

---

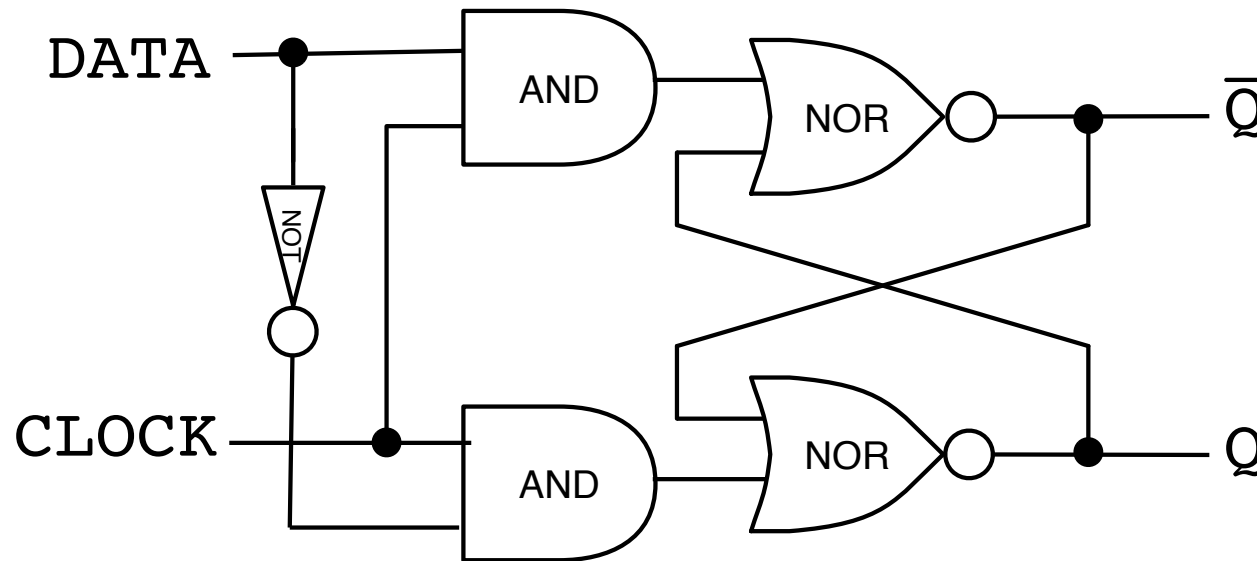
# Memory

Philipp Koehn

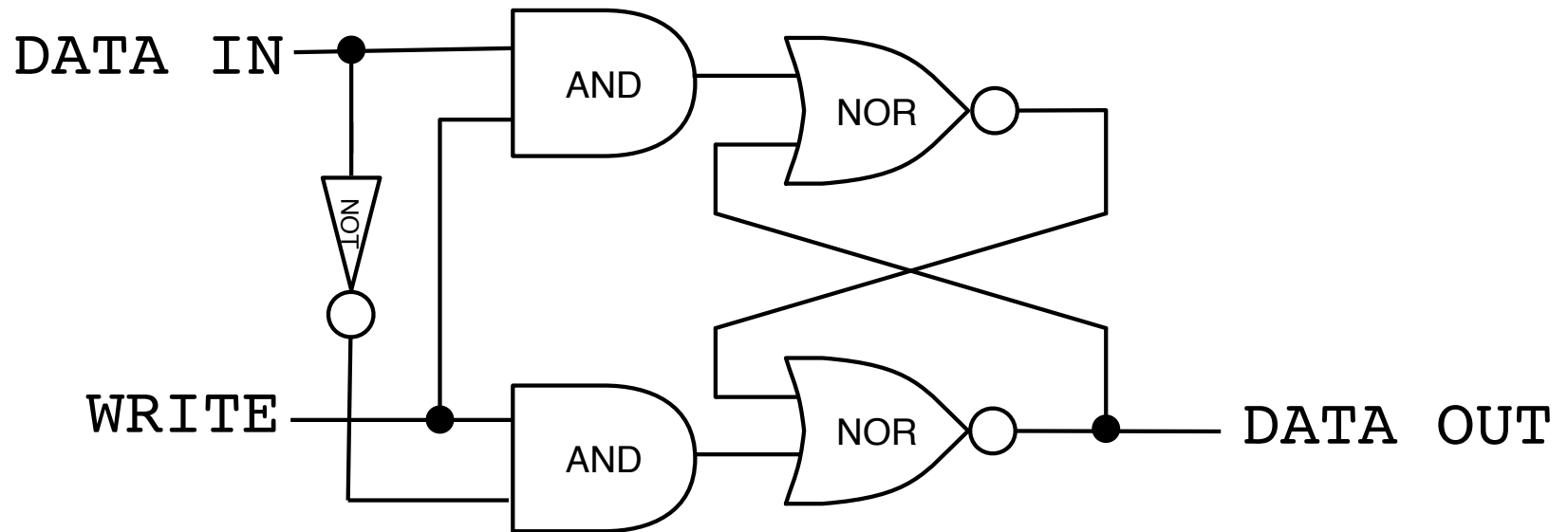
9 September 2019



# D-Type Level-Triggered