9 - memory and programable unit

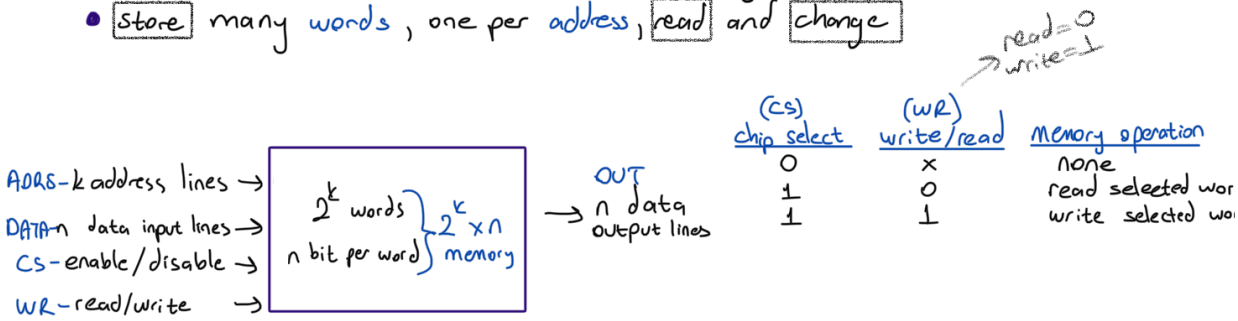
memory unit = storage cells + circuits need to tranfer data in and out

two types of memory (in digital systems)
① RAM - random access memory — can perform both read and write
② ROM - read-only memory — also considered a programmable logic device

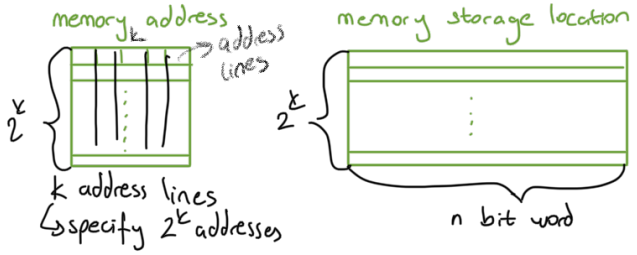
words = a memory unit stores binary information in groups of bits (mostly 8 bits=1 byt
↳ a memory word may represent a number, an instruction, a char, etc.

RAM

- bunch of registers connected together, allowing users to sellect a particular address to read
- access times for different locations are always the same
- store many words, one per address, read and change

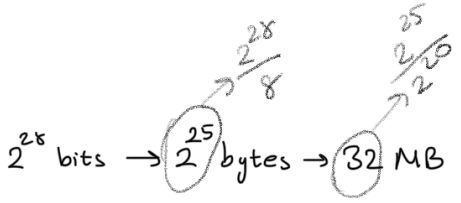


memory sizes



2^k x n memory
(storage capacity)

base 2
K - kilo - 2^{10} = 1.024
M - mega - 2^{20} = 1.048.576
G - giga - 2^{30} = 1.073.741.824



- pc's usually come with 8GB / 16GB of RAM, smartphones have 2 / 4GB of memory

virtual memory = makes the memory larger than RAM's capacity.
↳ the operating system uses hard disk as a virtual memory

reading RAM

- CS → 1 (enable)
- content of the

clock → memory unit does not have an internal clock
↳

- WS → 0 (read)
- ADRS → desired address (inputs) / address (output)

*data input unused

→ uses external devices (CPU's)

access time = select a word and read it

cycle time = complete a write operation

writing RAM

- CS → 1 (enable)
- WS → 1 (write)
- ADRS → desired address
- DATA → word to store

} inputs

* no output needed

types of memories

fast / cost more

① static ram (SRAM) = the stored data remains valid as long as the power is supplied
↳ use latches instead of flip-flops = because lathes are faster, cheaper, less hardware, but getting the timing exactly right is problematic

② dynamic ram (DRAM) = built with capacitors → 1: stored charge, 0: no charge
↳ capacitors lose their charge after a few milliseconds → memory requires constant refreshing to charge the capacitor (that is
↳ smaller than SRAM (DRAM = 1 bit 1 capacitor; SRAM = 1 cell 4-6 transistors)
↳ cheaper, and use less power than SRAM (↳ easier to use)
↳ disadvantage = its speed

