UTM
UNIVERSITI TEKNOLOGI MALAYSIA

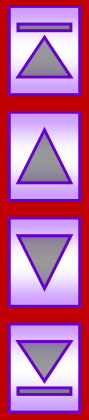
SKEE1223: Digital Electronics

14 – Programmable Logic Devices

Dr Michael Tan Loong Peng
PhD (Cambridge)
Senior Lecturer
Faculty of Electrical Engineering
Universiti Teknologi Malaysia

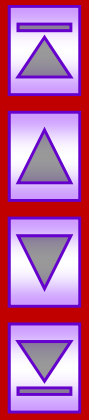
Lecture 14: Combinational Circuits Programmable Logic Devices

- Programmable Logic Devices
- PN Diode Operation
  - AND Logic Arrays
  - OR Logic Arrays
  - Two-level AND-OR Arrays
- Programmable Logic Array (PLA)
- Realising Logic Functions with PLAs



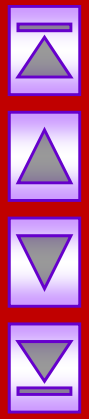
Lecture 14
Programmable Logic Devices

- Read-Only Memory (ROM)
- Programmable Read-Only Memory (PROM)
- Realising Logic Functions with PROMs
- Progammable Array Logic (PAL)
- Realising Logic Functions with PALs

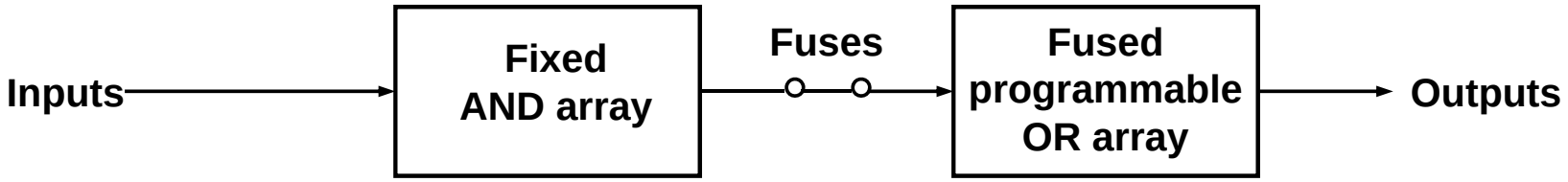


Programmable Logic Devices

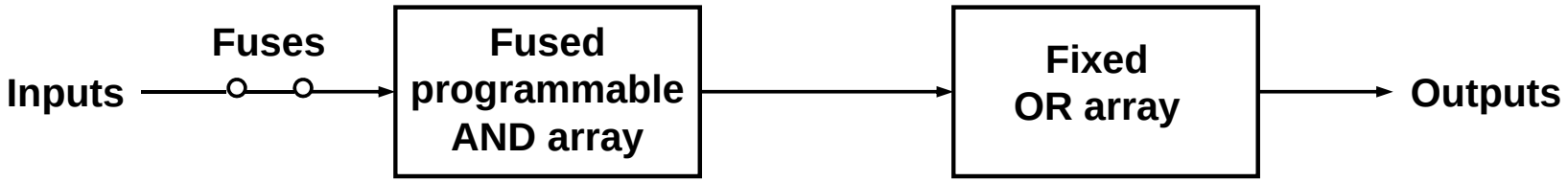
- Programmable Logic Devices (PLDs) are IC chips with internal logic gates connected by electronic fuses.
- These fuses can be ‘blown’ (by programming) to obtain different circuit configurations.
- Semi-customized chips that give high packing density at reasonable cost.
- Three classes of PLDs are :
  - Programmable Logic Array (PLA)
  - Programmable Read Only Memory (PROM)
  - Programmable Array Logic (PAL)



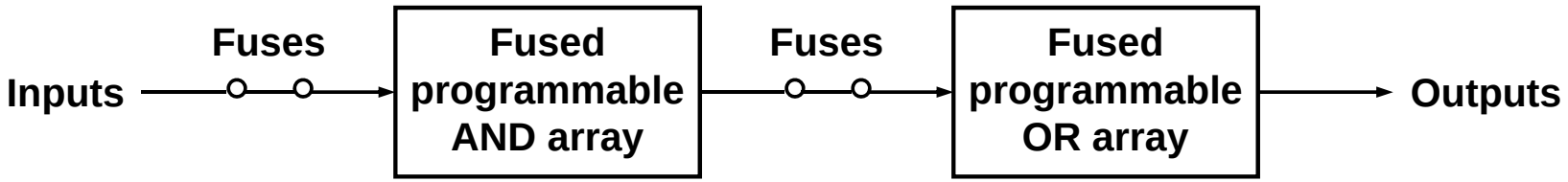
Programmable Logic Devices



Programmable Read Only Memory (PROM)



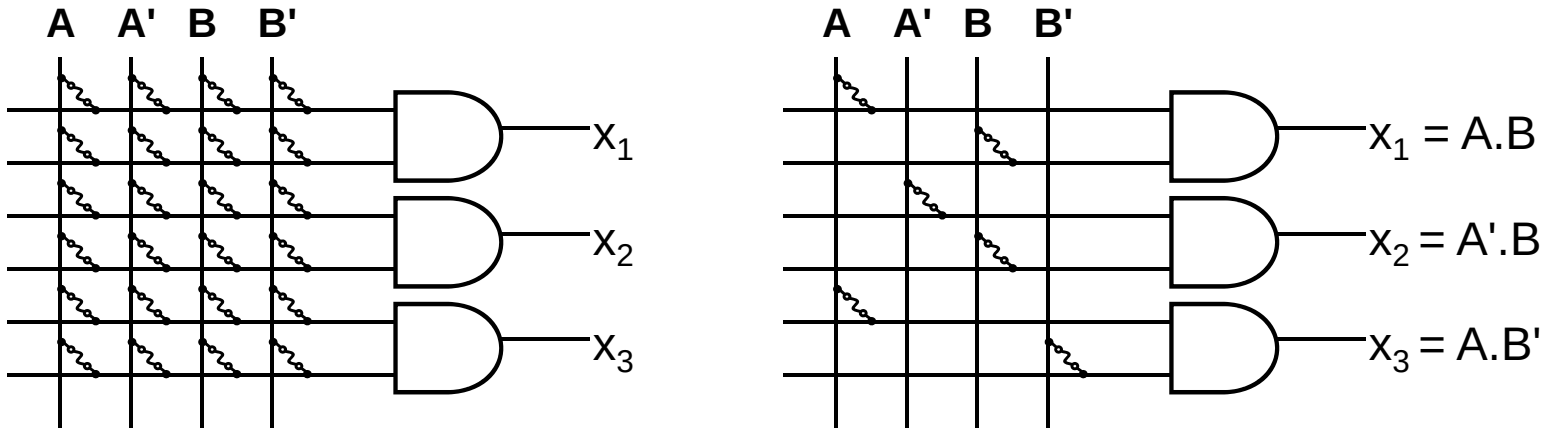
Programmable Array Logic (PAL)



Programmable Logic Array (PLA)

Programmable Logic Devices

- "Programming" an array – blowing the fuses.

