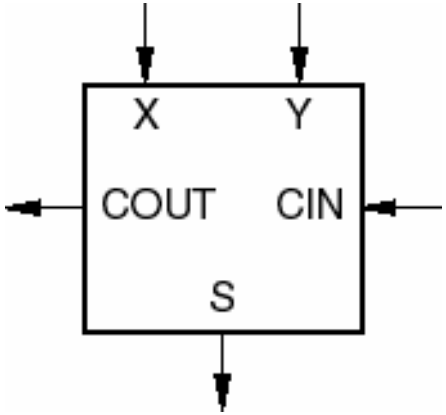
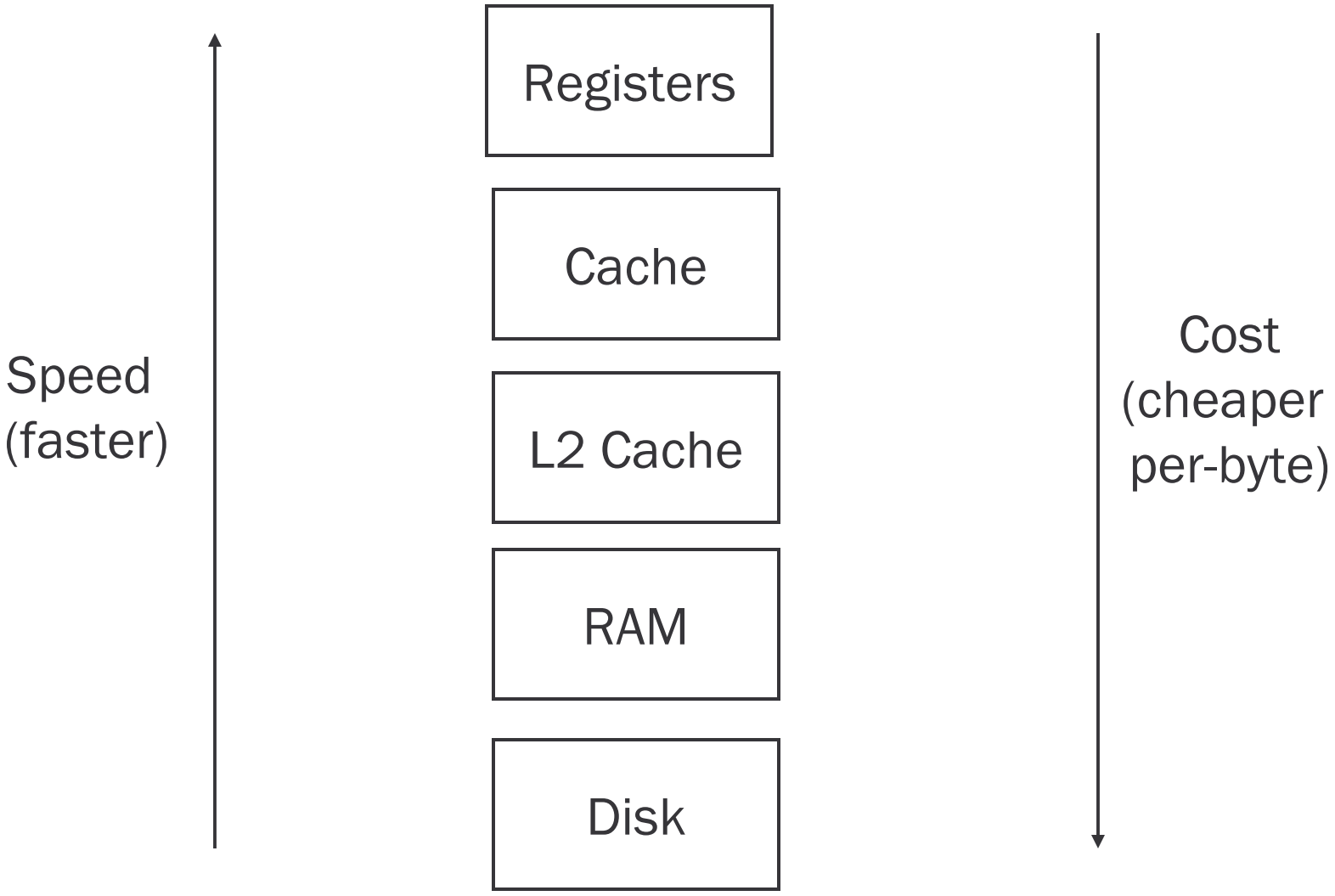
More Digital Design

ECGR2181

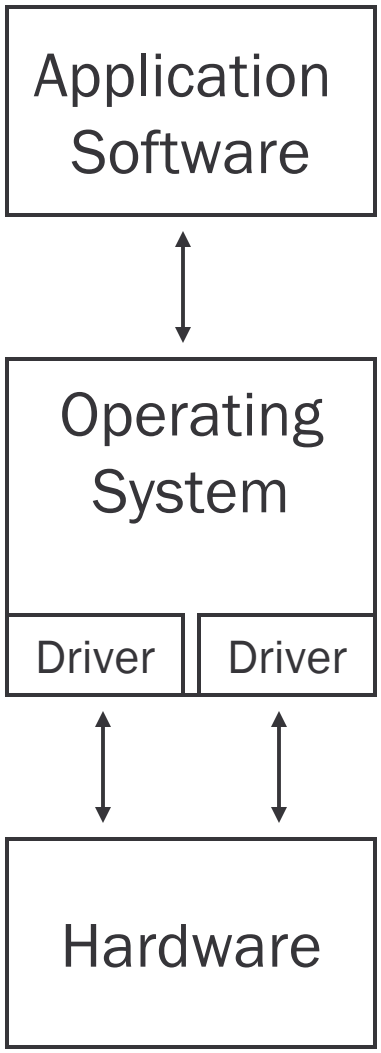


Reading: Chapter 9.1

Memory Hierarchy



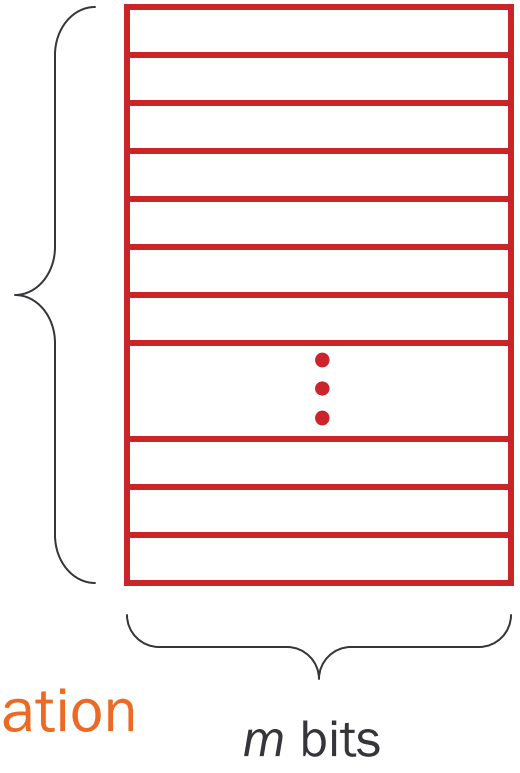
View of Computer System



Memory

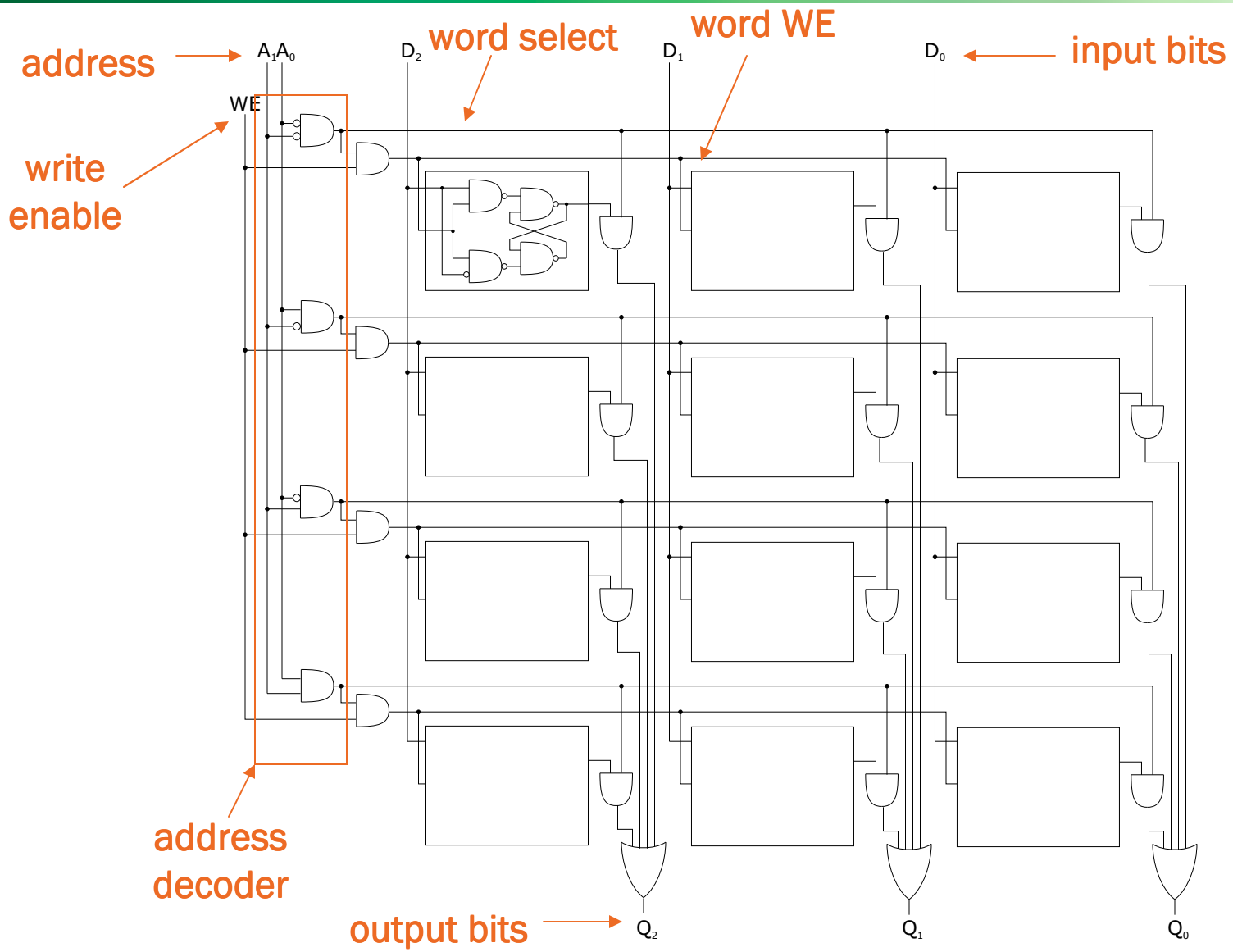
To build a memory -- a logical $k \times m$ array of stored bits.

