Complex Programmable Logic Devices (CPLD)

- reference → NPTEL, Digital System Design, IISC,

* PLD → SPLD (simple)
  → CPLD

* Idea → Memory as programmable logic in 80's
  → PROM → Programmable m/m

* Eg: EXOR gate using PROM

	A1	A0	D0
X →	0	0	0
	0	1	1
Y →	1	0	1
	1	1	0

↑ address — ↑ D0 — content at the address

OR

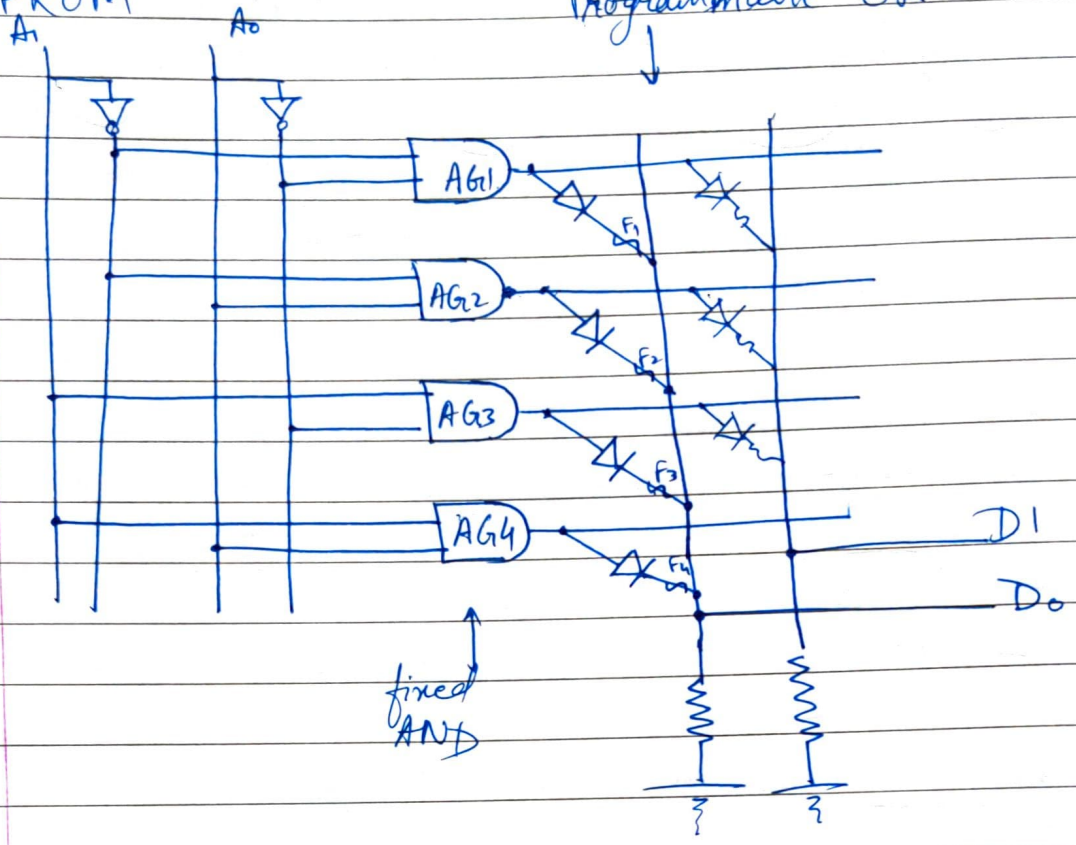
	1B each
11	0
10	1
01	1
00	0

m/m in read mode

Addr lines → i/p
Data line → o/p
Truth table → content.

NOTE: # M/m should be non-volatile so that the T.T. or the contents are not lost.

PROM



F: fuse

- blow fuses F1 & F4 (for XOR gate example). (How to blow is not necessary).
- blowing fuse = programming 0.
- when $A_1A_0 = 00$, AG1's o/p = 1 but F1 is blown so 1 does not get connected to D0 i.e. D0 is pulled down to 0 (gnd).
- when 10, D0 becomes '1'.
- If ~~i/p~~ no. of i/ps = n then no. of AND gates = 2^n

Q] What is the use of diode?