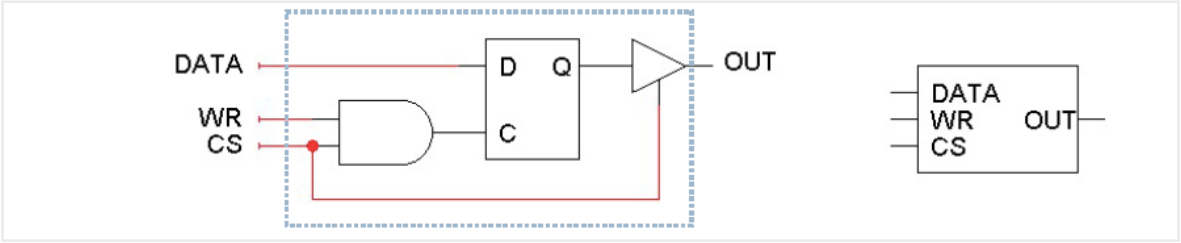
* dynamic ram is used for a computer's main memory

caches = real systems augment dynamic memory with small but fast selections of static memory (its size 128KB - 320KB, but it increas the computer's speed significantly)

Static RAM	Dynamic RAM
Made up of flip-flops.	Made up of capacitors.
Large in size.	Small in size.
Data store in the form of voltage.	Data store in the form of charge.
Much expensive as compare to dynamic RAM	Less expensive as compare to static RAM
Low storage capacity	High storage capacity.
Consume more power	Consume less power
Fast	Slow
Data sustain with time.	Data loses with time, so need refreshing circuit*.