Address Space:
number of locations
(usually a power of 2)



Addressability:
number of bits per location
(e.g., byte-addressable)

2^2 x 3 Memory


address
A1A0
D2
word select
D1
word WE
D0
input bits
WE
write
enable
address
decoder
output bits
Q2
Q1
Q0

More Memory Details

This is a not the way actual memory is implemented.

- fewer transistors, much more dense, relies on electrical properties

But the logical structure is very similar.

- address decoder
- word select line
- word write enable

Two basic kinds of memory (RAM = Random Access Memory)

Static RAM (SRAM)

- fast, maintains data without power refresh

Dynamic RAM (DRAM)

- slower but denser, bit storage must be periodically refreshed

Even More Memory Details

There are other types of “non-volatile” memory devices:

- ROM
- PROM
- EPROM
- EEPROM
- Flash

Can you think of other memory devices?

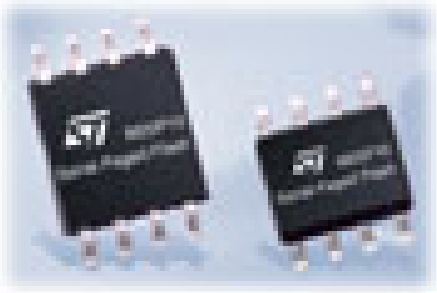
Electronics Packaging

- There are several packaging technologies available that an engineer can use to create electronic devices.
- Some are suitable for inexpensive toys but not miniature consumer products, and some are suitable for miniature consumer products but not inexpensive toys.
- These packages have metal leads that are the conductive wire that connect electricity from the outside world to the silicon inside the package.
- Leads between packages are connected with small copper traces on a printed circuit board (PCB), and the package leads are soldered to the PCB.

Examples of Electronics Packages

Dual In-line Package (DIP) Older technology, requires the metal leads to go through a hole in the printed circuit board.

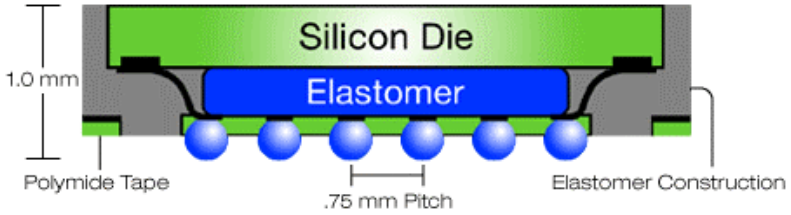
Dual Flat Pack (DFP) - A fairly recent technology, metal leads solder to the surface of the printed circuit board.



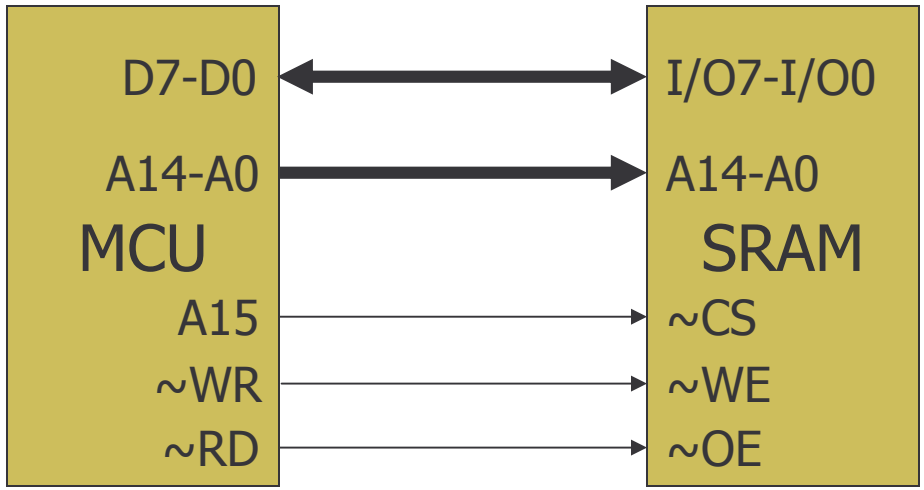
Examples of Electronics Packages

Quad Flat Pack (QFP) - like the Dual Flat Pack, except here are metal leads are on four sides.

Ball Grid Array (BGA) - The connections to the component are on the bottom of the chip, and have balls of solder on these connections.



