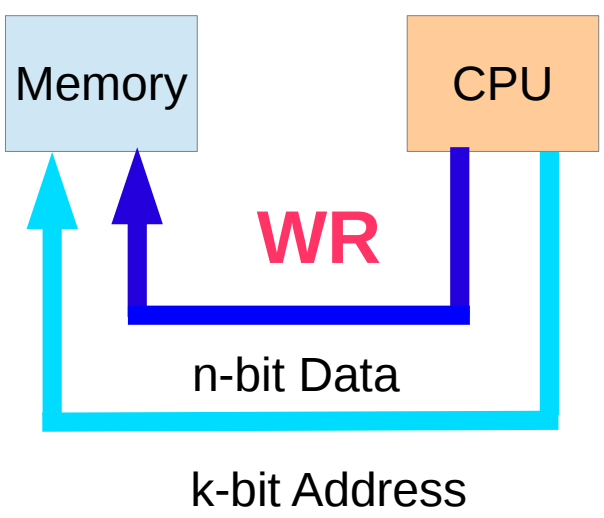
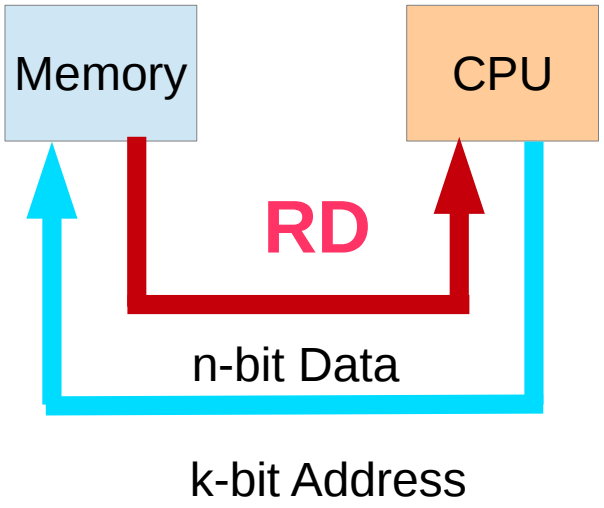
Memory Subsystem Background

Please send corrections (or suggestions) to youngwlim@hotmail.com.

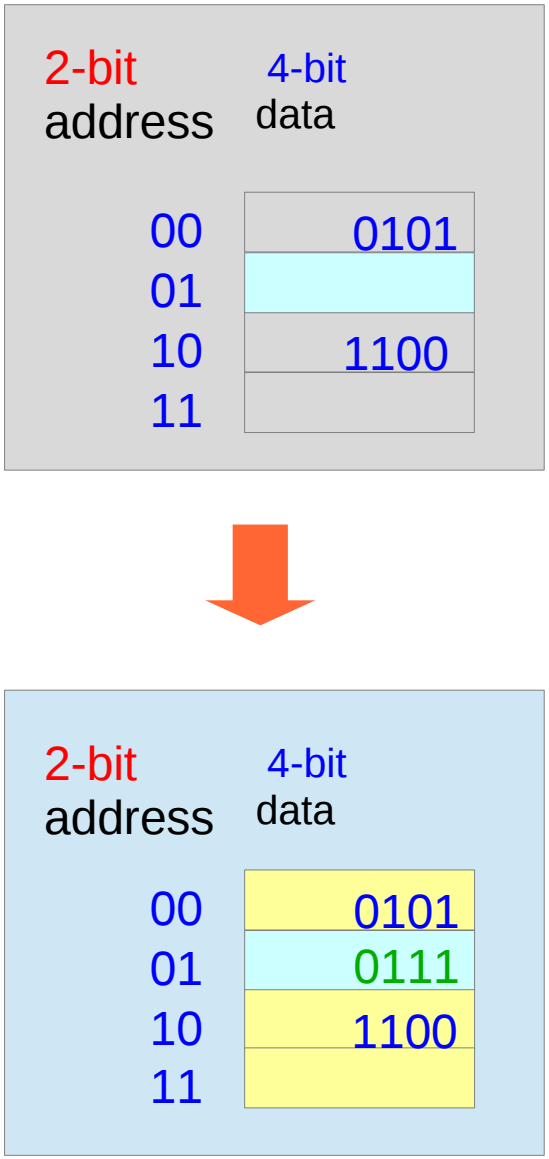
This document was produced by using OpenOffice and Octave.

