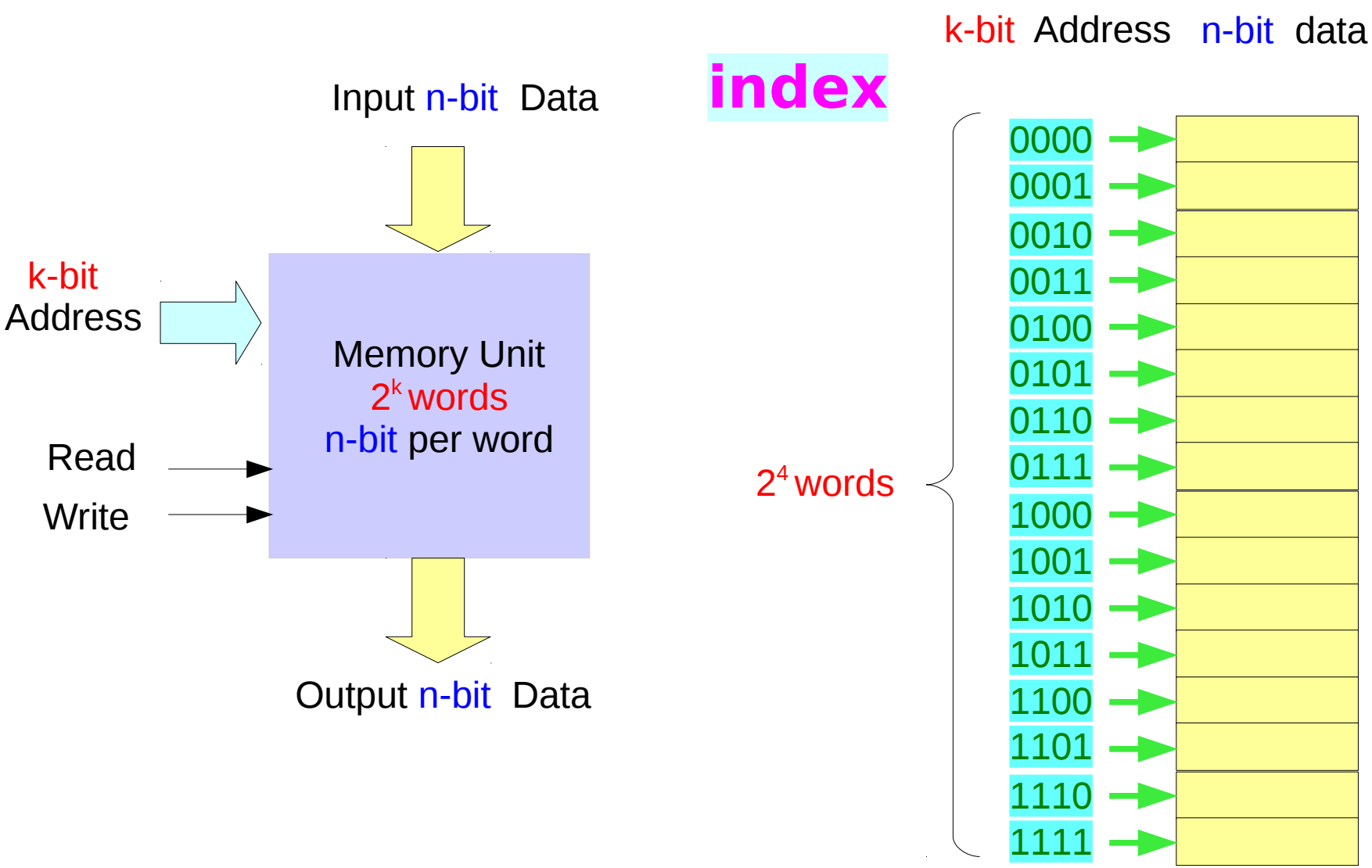
Cache Memory

Young Won Lim
5/23/16

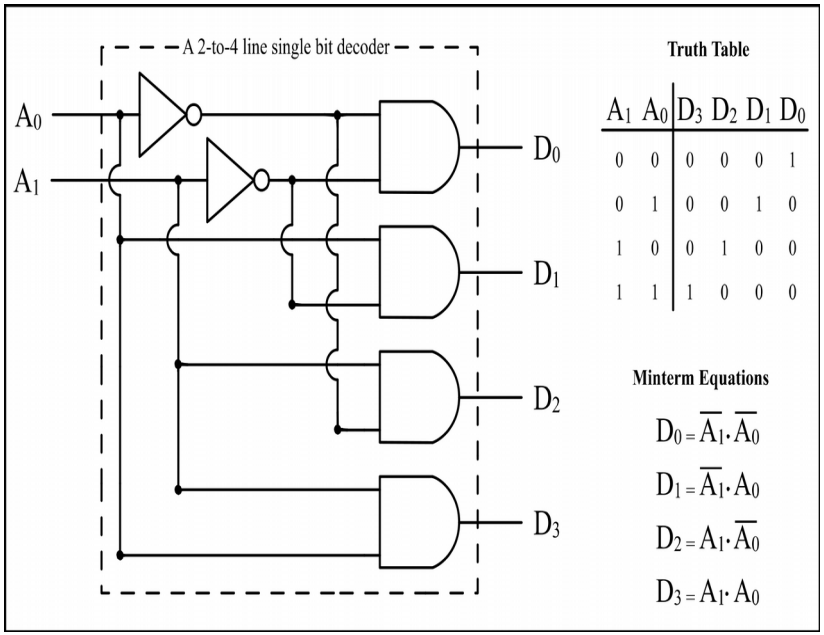
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

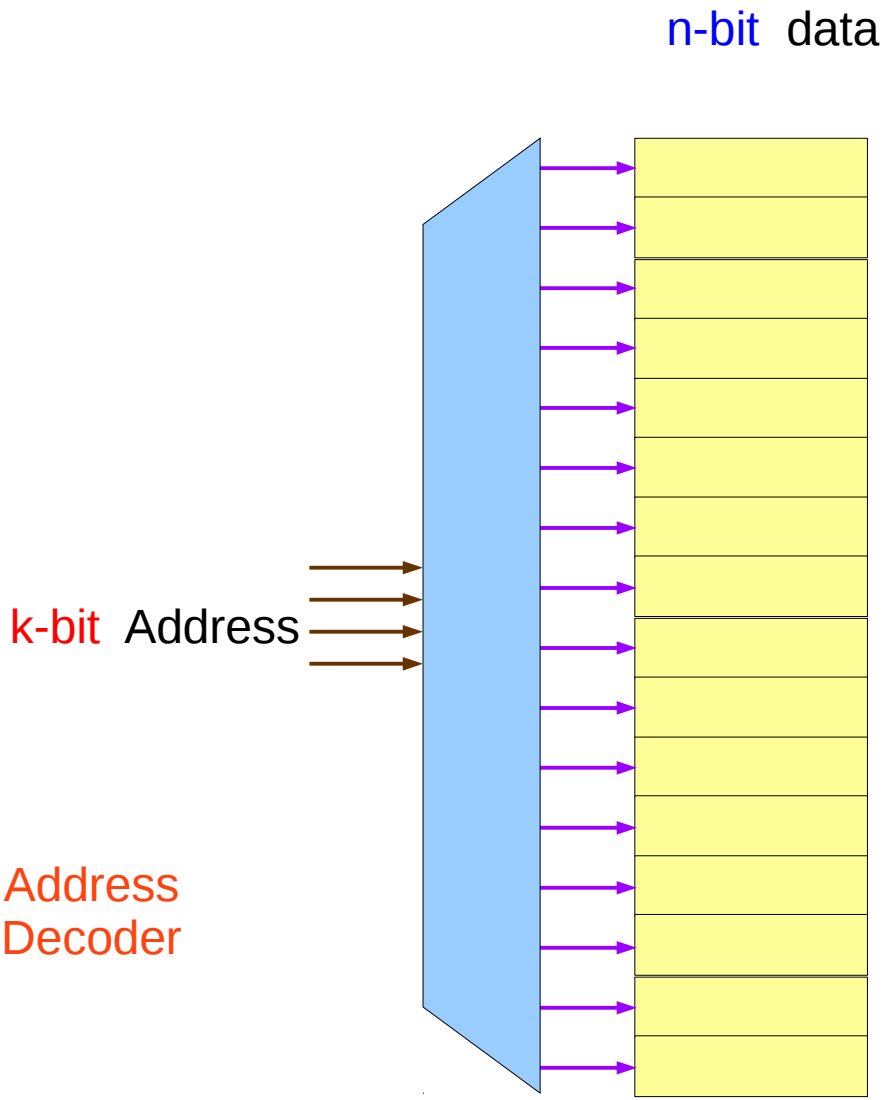
Address is used as an index to a data array