Latch



# Slightly Modified



# Operations



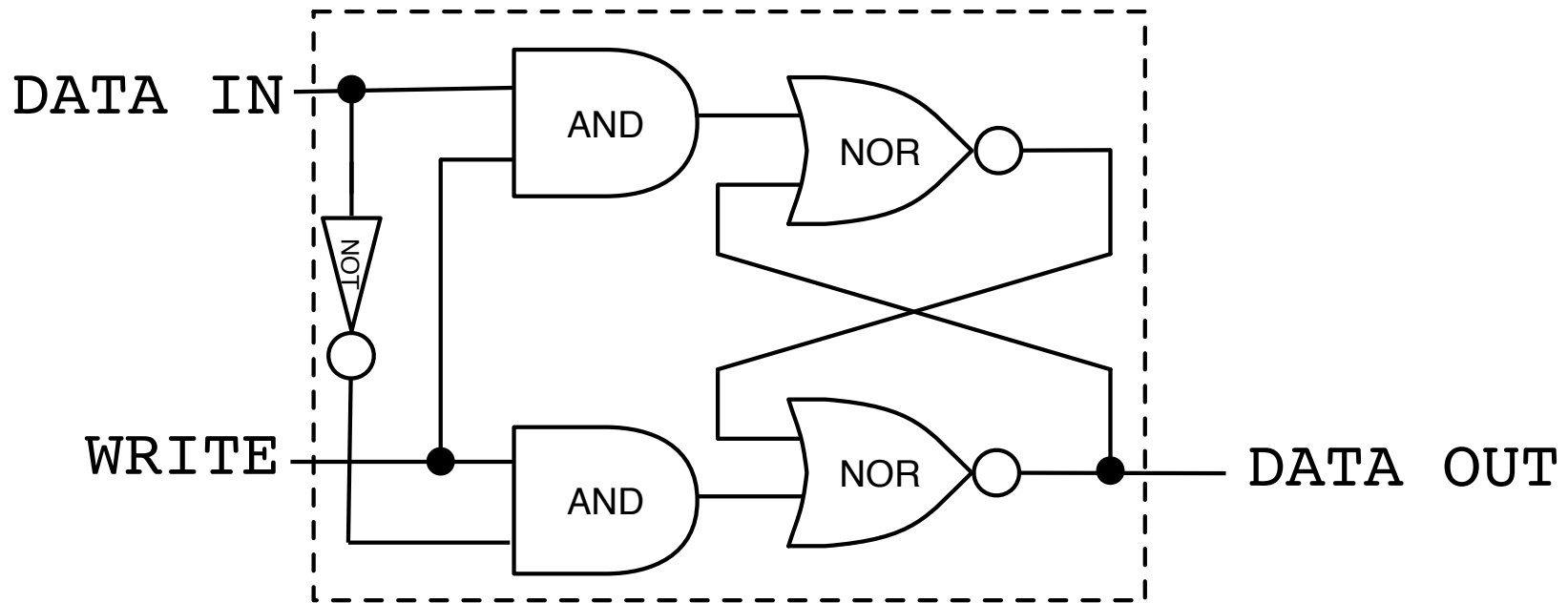
- Circuit latches on one bit of memory and keeps it around
- Truth table

