Memory Devices

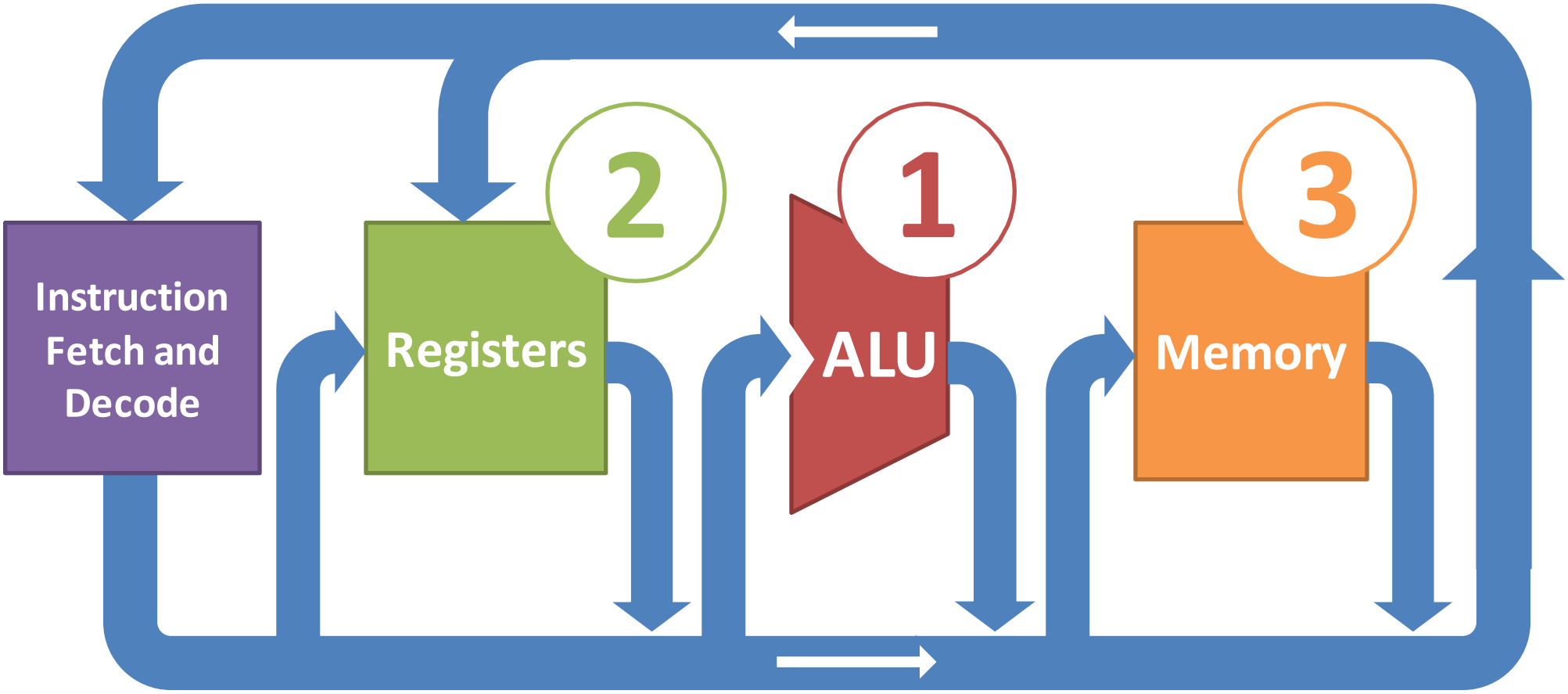
Small: **Register file** *(group of numbered registers)*

Medium: **SRAM** *(Static Random Access Memory)*

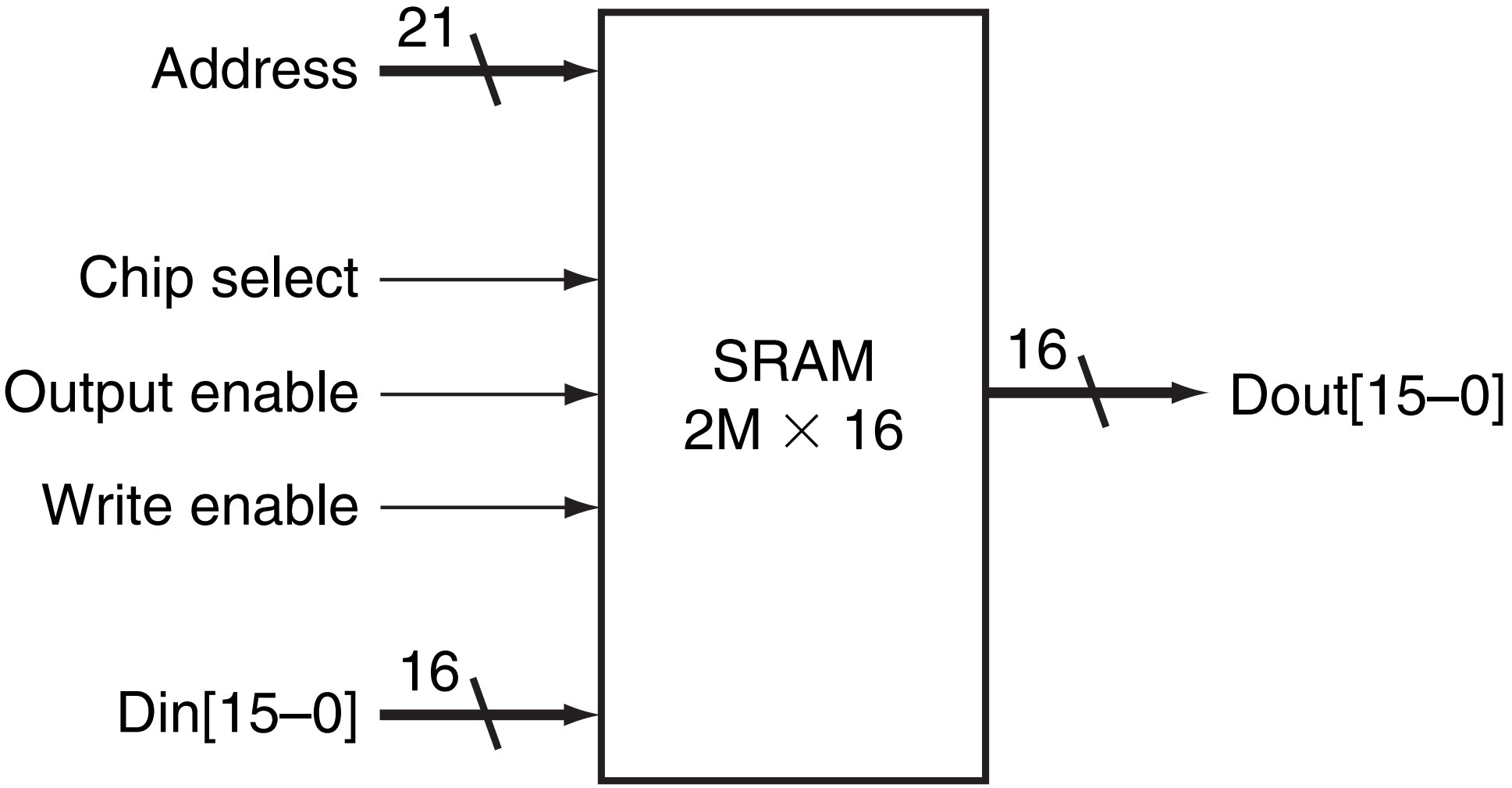
Large: **DRAM** *(Dynamic Random Access Memory)*

Future?

Processor: Data Path Components



SRAM: Static Random Access Memory



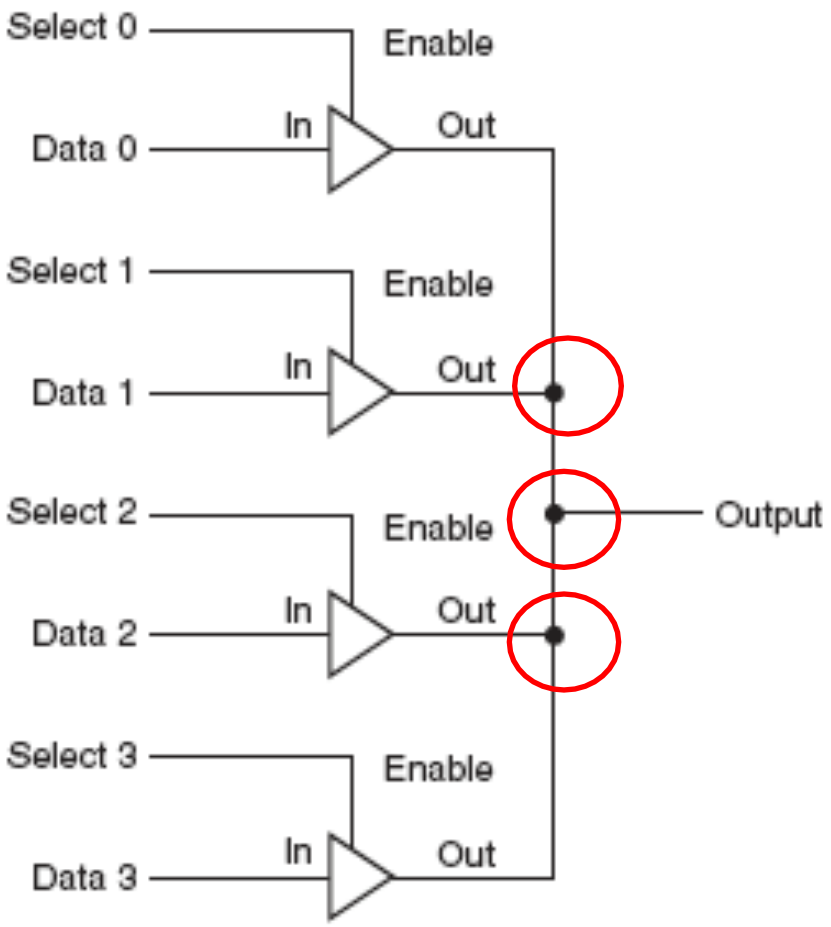
SRAM read port: data out

Large register files are impractical.

