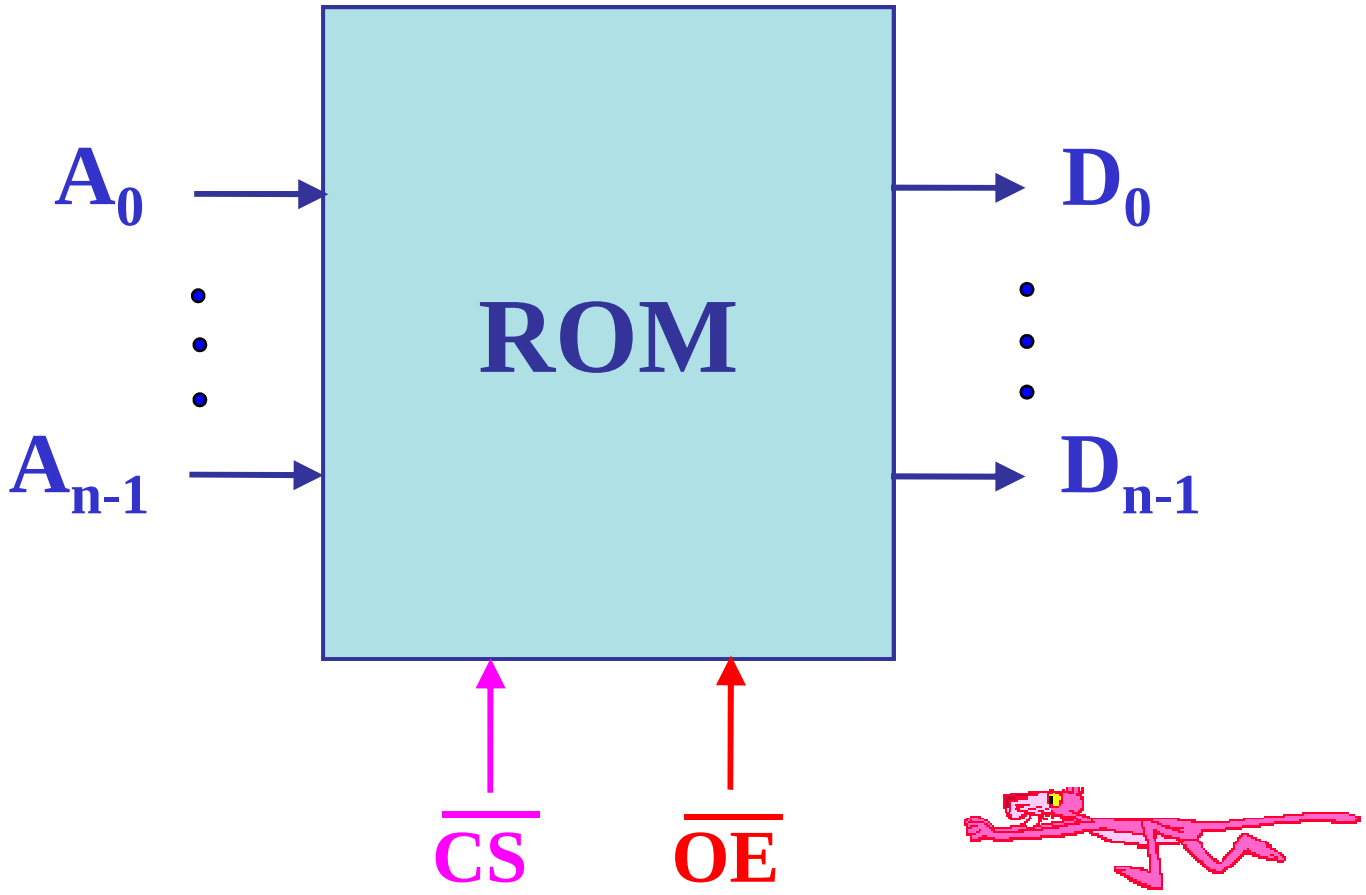
- Three inputs:
  - address bits: 16
  - *chip select*
  - *output enable*
- One output:
  - typically 8 bits of *tristate data output*
- *Access time* (time from stable address to stable data) *450 ns - 70 ns*

2.1 ROM: Read-Only Memory

- *Memoria de sólo lectura: ROM*
  - La ROM es un conversor de código
  - No es necesario almacenar los bits en flip-flops
  - Se puede implementar a partir de circuitos combinatorios: *decodificadores/codificadores*

2.1 Memoria ROM: Read Only Memory

- Diagrama a nivel de bloque: ROM



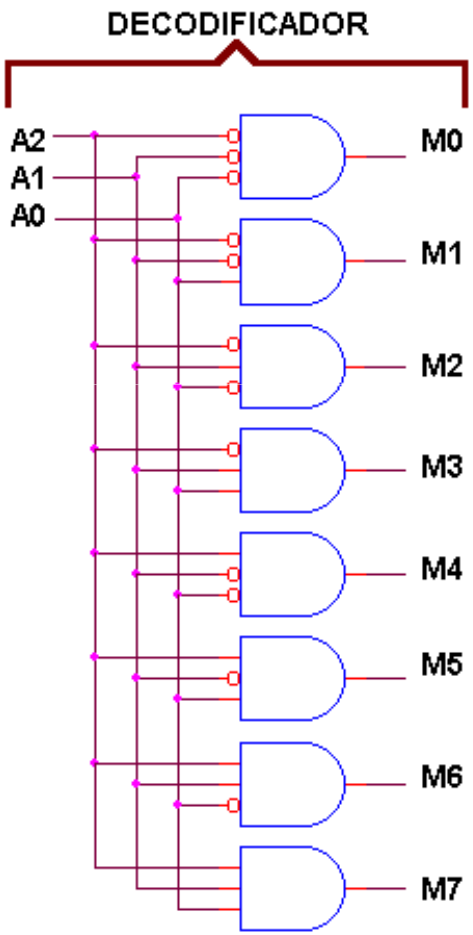
2.1 Memoria ROM: Read Only Memory

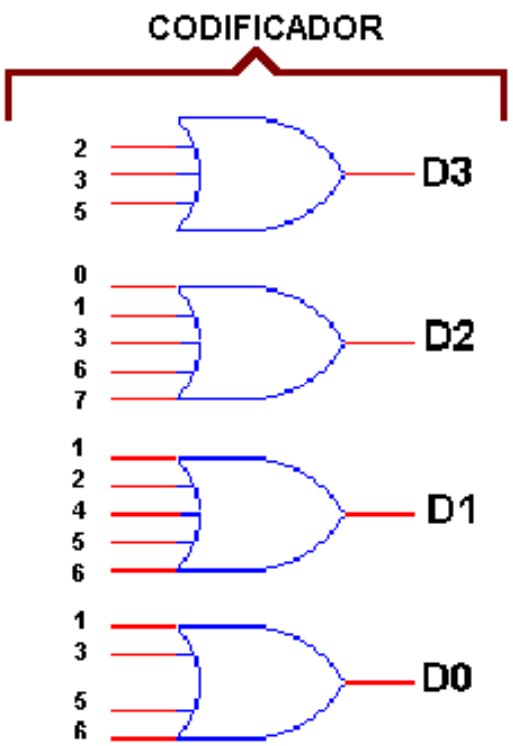
- Tabla de verdad: ROM

POSICIÓN DE MEMORIA	DIRECCIÓN			PALABRA DE DATOS			
	A2	A1	A0	D3	D2	D1	D0
M0	0	0	0	0	1	0	0
M1	0	0	1	0	1	1	1
M2	0	1	0	1	0	1	0
M3	0	1	1	1	1	0	1
M4	1	0	0	0	0	1	0
M5	1	0	1	1	0	1	1
M6	1	1	0	0	1	1	1
M7	1	1	1	0	1	0	0