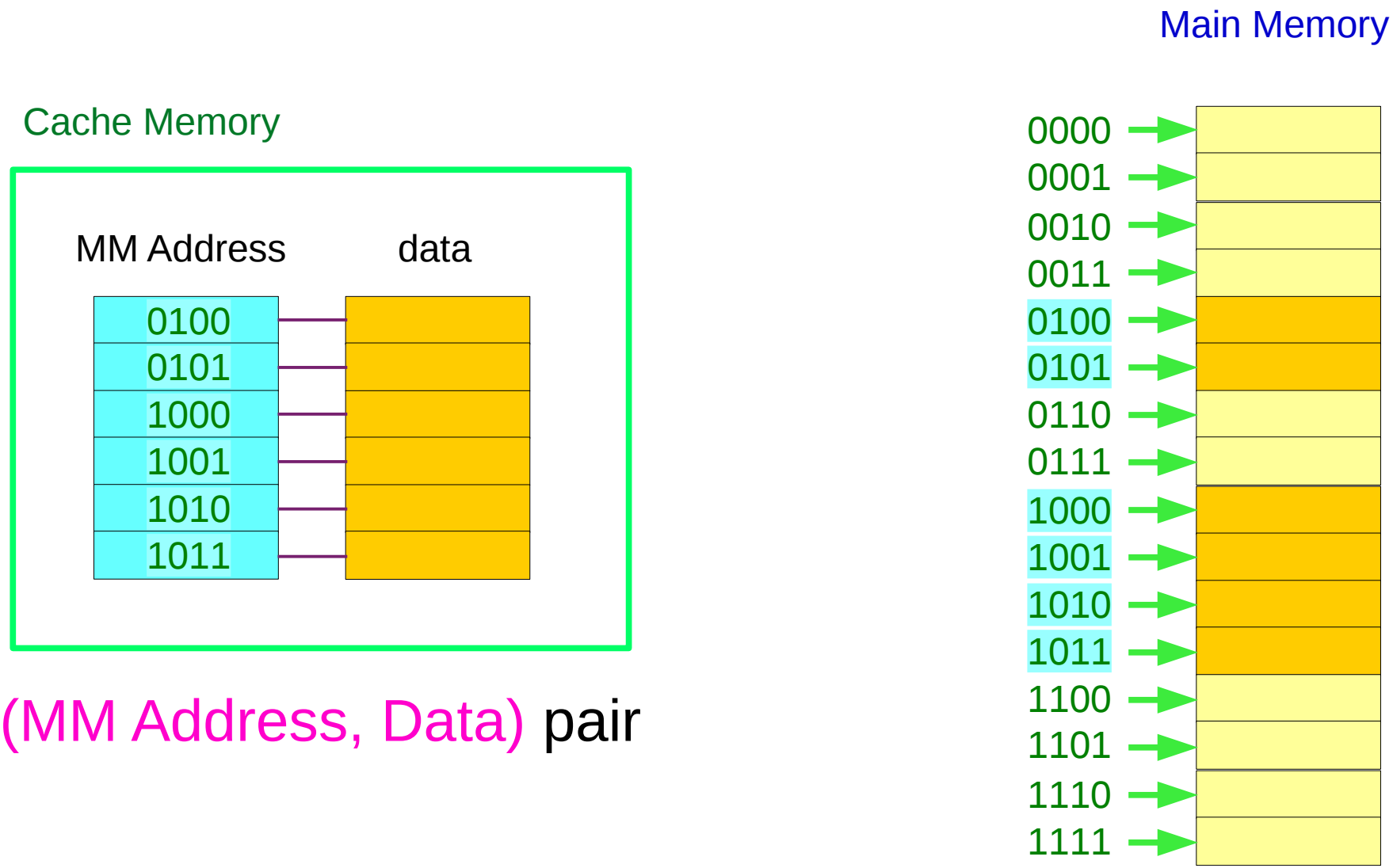
MM Address Decoder



en.wikipedia.org

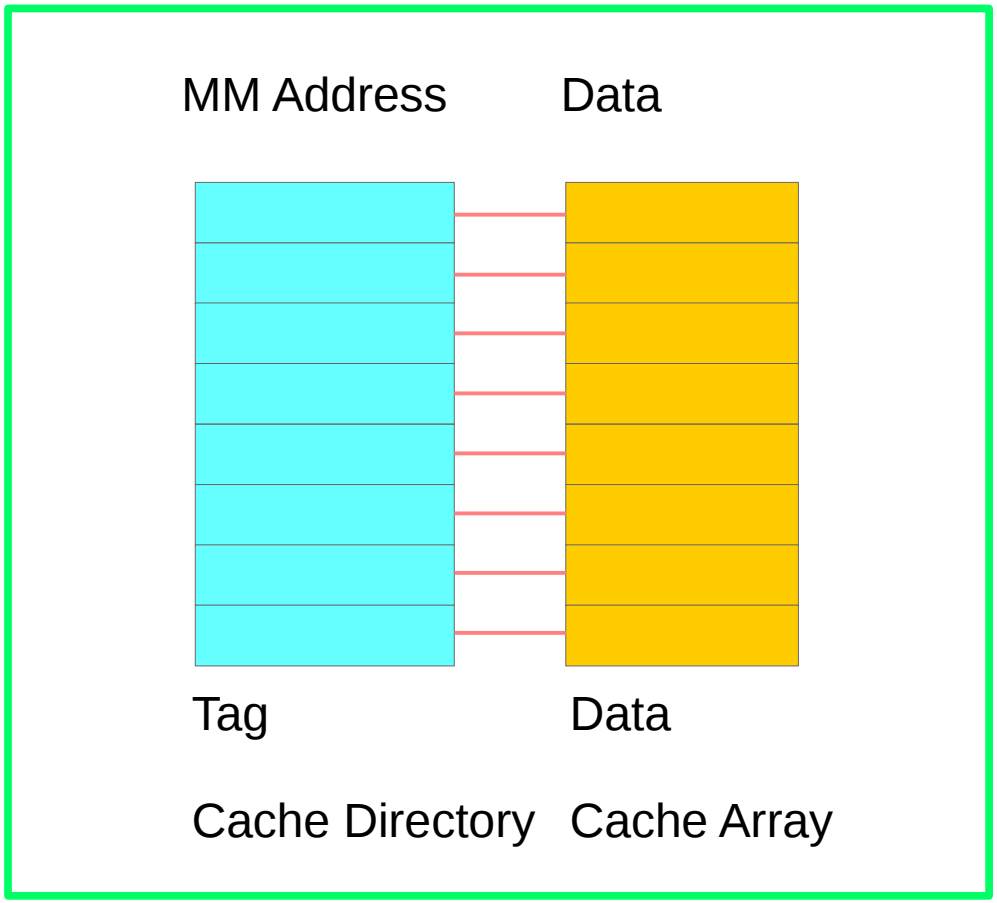


Cache : Storing a partial copy of MM



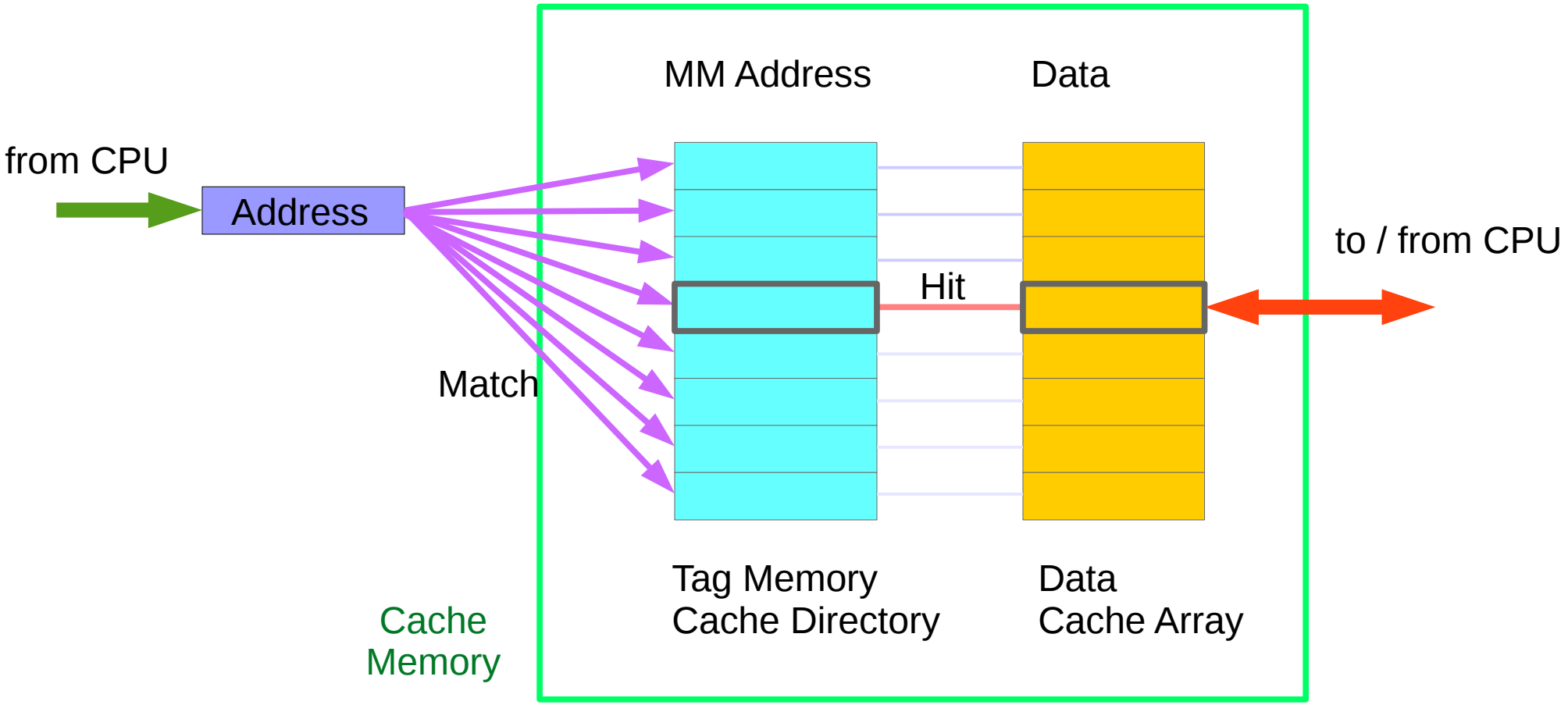
(MM Address, Data) pair

Cache Memory : two components

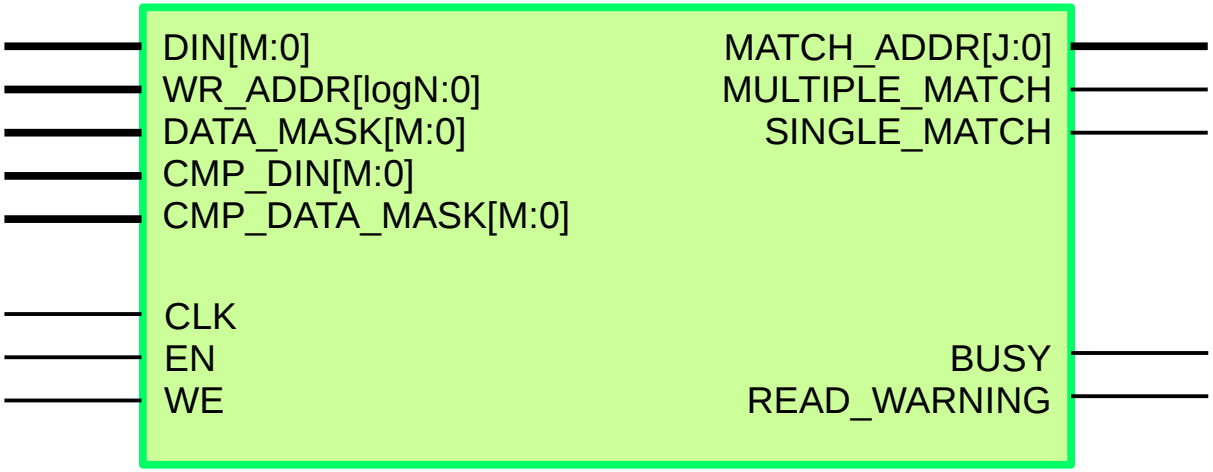


Cache
Memory

Accessing Cache Memory : Address Matching



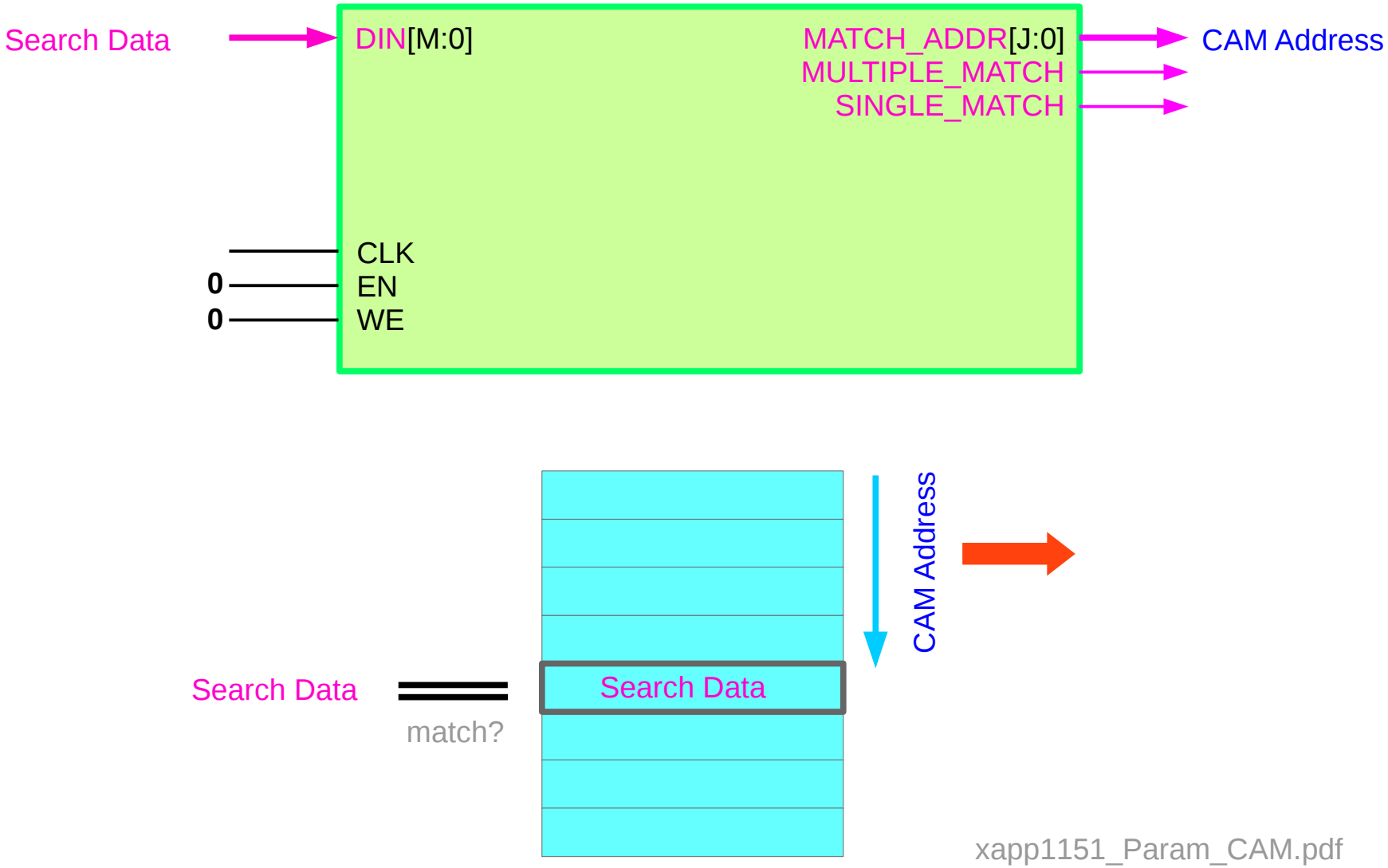
CAM (Content Addressable Memory) Interface