Data-In	Write	Data-Out
0	1	0
1	1	1
X	0	Data

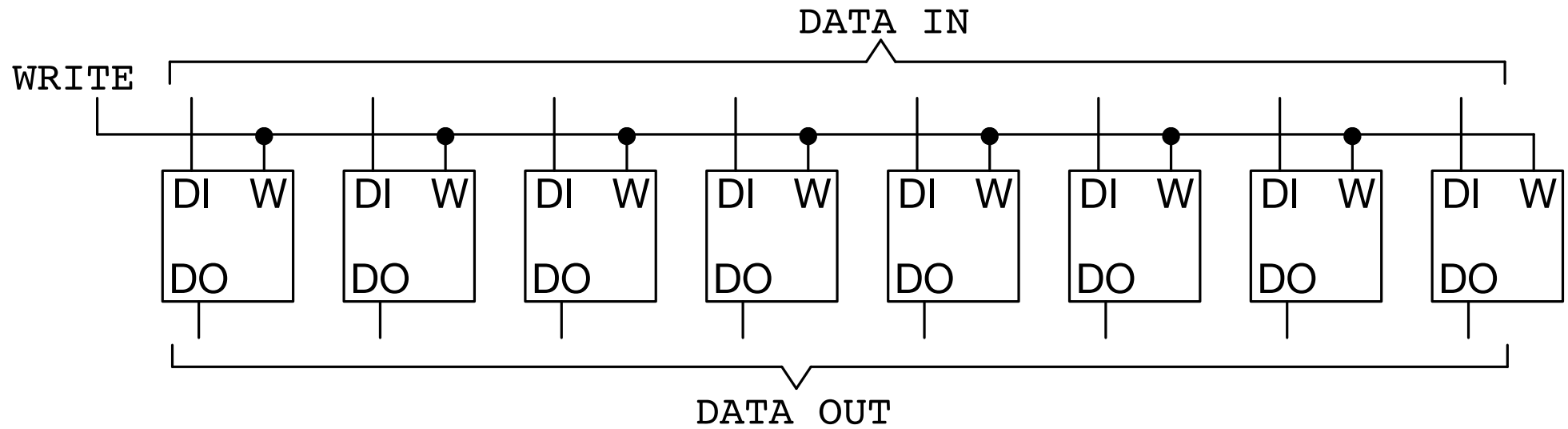
- Can write 1 bit and read content

# multi-bit storage

# 1 Bit Memory



# 8 Bit Memory



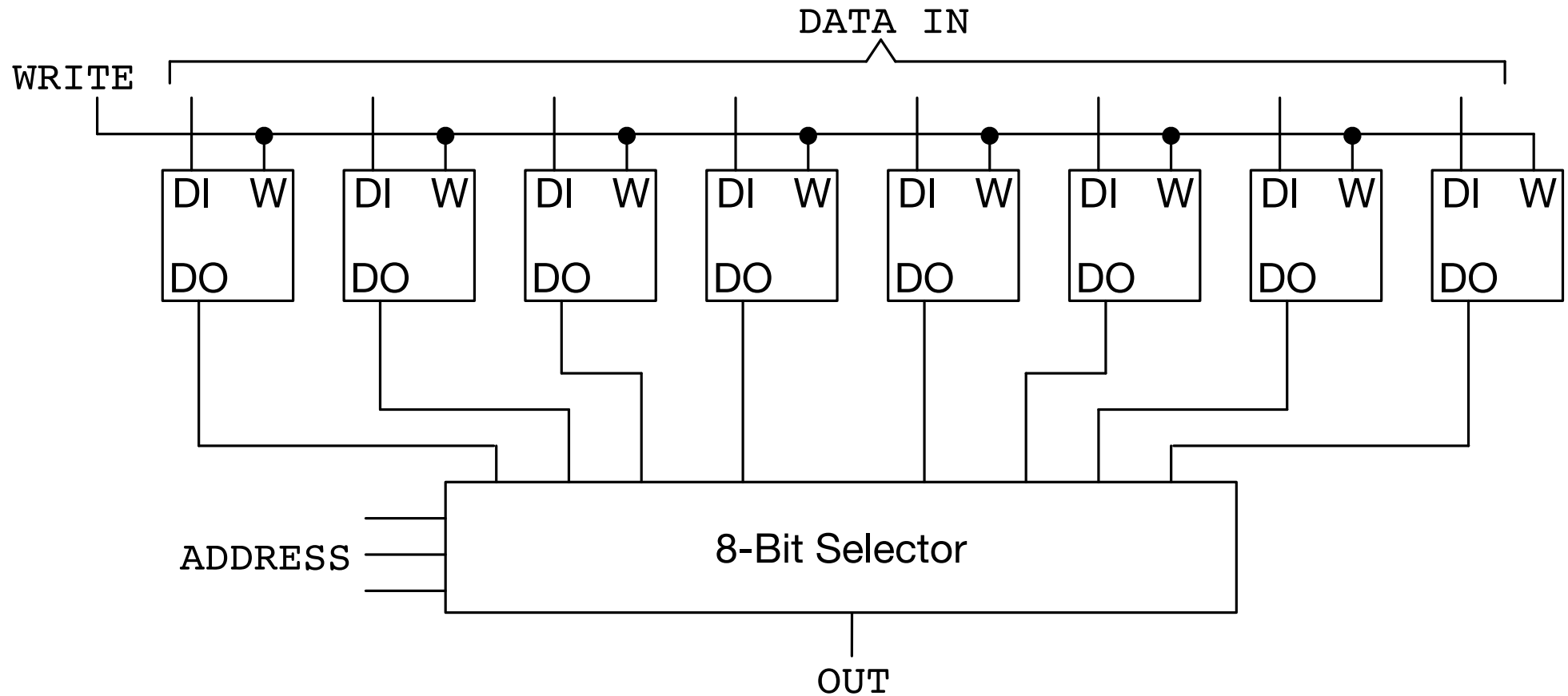
# Output Selector



- 8 Bit Latch contains 8 bits
- Now: only read 1 bit at a time
- Select the bit with an address
- Input: address
- Output: bit value



# Output Selector



# Output Selector

- Truth table

Address			Output
A2	A1	A0	OUT
0	0	0	D <sub>0</sub>
0	0	1	D <sub>1</sub>
0	1	0	D <sub>2</sub>
0	1	1	D <sub>3</sub>
1	0	0	D <sub>4</sub>
1	0	1	D <sub>5</sub>
1	1	0	D <sub>6</sub>
1	1	1	D <sub>7</sub> ■