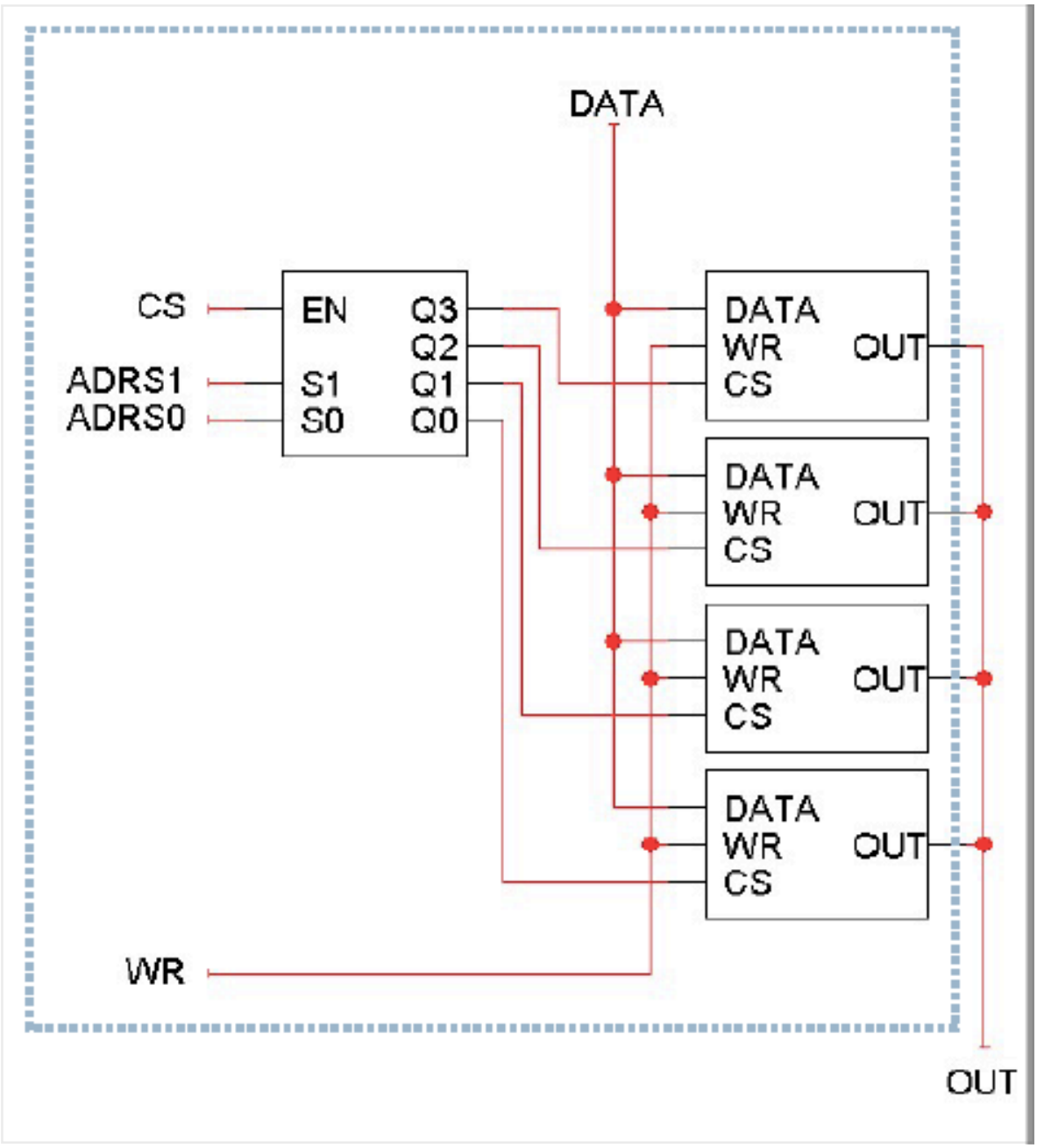
building RAM

one-bit ram cell: (no need address input, because it is just one-bit memory)

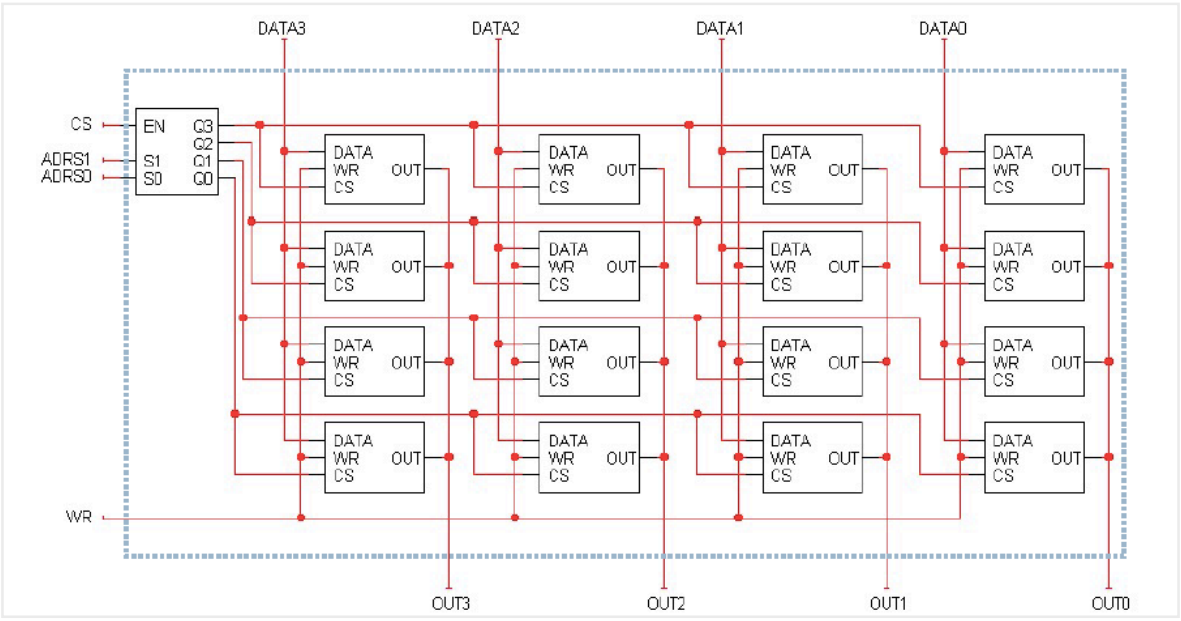


4x1 ram: 4 words, each 1 bit

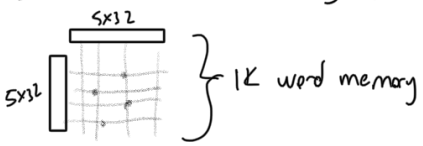
↳ only one cell can be read or written at a time


DATA
CS
ADRS1
ADRS0
EN
S1
S0
Q3
Q2
Q1
Q0
DATA
WR
CS
OUT
DATA
WR
CS
OUT
DATA
WR
CS
OUT
DATA
WR
CS
OUT
WR
OUT

4x4 ram


DATA3
DATA2
DATA1
DATA0
CS
ADRS1
ADRS0
WR
EN
S1
S0
Q3
Q2
Q1
Q0
DATA
WR
CS
OUT
OUT3
OUT2
OUT1
OUT0

* to select words we use decoders (in linear way it is too complex)
↳ 2d decoding



error correction

↳ we need to detect whether there is an error : parity bit, etc..