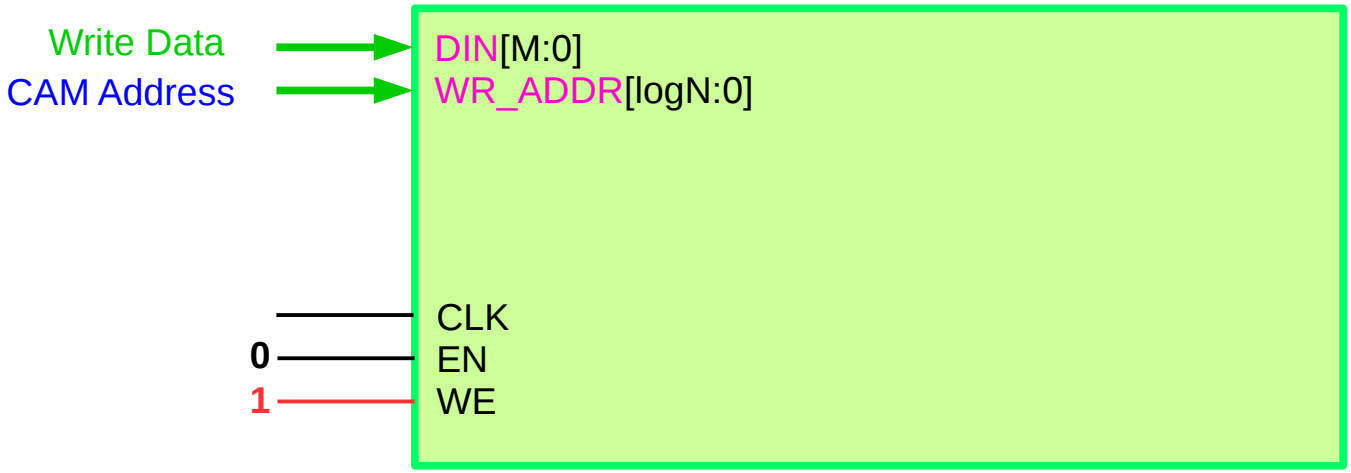
Xilinx CAM

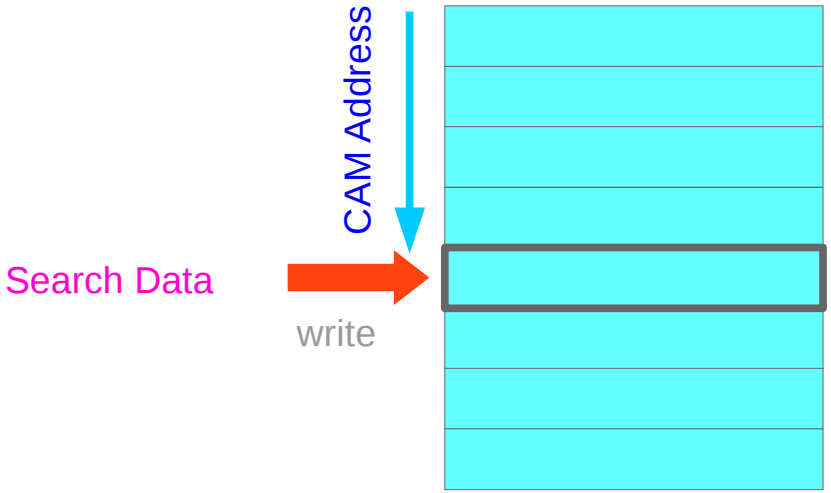
xapp1151_Param_CAM.pdf

CAM Read Operation – Search a key



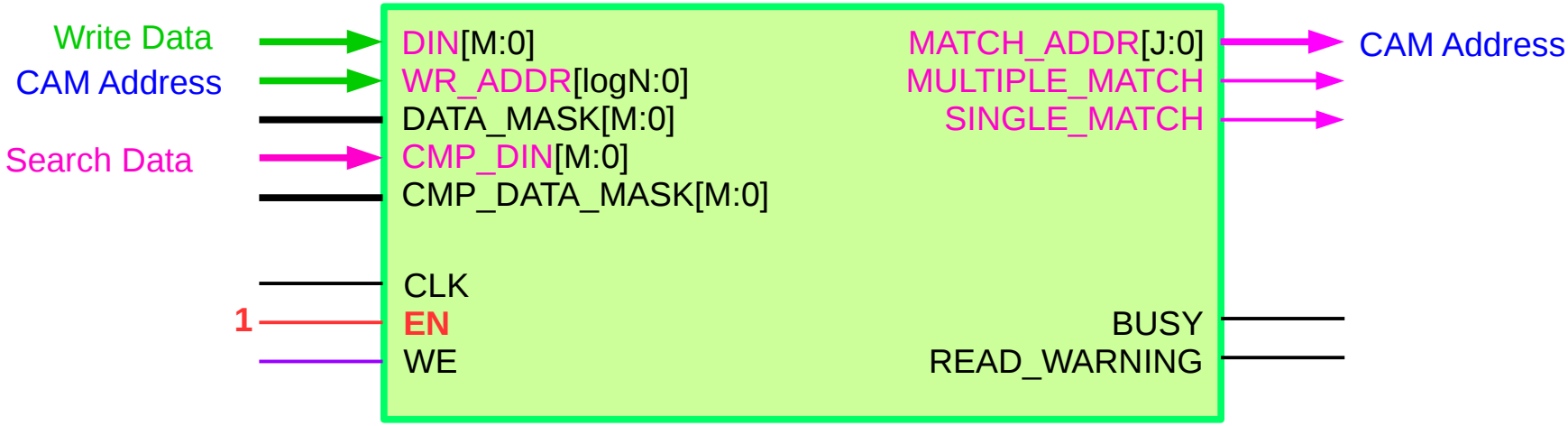
Write Operation





xapp1151_Param_CAM.pdf

Simultaneous Read / Write

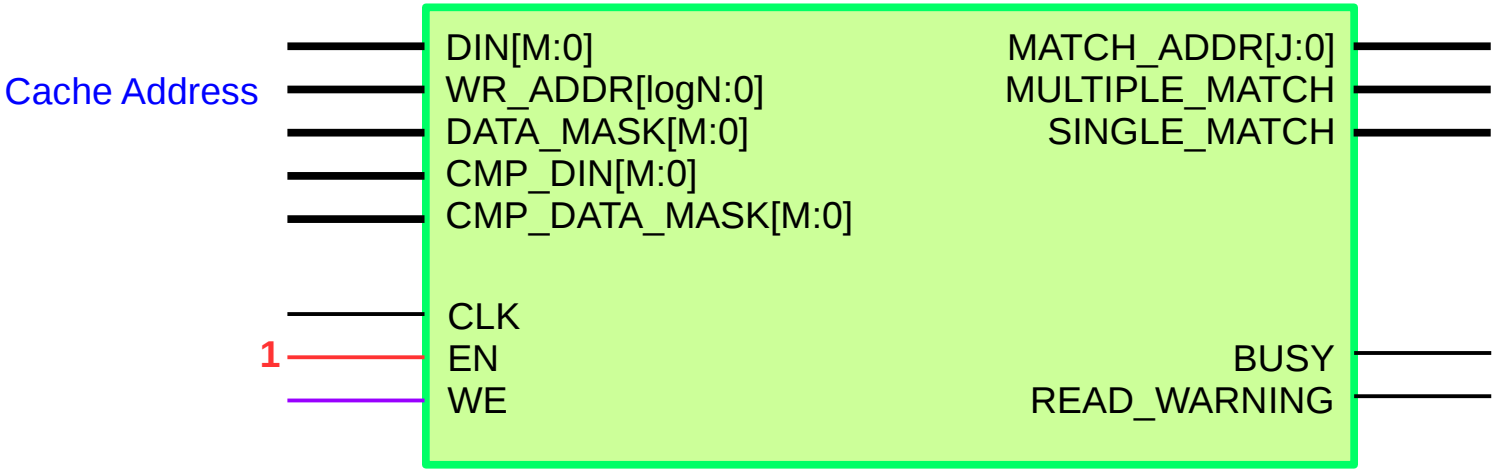


EN=1 simultaneous write/read

Simultaneous Read/Write
- Simultaneous write and search operations
- With an output to warn the user of possible collision
- Read warning flag:
  - The data applied to the CAM for a read
  - Matches the data that is currently being written into the CAM
  - By unfinished write operation

xapp1151_Param_CAM.pdf

CAM (Content Addressable Memory)



EN=1 simultaneous write/read

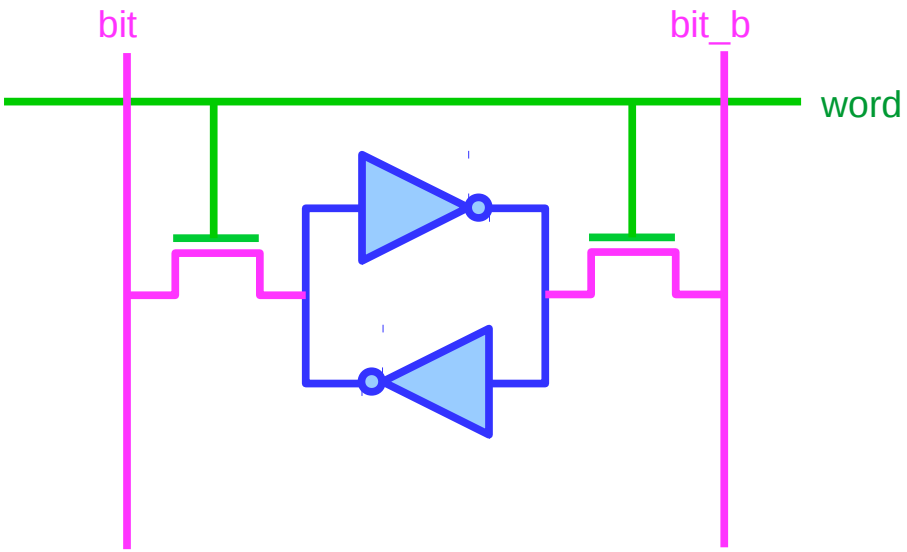
DIN[M:0] Data in Bus
The data to be written into
The data read from the CAM
Simultaneous read/write mode
CMP_DIN for the read operation
Standard Ternary mode