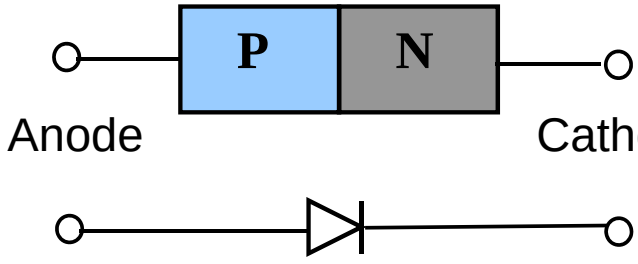


(a) Unprogrammed (b) Programmed

Example of a basic AND array

Programmable Logic Devices

- PLDs use diodes. A *PN diode* is an electronic device formed by creating a junction of two types of semi-conductor materials, *p* type and *n* type.
  - *Forward-biased*: When *p* side (anode) is more positive than *n* side (cathode), it behaves as a closed switch.
  - *Reverse-biased*: When cathode is more positive than anode, it behaves as an open circuit.

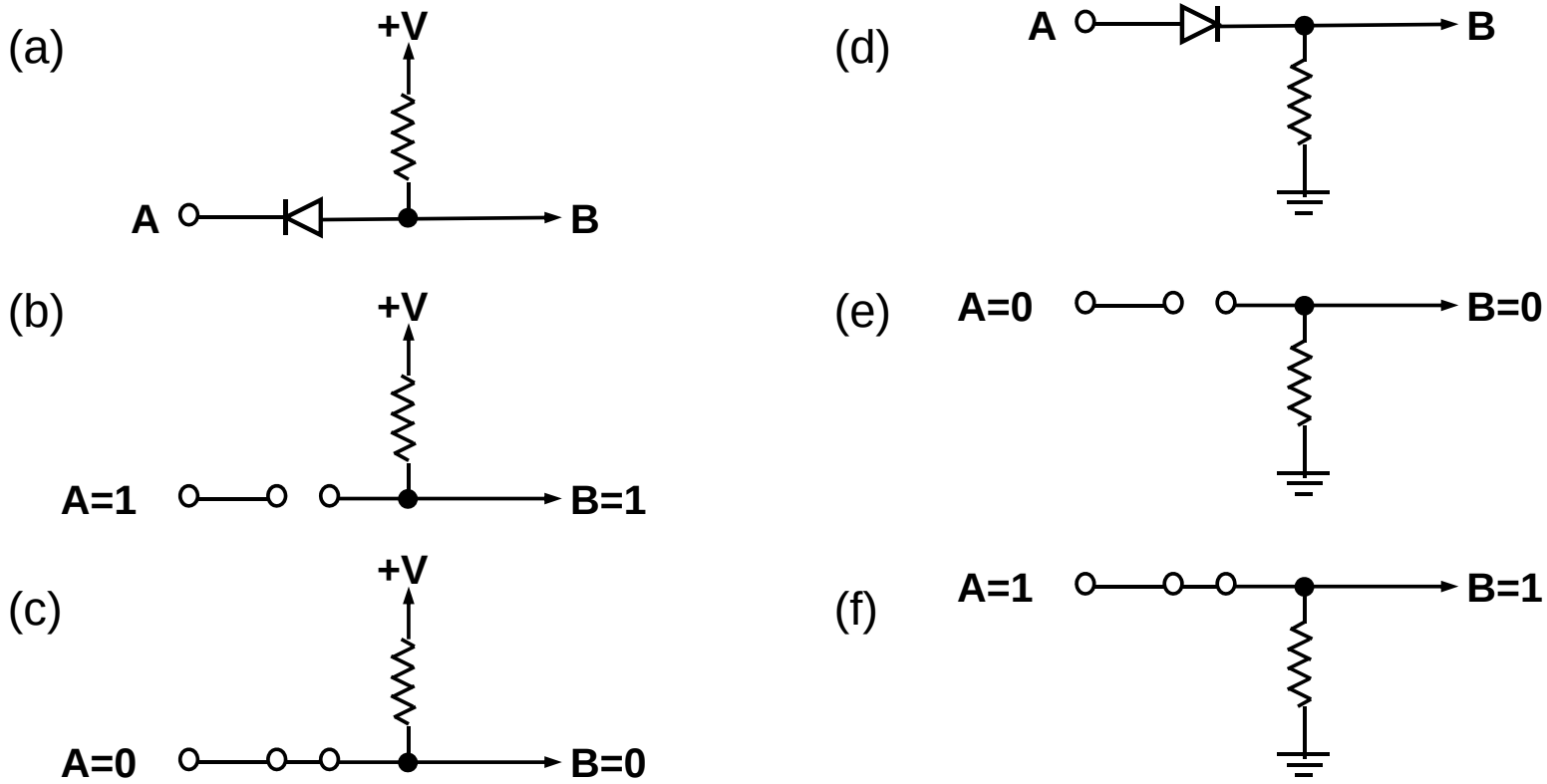


PN junction diode and schematic symbol.

Forward-biased (closed circuit)

Reverse-biased (open circuit)