2.1 Memoria ROM: Read Only Memory

- Implementación física: ROM



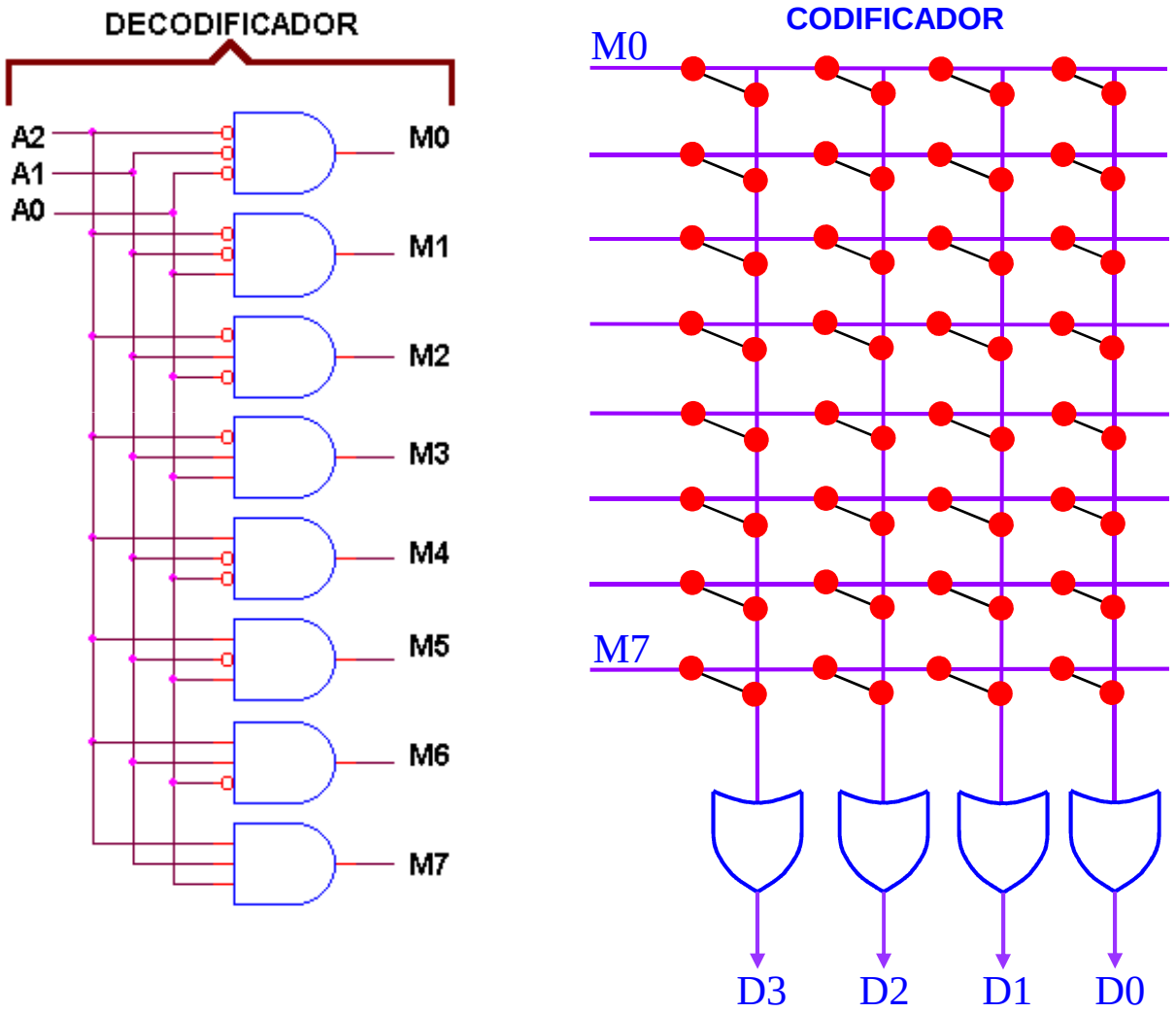


$$f_3 = \sum m(2,3,5)$$

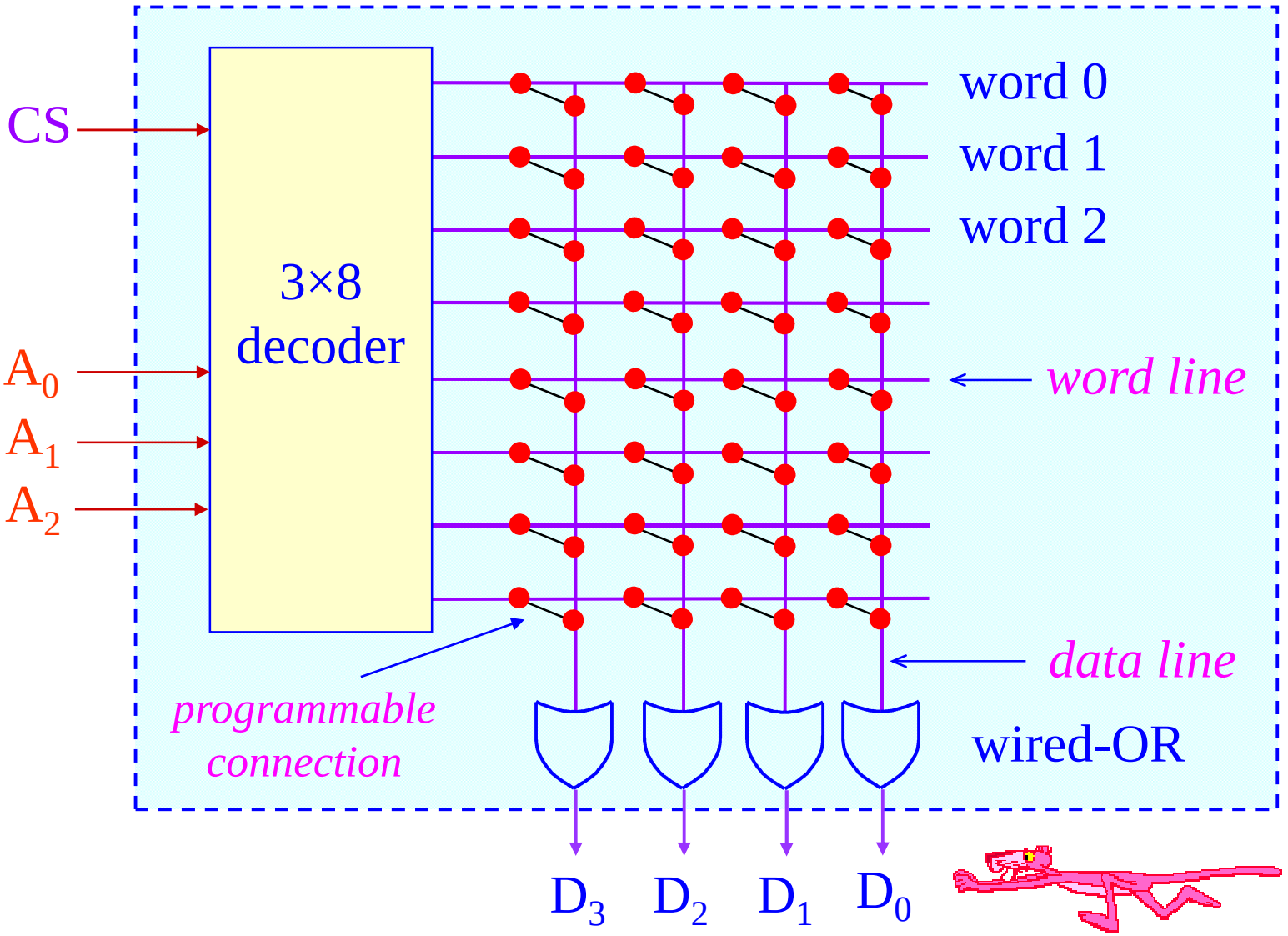
$$f_0 = \sum m(1,3,5,6)$$

2.1 Memoria ROM: Read Only Memory

- Arreglo de interconexión

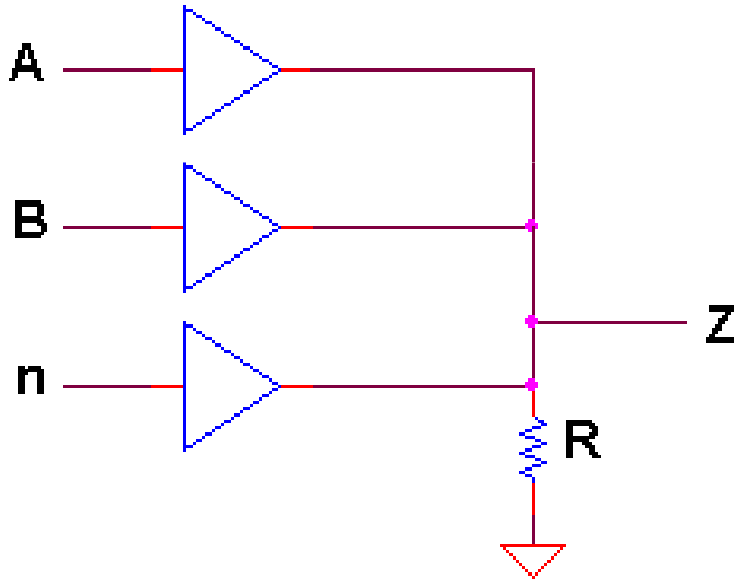


2.1 ROM 8 x 4 bits: Internal view



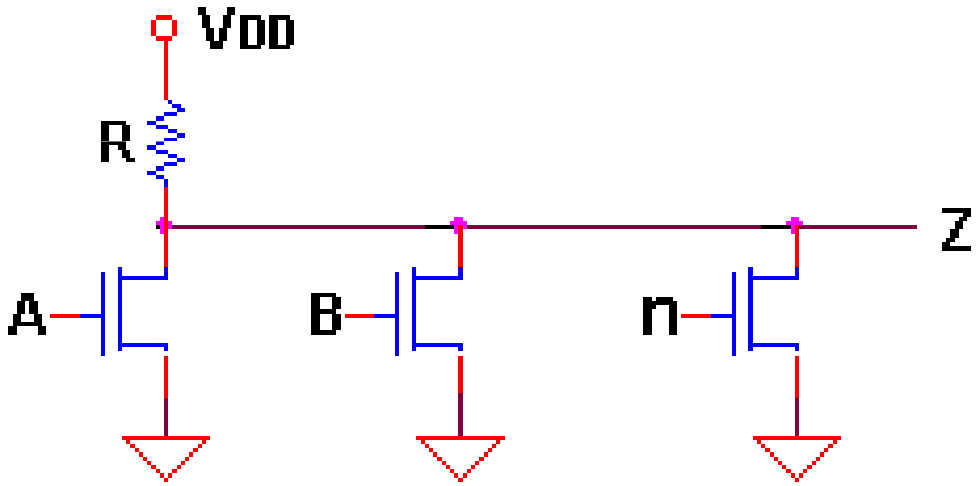
2.1 Memoria ROM: Read Only Memory

- Codificador:
  - función OR de N entradas: *diodos*



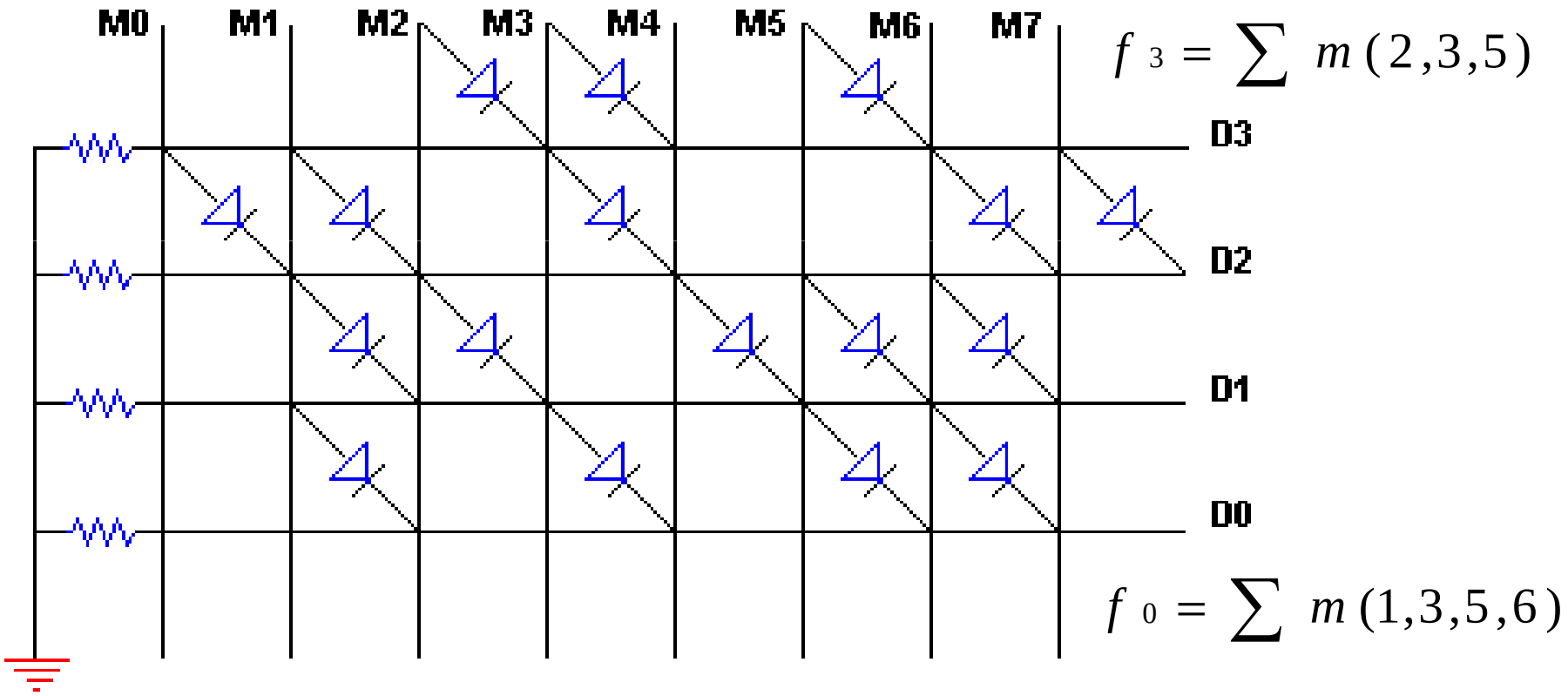
2.1 Memoria ROM: Read Only Memory

- Codificador:
  - función NOR de N entradas: *transistores*



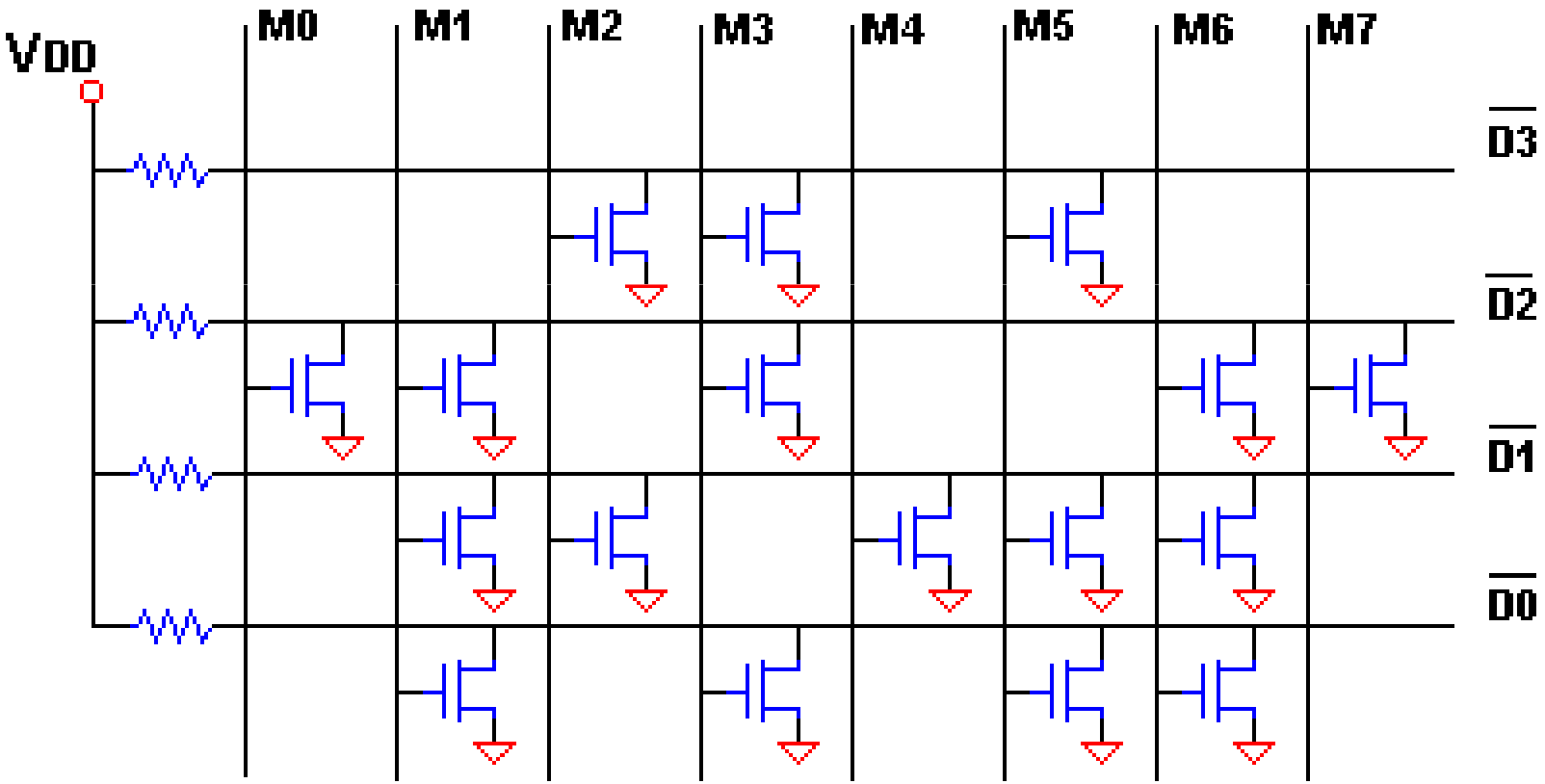
2.1 Memoria ROM: Read Only Memory

- Codificador ROM: Diodos



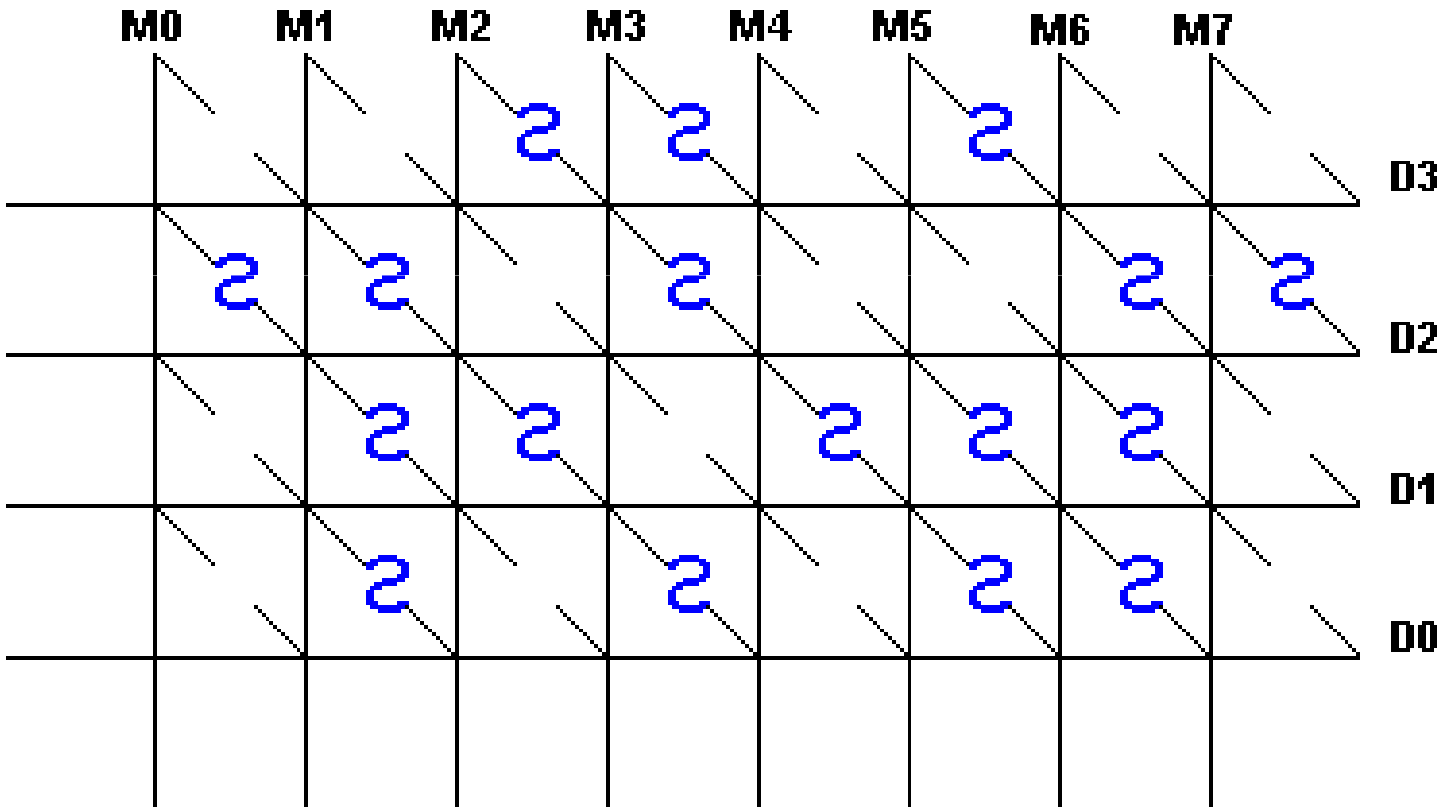
2.1 Memoria ROM: Read Only Memory

- Codificador ROM: Transistores

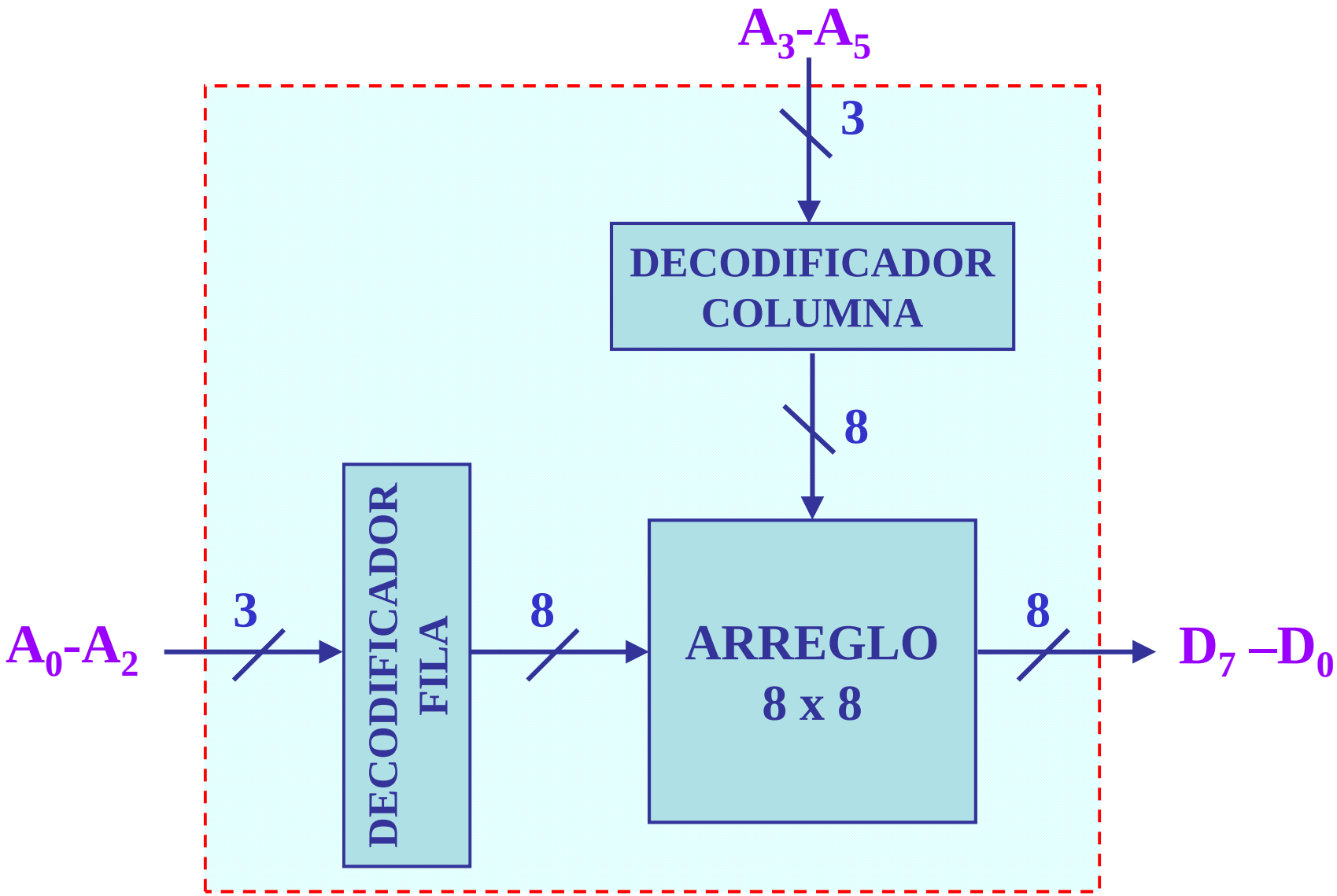


2.1 Memoria ROM: Read Only Memory

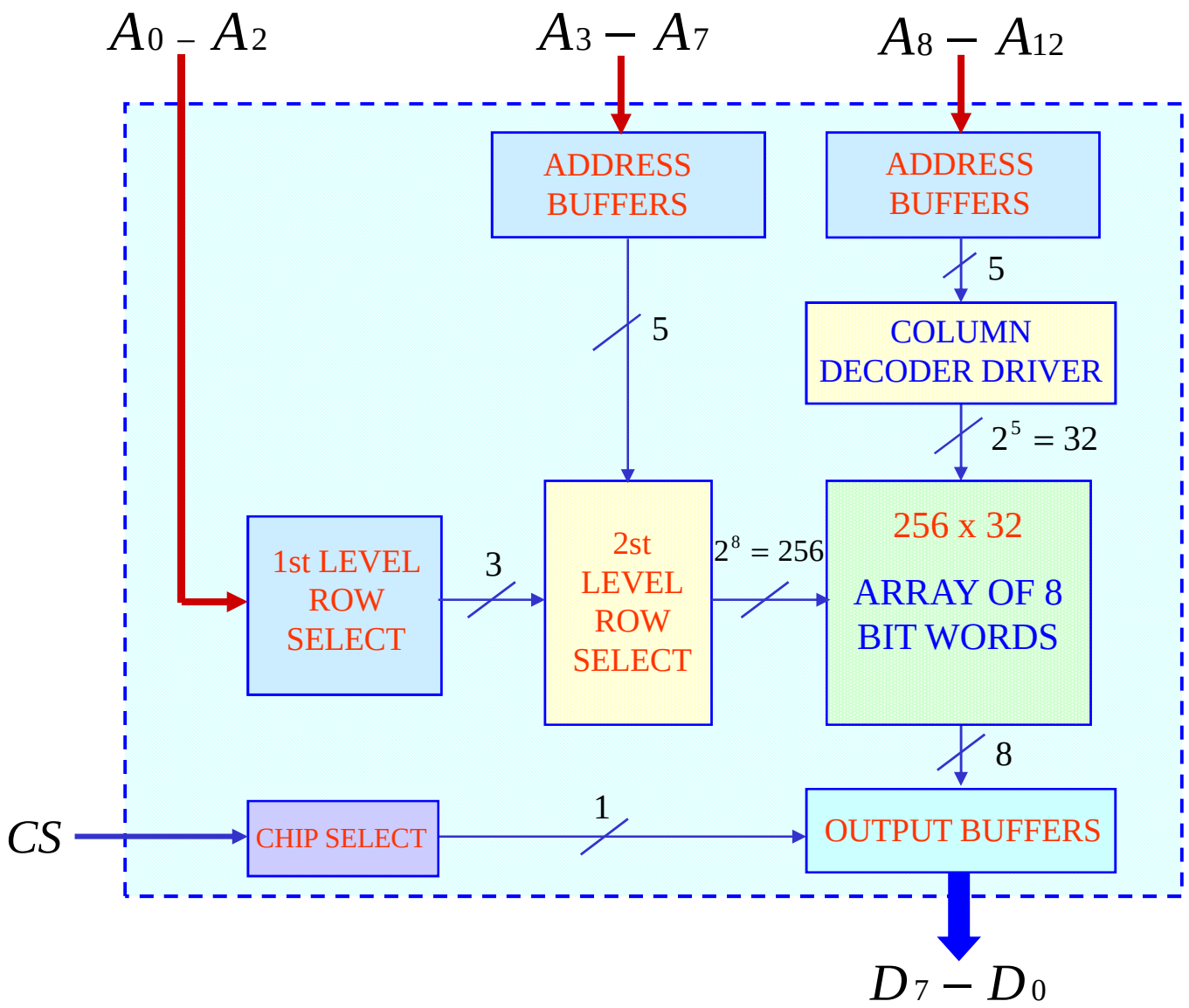
- Codificador ROM: Fusibles



2.1 Diagrama de bloques ROM: 64 x 8 bits

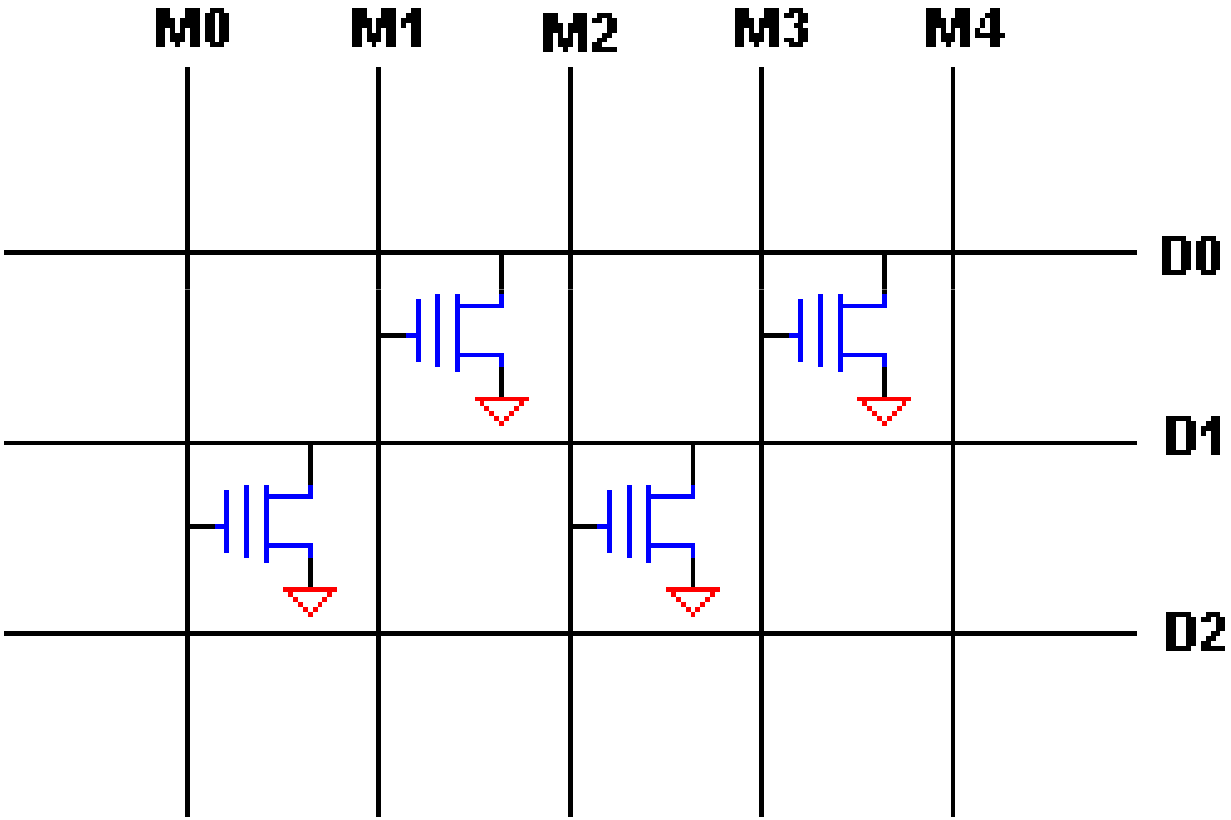


2.1 Diagrama de bloques ROM: 8K x 8 bits

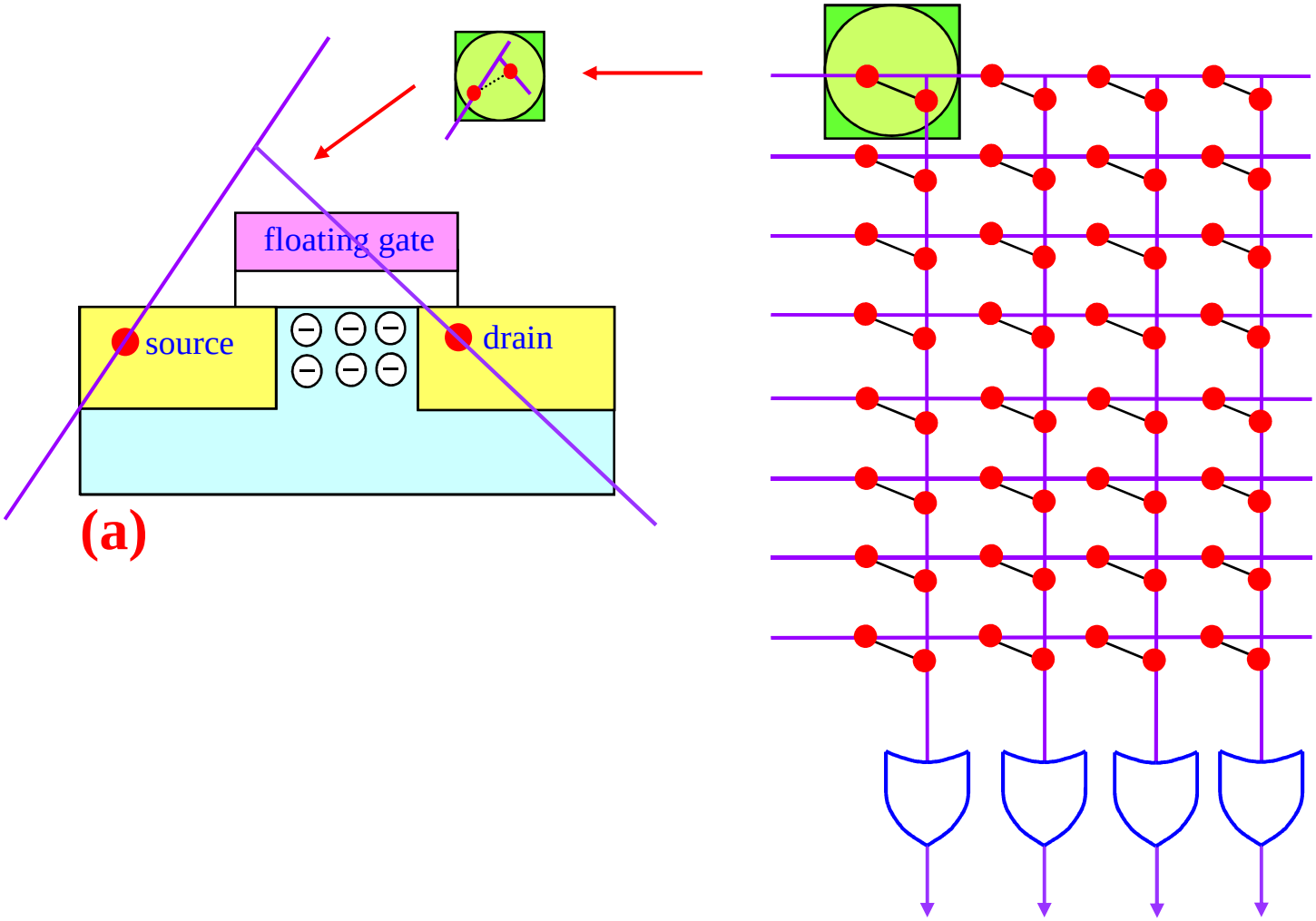


2.2 Memoria EPROM-EEPROM

- Codificador EPROM/EEPROM: Transistor puerta flotante

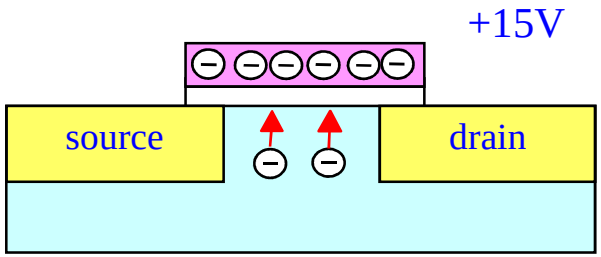


2.2 Memoria EPROM-EEPROM

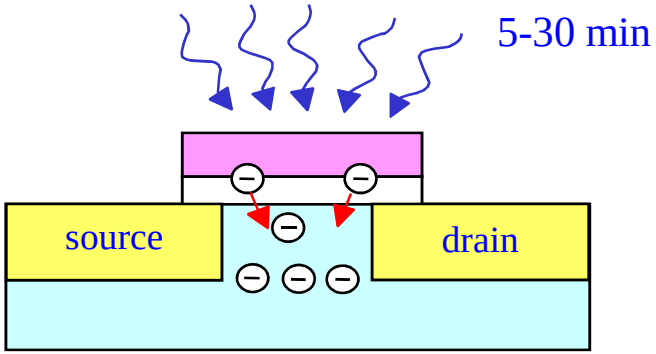


2.2 Memoria EPROM-EEPROM

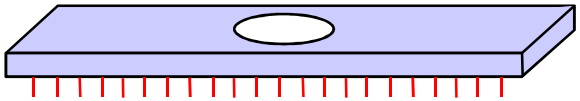
(b)



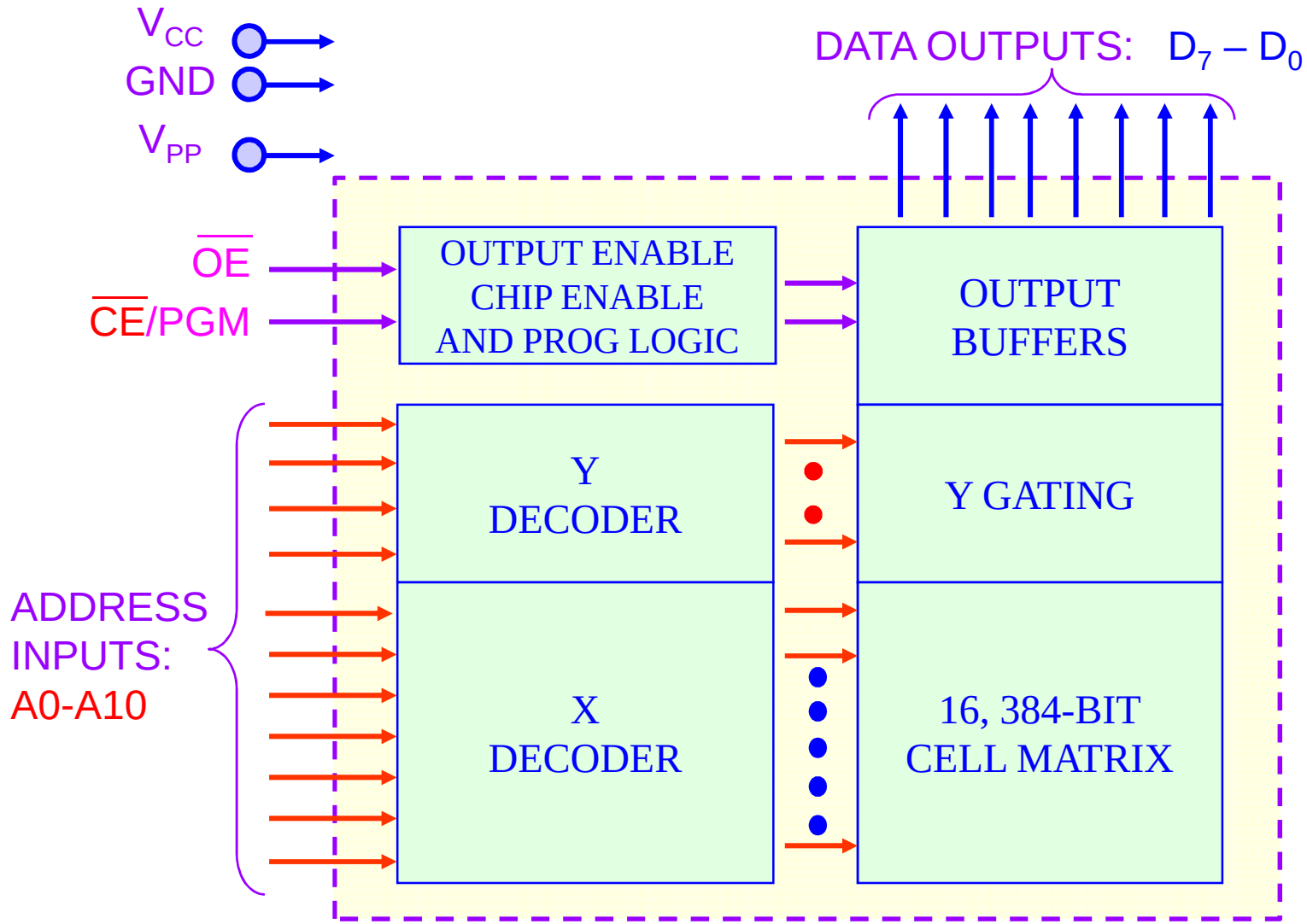
(c)