DIN	DATA_MASK	
0	0	0
1	0	1
0	1	X
1	1	X

CMP_DIN[M:0] Compare Data In Bus
Simultaneous read/write
The data read from the CAM
Ternary mode
One of the two input buses
To determine the bit value
During read operation

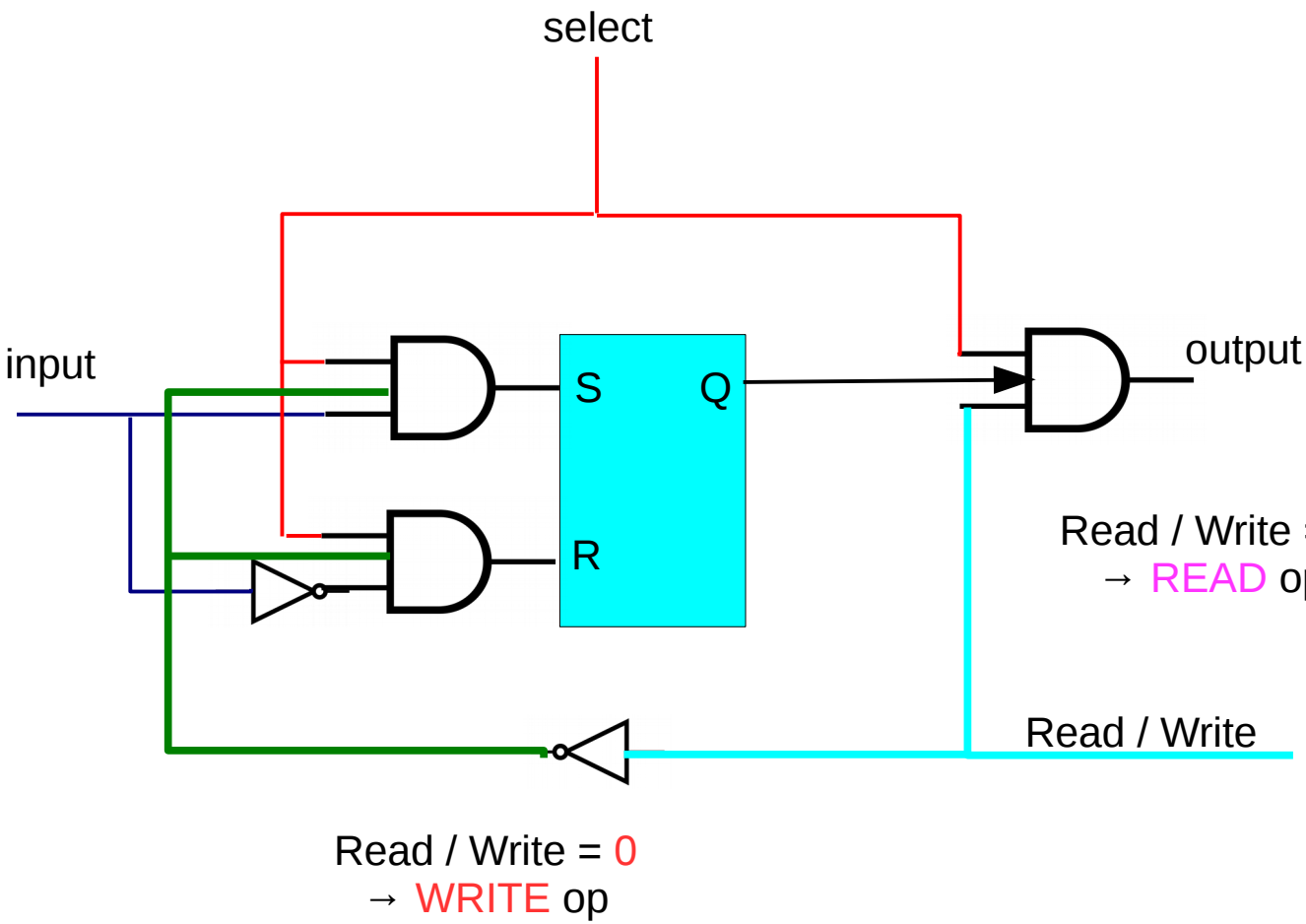
xapp1151_Param_CAM.pdf

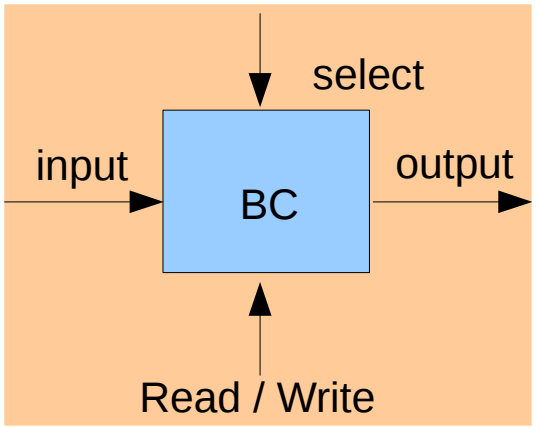
SRAM Cell



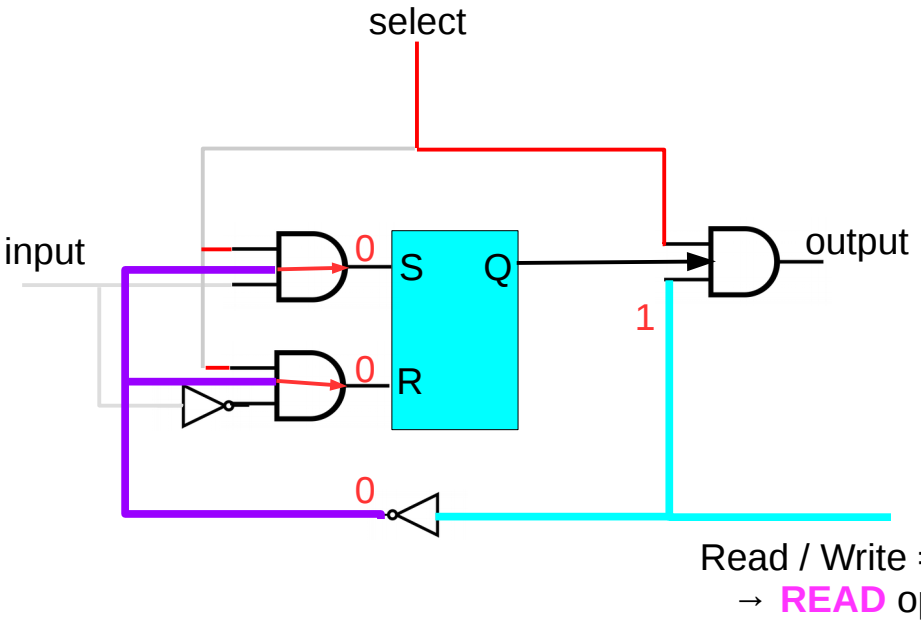
CMOS VLSI Design 4th ed, Weste

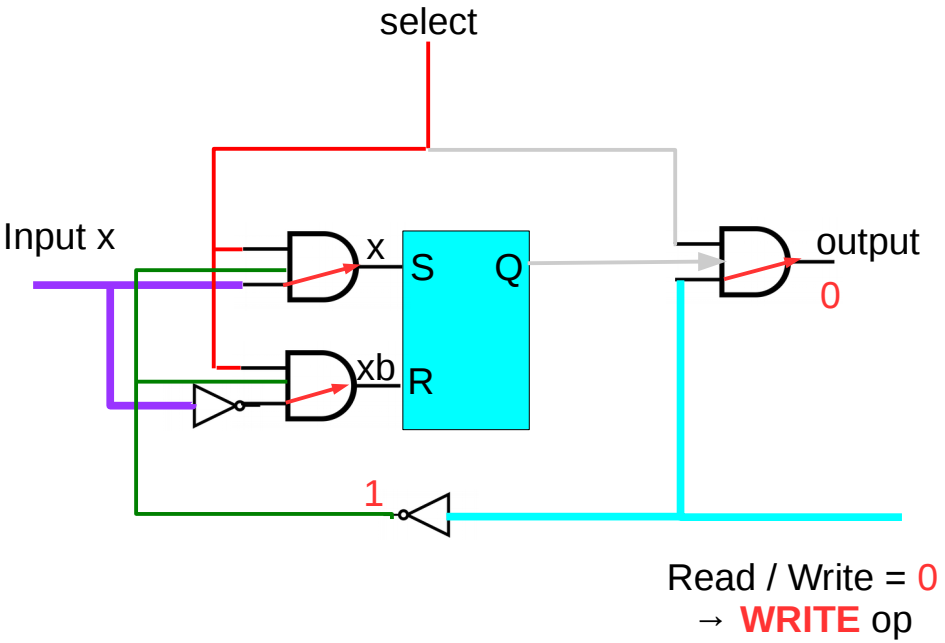
SRAM Bit Cell RTL Model





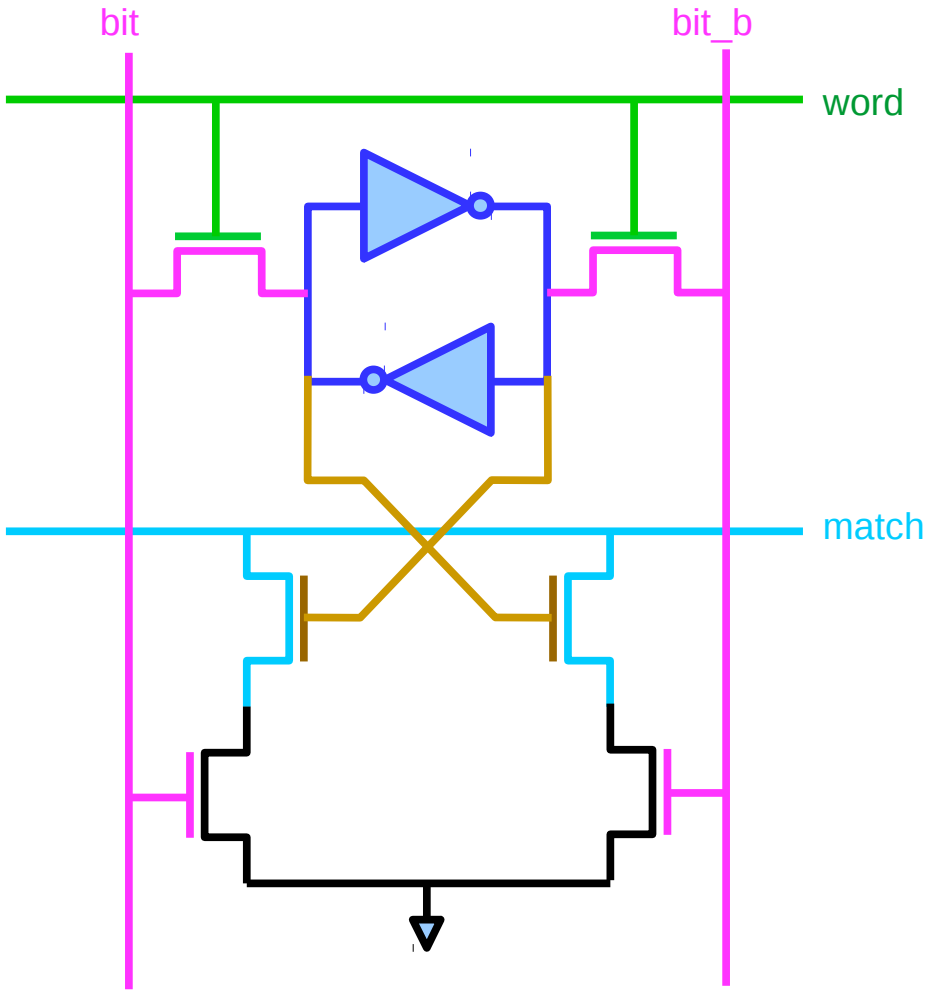
SRAM Bit Cell Read & Write Operations





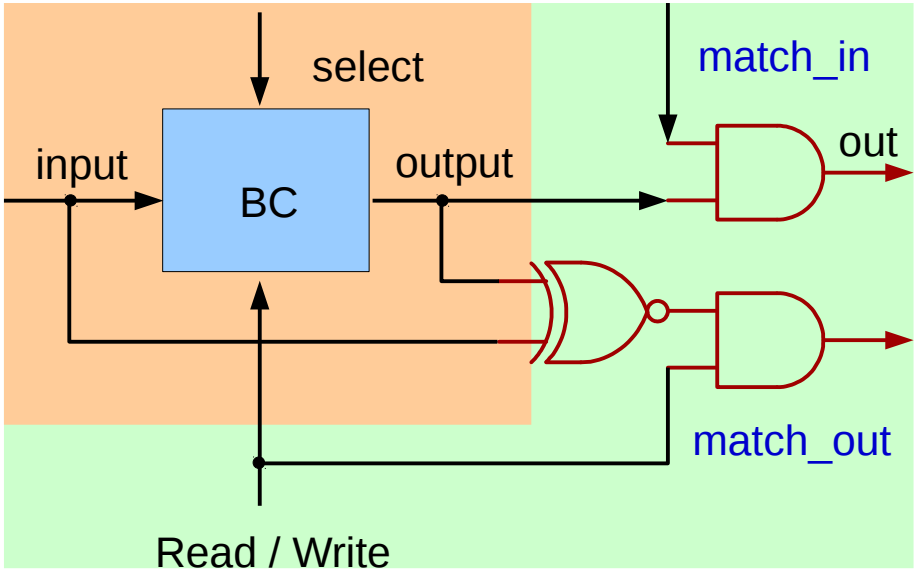