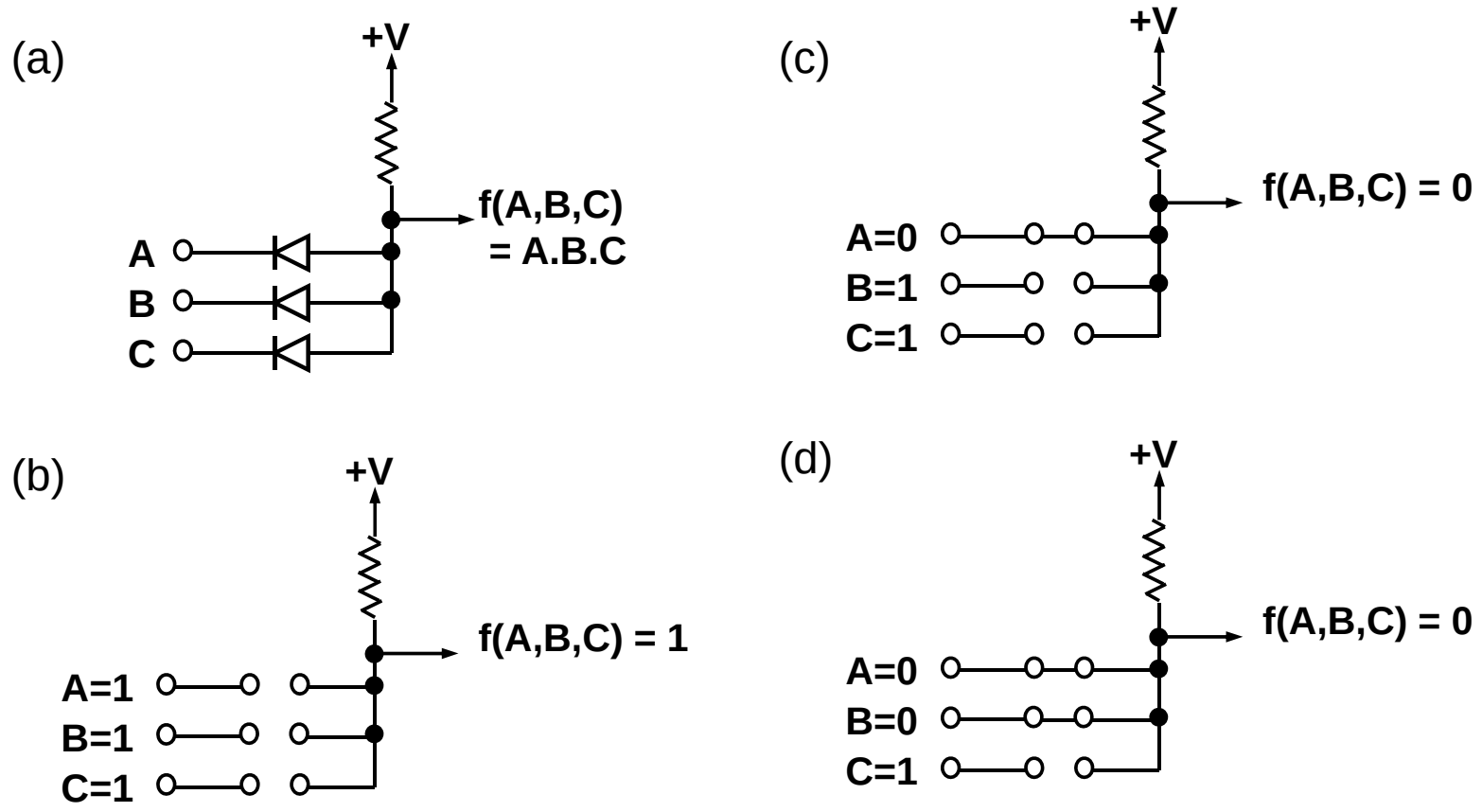
PN Diode Operation



PN diode operation for digital applications.

(a) With pull-up resistor.
(b) Reverse-biased: diode open; B pulled up to 1.
(c) Forward-biased: diode shorted, forcing B to 0.
(d) With pull-down resistor.
(e) Reverse-biased: diode open; B pulled down to 0.
(f) Forward-biased: diode shorted, forcing B to 1.

AND Logic Arrays



AND function realised with a diode array.

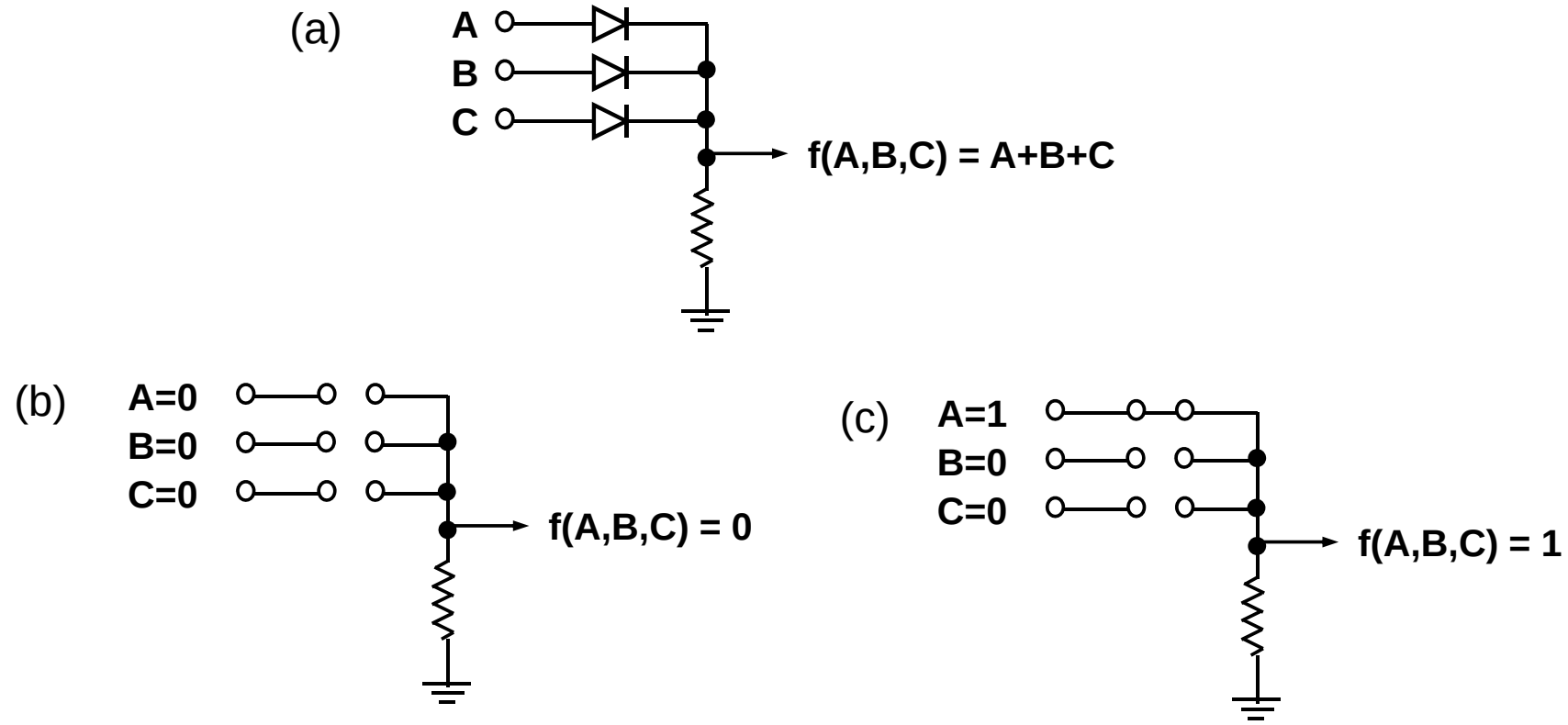
(a) Basic configuration.

(b) All diodes open; f pulled up to 1.

(c) One diode shorted, forcing f to 0.

(d) Multiple diodes shorted, forcing f to 0.

OR Logic Arrays



OR function realised with a diode array.

(a) Basic configuration.

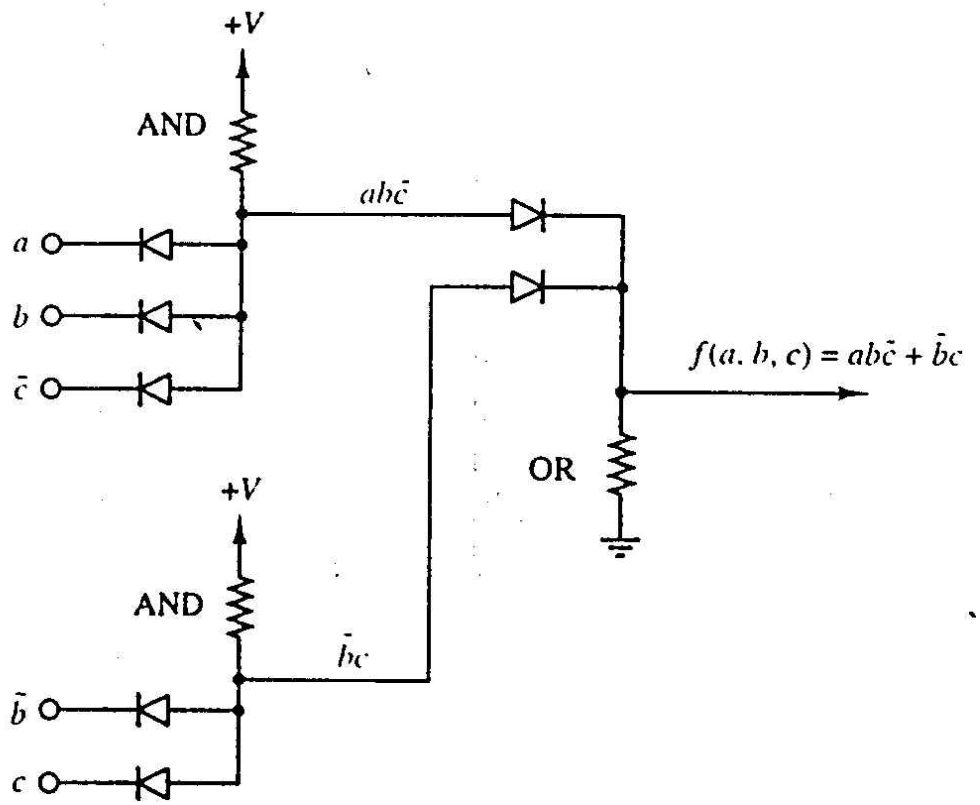
(b) All diodes open; f pulled up to 0.