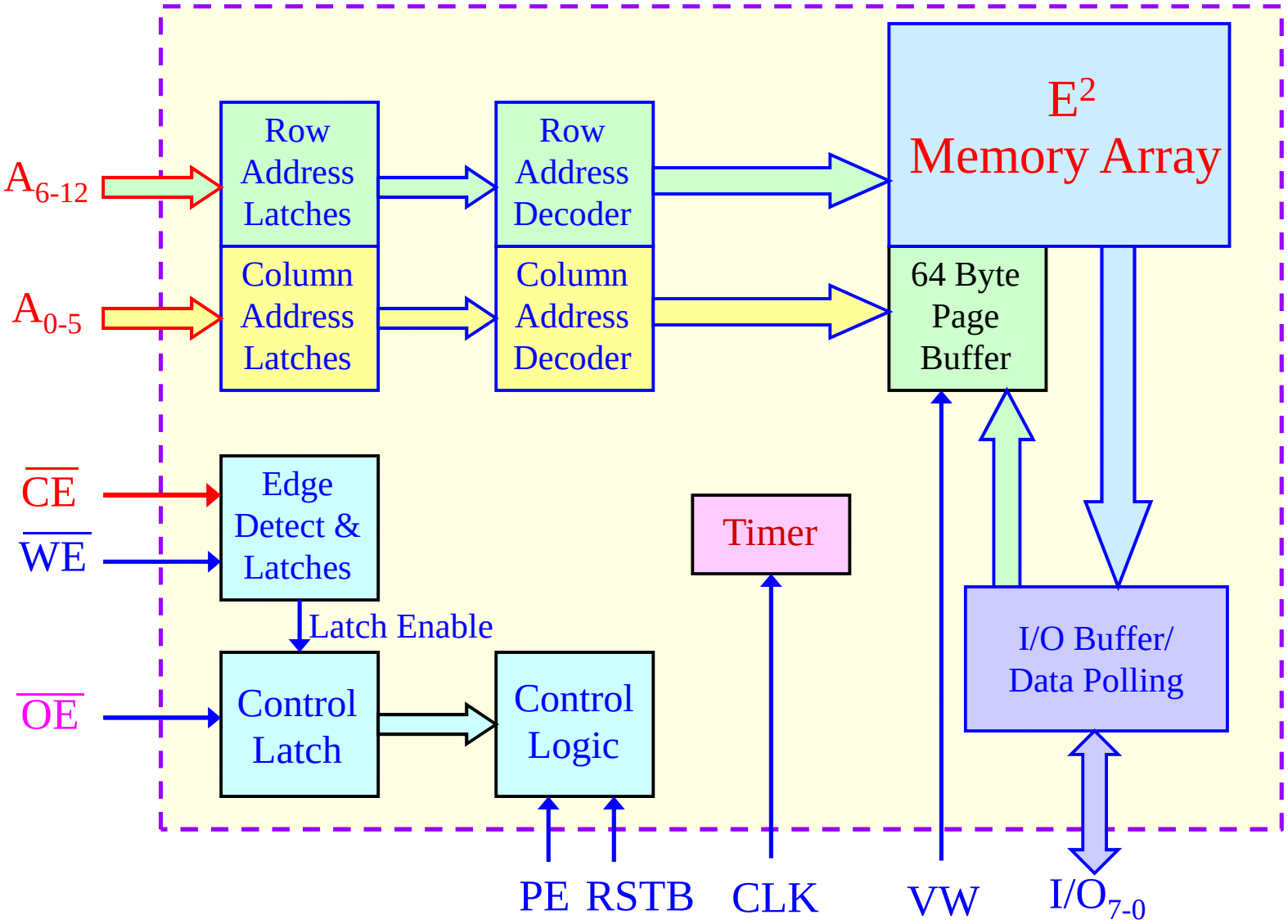
(d)



2.2 2716 EPROM (2k x 8bits)



2.2 W28C64 EEPROM *Simplified Block Diagram*



2.3 Memoria EEPROM Flash

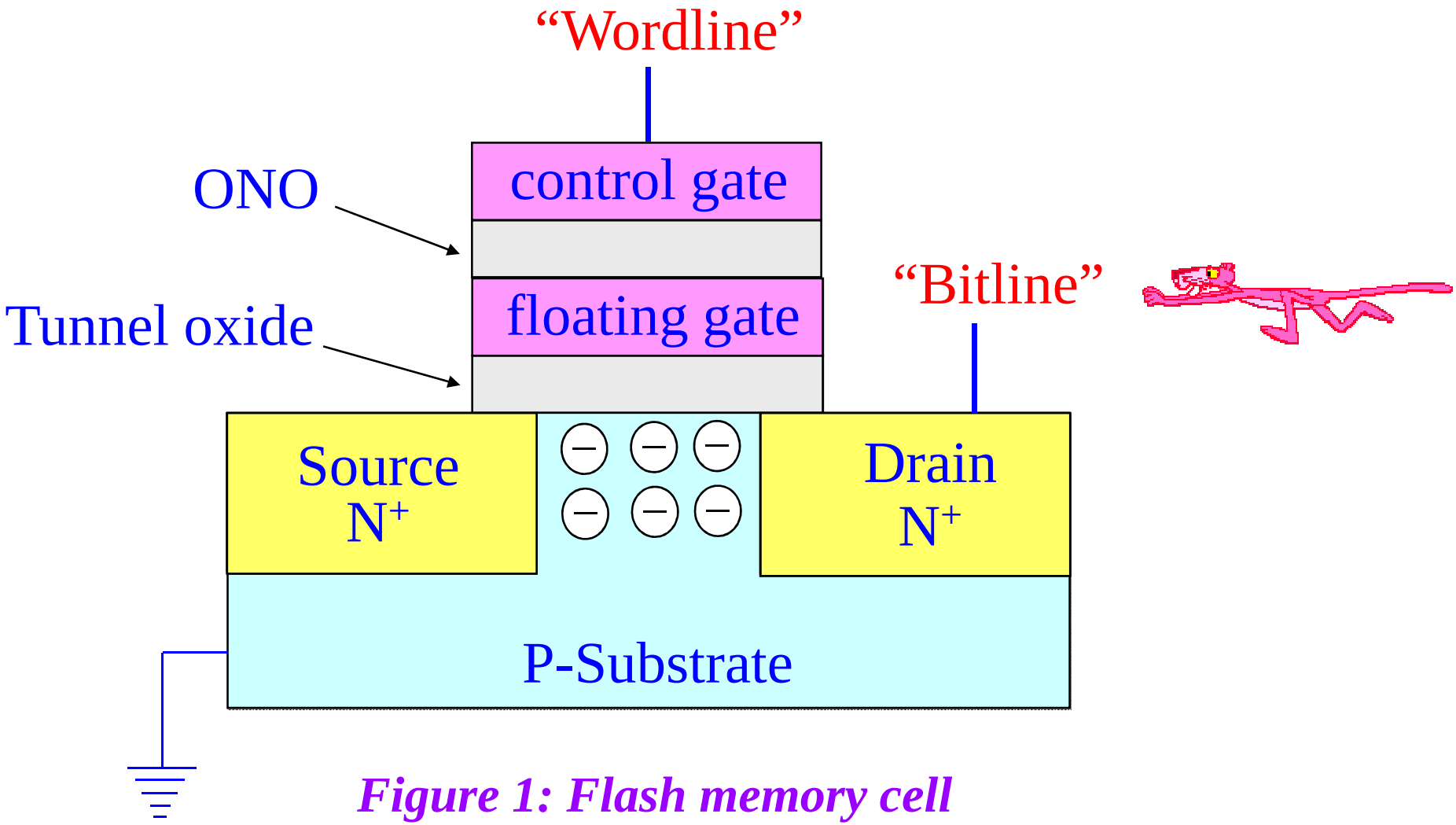
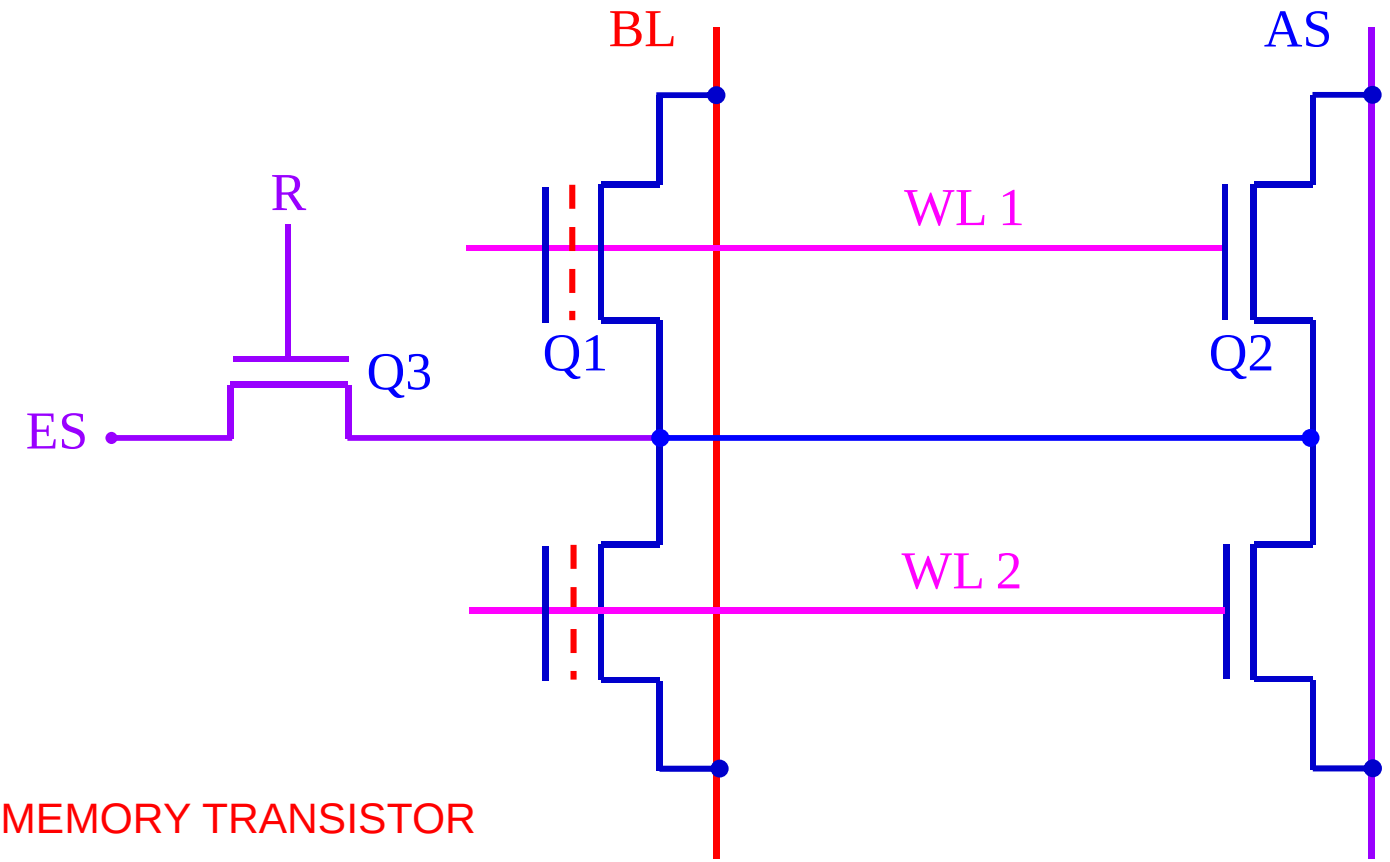


Figure 1: Flash memory cell

2.3 EEPROM Flash memory cell



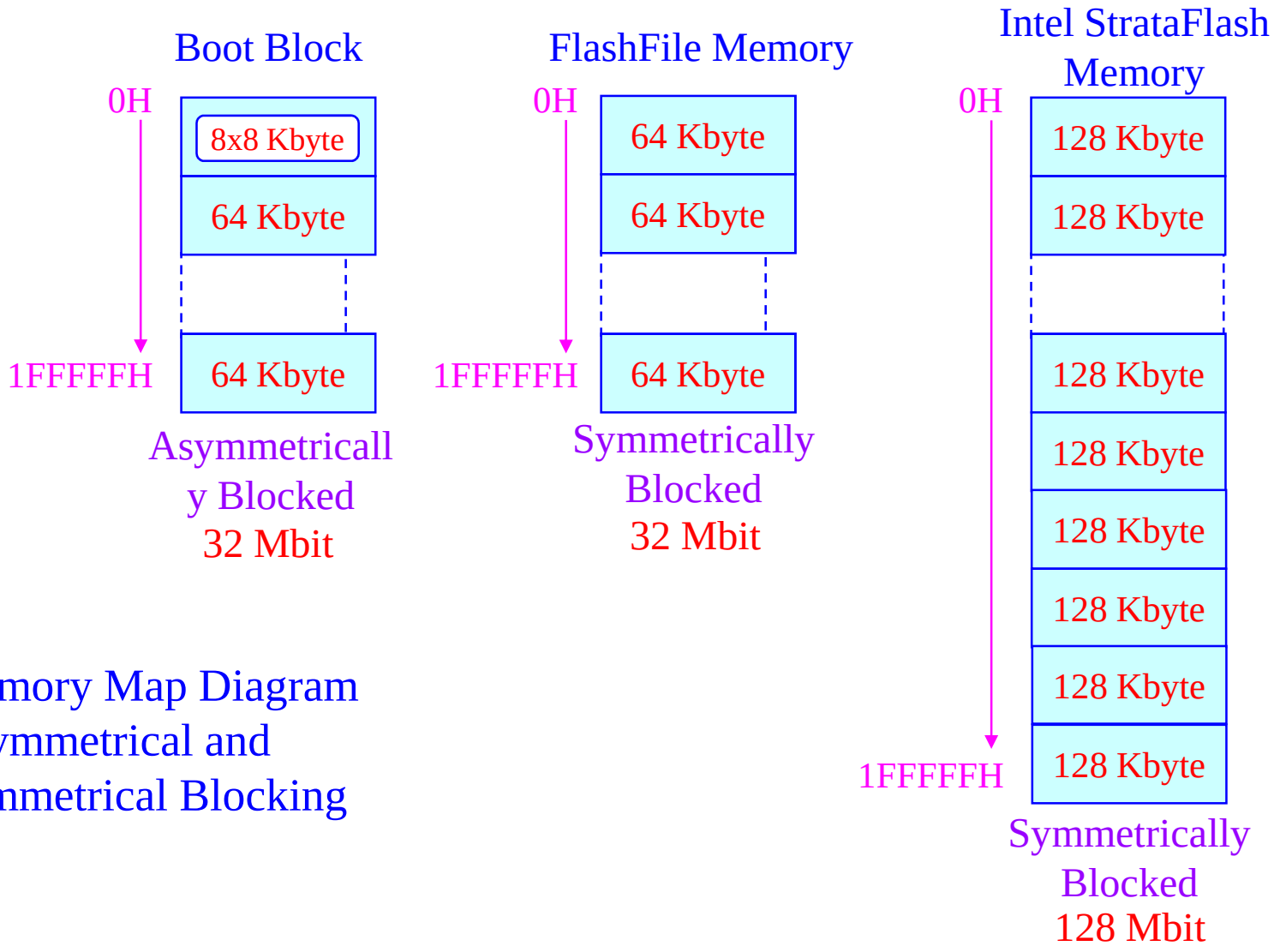
Q1 • MEMORY TRANSISTOR

Q2 • PASS GATE (EACH 16 COLUMNS)

Q3 • SECTOR SELECT (EACH 16 ROWS)

EVERY 16 ROWS • 2K BYTES • 1 SECTOR

2.3 Flash EEPROM: Memory Map Diagram



Memory Map Diagram
Asymmetrical and
Symmetrical Blocking

3. RAM: Random Access Memory

- "Random Access" (Read-Write) Memory
- RAM used for storage of transient objects
  - global and static variables in C
  - the stack (auto variables)
  - the heap (unallocated memory)
- RAM types
  - SRAM (Static RAM)
  - DRAM (Dynamic RAM)

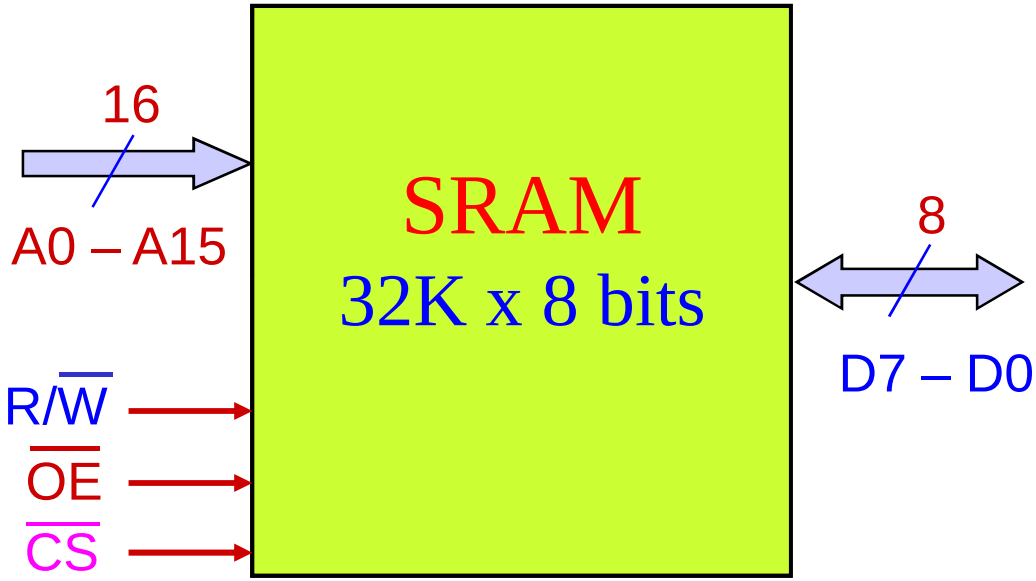
3.1 SRAM: Static RAM

- *Bit stored as the state of a flip-flop*
- Selected by AND-ed inputs: address + CS + OE + R/W
- Retains contents as long as power applied - no refresh
- Access time 12 ns to 200 ns
- Faster, but less dense than DRAM
- NVRAM: usually battery backed CMOS SRAM

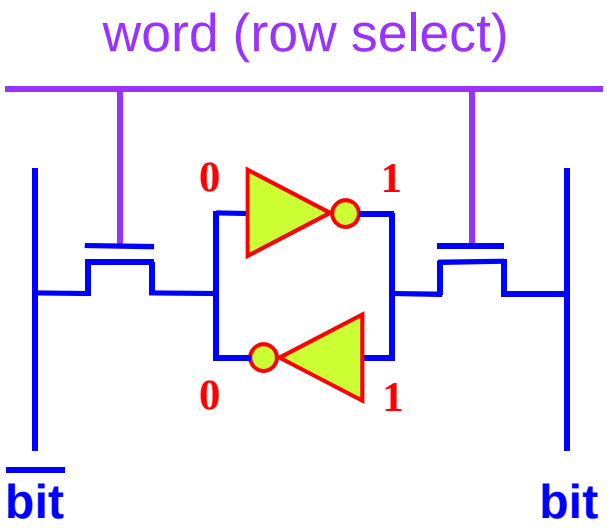
3.1 SRAM: Static RAM

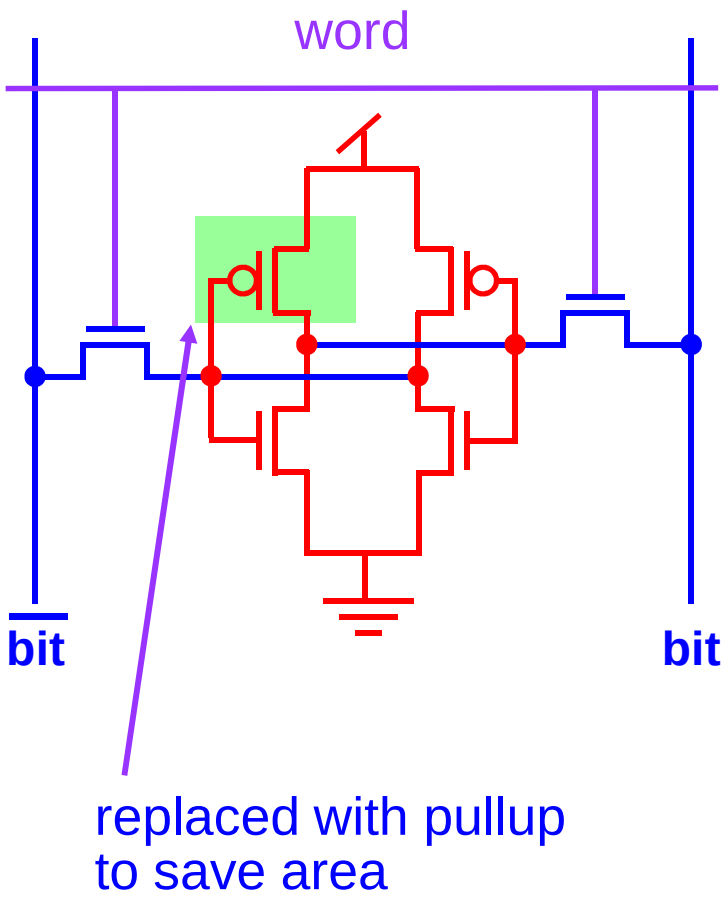
- Static RAM technology
  - Highest speed
    - < 10 ns
  - Density
    - ~ 1 Mbits / chip (Lower for fastest devices)
  - Packaging
    - **128k x 8 bit, 64k x 16 bit**
  - Access protocol
    - Simple:
      - Read: assert address (+ CE, R, OE), then read data
      - Write: assert address and data + CE, W
  - Pins
    - 17bits address (128k) + 8 data = 25 + control