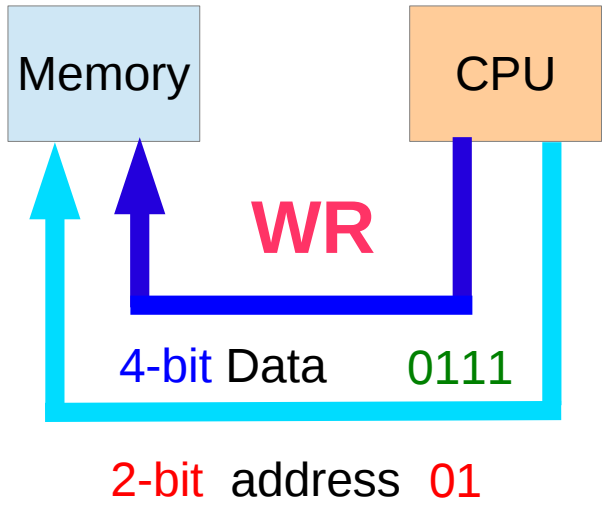
Memory Access Operations





A Memory Write Example





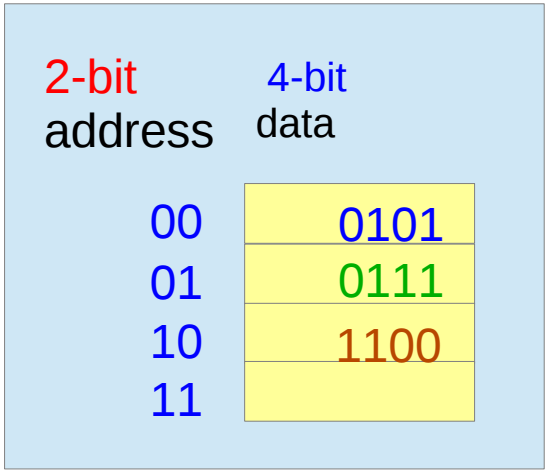
2-bit address 01

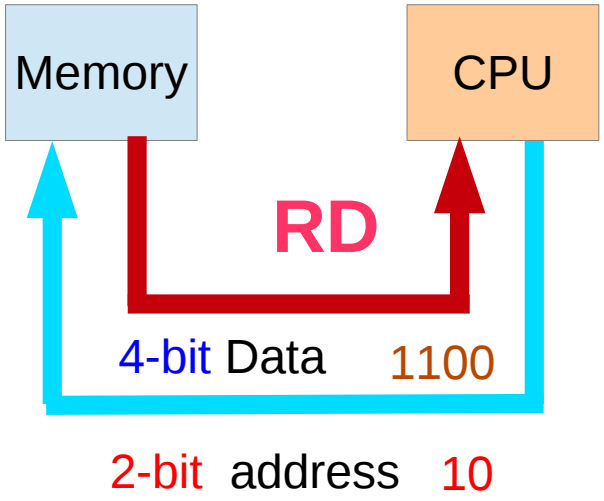
WR

Address: 01

Data: 0111

A Memory Read Example



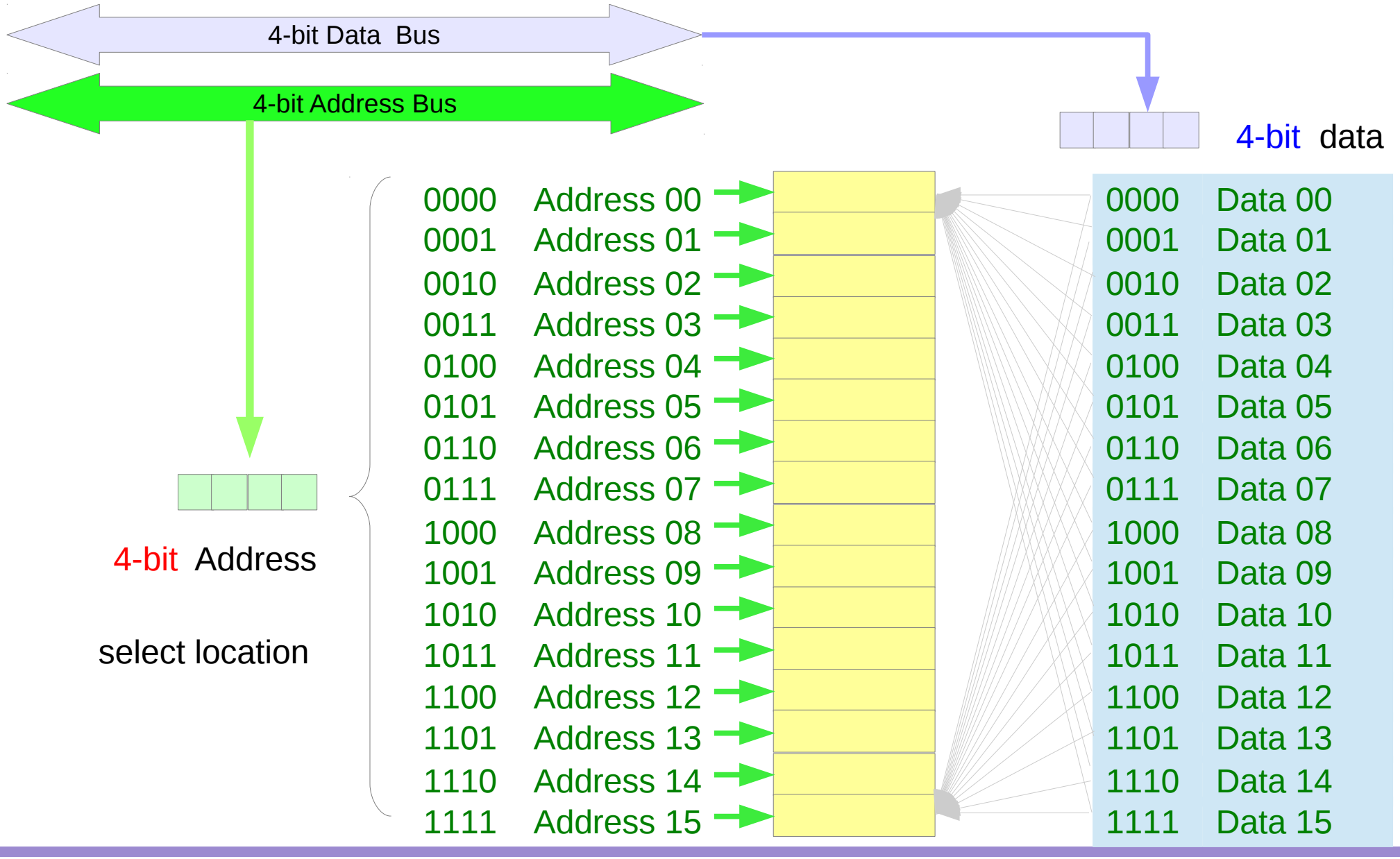


RD

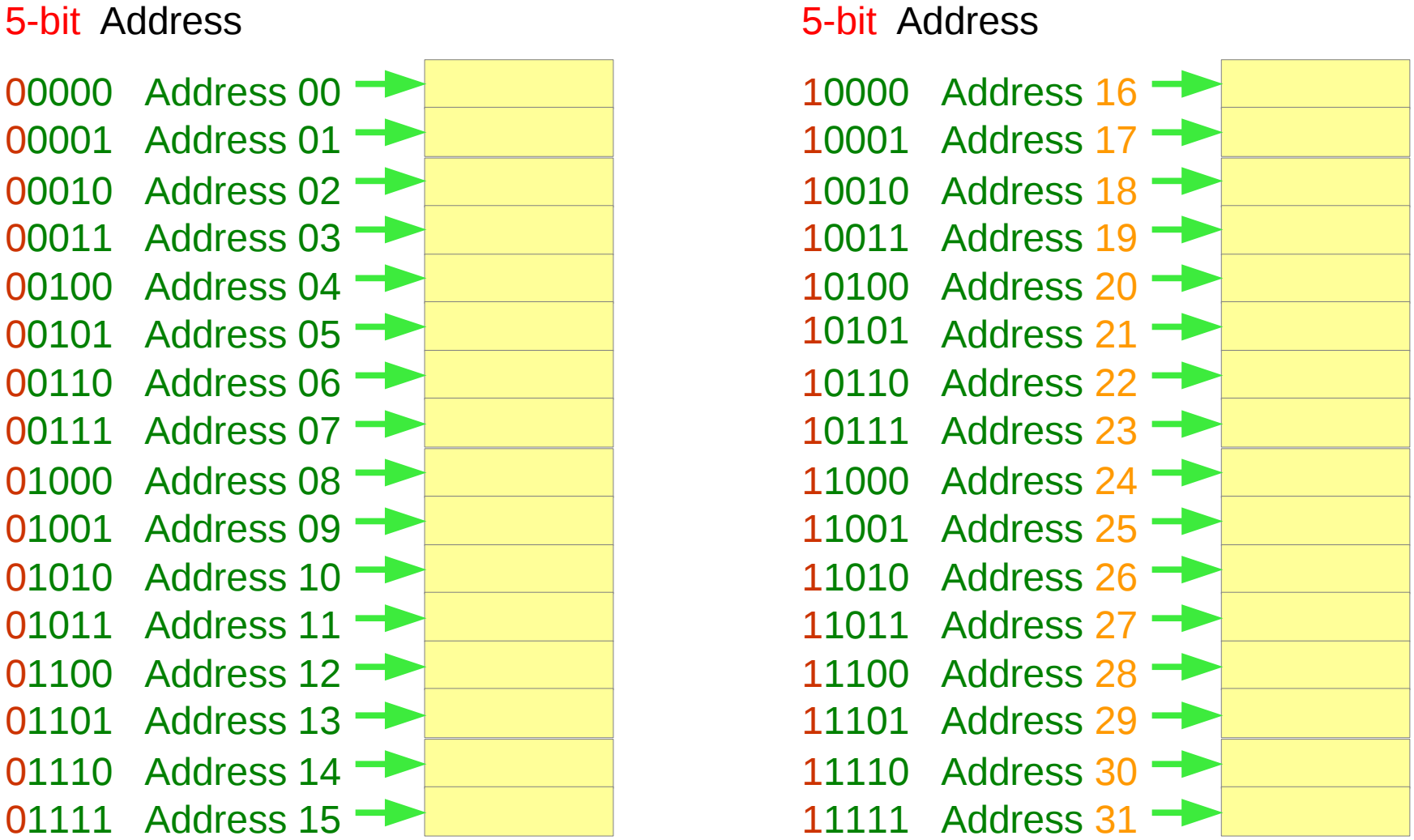
Address: 10

Data: 1100

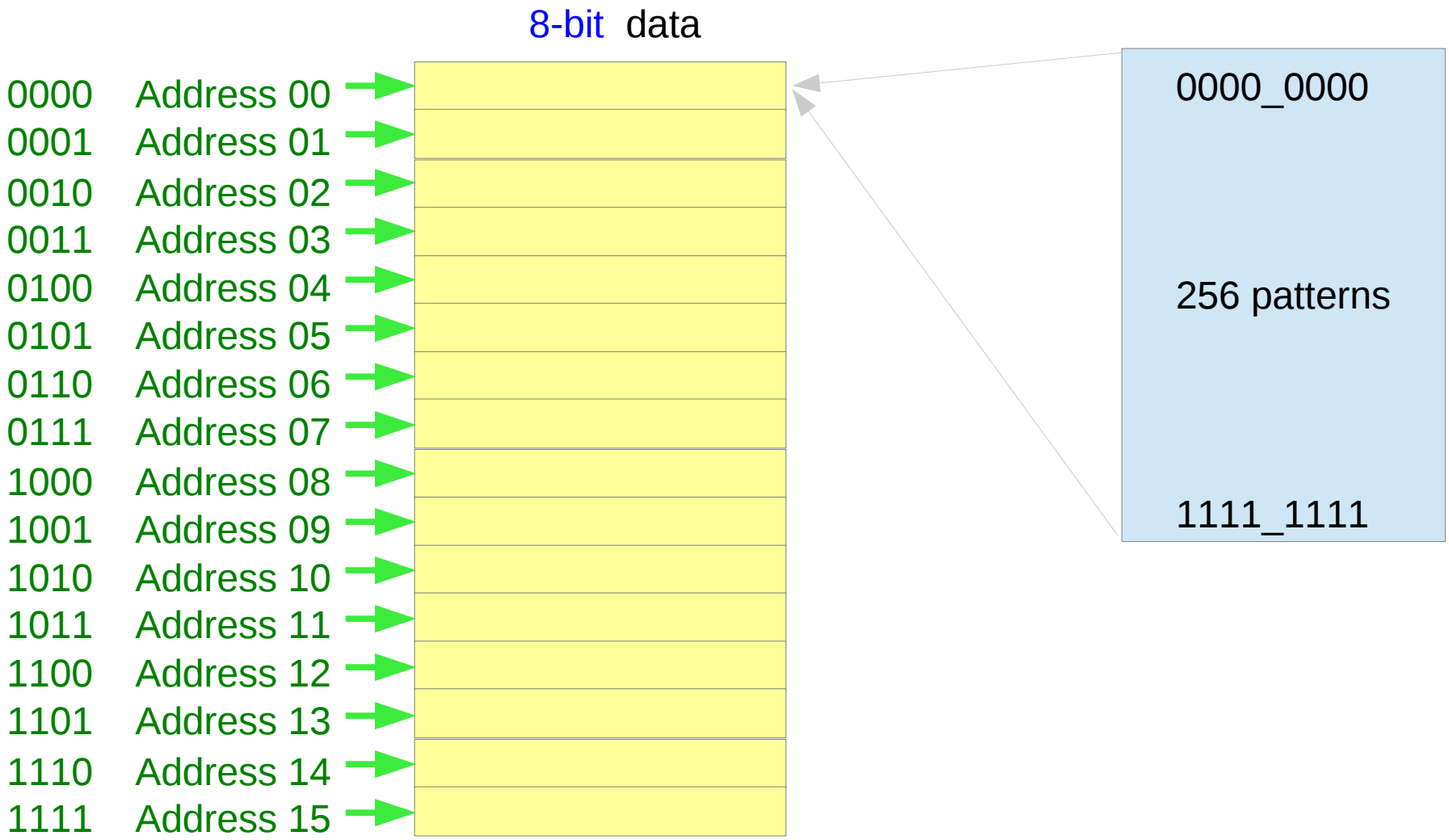
Address Bits and Data Bits



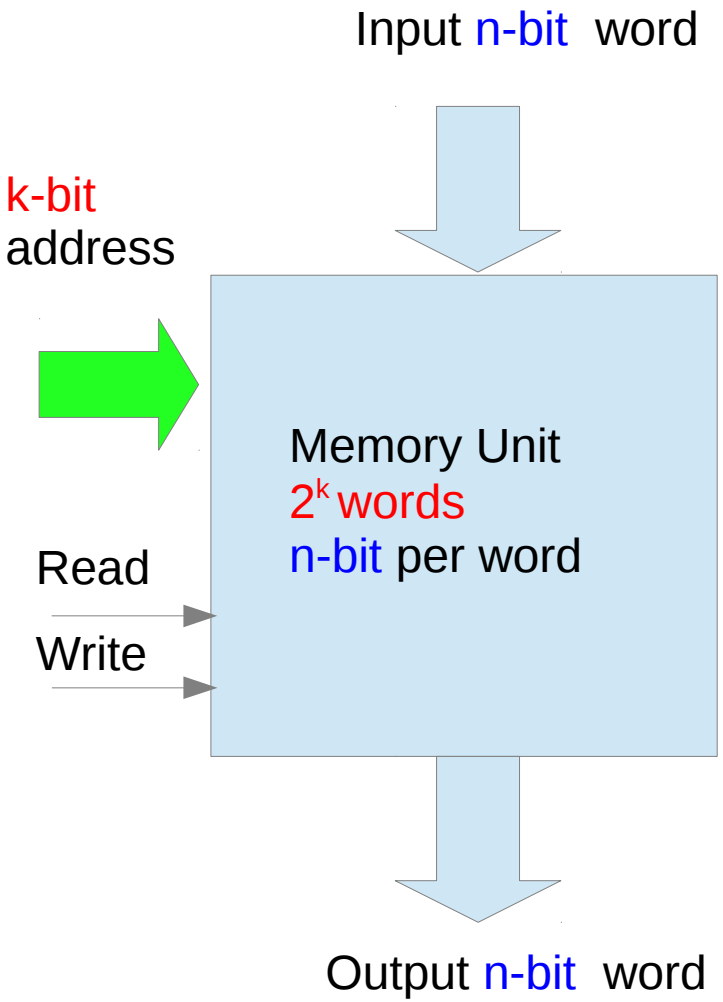
Increasing Memory Locations

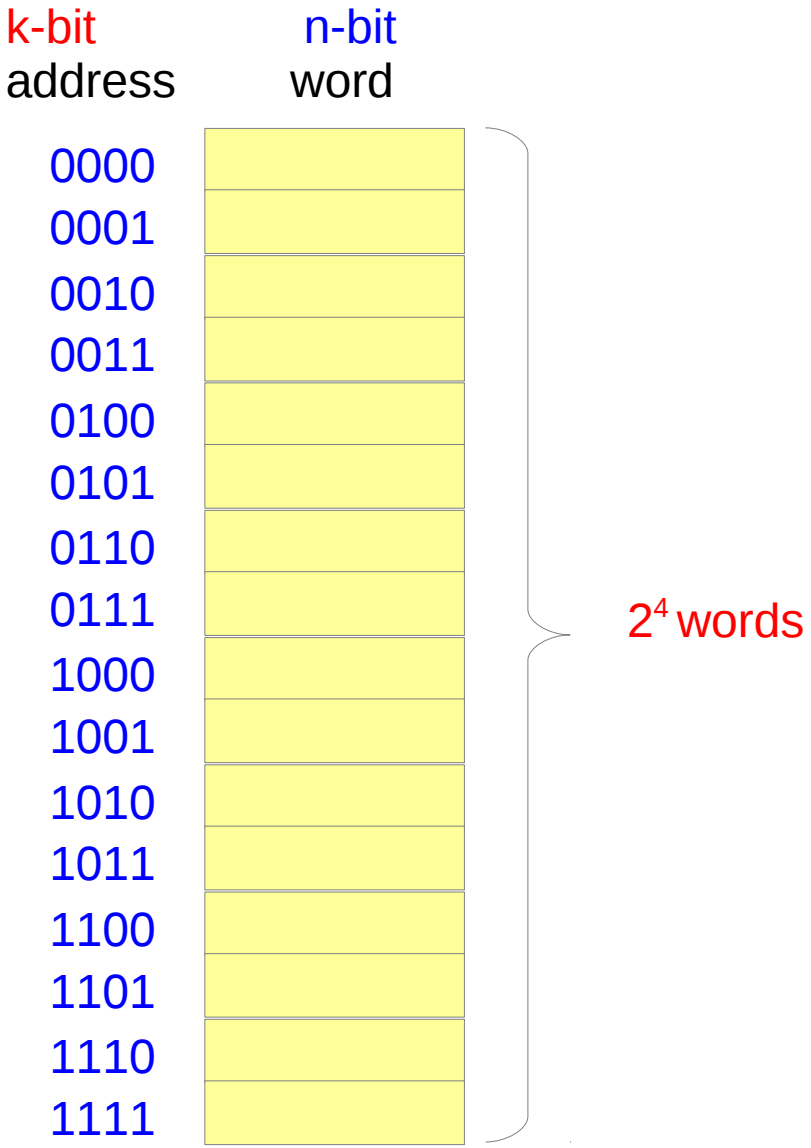


Increasing Memory Width

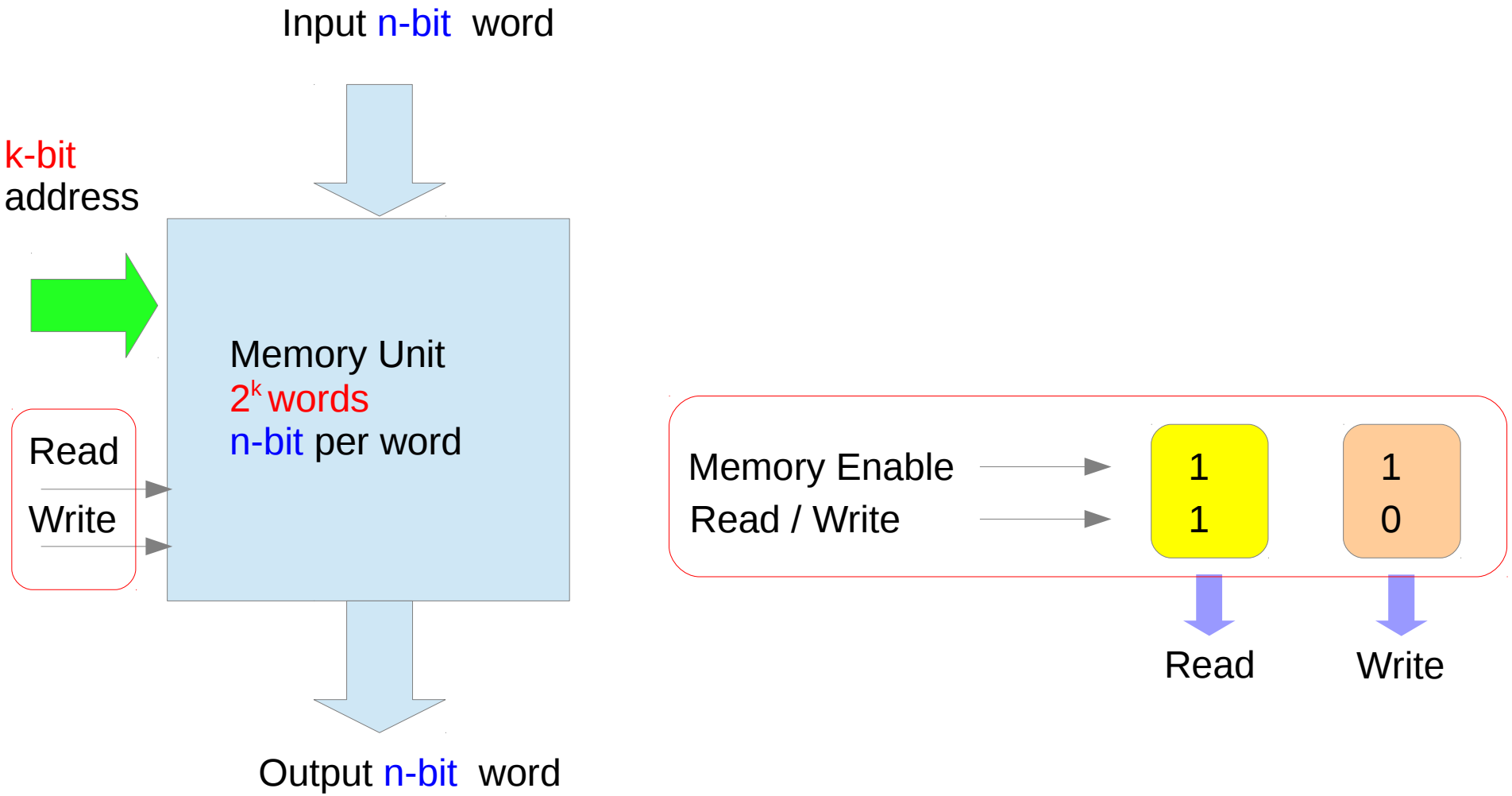


Block Diagram of a Memory Unit

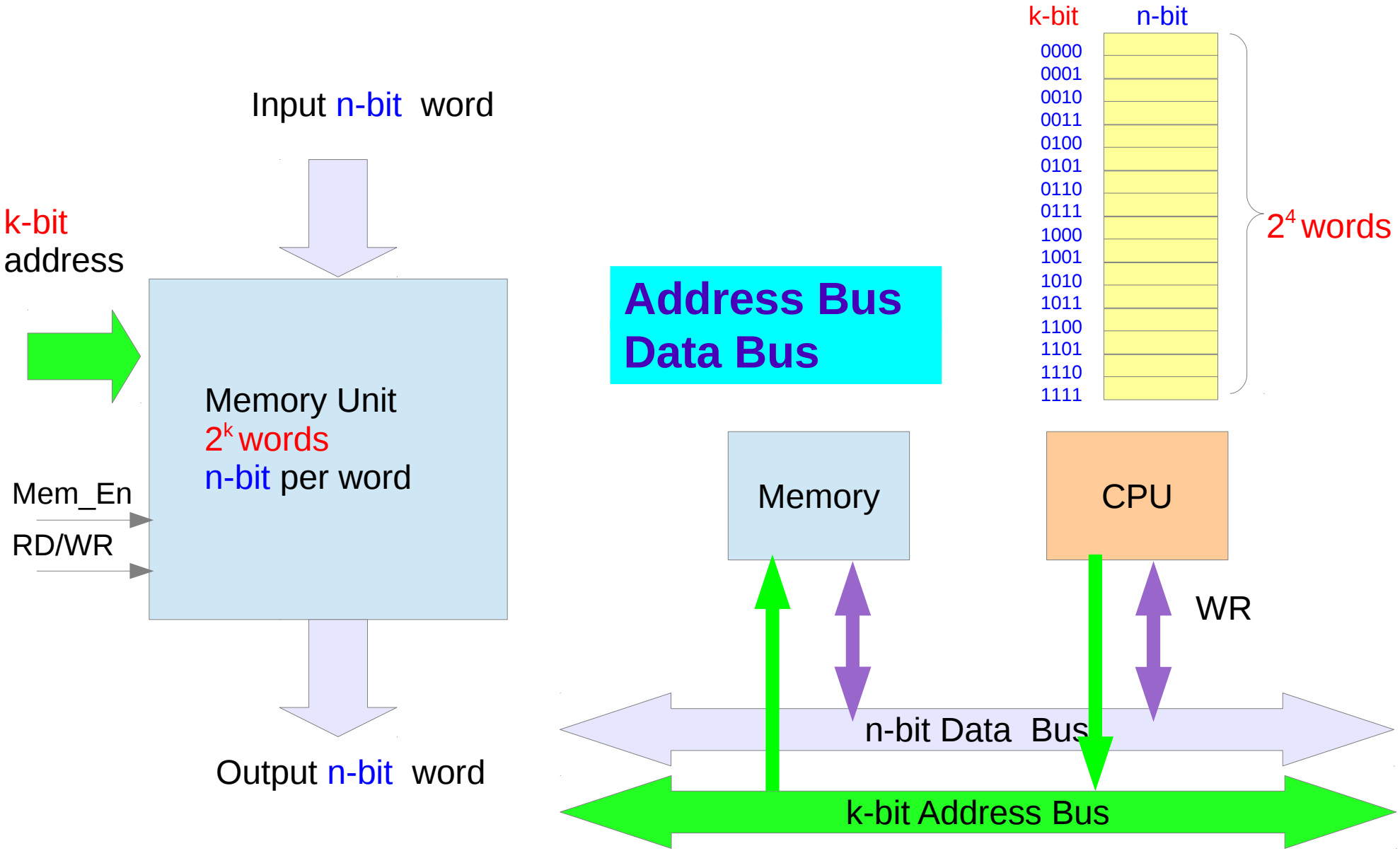




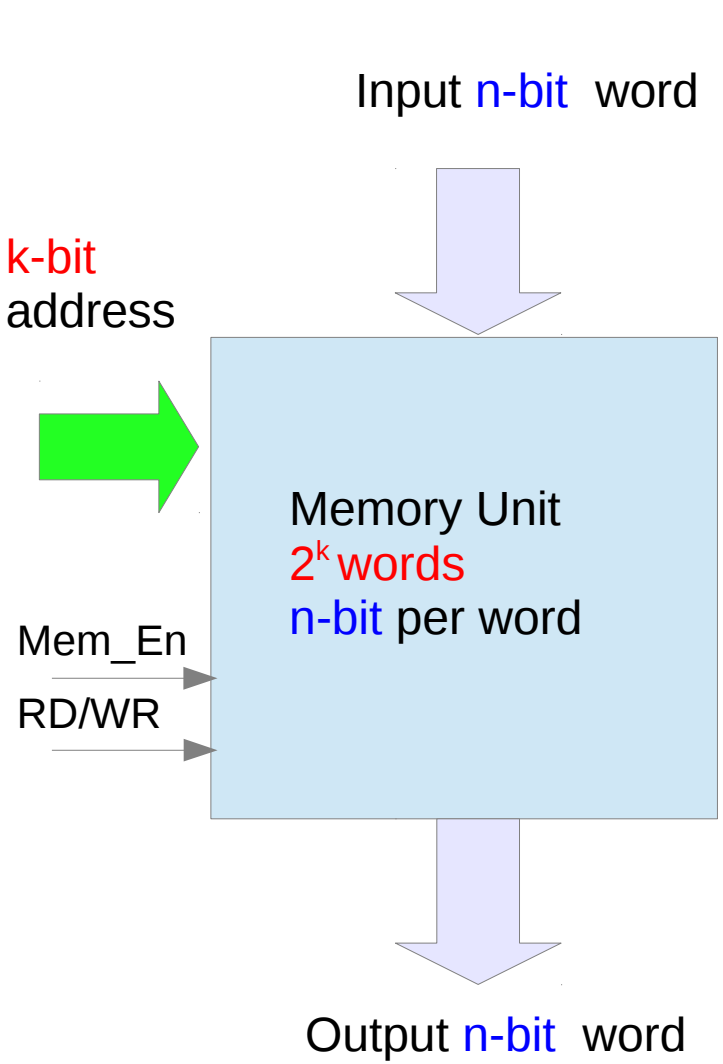
Memory Control Lines

