



Faculty of Engineering

**CSE115: Digital Design**

**Lecture 21:**  
**Combinational Multipliers & ROMs**

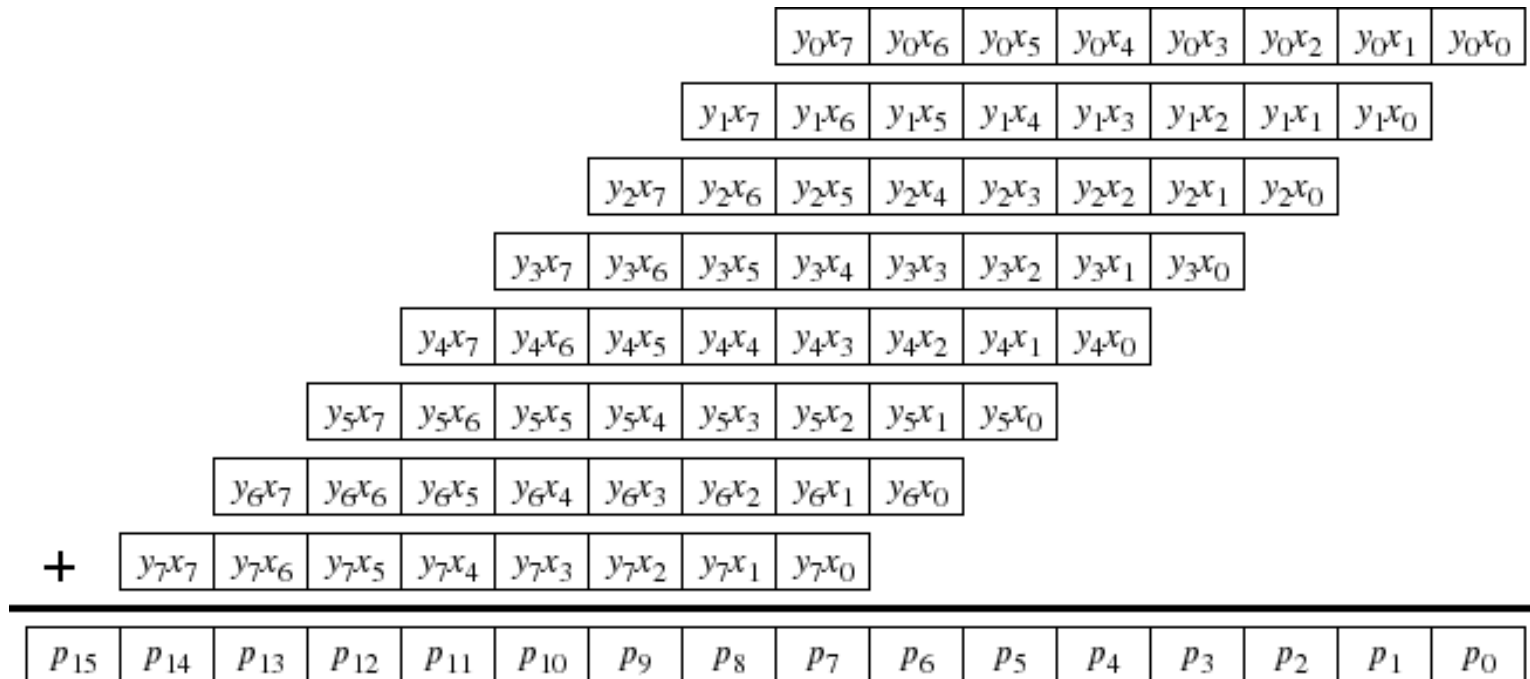
# Suggested Reading

- [Section 5.11](#)