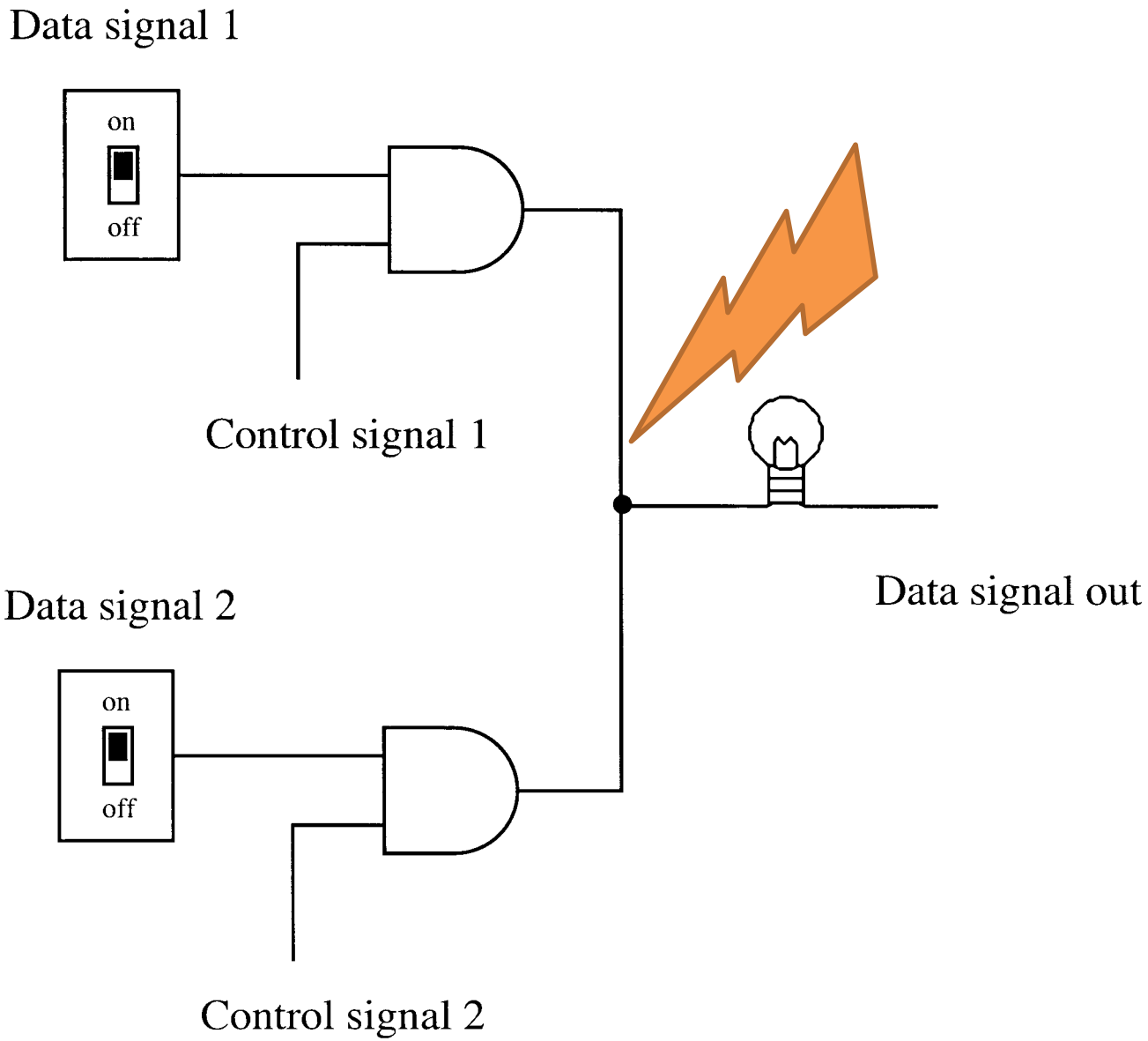
Big MUX = *significant* gate delay.

Large memories use a shared output line.

No central gates/MUX to choose output!

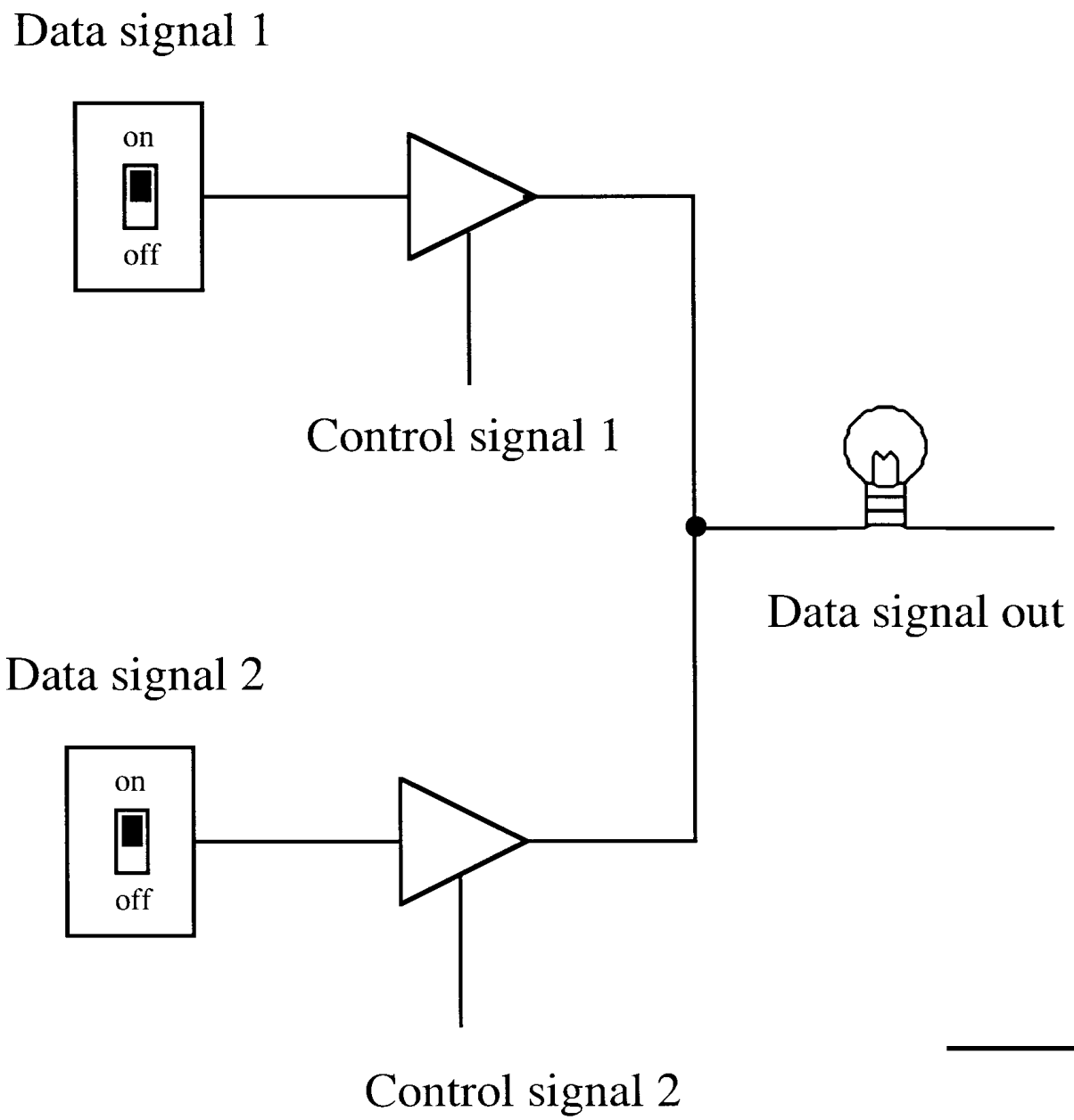


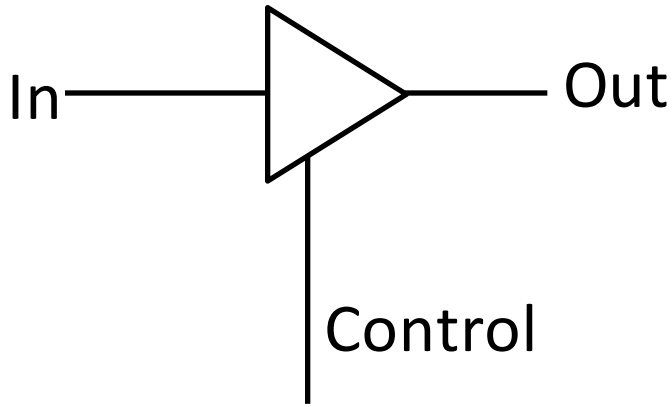
Wired ORs (don't try this at home/in the lab, kids)



[Danger, Will Robinson!]