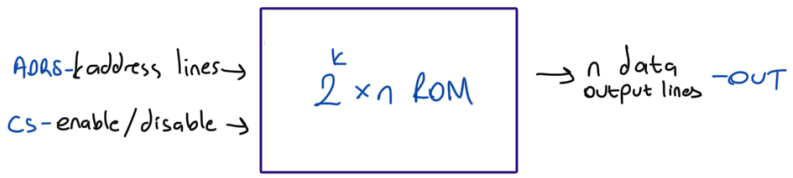
SDRAM = synchronous DRAM → common now (memory chips = modules)
DDR-RAM = double data rate RAM → newer type (faster, use less power → notebooks, phones)
↳ DDR2 → DDR3 → DDR4 → DDR5 (released in 2020)

programmable logic devices (PLD's)

↳ reconfigurable IC (integrated circuit), built with many gates
types: - programmable ROM
- programmable logic array (PLA)
- programmable array logic (PAL)
- FPGA

ROM

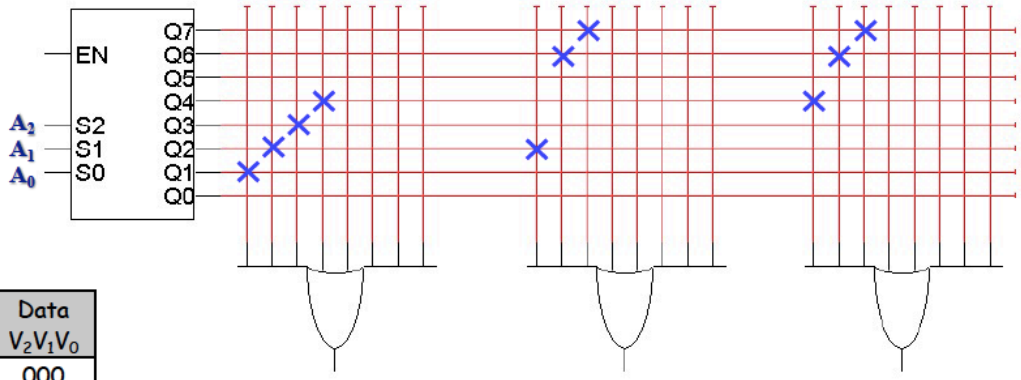
- contents cannot be easily modified
- computers use roms to store the system BIOS
- cell-phones, vending (automats) machines, game machines etc. contain rom.



* they are actually combinational devices (not sequential)
↳ rom table is just a truth table, implemented using decoders, and or gates

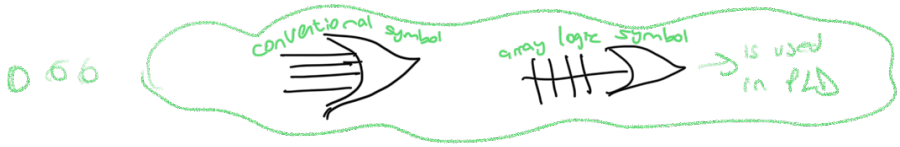
how it is programmable = implementing different functions between the connections between the decoder and the or gates

rom example =

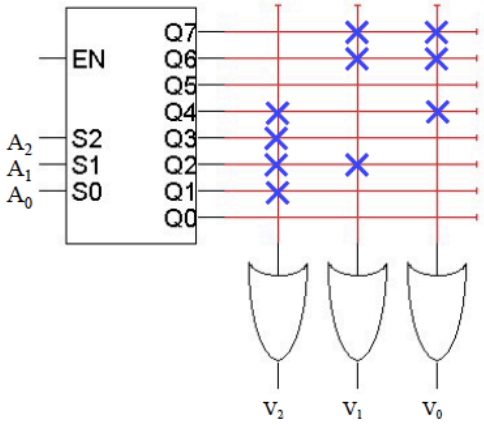


Address $A_2A_1A_0$	Data $V_2V_1V_0$
000	000
001	100
010	110
011	100
100	101
101	000
110	011
111	011