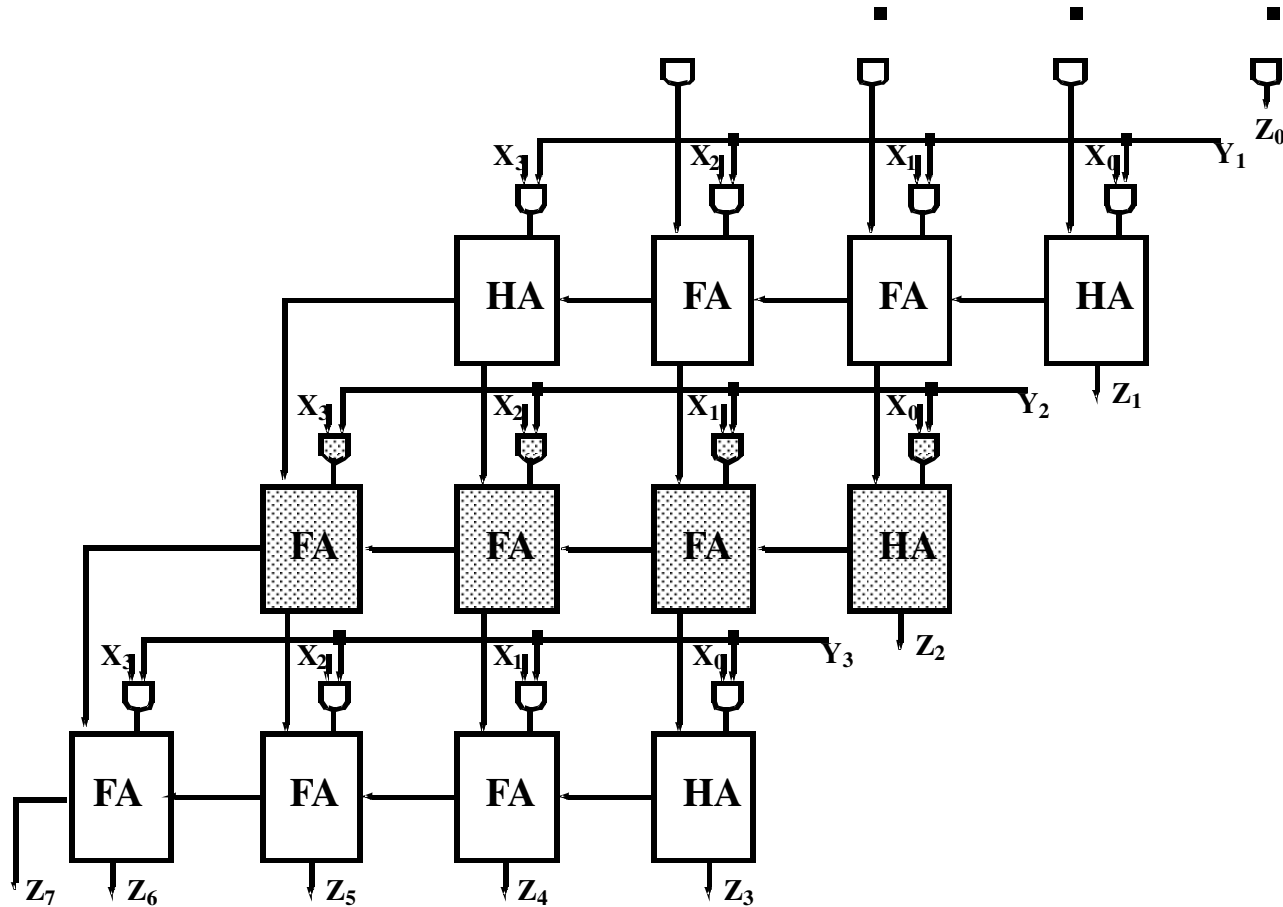


# Multipliers

8x8 multiplier



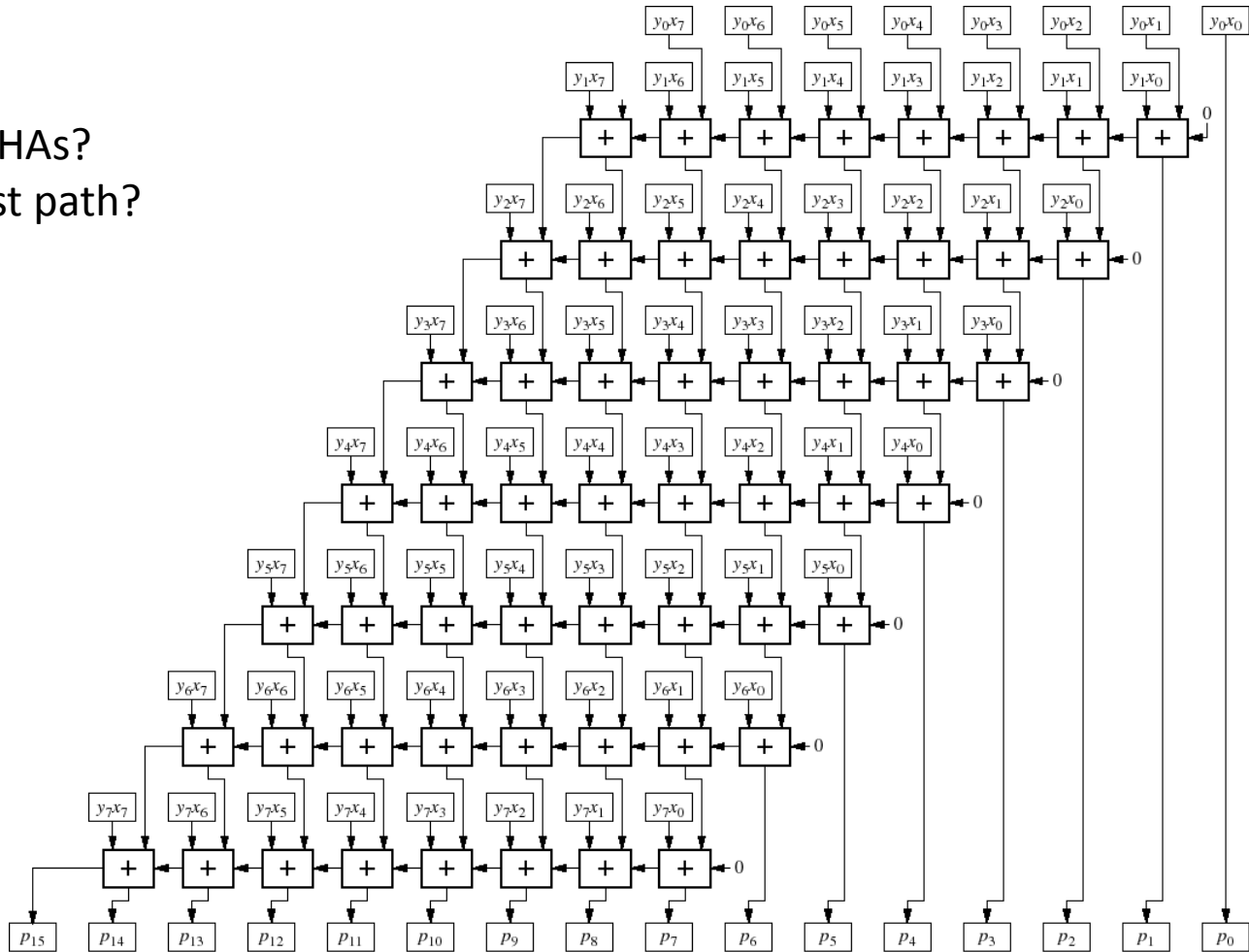
# The Array Multiplier





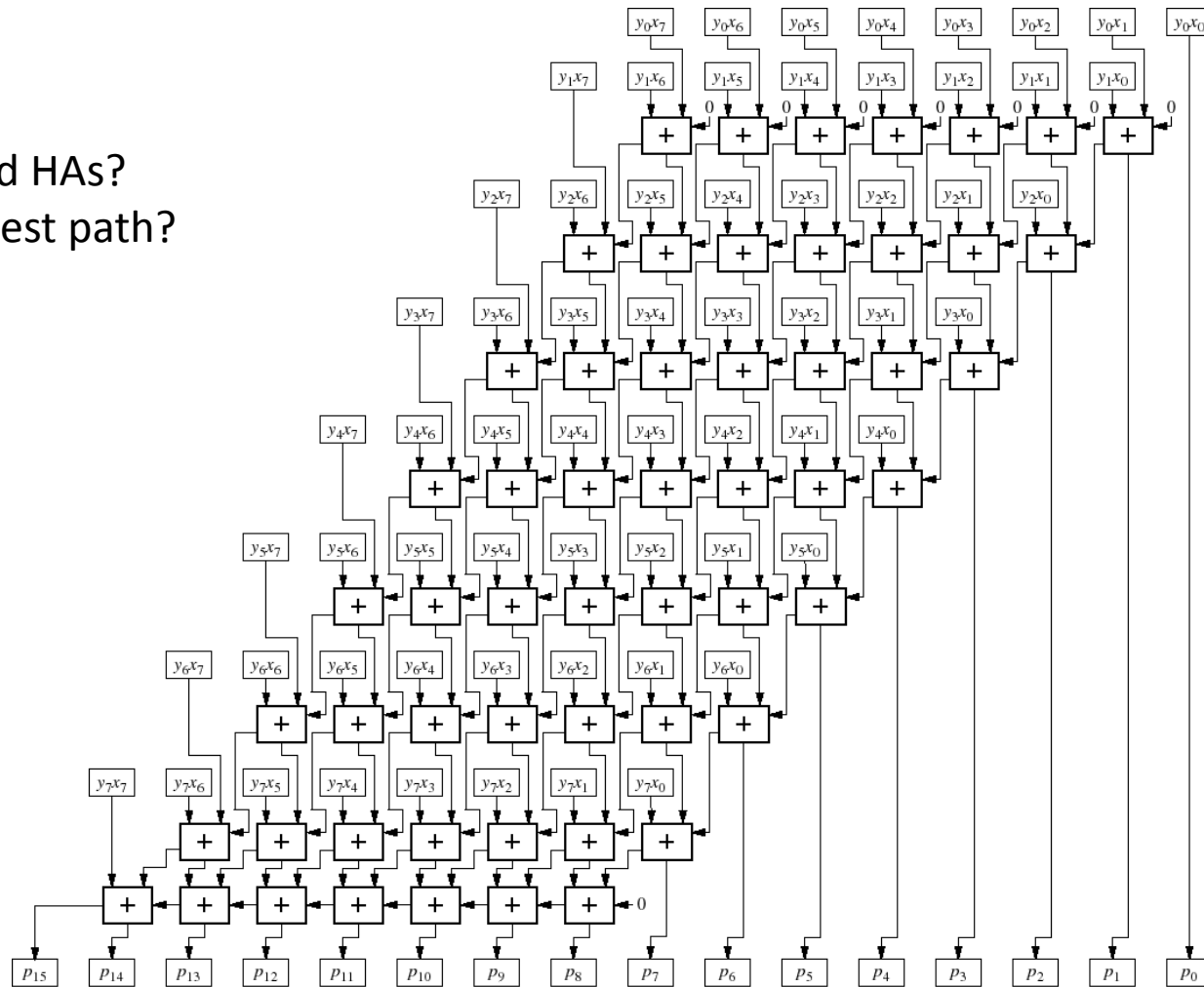
# Full-Adder Array

Number of FAs and HAs?  