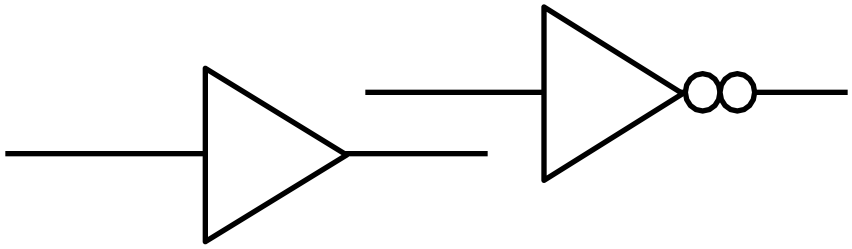
(noninverting) **tristate buffers**





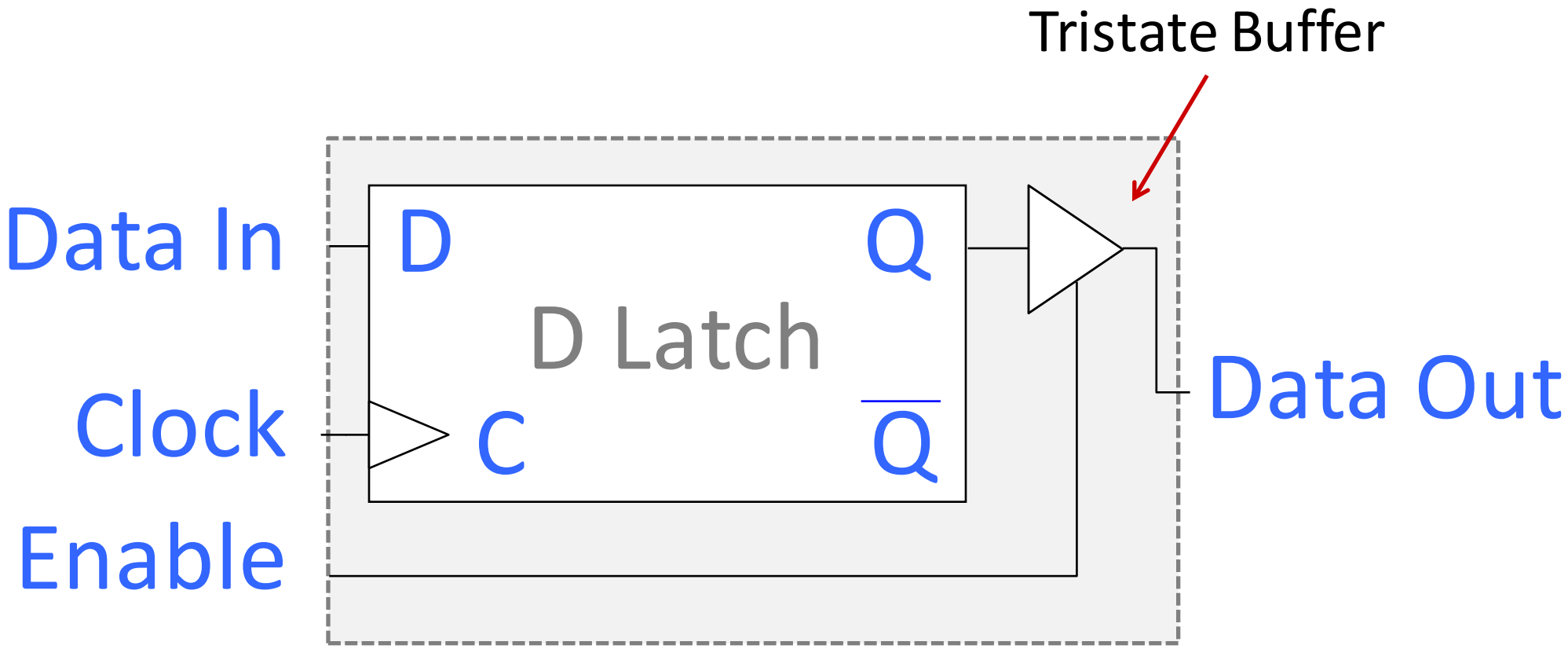
In	Control	Out
0	0	Z
1	0	Z
0	1	0
1	1	1

(active high)