(c) One diode shorted, forcing f to 1.

Two-level AND-OR Arrays

- AND and OR circuits can be interconnected to realise any arbitrary switching function.

Example: f(a,b,c)=a.b.c'+b'.c



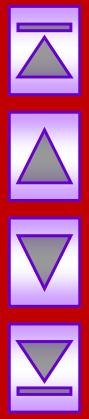
Programmable Logic Array (PLA)

- Combination of a programmable AND array followed by a programmable OR array.
- Example: Design a PLA to realise the following three logic functions and show the internal connections.

$f_1(A,B,C,D,E) = A'.B'.D' + B'.C.D' + A'.B.C.D.E'$

$f_2(A,B,C,D,E) = A'.B.E + B'.C.D'.E$

$f_3(A,B,C,D,E) = A'.B'.D' + B'.C'.D'.E + A'.B.C.D$

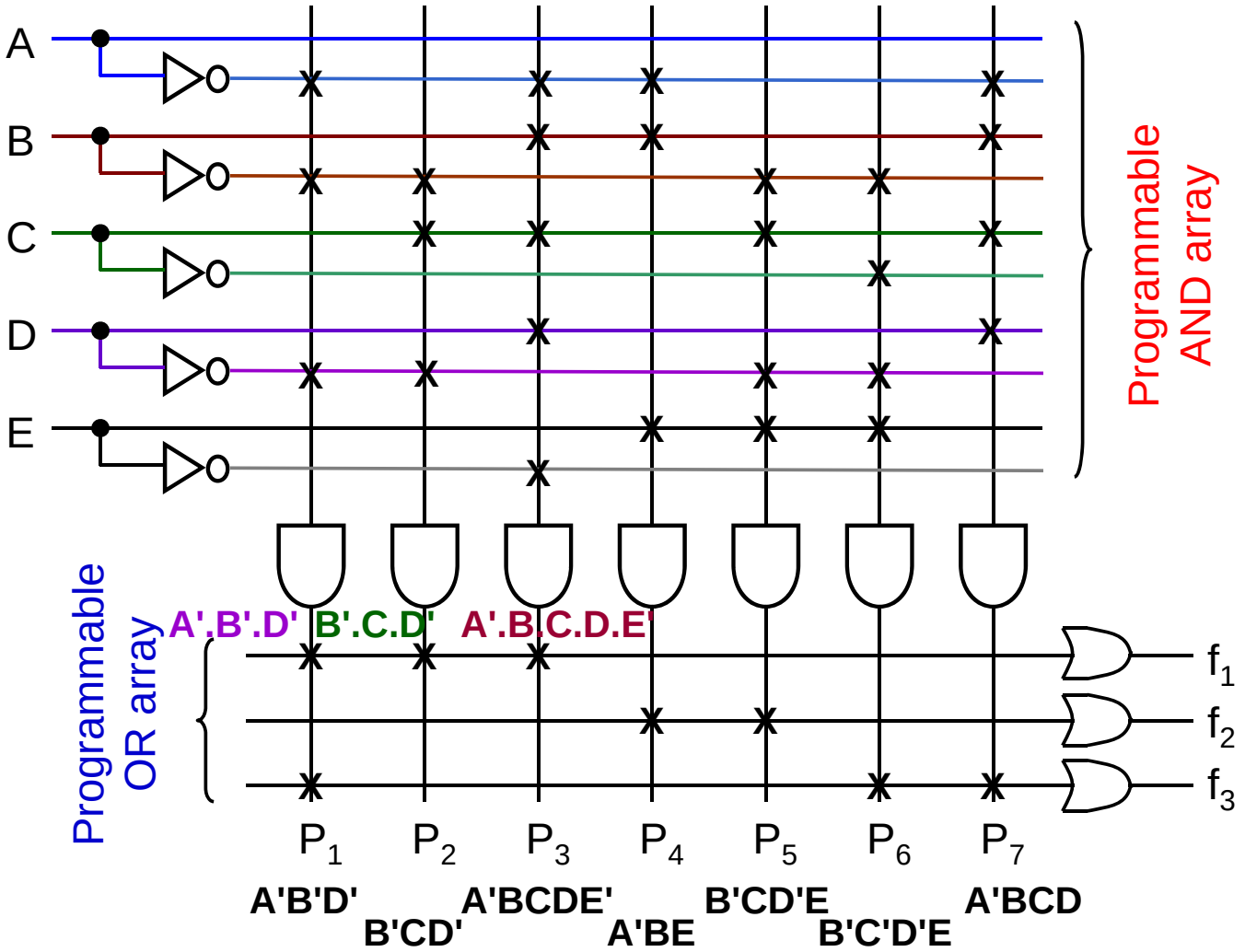


Realising Logic Functions with PLAs

$f_1(A,B,C,D,E) = A'.B'.D' + B'.C.D' + A'.B.C.D.E'$

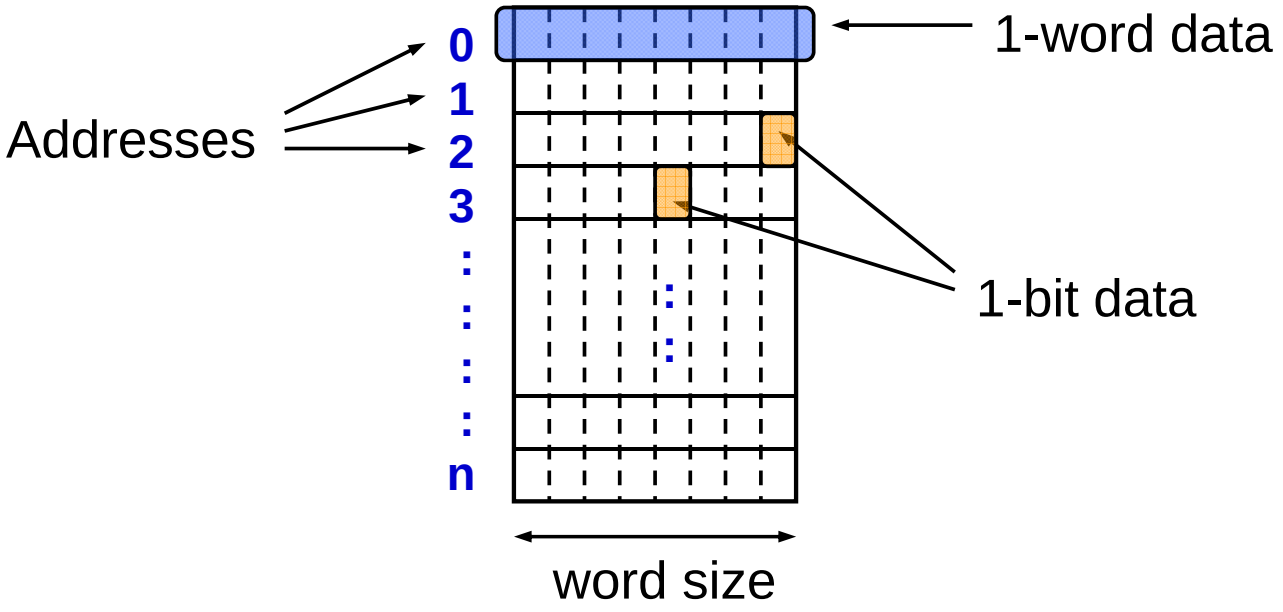
$f_2(A,B,C,D,E) = A'.B.E + B'.C.D'.E$

$f_3(A,B,C,D,E) = A'.B'.D' + B'.C'.D'.E + A'.B.C.D$



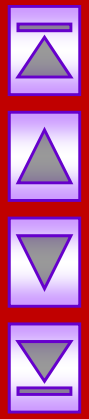
Read-Only Memory (ROM)

- A semi-conductor memory is a device where data can be stored and retrieved.
- Logically, this memory device can be regarded as a table of memory cells (data).