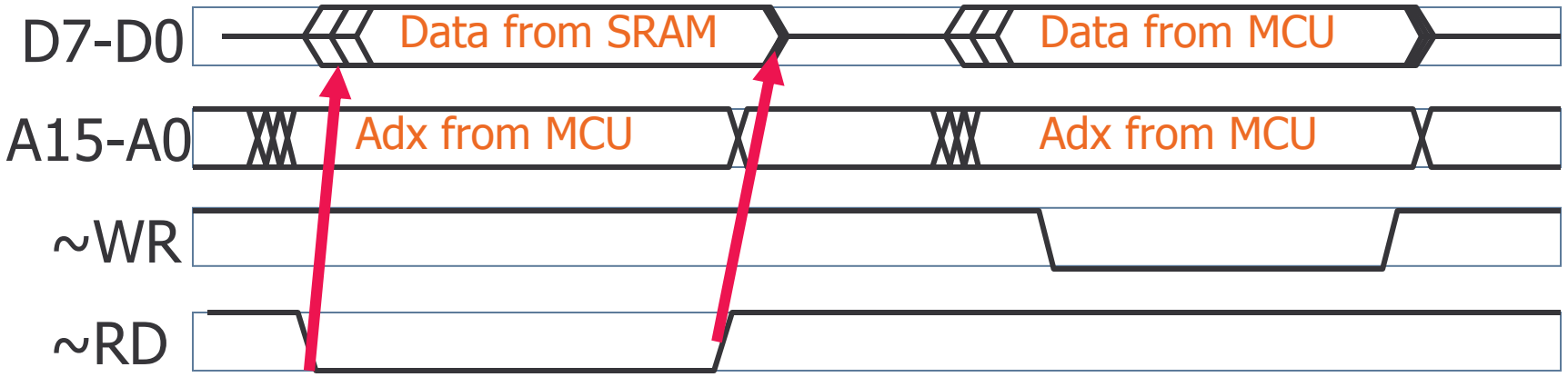
External Memory Access – Separate Buses



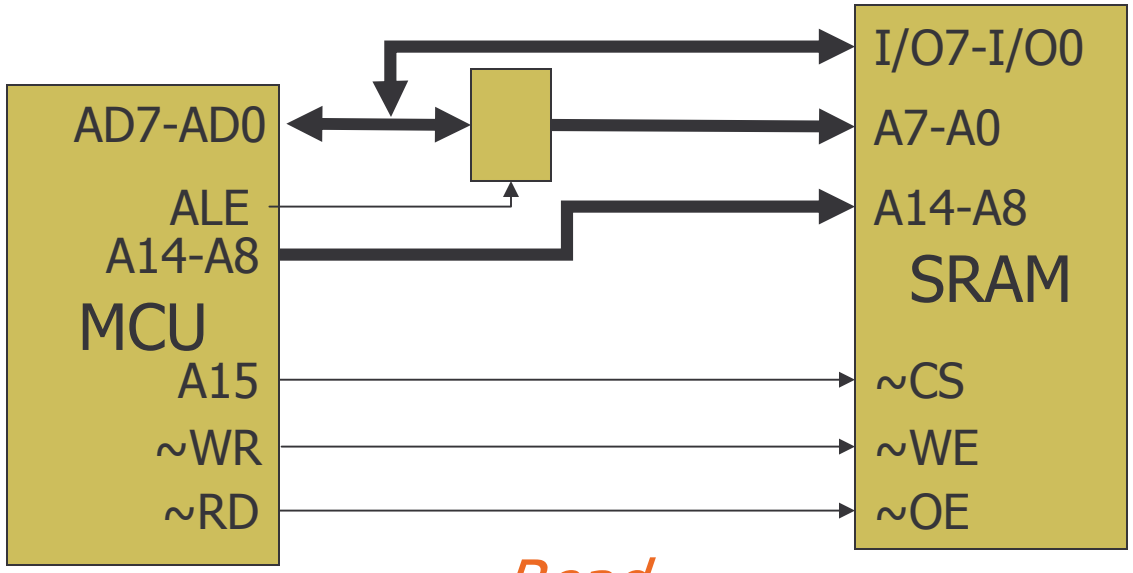
Chip Select

- Partial vs. Full Decoding
- Power Consumption

Read *Write*

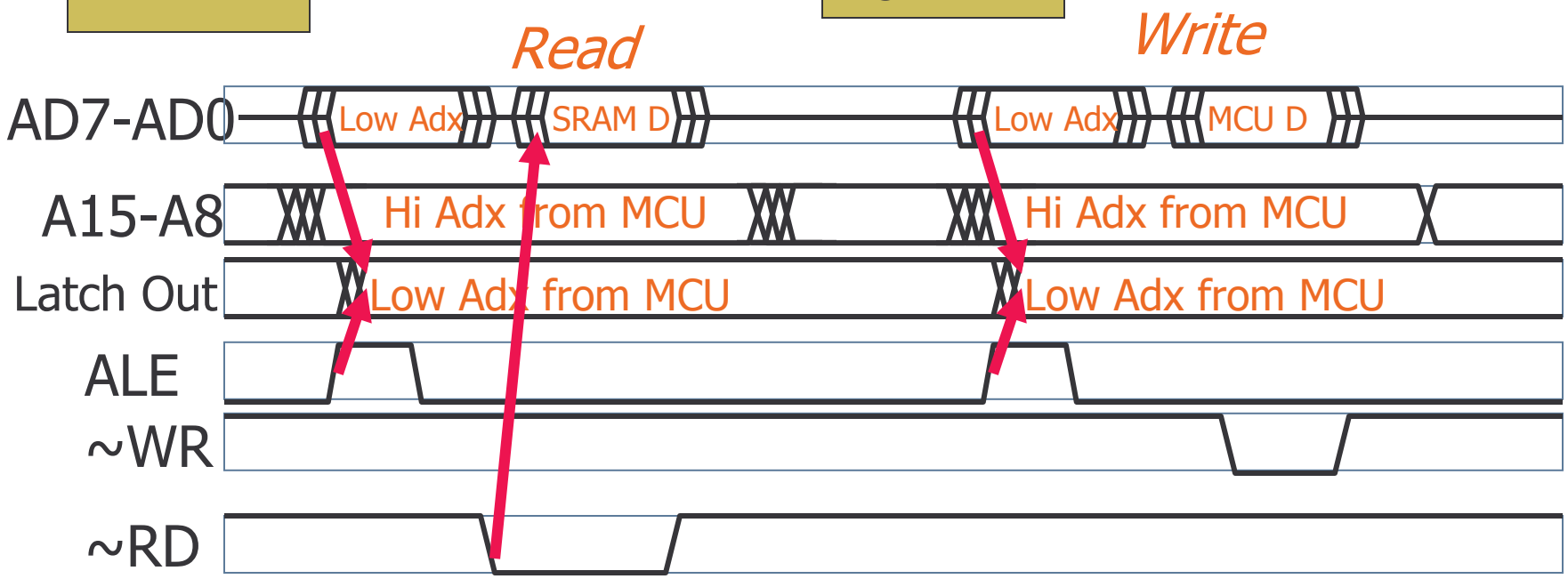


External Memory Access – Multiplexed Buses

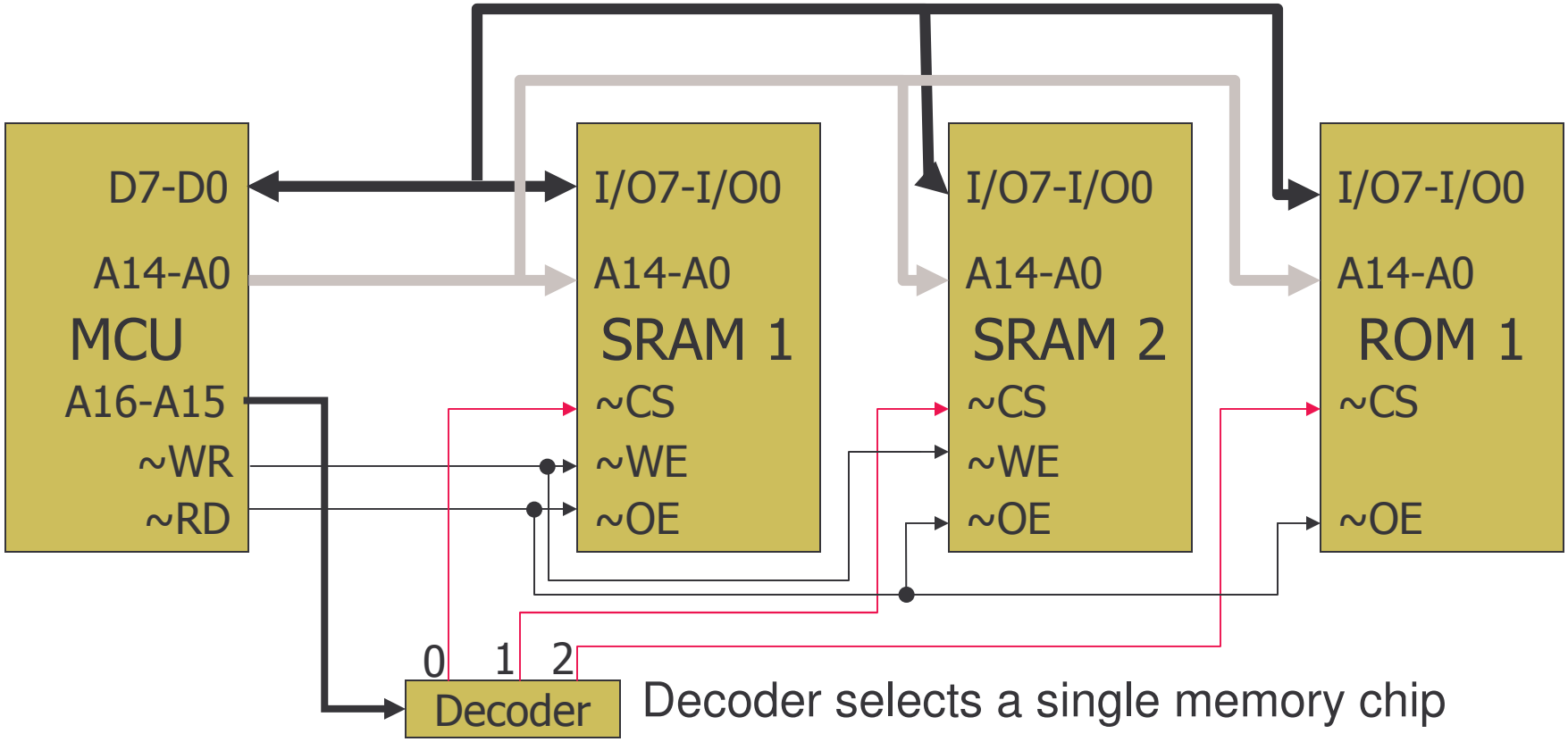


Use a latch to hold the low byte of the address

Saves pins



Multiple External Memories

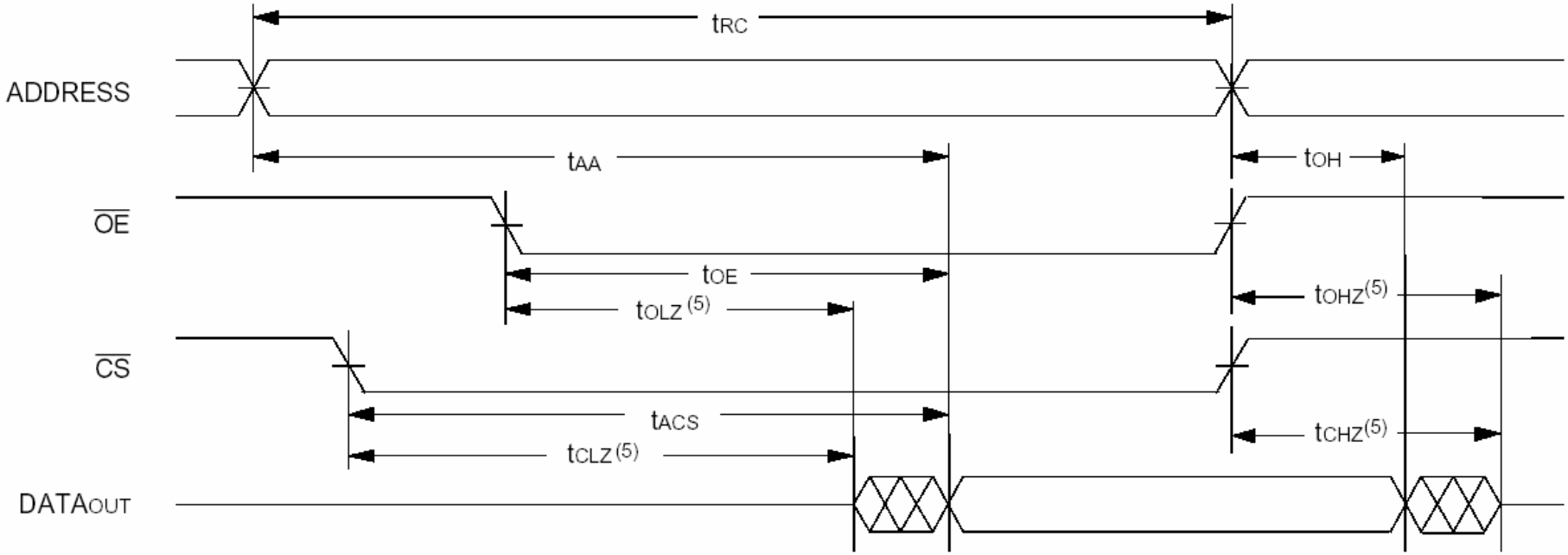


Decoder selects a single memory chip

- Output 0 active when A16:A15 = 00. Address = 0 0xxx xxxx xxxx xxxx = 00000h to 07FFFh
- Output 1 active when A16:A15 = 01. Address = 0 1xxx xxxx xxxx xxxx = 08000h to 0FFFFh
- Output 2 active when A16:A15 = 10. Address = 1 0xxx xxxx xxxx xxxx = 10000h to 17FFFh