3.1 SRAM 32k x 8bits

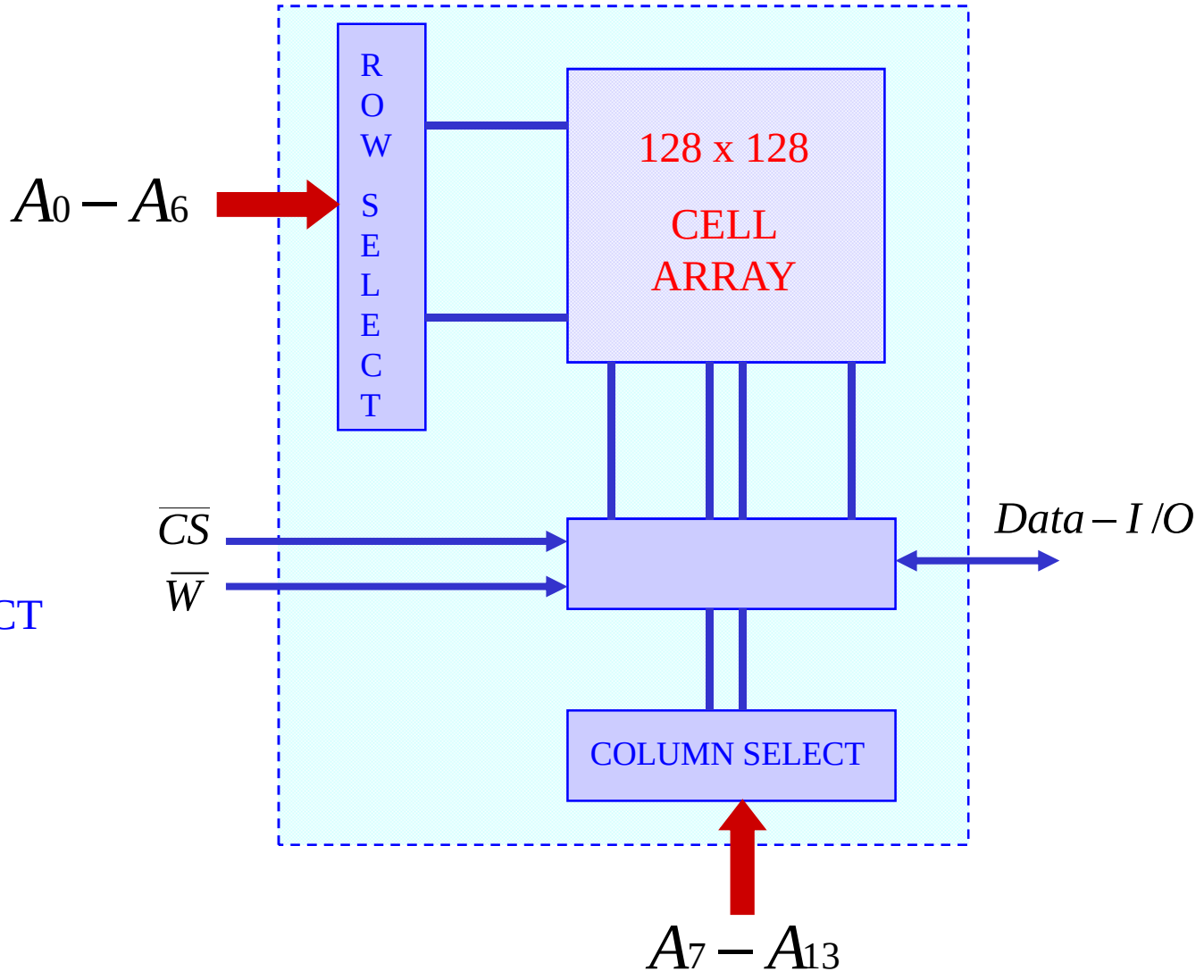


3.1 SRAM Cell: 6-Transistors



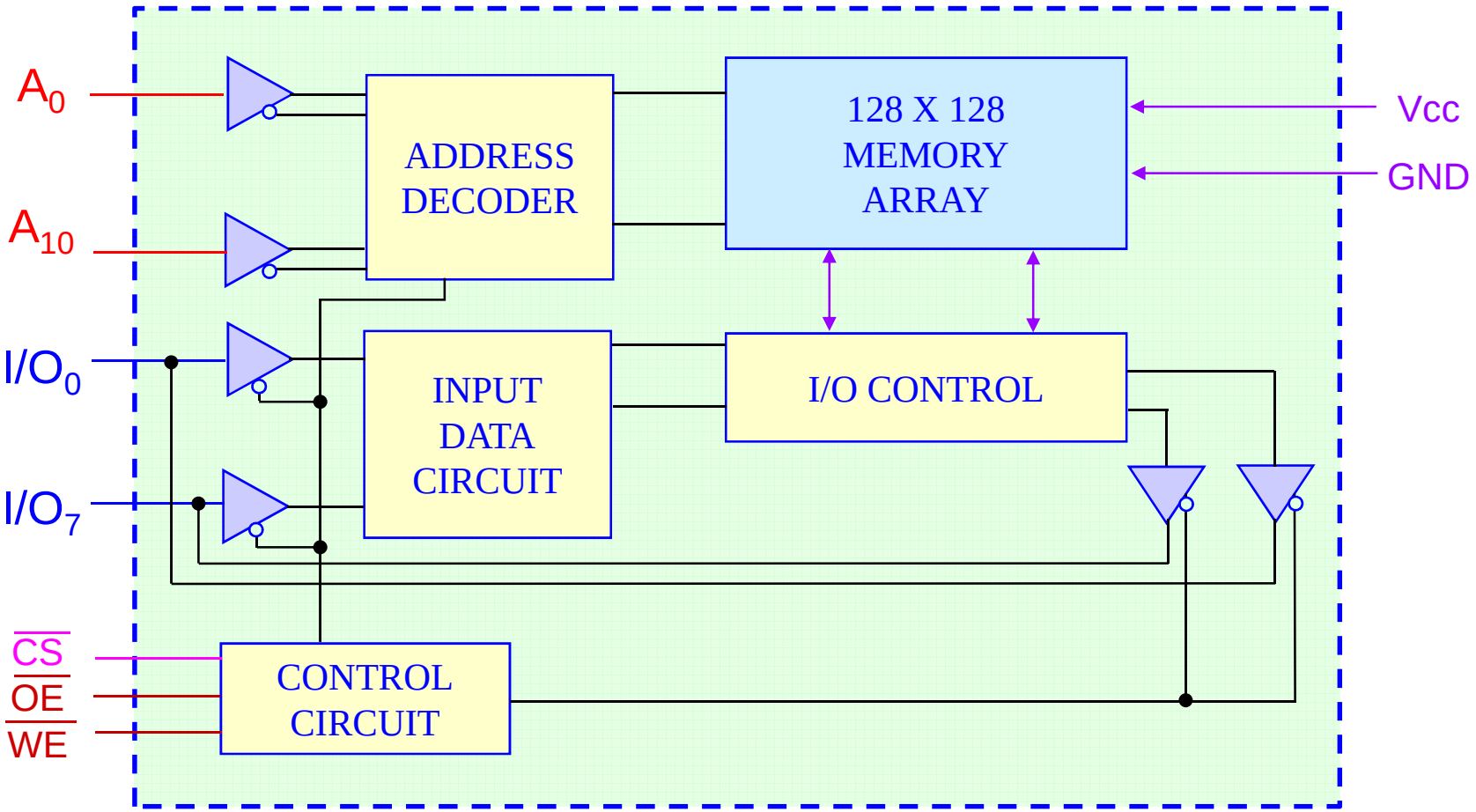


3.1 Diagrama de bloques SRAM: 16K x 1 bits



- $\overline{CS}$: CHIP SELECT
- $\overline{W}$ = 1: READ
- $\overline{W}$ = 0; WRITE
- DI: DATA IN
- DO: DATA OUT

3.1 6264 SRAM (8k x 8bits)

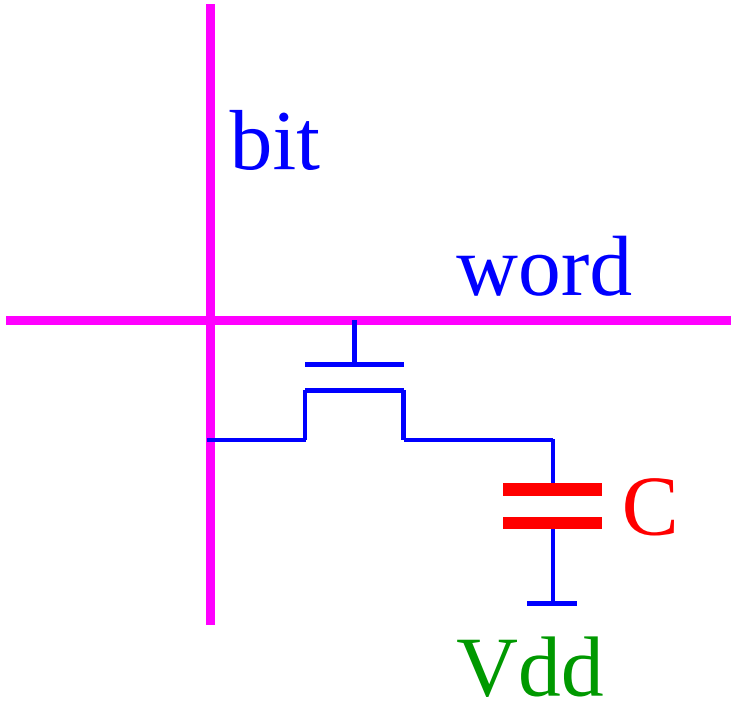


3.2 DRAM: Dynamic RAM

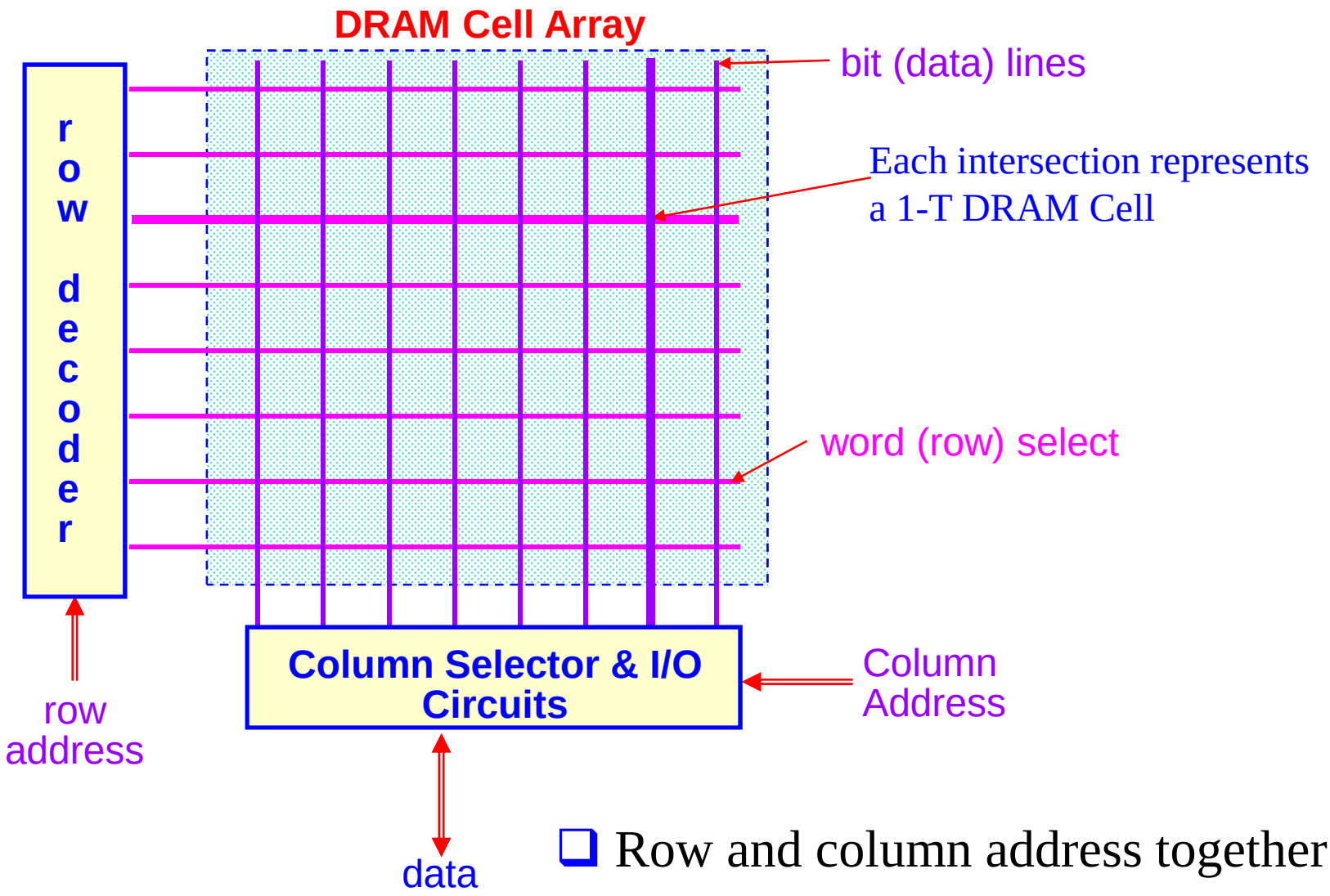
- Each bit stored as a charge on a storage capacitor
- Must be "refreshed" (write accessed) periodically - at intervals typically less than 8 ms
- Many μPs, all SIMMS have built-in refresh
- Less power, denser, but slower than SRAM

3.2 Dynamic RAM

- Refresh
- DRAM has one-transistor cells
- Charge leaks from storage capacitor
- Refresh interval ~ 1-4 ms
- RAM unavailable during refresh!
  - Some bandwidth loss



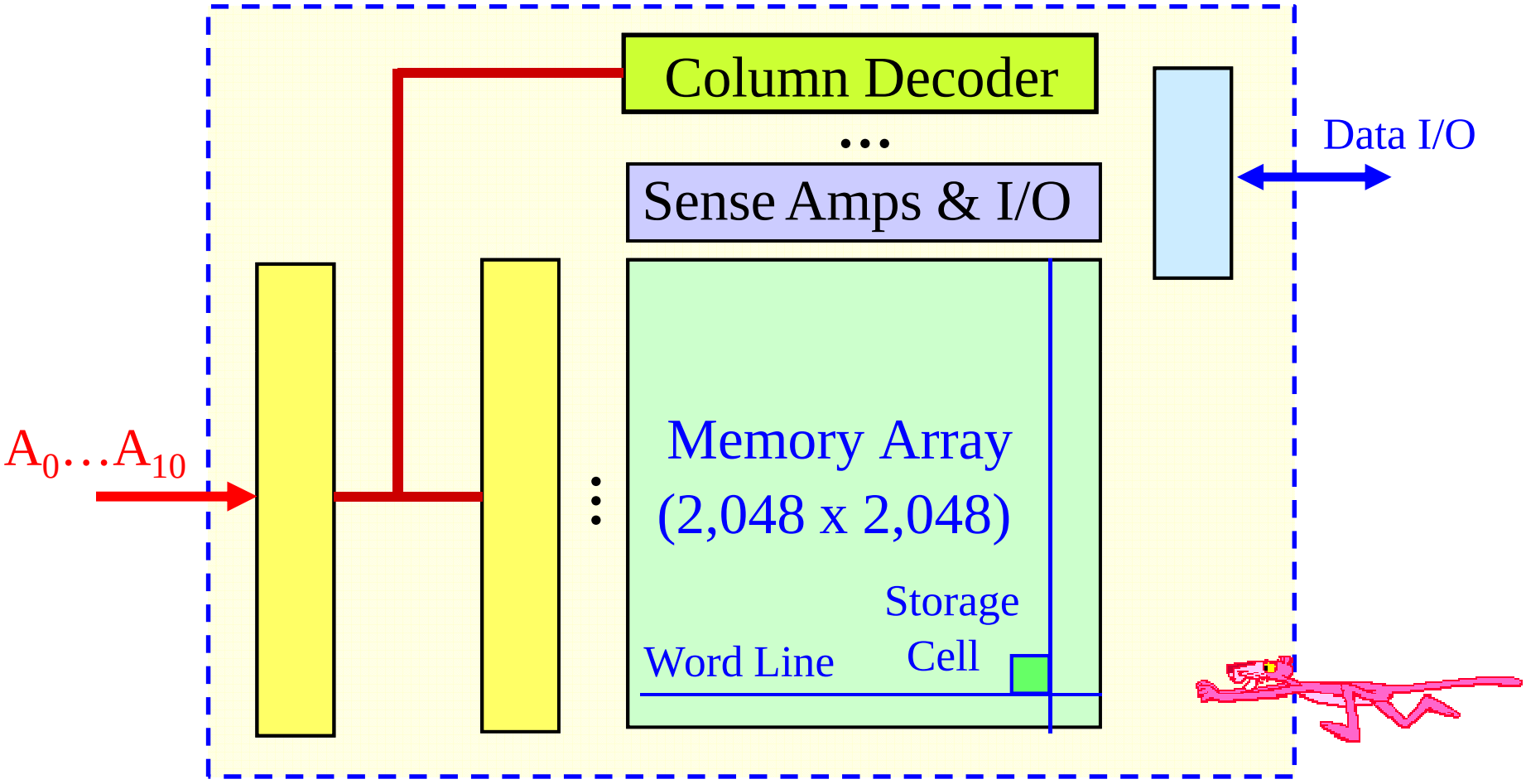
3.2 Classical DRAM Organization (Square)



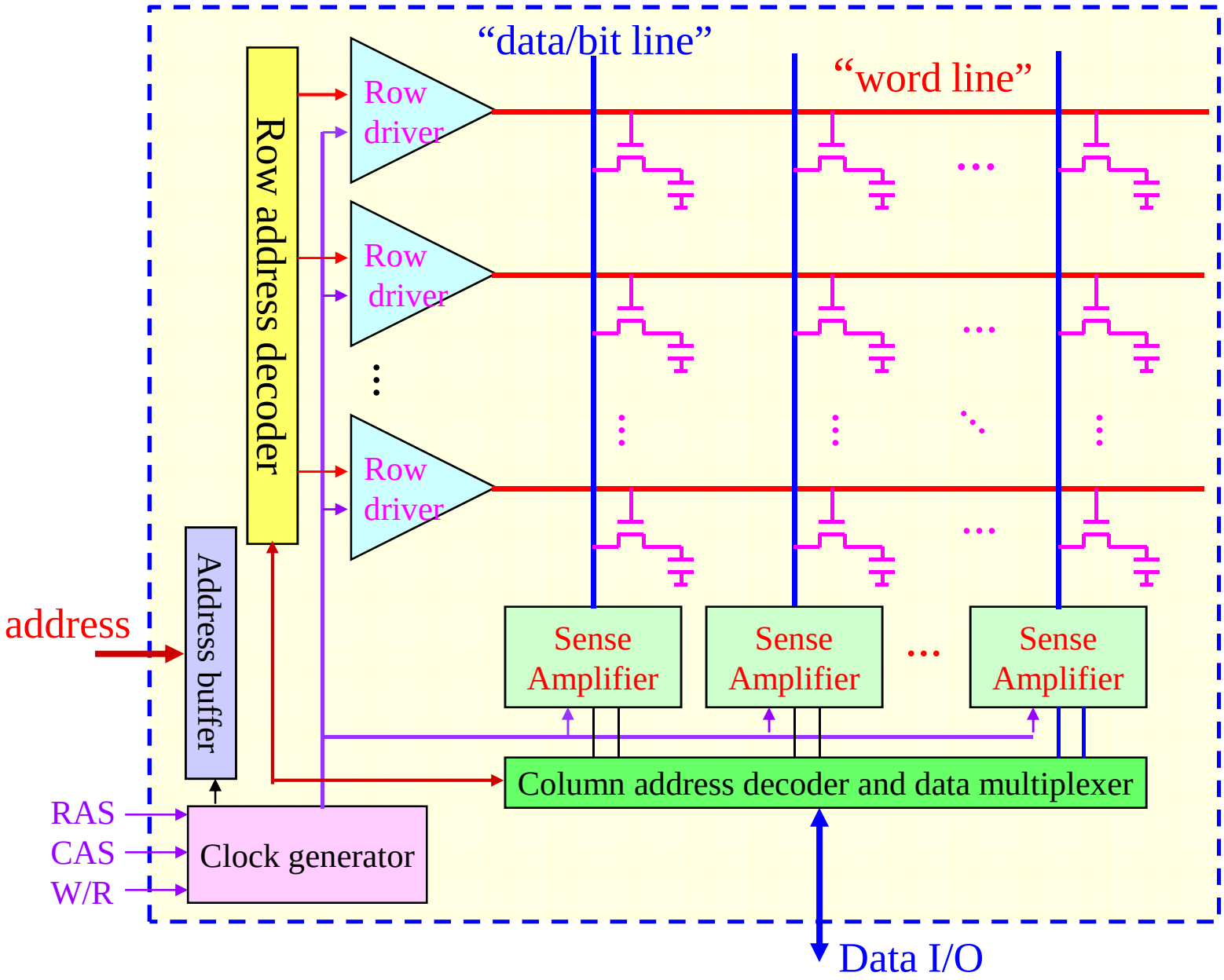
- Row and column address together:
  - Select 1 bit a time

3.2 DRAM Block Diagram: 4 Mbits

- ***Square root of bits per RAS/CAS***

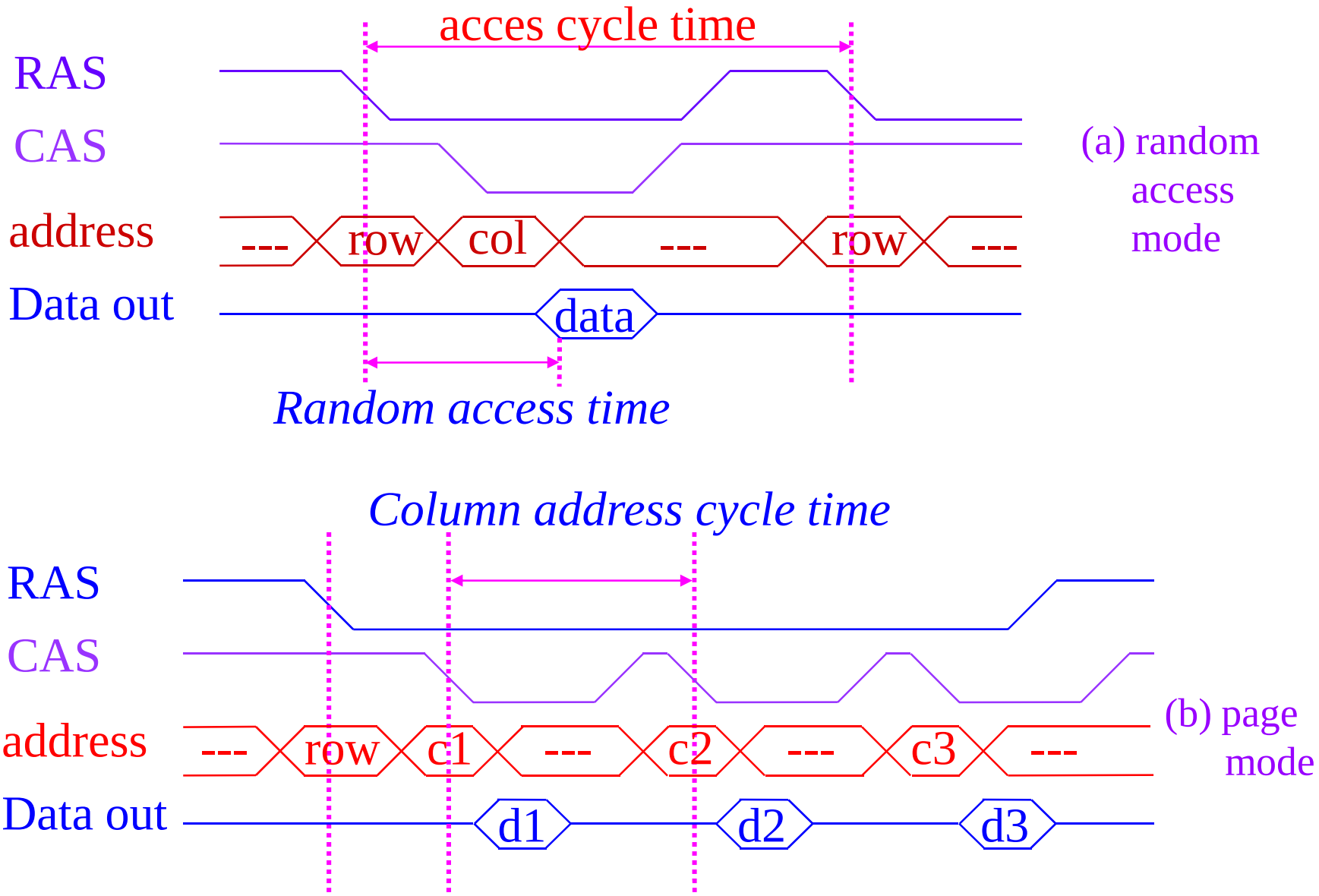


3.2 DRAM Memory Array

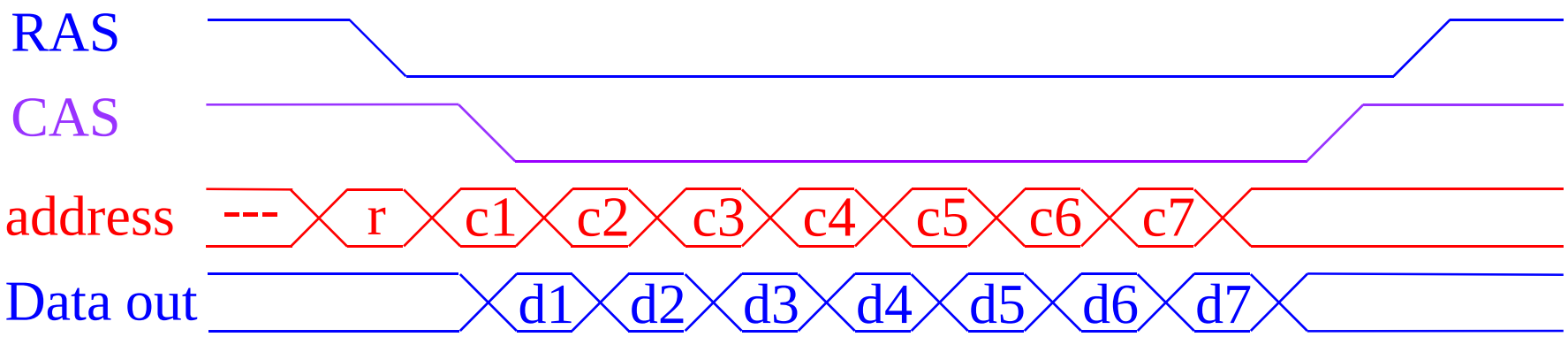


- precharge BL
- decode row
- discharge BL
- sense amp
- column select
- data > out