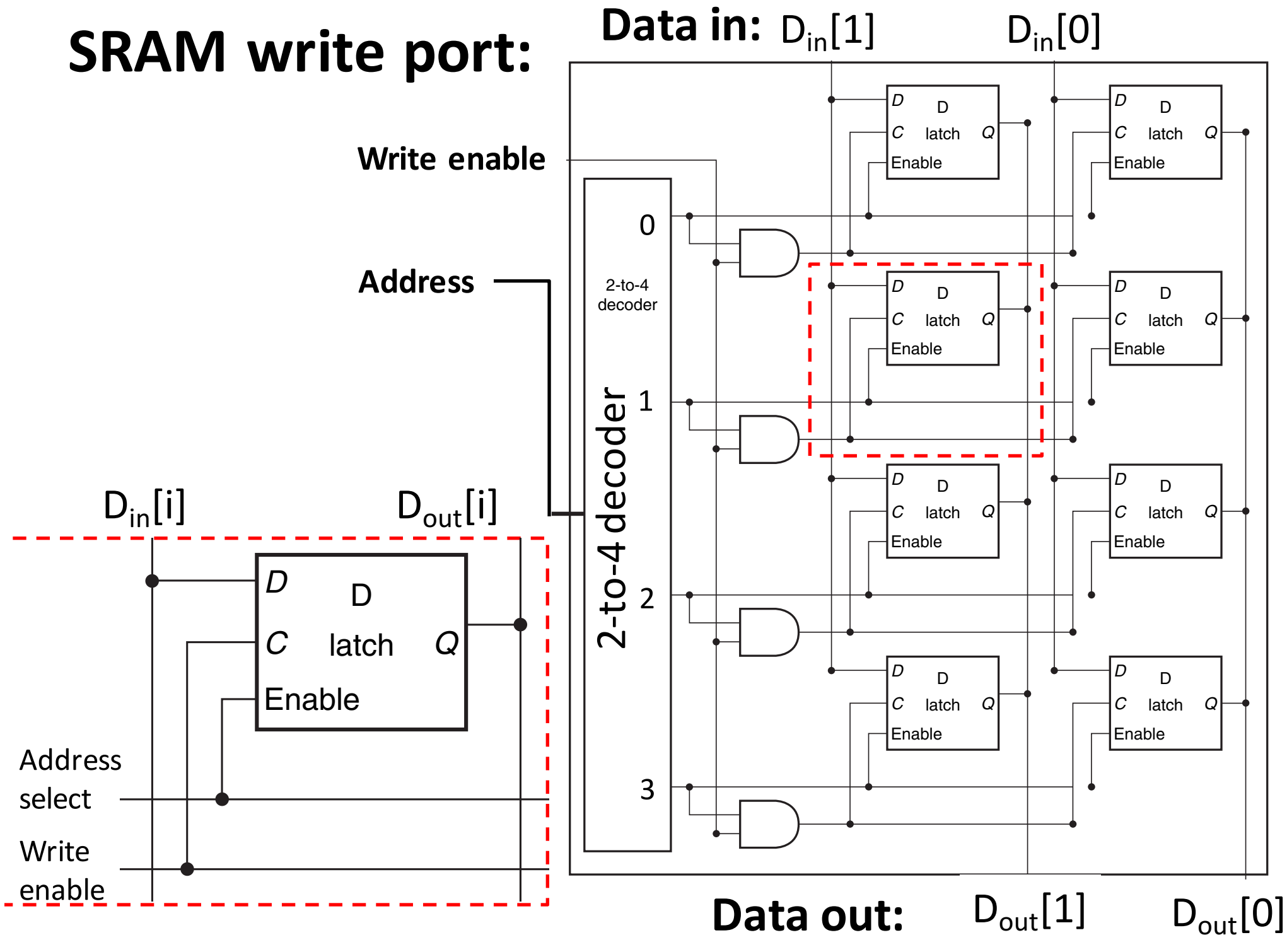


SRAM cell

one option

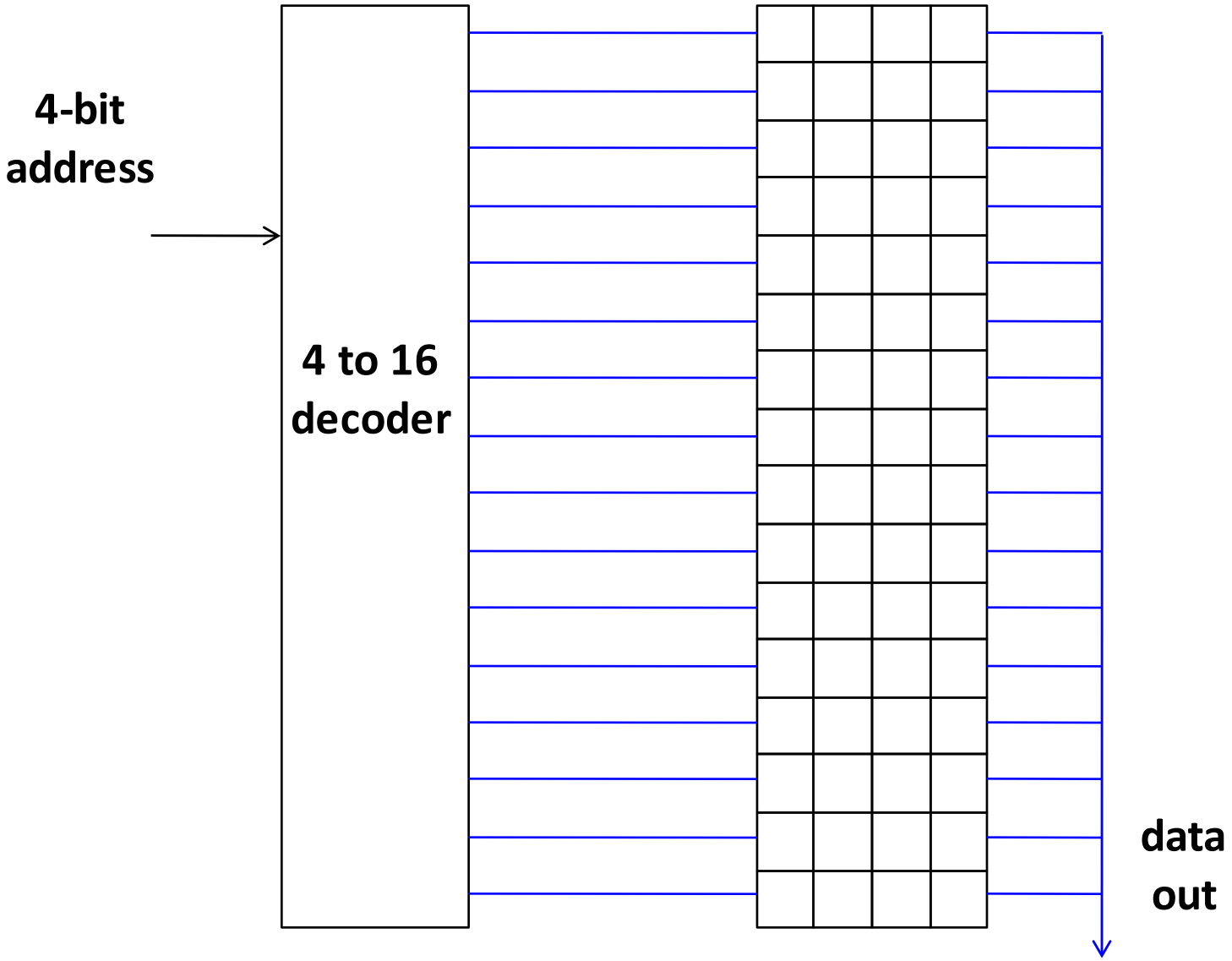


SRAM write port:

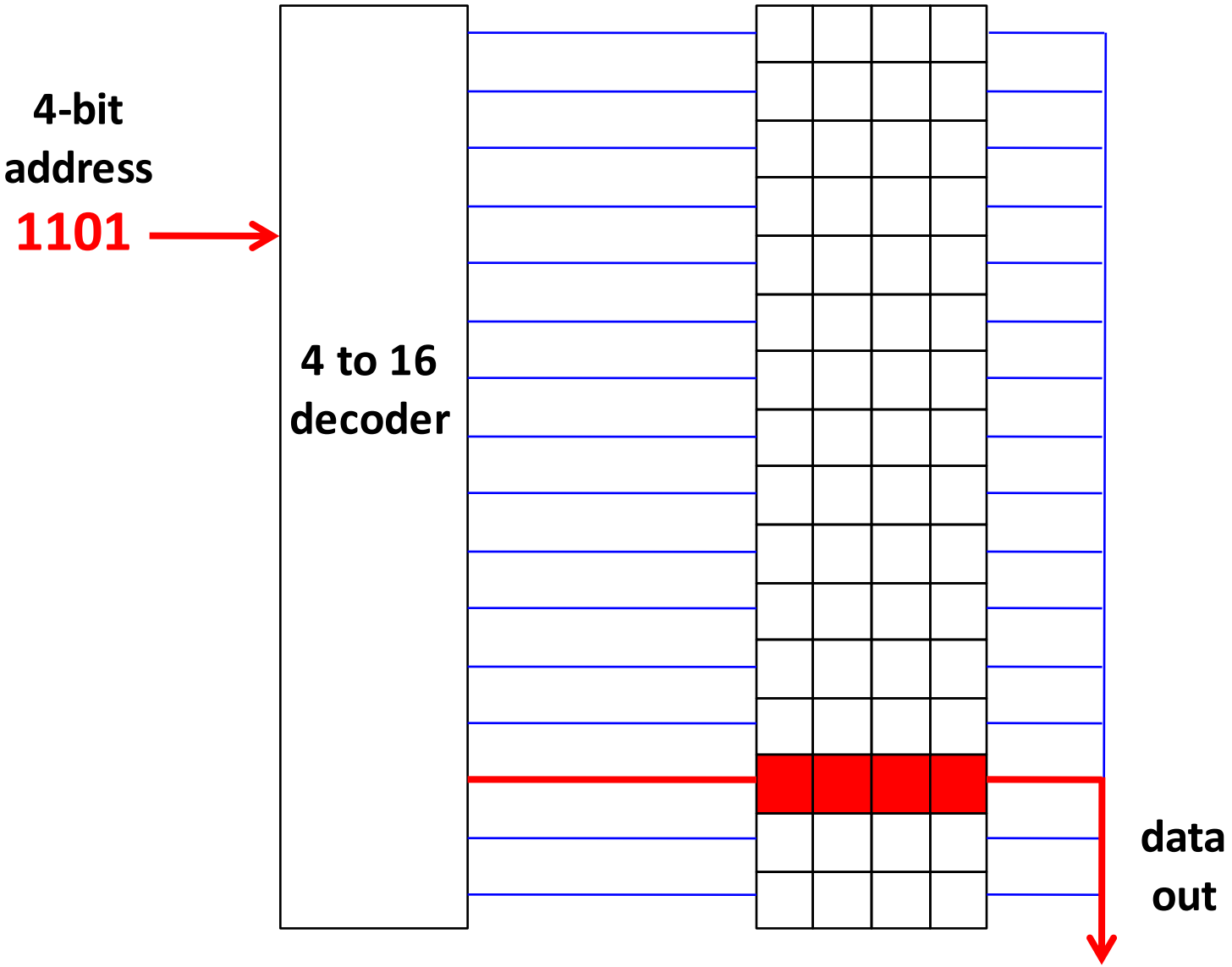


Organization of a 16 x 4 SRAM

(one option)