$V_2 = \Sigma m(1,2,3,4)$ $V_1 = \Sigma m(2,6,7)$ $V_0 = \Sigma m(4,6,7)$



$V_2 = \Sigma m(1,2,3,4)$
$V_1 = \Sigma m(2,6,7)$
$V_0 = \Sigma m(4,6,7)$



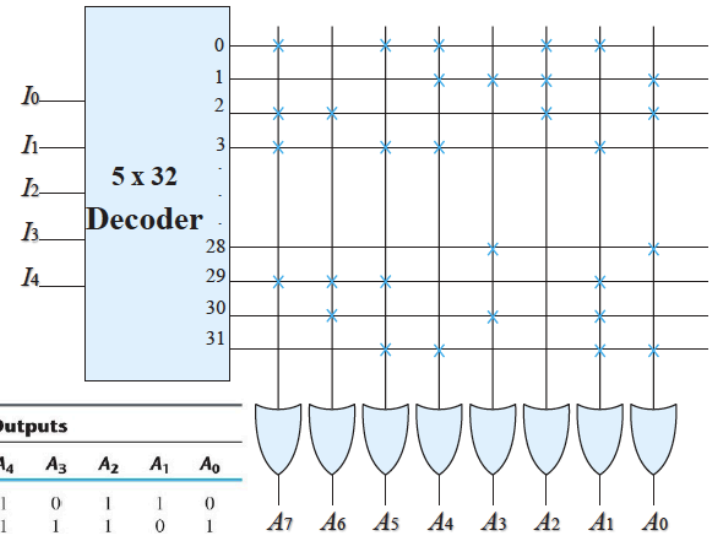


Table 7.3
ROM Truth Table (Partial)

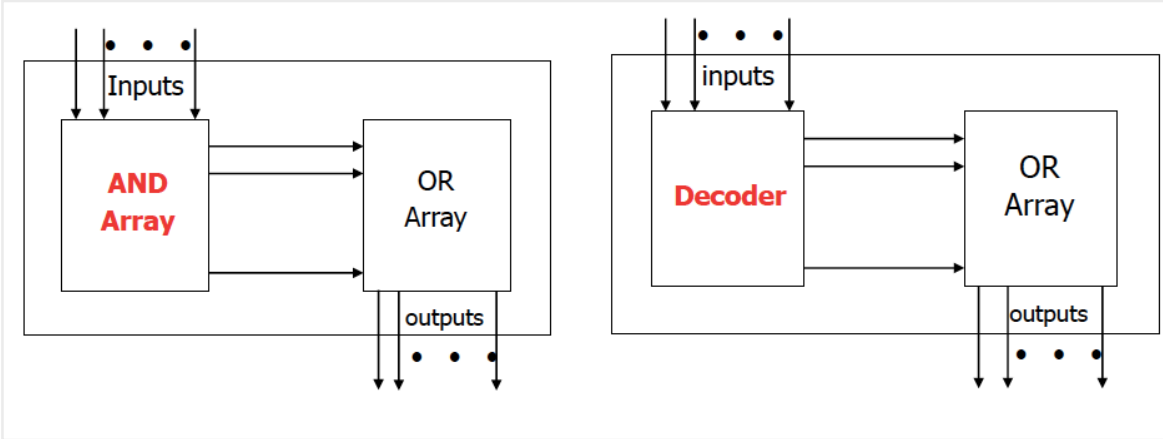
Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
⋮								⋮				
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