The length of longest path?



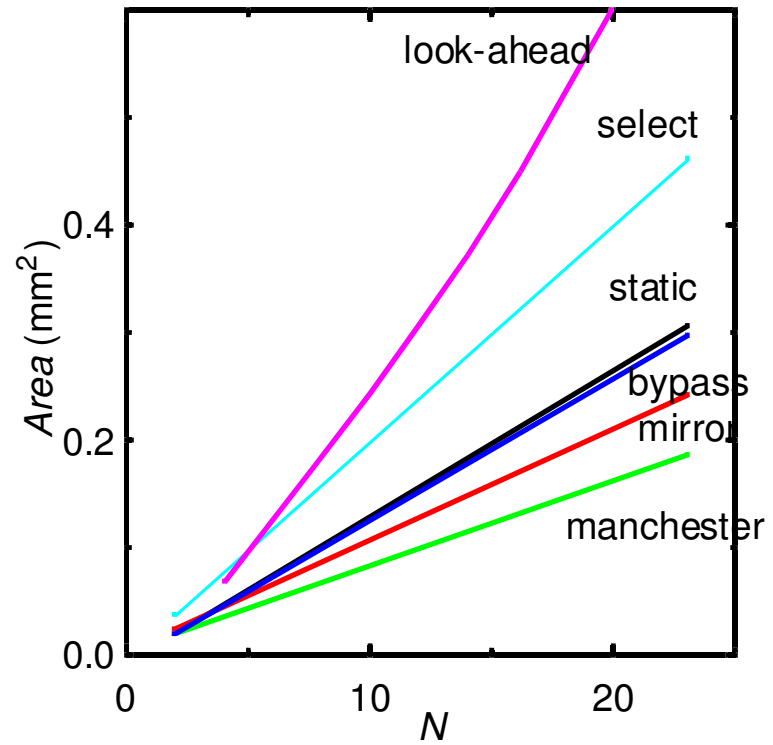
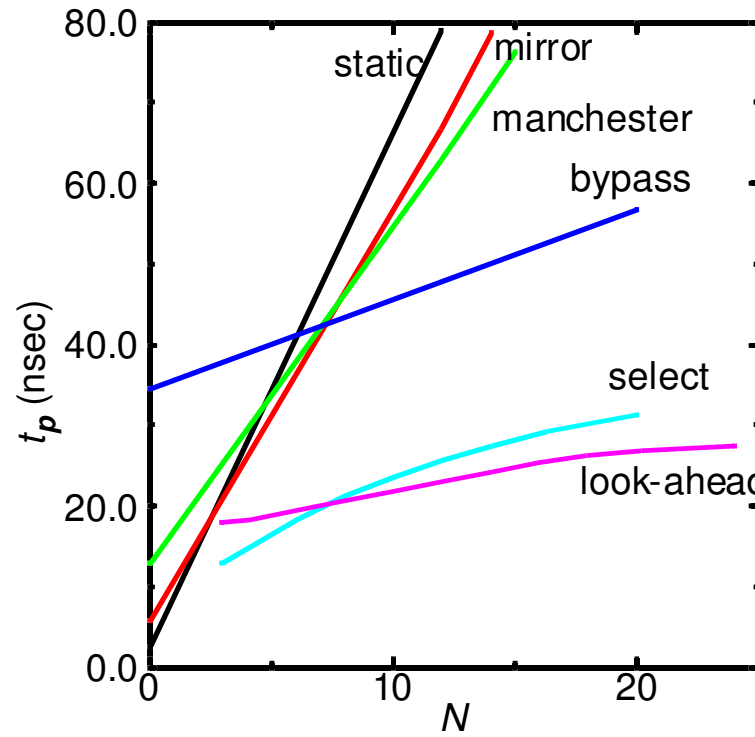
# Faster Carry Chain

Number of FAs and HAs?  
The length of longest path?