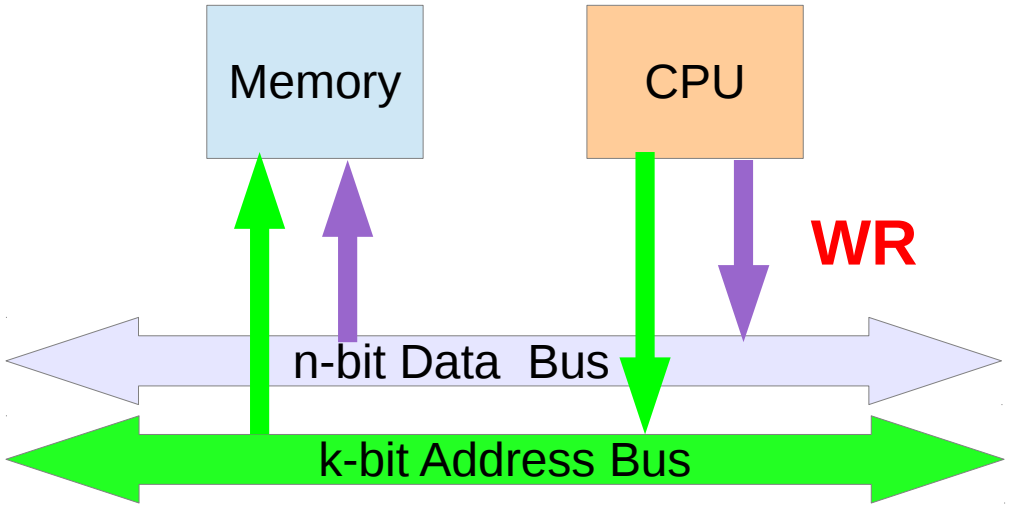


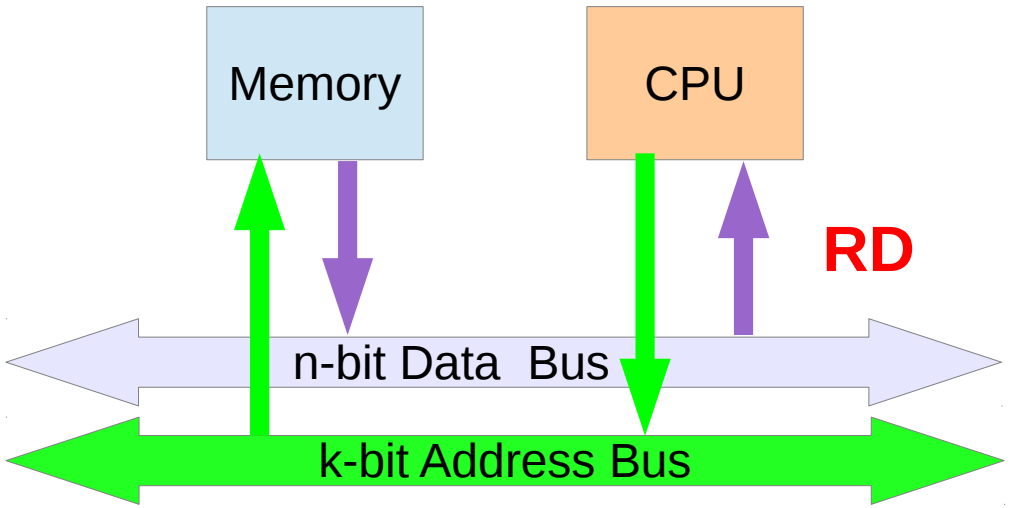
Address & Data Buses



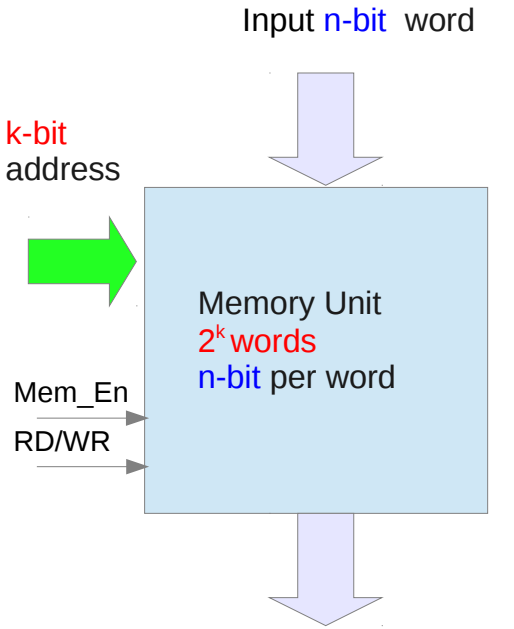
RD & WR Operations

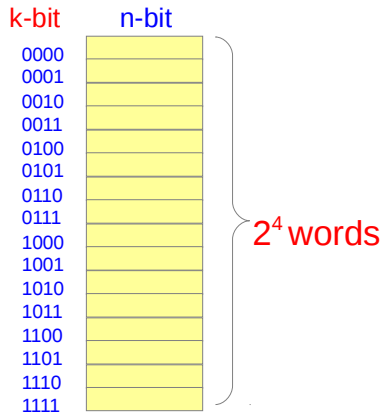


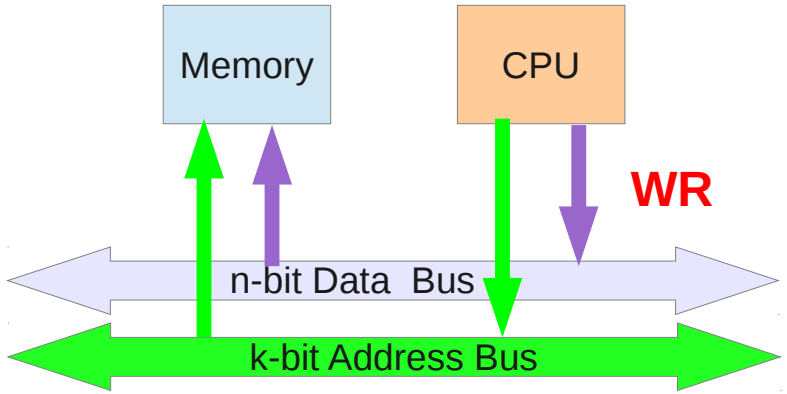


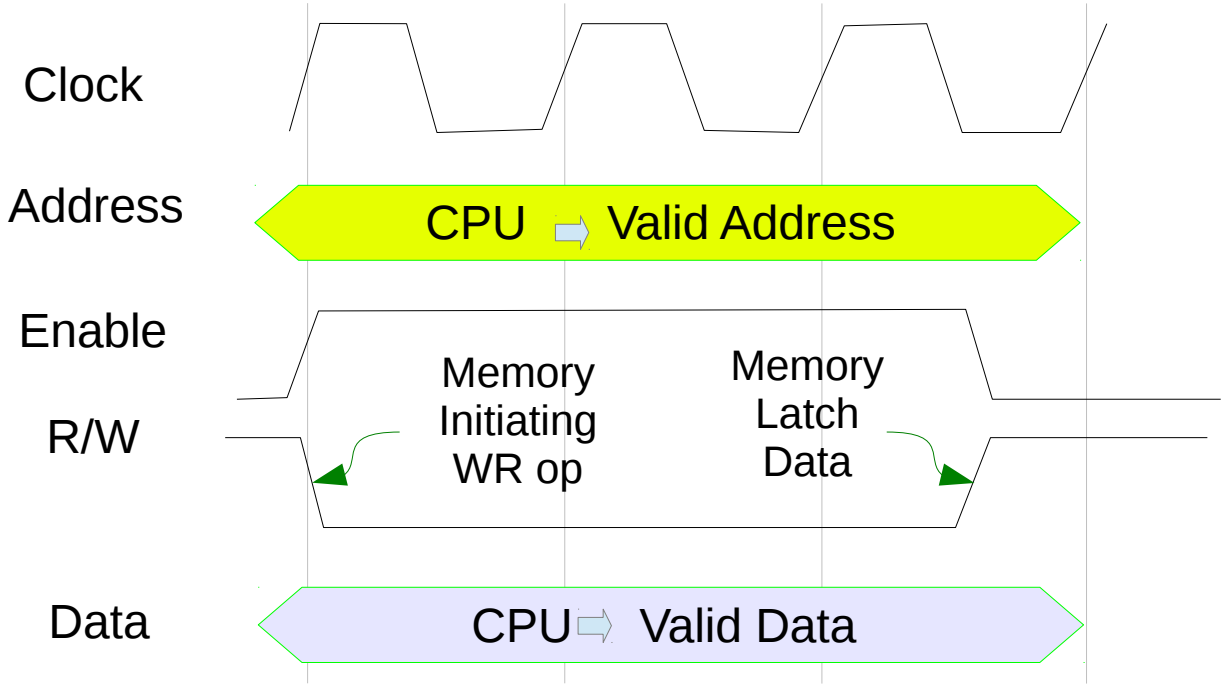


Memory Write Cycle

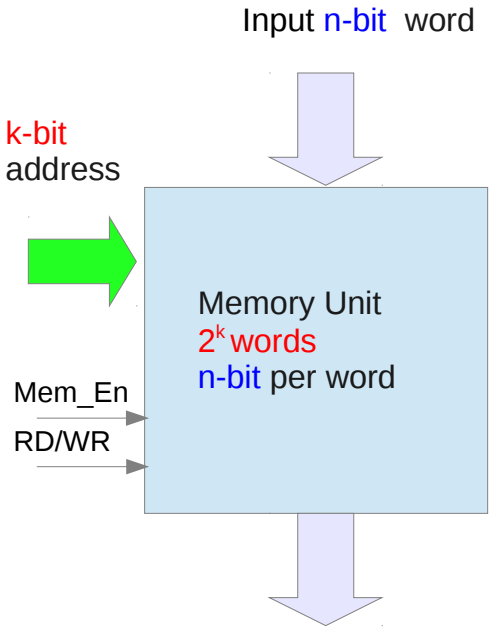


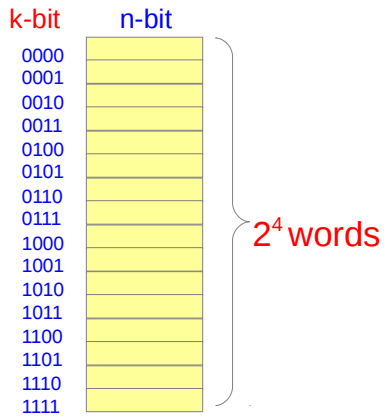


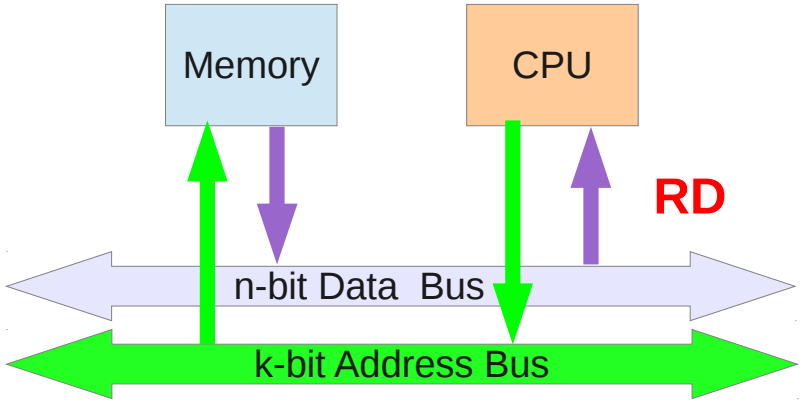


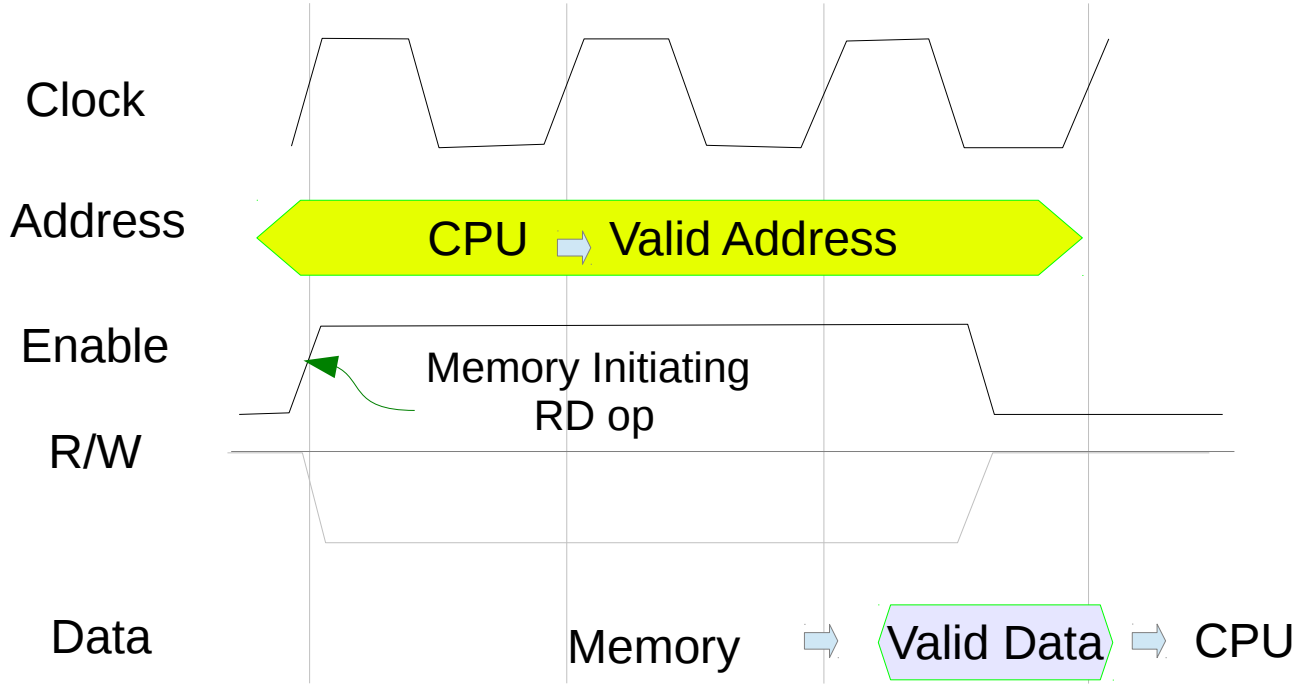


Memory Read Cycle

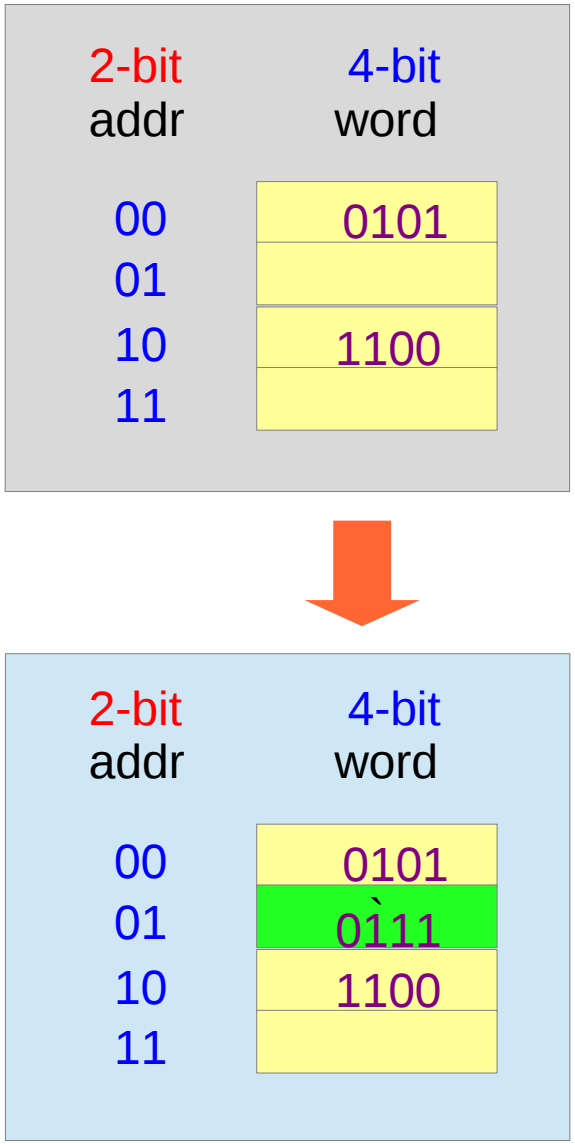


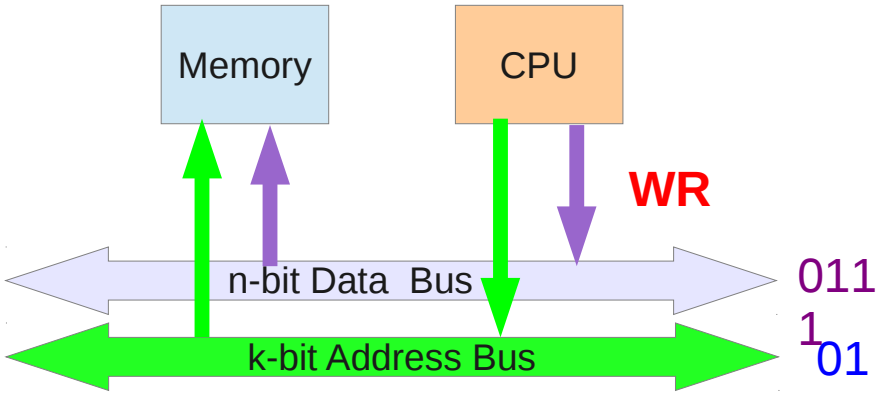


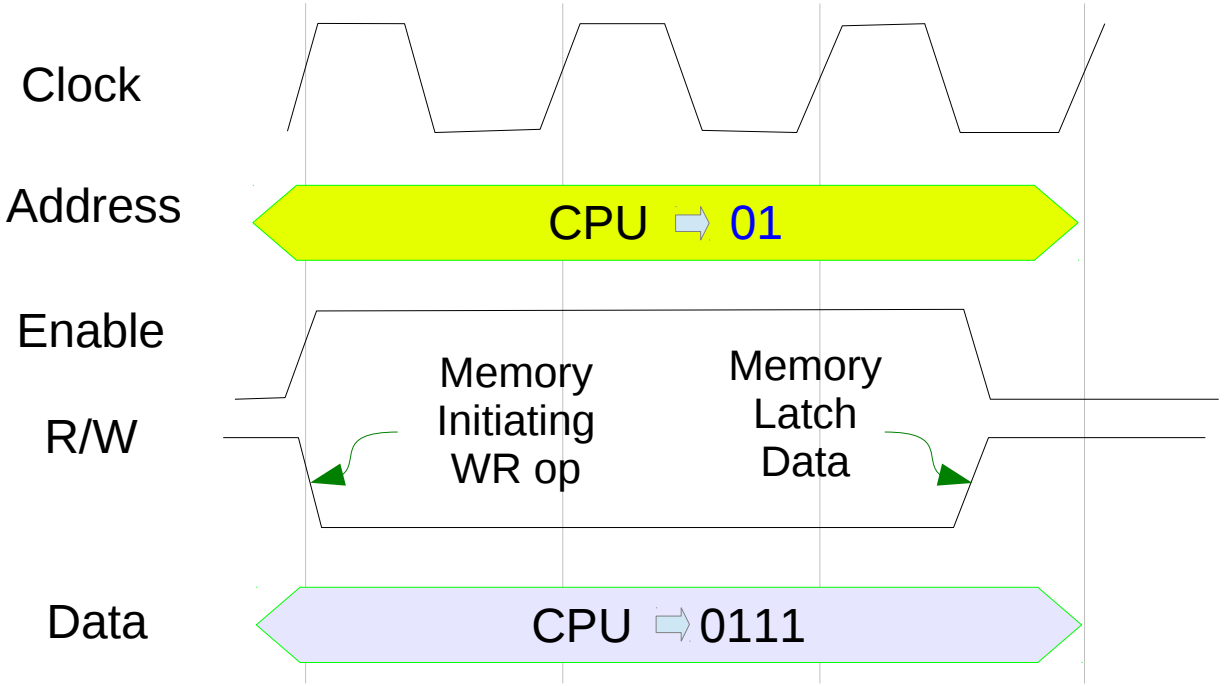




Memory Write Example

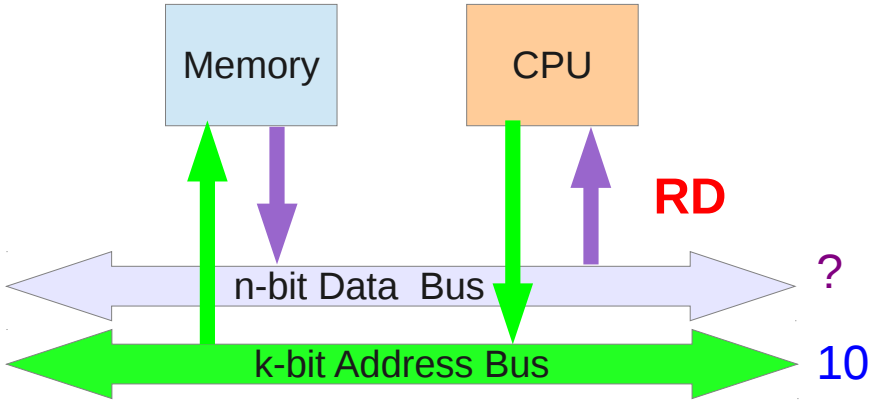


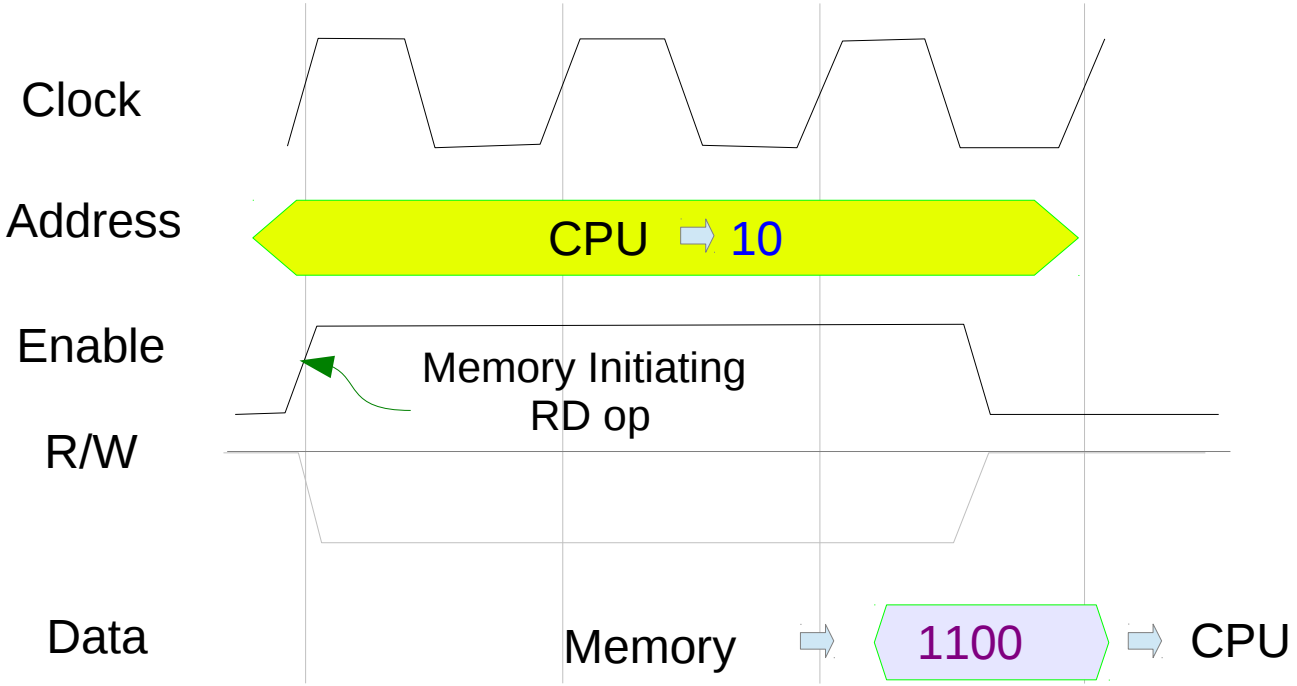




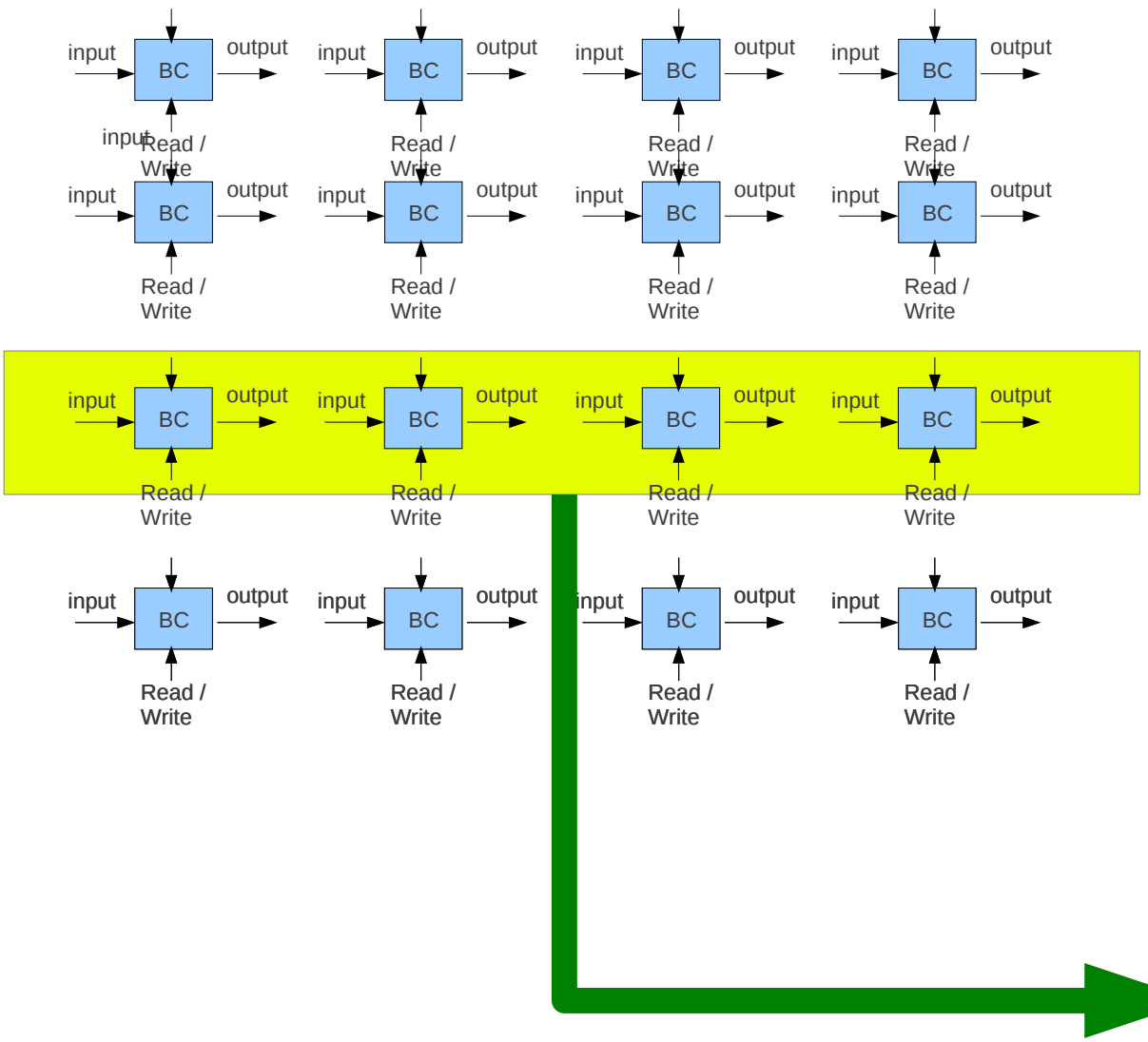