IDT71256L Read Timing

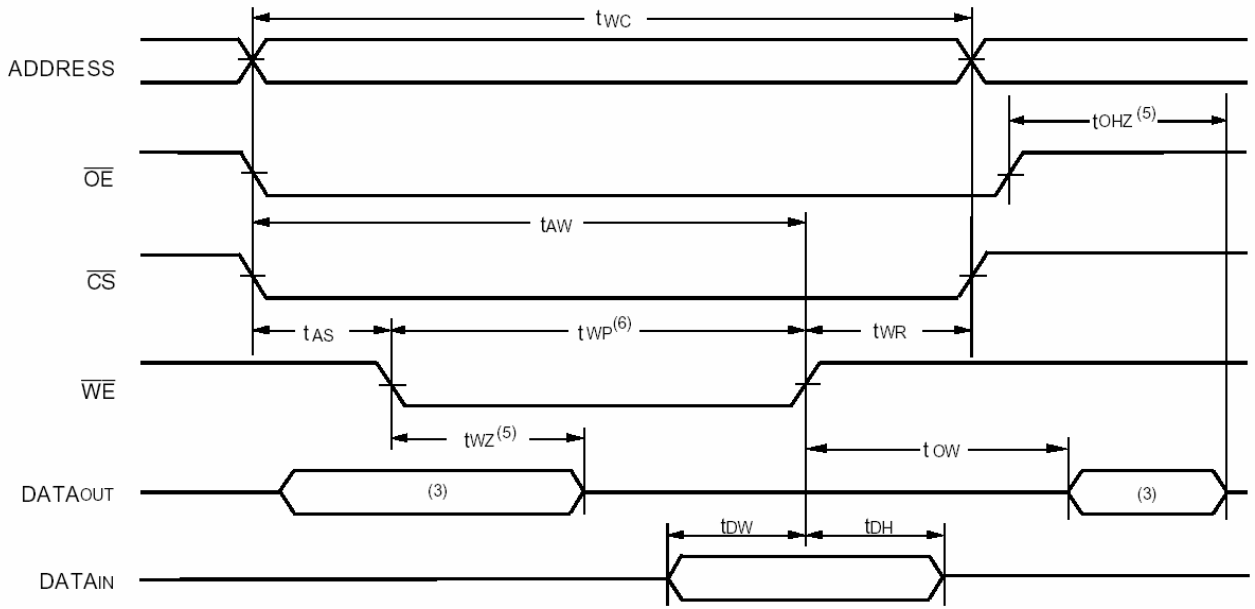
Timing Waveform of Read Cycle No. 1(1)



Symbol	Parameter	Minimum (ns)	Maximum (ns)
t_{RC}	Read Cycle Time	20	-
t_{AA}	Address Access time	-	20
t_{OLZ}	Output Enable to Output in Low-Z	2	-
t_{OE}	Output Enable to Output Valid	-	10
t_{OHZ}	Output Disable to Output in Hi-Z	2	8

IDT71256L Write Timing

Timing Waveform of Write Cycle No. 1 ($\overline{WE}$ Controlled Timing)(1,2,4,6)



Symbol	Parameter	Minimum (ns)	Maximum (ns)
t_{wc}	Write Cycle Time	20	-
t_{AW}	Address Valid to End-of-Write	15	-
t_{WP}	Write Pulse Width	15	-
t_{AS}	Address Set-Up Time	0	-
t_{DW}	Data to Write Time Overlap	11	-
t_{DH}	Data Hold from Write Time	0	-

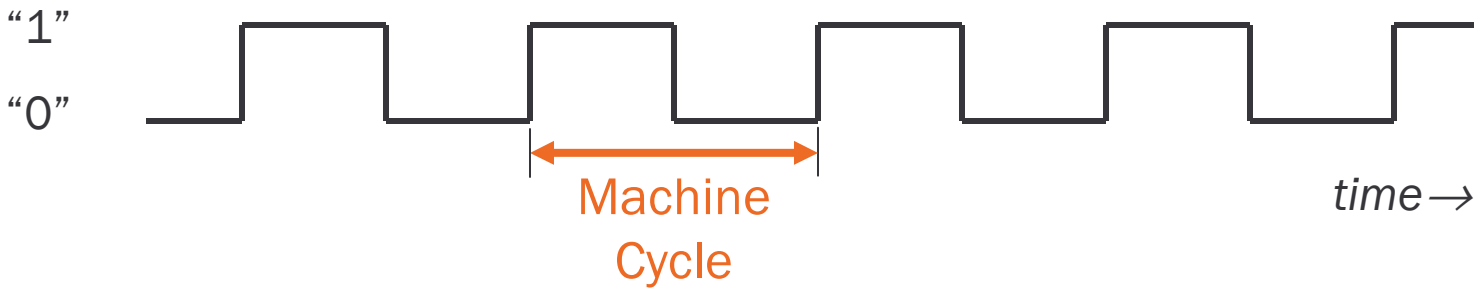
Driving Force: The Clock

The clock is a signal that keeps the control unit moving.

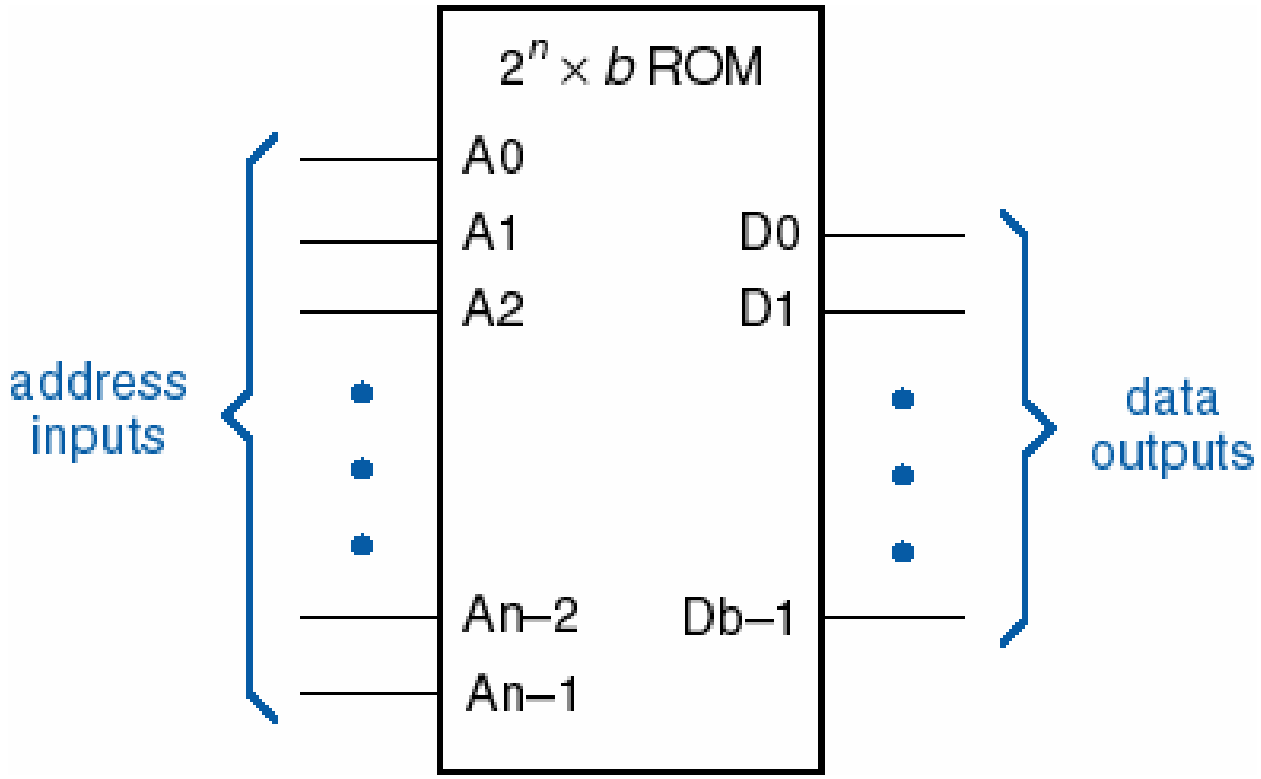
- At each clock "tick," control unit moves to the next machine cycle -- may be next instruction or next phase of current instruction.

Clock generator circuit:

- Based on crystal oscillator
- Generates regular sequence of "0" and "1" logic levels
- Clock cycle (or machine cycle) -- rising edge to rising edge



Read-Only Memories



Why “ROM”?