ROMs vs RAMs

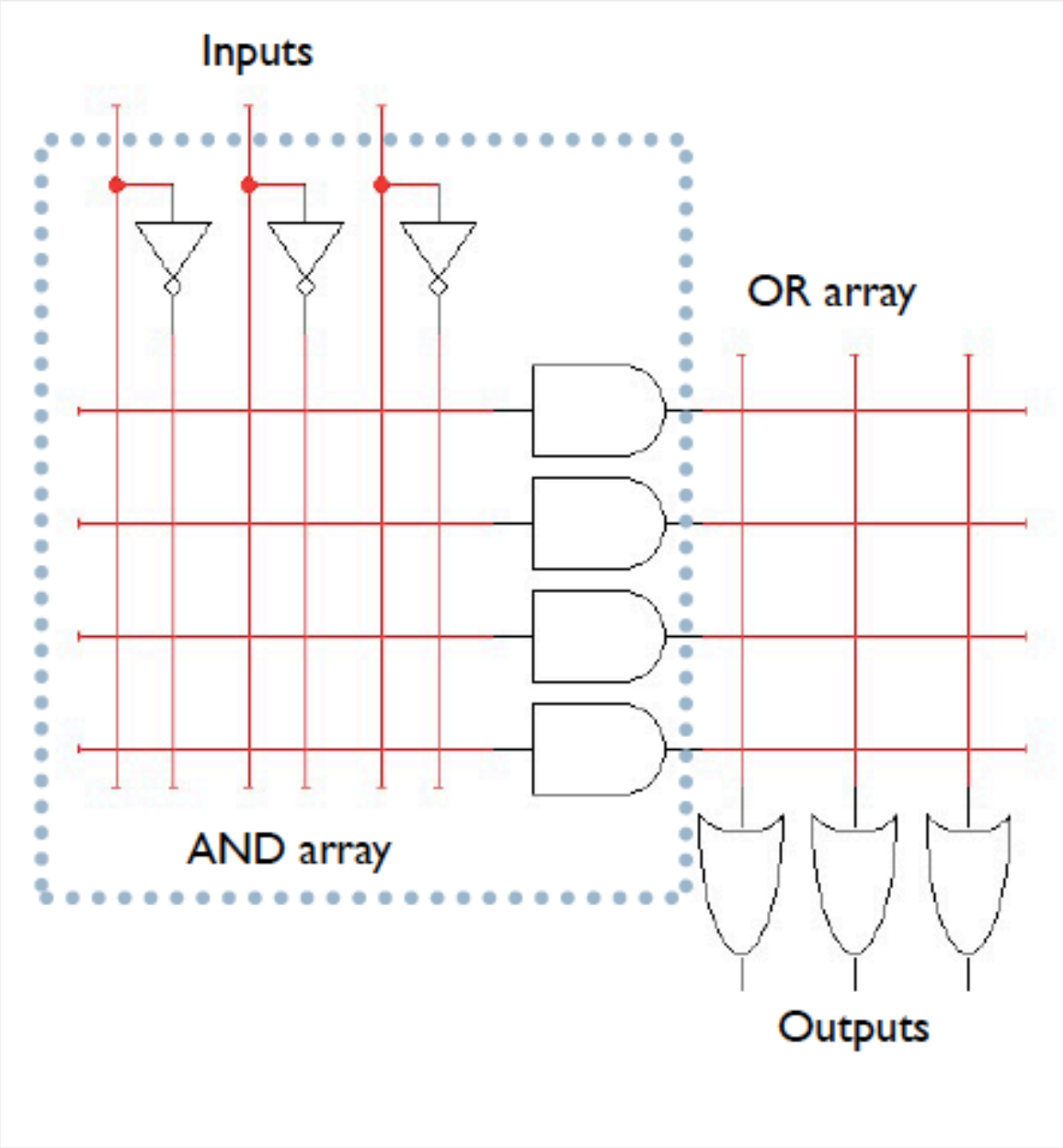
roms are "non-volatile" = data is preserved even without power
rams contents disappear once power is lost

Programmable Logic Arrays

- a rom is potentially inefficient because it uses a decoder → generates all possible minterms (no minimizatio
- PLA makes the decoder part of the rom "programmable" too → instead of generating all minterms, generate necessary products



3x4x3 PLA example



blank

Address $A_2A_1A_0$	Data $V_2V_1V_0$
000	000
001	100
010	110
011	100
100	101
101	000
110	011
111	011