Memory Read Example

2-bit addr	4-bit word
00	0101
01	
10	1100
11	

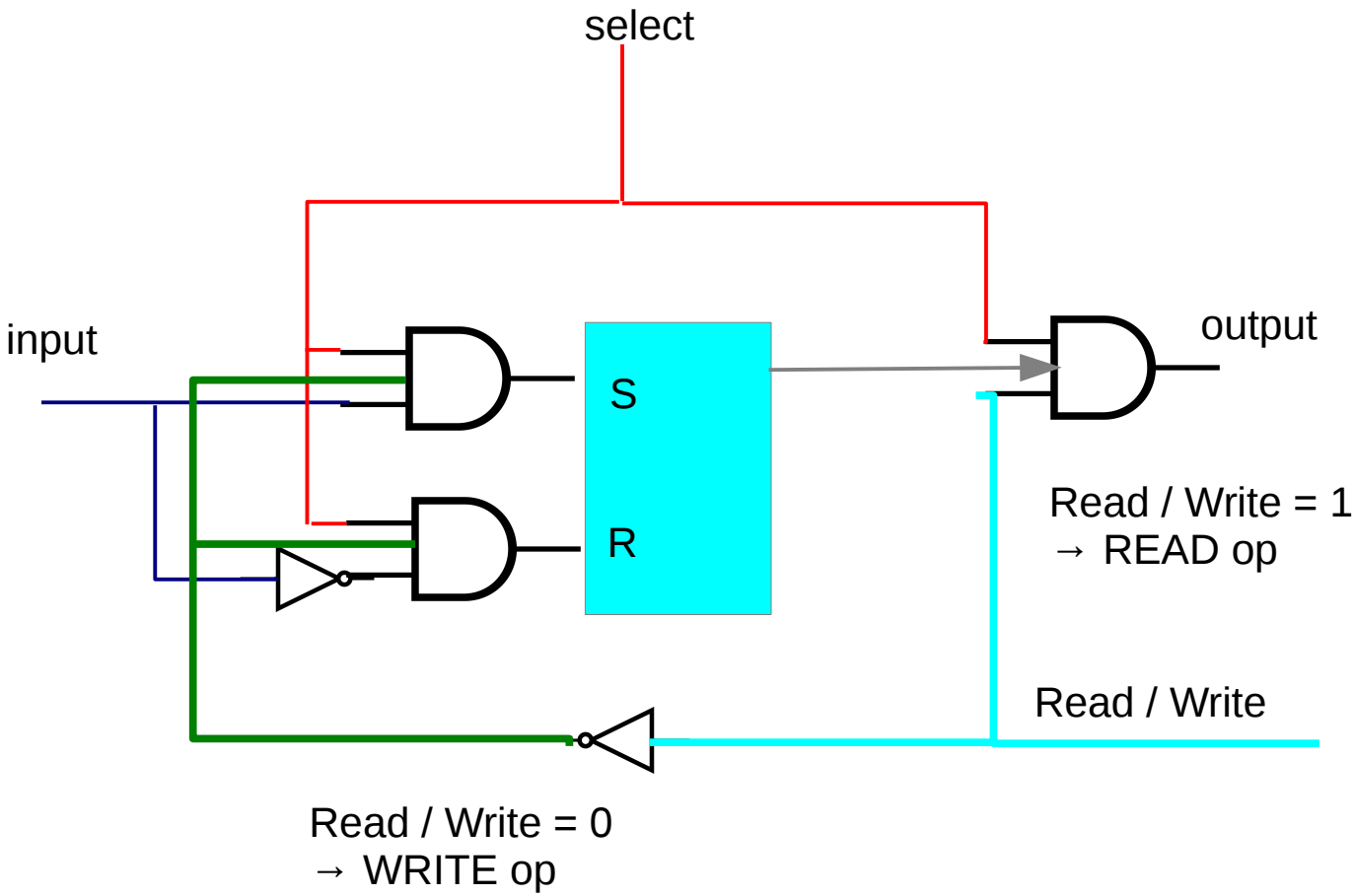


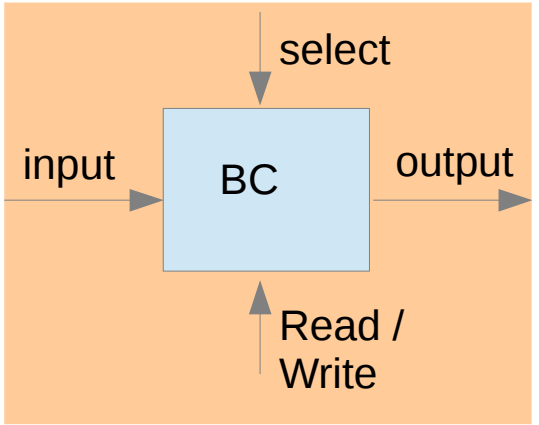


Selecting a Word

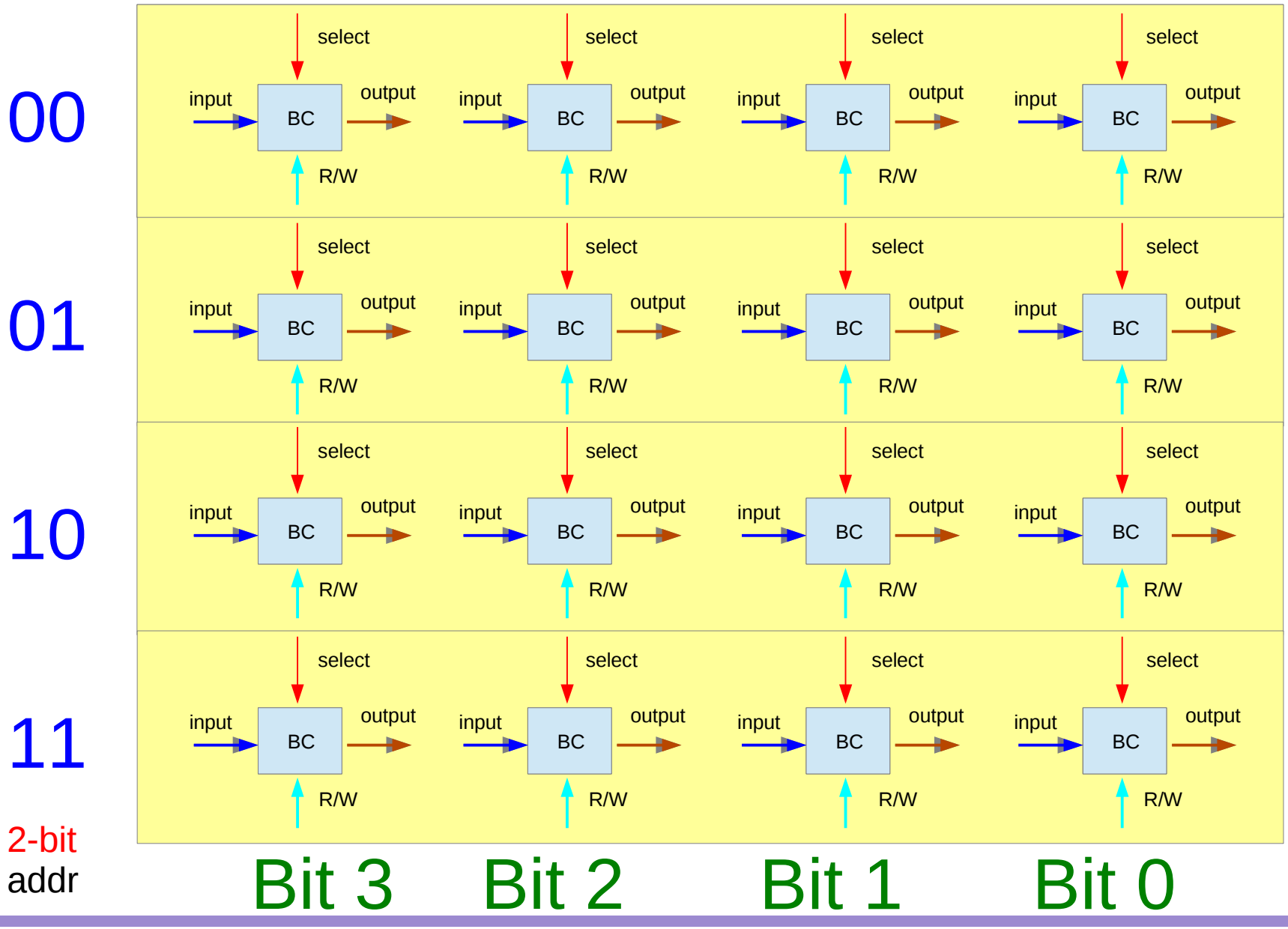


Selecting a Word

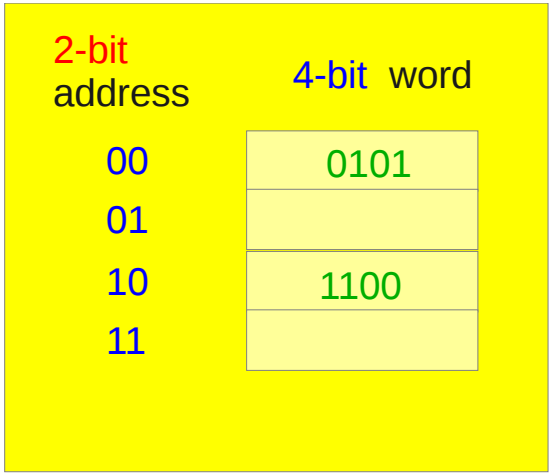


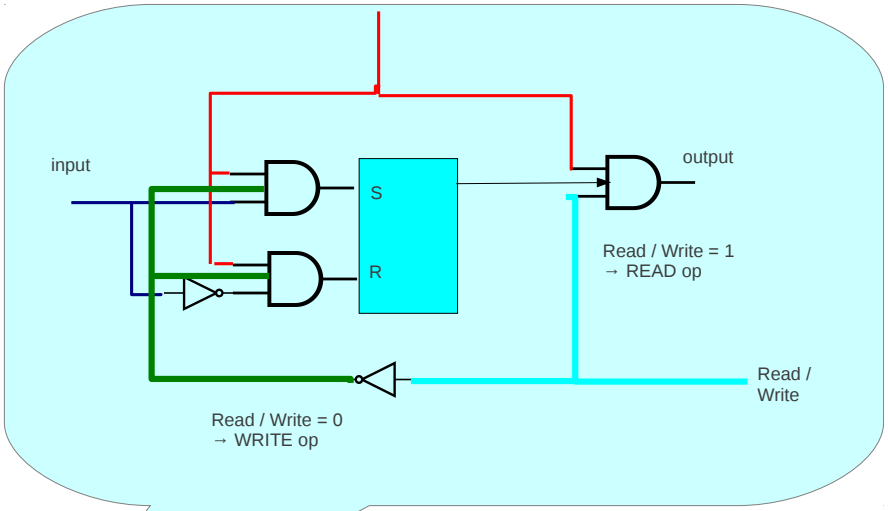


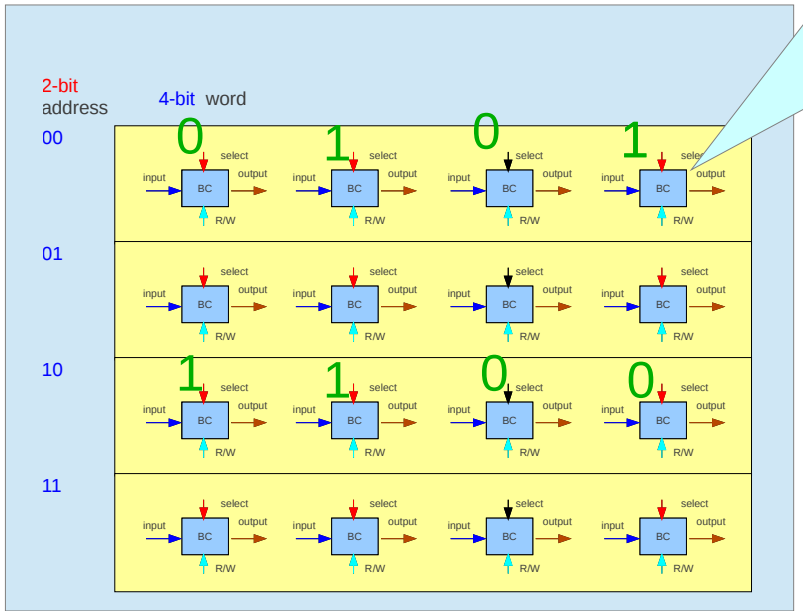
A Memory Map


00
01
10
11
2-bit
addr
select
input
output
BC
R/W
Bit 3
Bit 2
Bit 1
Bit 0

4x4 Memory


2-bit
address
4-bit word
00
01
10
11
0101
1100


input
S
R
output
Read / Write = 1
→ READ op
Read /
Write
Read / Write = 0
→ WRITE op


2-bit
address
4-bit word
00
01
10
11
select
input
BC
output
R/W

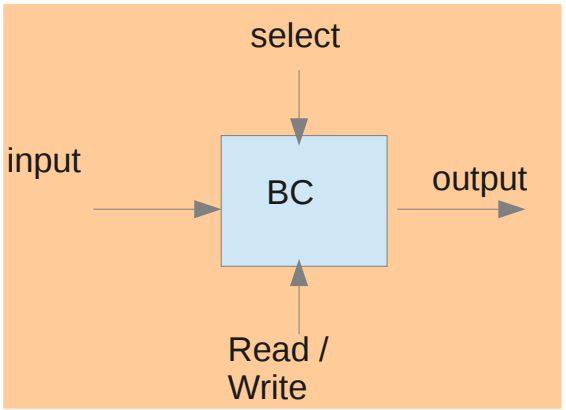
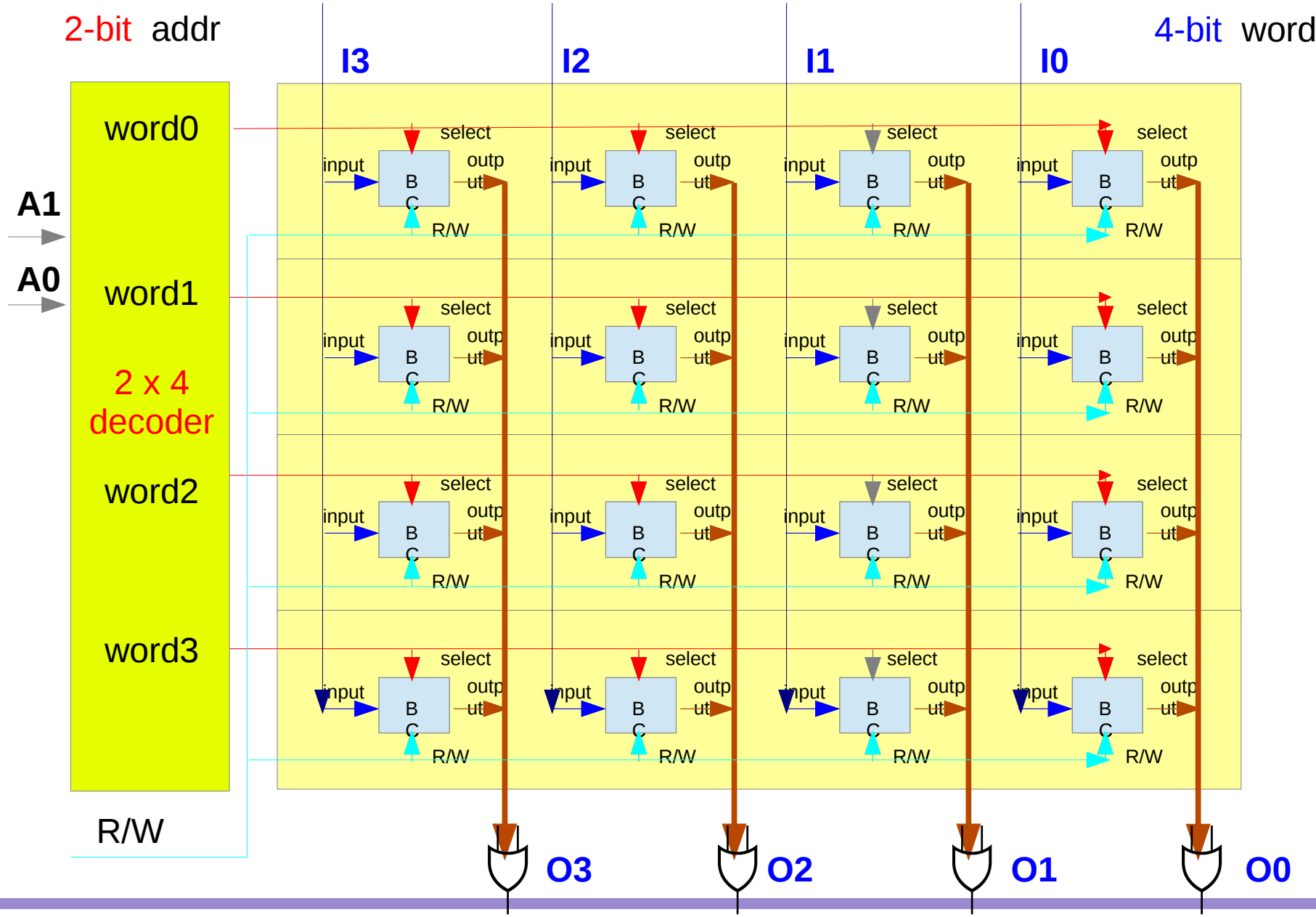
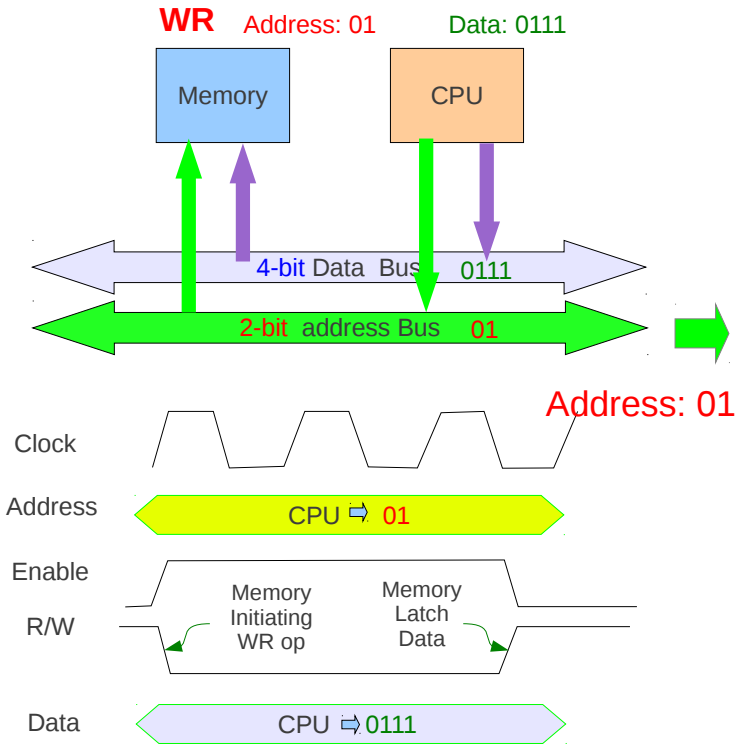