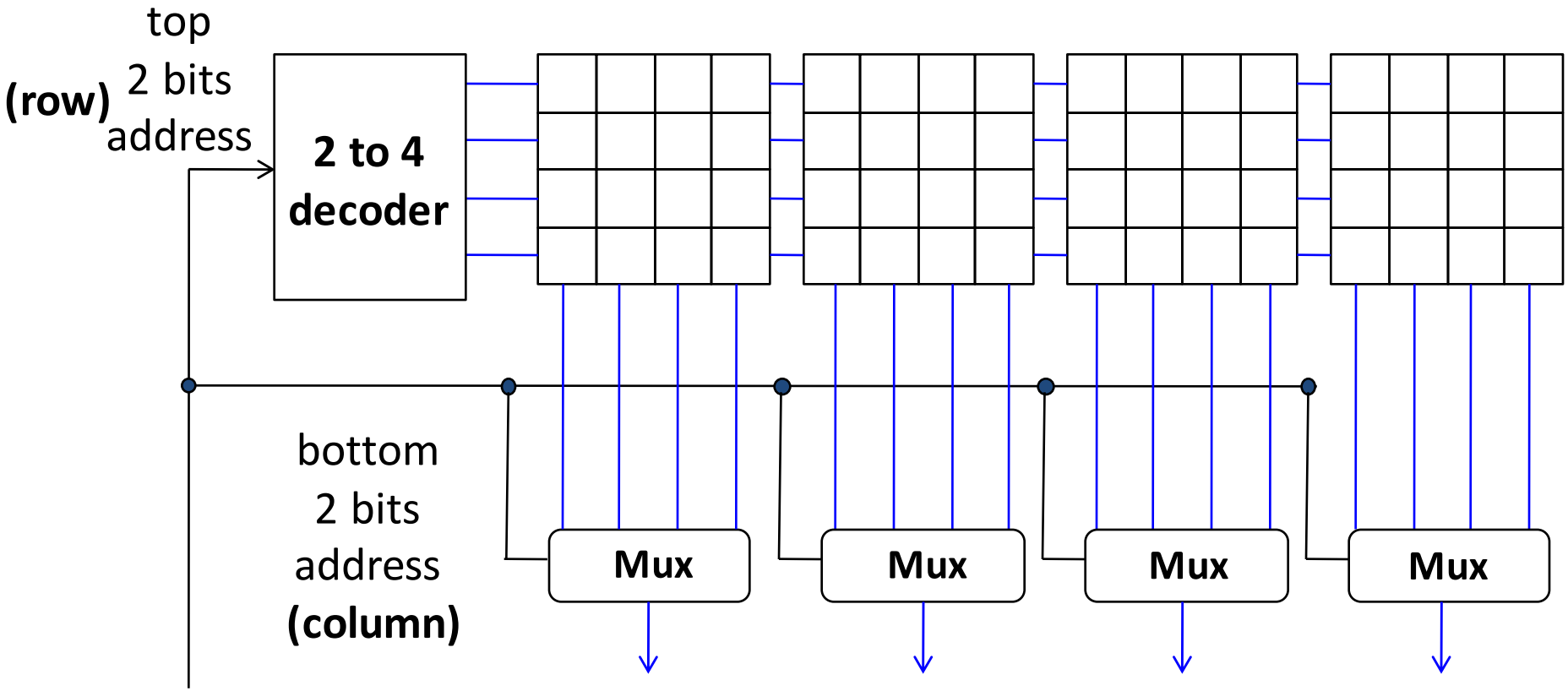


Selecting location 1101



Another organization of a 16 x 4 SRAM

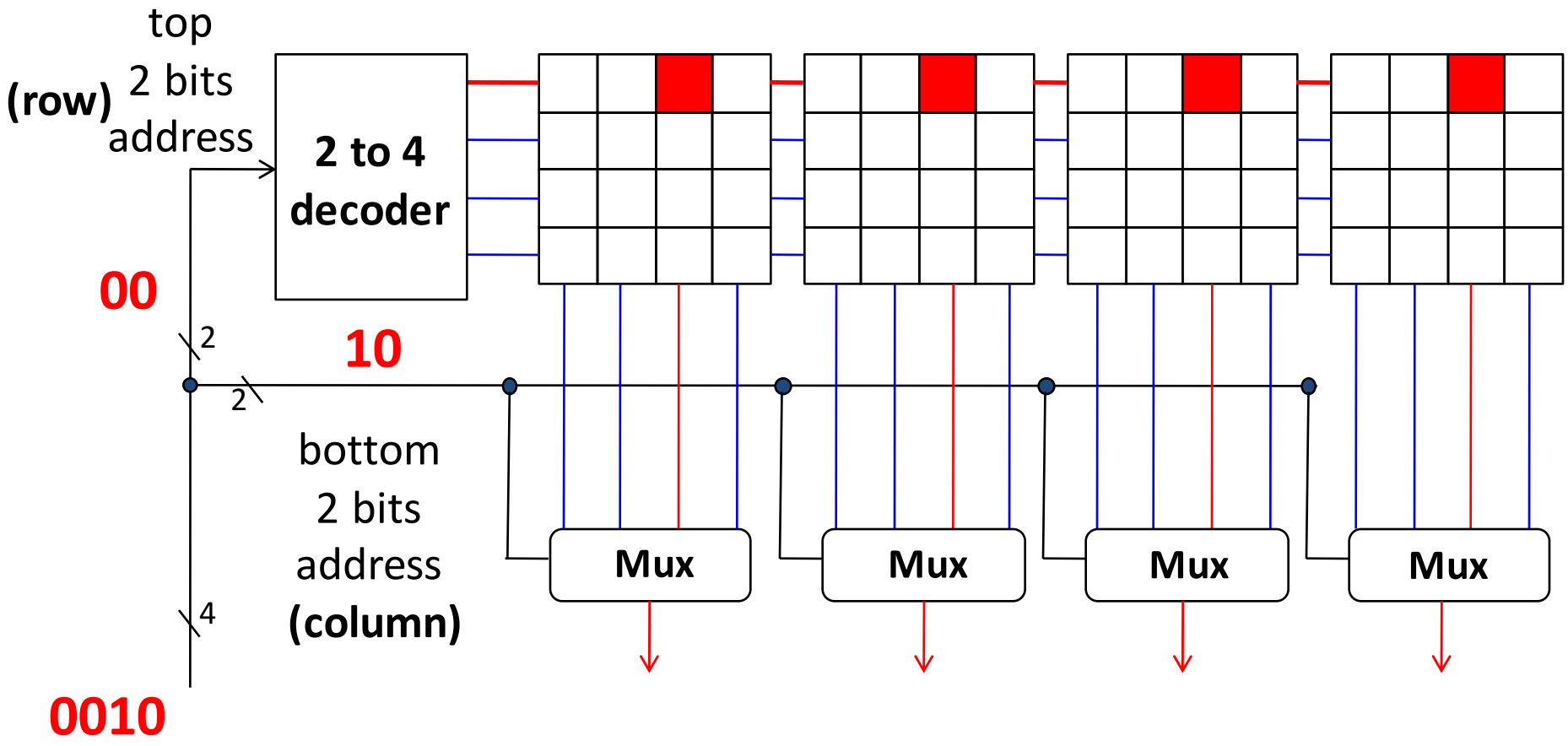
Split-level row/column addressing = physical multidimensional array!



Notice the smaller decoder... how does this affect timing?

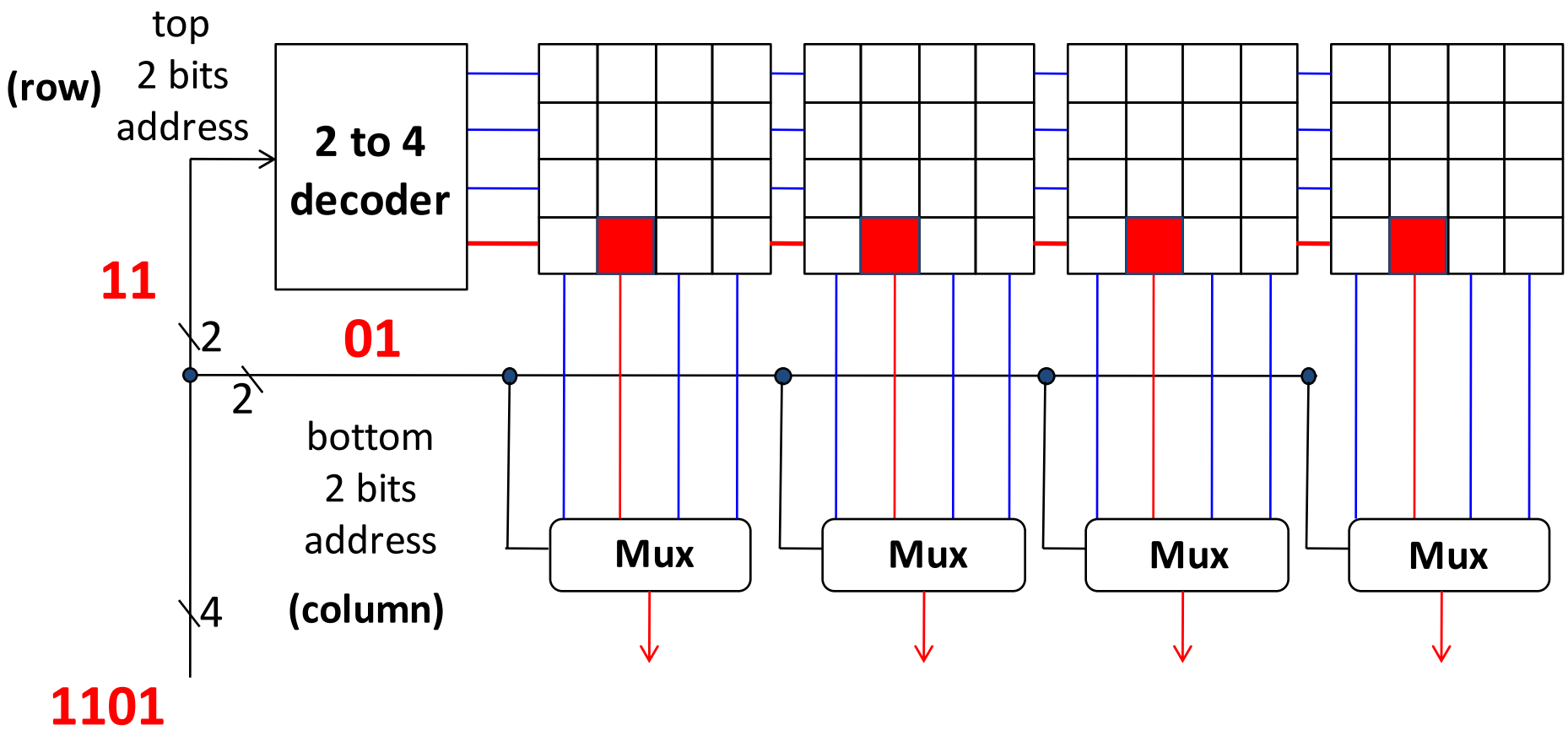
Selecting location 0010

Nibbles "striped" across 4 smaller memories.

