Memory

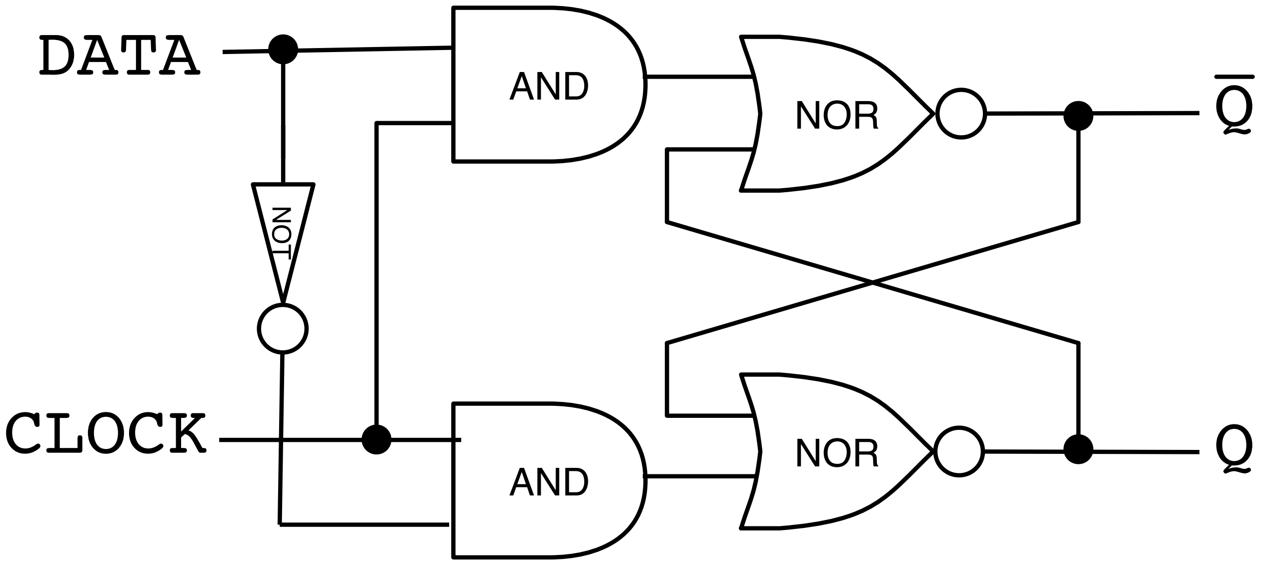
Philipp Koehn

9 September 2019

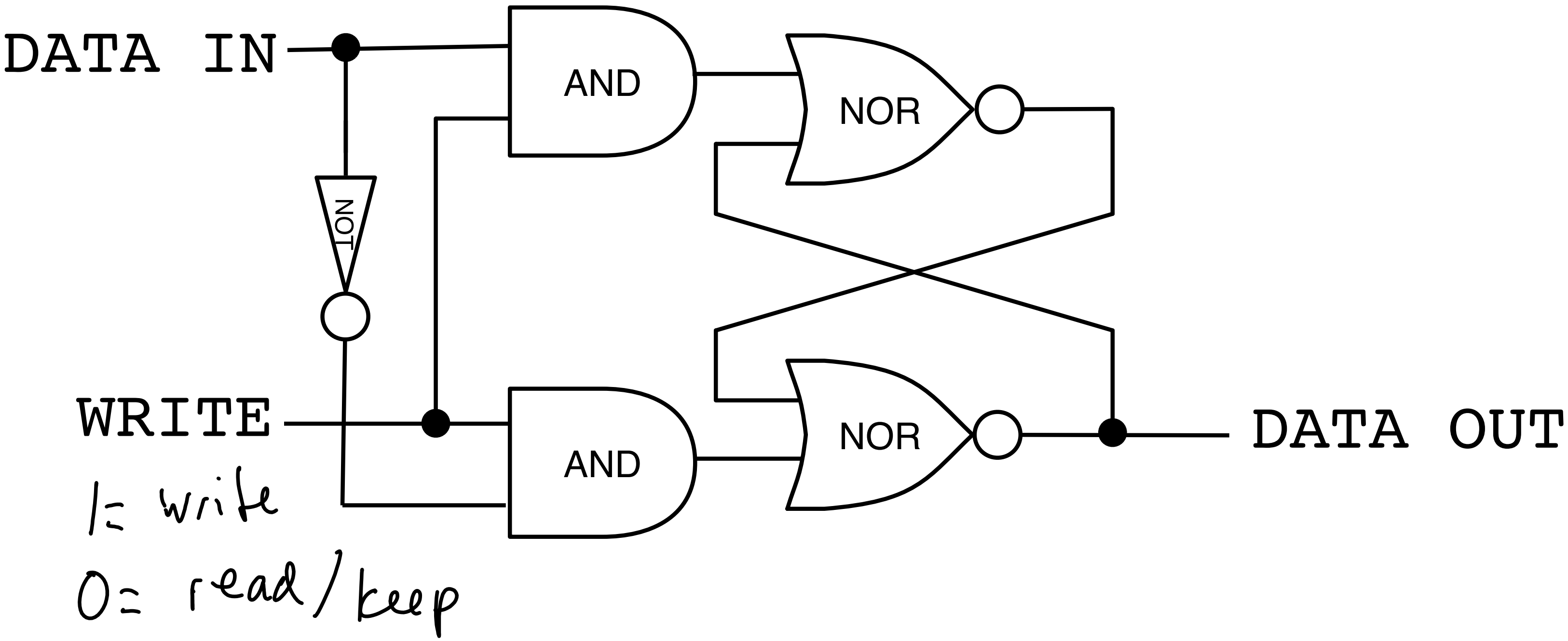
HW1 - due Friday 9/13

HW2 - soon