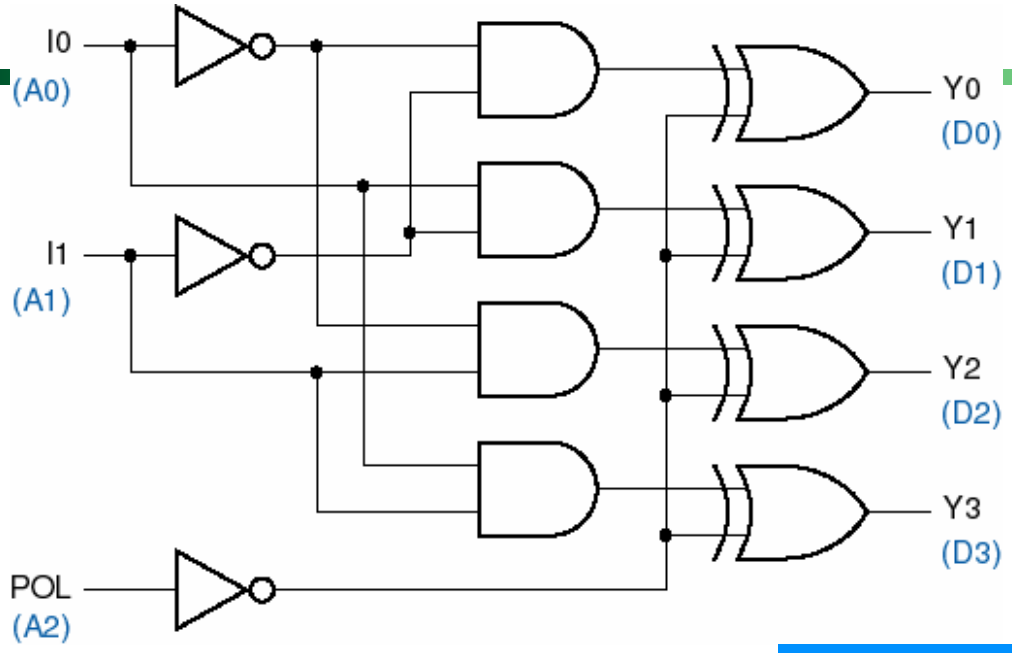
Program storage

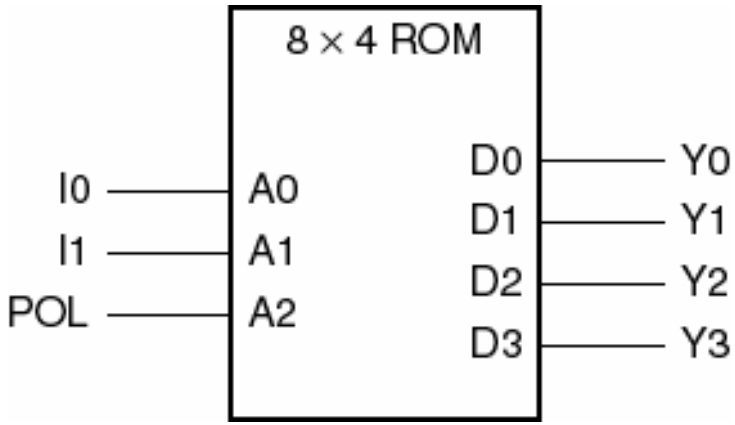
- Boot ROM for personal computers
- Complete application storage for embedded systems.

Actually, a ROM is a combinational circuit, basically a truth-table lookup.

- Can perform any combinational logic function
- Address inputs = function inputs
- Data outputs = function outputs

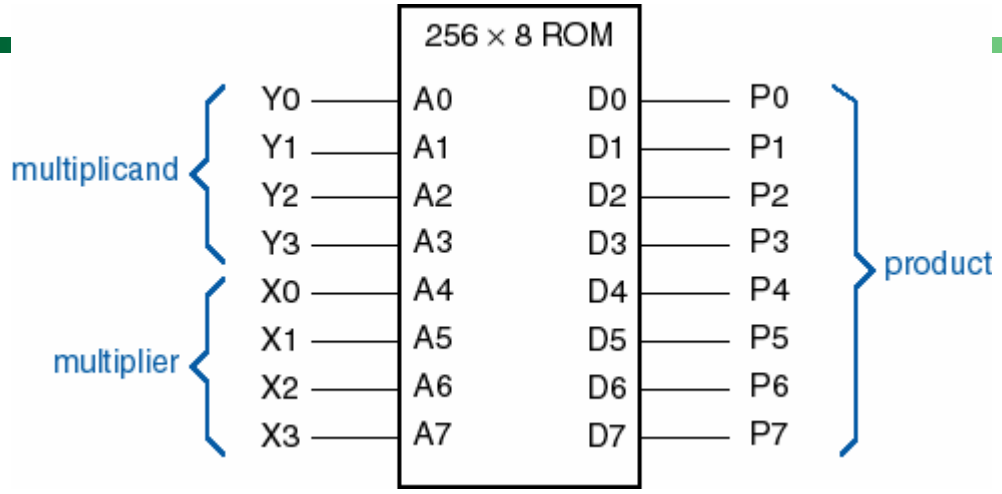
Logic-in-ROM example





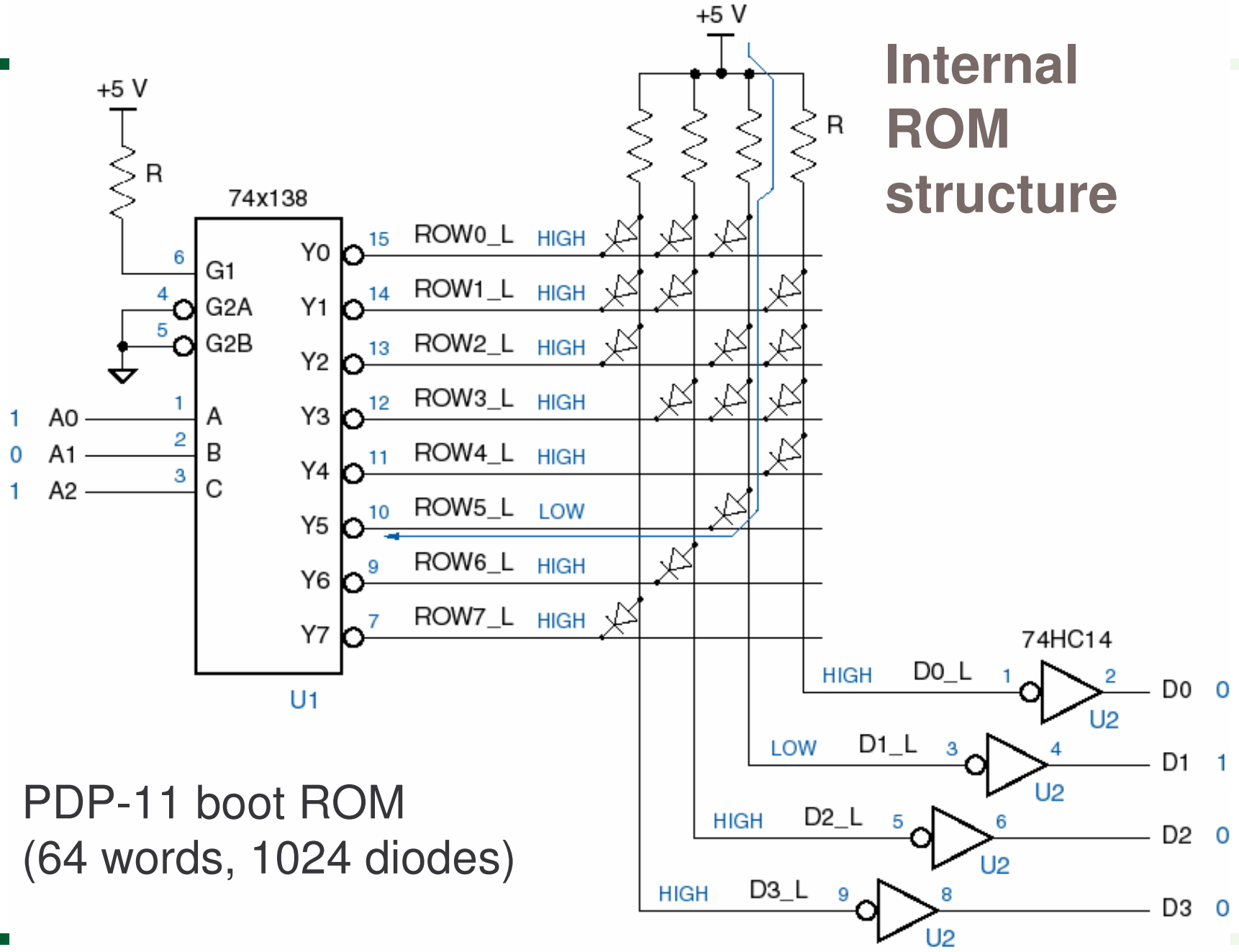
Inputs			*Outputs*			
A2	*A1*	*A0*	*D3*	*D2*	*D1*	*D0*
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