select
input
BC
output
Read /
Write

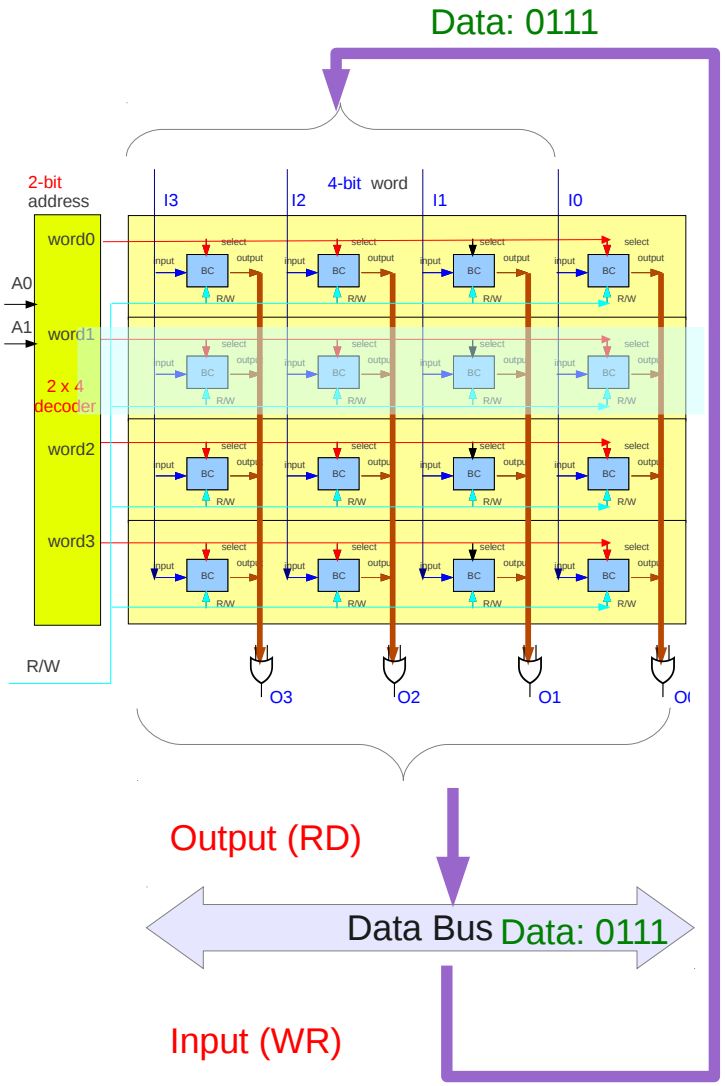
Diagram for a 4x4 Memory



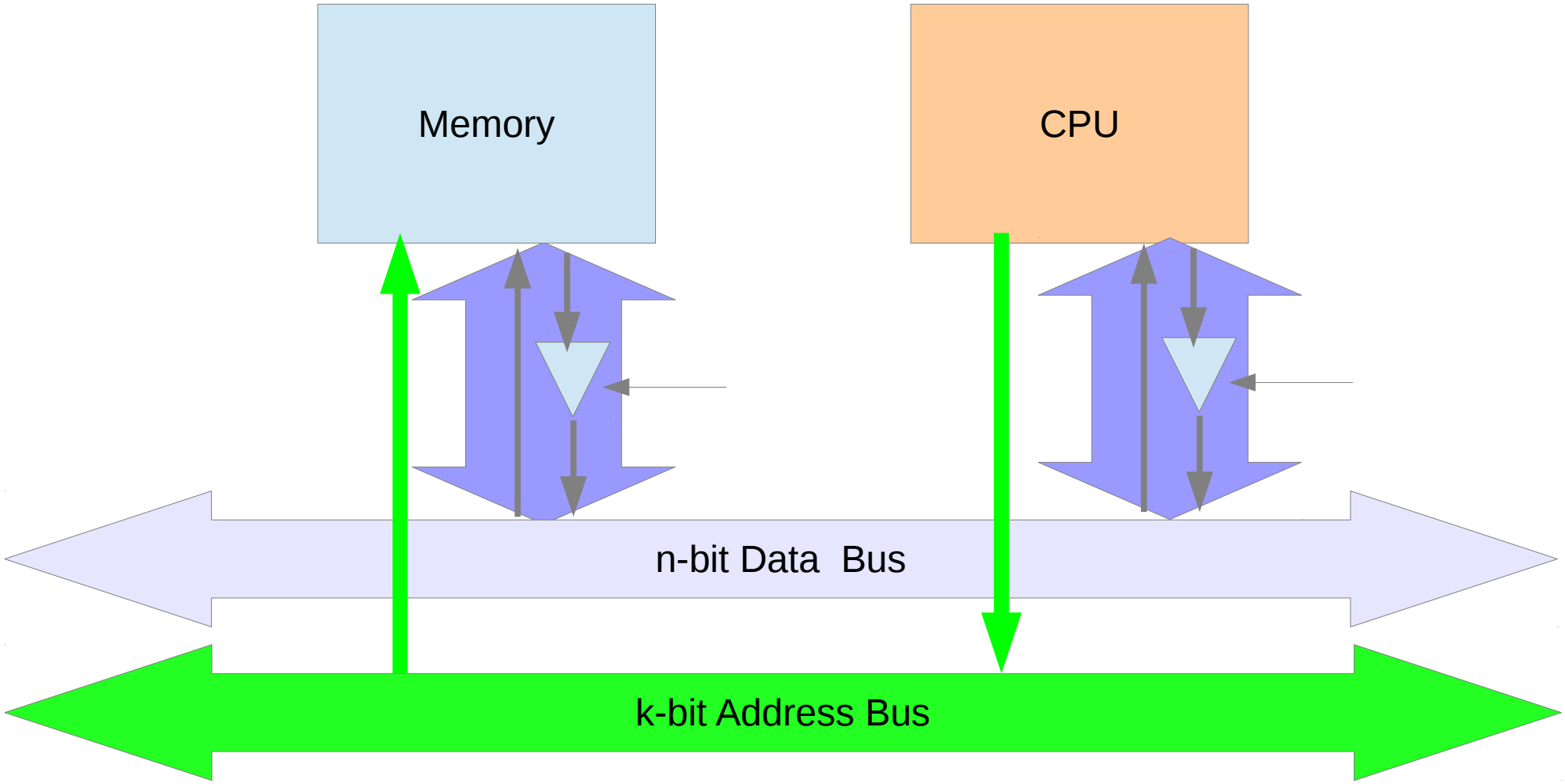
Write Cycle Example for a 4x4 Memory



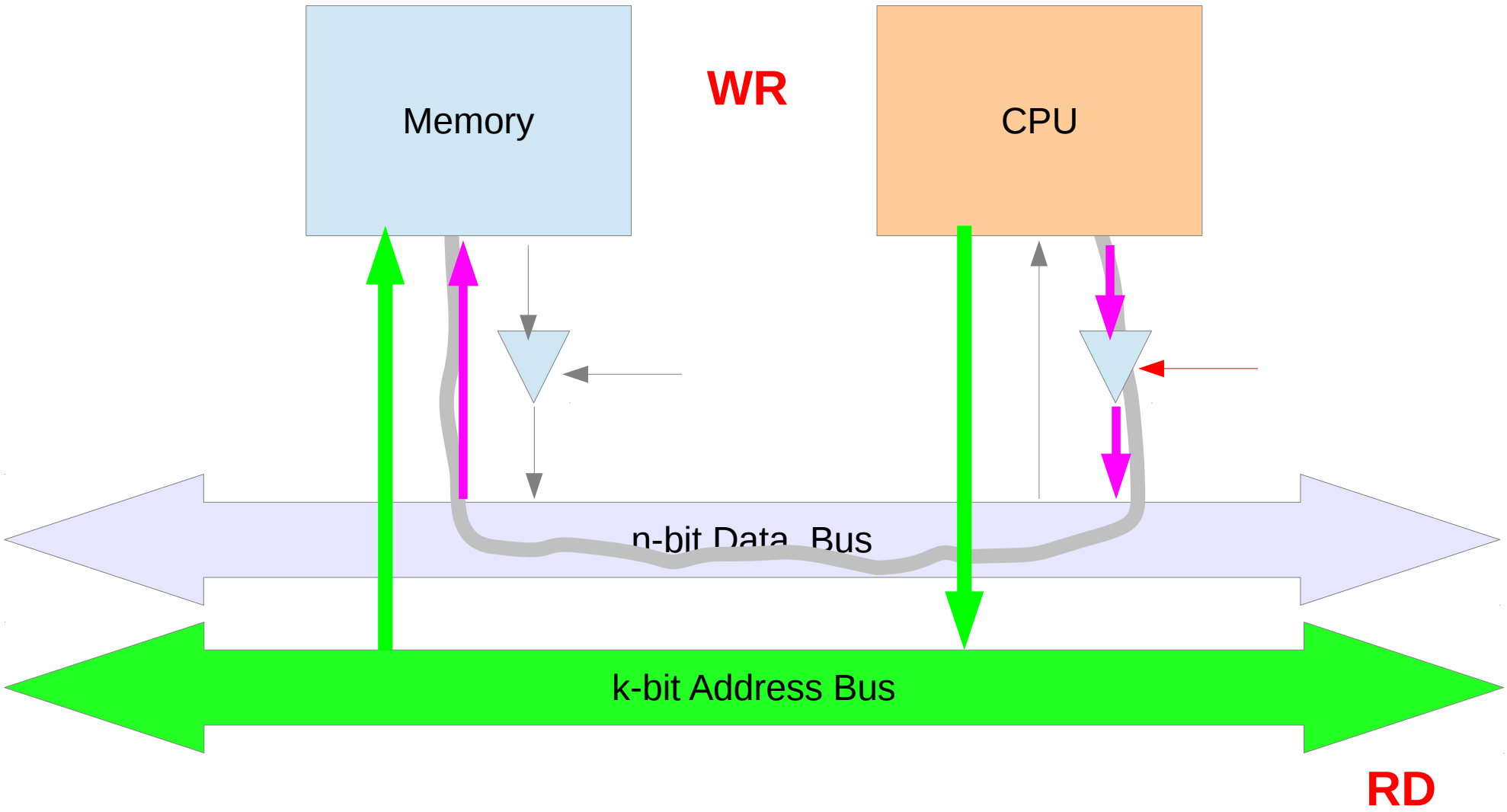
Data & Address Buses



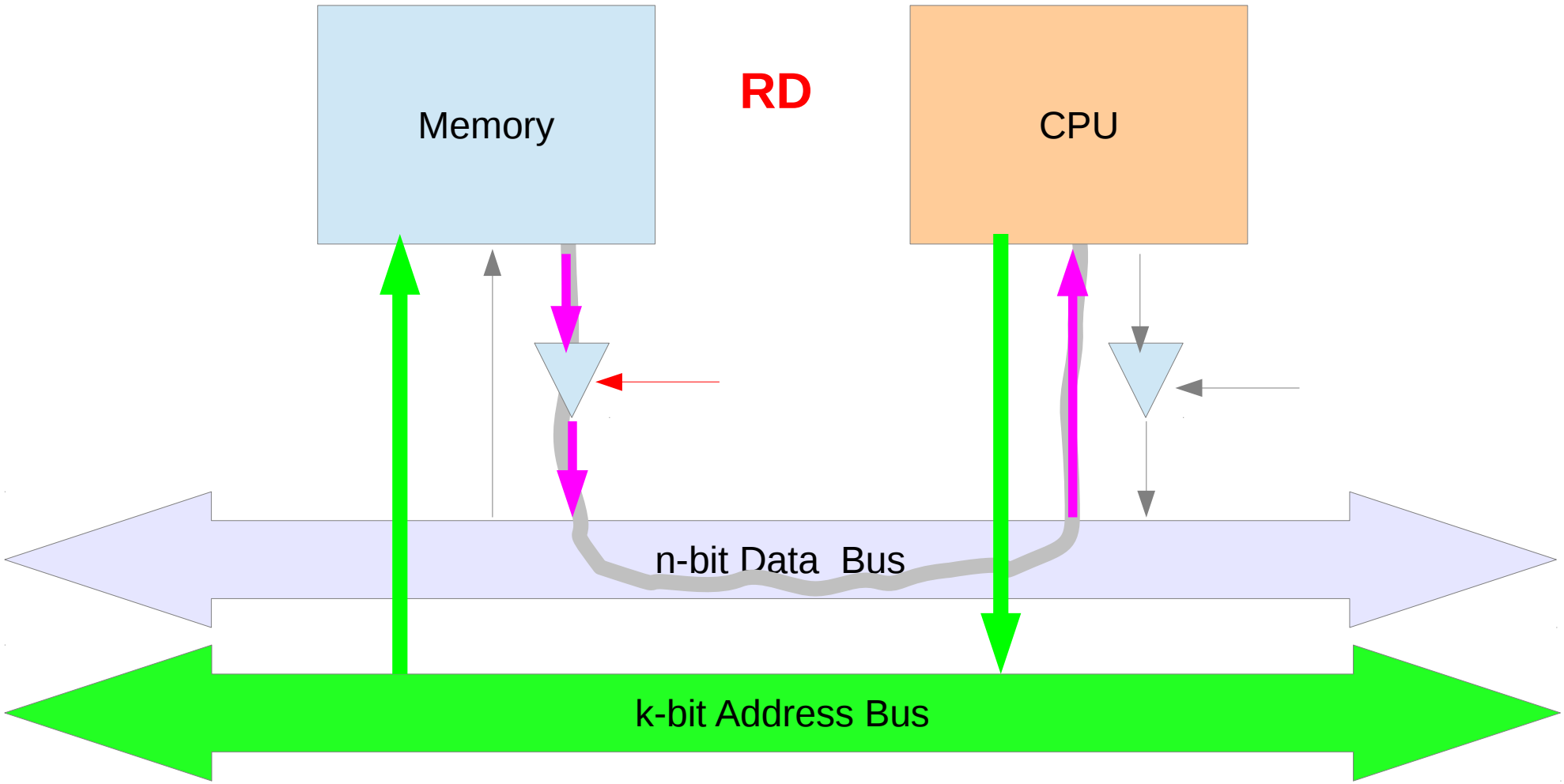
Bi-directional Data Bus



From CPU to Memory

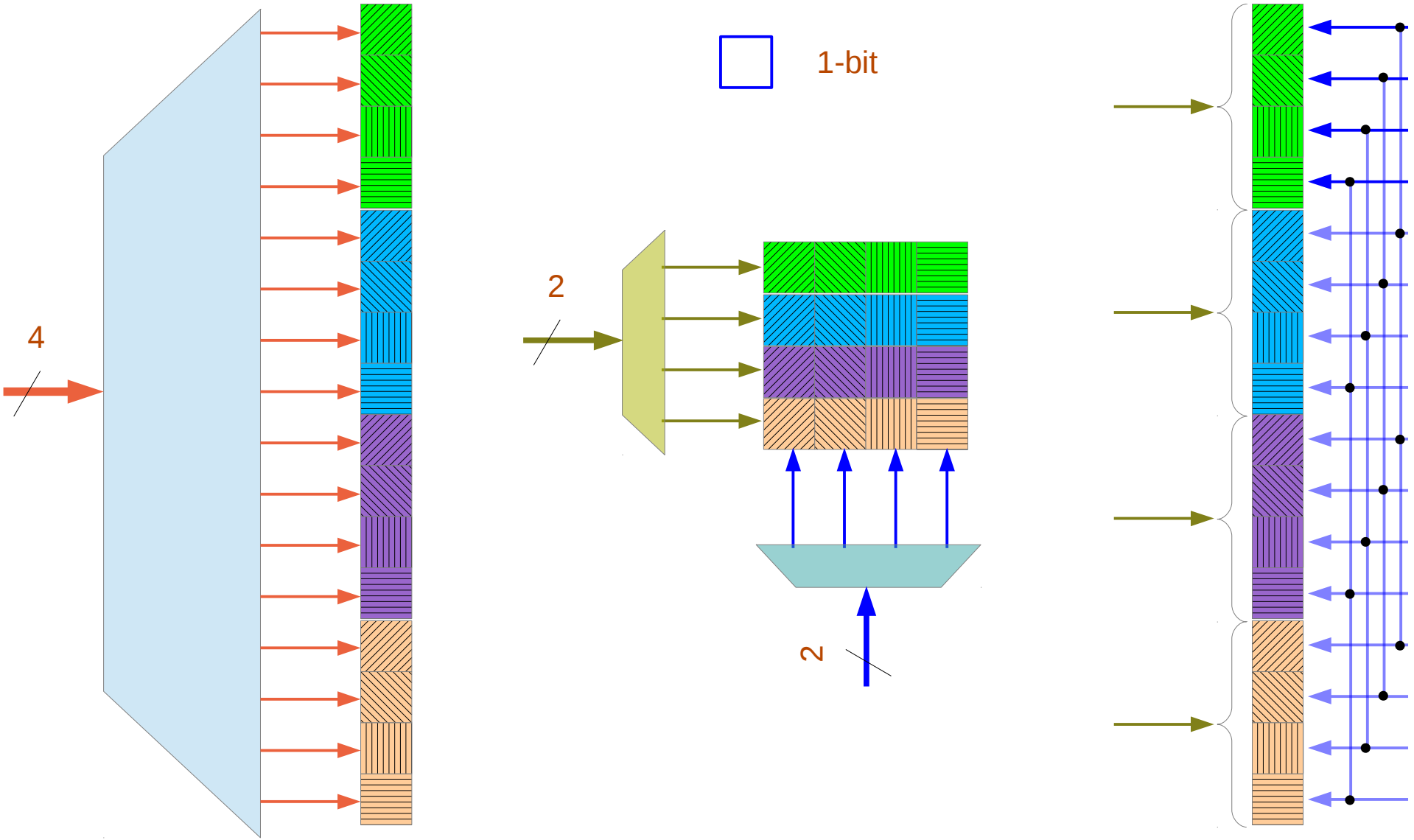


From Memory to CPU

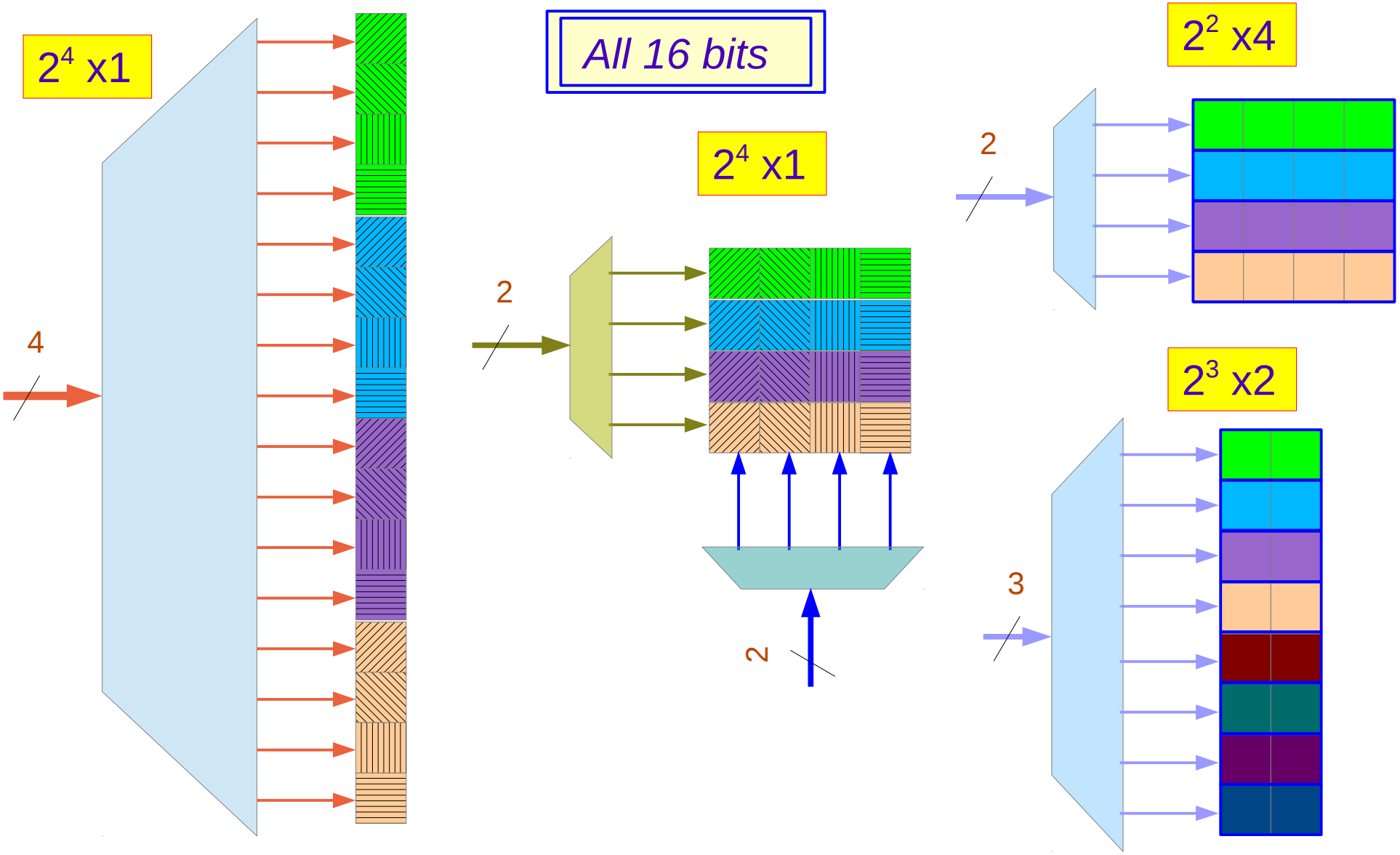


Address Decoding

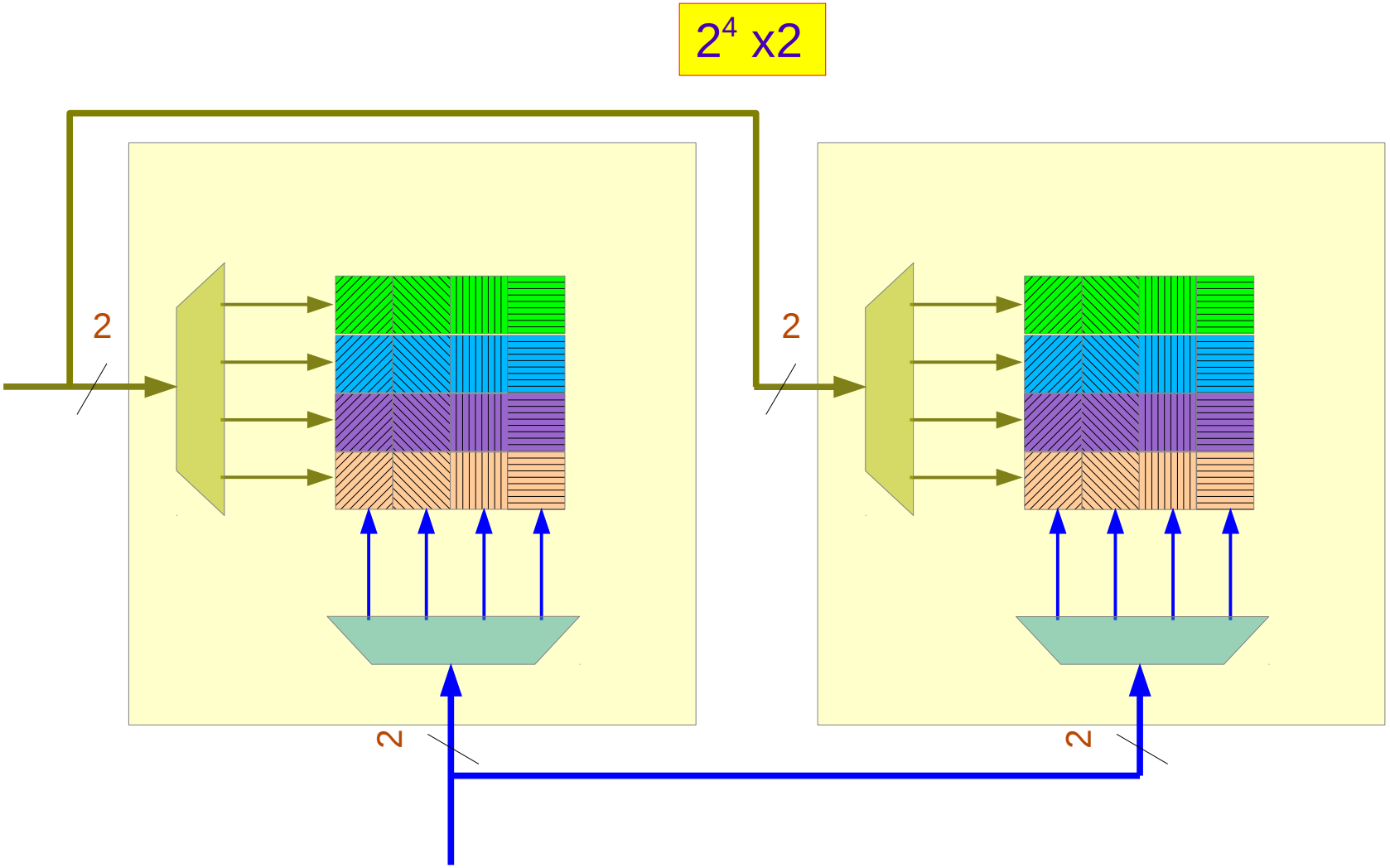
DRAM – Coincidence Selection (1)



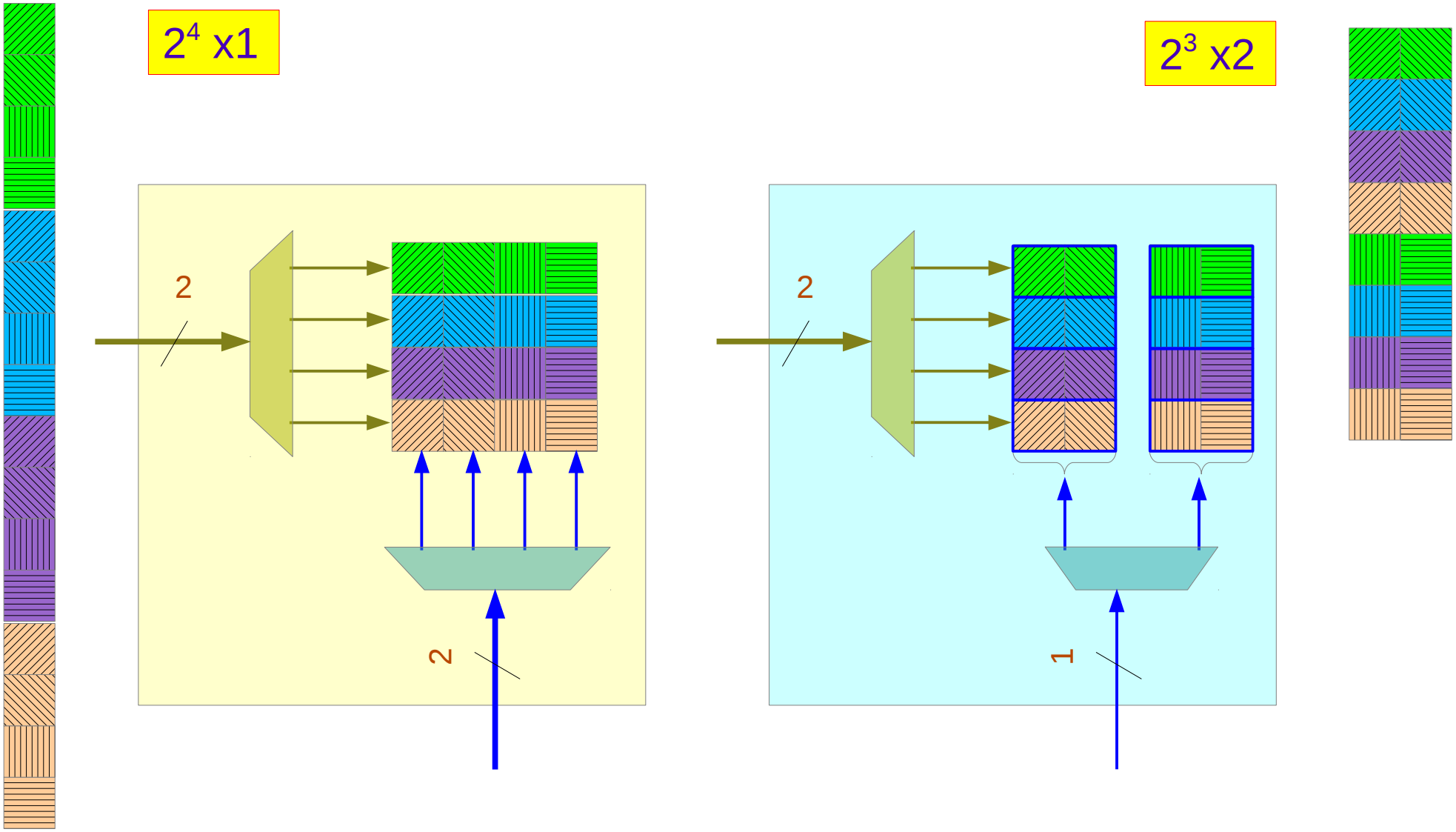
DRAM – Coincidence Selection (2)



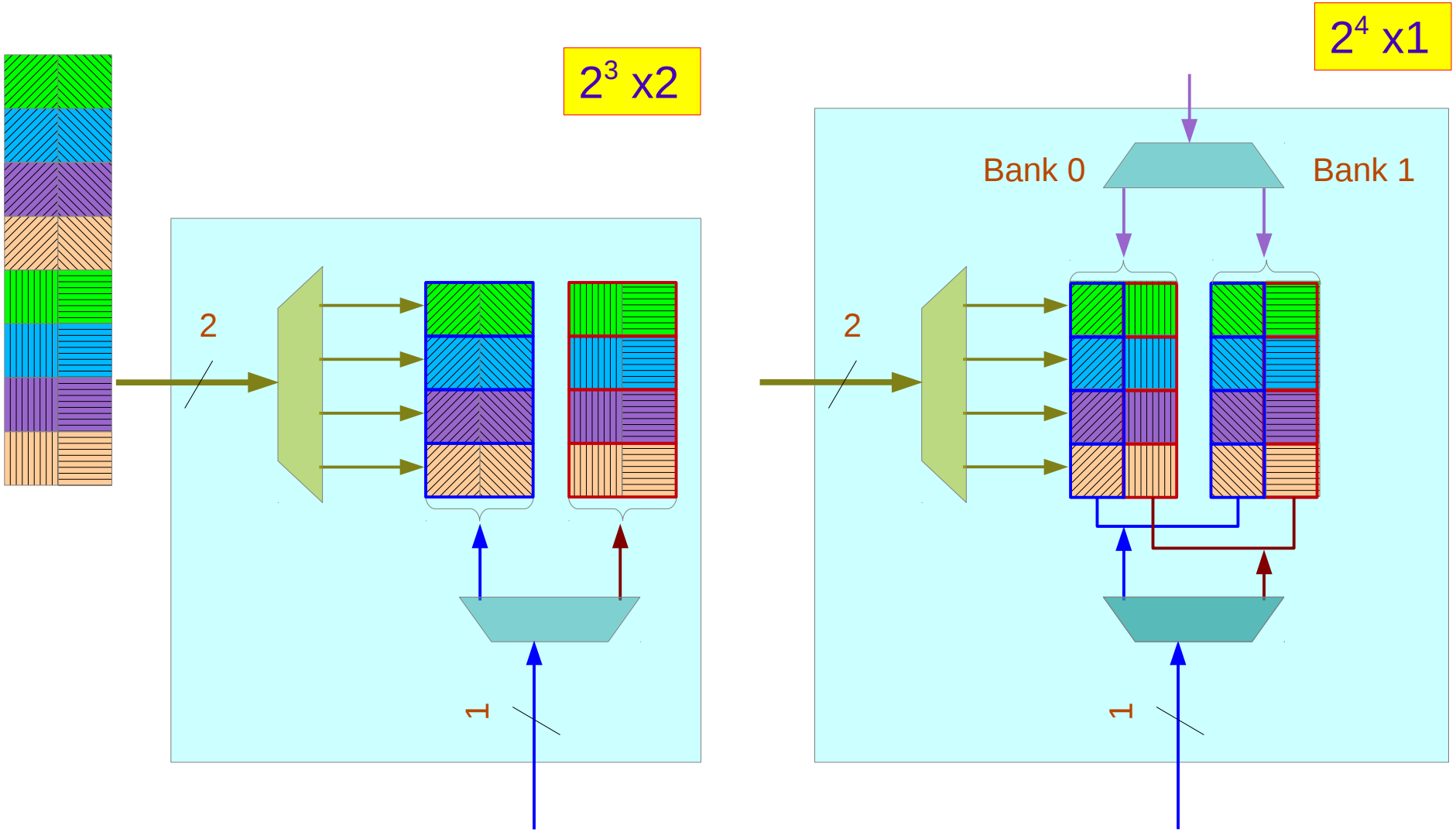
DRAM – Coincidence Selection (3)



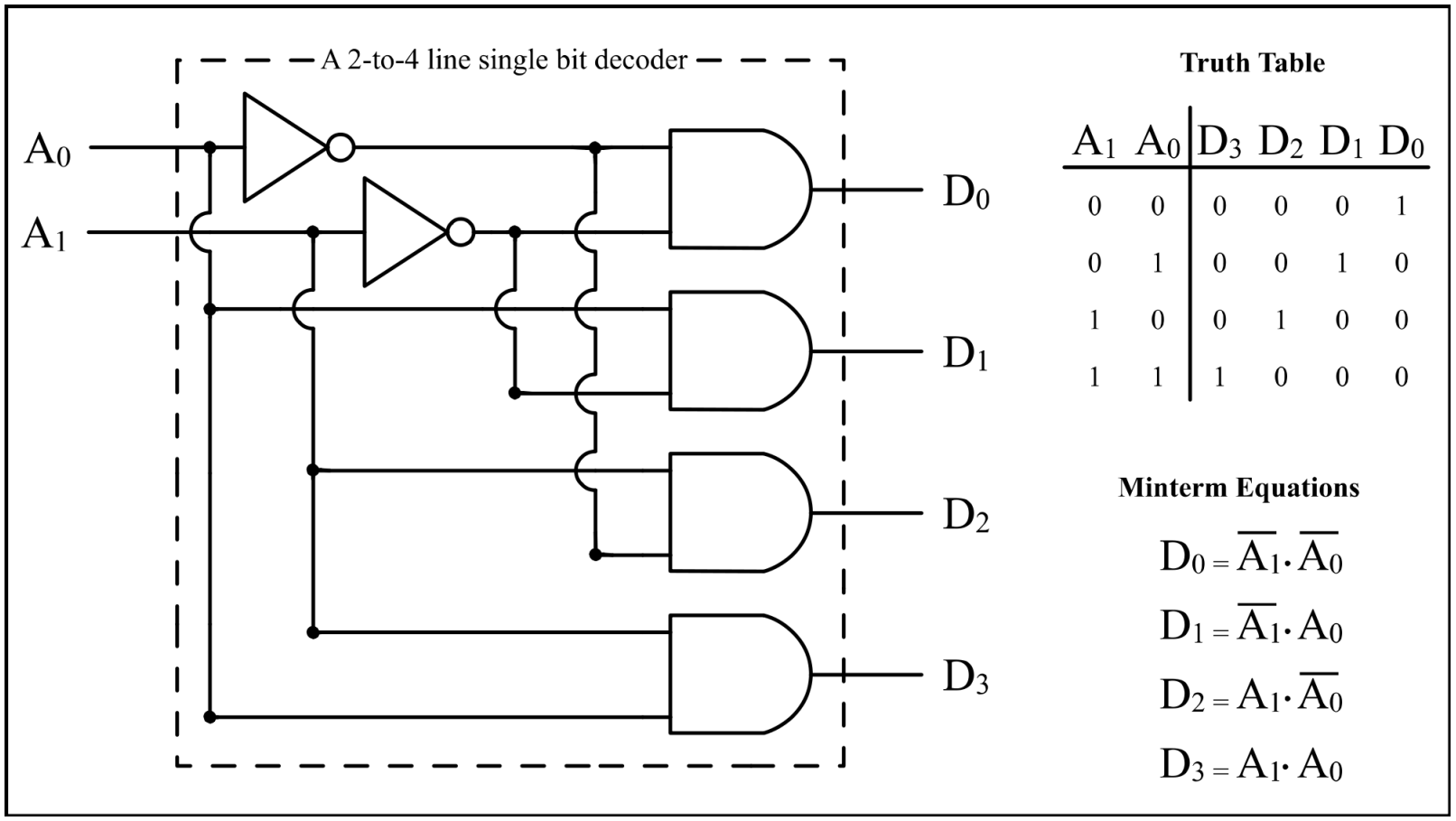
DRAM – Coincidence Selection (4)