3.2 DRAM Memory



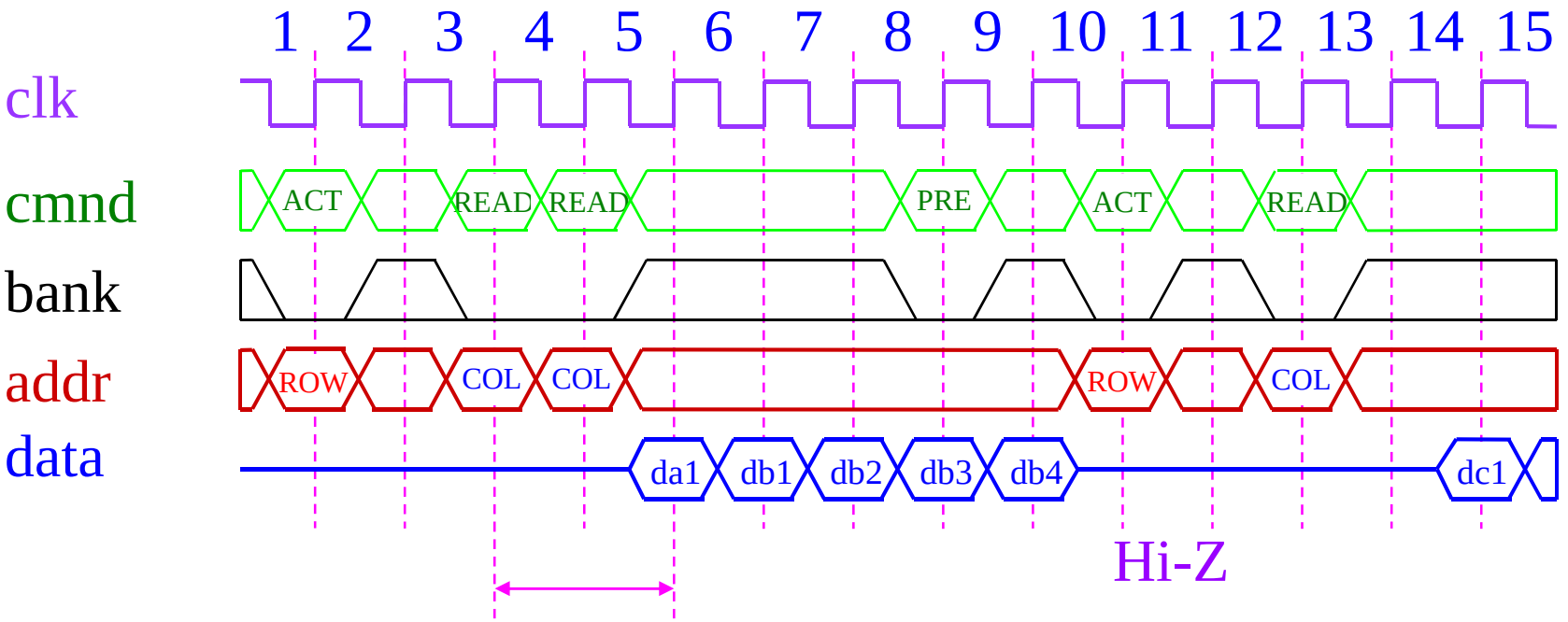
3.2 DRAM Memory



(c) static column mode

3.2 Synchronous DRAM: SDRAM

- introduce a clock (asynchronous => synchronous)
- pipelining



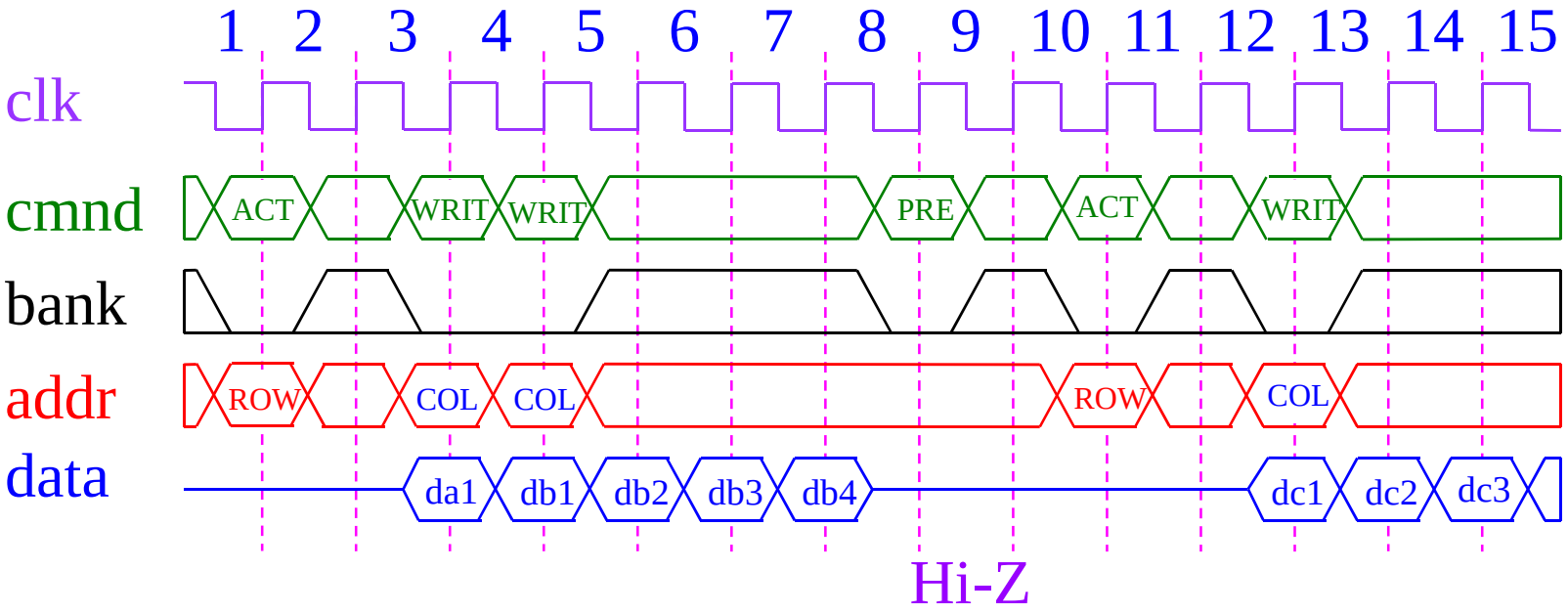
CAS latency

1/2/3 for 33/66/99 MHz

Burst length =

1/2/4/8/full page

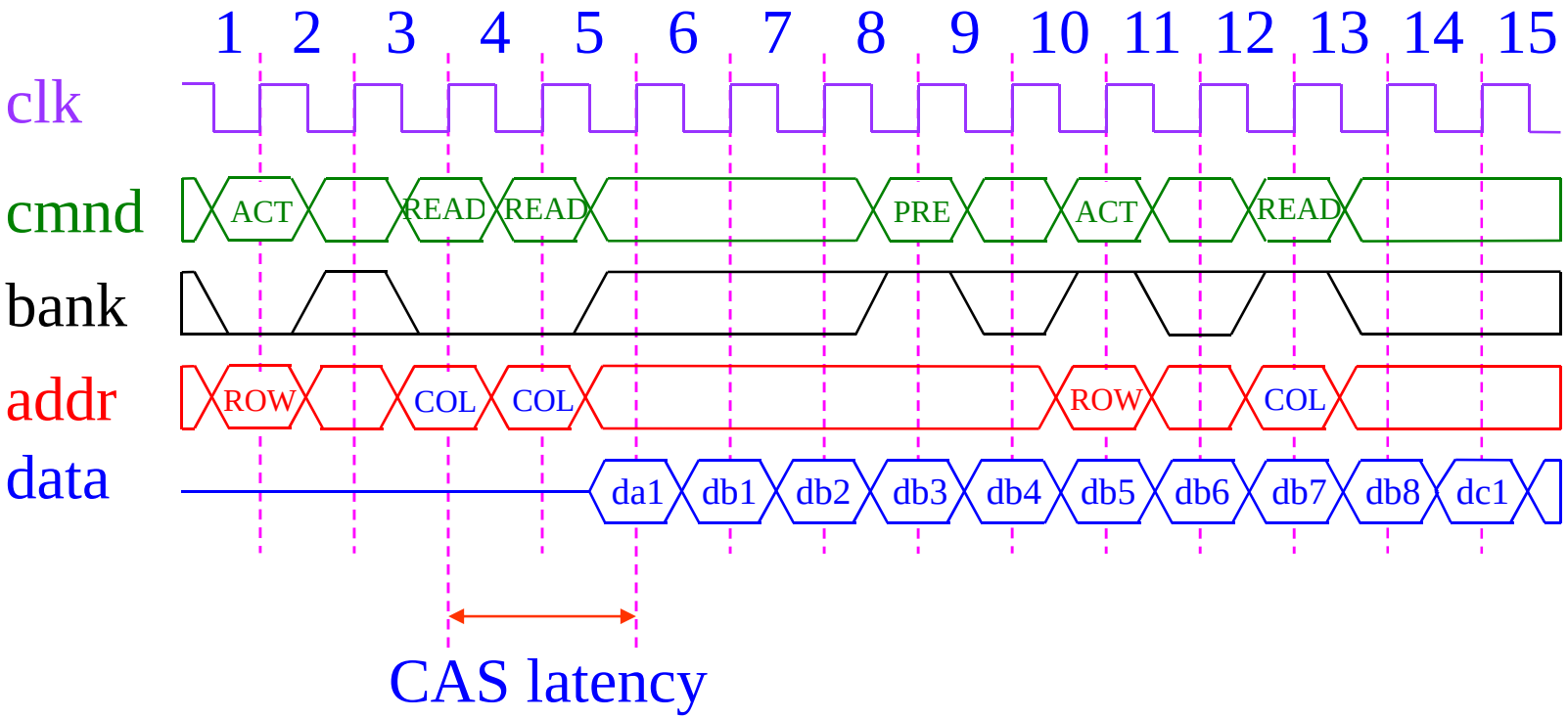
3.2 Synchronous DRAM: SDRAM



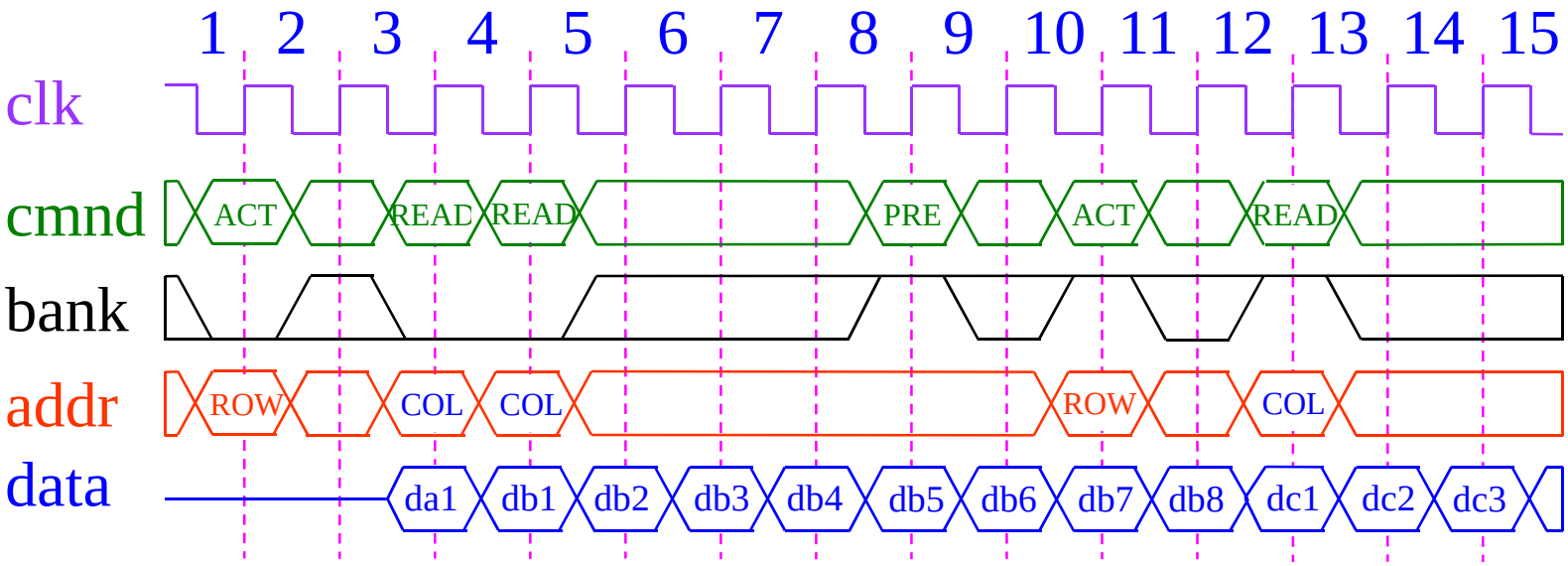
Final goal = 1 access each clock cycle

- banking
- burst length large enough (e.g. 8)

3.2 Synchronous DRAM: SDRAM

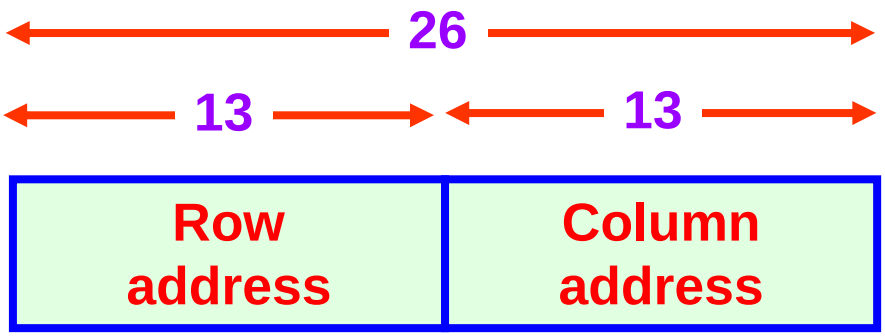


3.2 Synchronous DRAM: SDRAM



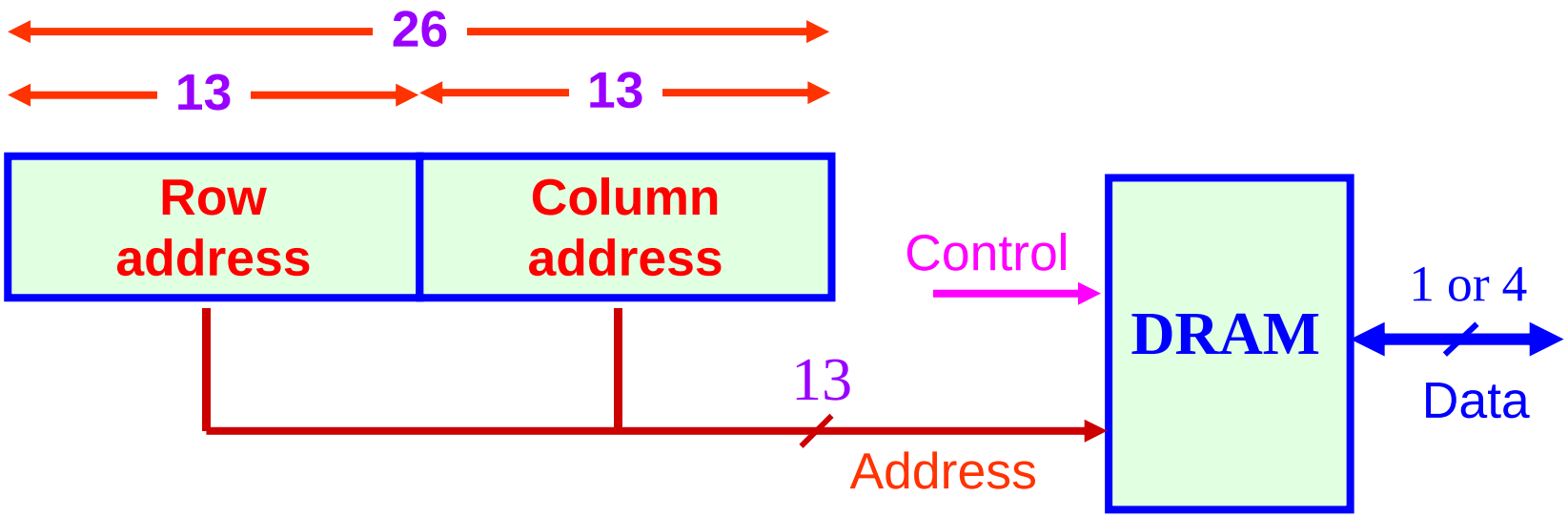
3.2 DRAM Memory

- Dynamic RAM technology
  - highest density
    - 64-256 Mbits / chip
    - semiconductor memories of several Gbytes feasible
  - packaging
    - 64M x 1 bit
    - 16M x 4 bit
  - reduced pin count
    - $64x10^6 = 2^{26}$
    - use 13 pins only
    - Row/Column multiplexing

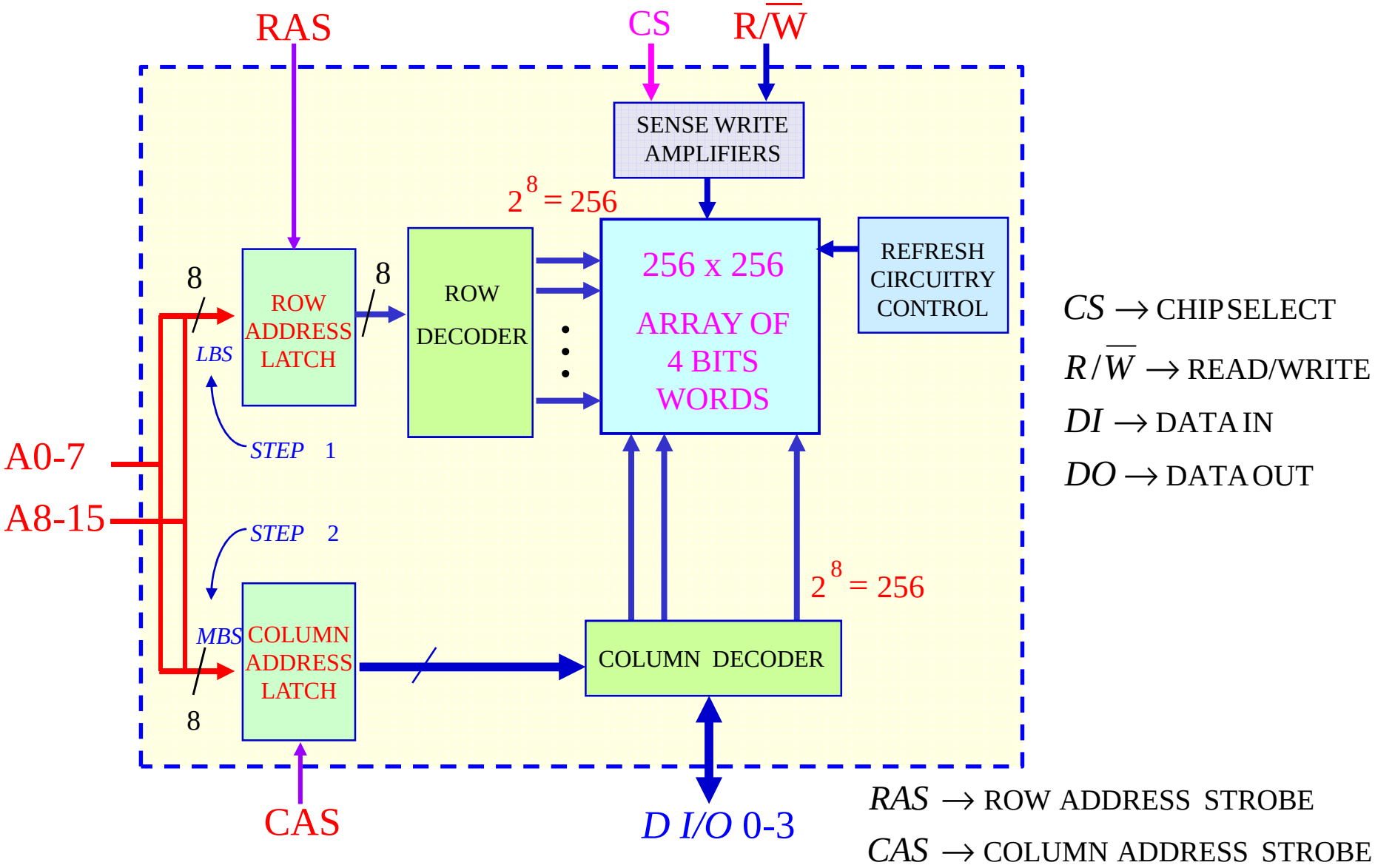


3.2 DRAM Memory

- Dynamic RAM technology
  - Row / Column multiplexing
  - Two “cycles” to access
    - Assert Row address
    - Assert Column address