4x4 multiplier example



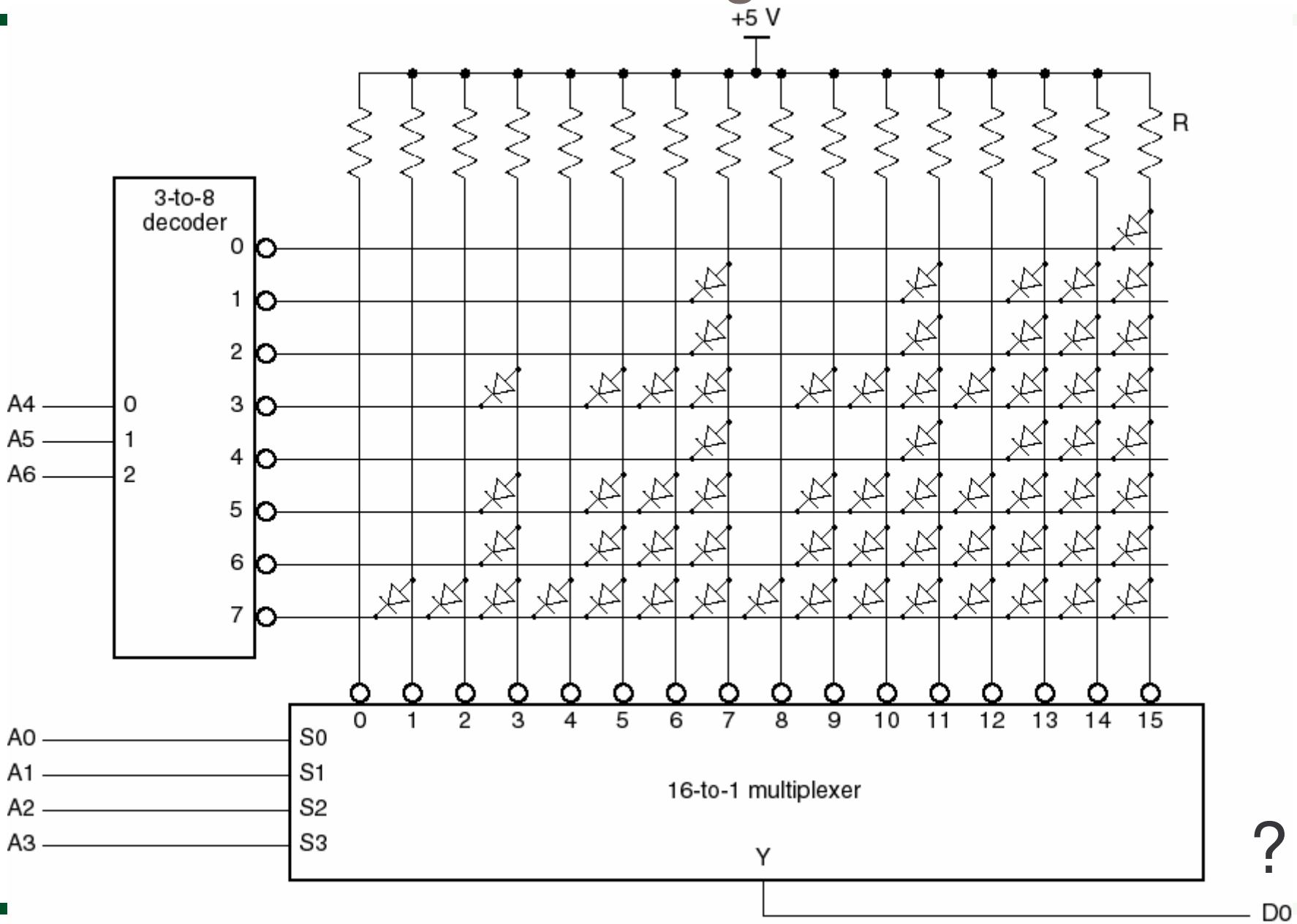

```
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
20: 00 02 04 06 08 0A 0C 0E 10 12 14 16 18 1A 1C 1E
30: 00 03 06 09 0C 0F 12 15 18 1B 1E 21 24 27 2A 2D
40: 00 04 08 0C 10 14 18 1C 20 24 28 2C 30 34 38 3C
50: 00 05 0A 0F 14 19 1E 23 28 2D 32 37 3C 41 46 4B
60: 00 06 0C 12 18 1E 24 2A 30 36 3C 42 48 4E 54 5A
70: 00 07 0E 15 1C 23 2A 31 38 3F 46 4D 54 5B 62 69
80: 00 08 10 18 20 28 30 38 40 48 50 58 60 68 70 78
90: 00 09 12 1B 24 2D 36 3F 48 51 5A 63 6C 75 7E 87
A0: 00 0A 14 1E 28 32 3C 46 50 5A 64 6E 78 82 8C 96
B0: 00 0B 16 21 2C 37 42 4D 58 63 6E 79 84 8F 9A A5
C0: 00 0C 18 24 30 3C 48 54 60 6C 78 84 90 9C A8 B4
D0: 00 0D 1A 27 34 41 4E 5B 68 75 82 8F 9C A9 B6 C3
E0: 00 0E 1C 2A 38 46 54 62 70 7E 8C 9A A8 B6 C4 D2
F0: 00 0F 1E 2D 3C 4B 5A 69 78 87 96 A5 B4 C3 D2 E1
```

Internal ROM structure

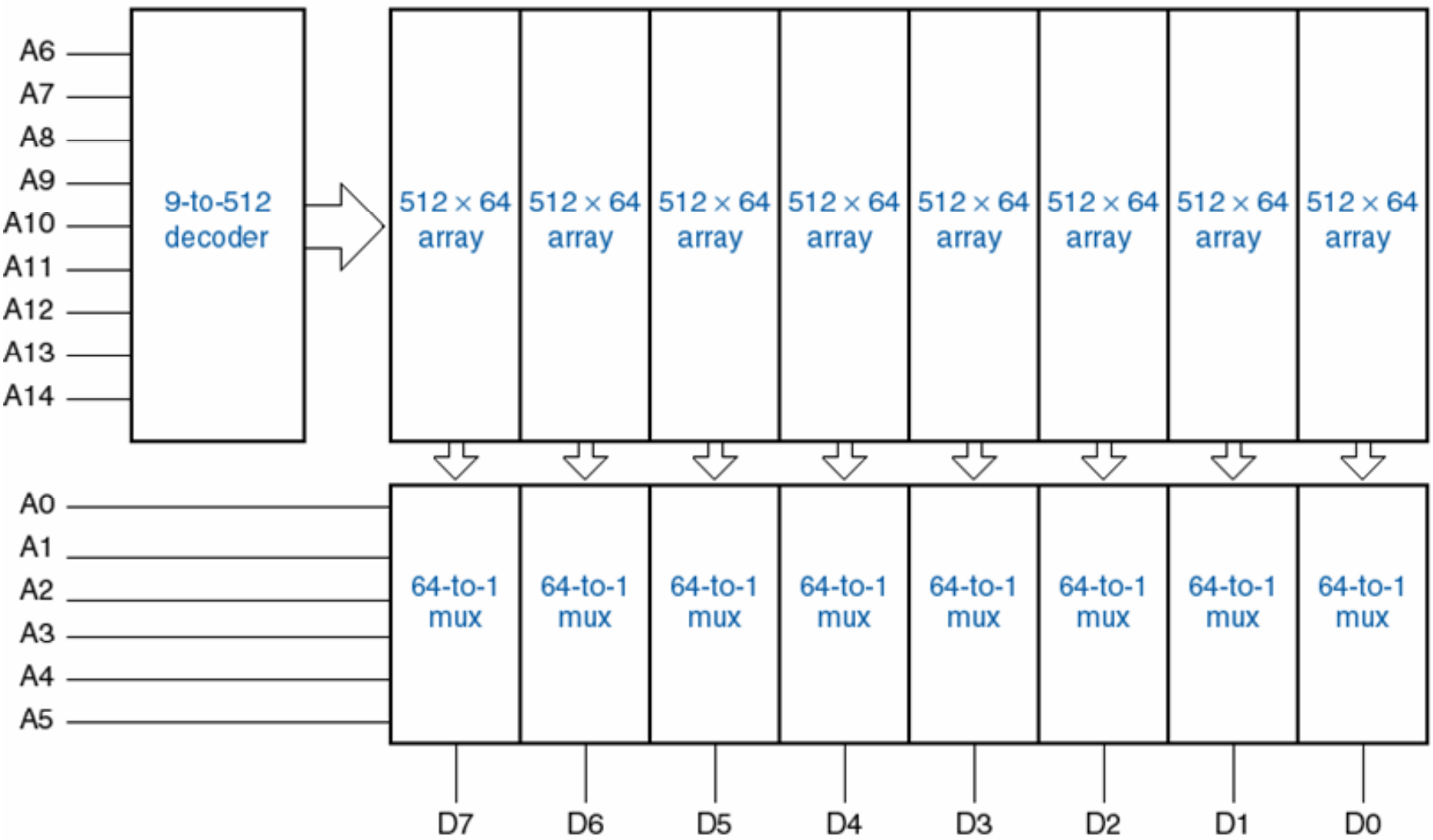


PDP-11 boot ROM
(64 words, 1024 diodes)