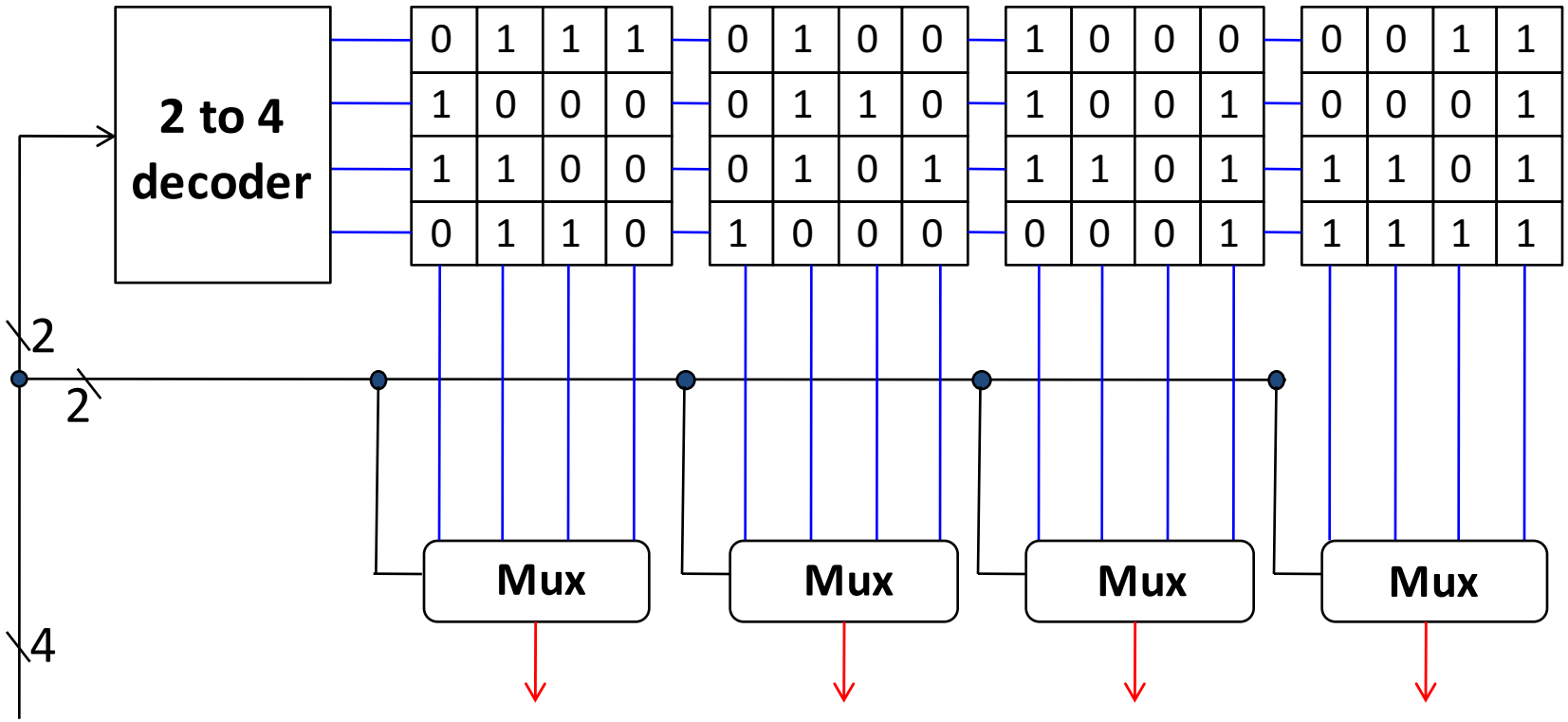


Selecting location 1101

Nibbles "striped" across 4 smaller memories.



What value does location 1010 hold?

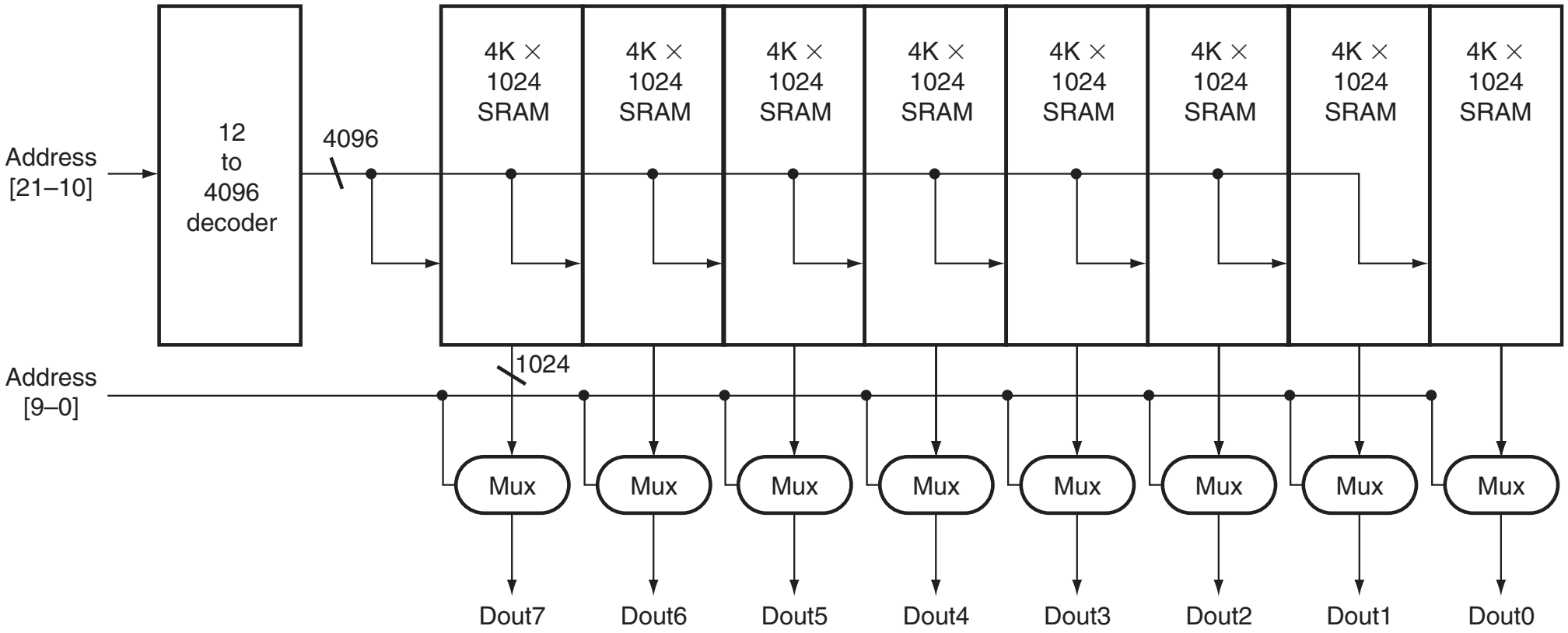


1010

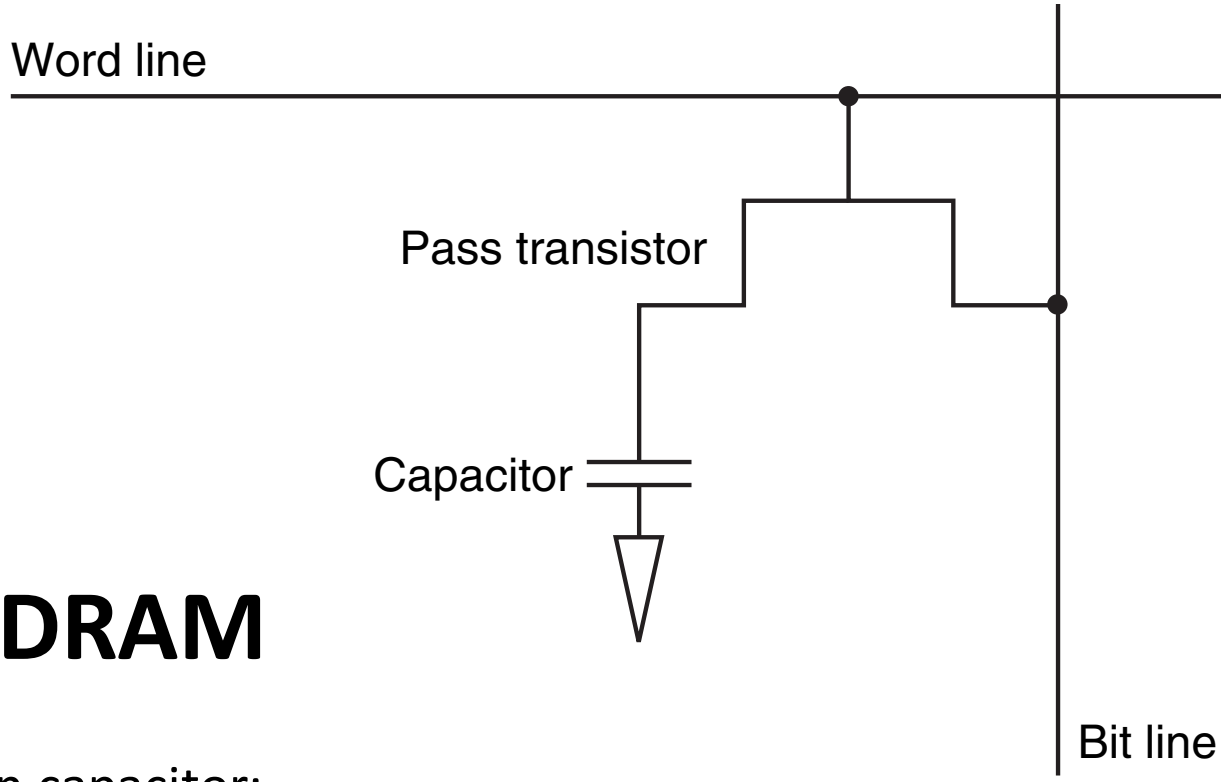
Organization of a 4M x 8 SRAM

(one option)

= 4 MB memory, size of a large cache for modern laptop



In practice, single set of data lines often time-shared for read (out)/write (in).