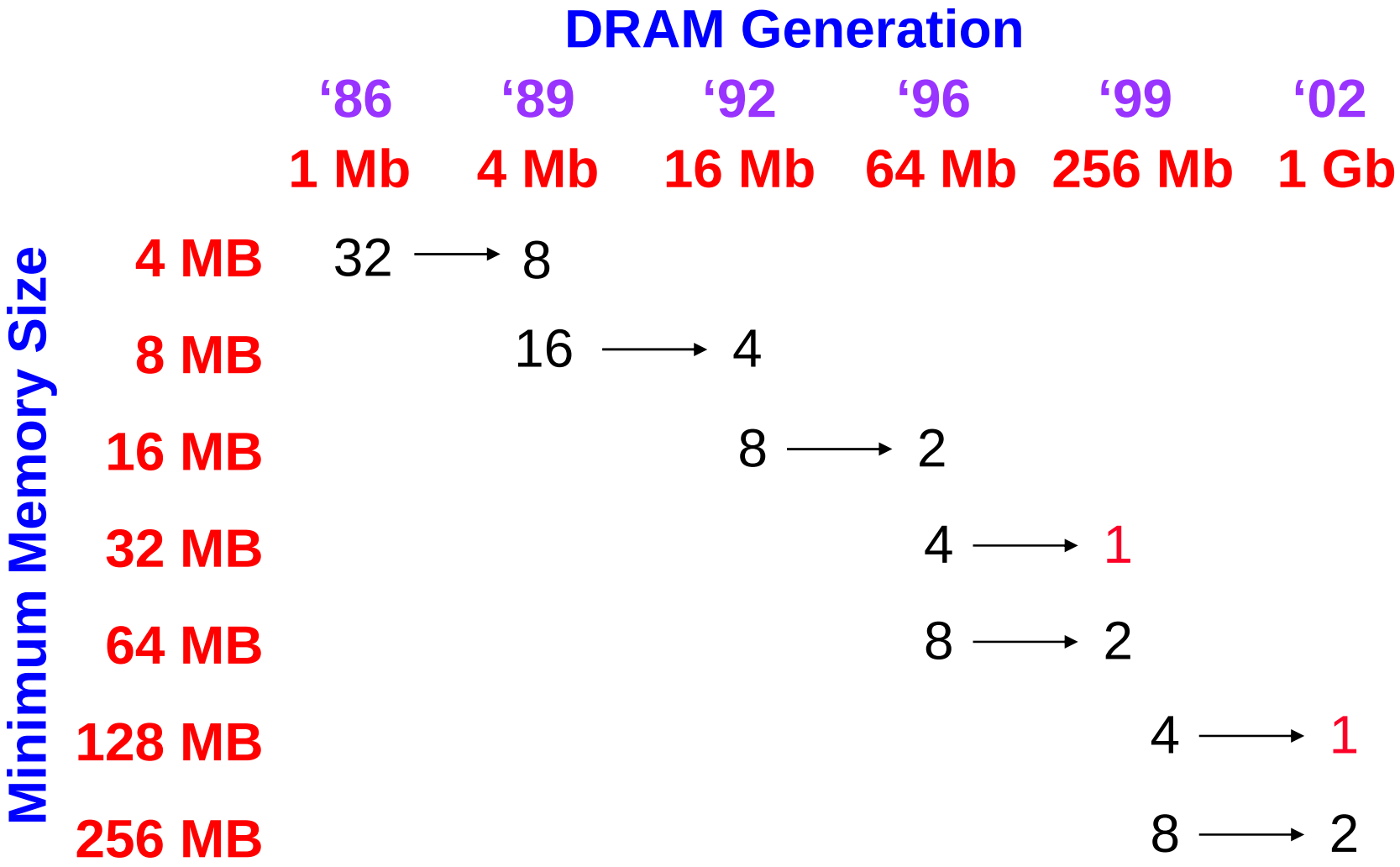
3.2 Diagrama de Bloques: DRAM 64K x 4



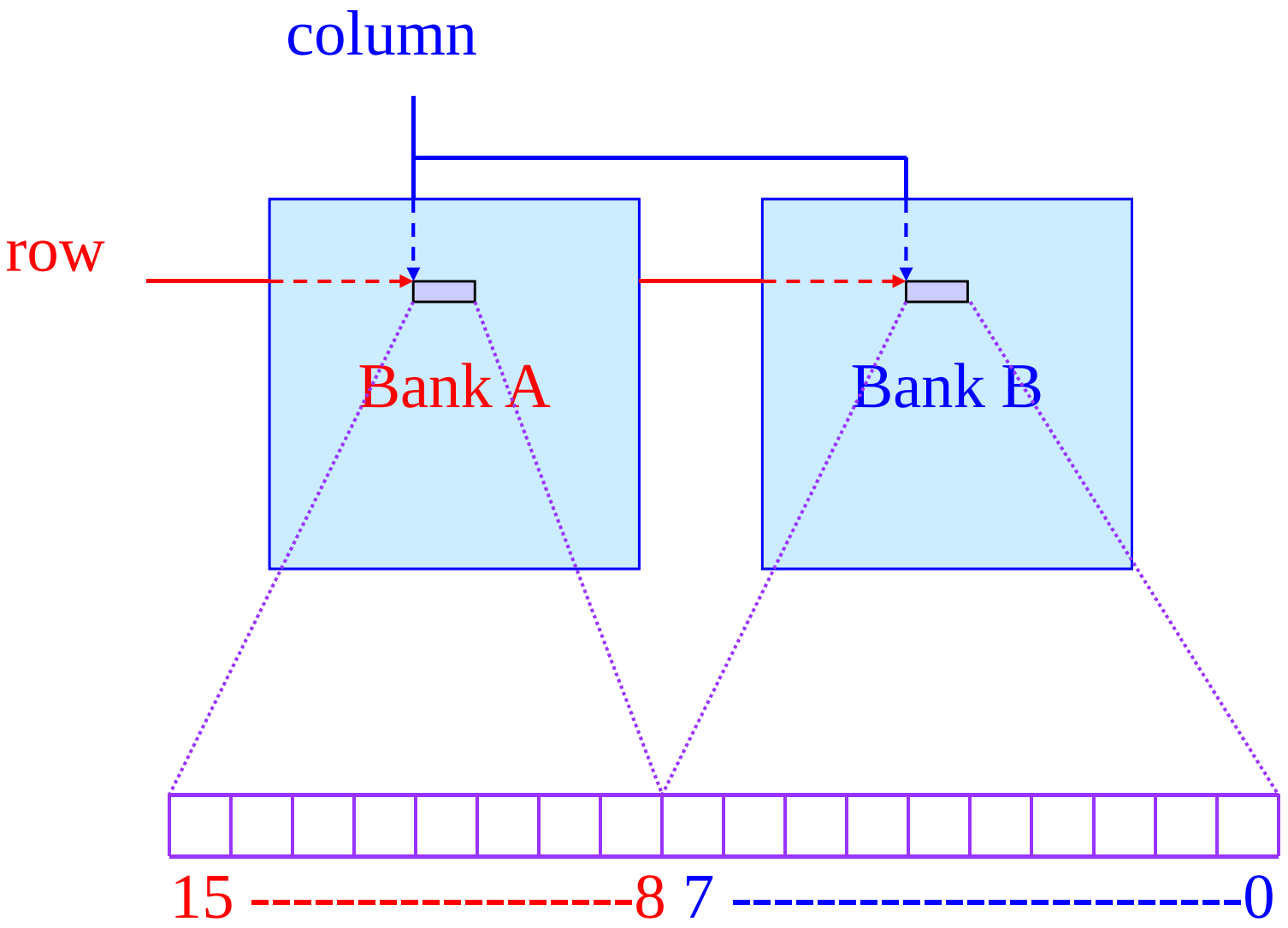
3.2 Advanced DRAM

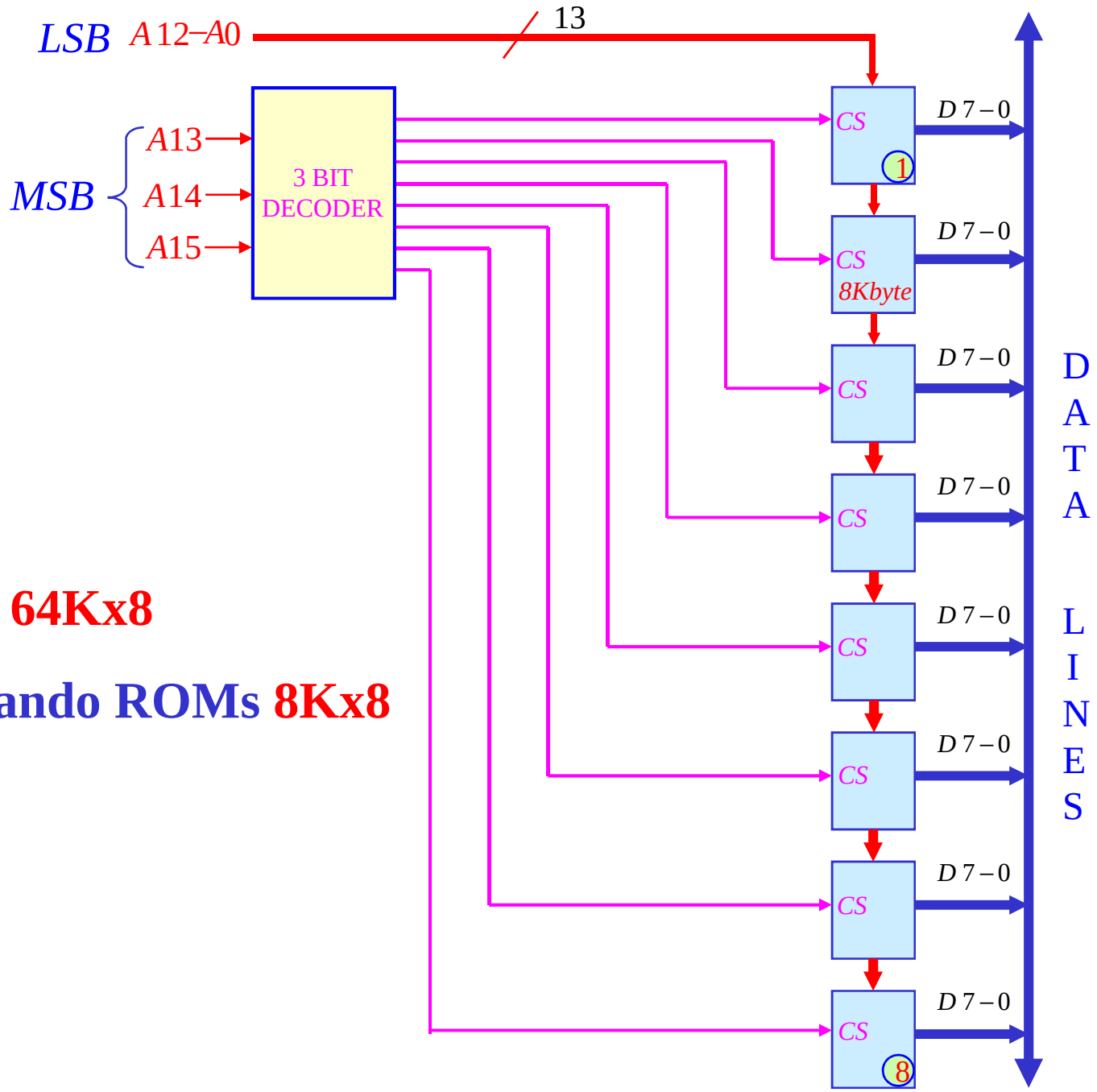
- DRAMs commonly used as main memory in processor based embedded systems
  - high capacity, low cost
- Many variations of DRAMs proposed
  - need to keep pace with processor speeds
  - FPM DRAM: fast page mode DRAM
  - EDO DRAM: extended data out DRAM
  - SDRAM/ESDRAM: synchronous and enhanced synchronous DRAM
  - RDRAM: rambus DRAM

3.2 DRAMs per PC over Time

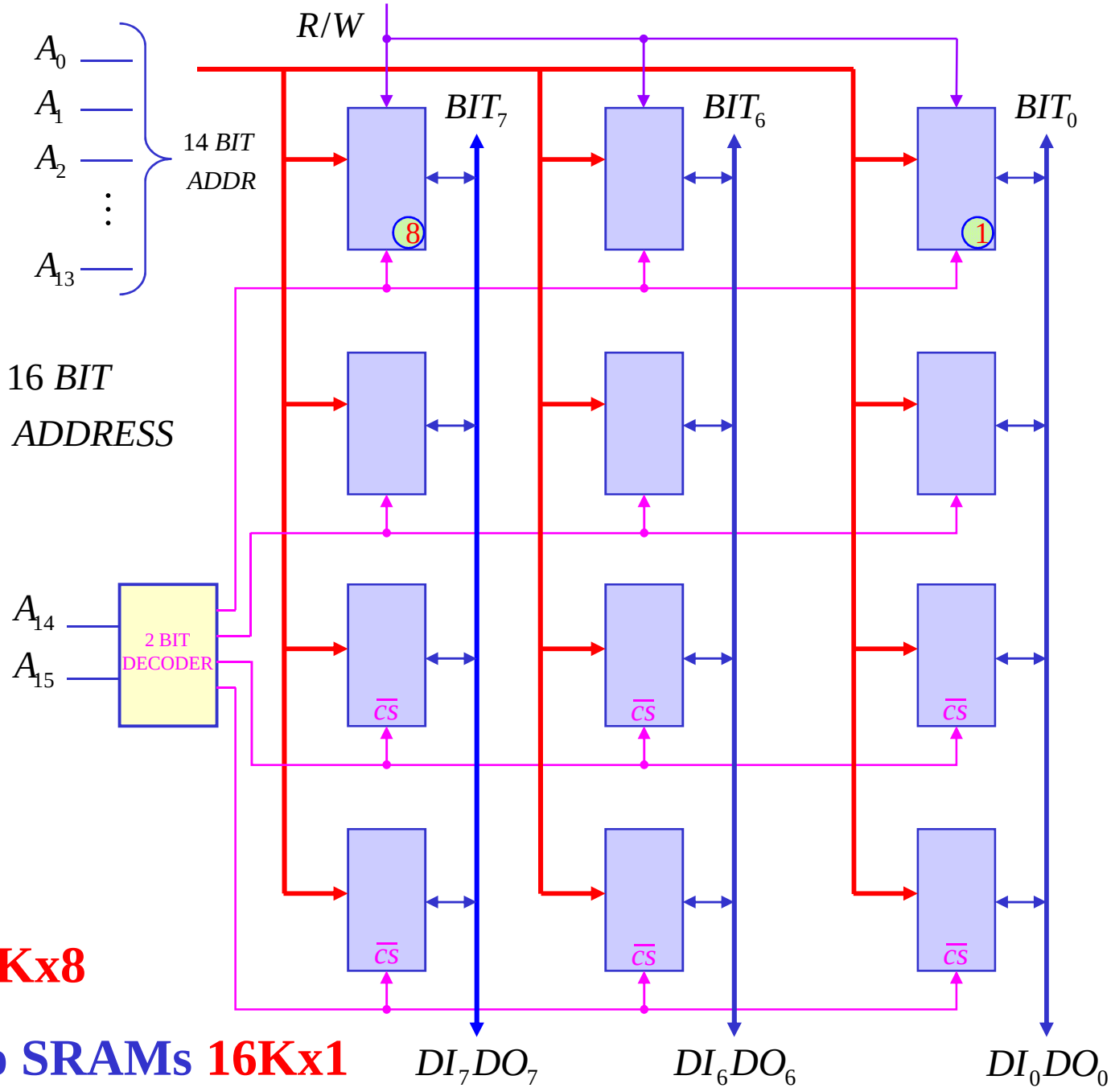


4. Memory Arrays:





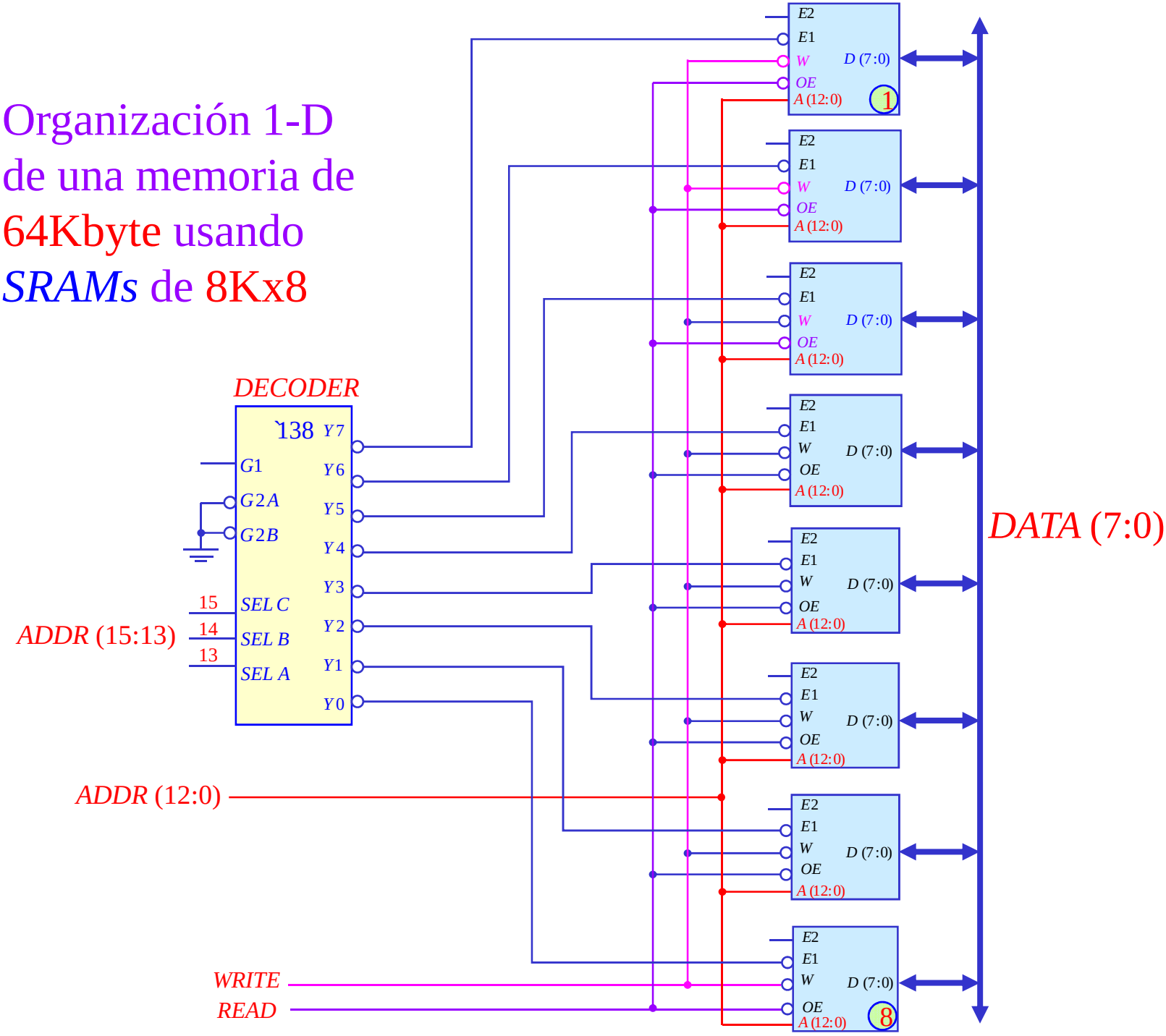
- **ROM 64Kx8**
  - **usando ROMs 8Kx8**



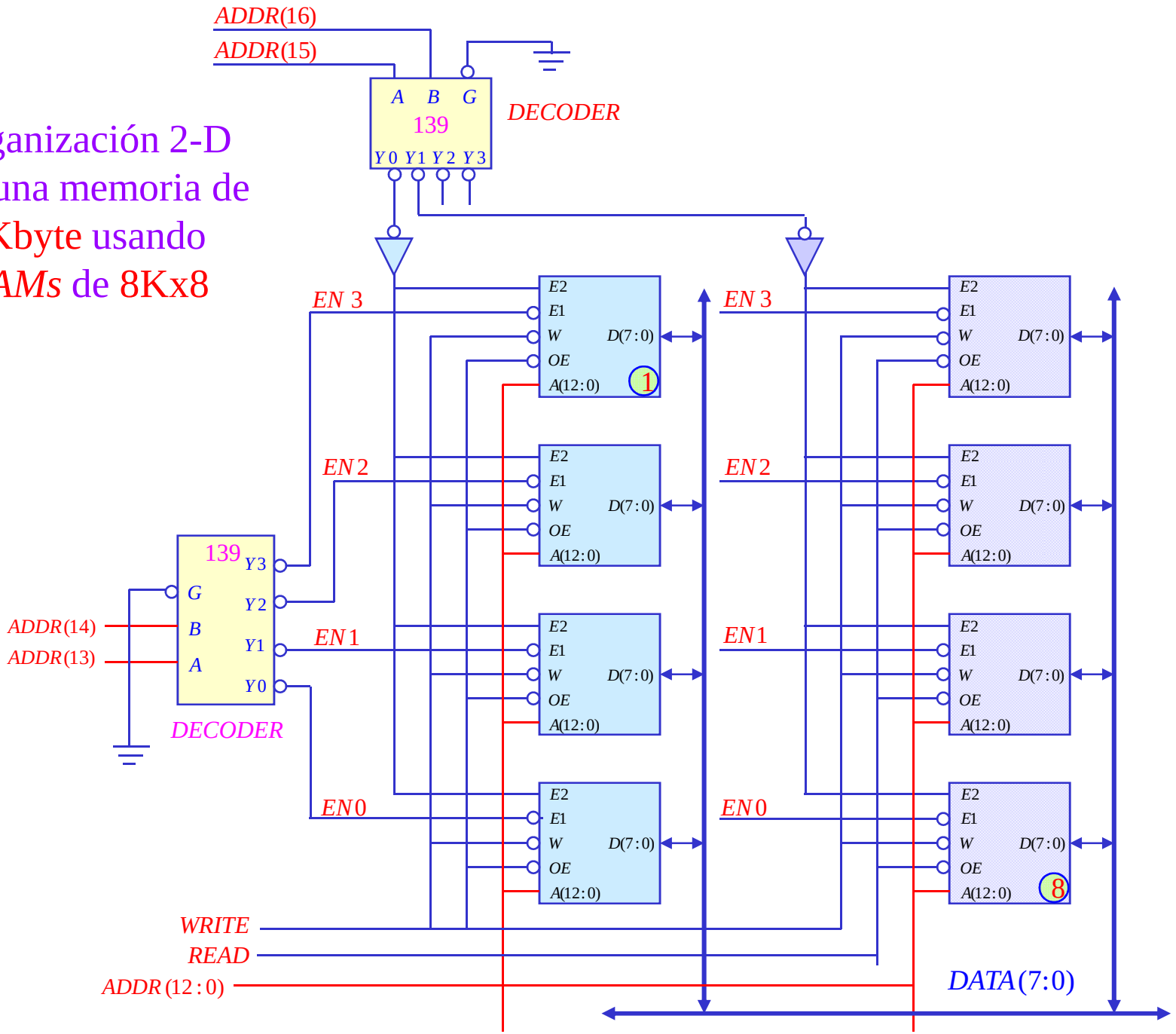
❖ **SRAM 64Kx8**

➢ **usando SRAMs 16Kx1**

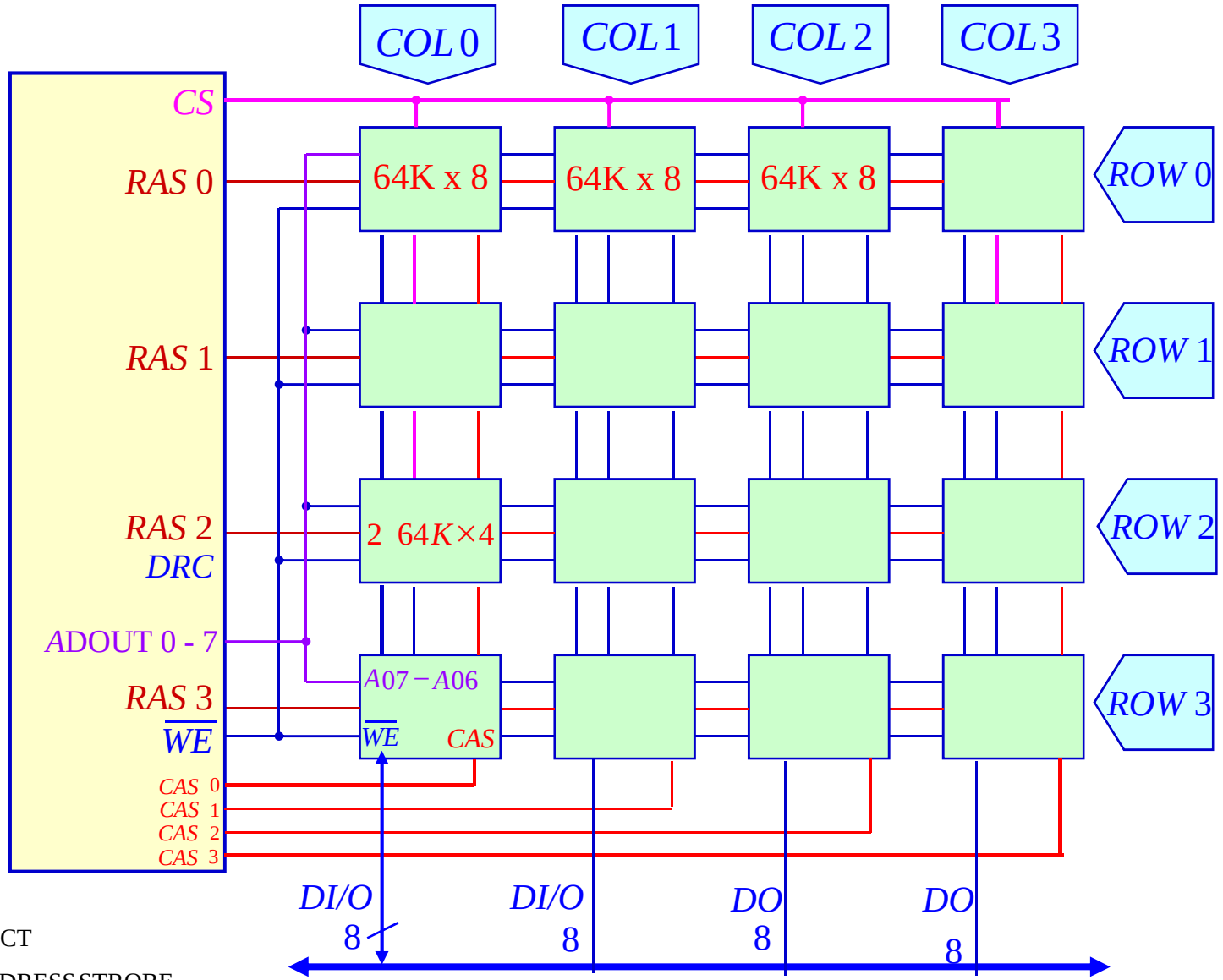
Organización 1-D de una memoria de 64Kbyte usando *SRAMs* de 8Kx8