Read-Only Memory (ROM)

- A Read-Only Memory (ROM) is a memory device where data are read from, but not written to.
- Writing is done at time of customisation, or, by special programming devices (programmable ROM).
- Any Boolean expression can be implemented using ROM. Procedure: Obtain a truth table, treat the inputs as addresses and outputs as data.
- Advantage: Boolean functions directly implemented.
- Disadvantages: Don't care conditions not used, and limited input variables (e.g. 10 inputs – 1K, 16 inputs – 64K, 20 inputs – 1M).

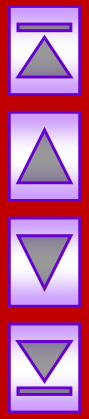


Read-Only Memory (ROM)

- Different types of ROM devices available:
  - ROM: Read-Only Memory

    Data written into memory by mask programming during manufacturing time. Expensive start-up cost but economical for high volume. Cannot be erased after data are programmed in.
  - PROM: Programmable ROM

    Semi-custom chip. Fuses can be broken by special hardware programmer unit. Cost-effective for low volumes. Cannot be erased after programming.



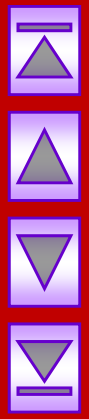
Read-Only Memory (ROM)

- EPROM: Erasable PROM

  Similar to PROM except that data can be completely erased by exposure to ultra-violet light.

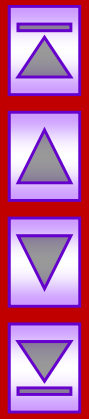
- EEPROM: Electrically Erasable PROM

  A PROM where data can be selectively erased by hardware programmer unit, rather than by ultra-violet light. Useful for remote devices which can be re-programmed from a distance.

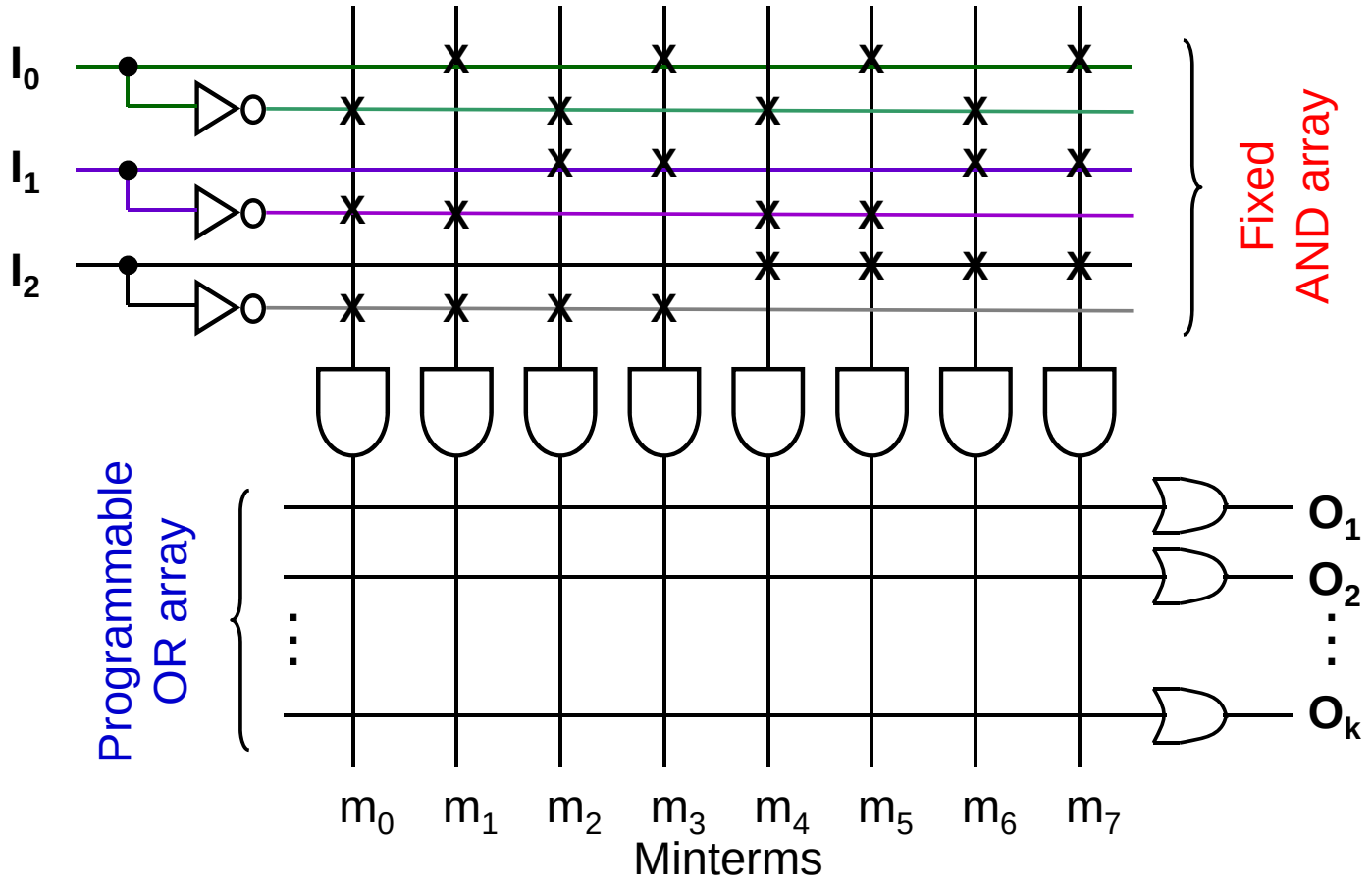


Programmable Read-Only Memory (PROM)

- Devices with fixed AND array (which is a decoder) and programmable OR array.
- The AND array (decoder) generates all 2^n possible minterm products of its n inputs (often referred to as n-to-2^n decoder).
- n input lines, m output lines.
- Bit combination of input variables – address.
- Bit combination of output lines – word (each word contains m bits).

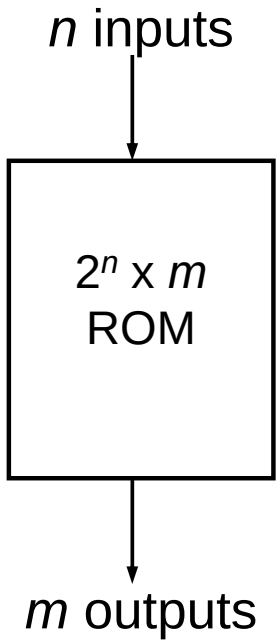


Programmable Read-Only Memory (PROM)



Programmable read-only memory (PROM) can realize K functions $f(I_2,I_1,I_0)$.