D-Type Level-Triggered Latch



Slightly Modified



Operations

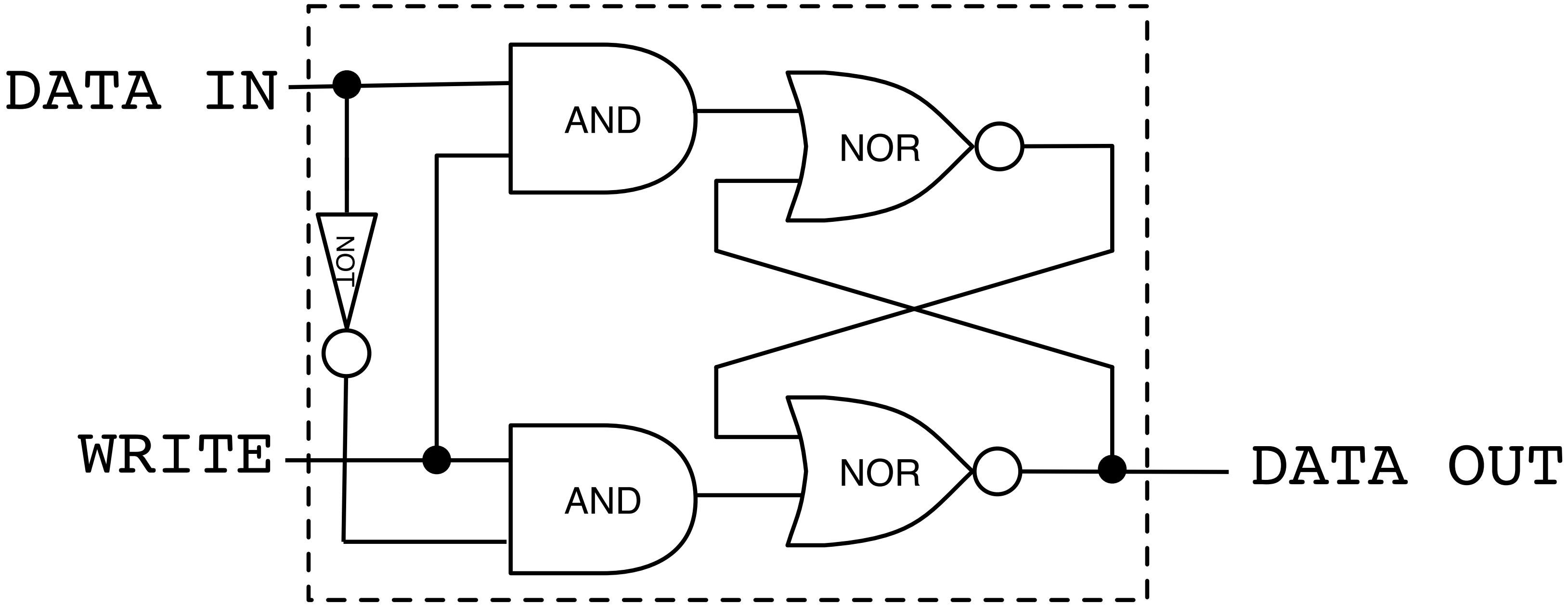
- Circuit latches on one bit of memory and keeps it around
- Truth table