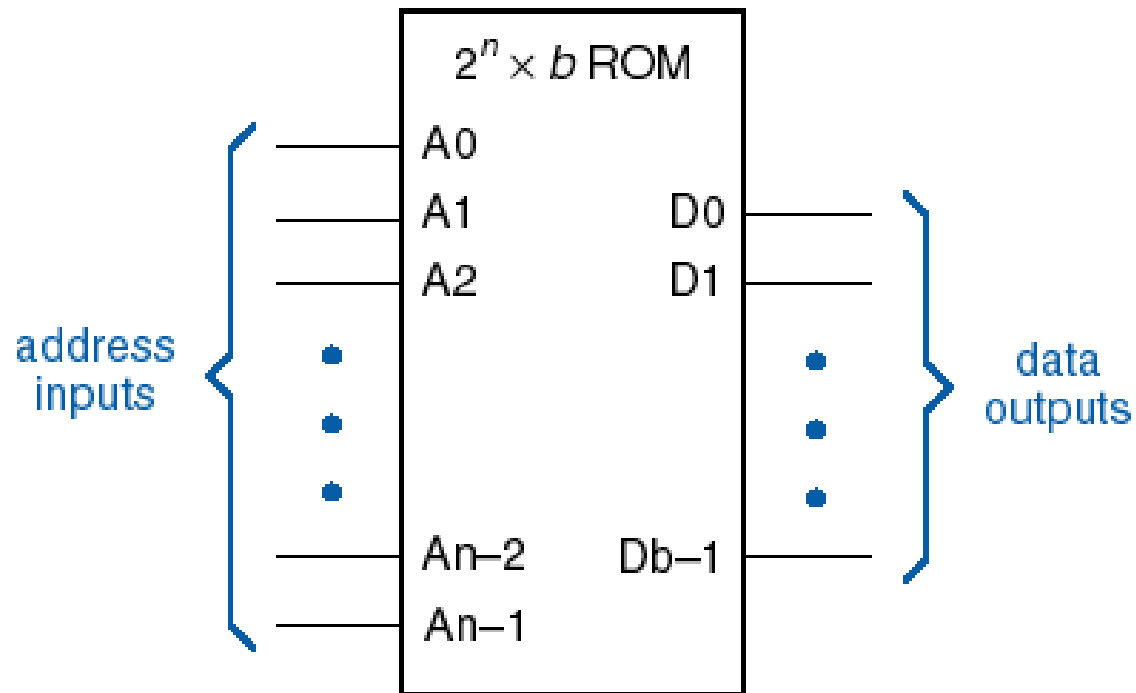




# Design as a Trade-Off



# Read-Only Memories



# Why 'ROM'?

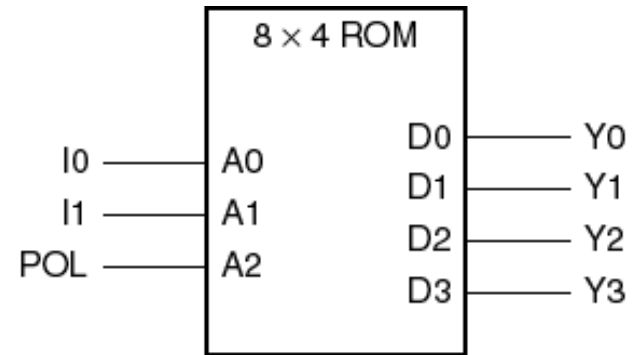
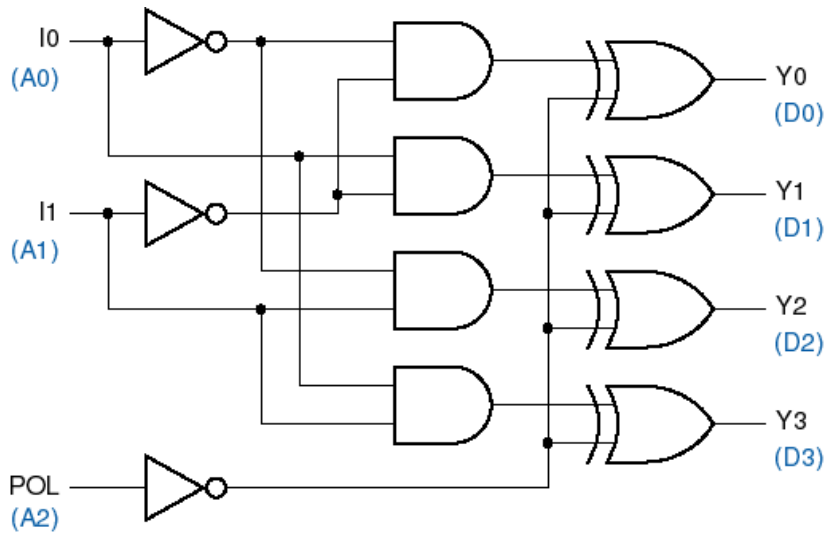
## Program storage

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

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

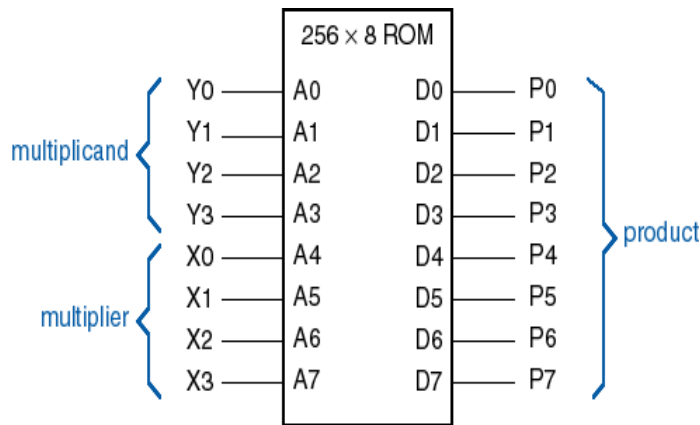
- Can perform any combinational logic function
- **Address** inputs = **function** inputs
- **Data** outputs = **function** outputs

# Logic-in-ROM Example



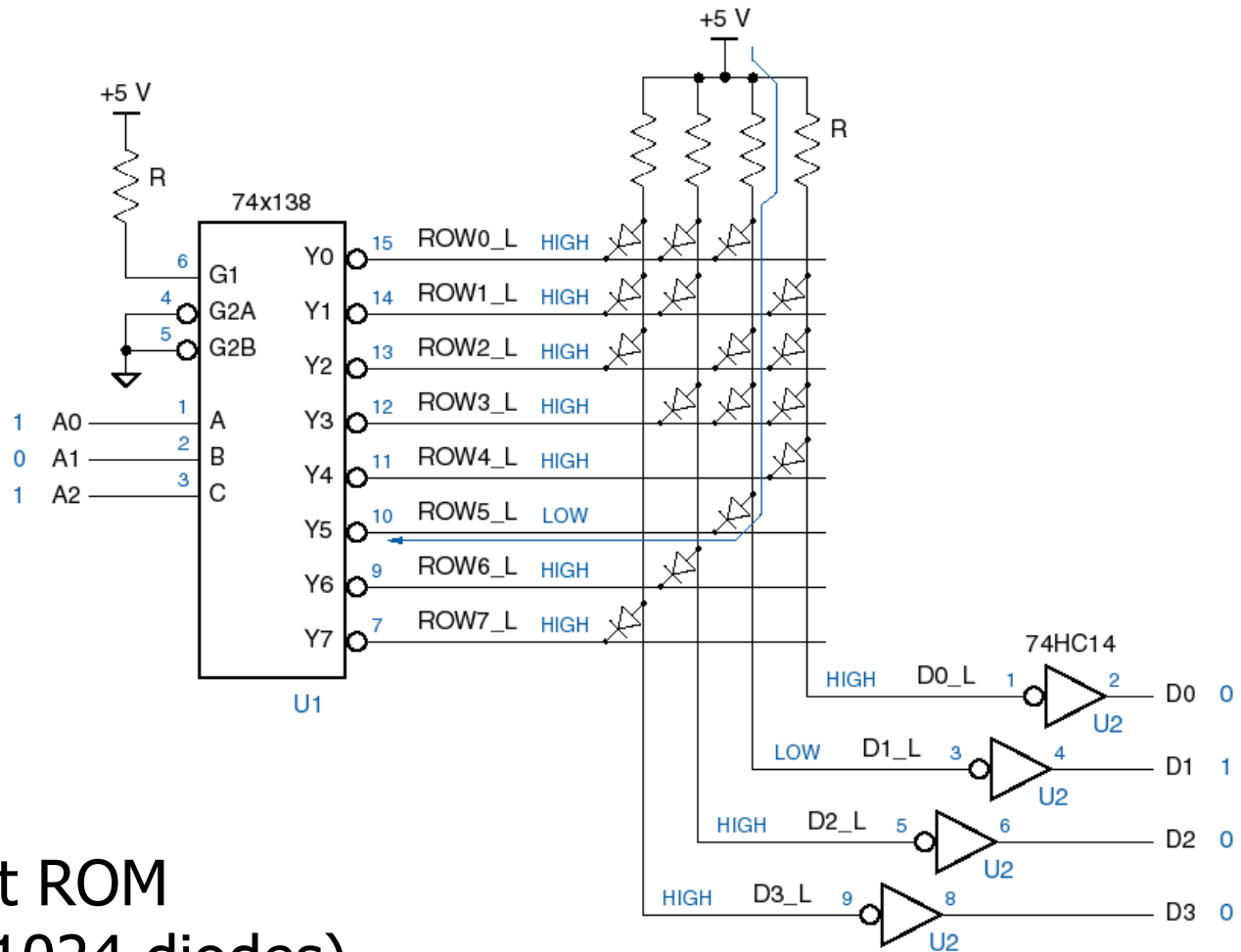
Inputs			Outputs			
A2	A1	A0	D3	D2	D1	D0
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