→ $A_1A_0 = 10$, o/p of AG2 = 1 & AG3 = 0, In absence of diode, AG2 = AG3 i.e we are shorting 2 lines have +Vcc & GND level!

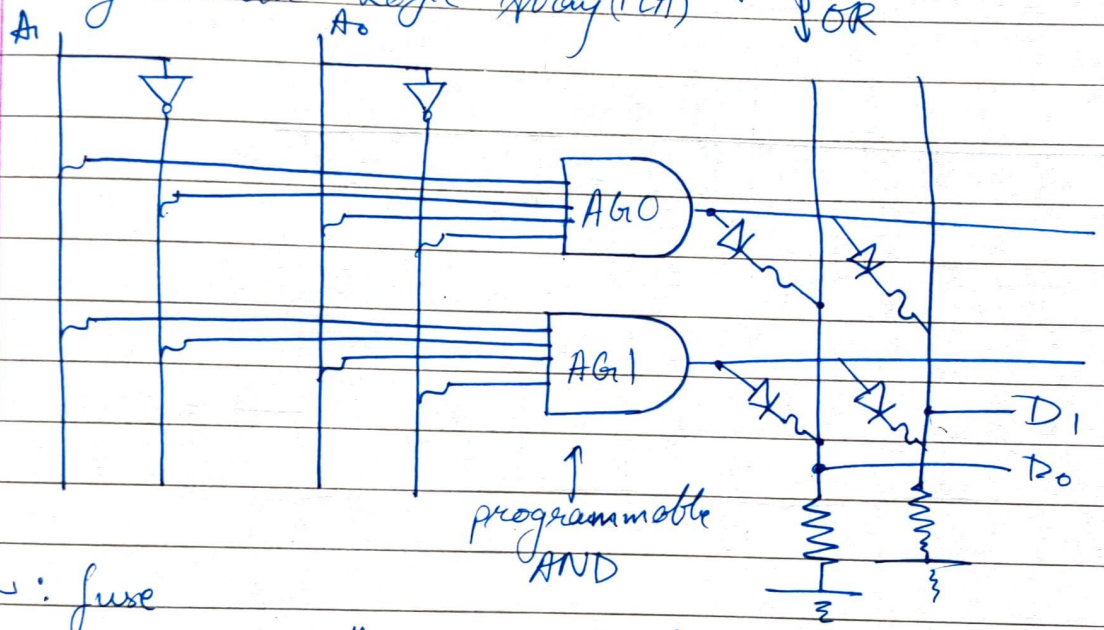
Q] How to blow the fuse?

→ Apply -ve voltage to D0, provide proper A_1A_0. So the current exceed fuse rating current then the fuse is blown.