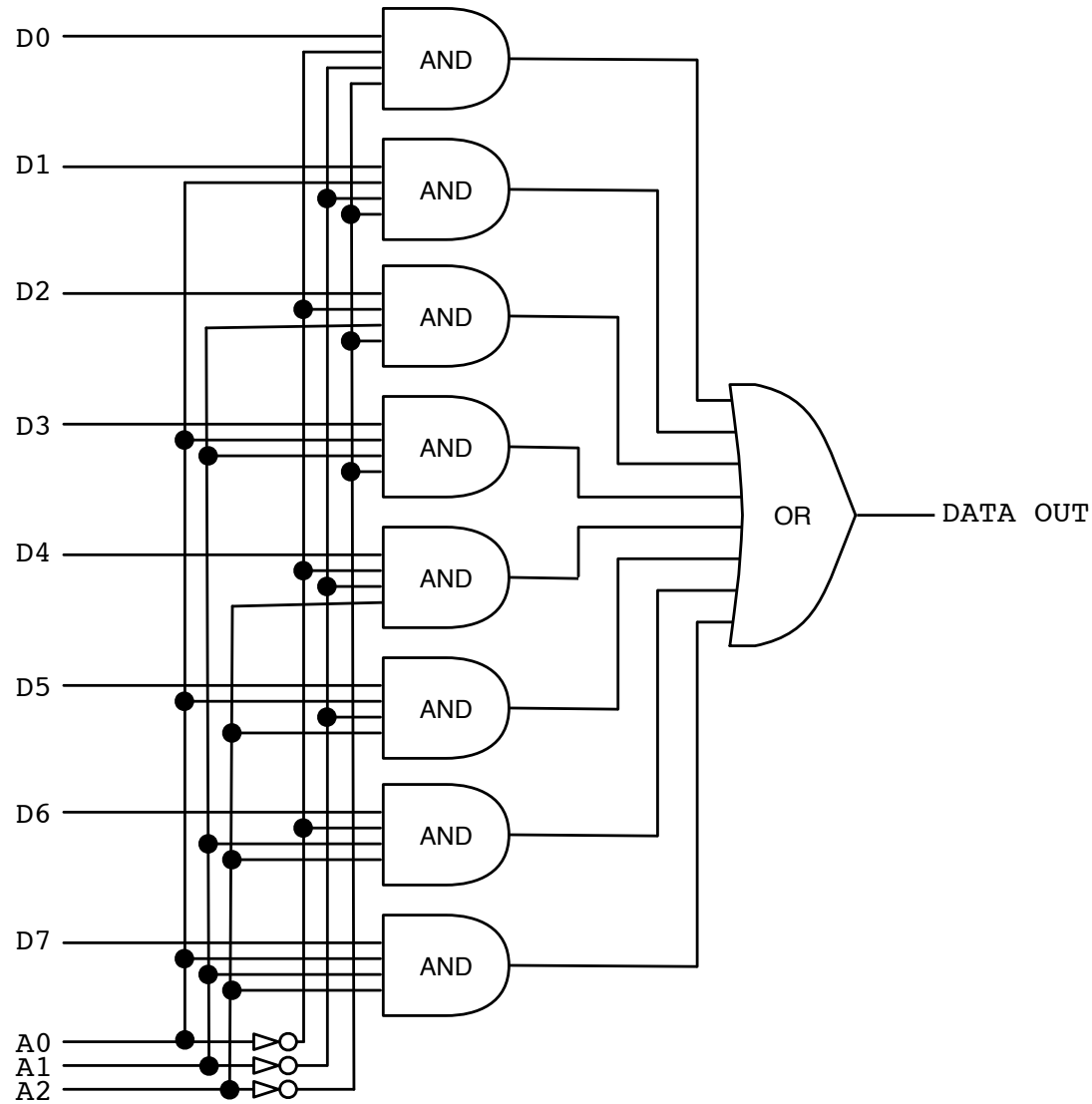
- What Boolean operation returns the correct value for address 000? ■  
(NOT A2) AND (NOT A1) AND (NOT A0) AND D0

# Output Selector

- Full Boolean formula

```
( (NOT A2) AND (NOT A1) AND (NOT A0) AND D0 ) OR
( (NOT A2) AND (NOT A1) AND      A0  AND D1 ) OR
( (NOT A2) AND      A1  AND (NOT A0) AND D2 ) OR
( (NOT A2) AND      A1  AND      A0  AND D3 ) OR
(      A2  AND (NOT A1) AND (NOT A0) AND D4 ) OR
(      A2  AND (NOT A1) AND      A0  AND D5 ) OR
(      A2  AND      A1  AND (NOT A0) AND D6 ) OR
(      A2  AND      A1  AND      A0  AND D7 )
```