HOLD	S=0 R=0	Q=old Q $\overline{Q}$=old $\overline{Q}$	
Input x=1	SET	S=**1** R=0	Q=1 $\overline{Q}$=0
Input x=0	RESET	S=0 R=**1**	Q=0 $\overline{Q}$=1

10T CAM Cell

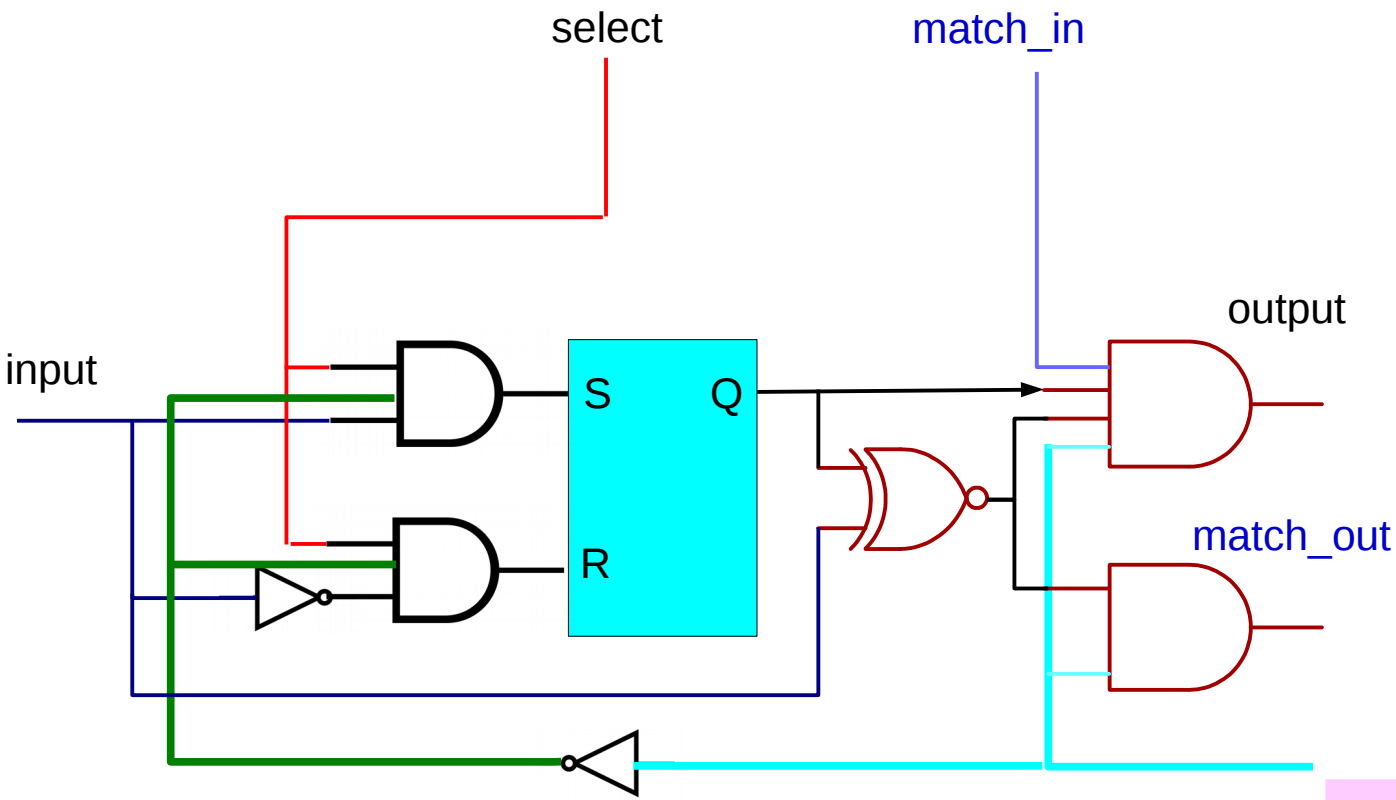


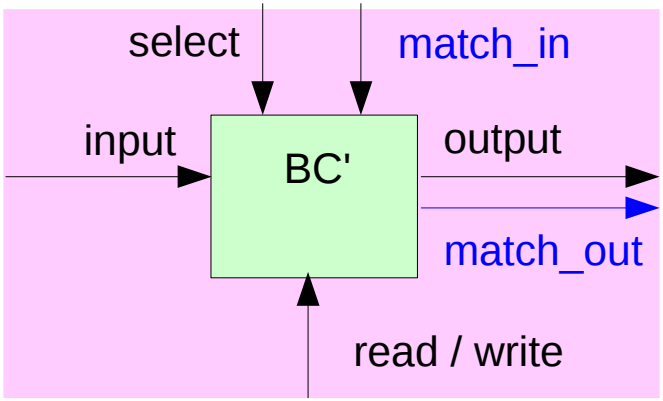
CMOS VLSI Design 4th ed, Weste

SRAM Bit Cell RTL Model

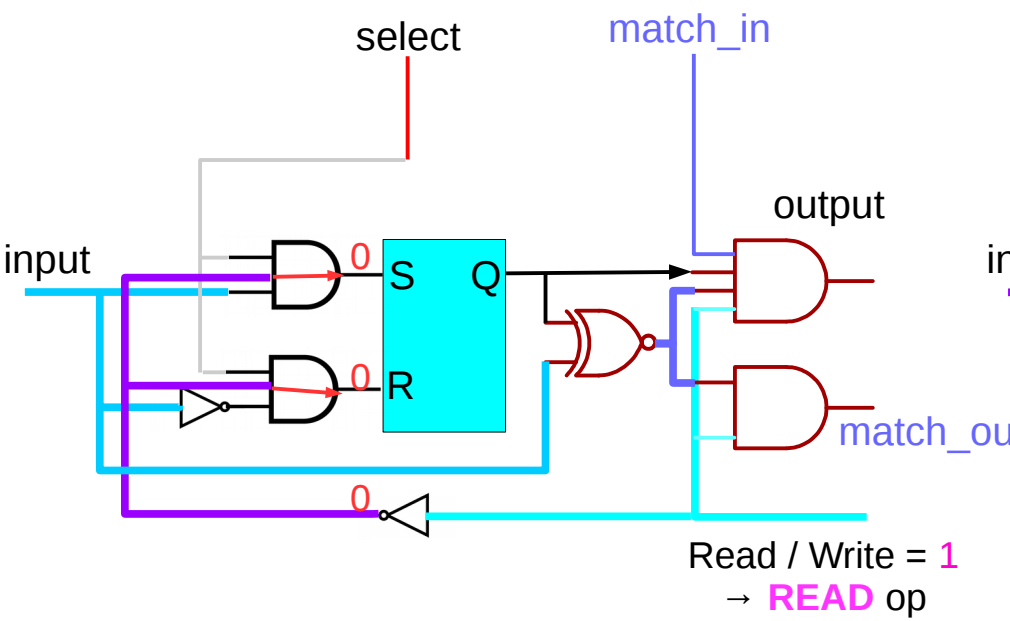


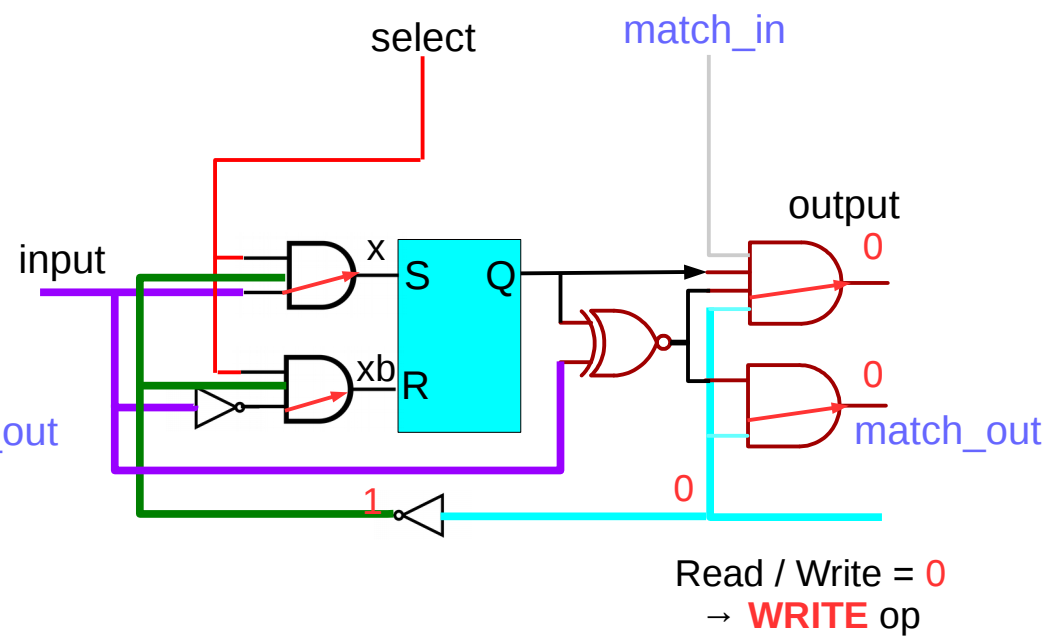
CAM Bit Cell RTL Model





CAM Bit Cell RTL Model – Read / Write Operations





HOLD	S=0 R=0	Q=old Q $\overline{Q}$=old $\overline{Q}$
Input x=1 SET	S=**1** R=0	Q=1 $\overline{Q}$=0
Input x=0 RESET	S=0 R=**1**	Q=0 $\overline{Q}$=1