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

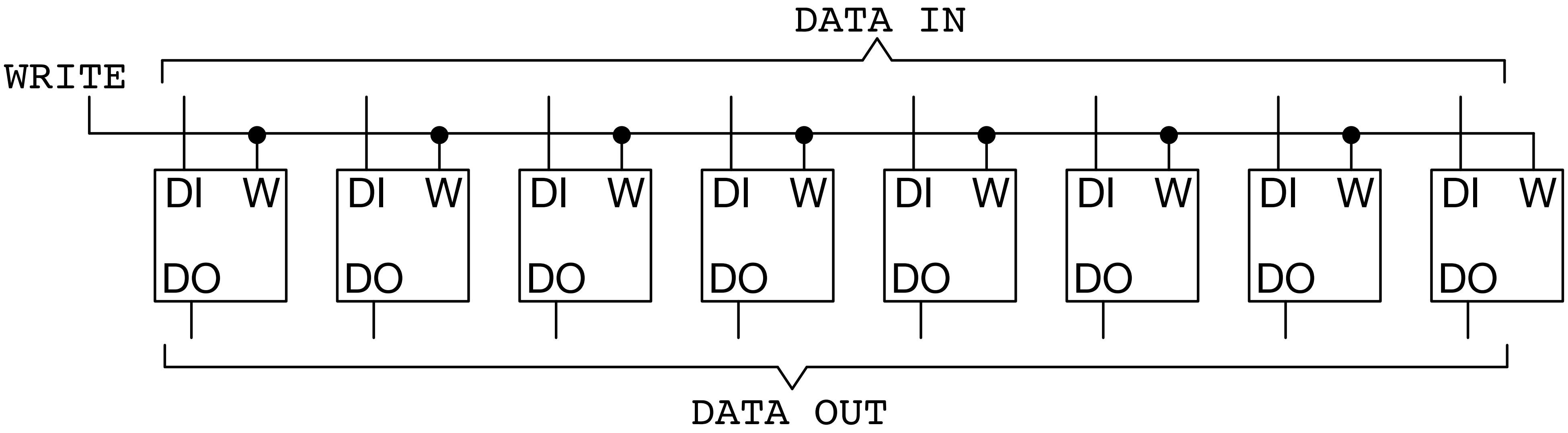
- Can write 1 bit and read content

multi-bit storage

1 Bit Memory



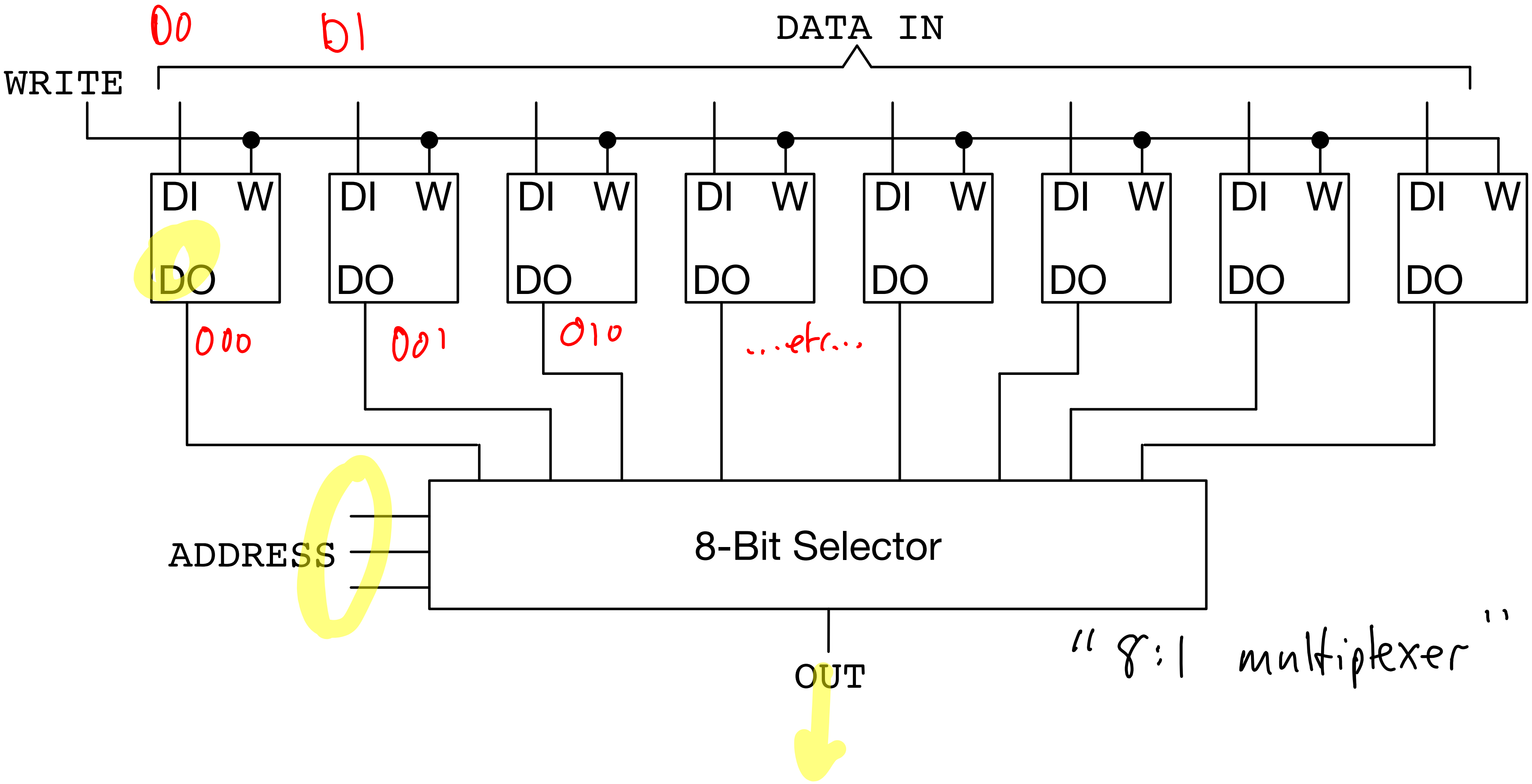
8 Bit Memory



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



"8:1 multiplexer"

Output Selector

- Truth table

Address			Output
A2	A1	A0	OUT
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Output Selector

- Truth table

Address			Output
A2	A1	A0	OUT
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

- What Boolean operation returns the correct value for address 000?

Output Selector

- Truth table

Address			Output
A2	A1	A0	OUT
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

- What Boolean operation returns the correct value for address 000?