- No. of AND gates required = $2^{(\text{no. of i/ps})}$

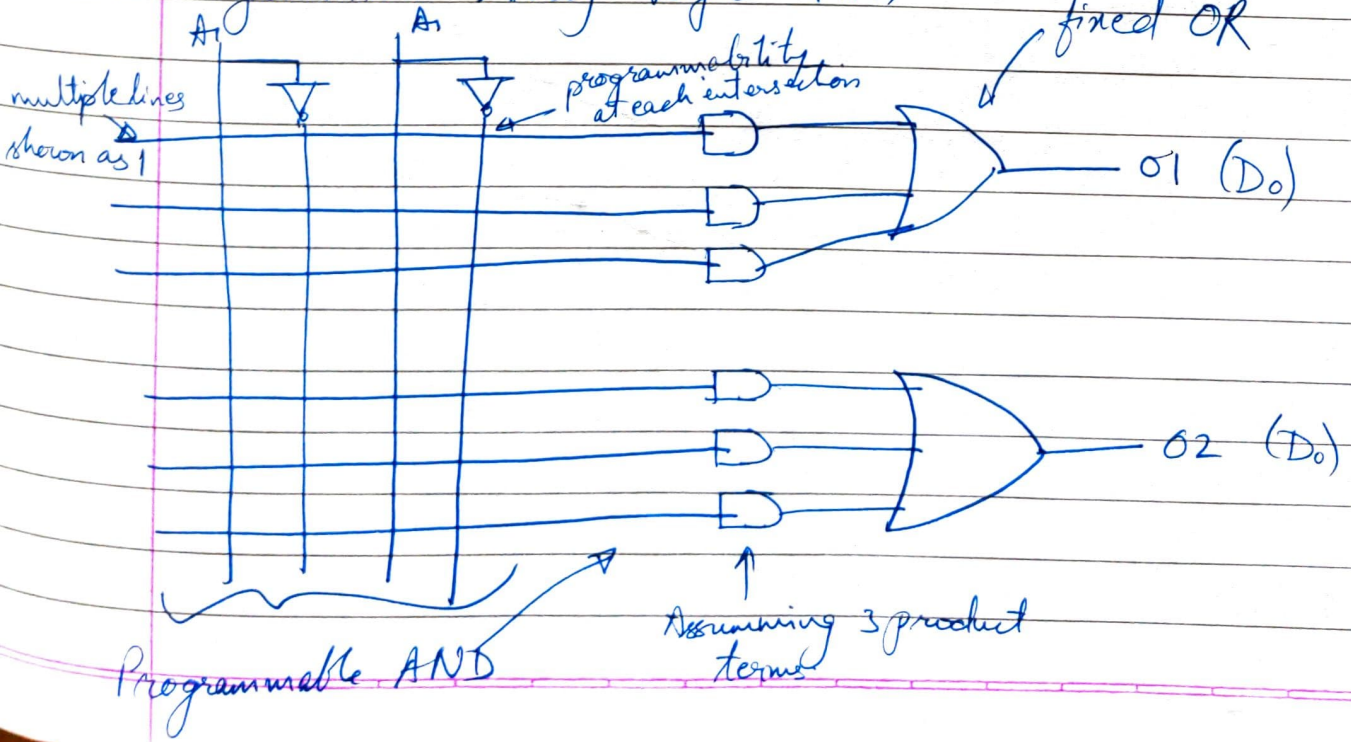
Q] Can we ~~rega~~ reduce the no. of AND gates?
→ Yes, but the AND gates should be programmable
& Minterms → Product terms
↳ This is the the beginning of PLA.

☆ Programmable Logic Array (PLA)



~ : fuse
AG0 & AG1 will give o/p = 1 for A_1A_0 = 01 or 10 ie. XOR

☆ Programmable Array Logic (PAL)



* PAL is the real architecture of PLD.

* PAL: 1) programmable AND
2) fixed OR
3) dedicated product terms for outputs.

~~PROM: fixed AND~~

	AND	OR	
PROM	fixed (F)	programmable (P)	
PLA	P	P	
PAL	P	F	→ = CPLD

* Check PAL16L8 on net for idea of PAL
→ Texas Instruments
→ This is a practical example of SPLD.

* Limitation of CPLD : ① Large no. of switches are required
② Complex design
③ Many product terms are rarely used but still a AND gate is allocated

✡ CPLD VS FPGA

Feature	PLD	FPGA
1) Logic	AND-OR	MUX / LUT / Gates
2) Register to logic ratio	small (low m/w)	large
3) Timing	simple	complex
4) Architecture variation	small	large
5) Programming technology	flash	anti-fuse, SRAM
6) Capacity	10K	few million logic cells + few MB RAM.

✡ FPGA

ASIC → Built from scratch, large cost & time, high design cost (NRE: Non recurring expenditure).