2x2 CAM Bit Cell RTL Model

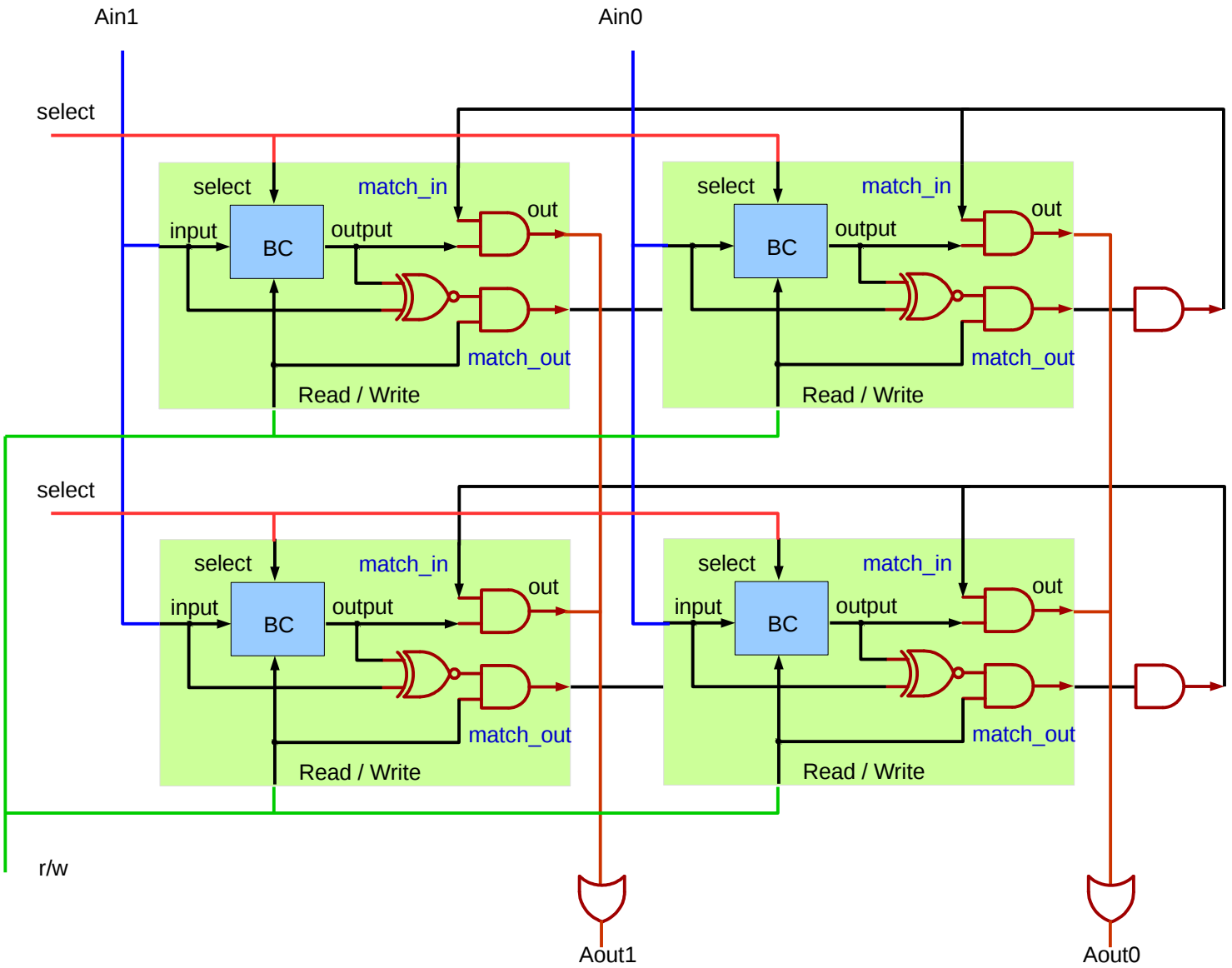
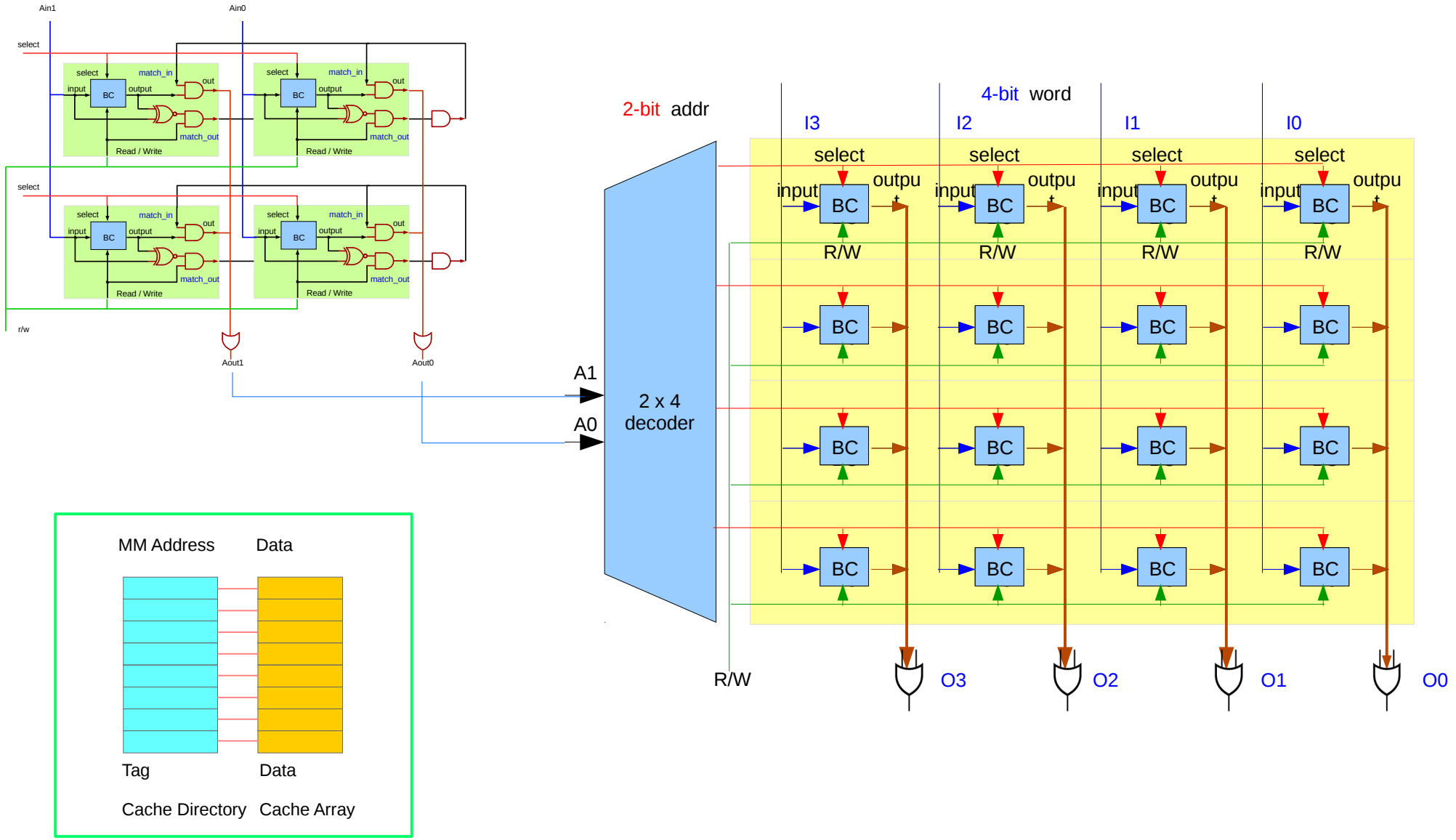
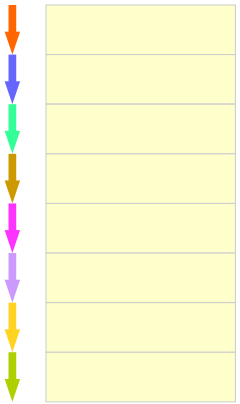