(NOT A2) AND (NOT A1) AND (NOT A0) AND D0

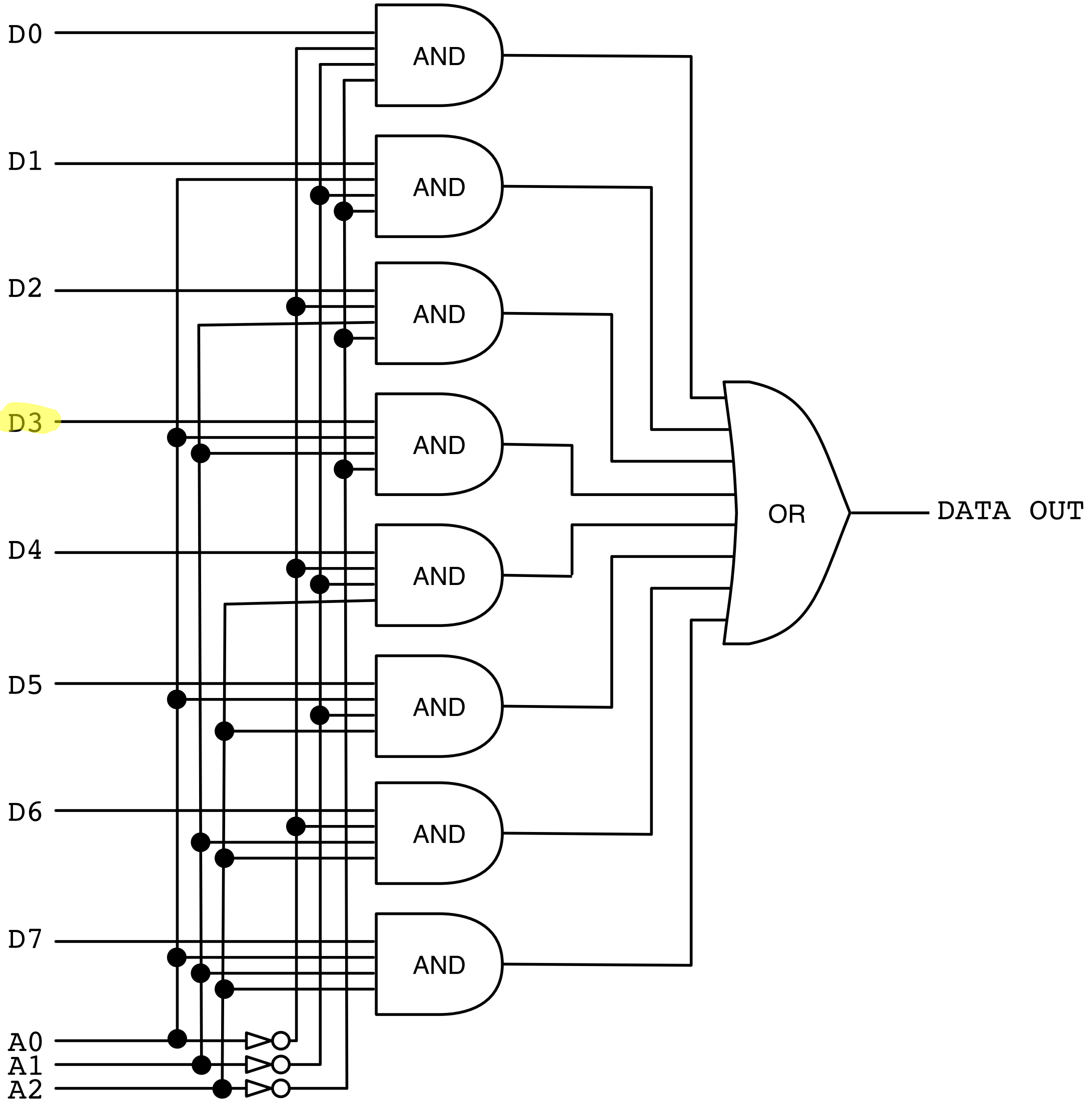
Output Selector

Addr = 011

- 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 )
```

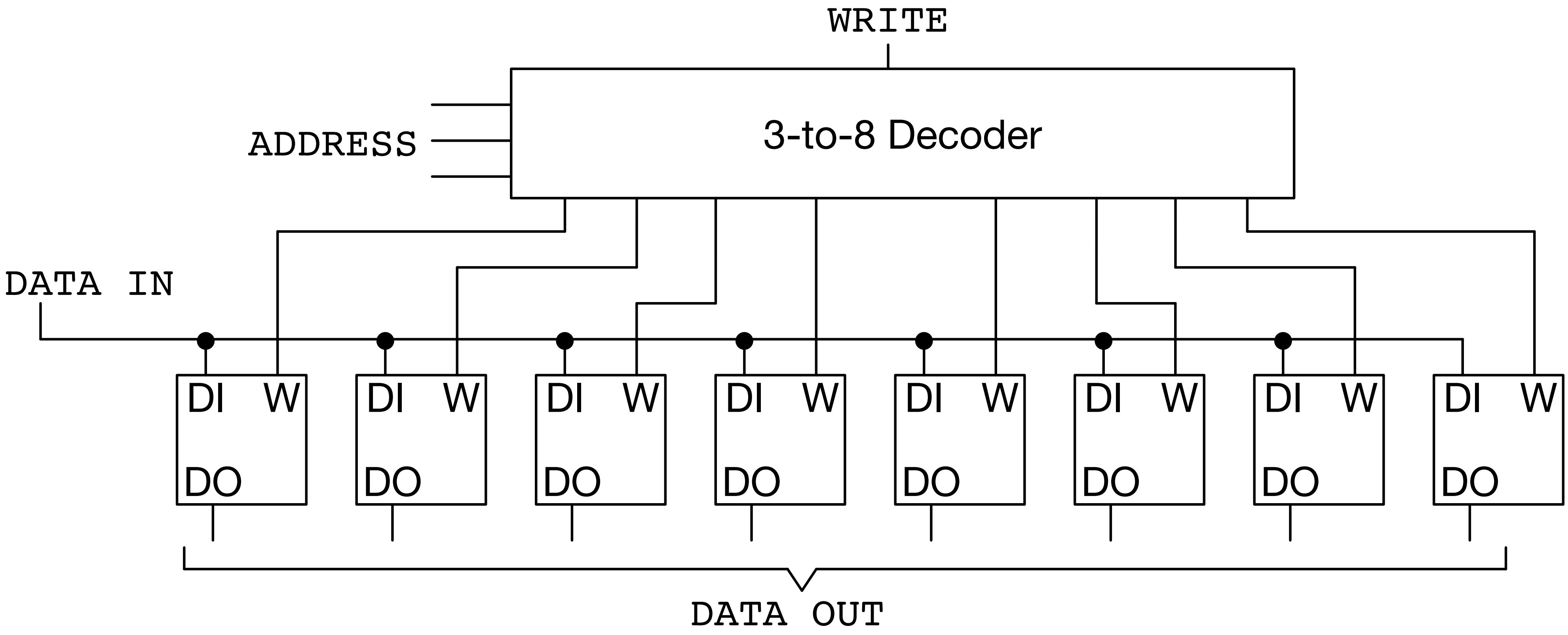
Output Selector



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?

Input Decoder

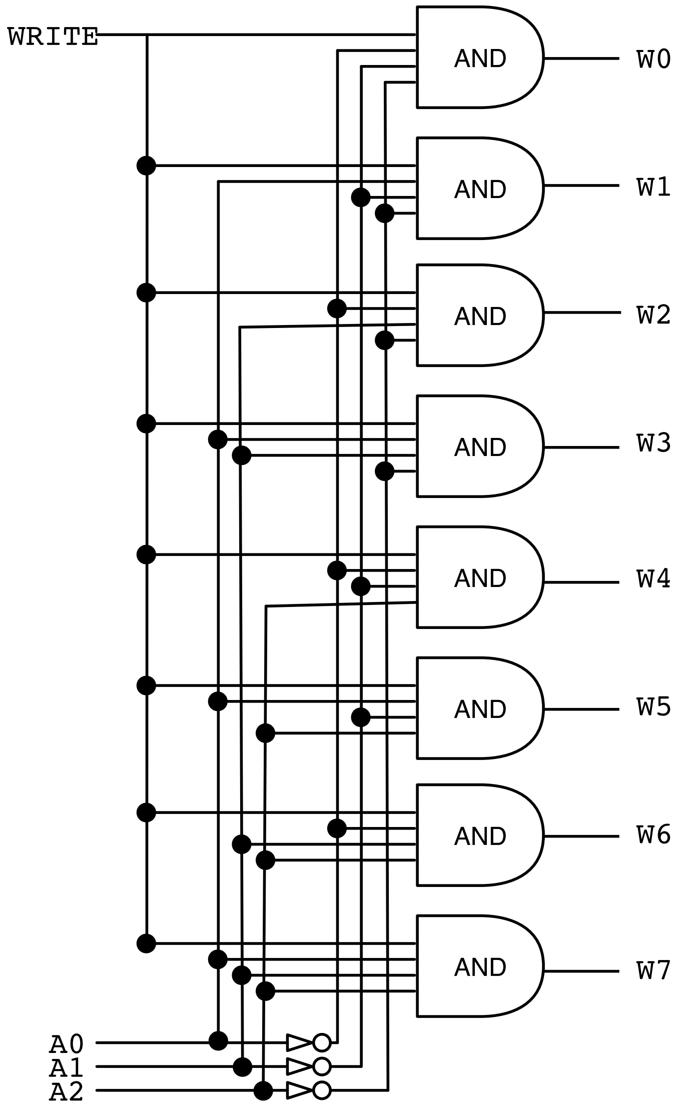
- Truth table

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

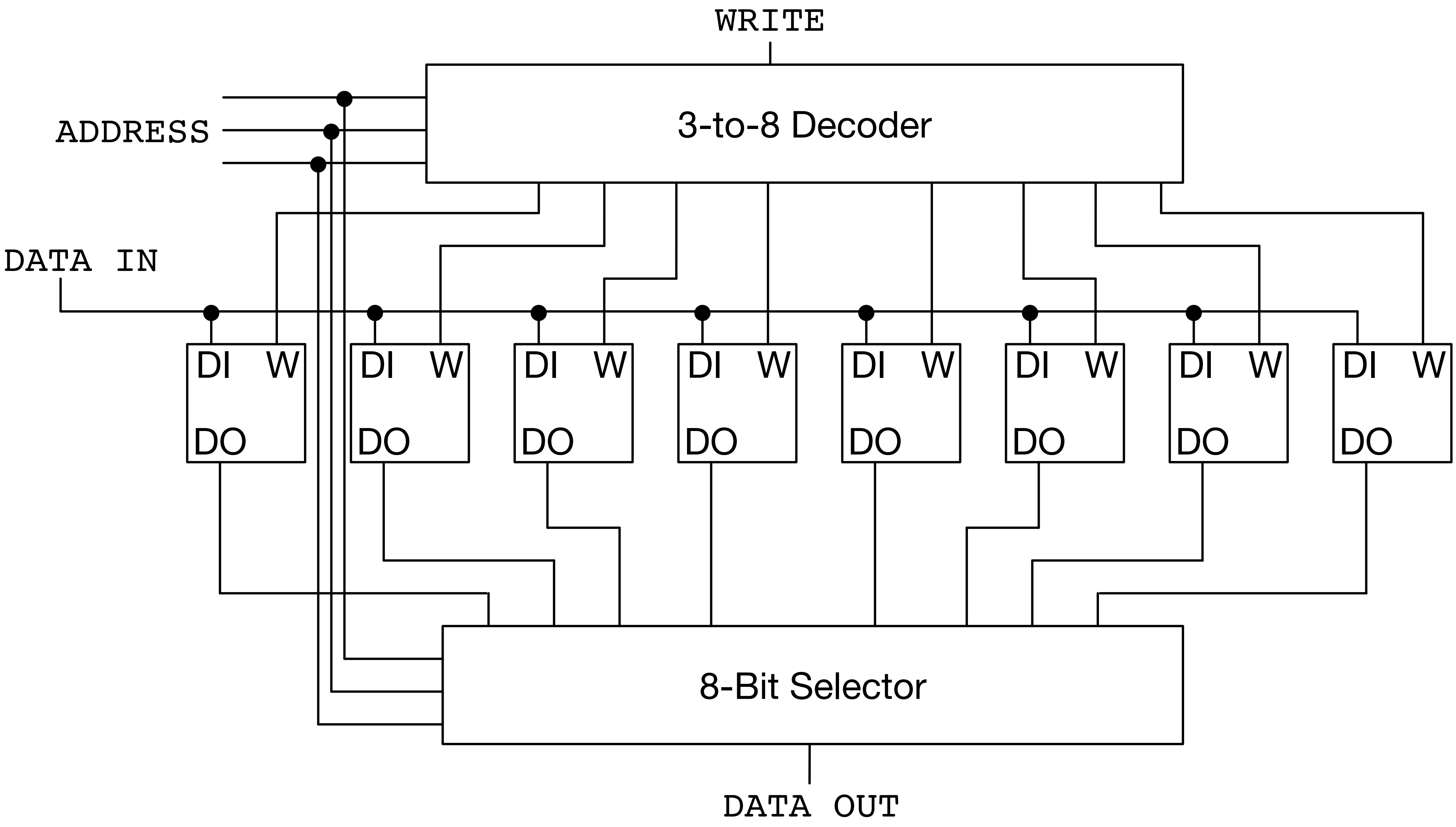