# 4x4 Multiplier Example



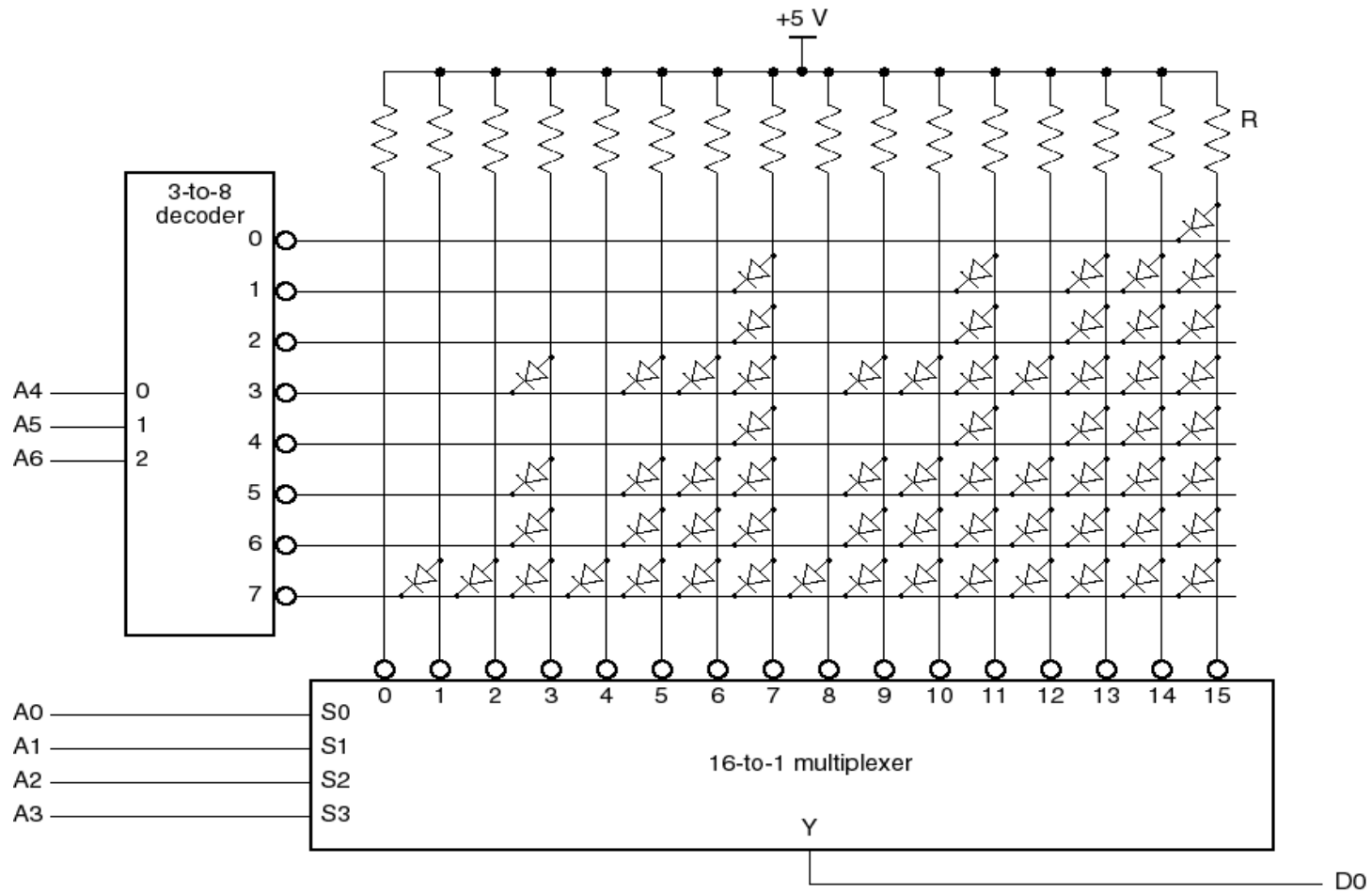
00:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
20:	00	02	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
30:	00	03	06	09	0C	0F	12	15	18	1B	1E	21	24	27	2A	2D
40:	00	04	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C
50:	00	05	0A	0F	14	19	1E	23	28	2D	32	37	3C	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