Dynamic RAM = DRAM

DRAM stores bit as charge on capacitor:

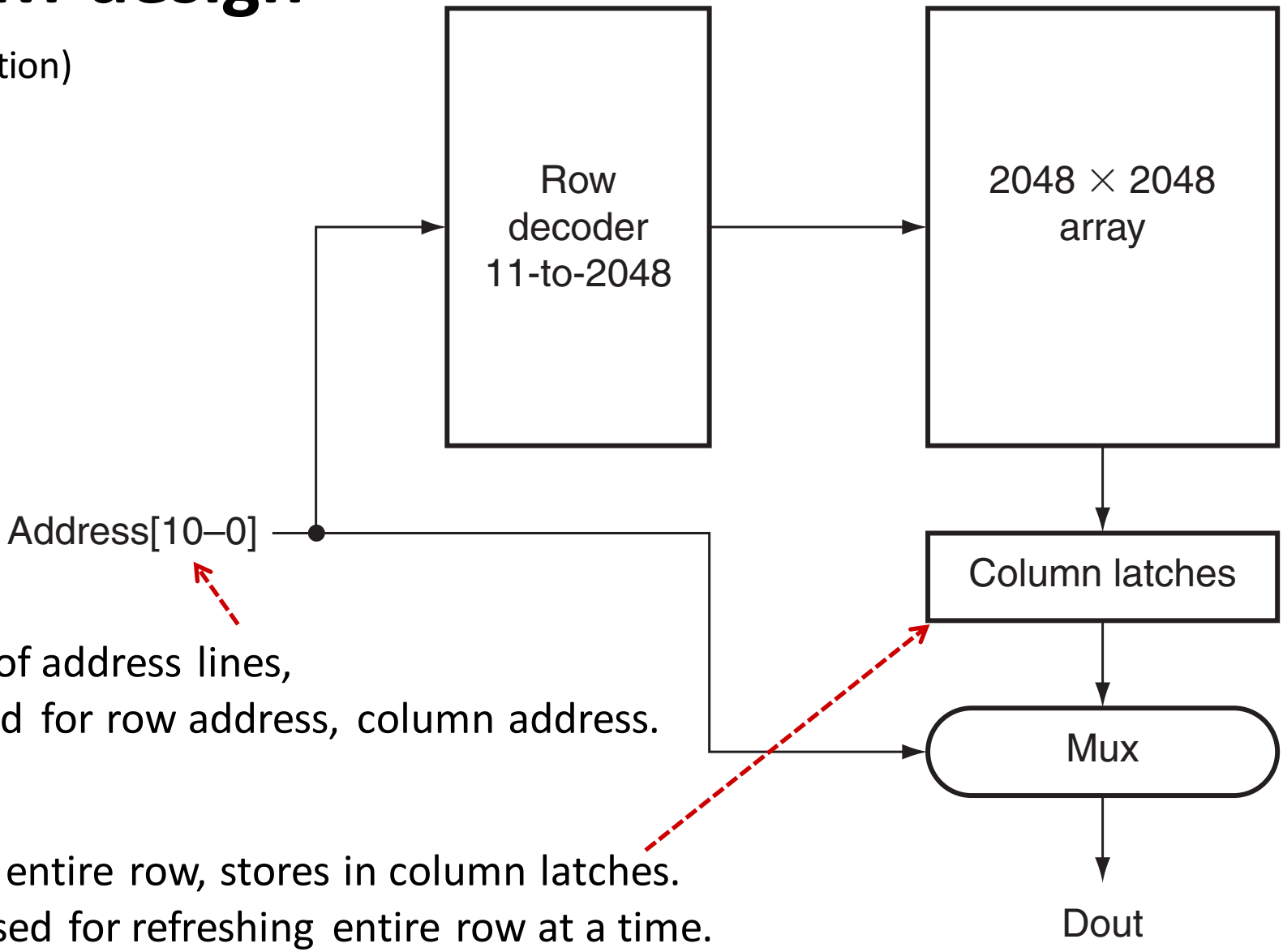
- 1 transistor accesses stored charge.
- requires periodic refresh = read-write (dynamic power)

SRAM stores bit on pair of inverting gates:

- several transistors
- requires continuous (static) power.

DRAM design

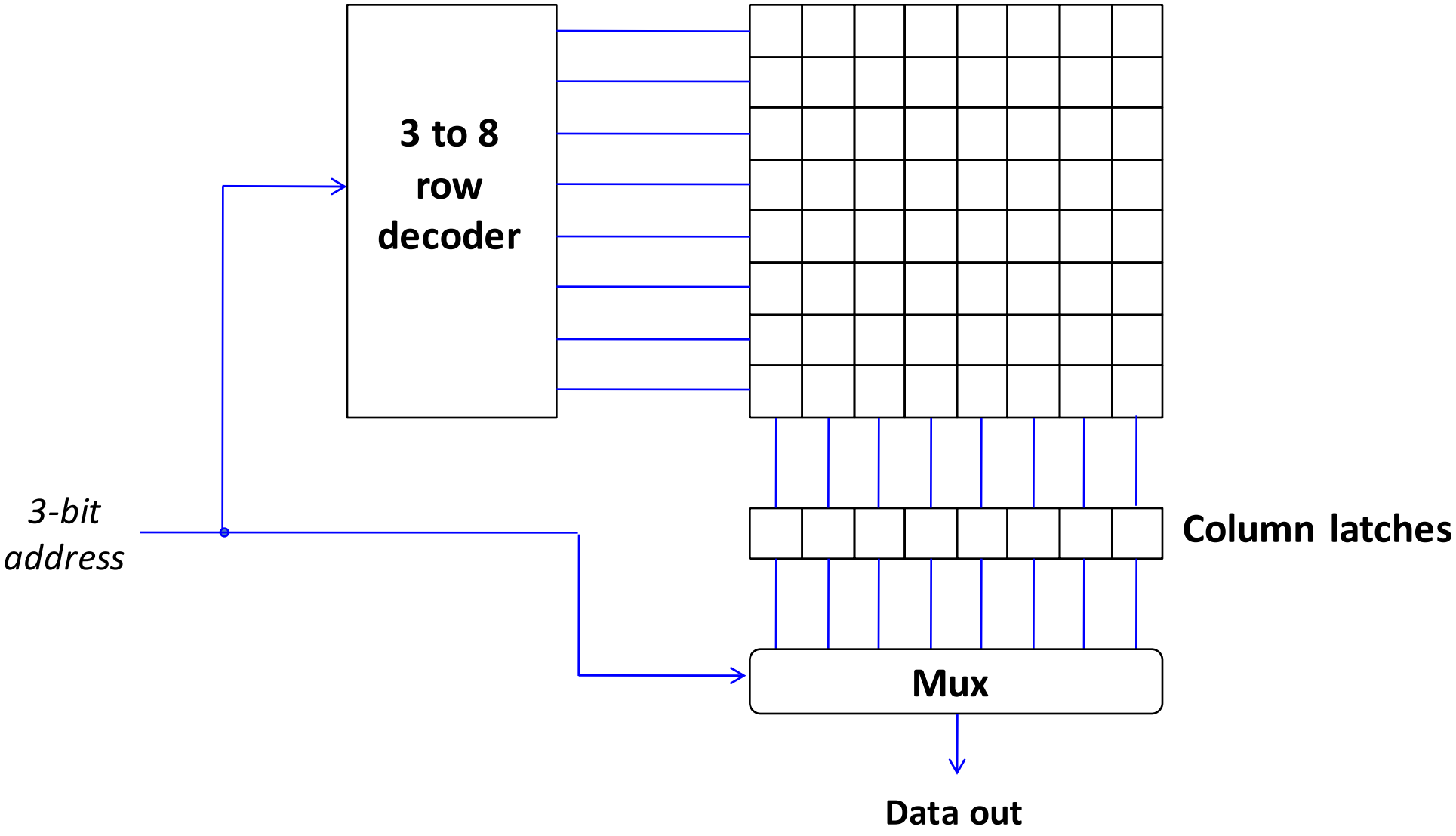
(one option)



Single set of address lines,
time-shared for row address, column address.

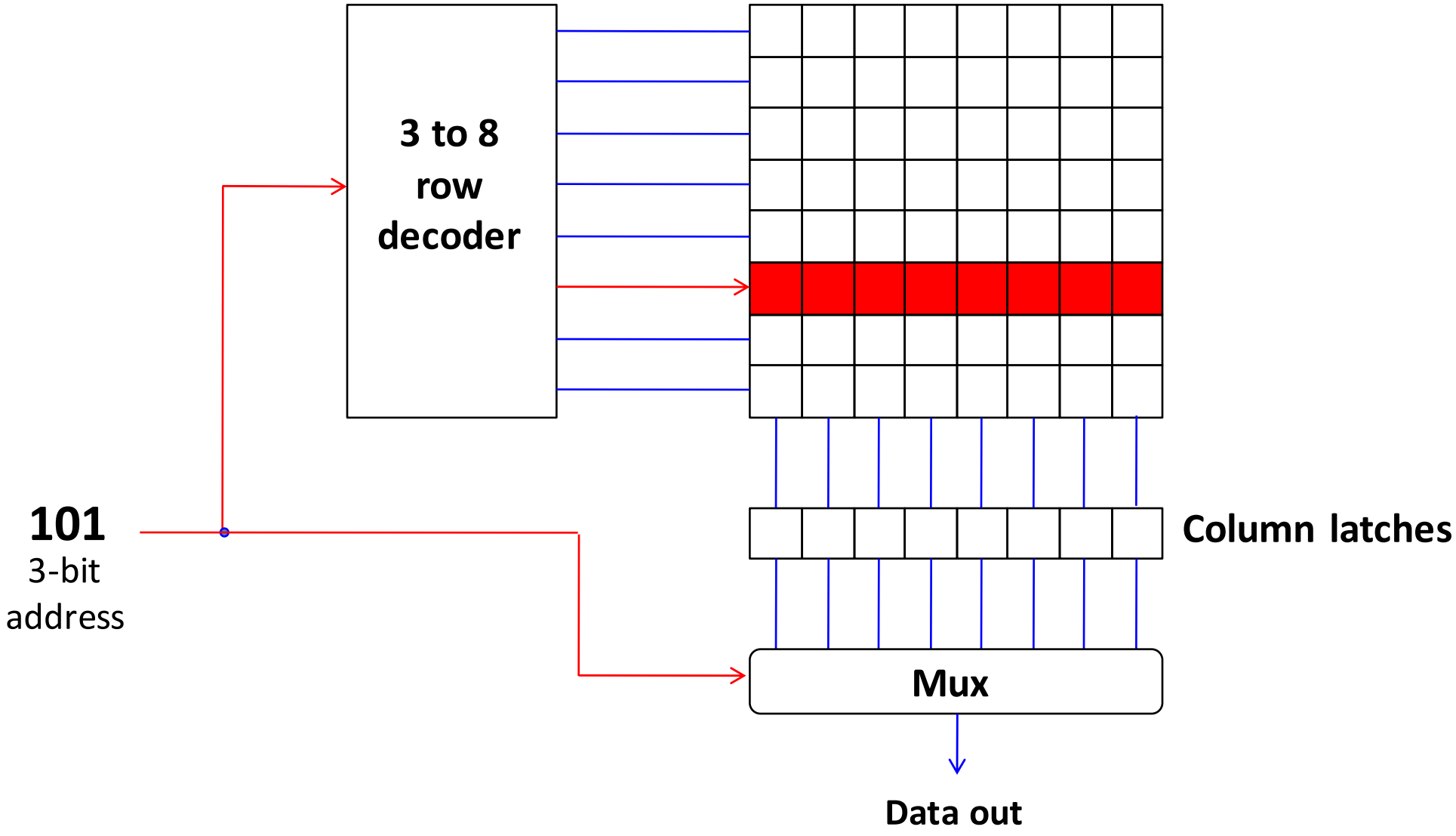
Accesses entire row, stores in column latches.
Mainly used for refreshing entire row at a time.
Accessing other columns in same row again cheaper…?