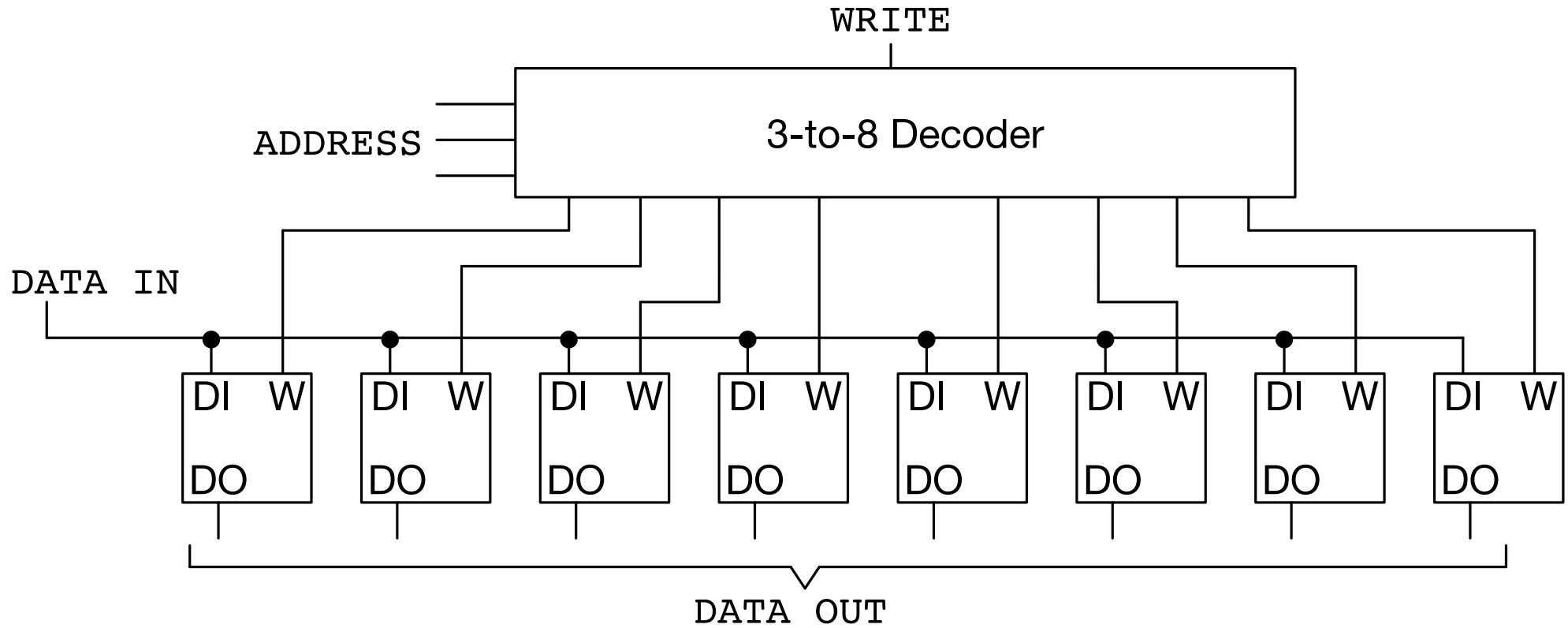
# Output Selector



# Input Decoder

- 8 Bit Latch allows 8 bits to be written at the same time
- Now: only write 1 bit at a time
- Select the bit with an address
- Input
  - address
  - write flag
  - data bit

# Input Decoder



# Input Decoder

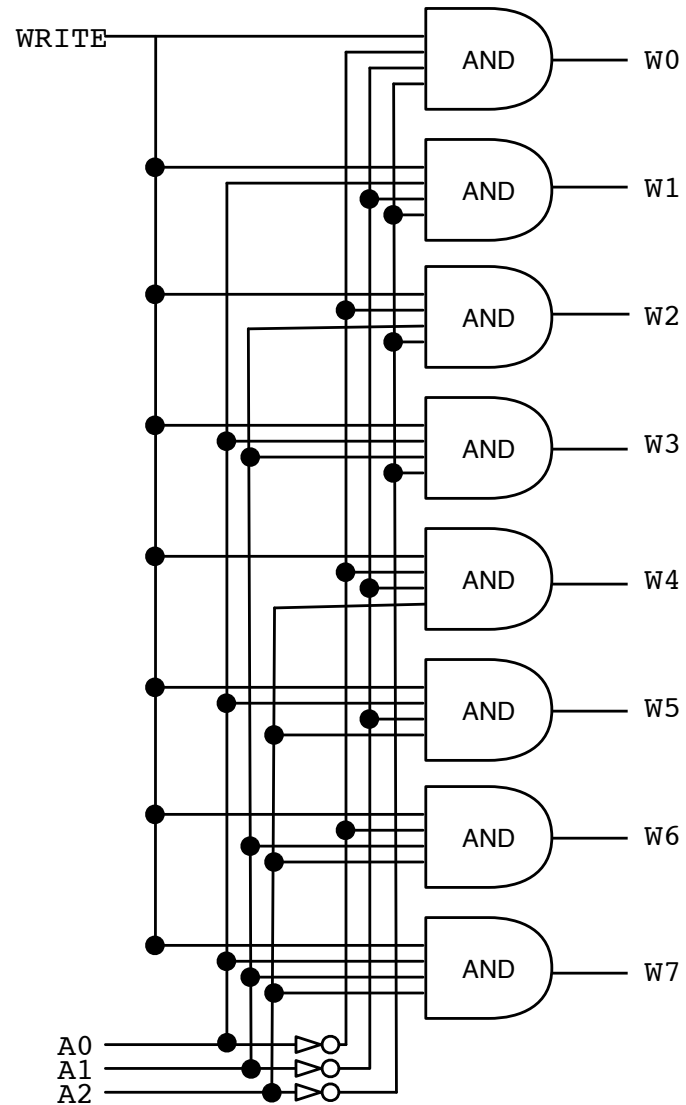
- Truth table

Address			Output							
A2	A1	A0	W7	W6	W5	W4	W3	W2	W1	W0
0	0	0	0	0	0	0	0	0	0	WRITE
0	0	1	0	0	0	0	0	0	WRITE	0
0	1	0	0	0	0	0	0	WRITE	0	0
0	1	1	0	0	0	0	WRITE	0	0	0
1	0	0	0	0	0	WRITE	0	0	0	0
1	0	1	0	0	WRITE	0	0	0	0	0
1	1	0	0	WRITE	0	0	0	0	0	0
1	1	1	WRITE	0	0	0	0	0	0	0