Two-dimensional decoding



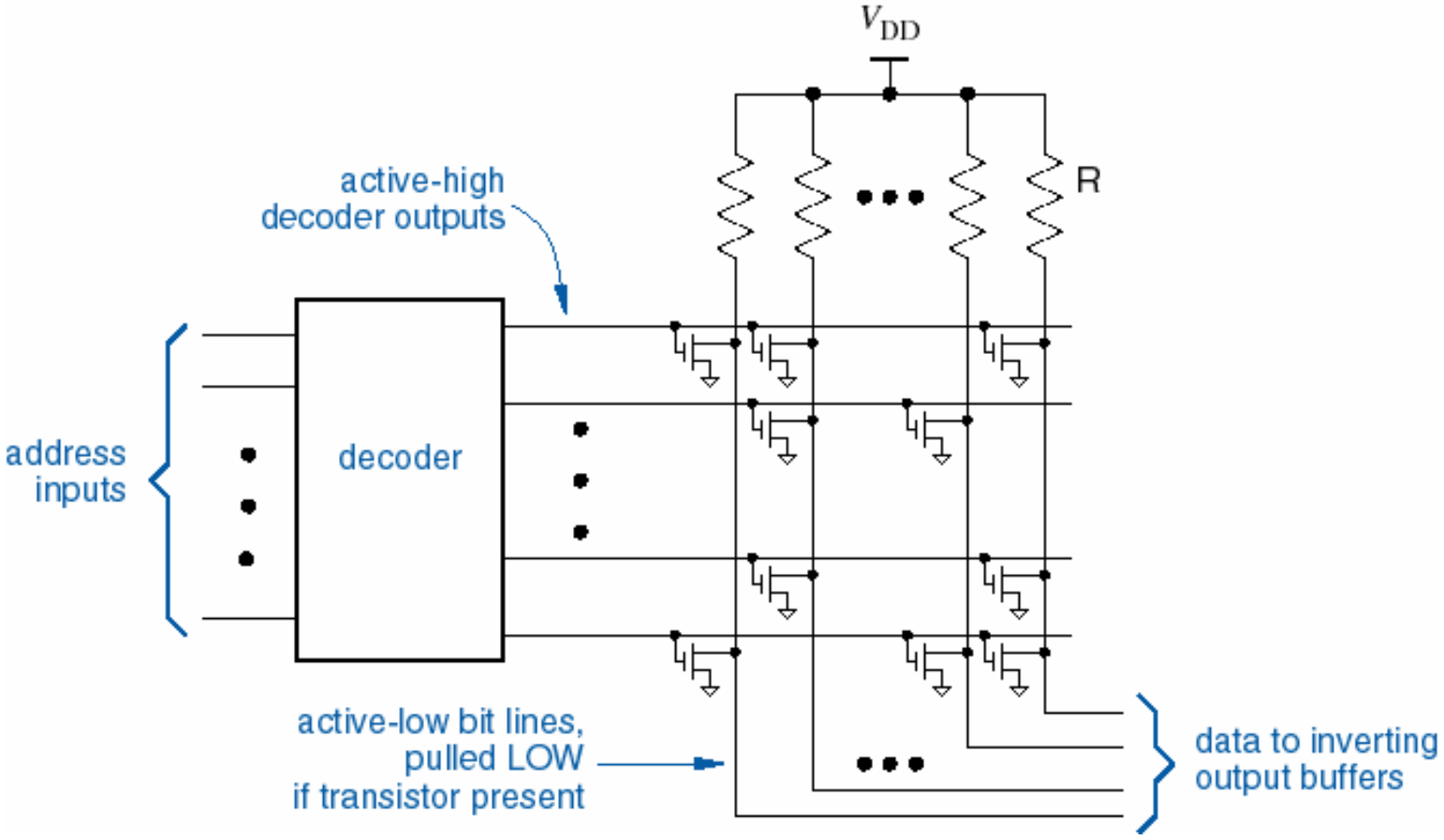
Larger example, 32Kx8 ROM



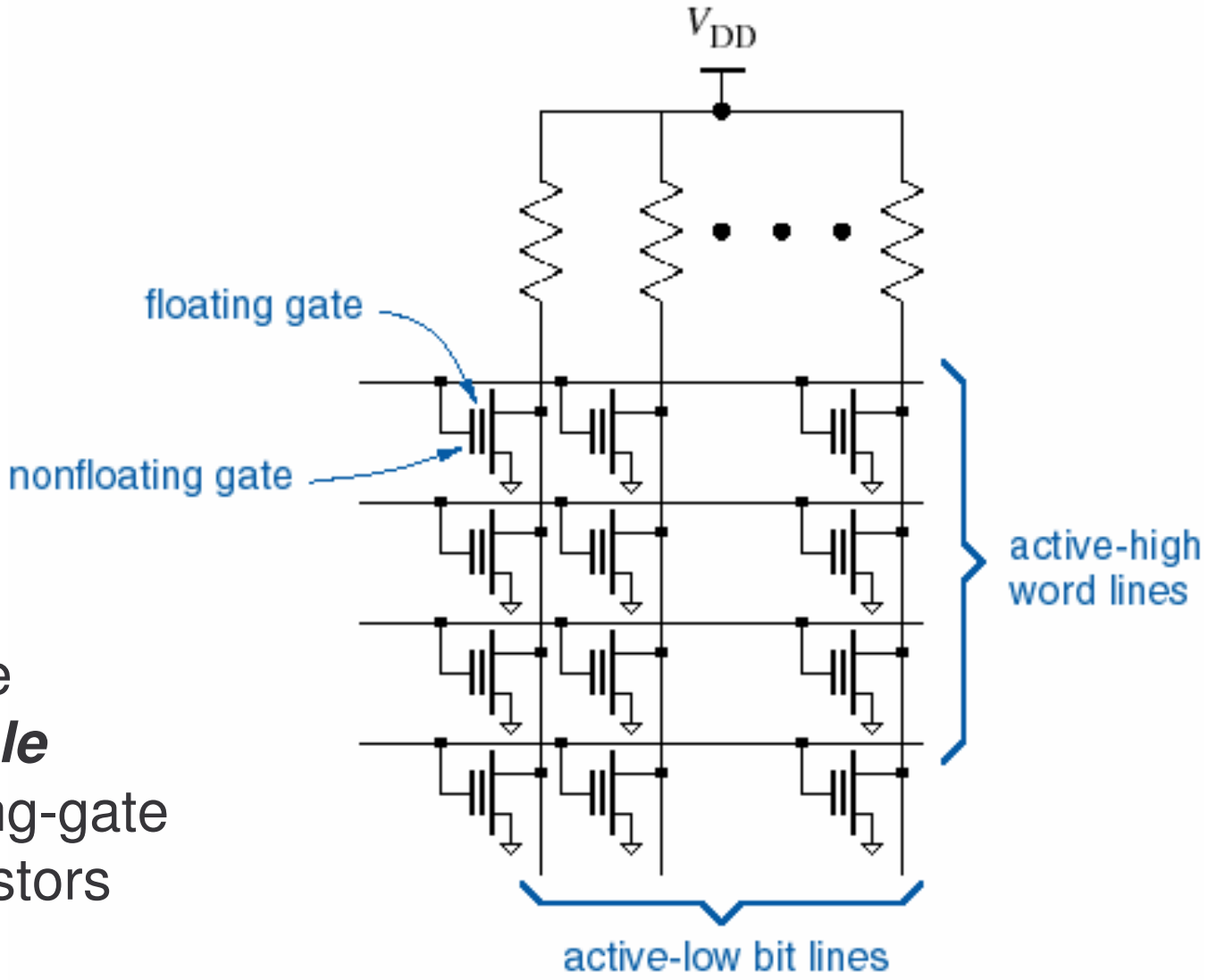
Today's ROMs

256K bytes, 1M byte, or larger

Use MOS transistors

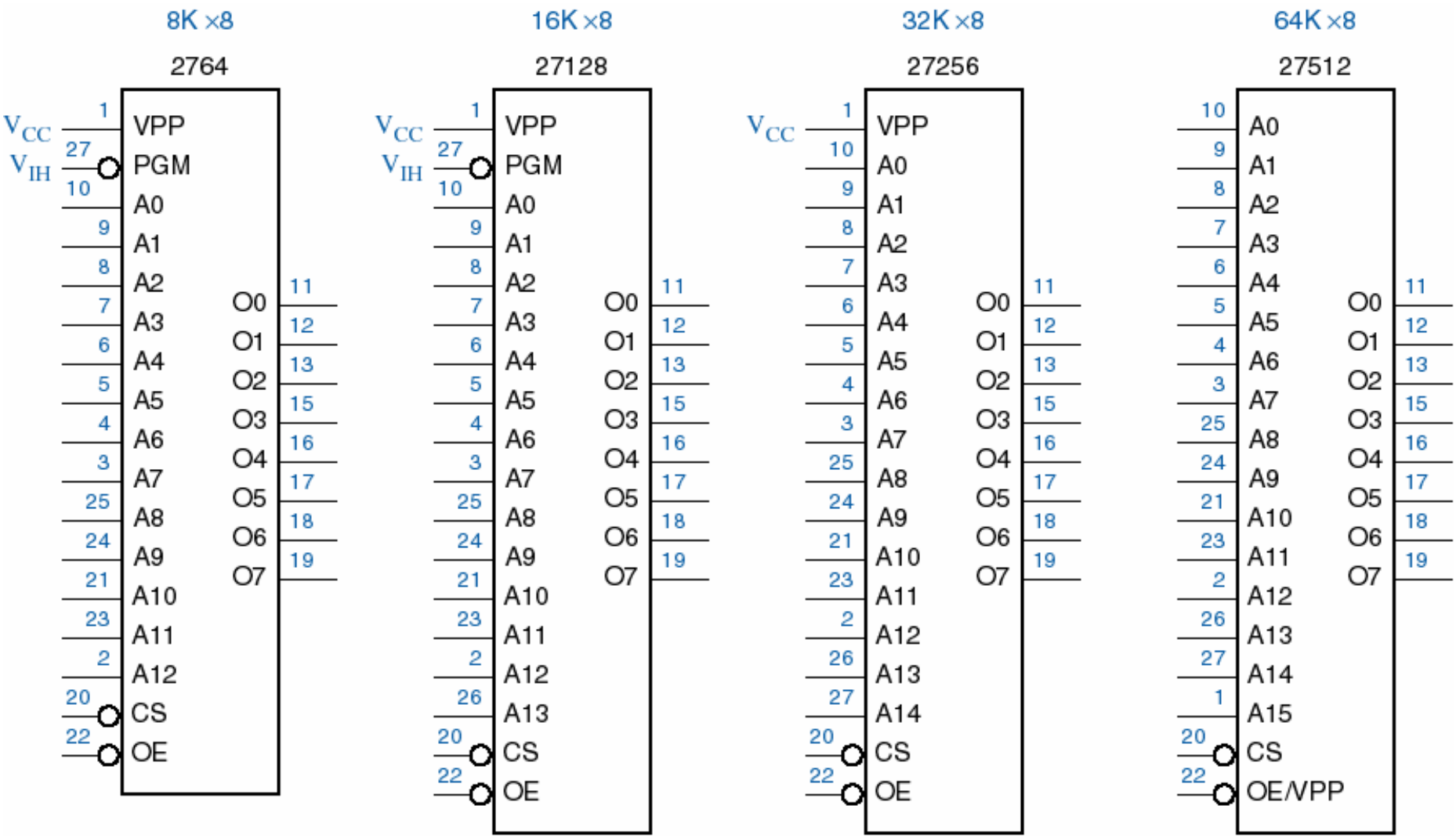


EEPROMs, Flash PROMs



Programmable
and ***erasable***
using floating-gate
MOS transistors

Typical commercial EEPROMs



EEPROM programming

Apply a higher voltage to force bit change

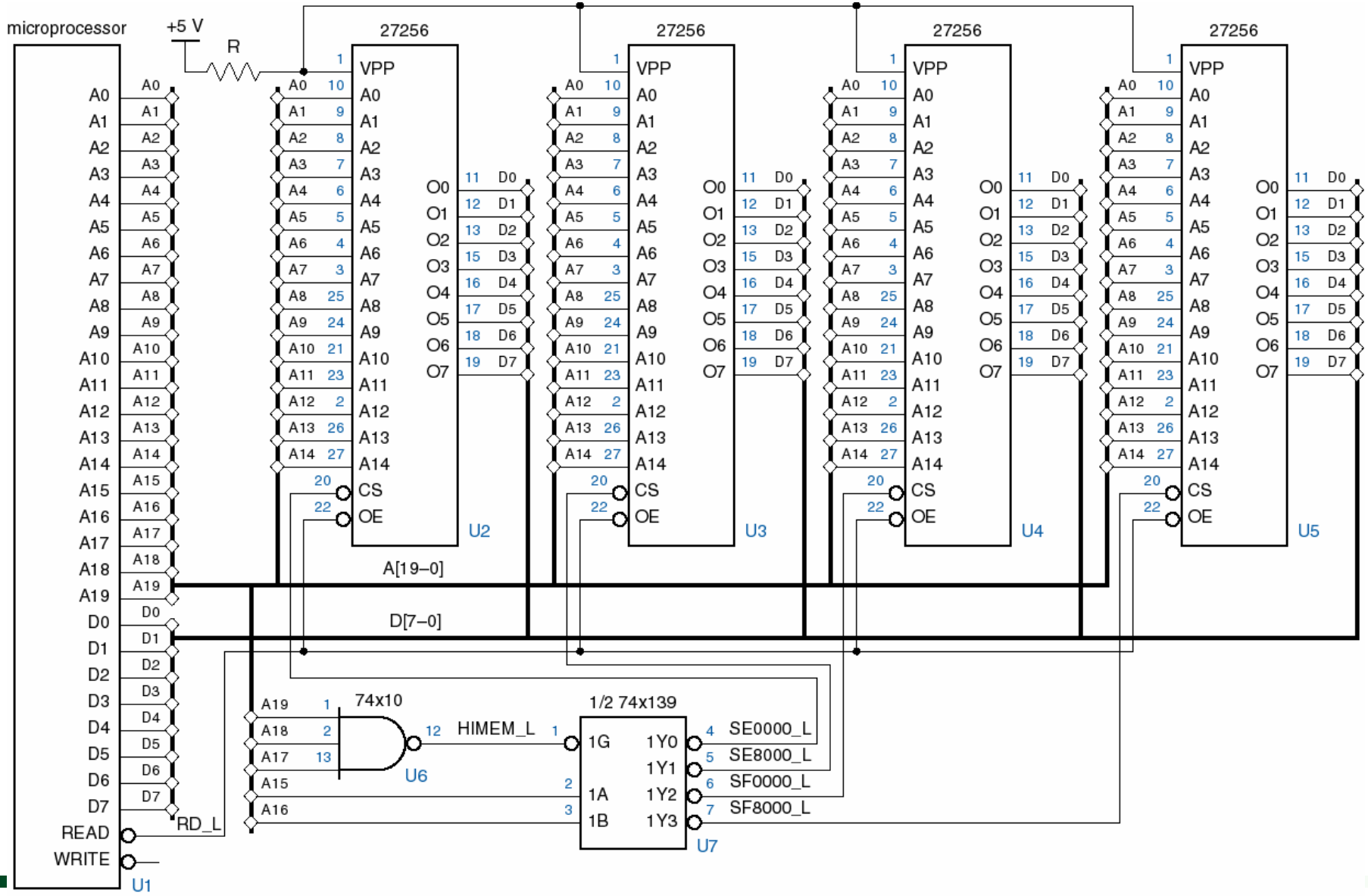
- E.g., VPP = 12 V