Organización 2-D de una memoria de 64Kbyte usando *SRAMs* de 8Kx8



Organización de una memoria 1Mx8 usando *DRAMs* de 64Kx4



CS → CHIP SELECT

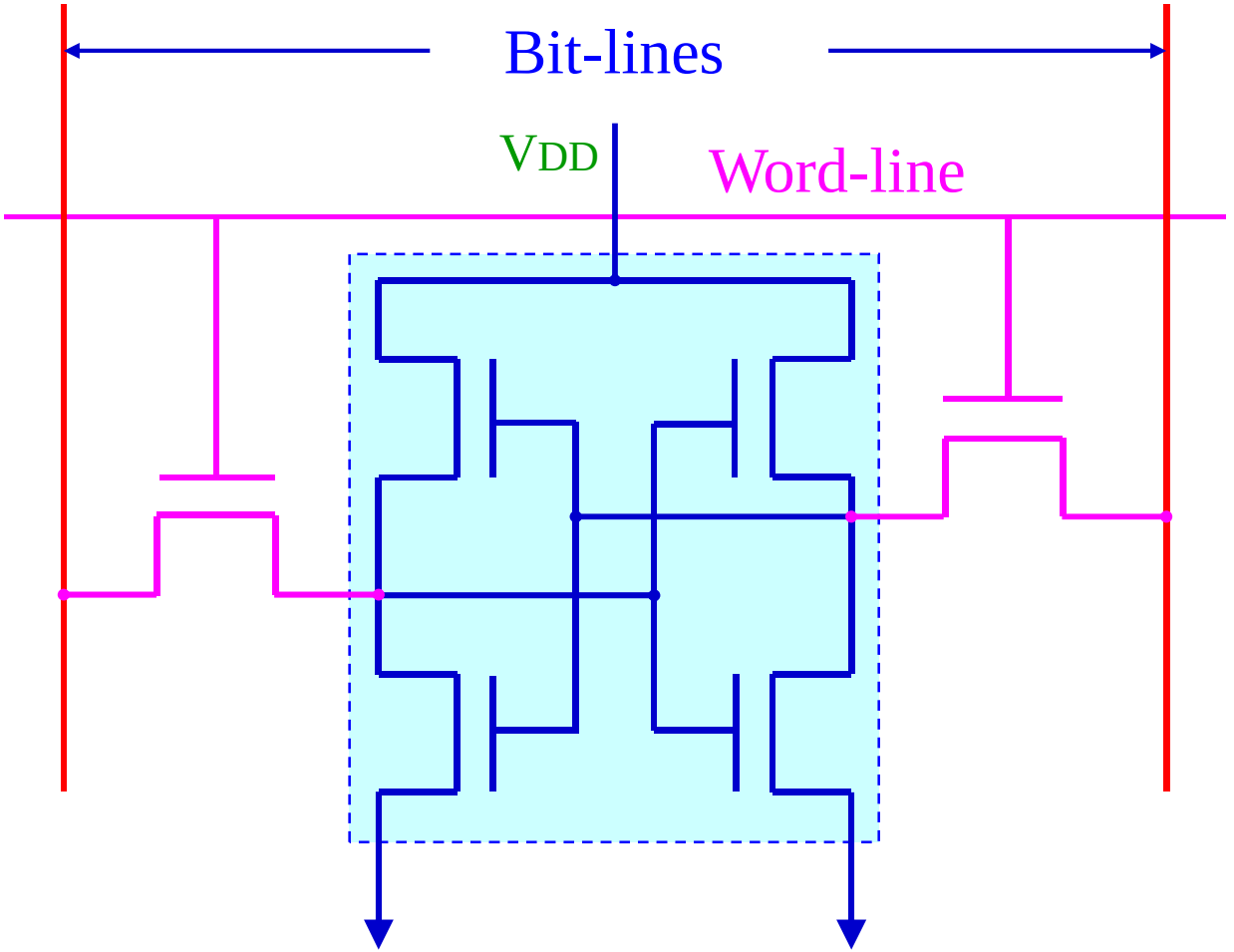
RAS → ROW ADDRESS STROBE

CAS → COLUMN ADDRESS STROBE

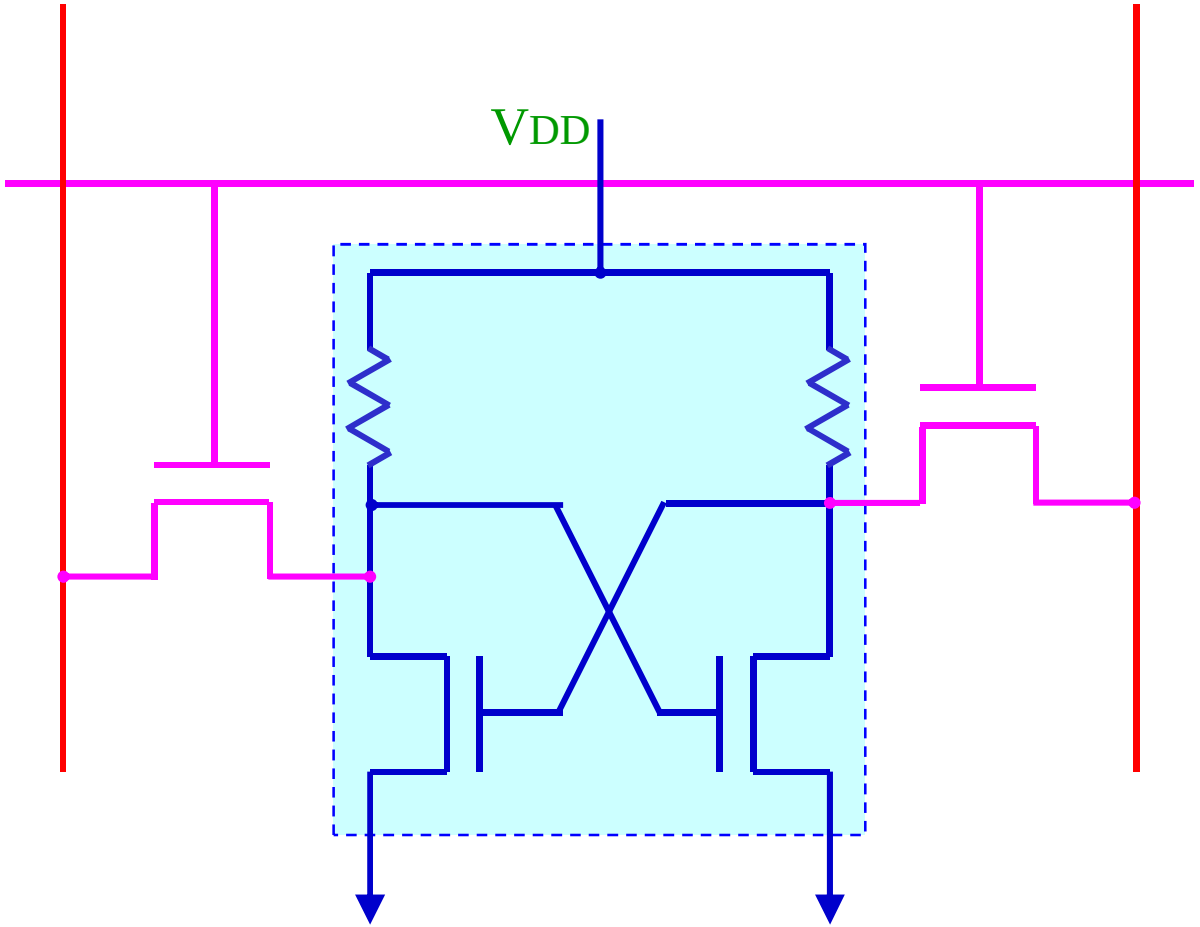
DI / O → DATA IN/OUT

BI-DIRECTIONAL DATA BUS DATA 7-0

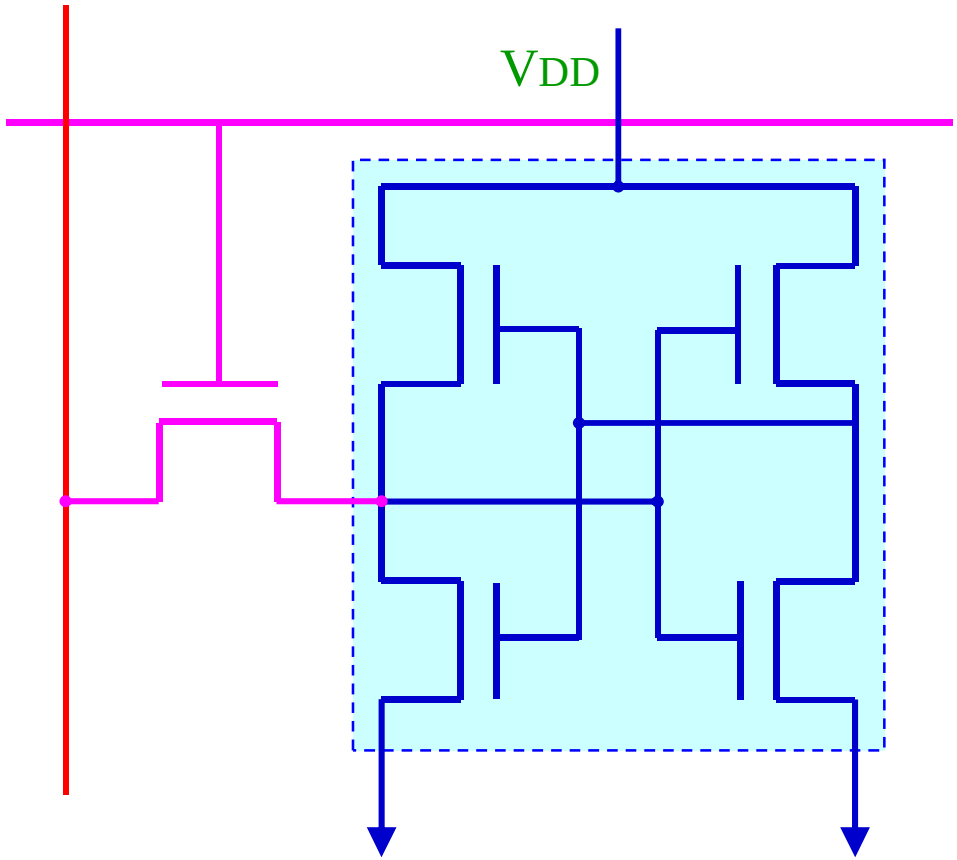
5.1 SRAM cell: CMOS cell



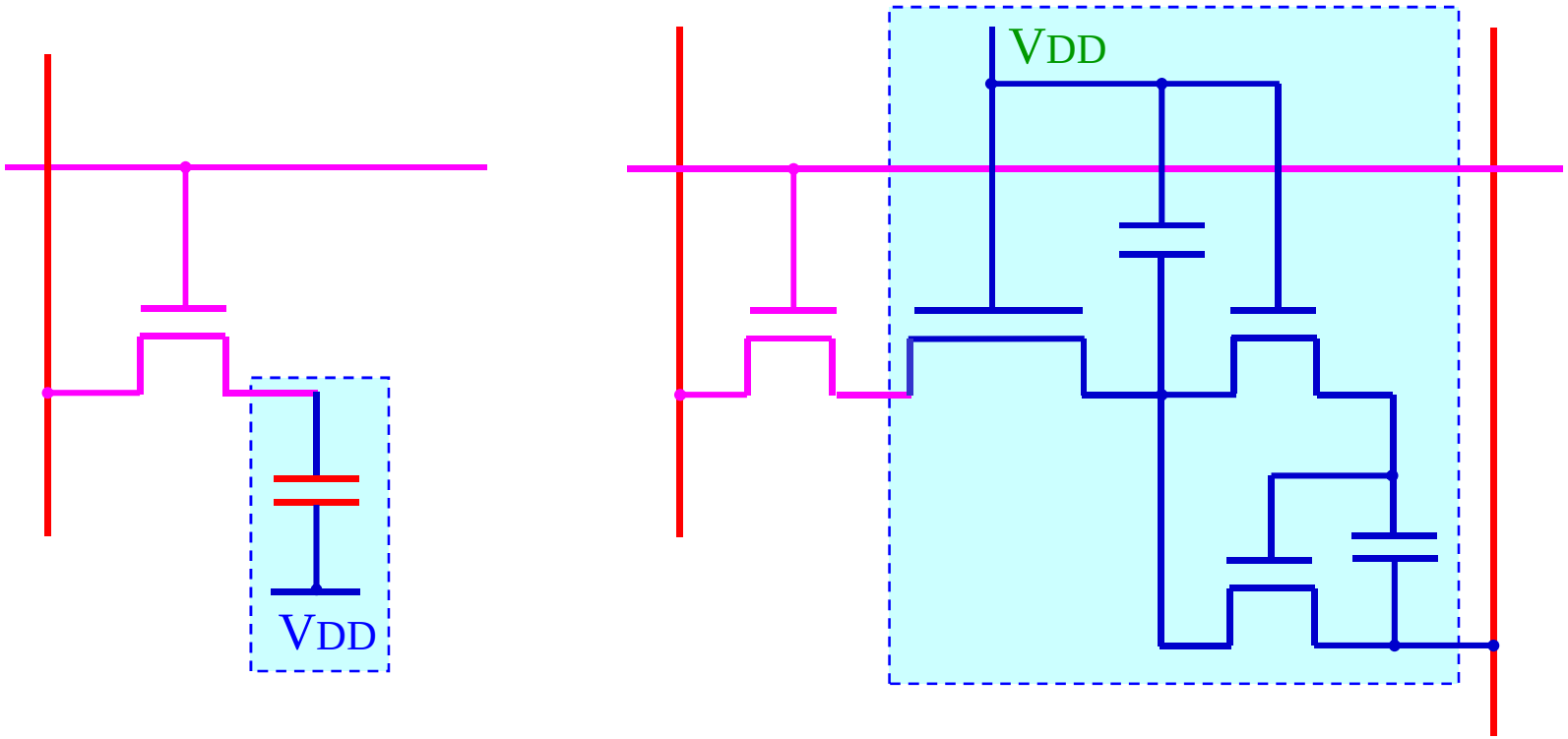
5.1 SRAM cell: RMOS cell



5.1 SRAM cell: CMOS single-ended cell



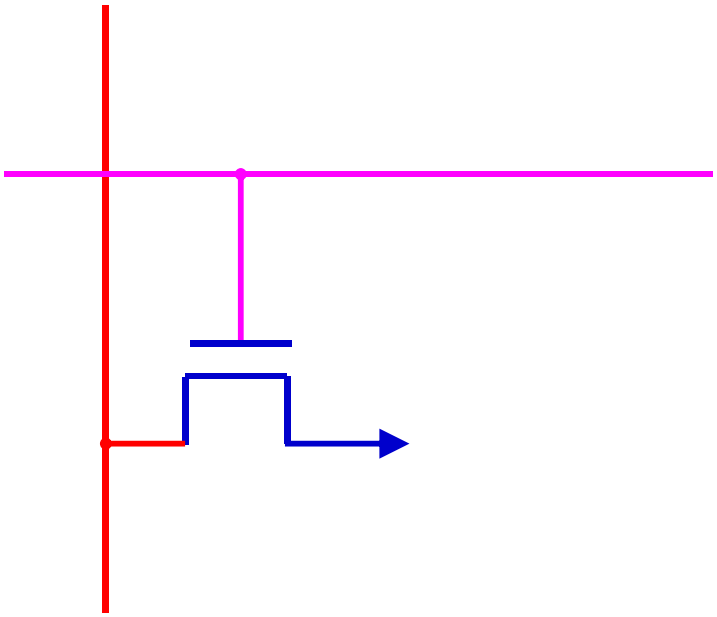
5.2 DRAM cell



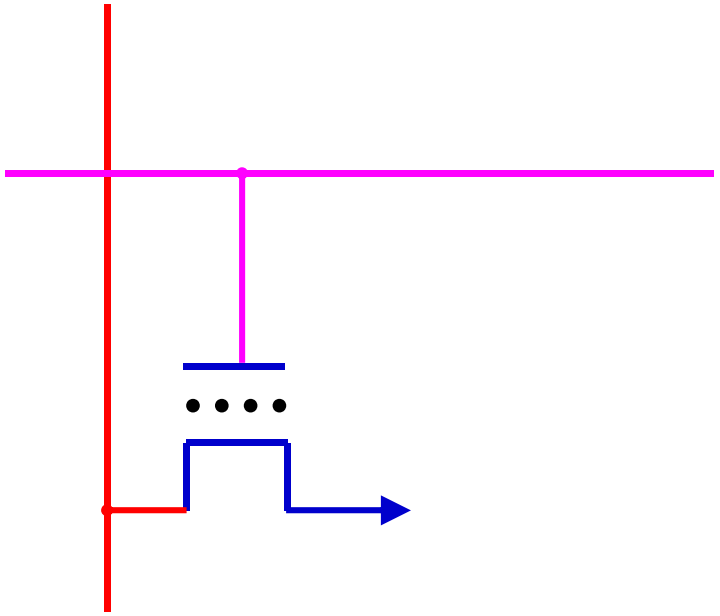
a) *DRAM cell*

b) *DRAM self-refresh cell*

5.3 EEPROM cell

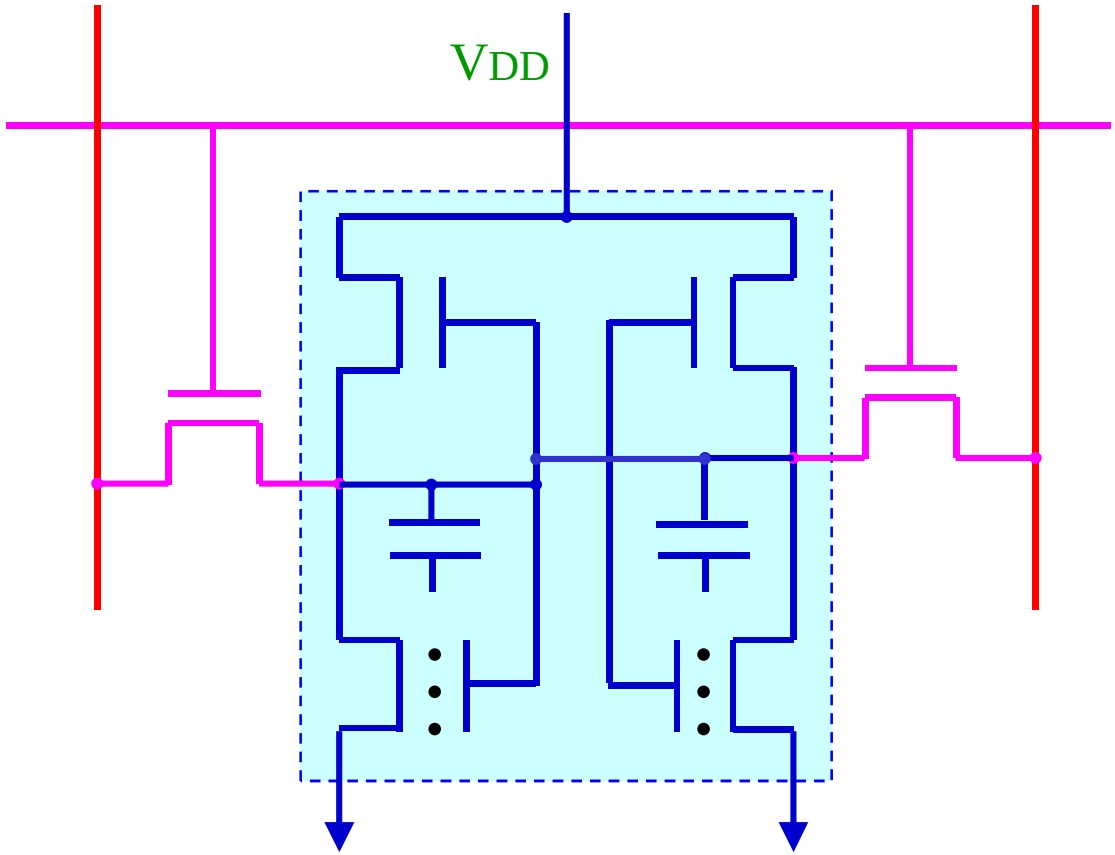


a) *ROM cell*

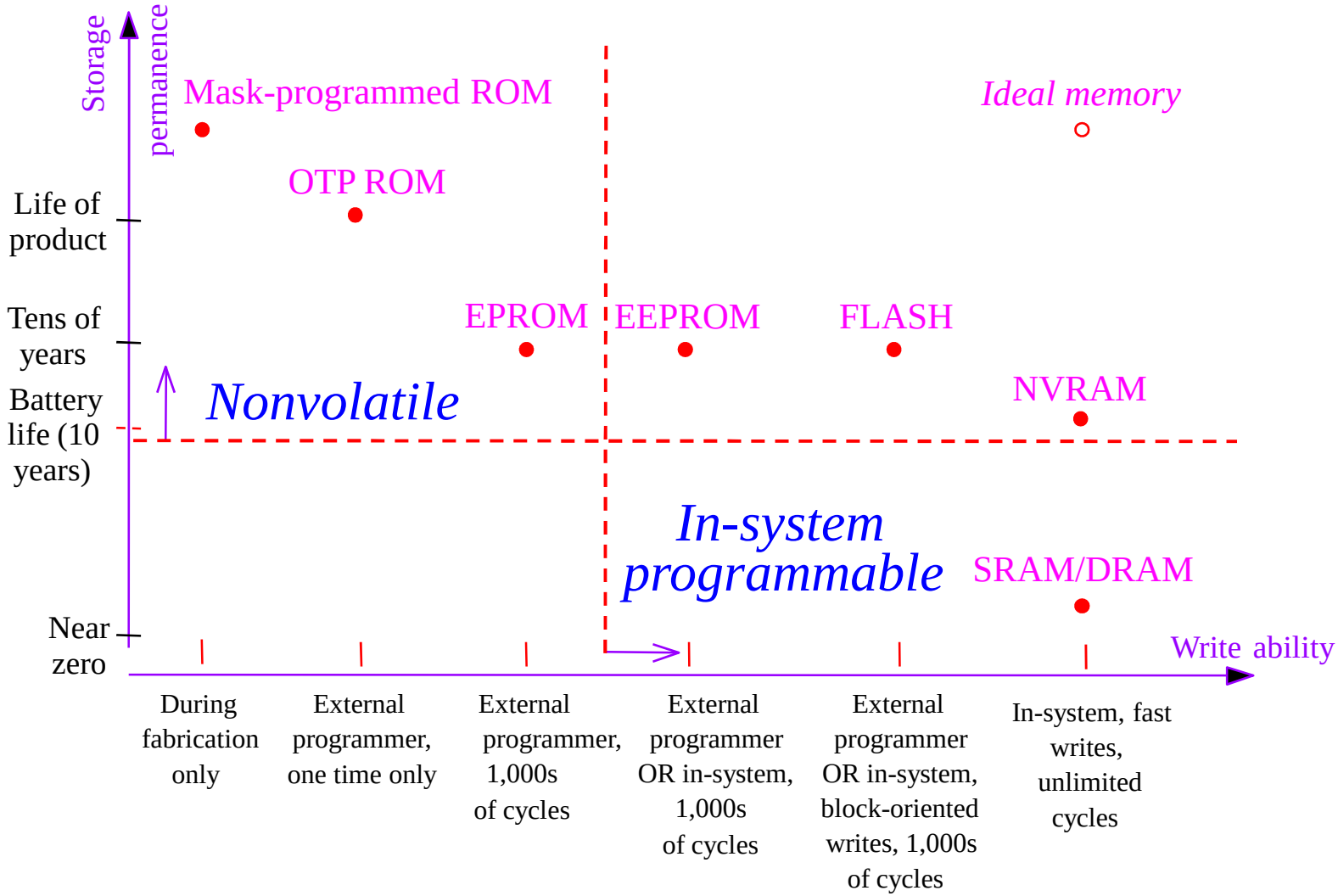


b) *EEPROM cell*

5.4 NVRAM cell



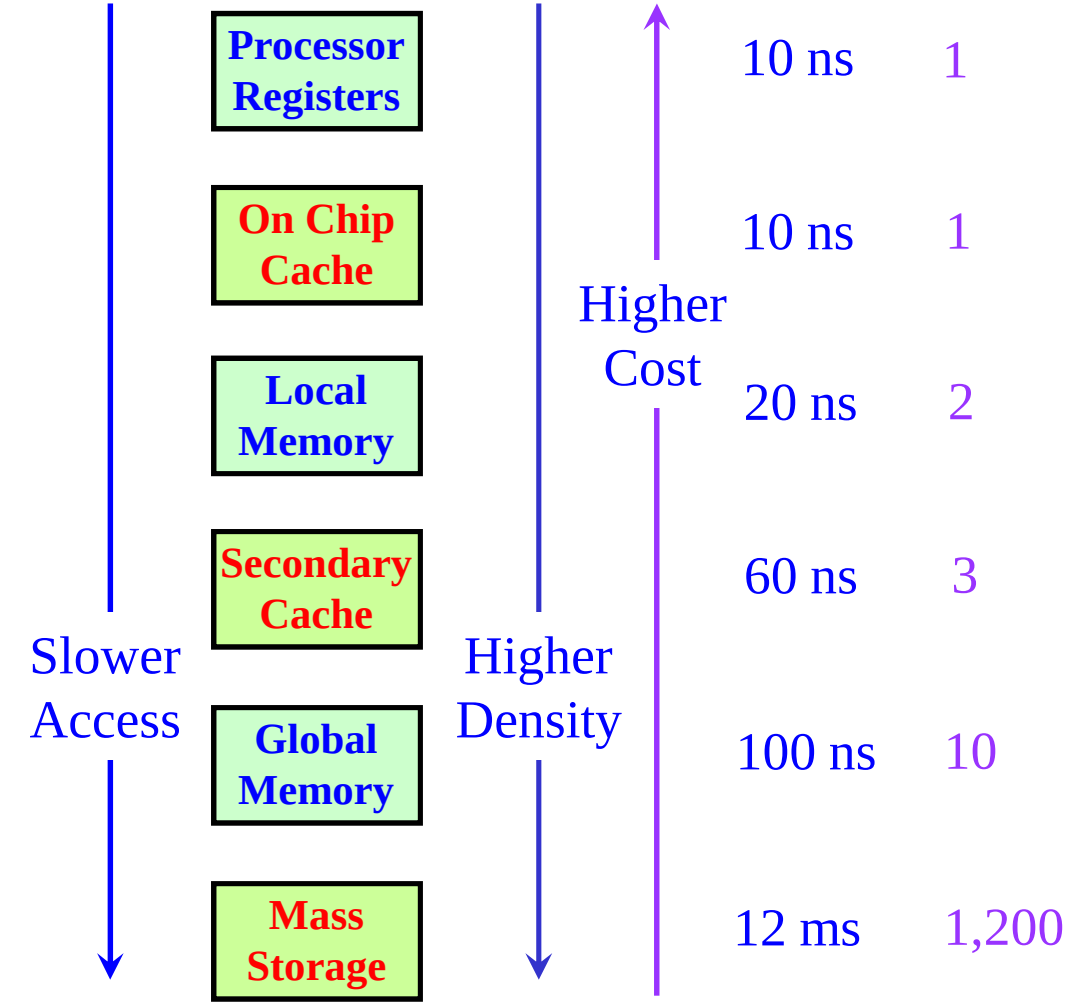
6. Write ability/ storage permanence



Write ability and storage permanence of memories, showing relative degrees along each axis (not to scale)