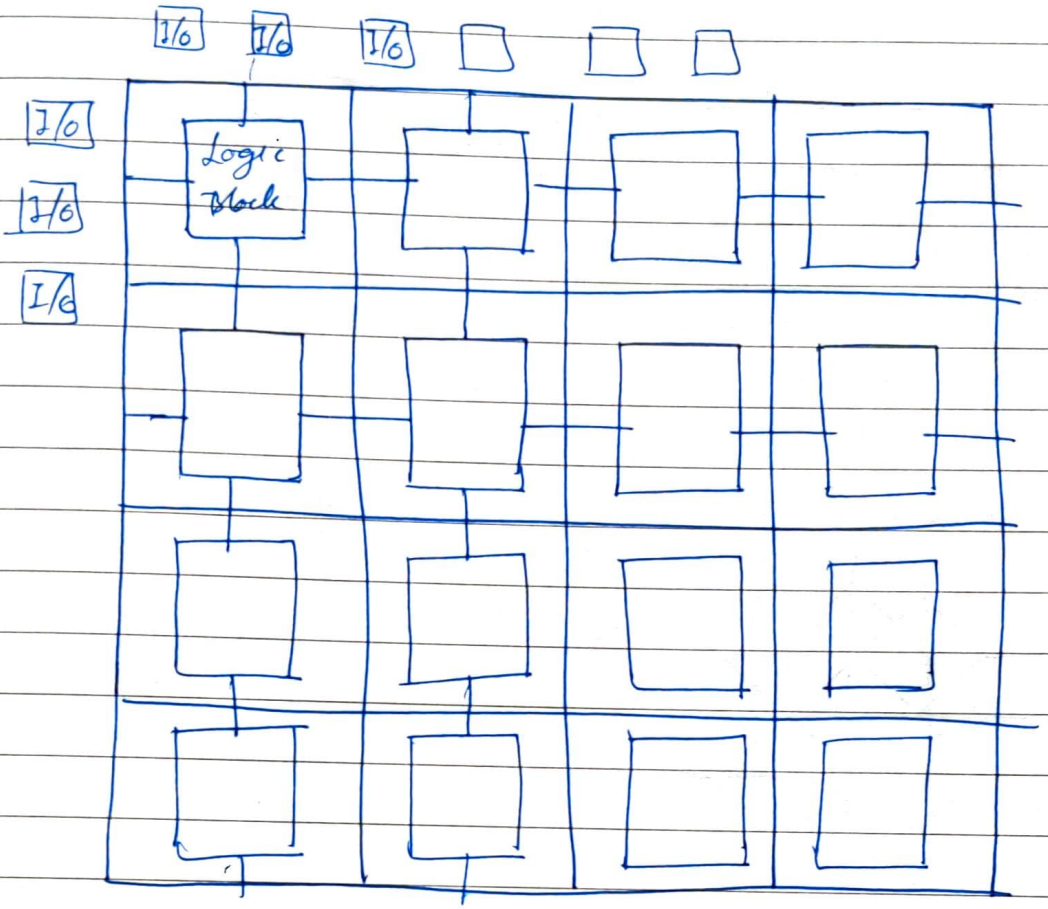
MPGA (Mask Programmable Gate Array)
- → Array of transistors
- → Interconnections are made during fabrication & cannot be altered on "FIELD".

FGPA → Programmable interconnections / on field.
- → Array of logic resources

* Logic Resources: LUT (similer to CPLD concept)
  Multiplexers
  Gates
* Programmable Interconnections: SRAM
  Flash transistor
  Anti-Fuse (opp. of a fuse) (make connect by applying voltage)
* Special Resources: PLL/DLL, RAMs, FIFOs
  ↑ Delay Locked Loop
* Memory Controllers, Network Interfaces, Processors (in complex FPGAs)

* Commercial FPGA (main manufacturers)
- Xilinx
  - spartan-3, ——— -6 : Low cost FPGA
  - Virtex-4, ——— 5, ——— 6 : complex FPGA (previous gen.)
  - Artix-7, Kintex-7, Virtex-7, Zynq: ——— current generate
- Altera
  - Cyclone, ——— II, III, IV, V : Low cost
  - Arria II, V : } Complex versions
  - Stratix II, III, IV, V : }

✡ Structure of a FPGA



* Pins: Vcc, GND, I/O
* Logic block: Small logic & flipflops
* Lot of interconnections b/w logic blocks.
* Interconnections delay may be > logic computation delay. but large interconnections provide flexibility.
* Interconnections: Switch Blocks