DRAM – Coincidence Selection (5)



2 to 4 Decoder



Truth Table

A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Minterm Equations

$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

$$D_2 = A_1 \cdot \overline{A_0}$$

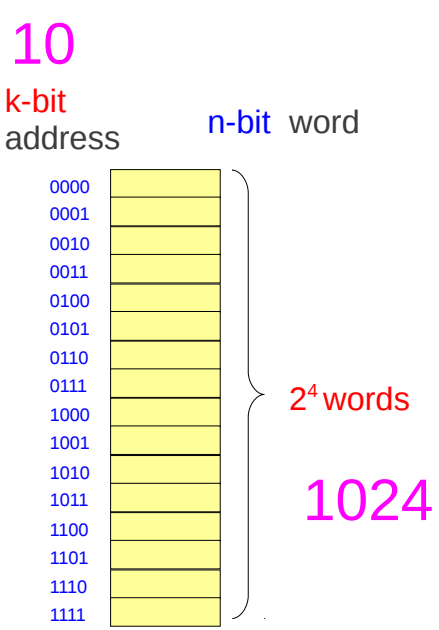
$$D_3 = A_1 \cdot A_0$$

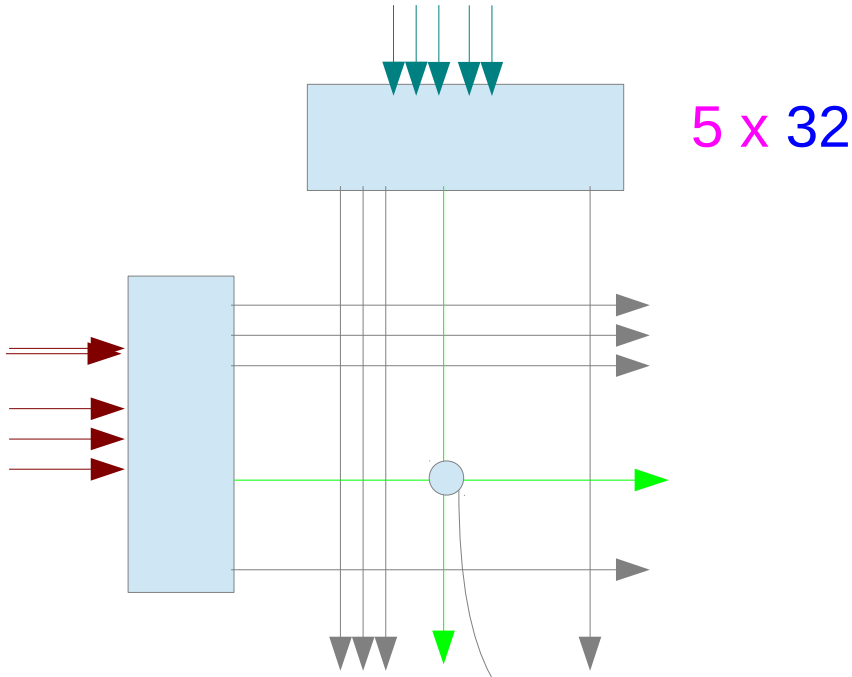
2 x 4 decoder : 4 AND gates

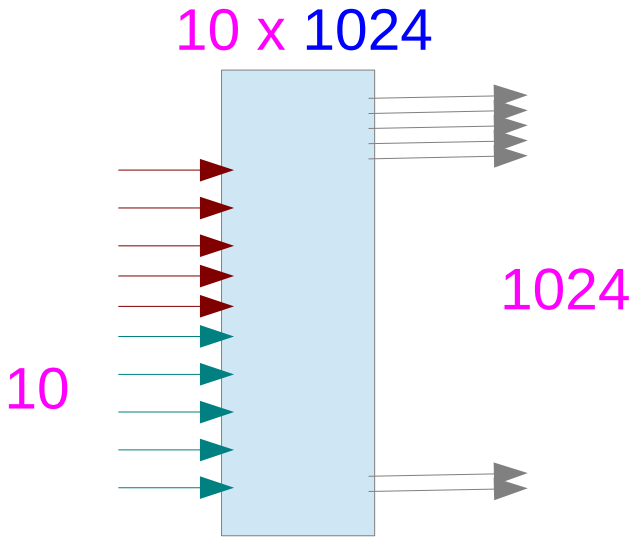
10 x 1024 decoder : 1024 AND gates

4 x 16 decoder : 16 AND gates

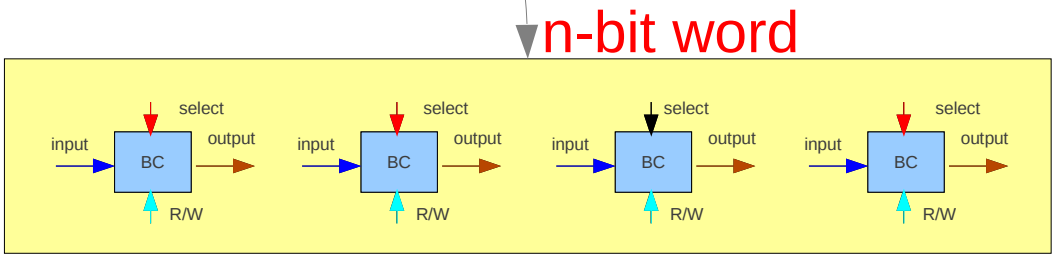
Coincidence Decoder



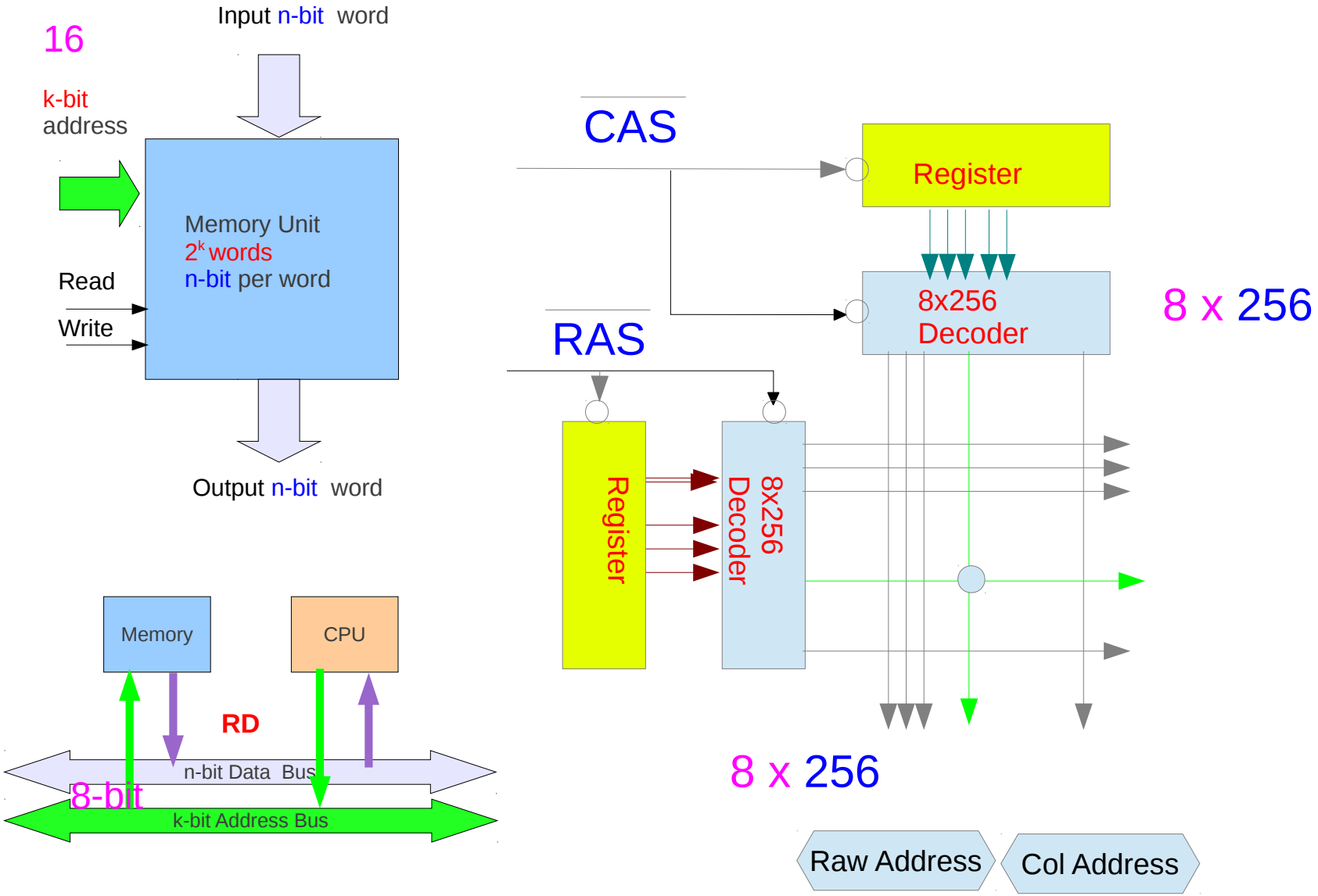




404 → 01100 10110



Time Sharing Address Bus



DRAM – PreCharge

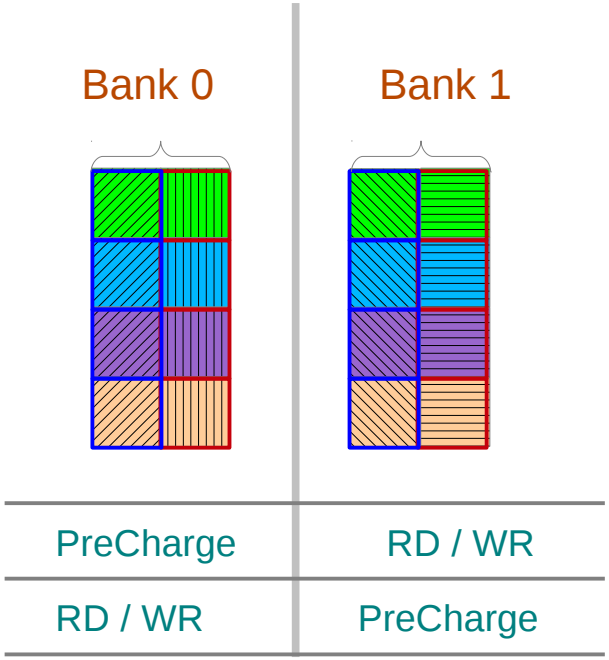
DDR RAM executes commands given by the chipset.
(Activate, Read, Write, Precharge, etc)

To activate a bank with a row for reading or writing,
the bank is to be charged.

This amplifies the signal from that row for a read

A memory bank is usually precharged
without waiting for a read/write request and then charging.

Precharging one memory bank can usually be overlapped
with **accessing** another memory bank.

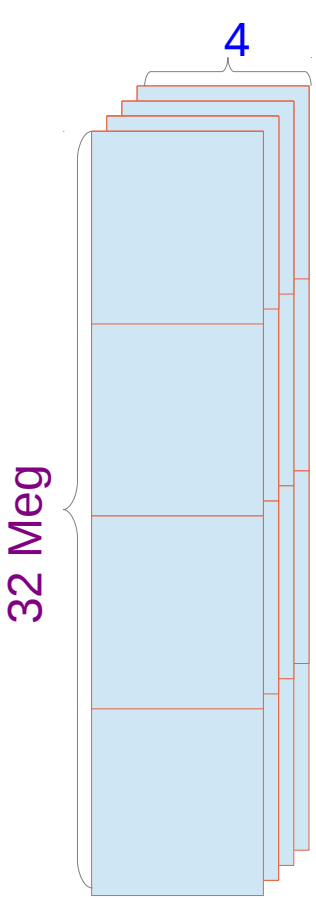


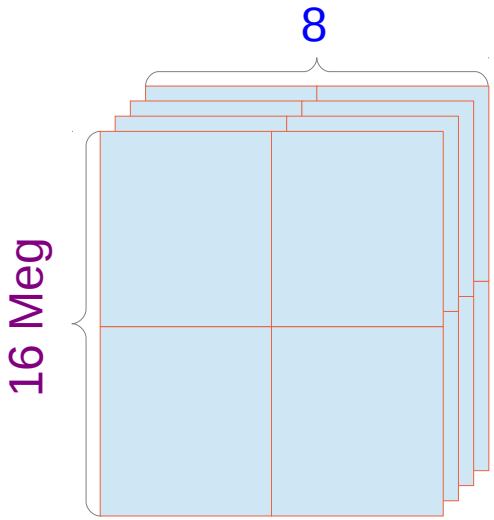
DRAM – 512Mb Example

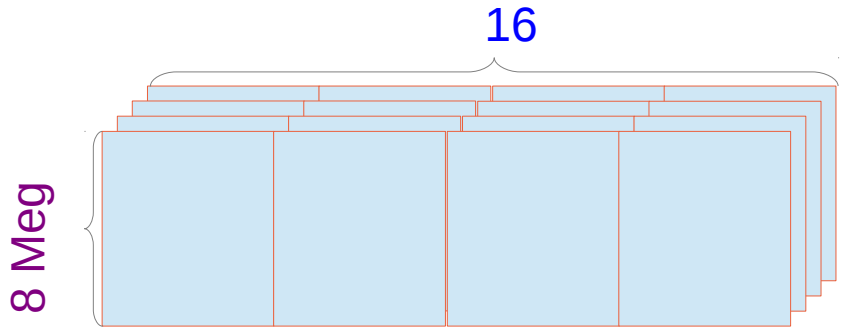
MT46V128M4 – 32 Meg x 4 x 4 banks
MT46V64M8 – 16 Meg x 8 x 4 banks
MT46V32M16 – 8 Meg x 16 x 4 banks

2^{25} x 2^{2} x 4 banks
2^{24} x 2^{3} x 4 banks
2^{23} x 2^{4} x 4 banks

All 512 Mbits







2^{27} x 4 banks = 128 Meg x 4 banks

2^{25} x 2^{2} x 4 banks
2^{24} x 2^{3} x 4 banks
2^{23} x 2^{4} x 4 banks

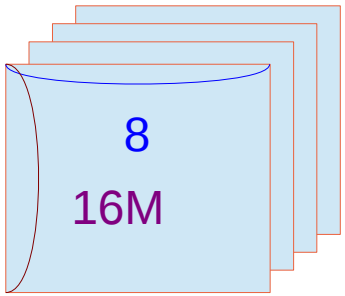
Row: 13 bits Col: 12 bits Bank: 2 bits Data: 4 bits
Row: 13 bits Col: 11 bits Bank: 2 bits Data: 8 bits
Row: 13 bits Col: 10 bits Bank: 2 bits Data: 12 bits

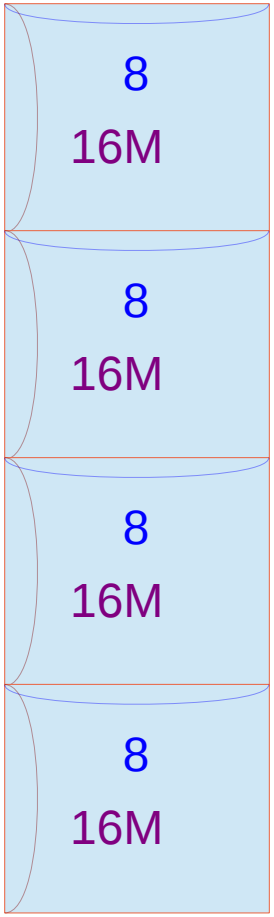
DIMM Example (1)

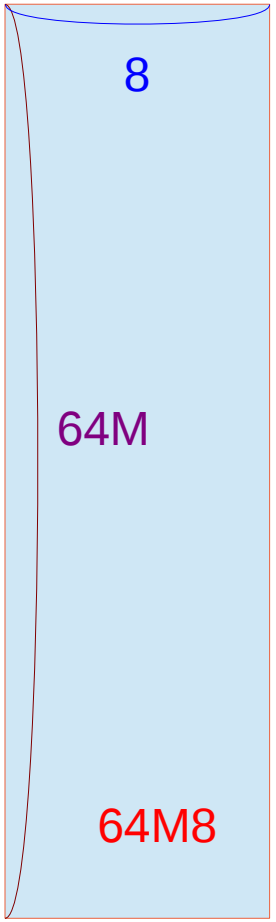
MT46V64M8 – 16 Meg x 8 x 4 banks 2^{24} x 2^{3} x 4 banks

All 512 Mbits

Row: 13 bits Col: 11 bits Bank: 2 bits Data: 8 bits







DIMM Example (2)

MT46V64M8 – 16 Meg x 8 x 4 banks 2^{24} x 2^{3} x 4 banks

All 512 Mbits

Row: 13 bits Col: 11 bits Bank: 2 bits Data: 8 bits

128M x 8 Bytes = $\mathbf{2^{27}}$ **x** $\mathbf{2^{3}}$ Bytes = 1 GBytes

Access Cycles

Memory Types

RAM (Random Access Memory)

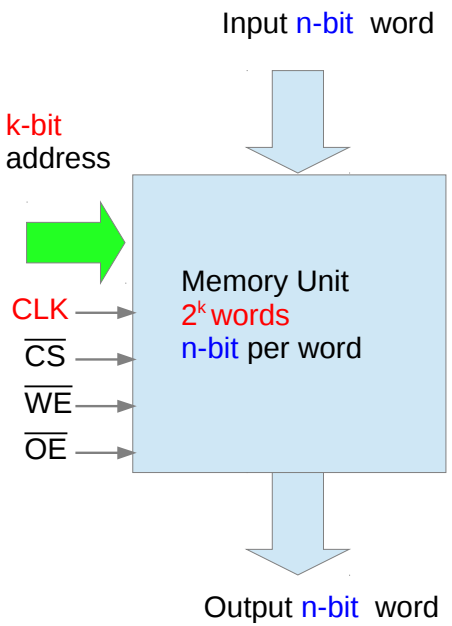
SRAM (Static RAM) – Fast
Synchronous SRAM
Asynchronous SRAM

DRAM (Dynamic RAM) – High Density

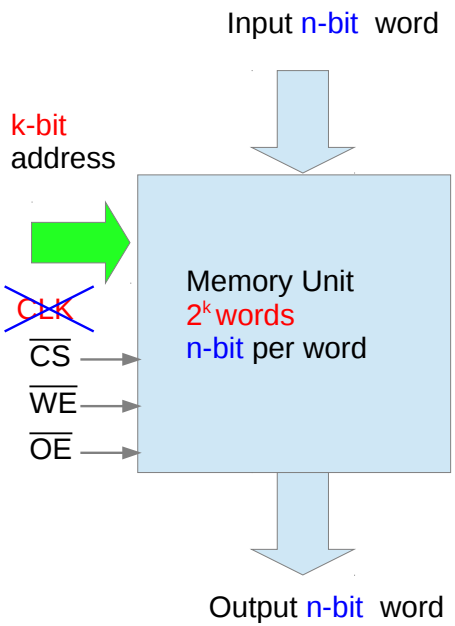
ROM (Read Only Memory)