Programmable Read-Only Memory (PROM)

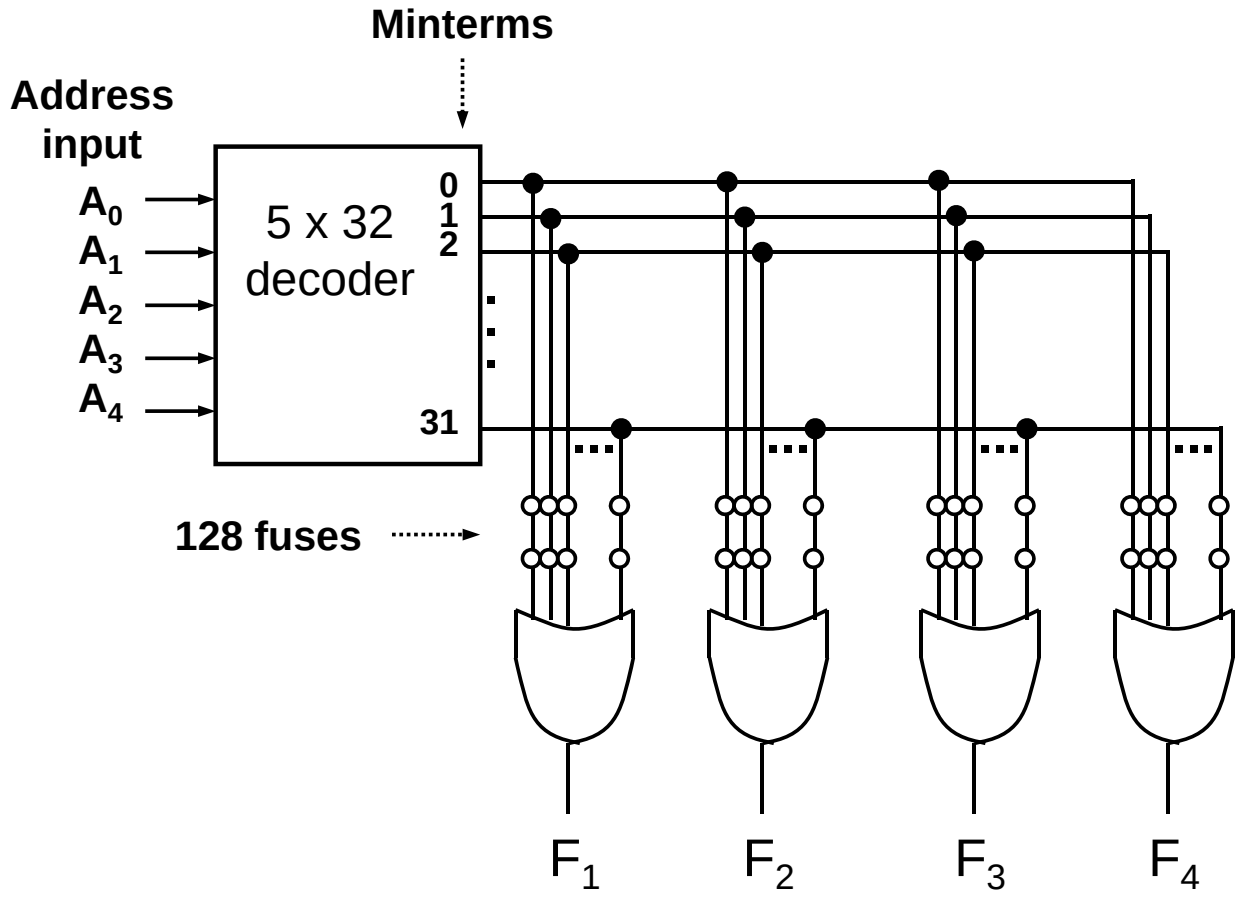


2^n x m ROM

=> 2^n words, each word m bits

=> 2^n x m bits

Programmable Read-Only Memory (PROM)



Logic construction of a 32 x 4 ROM.

Realising Logic Functions with PROMs

- Example (8 x 3 ROM):

  $f_1(A,B,C) = A.B + B'.C$

  $f_2(A,B,C) = (A+B'+C).(A'+B)$

  $f_3(A,B,C) = A + B.C$

- First, we convert each function to canonical SOP form.