- What Boolean operation returns the correct value for output W0?

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

Input Decoder

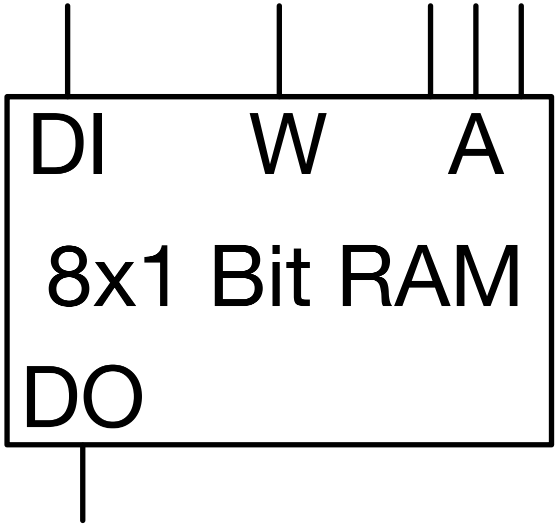


8 Bit RAM


WRITE
ADDRESS
3-to-8 Decoder
DATA IN
DI W
DO
DI W
DO
DI W
DO
DI W
DO
DI W
DO
DI W
DO
DI W
DO
DI W
DO
8-Bit Selector
DATA OUT

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)

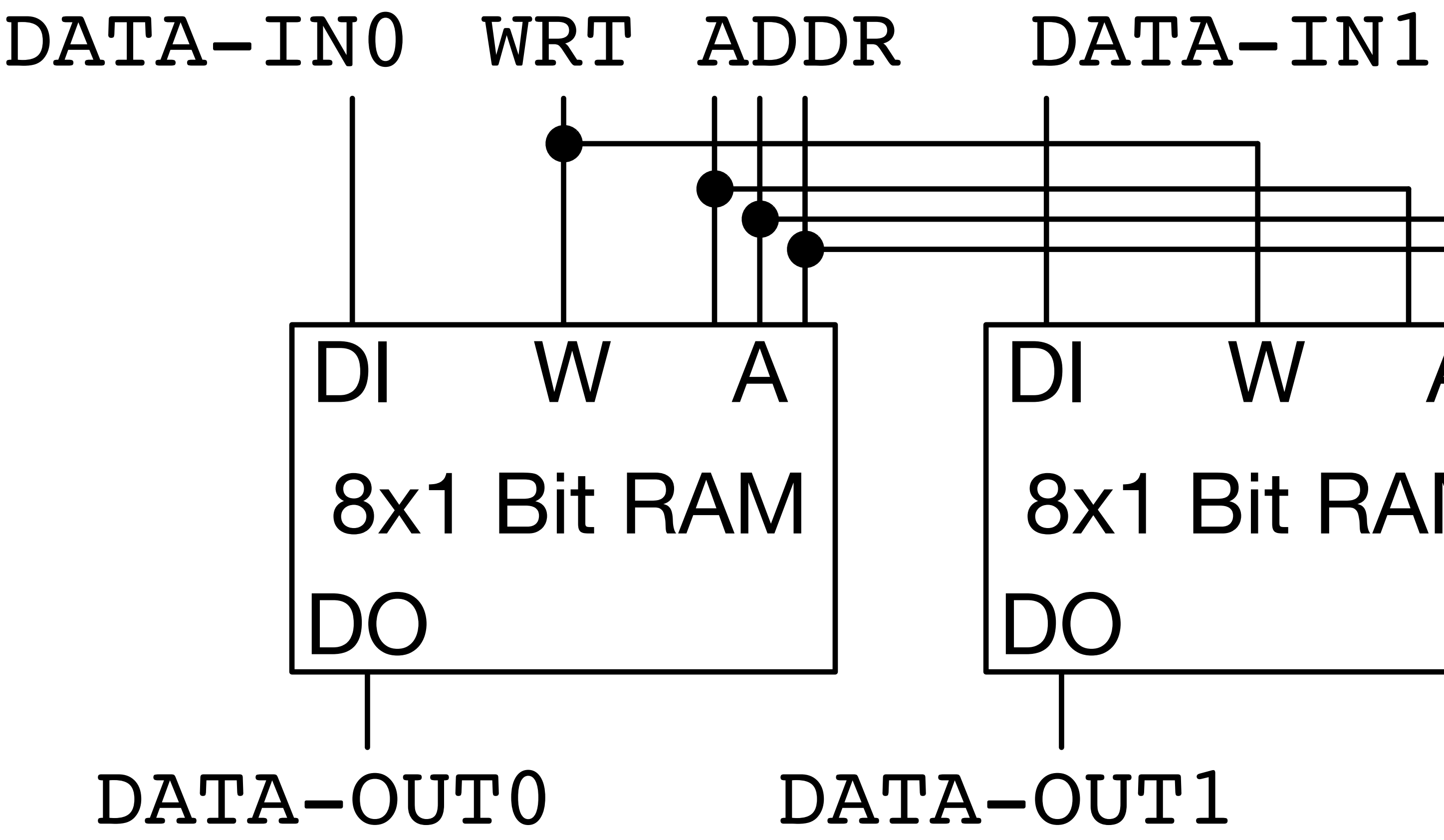
8x2 Bit RAM