- What Boolean operation returns the correct value for output W0? ■

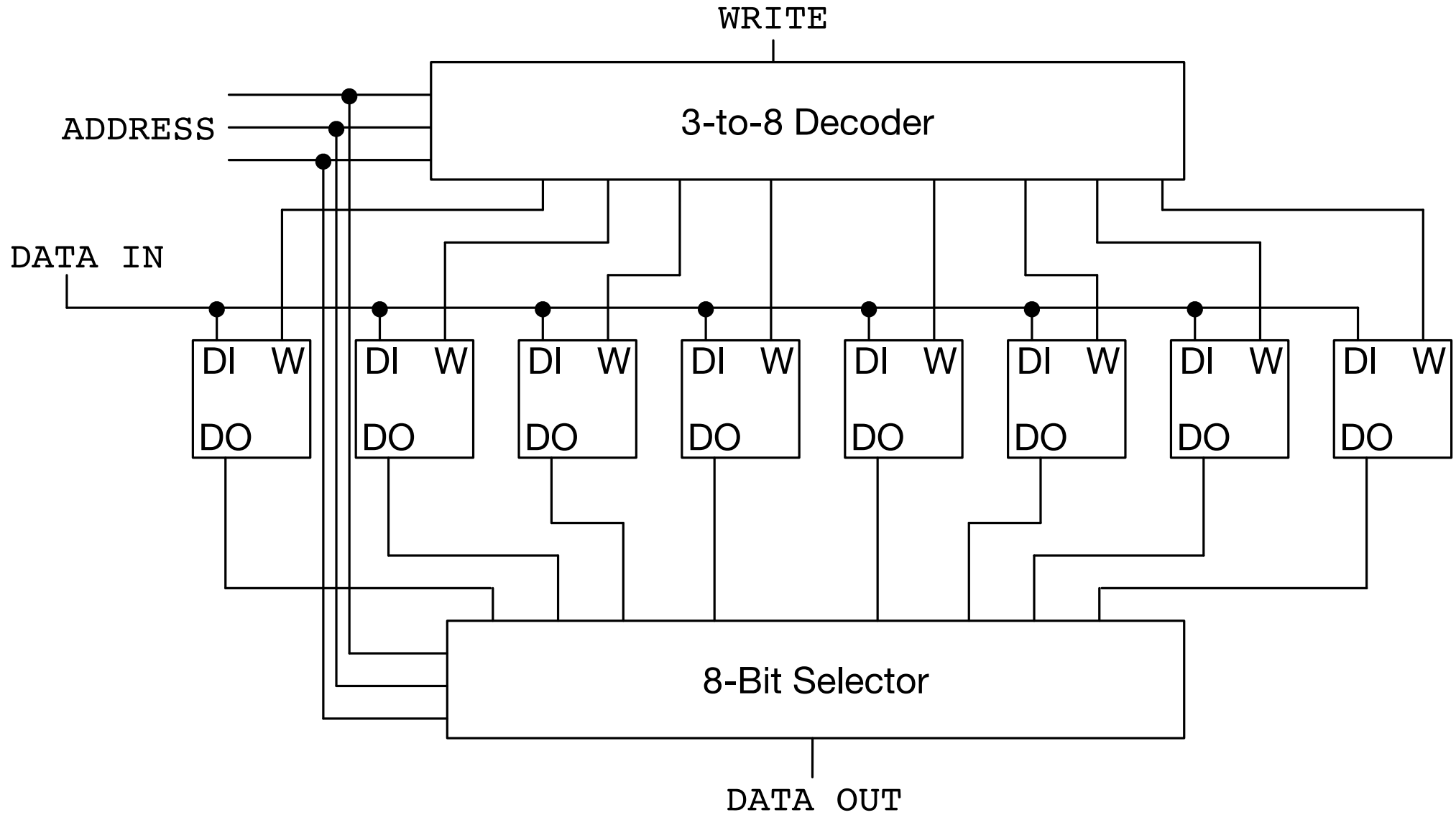
(NOT A2) AND (NOT A1) AND (NOT A0) AND WRITE

# Input Decoder





# 8 Bit RAM



# 8 Bit RAM

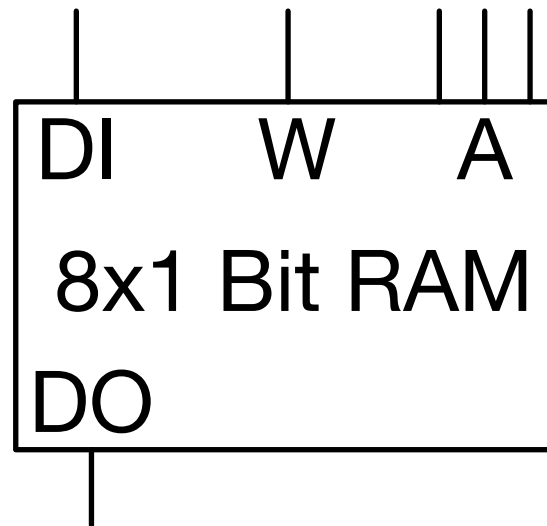
- 8 Bit Random Access Memory (RAM)