6. Memory Hierarchy

- Various *memory devices* are used in a *computer system*
- They have different *speeds and costs*



6. Cache Memory

- Two main problems:
  - mismatch in speed between processor and main memory
  - contention for contents of shared memory, or for switching, significantly increases latency

- Answer: Use a cache

6. What is a Cache?

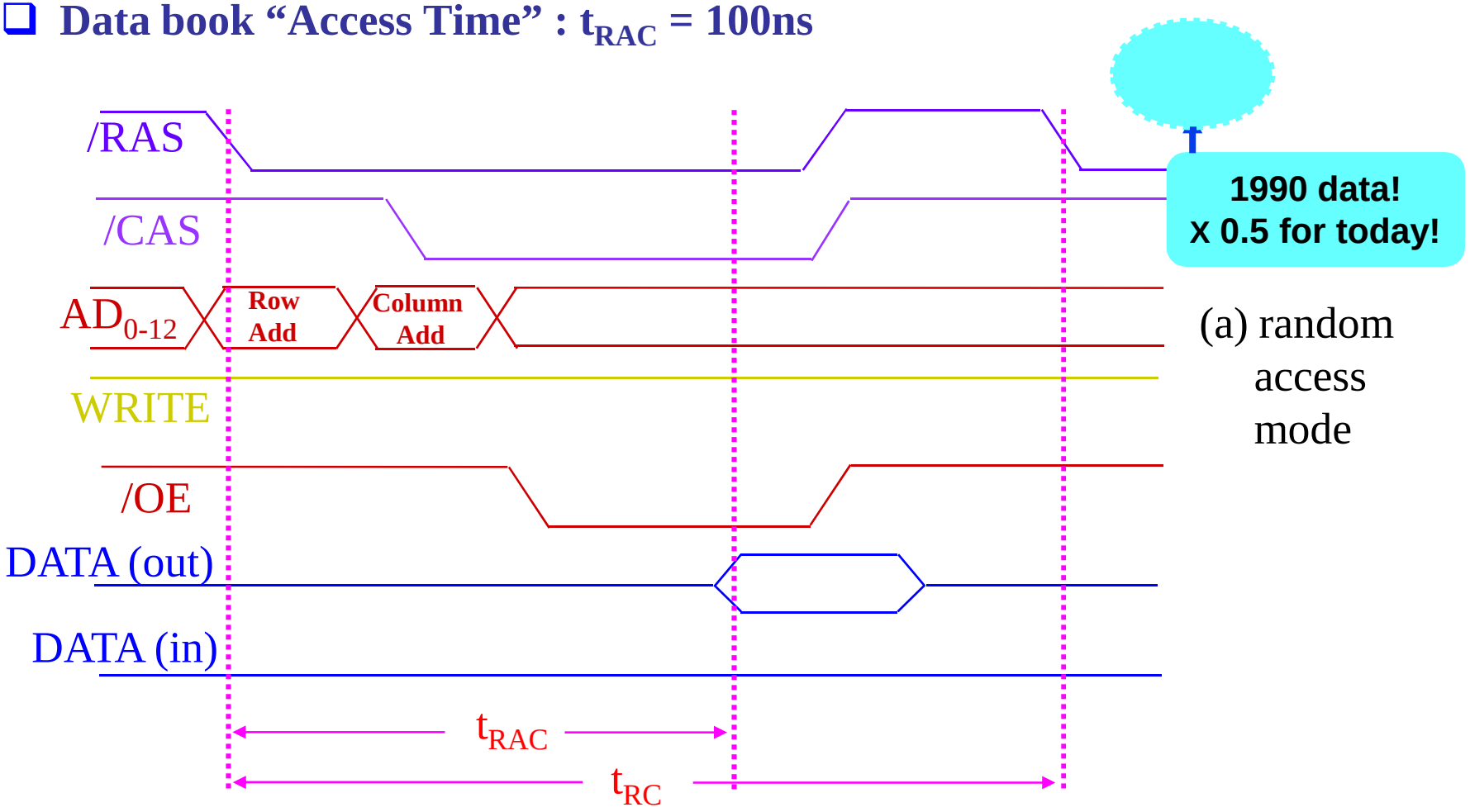
- High-speed memory module connected to a processor for its private use
- Contains *copies* of actively referenced material
- Copied between cache & memory in *lines* or *blocks*
- Caching works because of
  - *temporal locality* - repeated references to same memory in a small time
  - *spatial locality* - references to nearby memory addresses in a small time

6. Primary and Secondary

- *Primary cache* is usually on the CPU chip, for speed
- Few tens of kbytes, very fast
- May be separate data and instruction caches
- *Secondary cache* is off-chip
- Much larger (several Mbytes) than primary cache

Dynamic RAM - Access Time Myth

- **Data book "Access Time" : t_{RAC} = 100ns**



- Time between accesses: t_{RC} = 190ns

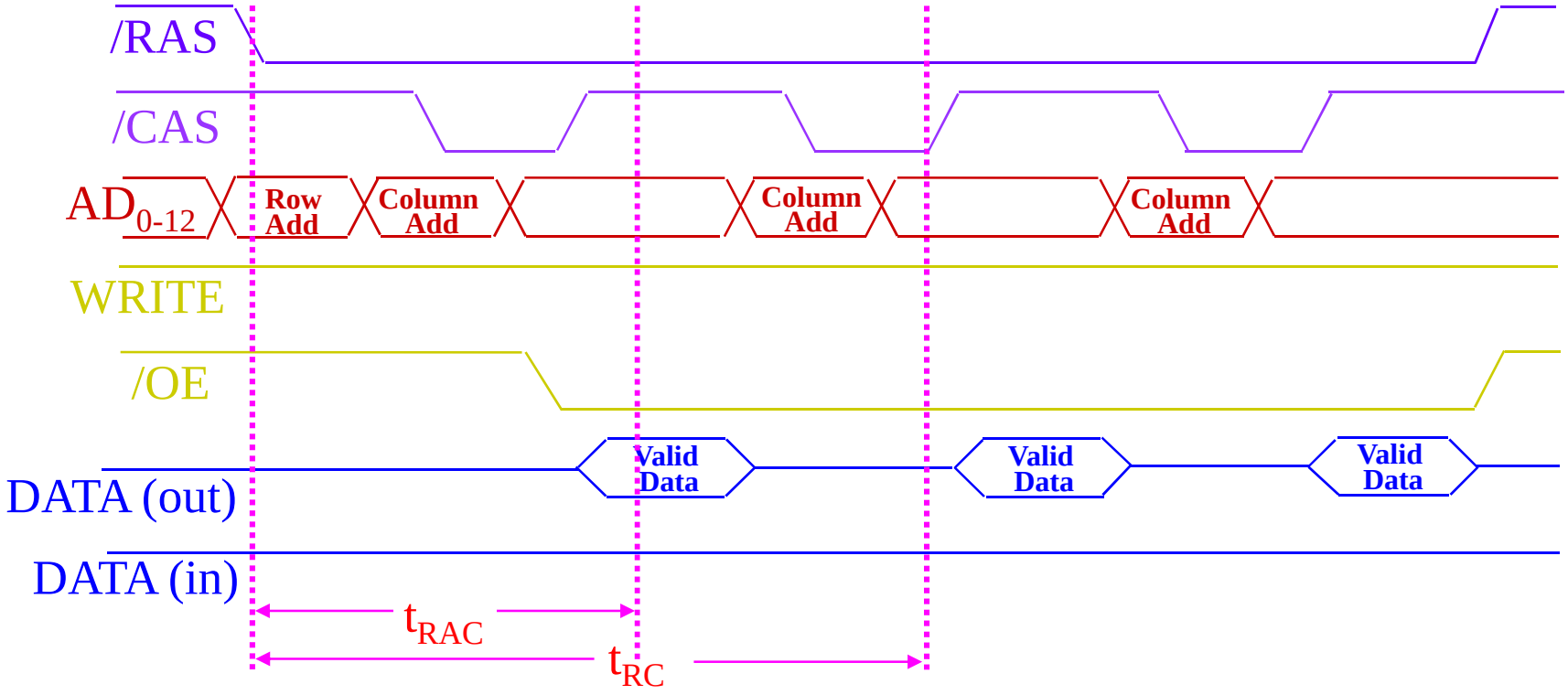
1990 Data: apply appropriate factor! (~0.5 for 1998)

Dynamic RAM - Improving Bandwidth

- Bandwidth = overall data rate $\propto 1/t_{RC}$
- t_{RC} determined by
  - packaging (stray C, L, lead R, etc)
  - need to drive PCB traces
  - program size
    - the IBM/Microsoft complexity factor J
    - DRAM manufacturers
      - *Go for size rather than speed!*
  - Dynamic RAM
    - refresh circuitry adds complexity
  - halved in 8 years!

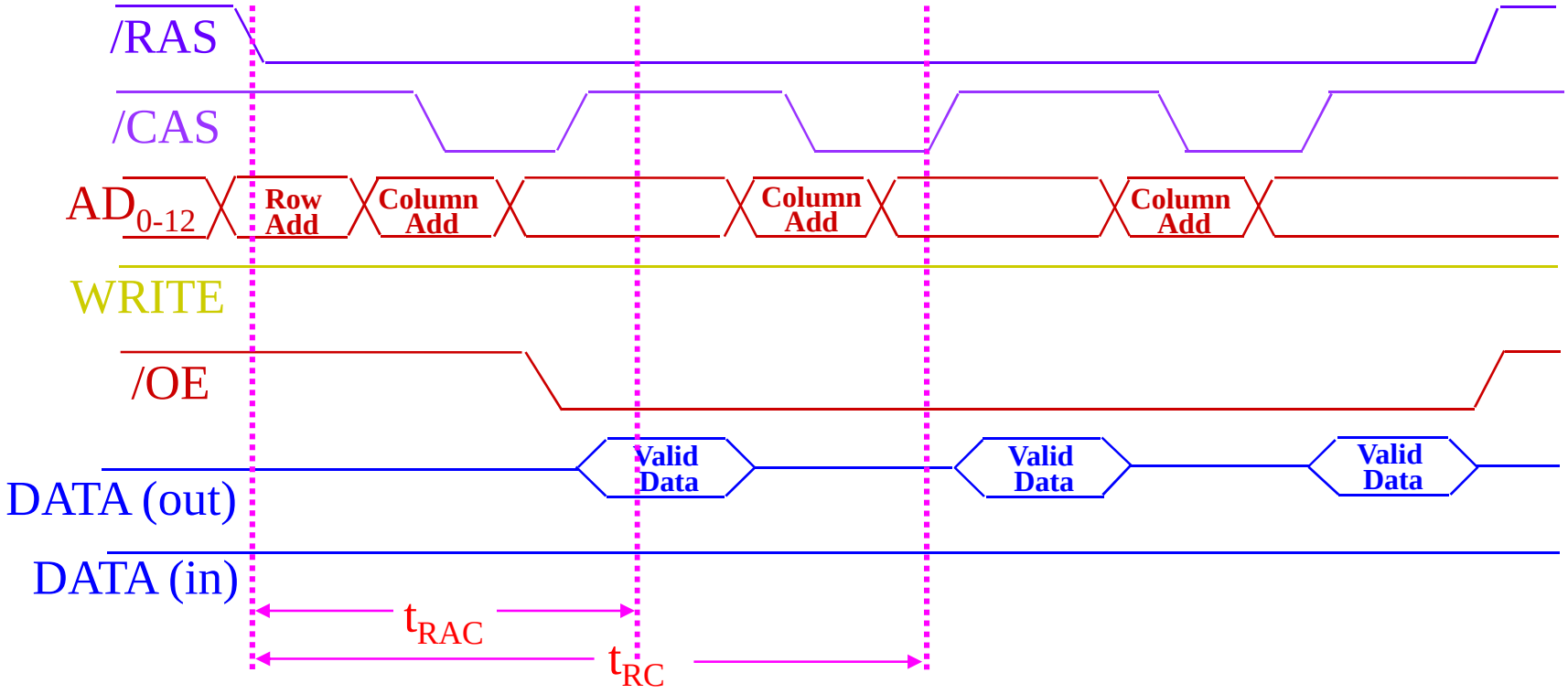
Dynamic RAM - Improving Bandwidth

- t_{RC} is difficult to reduce
- *So* increase amount of data per t_{RC}
- ← Page mode DRAMs



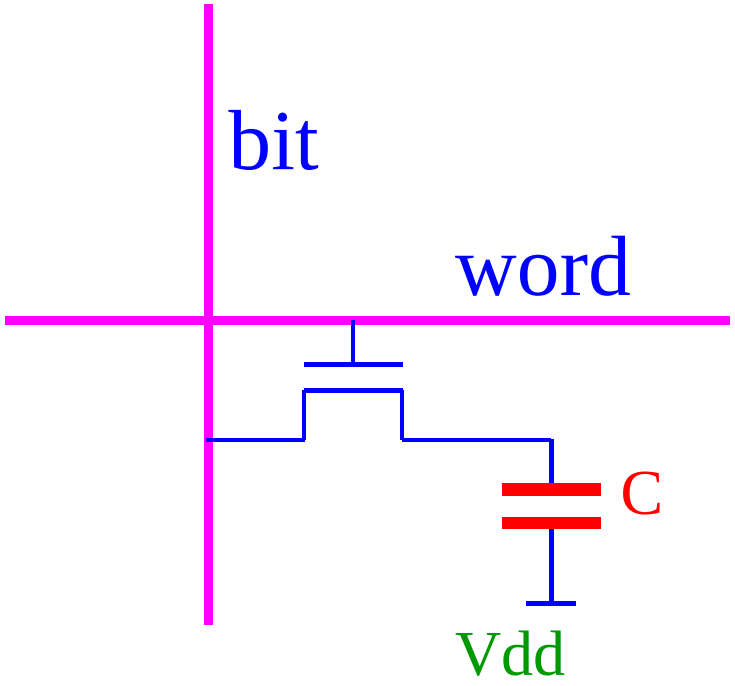
Dynamic RAM - Improving Bandwidth

- Assert one row address, then multiple column addresses
- Locality of reference - many accesses within a page



5. Dynamic RAM

- Refresh
- DRAM has one-transistor cells
- Charge leaks from storage capacitor
- Refresh interval ~ 1-4 ms
- RAM unavailable during refresh!
  - Some bandwidth loss



Dynamic RAM - EDO?

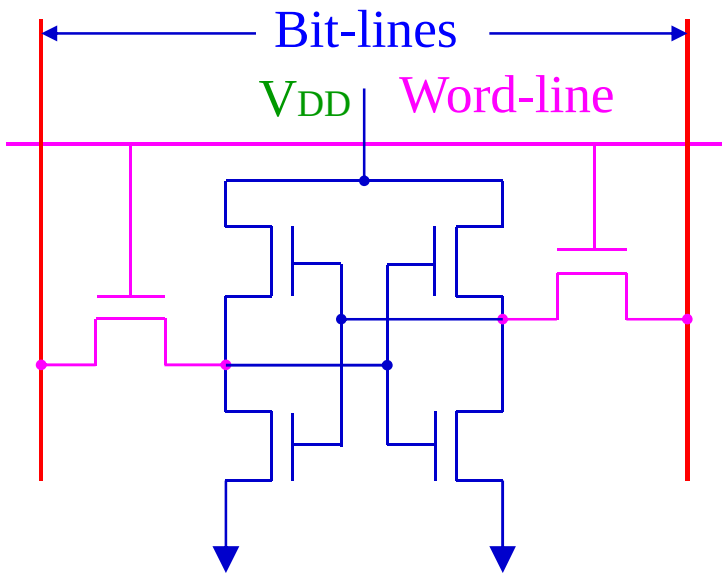
- EDO = Extended Data Out
- Extends time for which data is held
- Allows new column address to be applied while data is being read
- 10-15% improvement (by shrinking effective cycle time!)

SDRAM/SLDRAM - Just more acronyms?

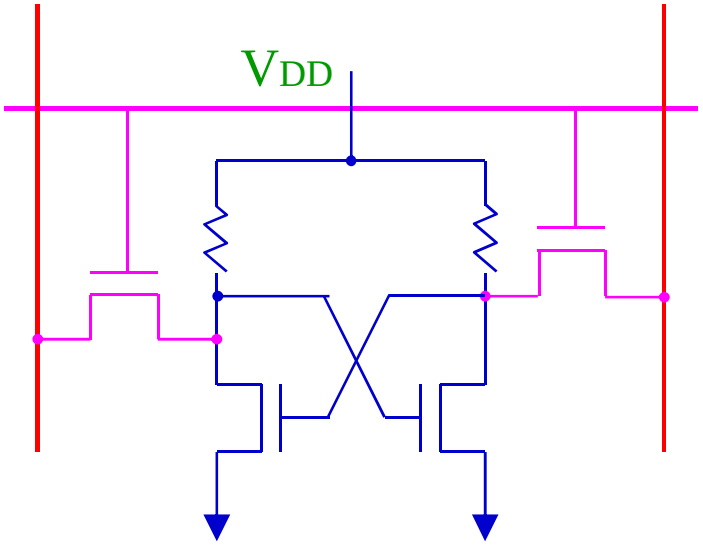
- Synchronous
  - more constraints on the designer
    - everything must respond in time!
  - faster
    - less time wasted in REQ/GNT protocols
    - time slots for commands/data
- Bursts data for high throughput
- SLDRAM - Synchronous Link DRAM
  - memory access protocol
  - SLDRAM memory is a subsystem

Dynamic RAM - RAMBUS

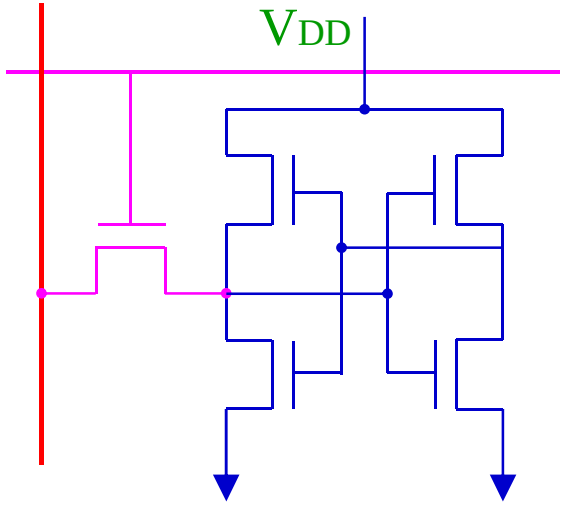
- "Packet" mode of operation
- Request large bursts of bits in one access
- 250MHz clock
- Data bit on each clock edge
  - ⇐ 500Mbits / second / pin!
  - suitable for graphics
  - large blocks of data streamed from memory to device
  - game machines!



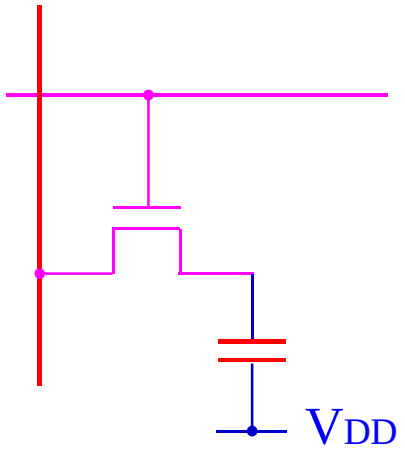
a) *CMOS cell*



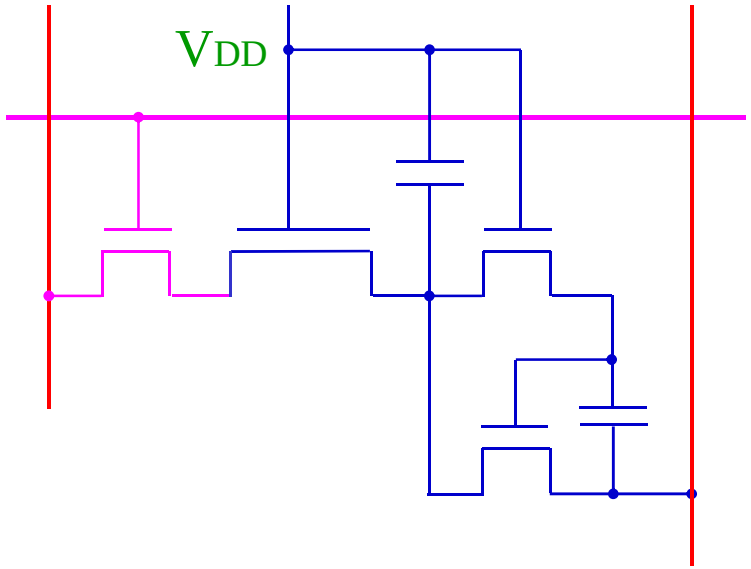
b) *RMOS cell*



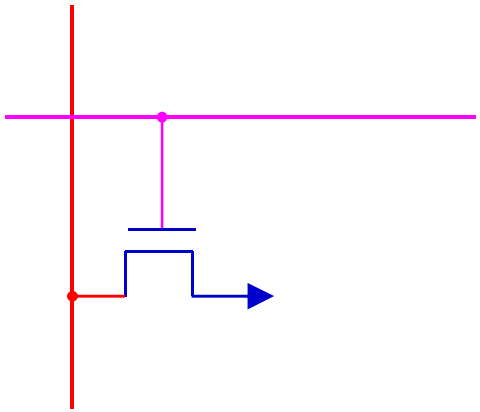
c) *CMOS single-ended cell*



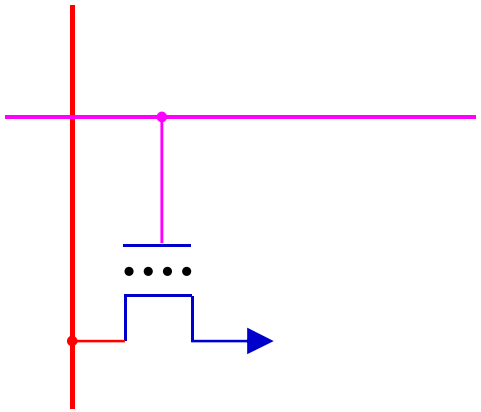
d) *DRAM cell*



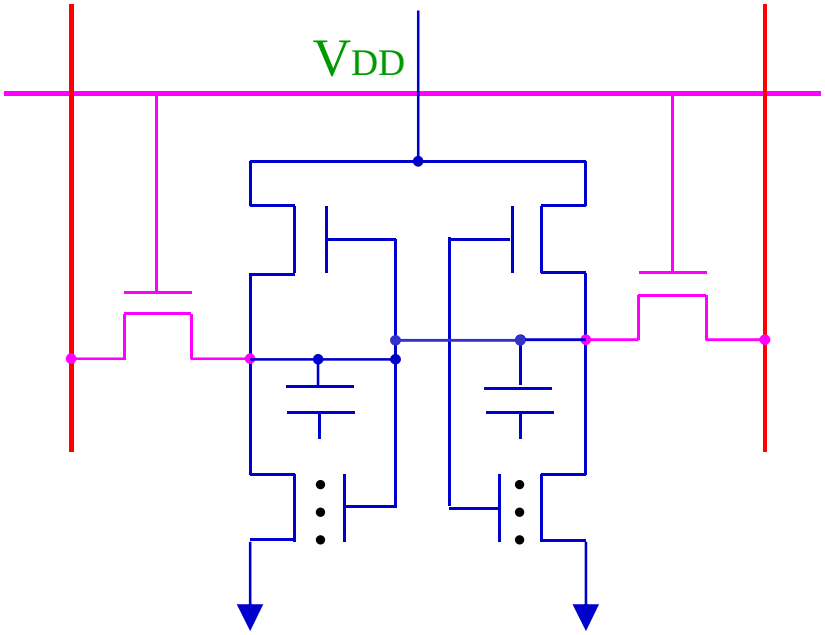
e) *DRAM self-refresh cell*



f) *ROM cell*



g) *EEPROM cell*



h) *NOVRAM cell*