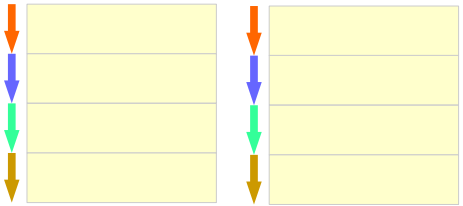
Diagram for a 4x4 Memory

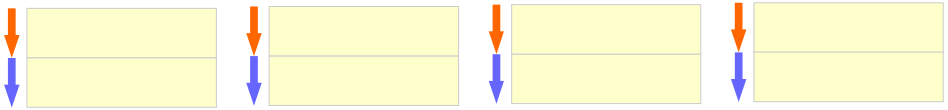


Cache Organization

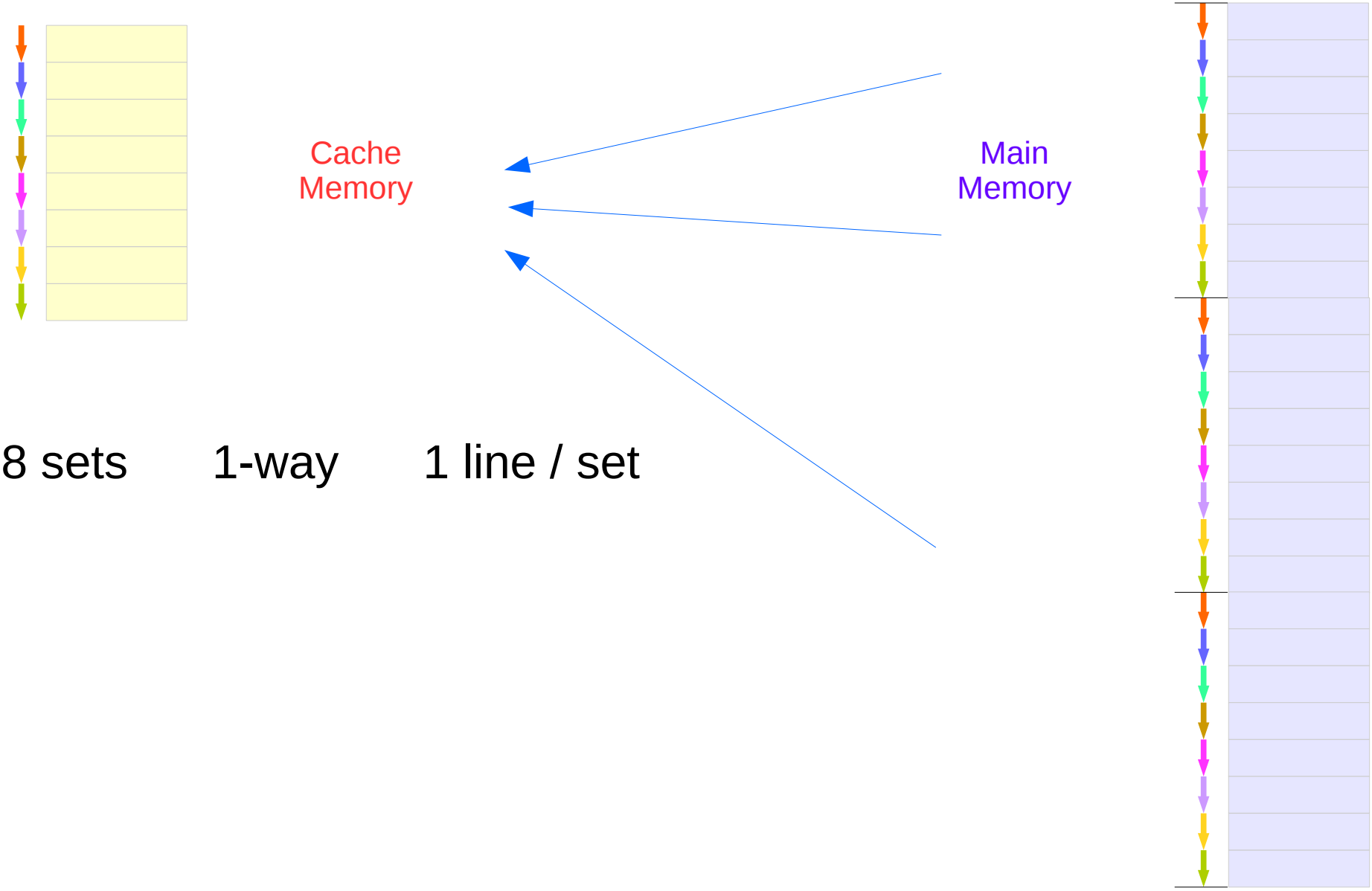


8 sets	1-way	1 line / set
4 sets	2-way	2 lines / set
2 sets	4-way	4 lines / set
1 set	8-way	8 lines / set