  $f_1(A,B,C) = A.B + B'.C = A.B.C' + A.B.C + A'.B'.C + A.B'.C$

  $= \Sigma\, m(1,5,6,7)$

  $f_2(A,B,C) = (A+B'+C).(A'+B)$

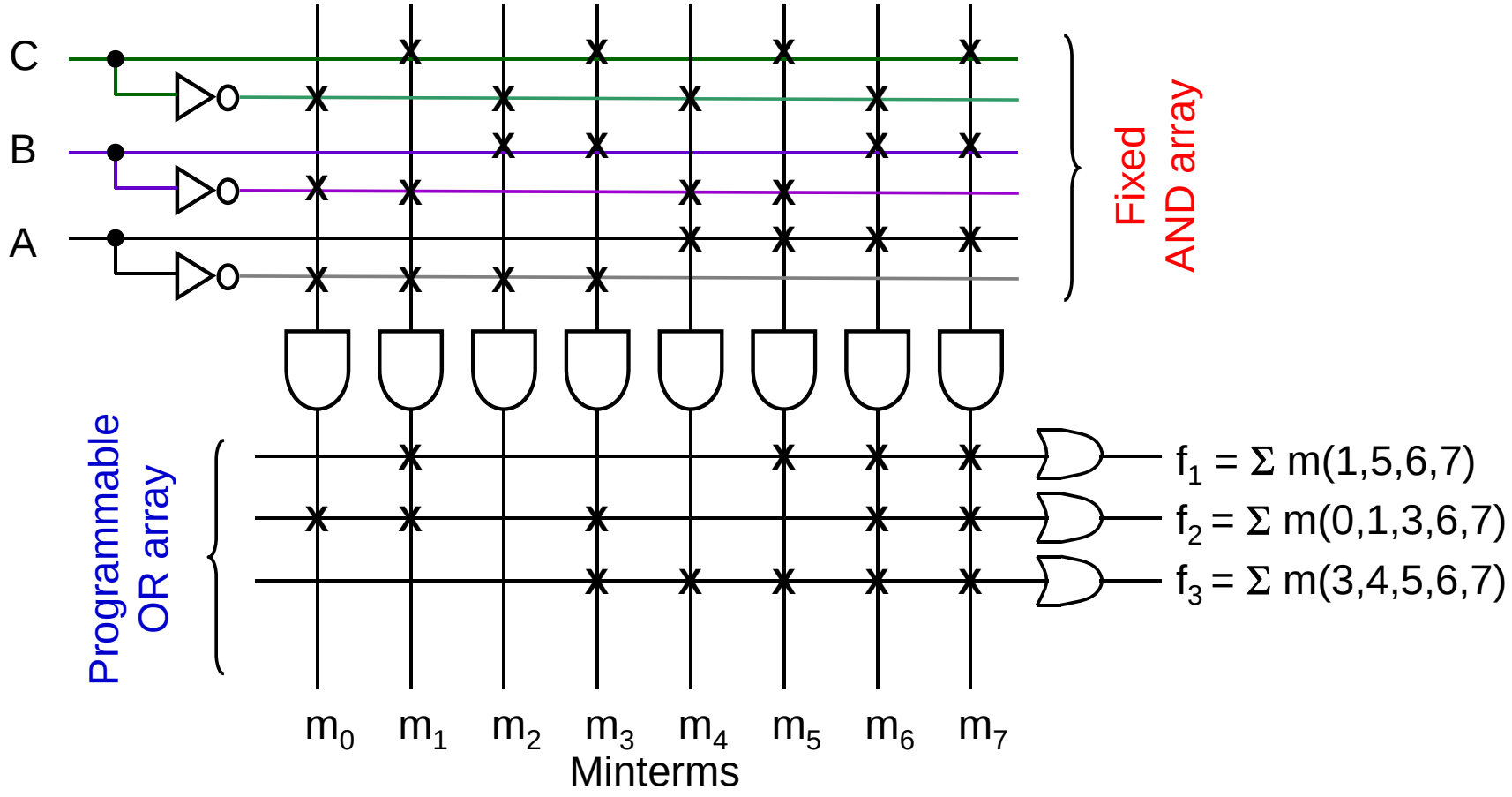
  $= (A+B'+B).(A'+B+C').(A'+B+C)$

  $= \Pi\, M(2,4,5) = \Sigma\, m(0,1,3,6,7)$

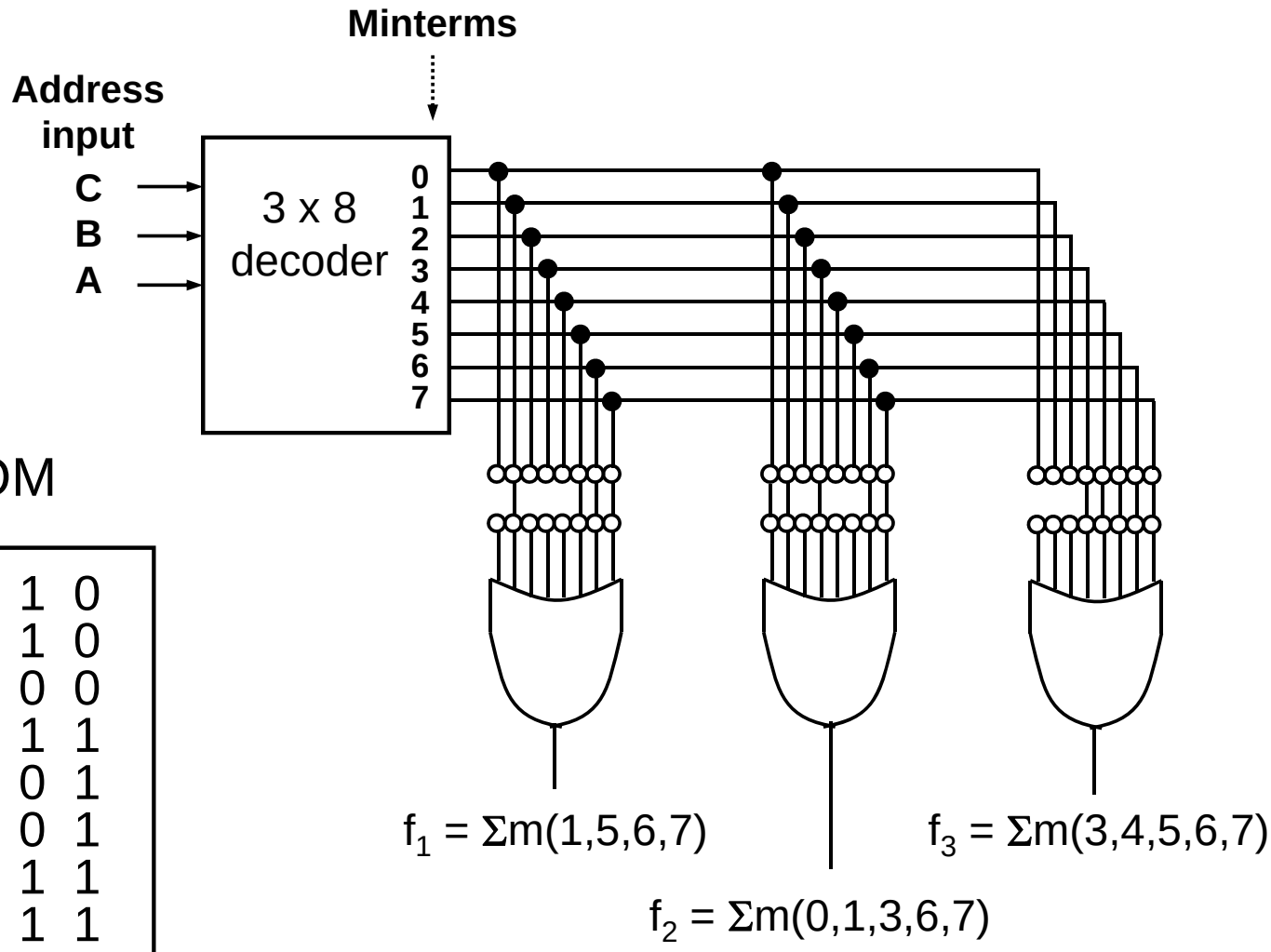
  $f_3(A,B,C) = A + B.C = A.B'.C' + A.B'.C + A.B.C' + A.B.C + A'.B.C$