V_2

X \ YZ	00	01	11	10
0	0	1	1	1
1	1	0	0	0

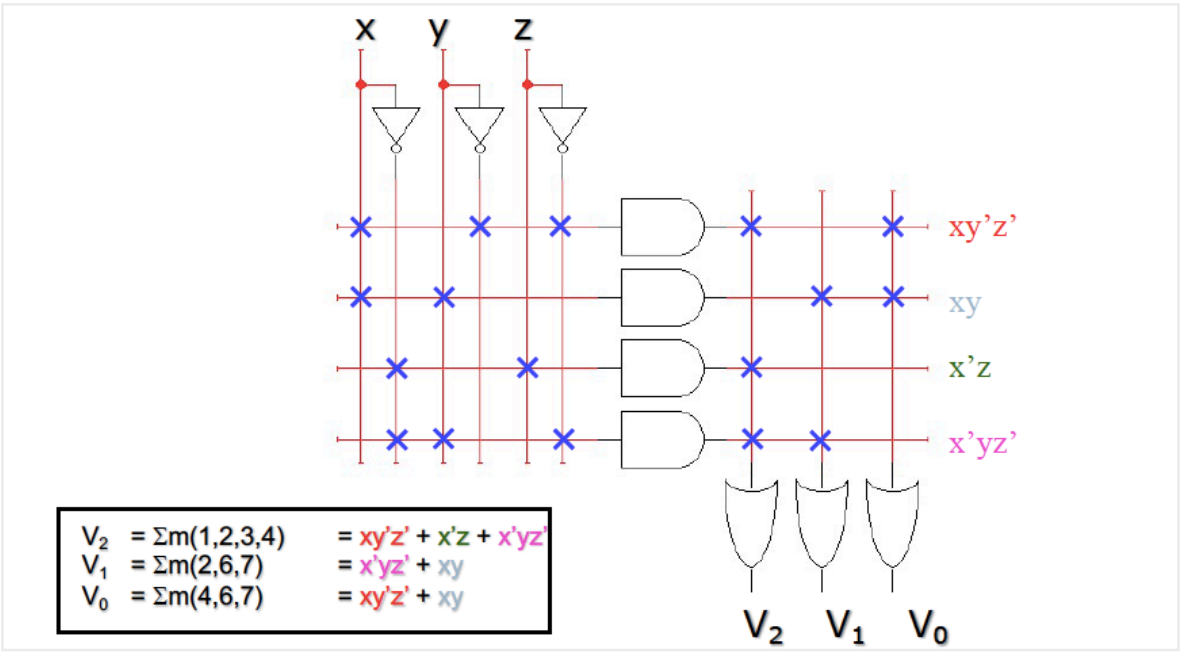
V_1

X \ YZ	00	01	11	10
0	0	0	0	1
1	0	0	1	1

V_0

X \ YZ	00	01	11	10
0	0	0	0	0
1	1	0	1	1

$V_2 = \Sigma m(1,2,3,4)$
$V_1 = \Sigma m(2,6,7)$
$V_0 = \Sigma m(4,6,7)$



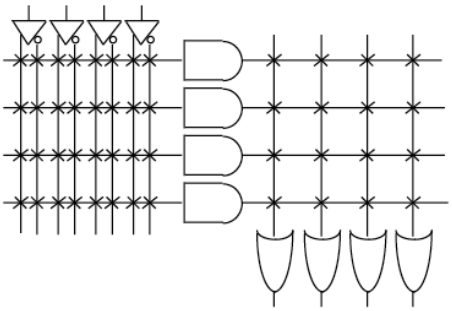
$V_2 = \Sigma m(1,2,3,4) = xy'z' + x'z + x'yz'$
$V_1 = \Sigma m(2,6,7) = x'yz' + xy$
$V_0 = \Sigma m(4,6,7) = xy'z' + xy$

another example

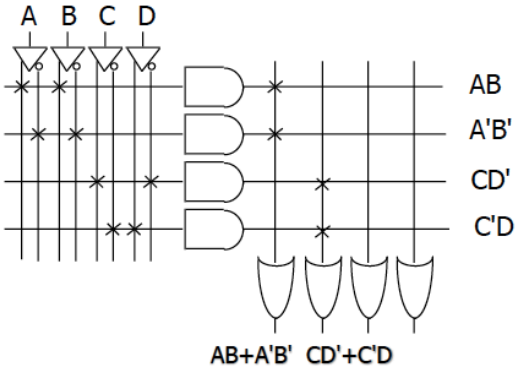
Implement the following using a 4x4x4 PLA:

$F_0 = A\,B + A'\,B'$
$F_1 = C\,D' + C'\,D$

Before programming:

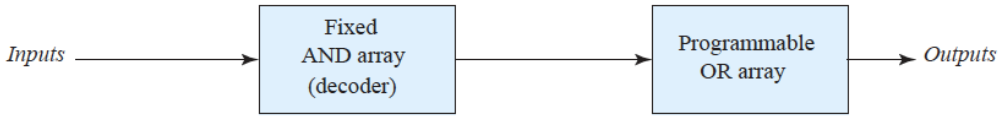


After programming:

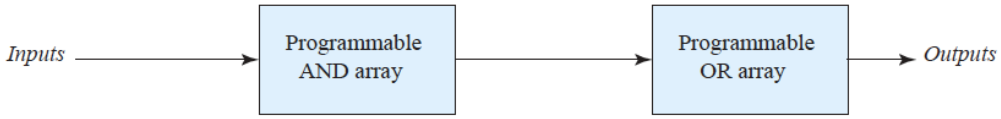


k x m x n n→ functions k→ inputs m→ expressible max m product terms

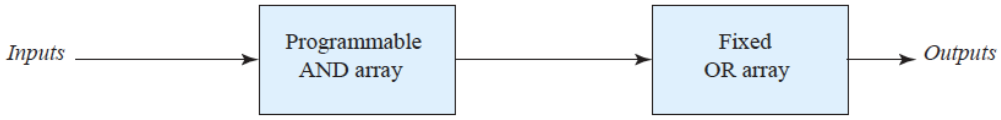
Programmable Logic Devices (PLDs)



(a) Programmable read-only memory (PROM)

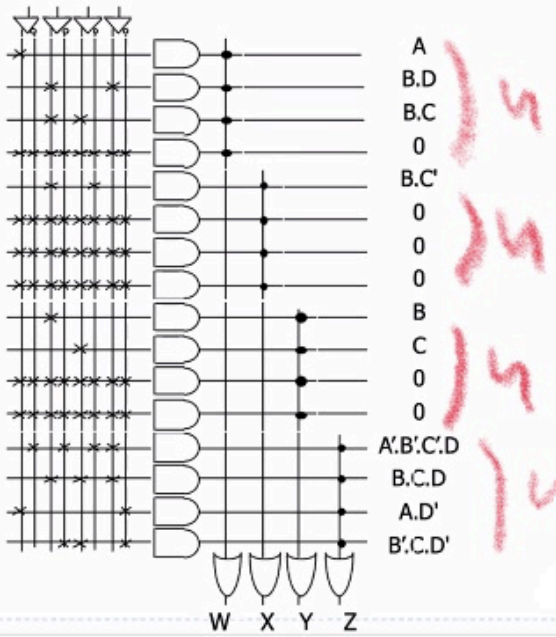


(c) Programmable logic array (PLA)



(b) Programmable array logic (PAL)

Code converter: programmed PAL



Minimized Functions:

W = A + B.D + B.C
X = B.C'
Y = B + C
Z = A'.B'.C'.D + B.C.D + A.D' + B'.C.D'

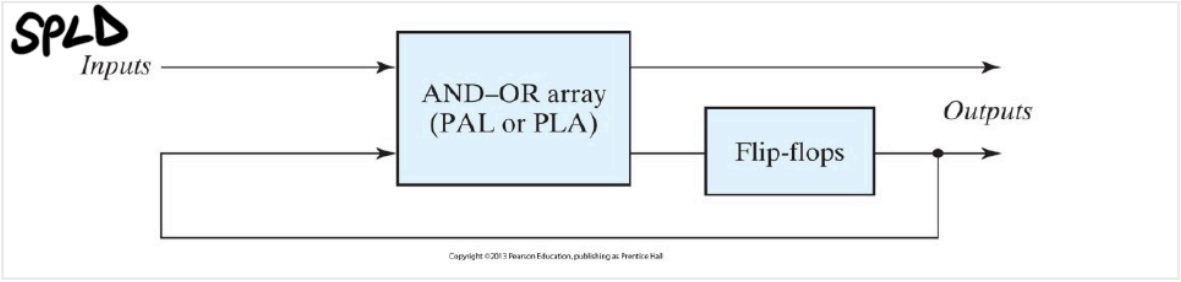
4 or gate inputs

in PAL you restrict the used or gates

Sequential programmable devices → programmable combinational logic + memory

1. SPLD (sequential / simple programmable logic devices)
2. CPLD (complex programmable logic devices)
3. FPGA (field-programmable gate array)

SPLD



CPLD