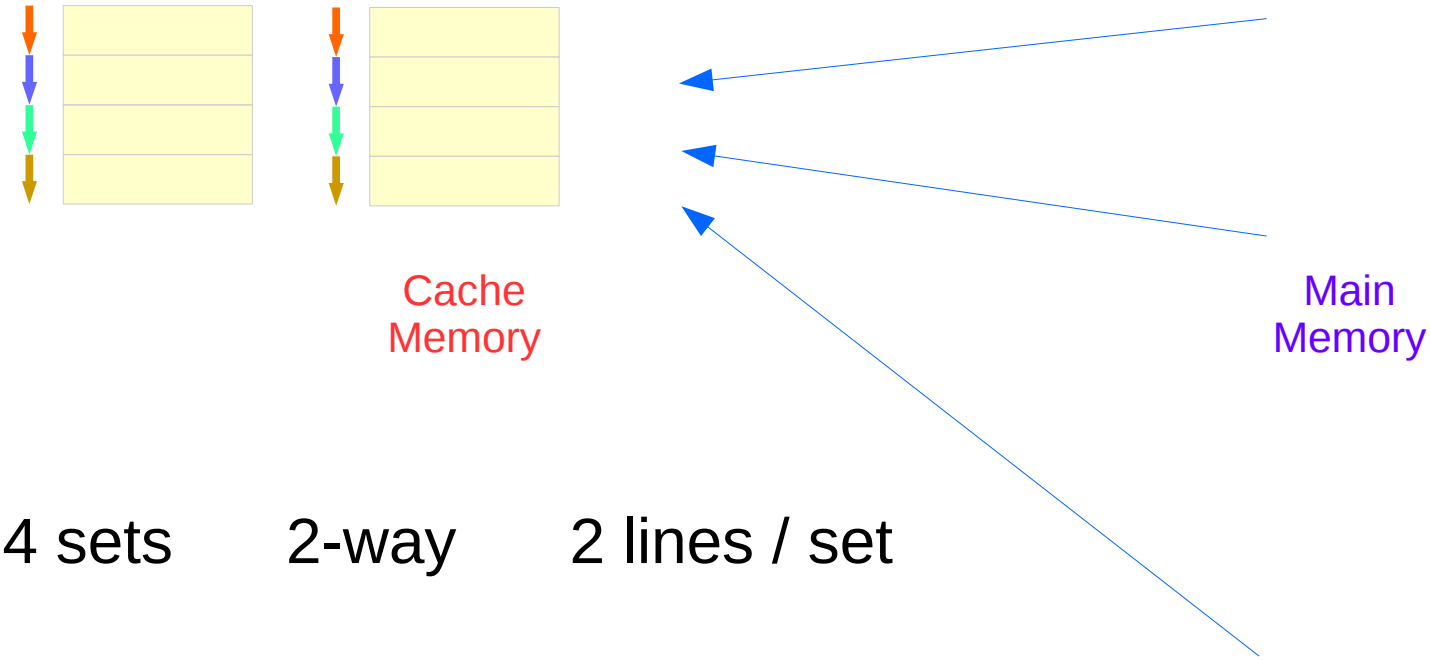


Direct Mapping



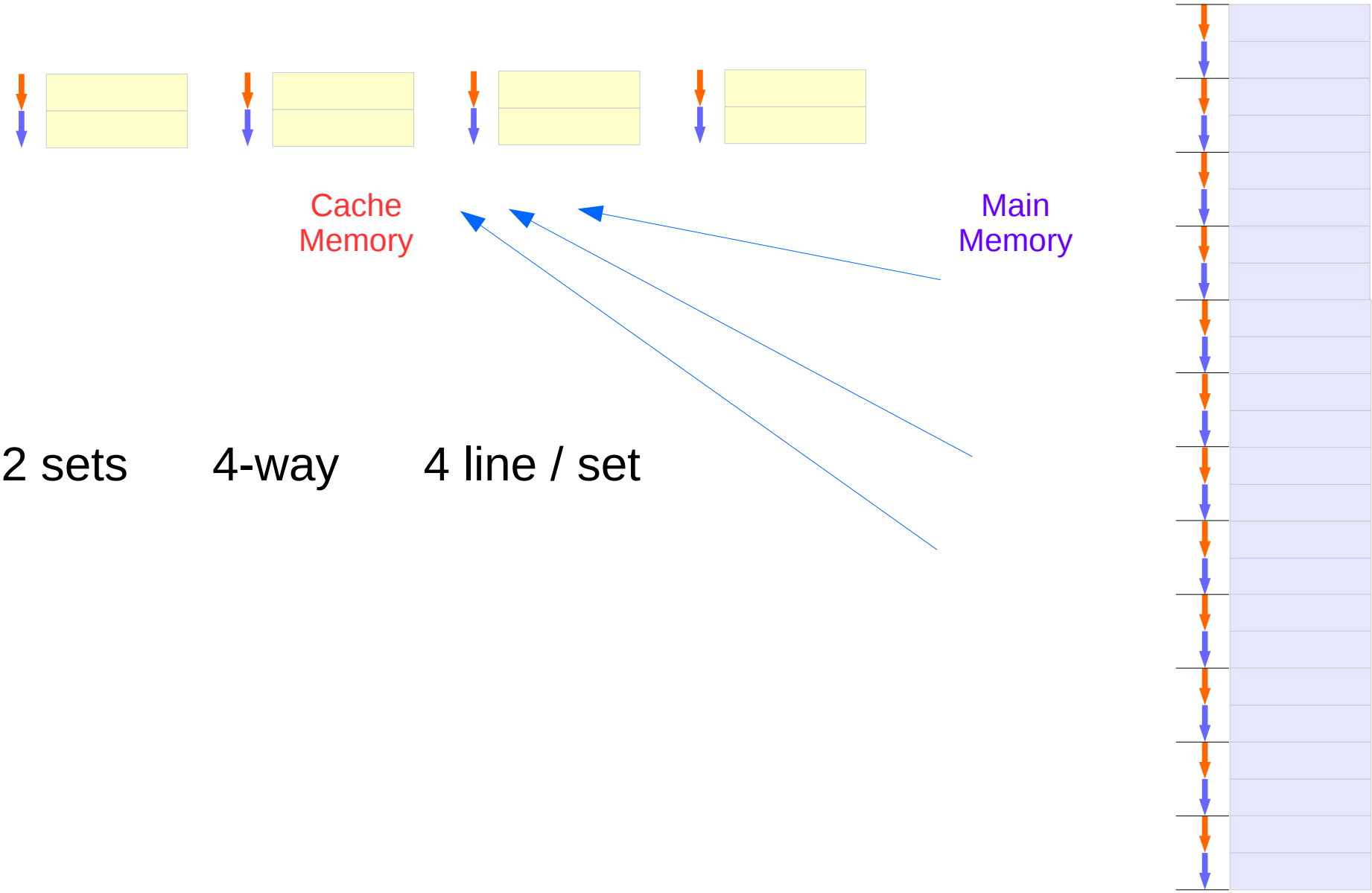
8 sets 1-way 1 line / set

2-Way Set Associative Mapping

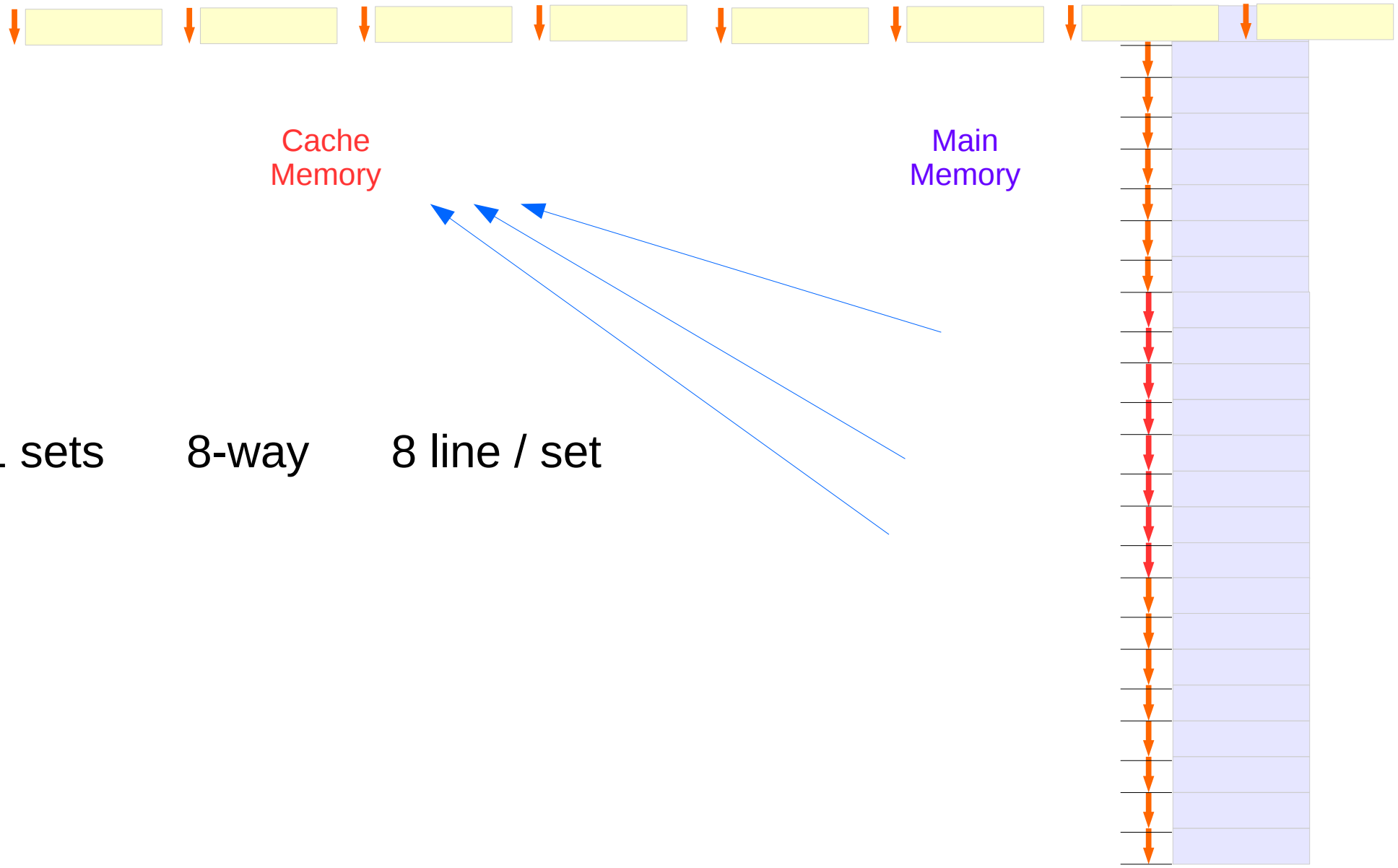


4 sets 2-way 2 lines / set

4-way Set Associative Mapping

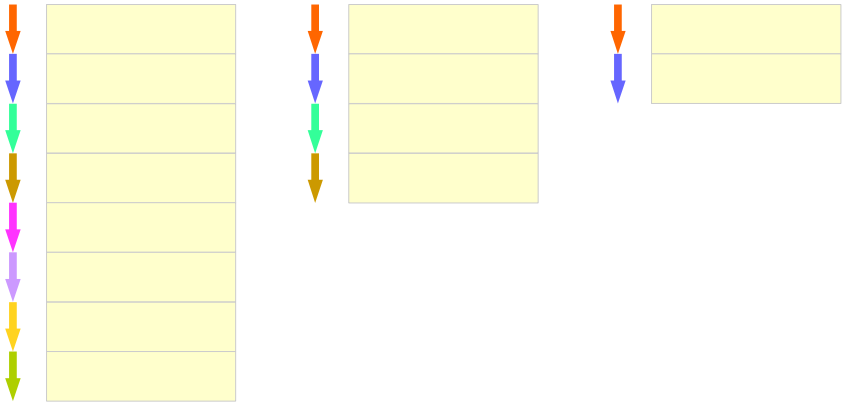


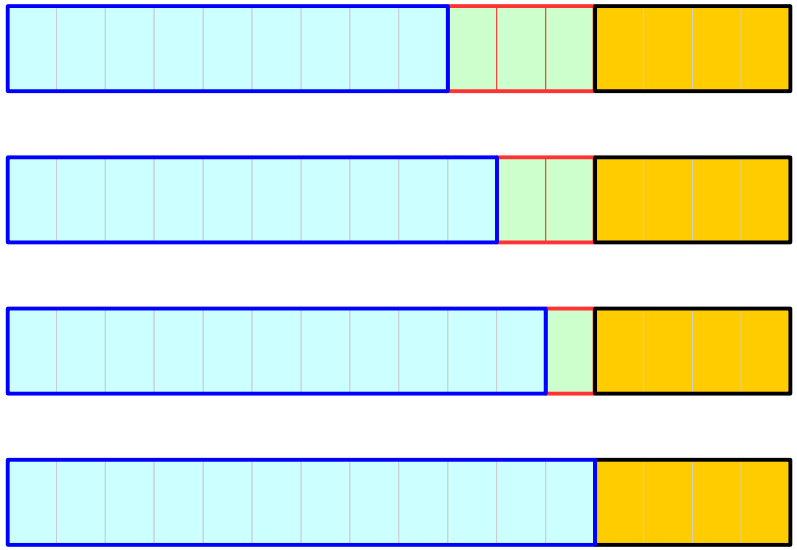
Fully Associative Mapping



1 sets 8-way 8 line / set

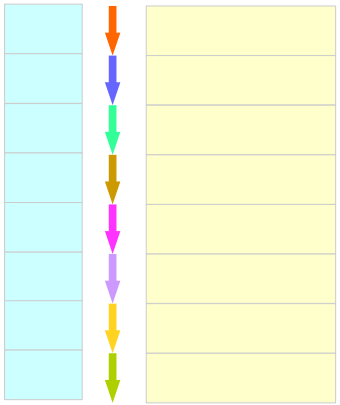
Tag Field



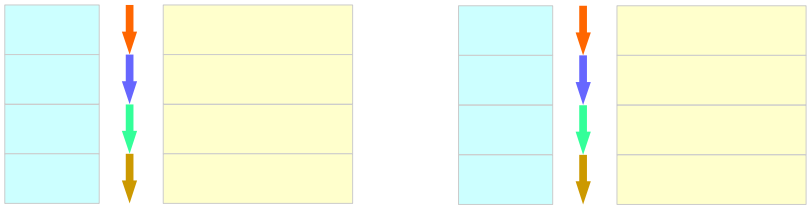


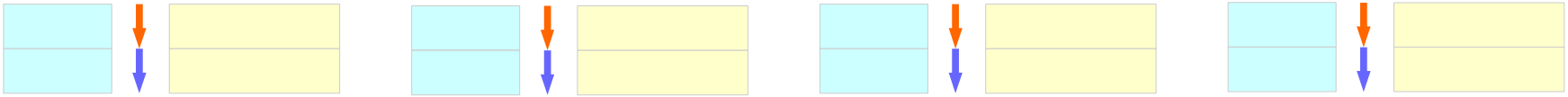
8 sets	1-way	1 line / set
4 sets	2-way	2 lines / set
2 sets	4-way	4 lines / set
1 set	8-way	8 lines / set

Cache Mapping Method (set-view)

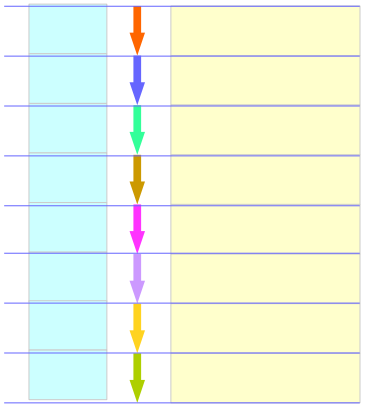
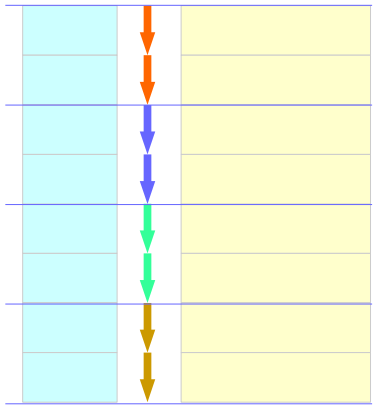
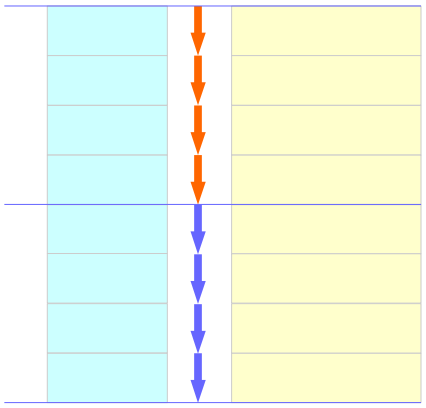
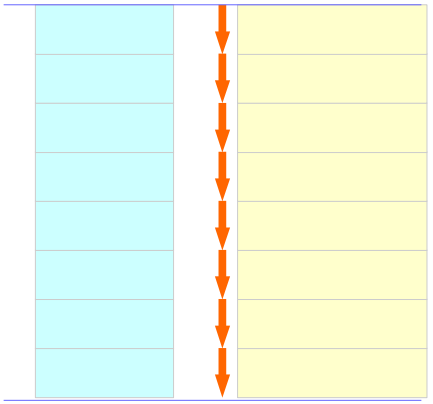


8 sets	1-way	1 line / set
4 sets	2-way	2 lines / set
2 sets	4-way	4 lines / set
1 set	8-way	8 lines / set





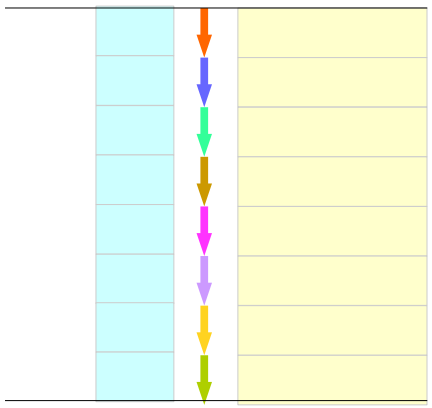
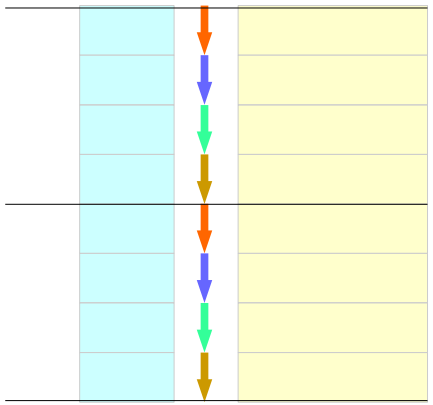
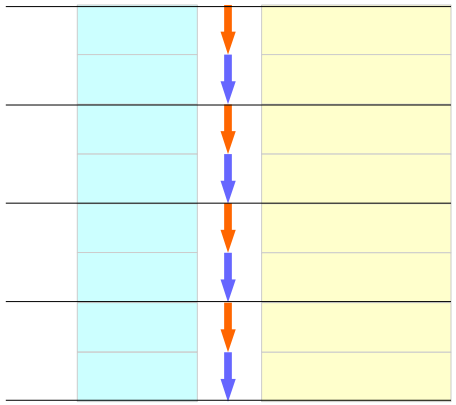
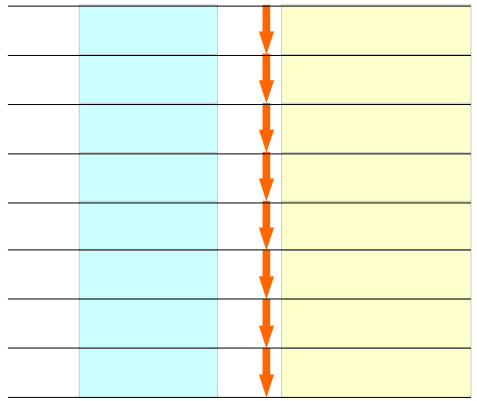
Cache Mapping Method (set-view)

8 sets	1-way	1 line / set
4 sets	2-way	2 lines / set
2 sets	4-way	4 lines / set
1 set	8-way	8 lines / set

• • •

Cache Mapping Method (way-view)

8 sets	1-way	1 line / set
4 sets	2-way	2 lines / set
2 sets	4-way	4 lines / set
1 set	8-way	8 lines / set

• • •

CAM (Content Addressable Memory)

CAM (Content Addressable Memory)

References

[1] http://en.wikipedia.org/
[2] https://en.wikiversity.org/wiki/The_necessities_in_SOC_Design
[3] https://en.wikiversity.org/wiki/The_necessities_in_Digital_Design
[4] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Design
[5] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Architecture
[6] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Organization
[7] https://en.wikiversity.org/wiki/Understanding_Embedded_Software
[8] Digital Systems, Hill, Peterson, 1987