- Input

- address
- write flag
- data bit

- Output

- data bit

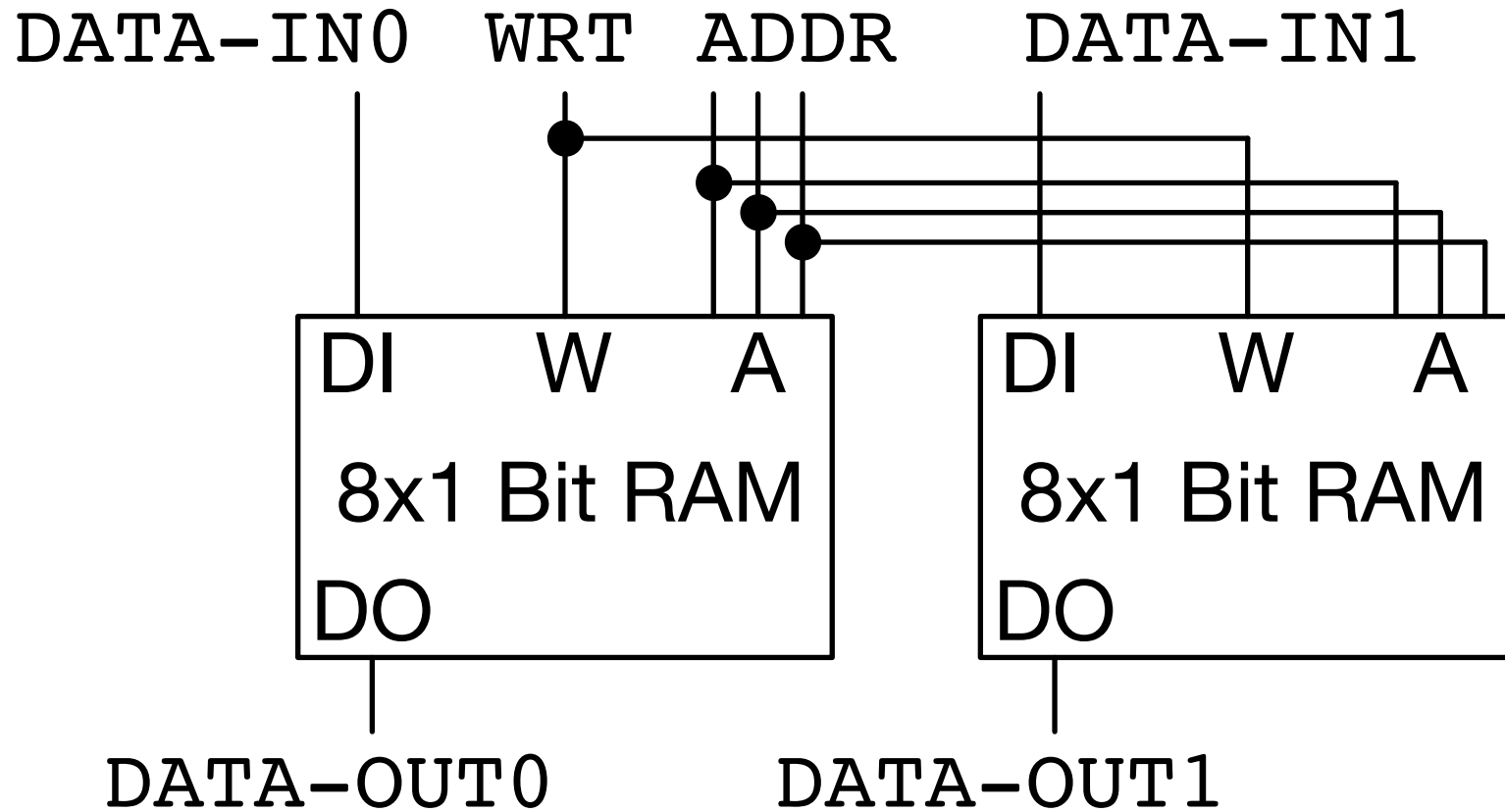


# 8x2 Bit RAM

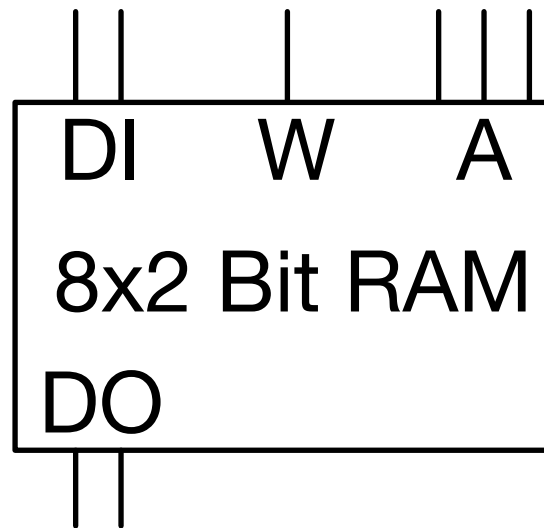
- 8x1 bit RAM allows read/write of 1 bit at a time
- What if we want to read/write 2 bits at a time?  
(and ultimately 8 bits (1 byte) and more)■