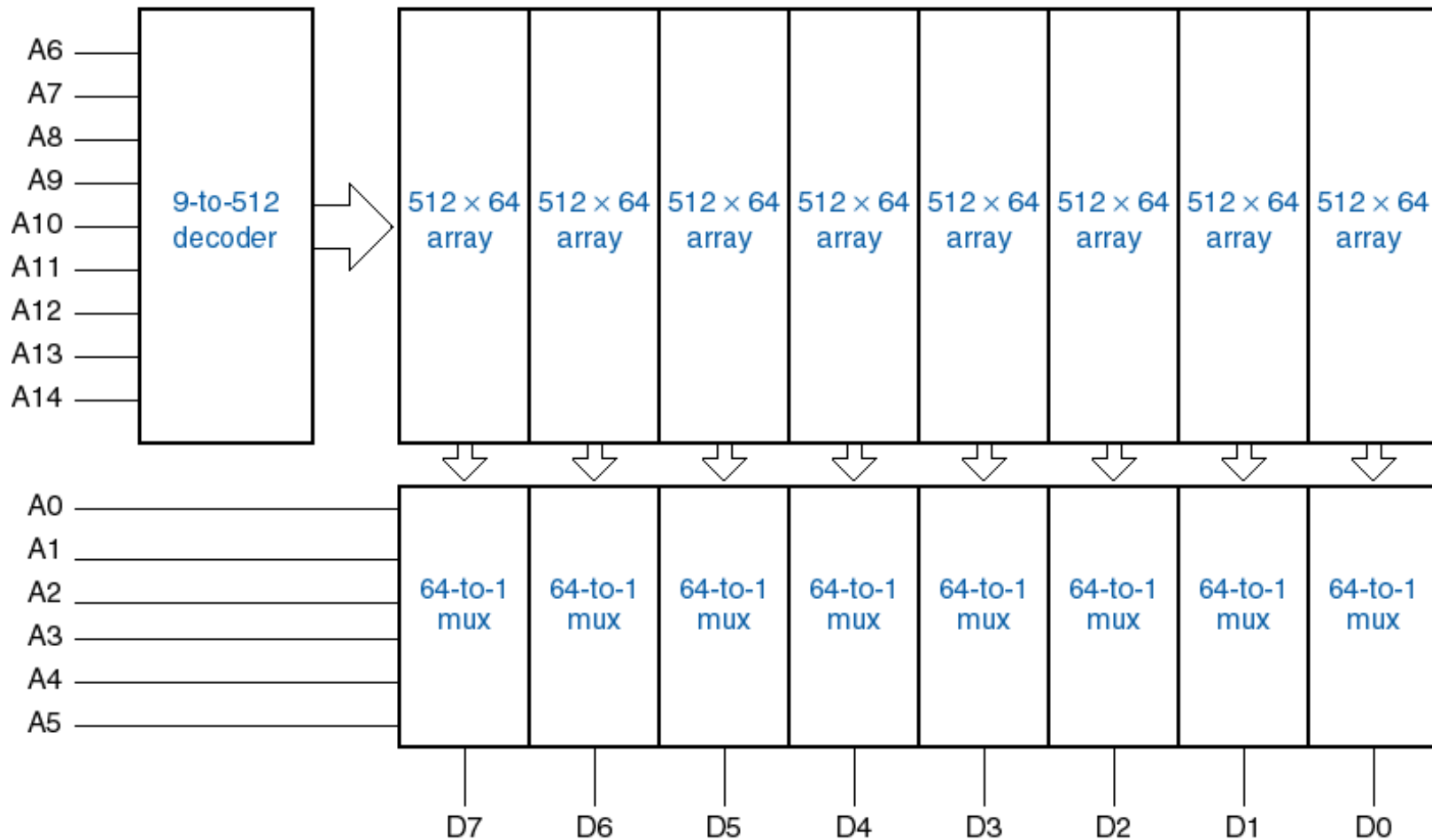


PDP-11 boot ROM  
(64 words, 1024 diodes)