- On-chip high-voltage “charge pump” in newer chips

Erase bits

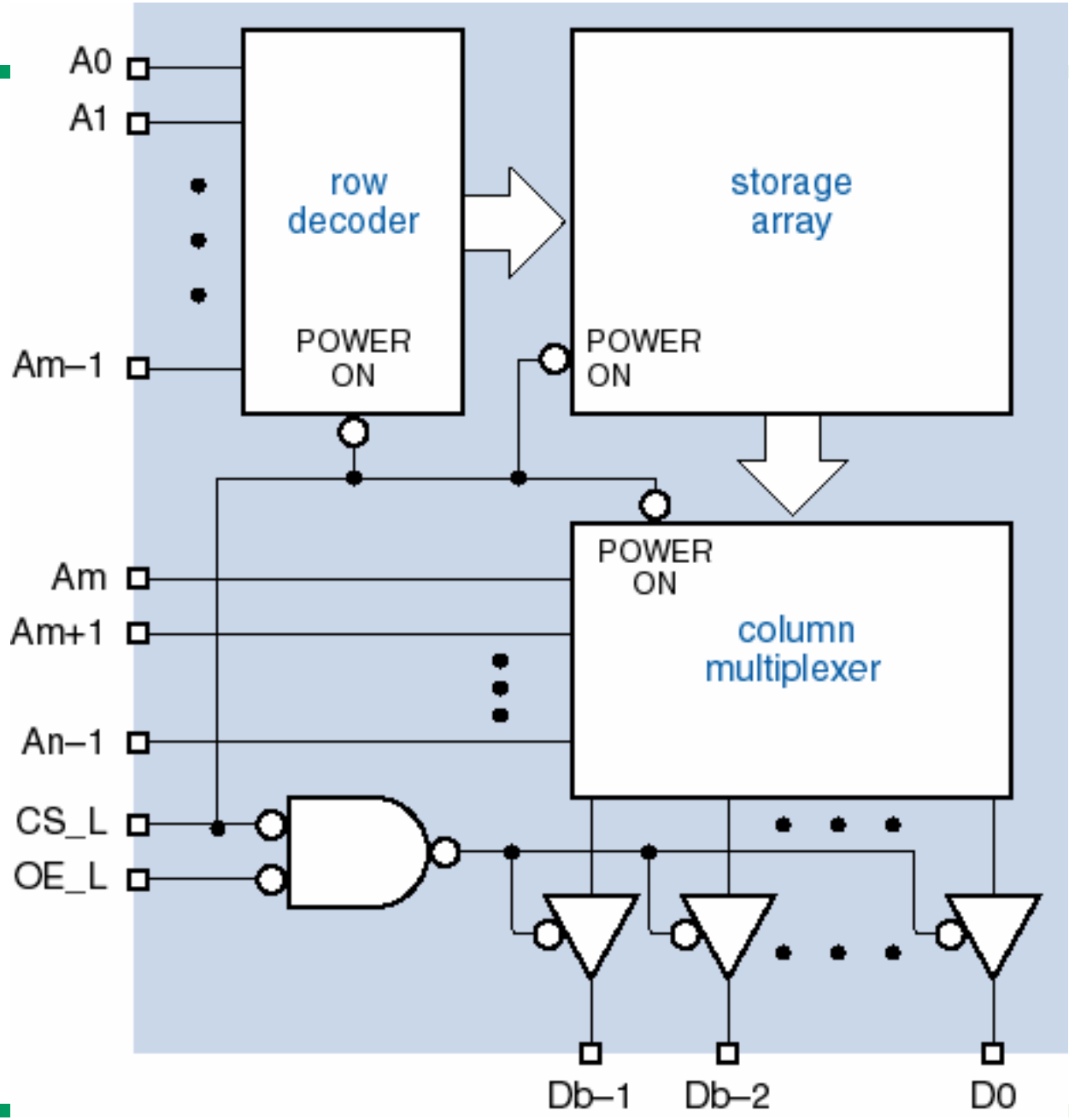
- Byte-byte
- Entire chip (“flash”)
- One block (typically 32K - 66K bytes) at a time

Programming and erasing are a lot slower than reading (milliseconds vs. 10’s of nanoseconds)

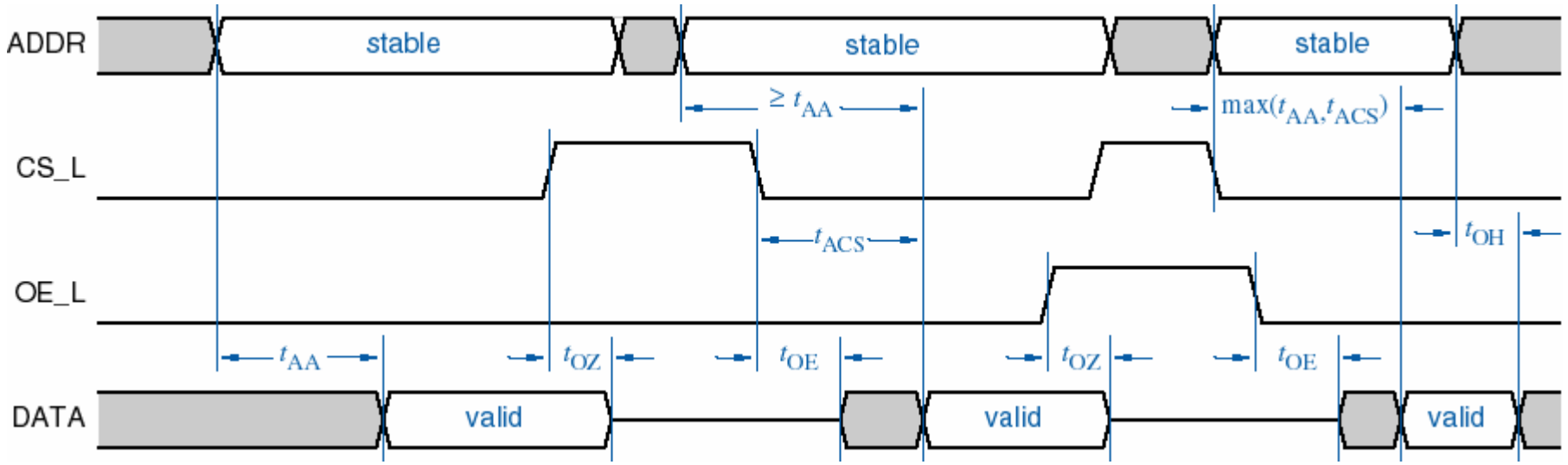
Microprocessor EPROM application



ROM control and I/O signals



ROM timing


ADDR
CS_L
OE_L
DATA
stable
stable
stable
≥ t_{AA}
max(t_{AA},t_{ACS})
t_{ACS}
t_{OH}
t_{AA}
t_{OZ}
t_{OE}
t_{OZ}
t_{OE}
valid
valid
valid