  $= \Sigma\, m(3,4,5,6,7)$

Realising Logic Functions with PROMs



Realising Logic Functions with PROMs

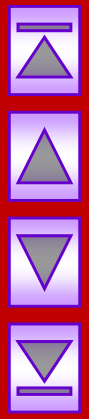


8 x 3 ROM

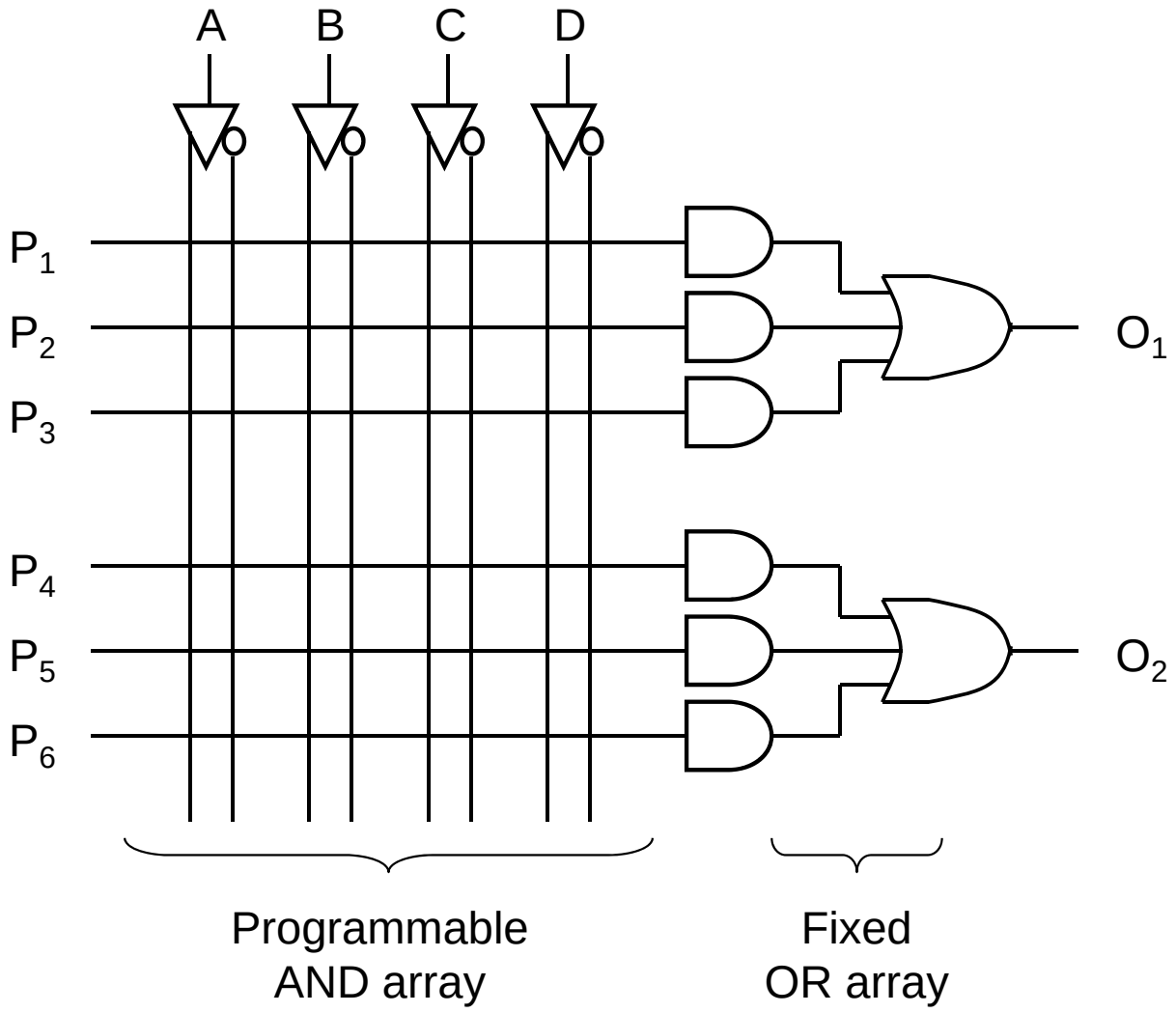
0:	0	1	0
1:	1	1	0
2:	0	0	0
3:	0	1	1
4:	0	0	1
5:	1	0	1
6:	1	1	1
7:	1	1	1

Programmable Array Logic (PAL)

- Introduced in late 1970s by Monolithic Memories Inc. as lower cost replacement for logic gates, PROMs, and PLAs.
- PAL has programmable AND array and fixed OR array.
- Less general than PLA but easier to manufacture and design.
- Product terms belong to different OR gates, cannot be shared.



Programmable Array Logic (PAL)



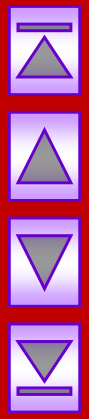
Realising Logic Functions with PALs

- Procedure is to obtain minimal SOP expressions.
- Example:

$$f_{\alpha}(A,B,C,D) = A'.B'.D' + B'.C.D' + A'.B.C.D$$

$$f_{\beta}(A,B,C,D) = A'.B + B'.C.D'$$

$$f_{\chi}(A,B,C,D) = A'.B'.D' + B'.C'.D' + A'.B.C.D$$



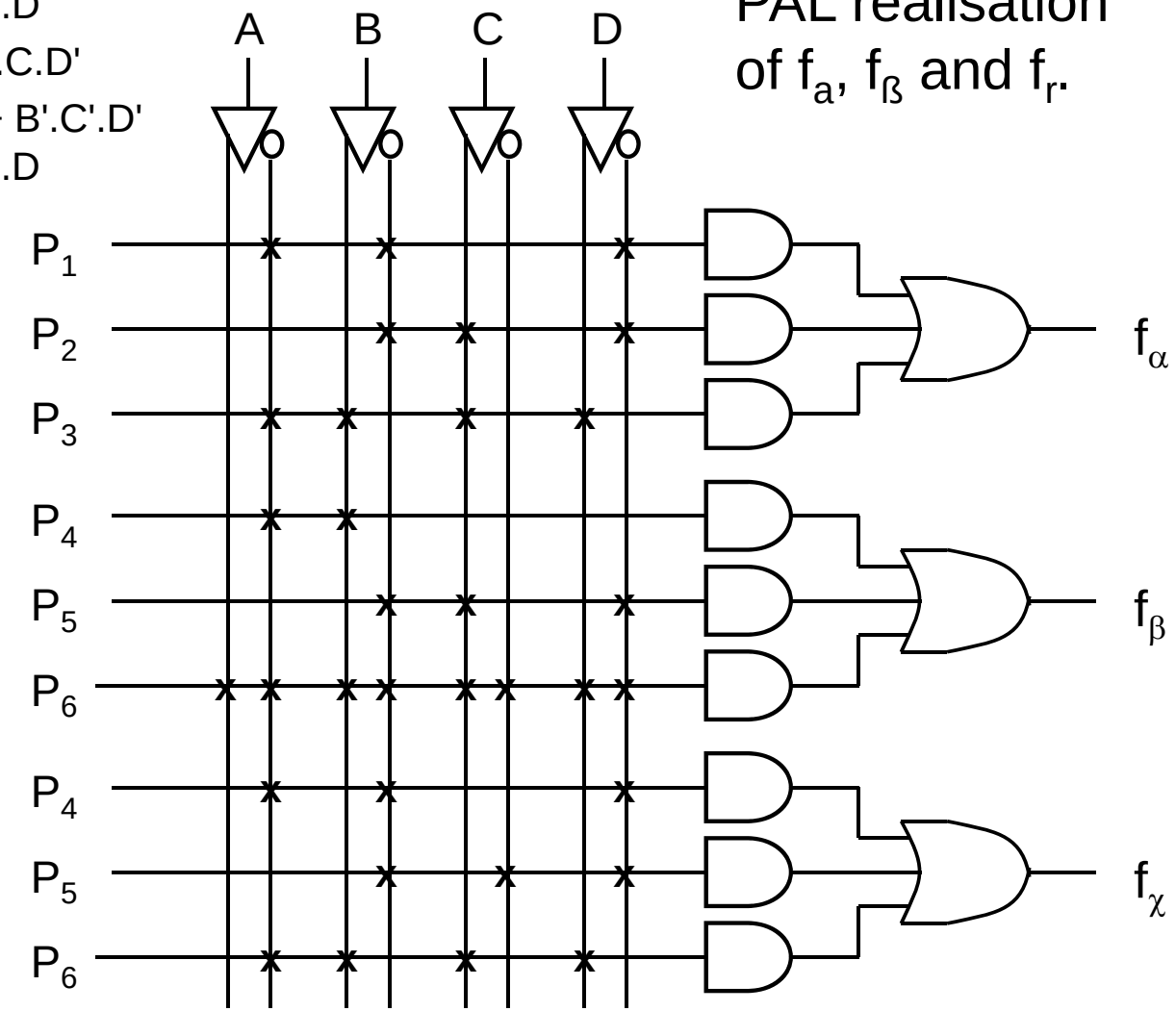
Realising Logic Functions with PALs

$f_\alpha(A,B,C,D) = A'.B'.D' + B'.C.D' + A'.B.C.D$

$f_\beta(A,B,C,D) = A'.B + B'.C.D'$

$f_\chi(A,B,C,D) = A'.B'.D' + B'.C'.D' + A'.B.C.D$

PAL realisation of f_a, $f_ß$ and f_r.



End of segment