- 8x1 bit RAM allows read/write of 1 bit at a time
- What if we want to read/write 2 bits at a time?

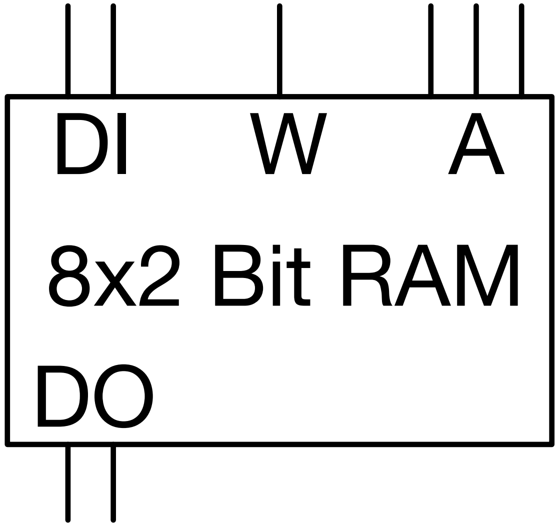
  (and ultimately 8 bits (1 byte) and more)

⇒ Arrange them together

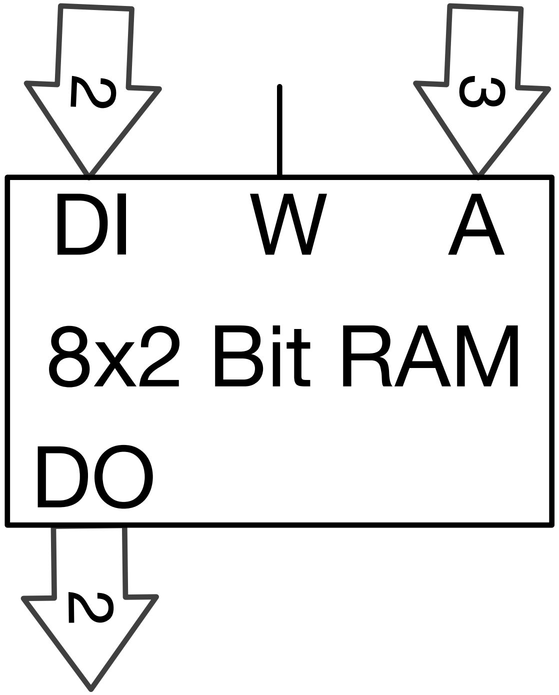
8x2 Bit RAM


DATA–IN0
WRT
ADDR
DATA–IN1
DI
W
A
8x1 Bit RAM
DO
DI
W
A
8x1 Bit RAM
DO
DATA–OUT0
DATA–OUT1

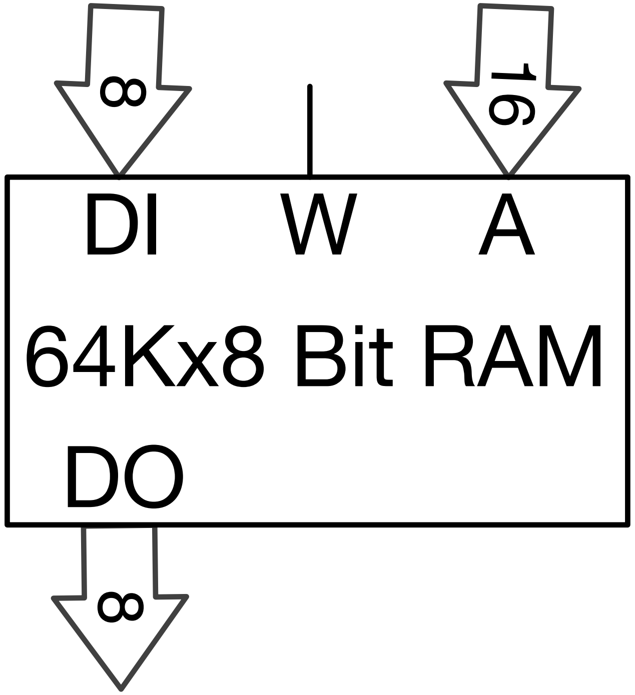
8x2 Bit RAM



8x2 Bit RAM



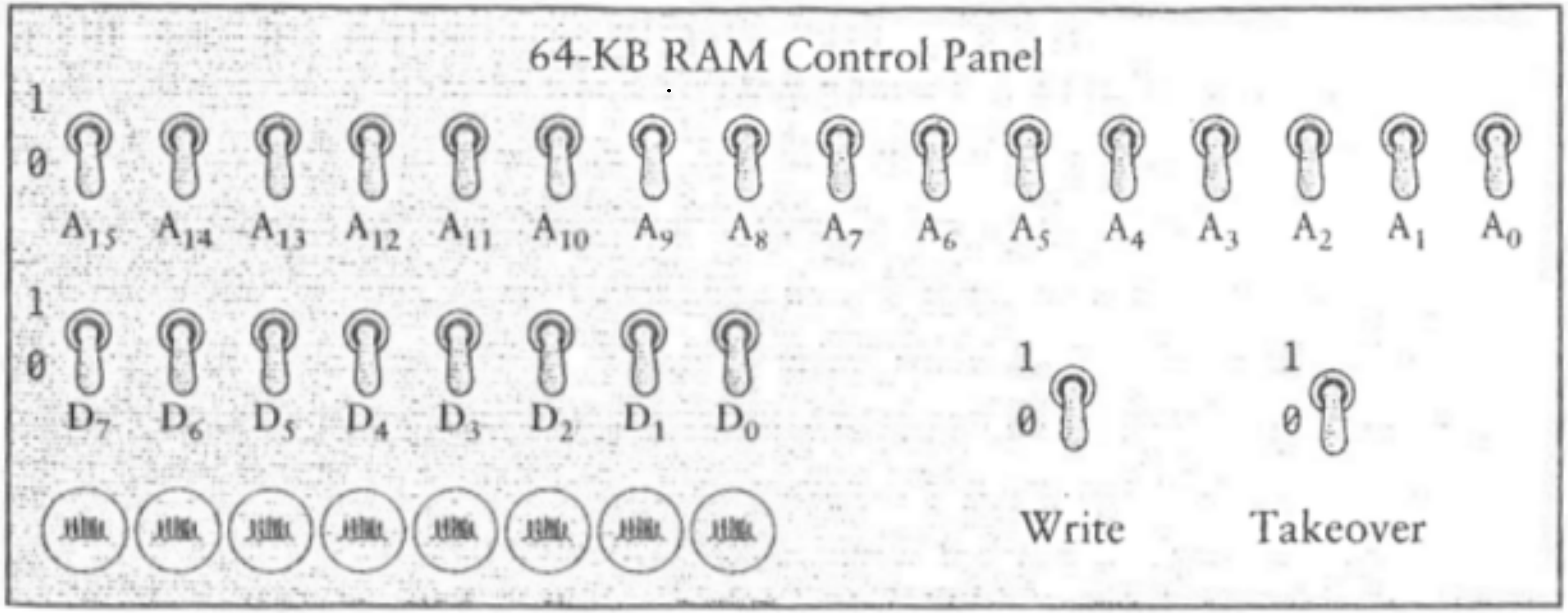