Memory Types

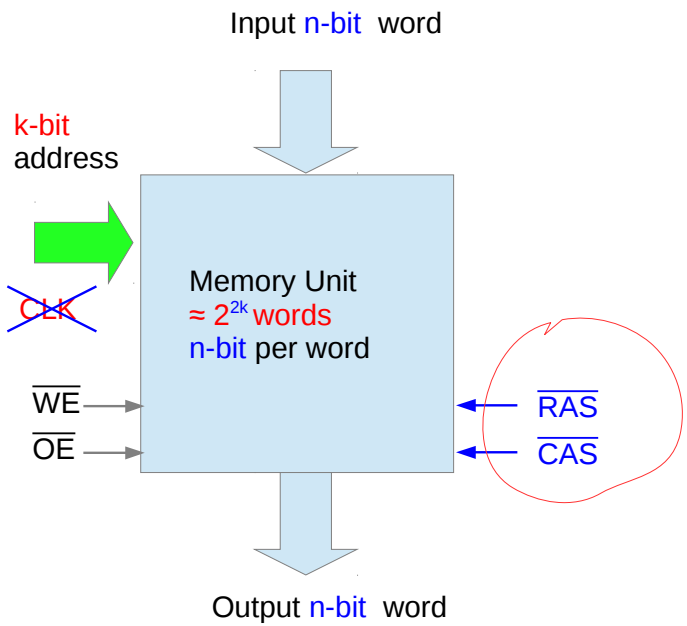
Synchronous SRAM



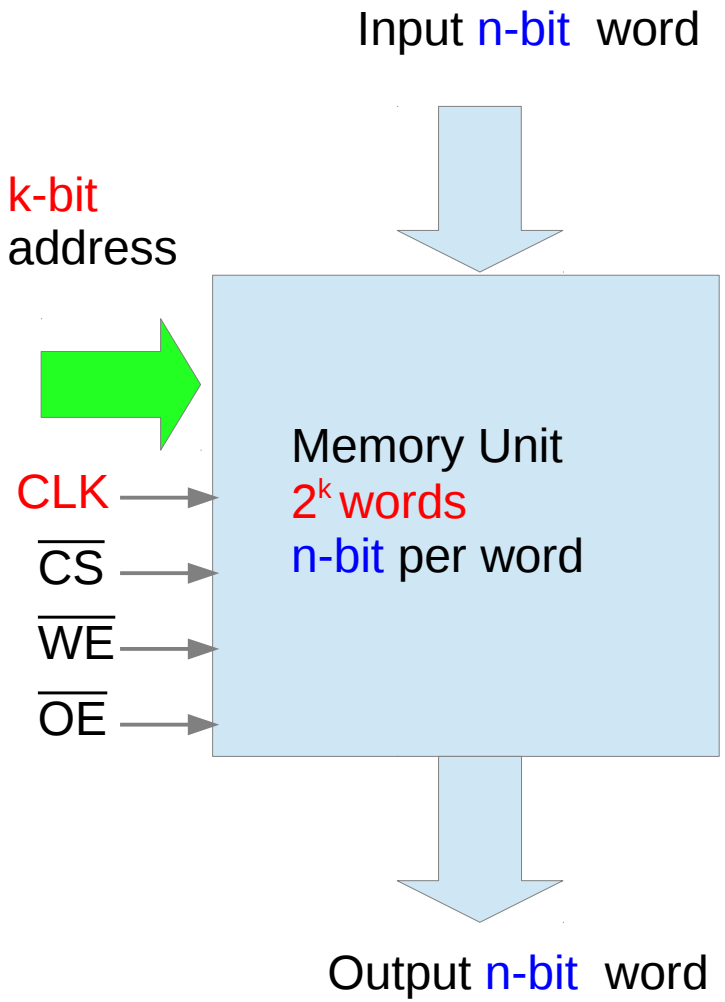
Asynchronous SRAM



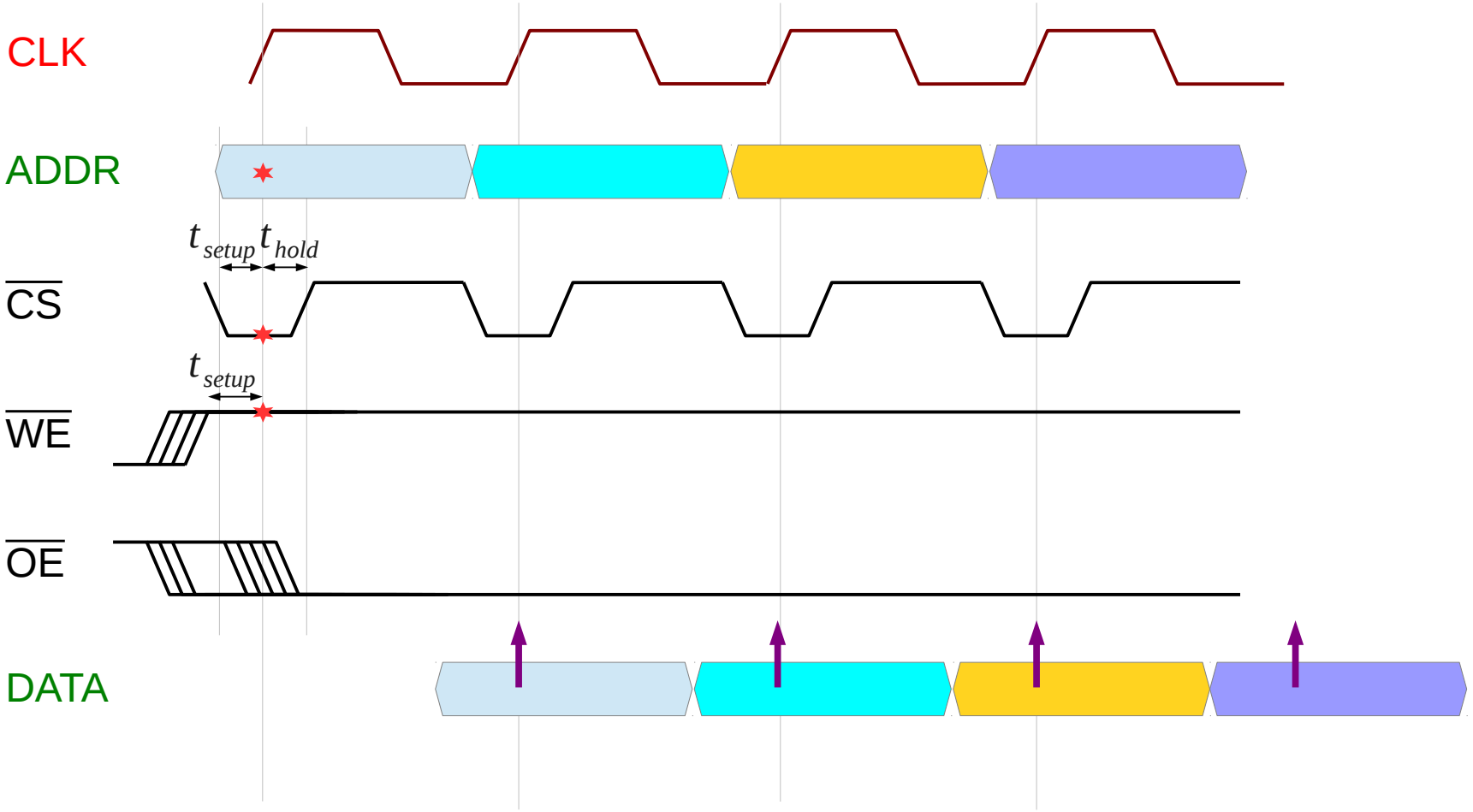
DRAM



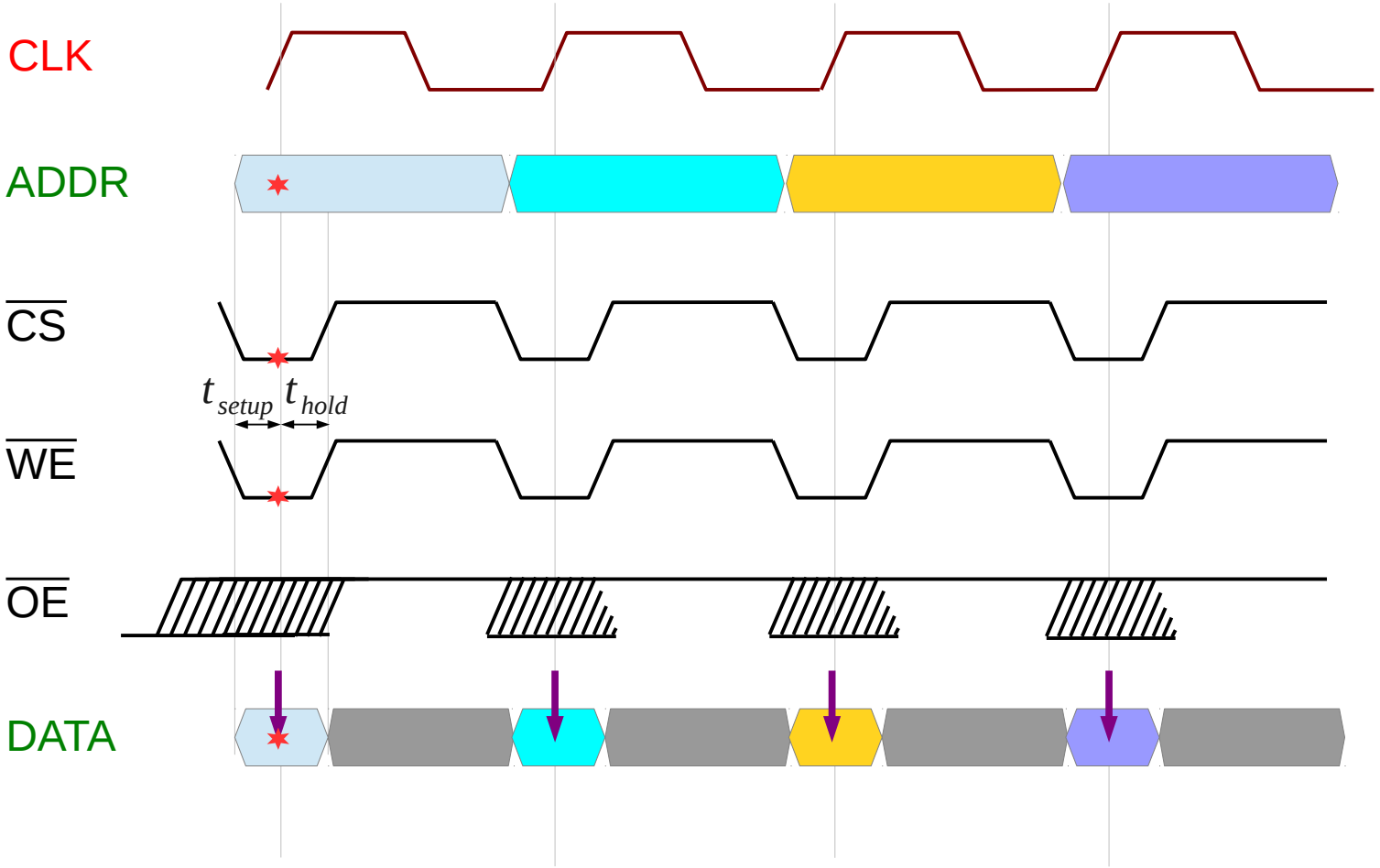
Synchronous SRAM



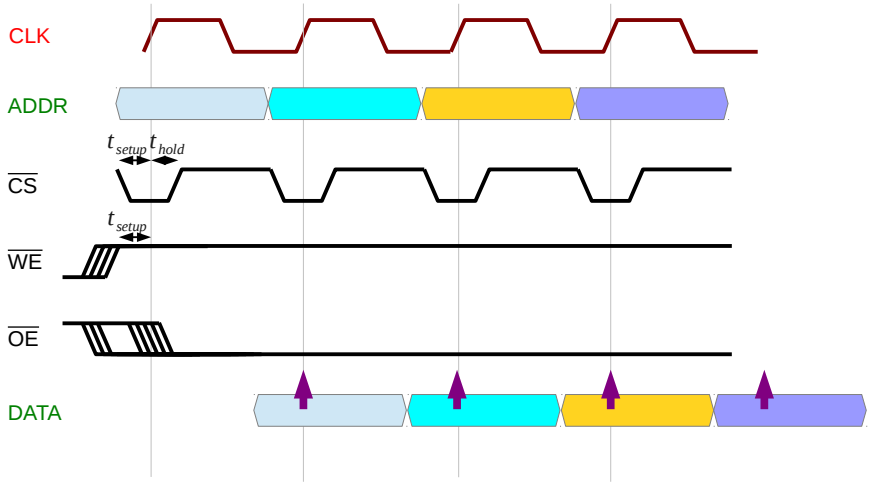
Synchronous SRAM Read Cycle



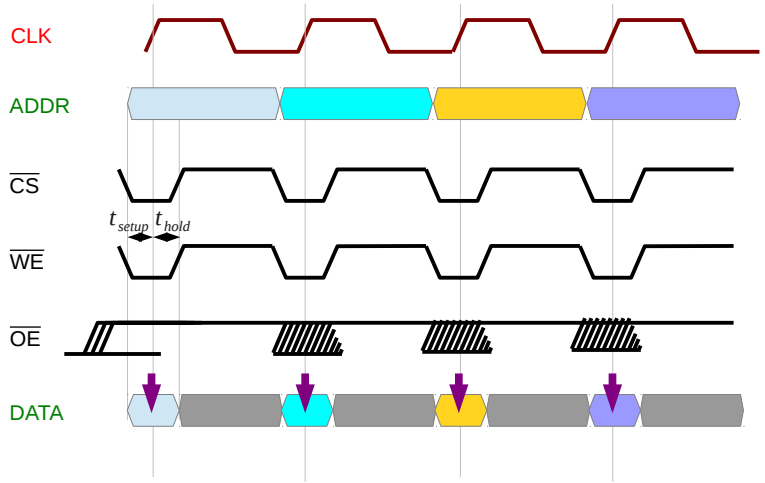
Synchronous SRAM Write Cycle



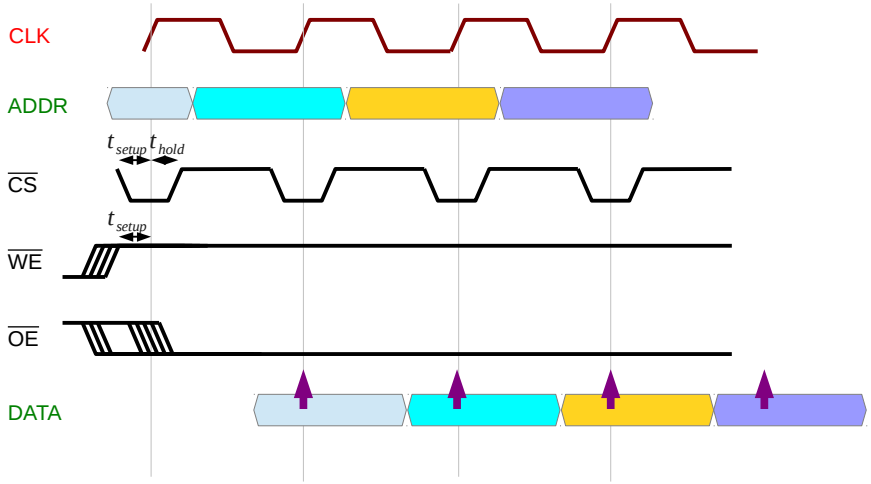
Synchronous SRAM Cycle

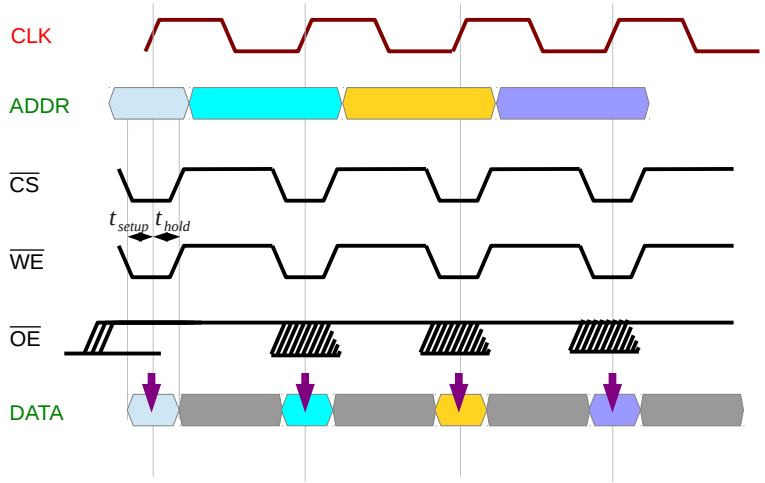


Sync SRAM RD Cycle

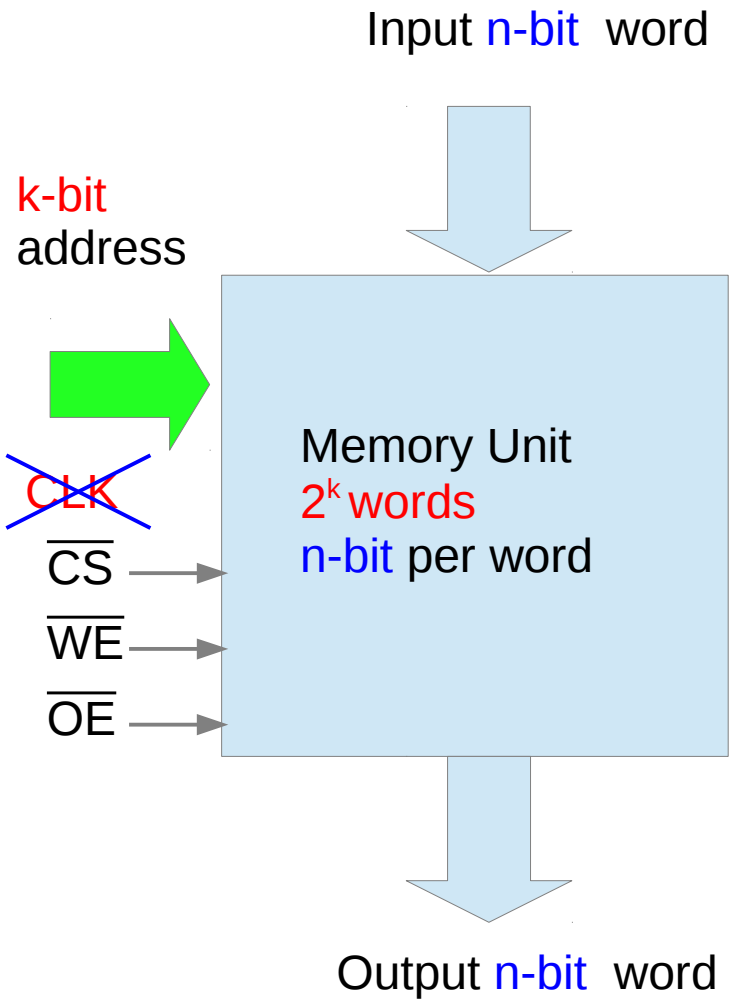


Sync SRAM WR Cycle



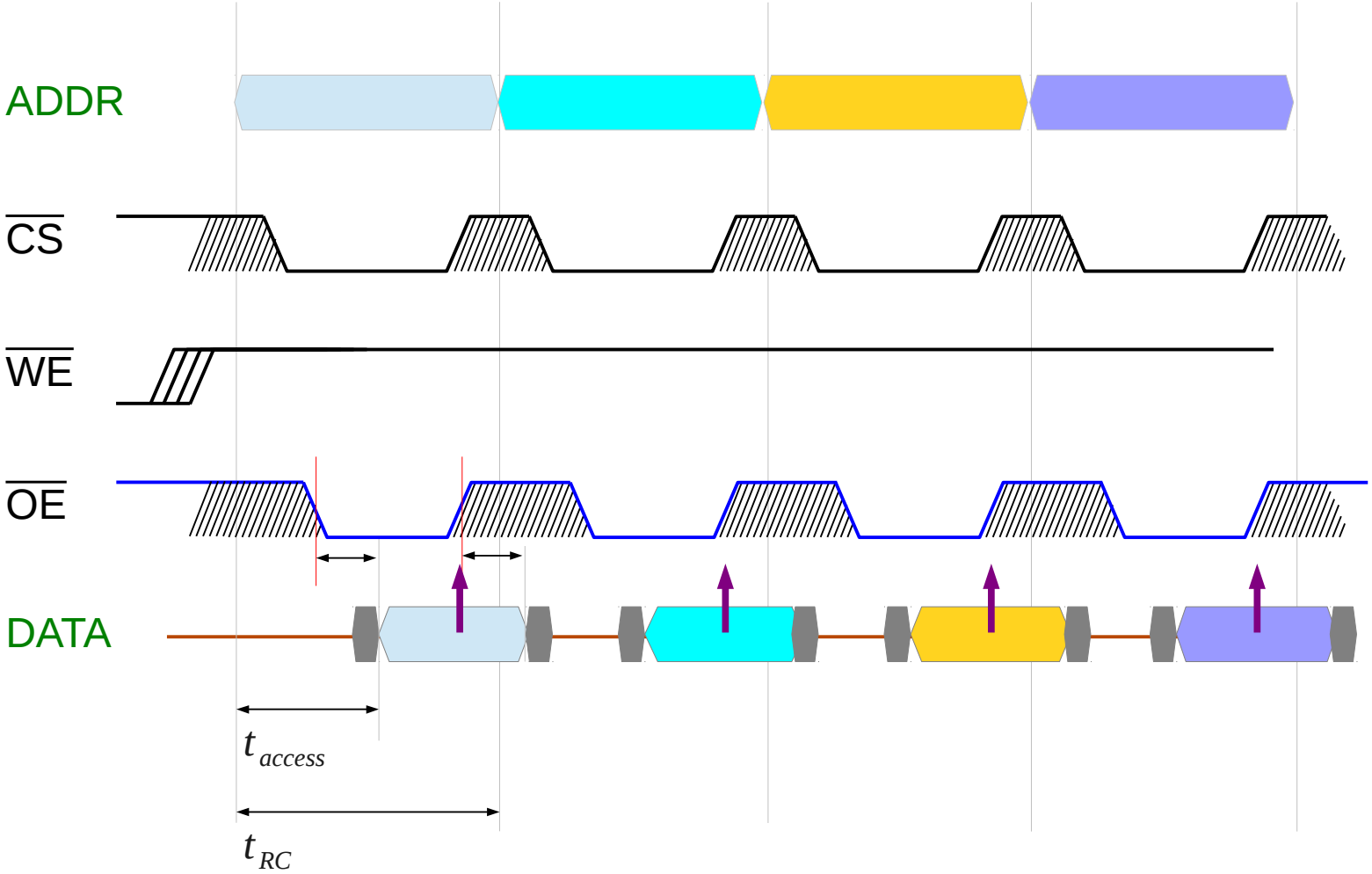


Asynchronous SRAM



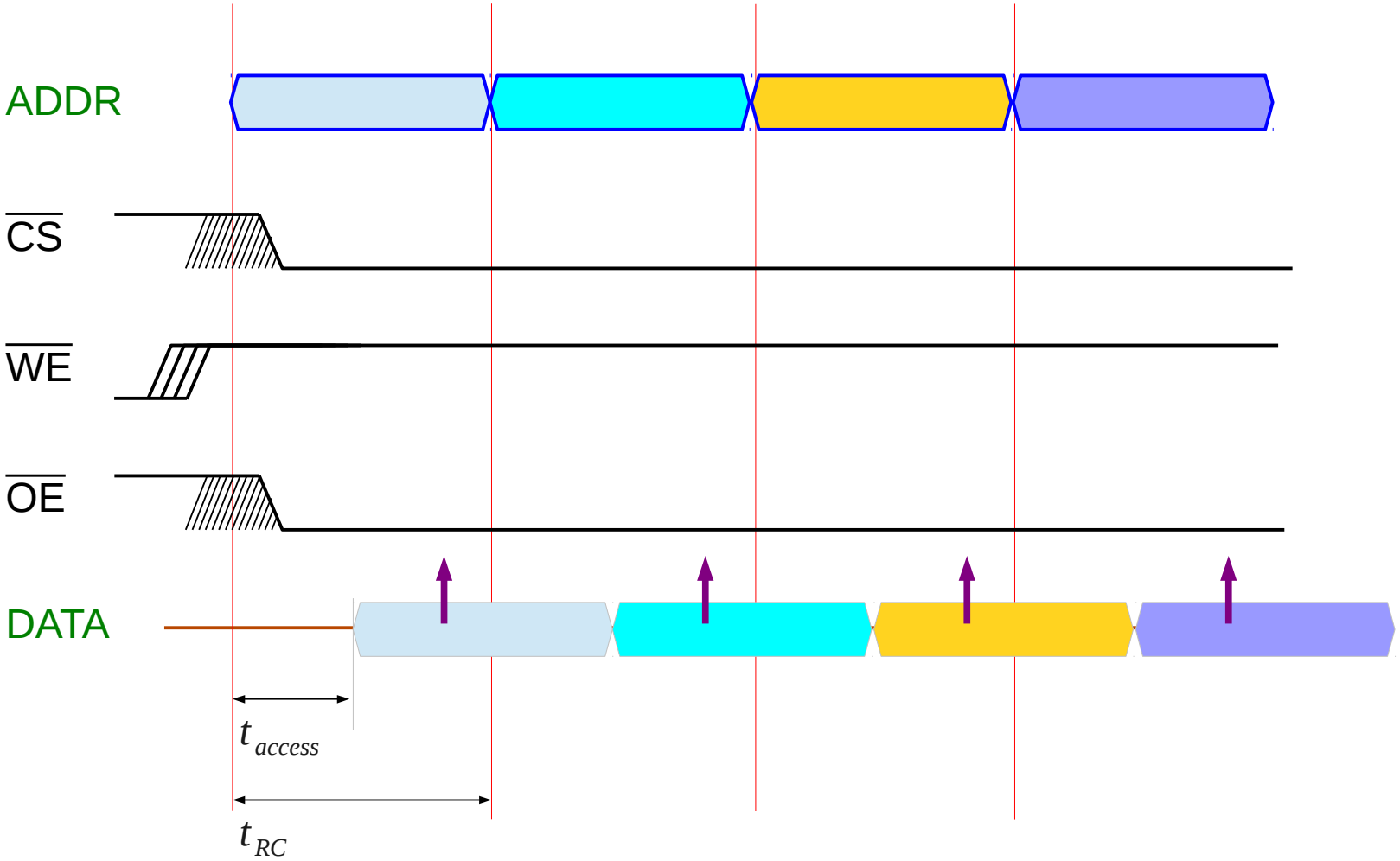
Asynchronous SRAM Read Cycle

$\overline{OE}$ Controlled



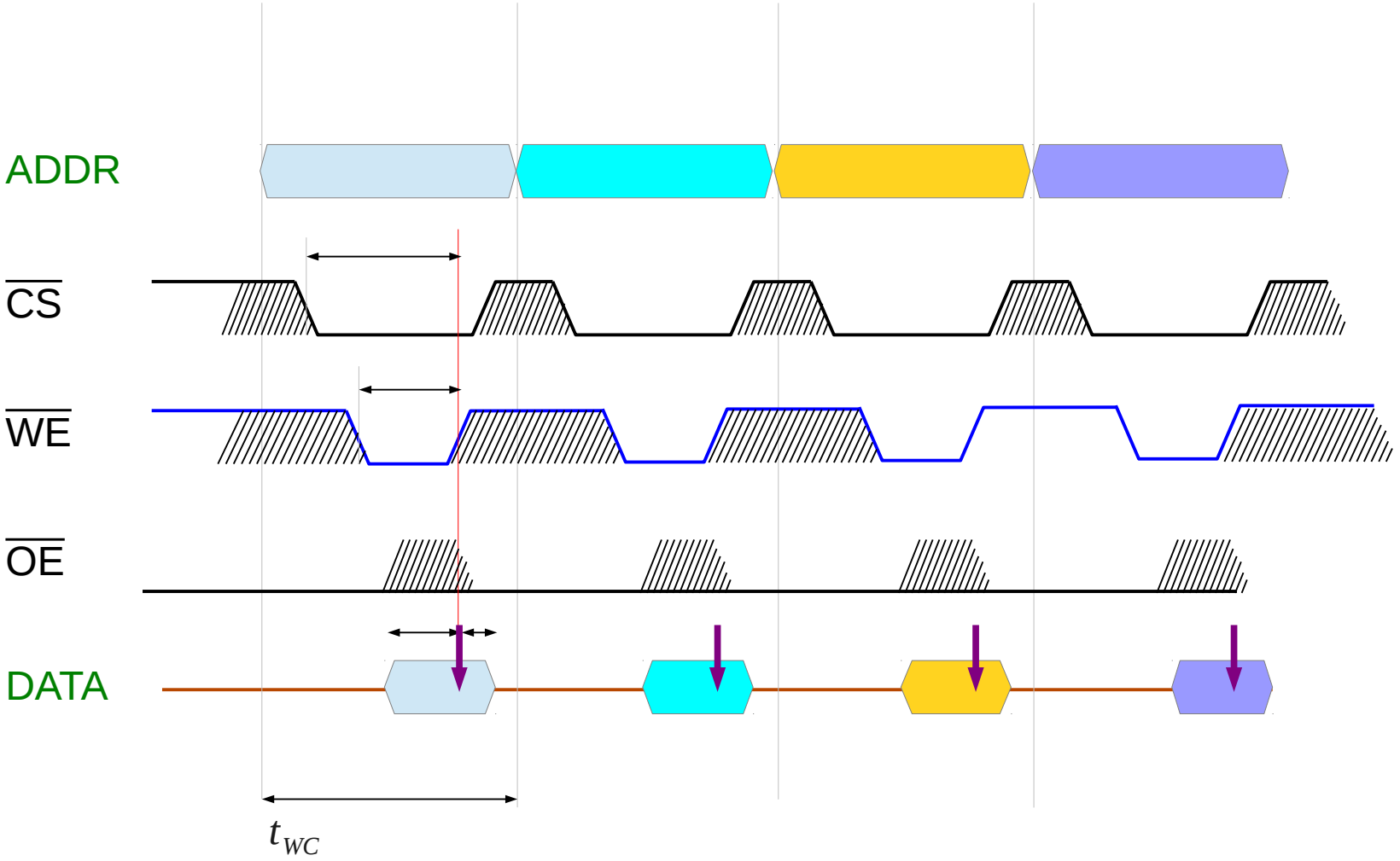
Asynchronous SRAM Read Cycle

Address Transition Controlled



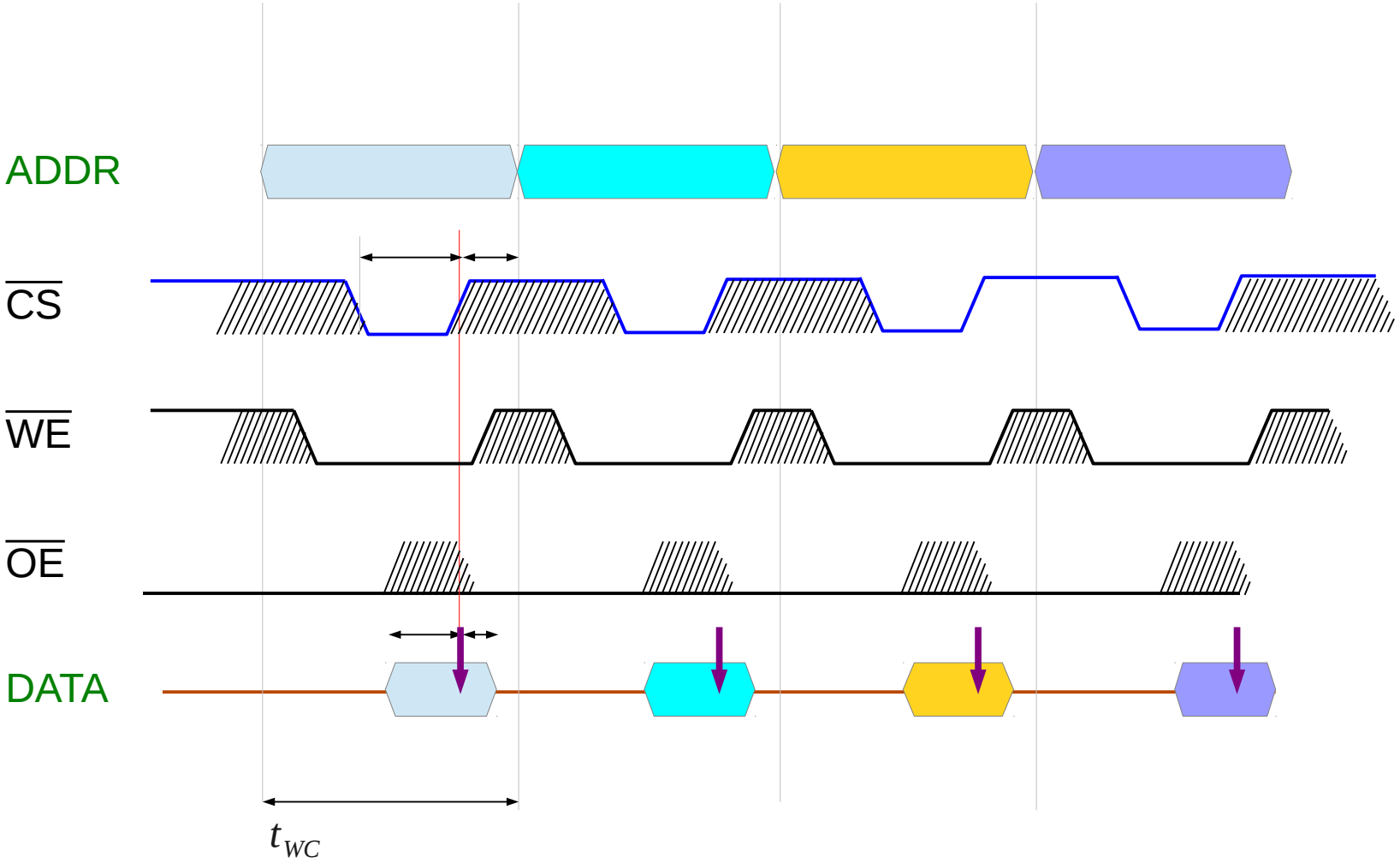