64-bit DRAM



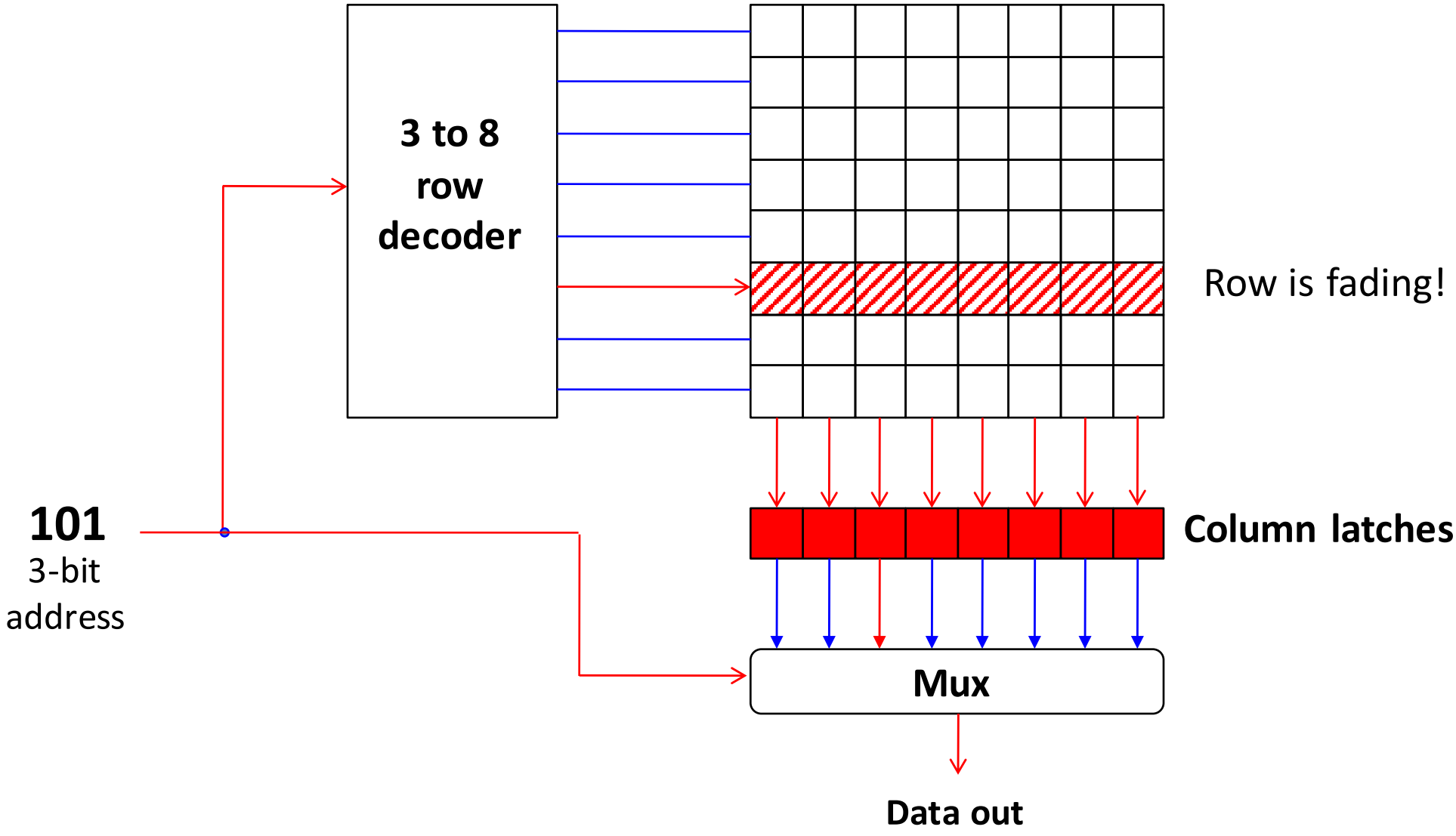
Reading bit at address 101011

1. Select row



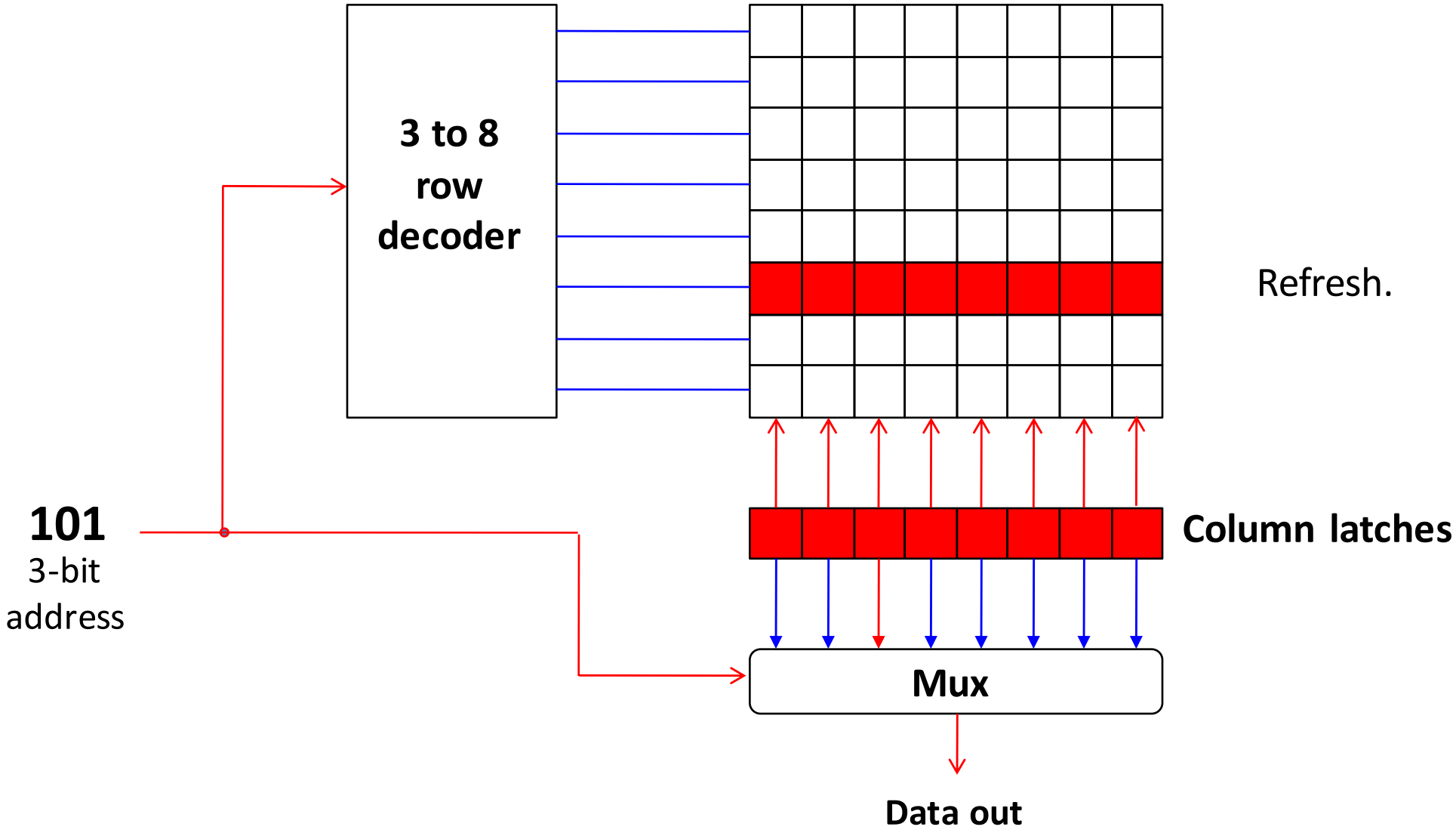
Reading bit at address 101011

2. Copy row to latches



Reading bit at address 101011

3. Refresh row from latches



Reading bit at address 101011

4. Select column from latches