64 KB RAM



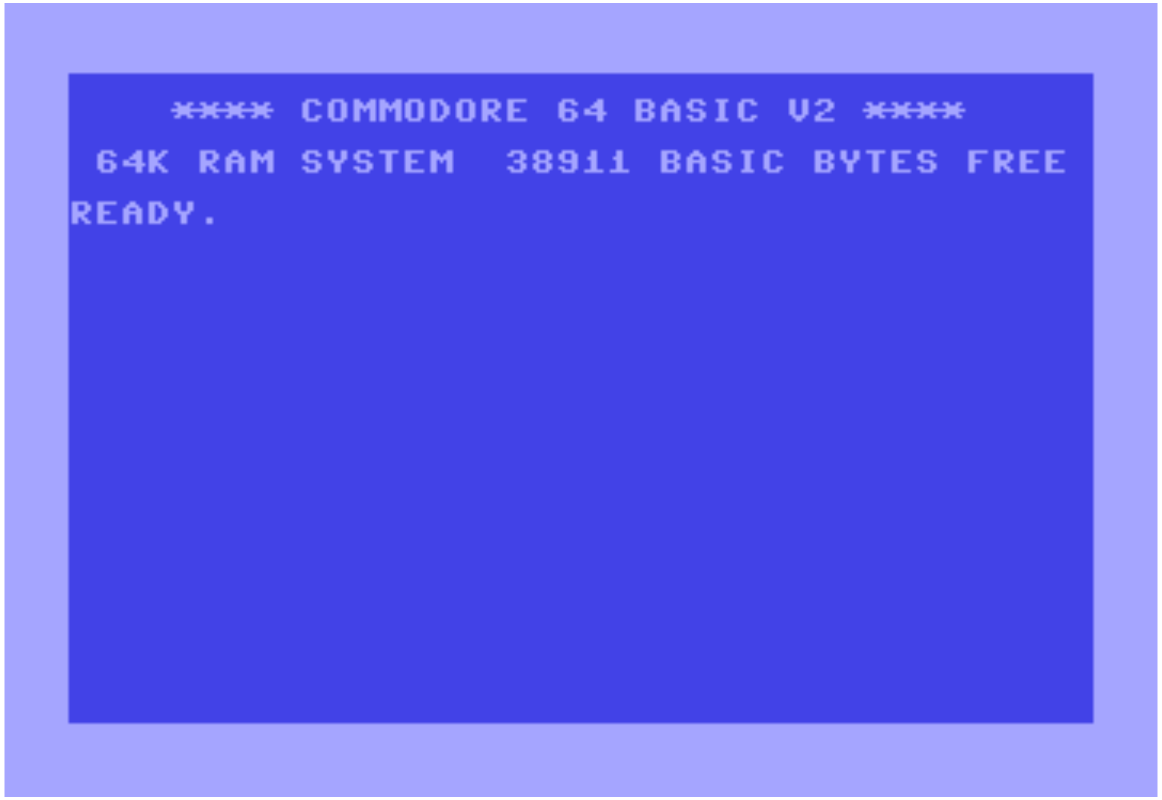
- 64KB = 65,536 bytes
- 16 bit address space ($2^{16} = 65536$)
- Common memory size in the 1980s: we will use it with 6502 assembly

Control Panel

Altair 8800



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

Bigger Memories

- Early 1980s: 16 bit address space, up to 64 KB

- 1990s: 32-bit address space, up to 4 GB

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)

Bigger Memories

- Early 1980s: 16 bit address space, up to 64 KB
- 1990s: 32-bit address space, up to 4 GB $= 2^{32}$
- Today: 64-bit address space, up to 16 EB (exa-byte) $= 2^{64}$
- Actually supported by Intel/AMD 64-bit processors
  - 52 bits for physical memory: 4 peta-byte
  - 48 bits for virtual memory: 256 tera-byte

→ flexible mapping of addresses to physical memory

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 ∼2 years)