# Two-Dimensional Decoding



# Larger Example, 32Kx8 ROM

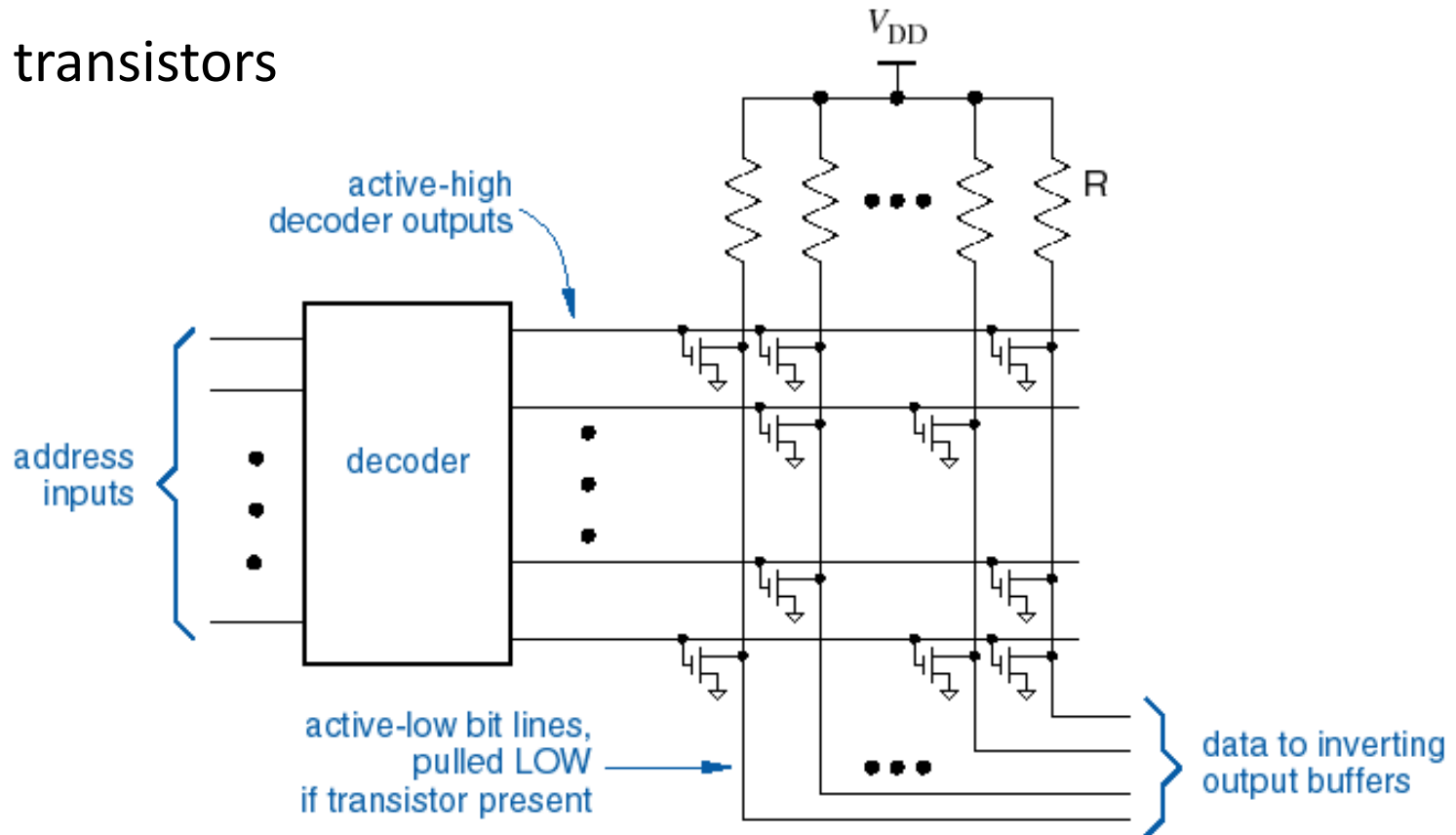




# ROMs

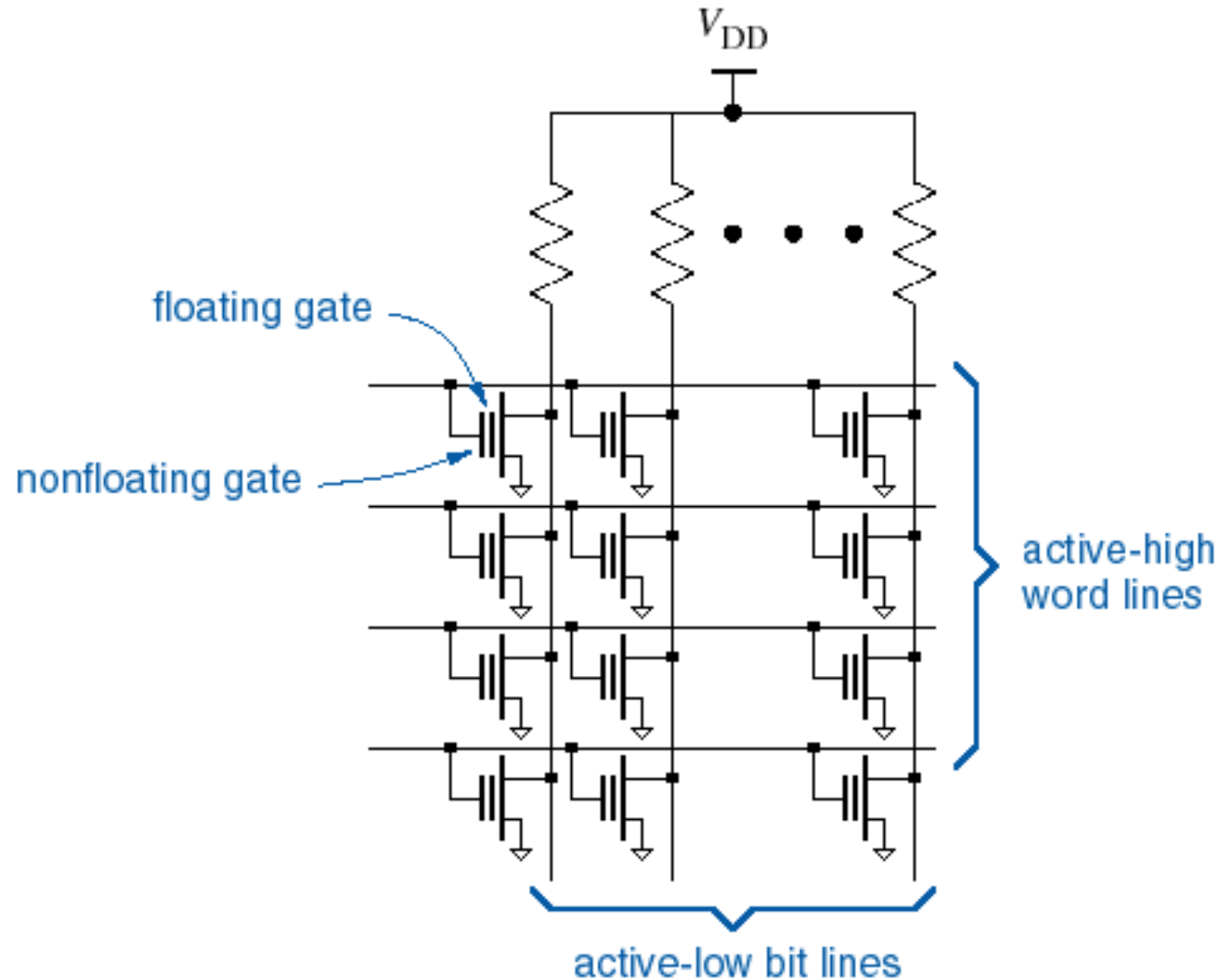
256K bytes, 1M byte, or larger

Use MOS transistors

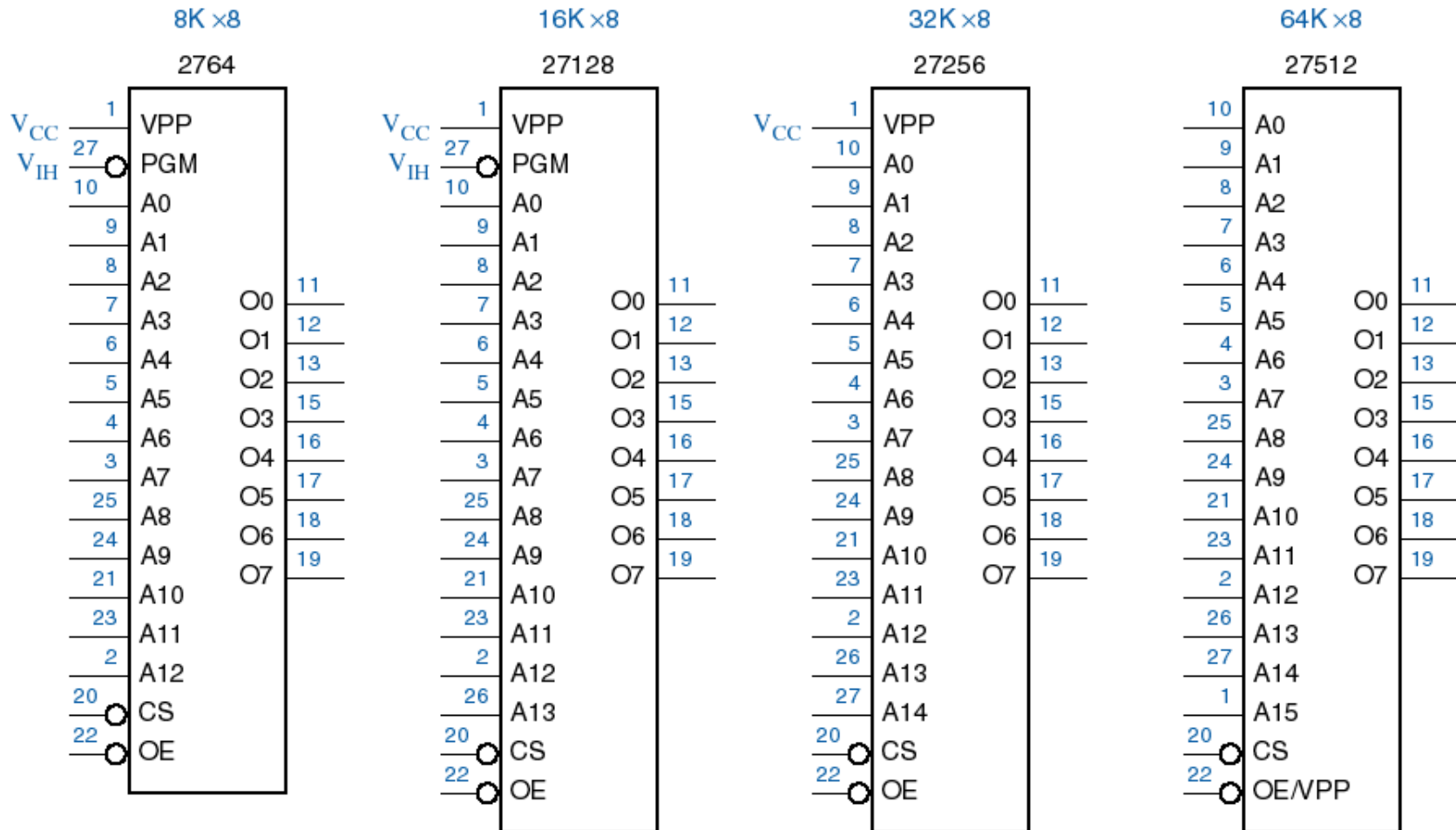


# EEPROMs, Flash PROMs

Programmable  
and *erasable*  
using floating-gate  
MOS transistors



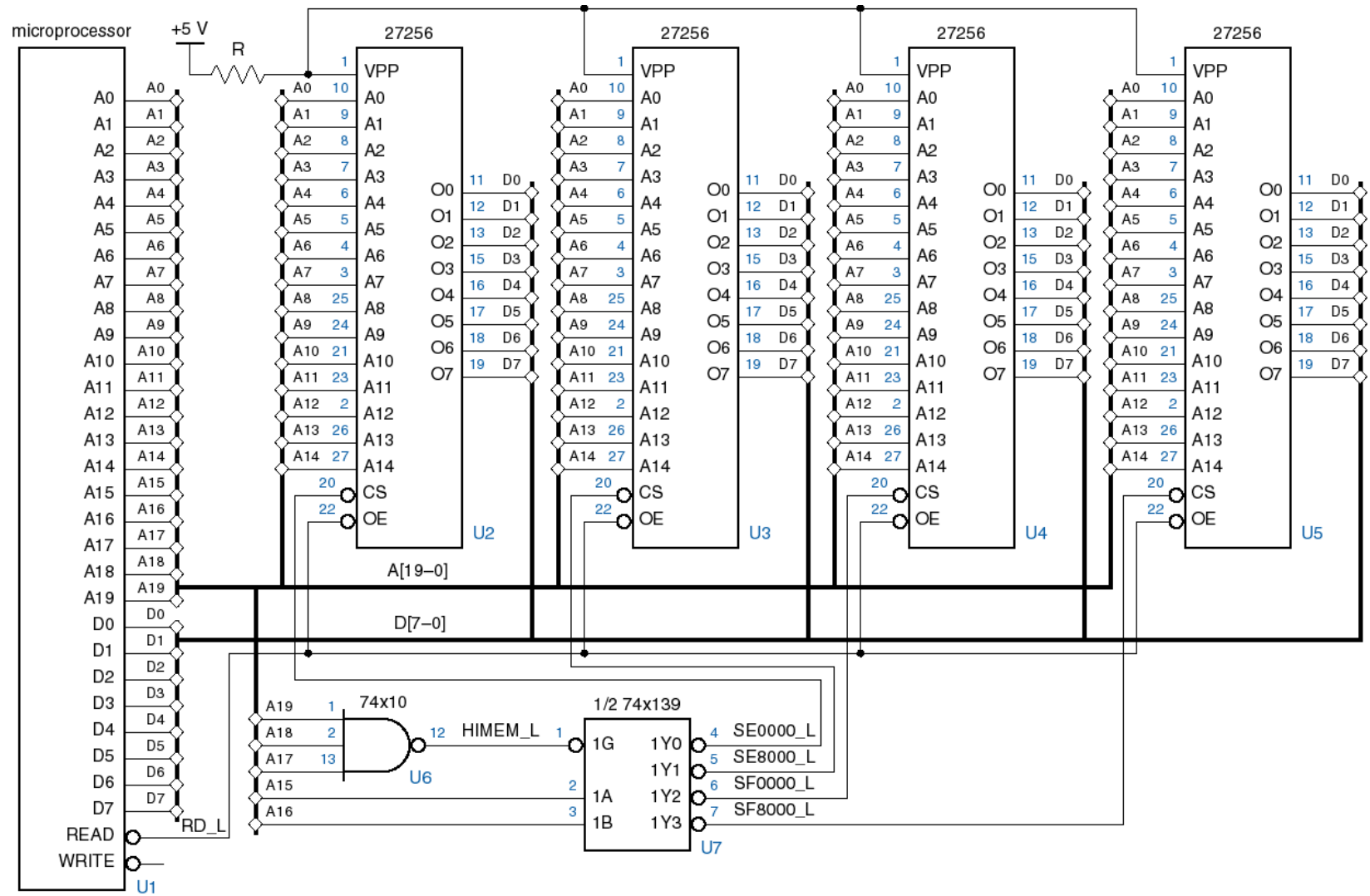