⇒ Arrange them together

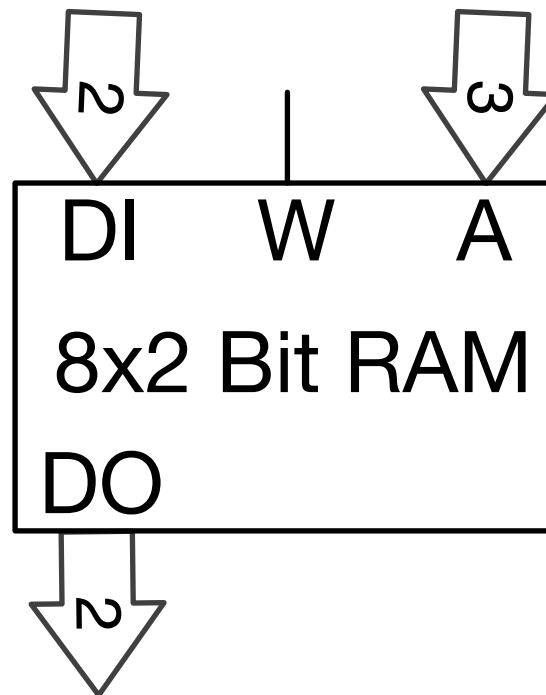
# 8x2 Bit RAM



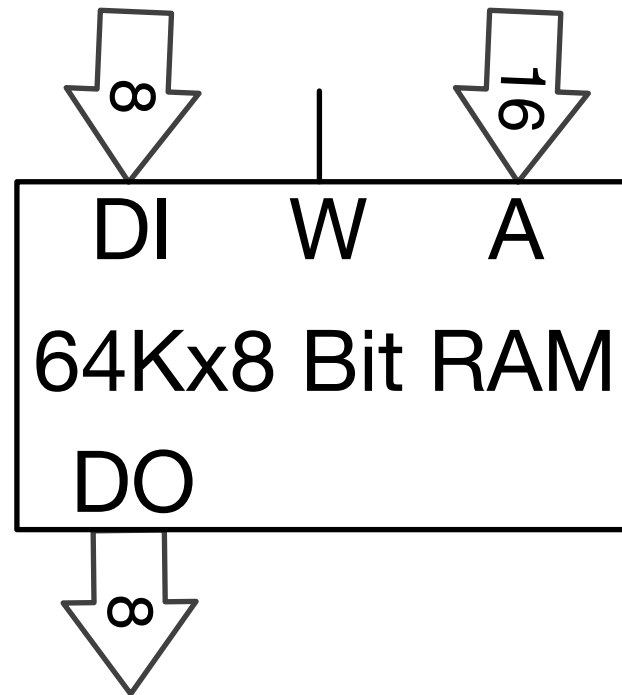
# 8x2 Bit RAM



# 8x2 Bit RAM

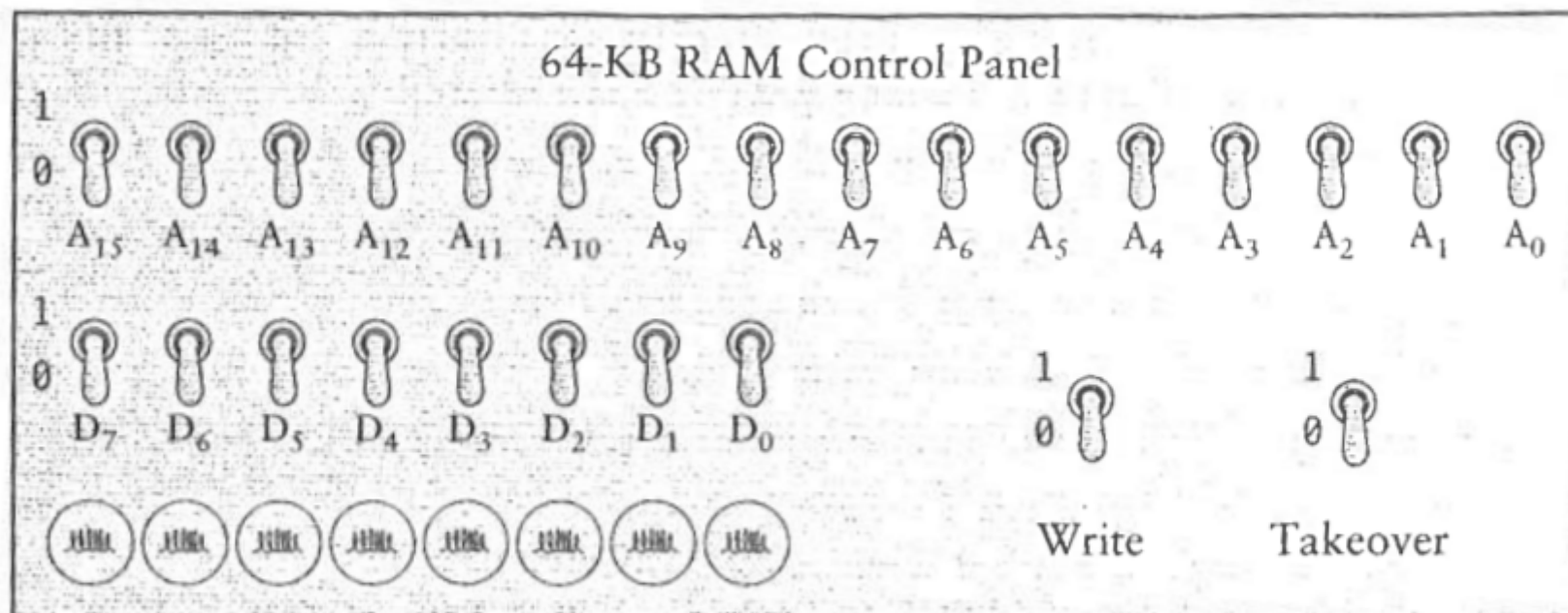


# 64 KB RAM



- 64KB = 65,536 bytes
- 16 bit address space ( $2^{16} = 65536$ )
- Common memory size in the 1980s: we will use it with 6502 assembly

# Control Panel





# Memories

```
**** COMMODORE 64 BASIC V2 ****  
64K RAM SYSTEM 38911 BASIC BYTES FREE  
READY.
```

Early 1980s: 64 KB RAM, 16 bit address space

# Bigger Memories



- Early 1980s: 16 bit address space, up to 64 KB
- 1990s: 32-bit address space, up to 4 GB
- Today: 64-bit address space, up to 16 EB (exa-byte)
- Actually supported by Intel/AMD 64-bit processors
  - 52 bits for physical memory: 4 peta-byte
  - 48 bits for virtual memory: 256 tera-byte
- Actually existing RAM: my lab biggest RAM machine: 768 GB (doubles every  $\sim 2$  years)