Asynchronous SRAM Write Cycle

$\overline{WE}$ Controlled

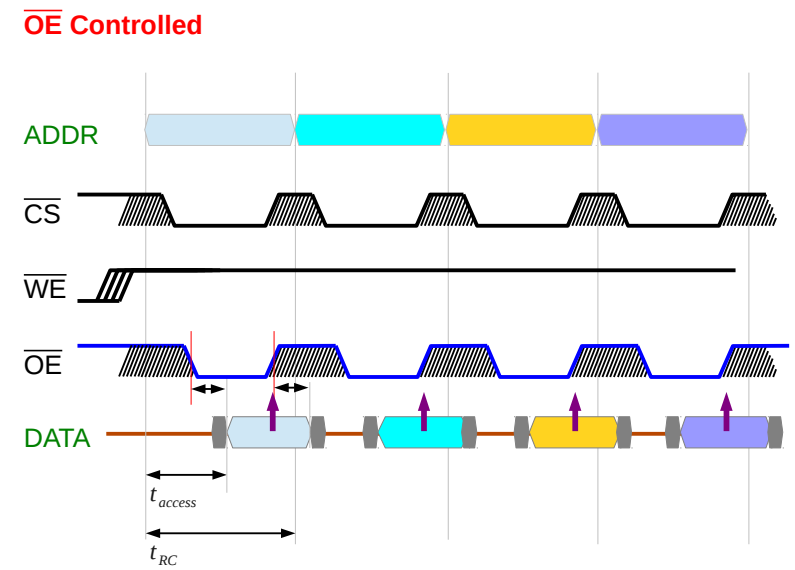


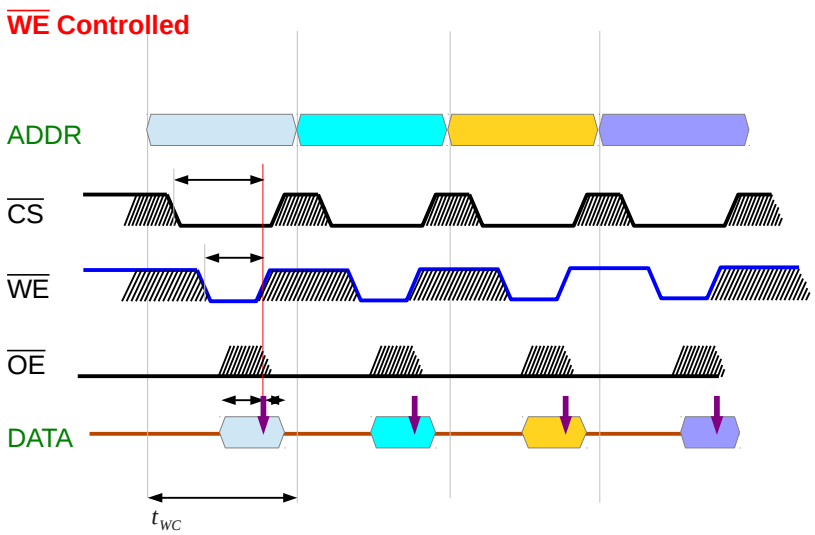
Asynchronous SRAM Write Cycle

$\overline{CS}$ Controlled

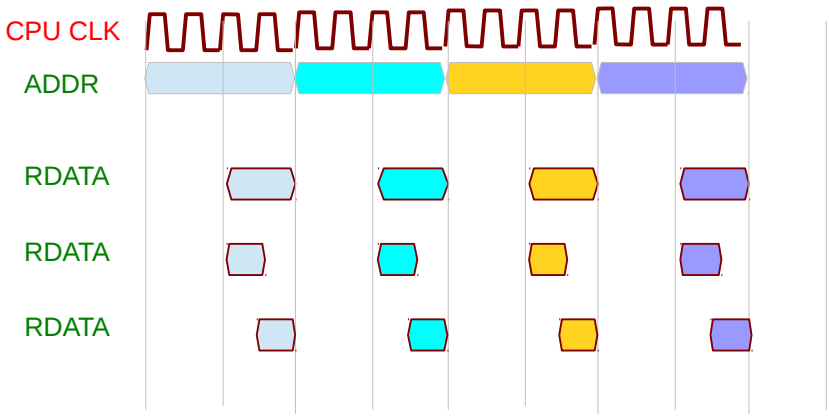


Asynchronous SRAM Cycle

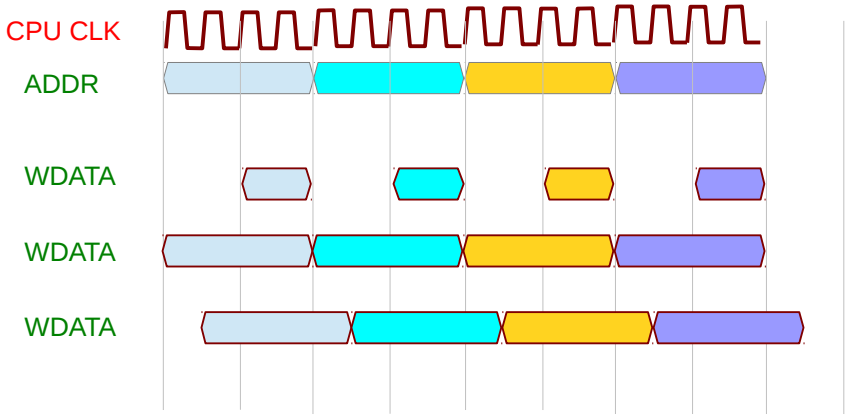




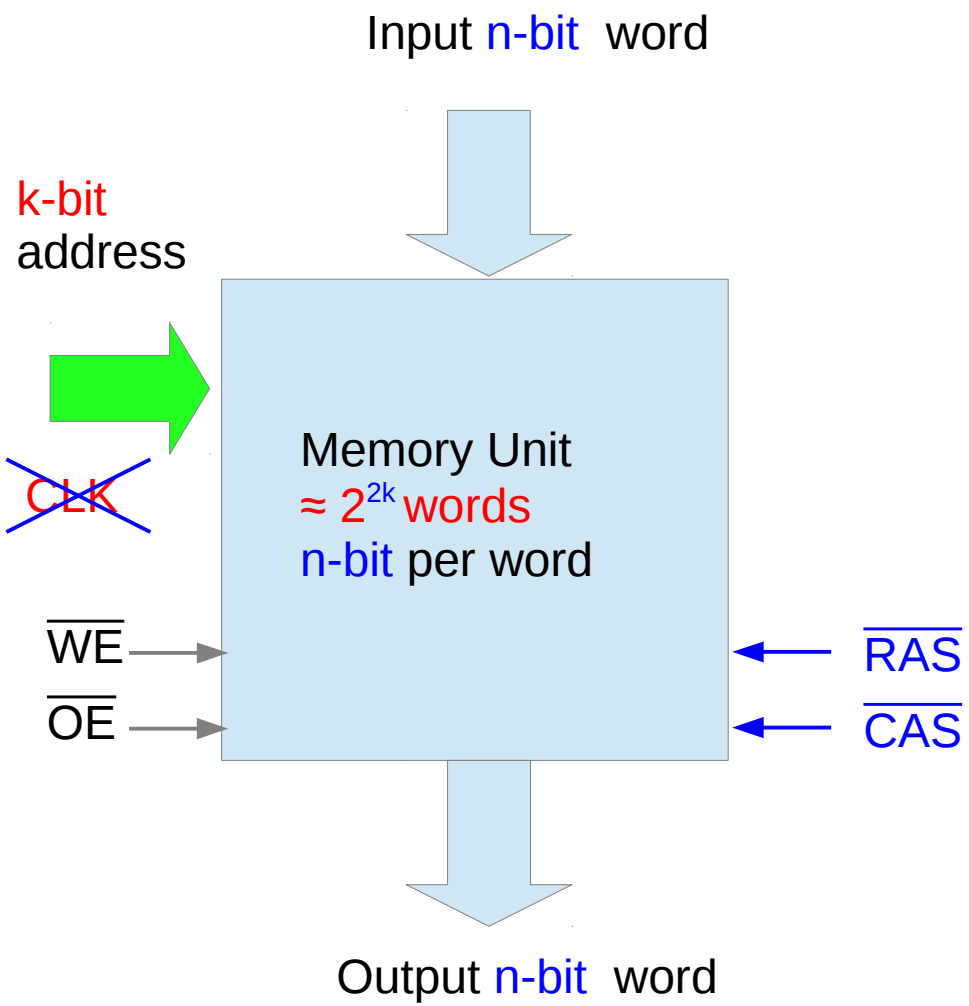
Async SRAM RD Cycle



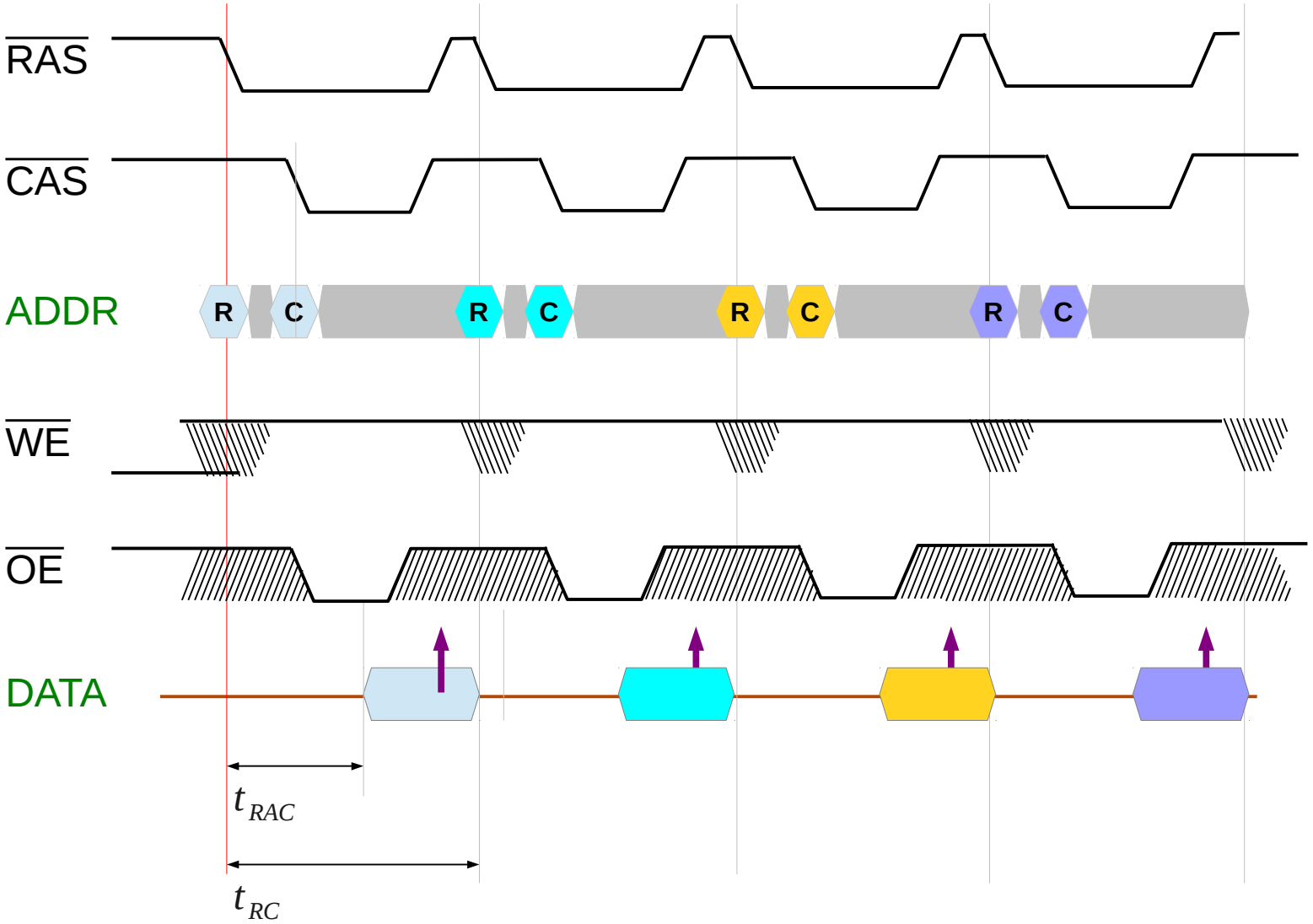
Async SRAM WR Cycle



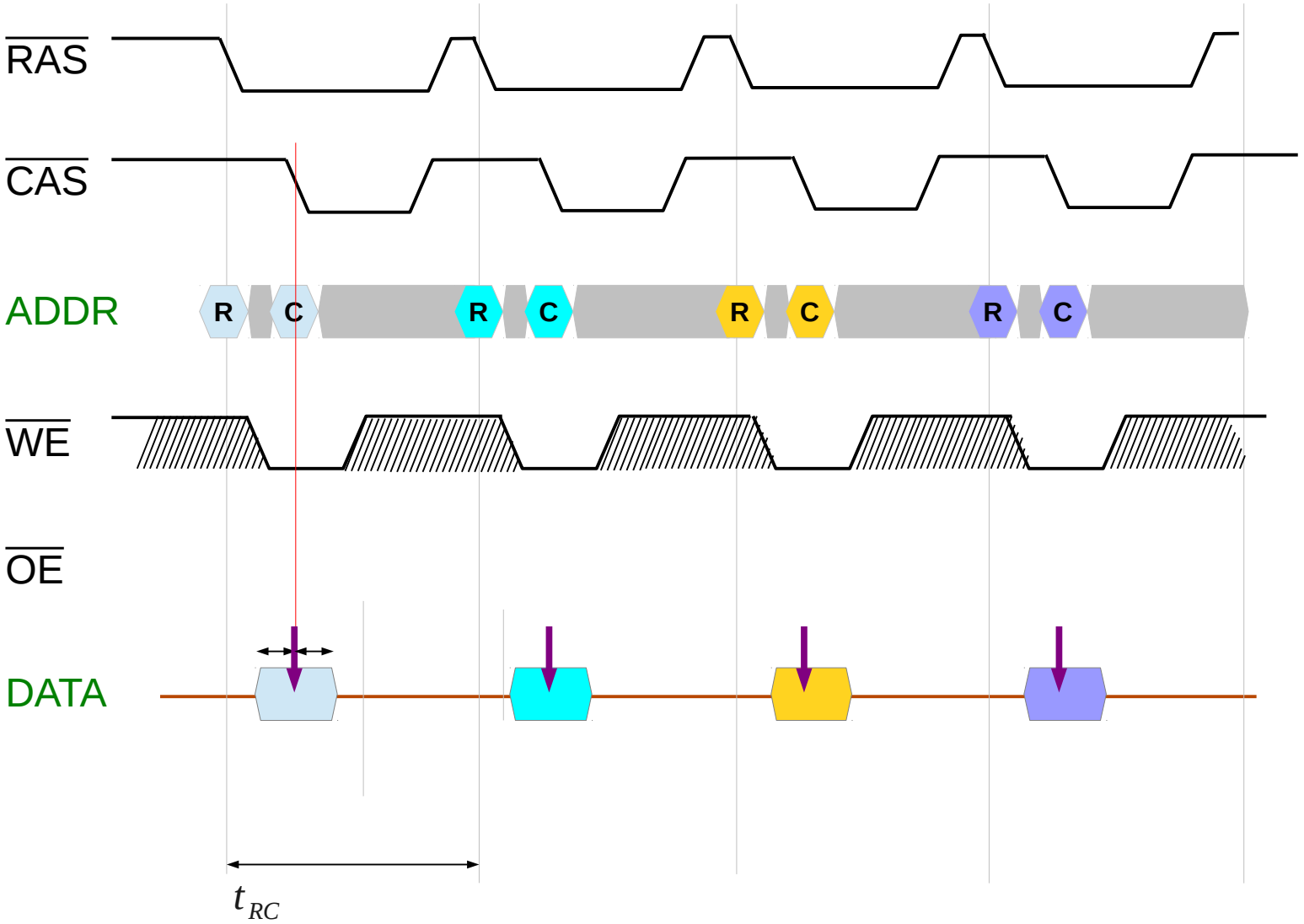
DRAM


Input n-bit word
k-bit
address
CLK
Memory Unit
≈ 2^2k words
n-bit per word
WE
OE
RAS
CAS
Output n-bit word

DRAM Read Cycle



DRAM Write Cycle



References

[1] http://en.wikipedia.org/
[2] M. M. Mano, C. R. Kime, “Logic and Computer Design Fundamentals”, 4th ed.
[3] J. Stephenson, Understanding Metastability in FPGAs. Altera Corporation white paper. July 2009.