# Typical Commercial EEPROMs



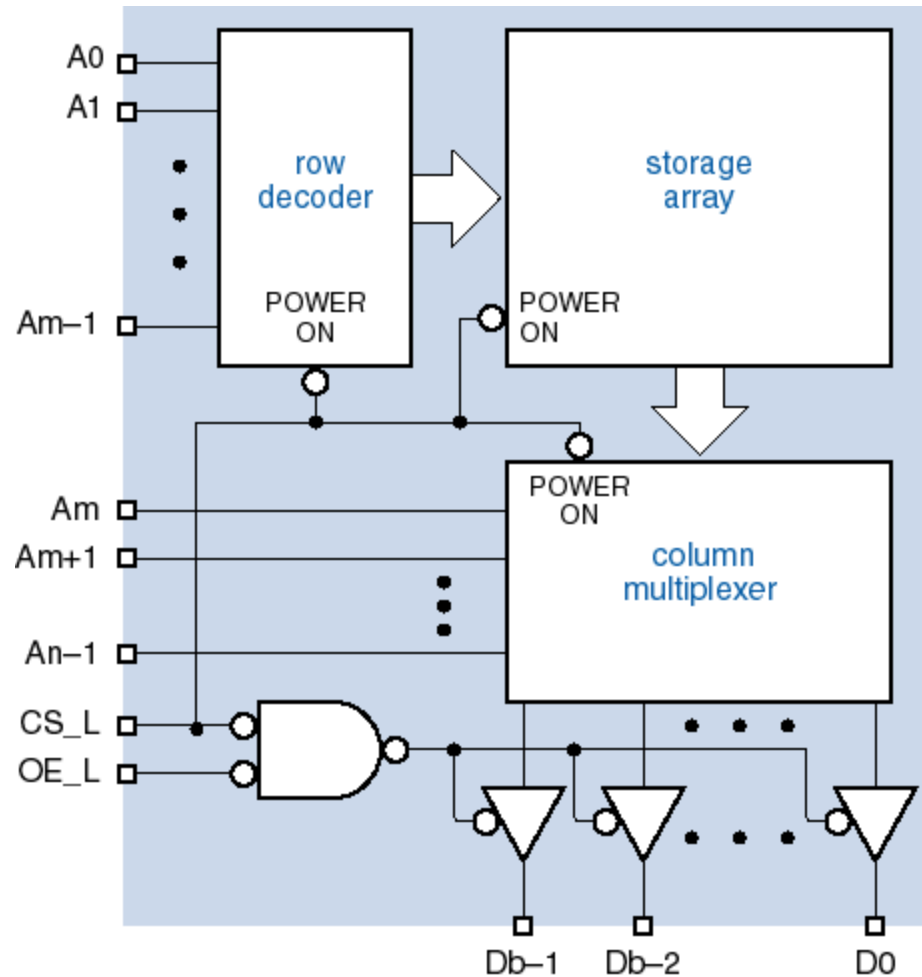
# EEPROM Programming

- Apply a **higher voltage** to force bit change
  - e.g.,  $V_{pp} = 12\text{ V}$
  - On-chip high-voltage “charge pump” in newer chips
- **Erase bits**
  - Byte-byte
  - One block (typically 32K - 66K bytes) at a time
  - Entire chip (‘flash’)
- 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

