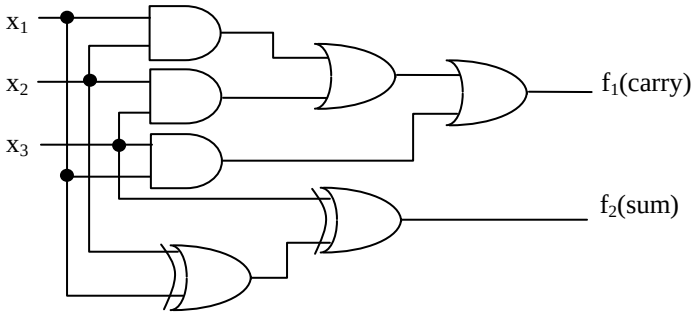
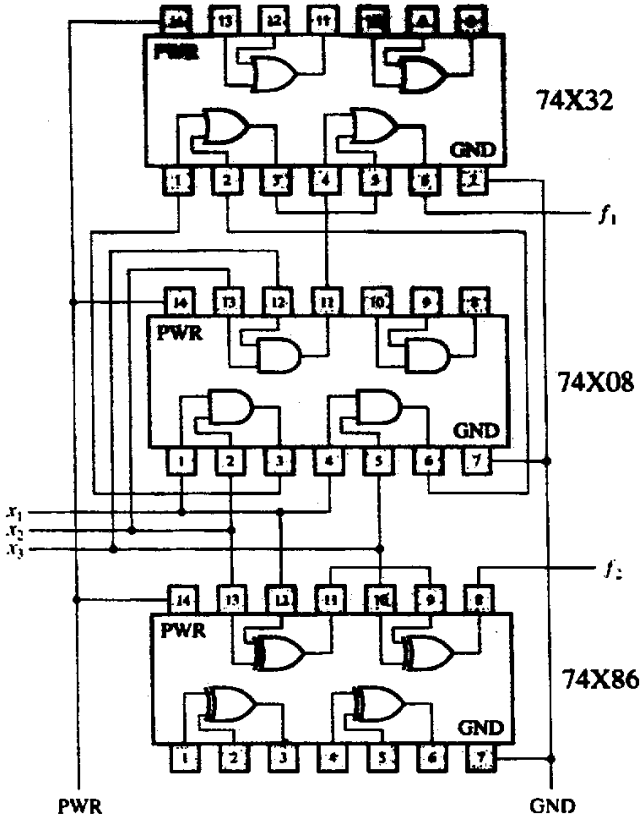
Programmable Logic Devices

Motivation for *Programmable Logic Devices*

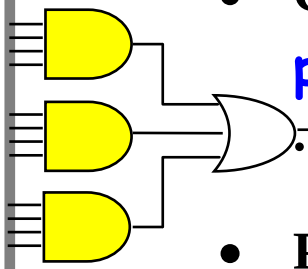
- **Complex digital circuits require many such combinational circuits requiring many such SSI/MSI components**

- **Modern circuits rarely use SSI devices to implement logic systems.**





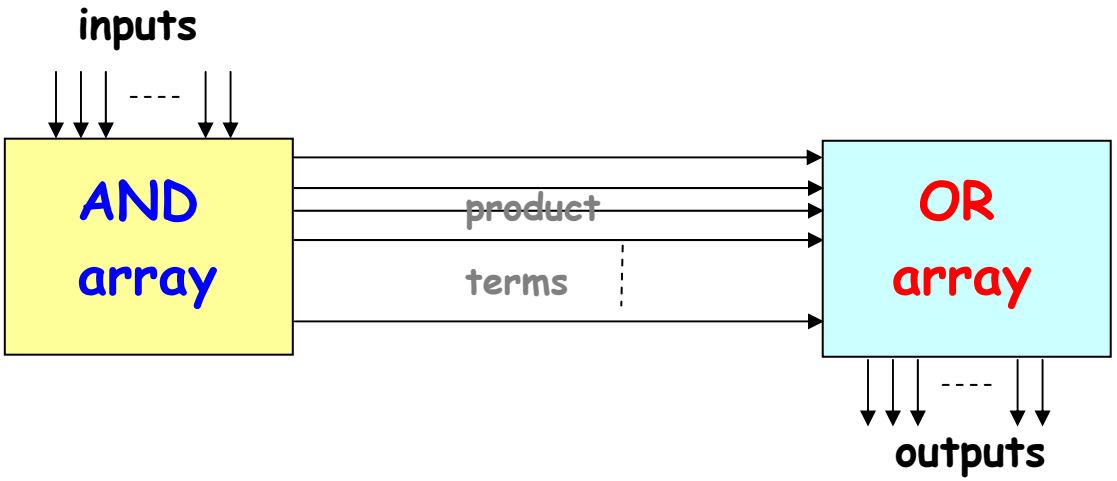
Programmable Logic Arrays



- **Combinational circuits are realized as two-level sum of products (SOP), AND-OR-INVERT structures such as**

  $Z = A\bar{B}CD + AB\bar{C}D + \ldots$

- **Programmable array block diagram for SOP form:**

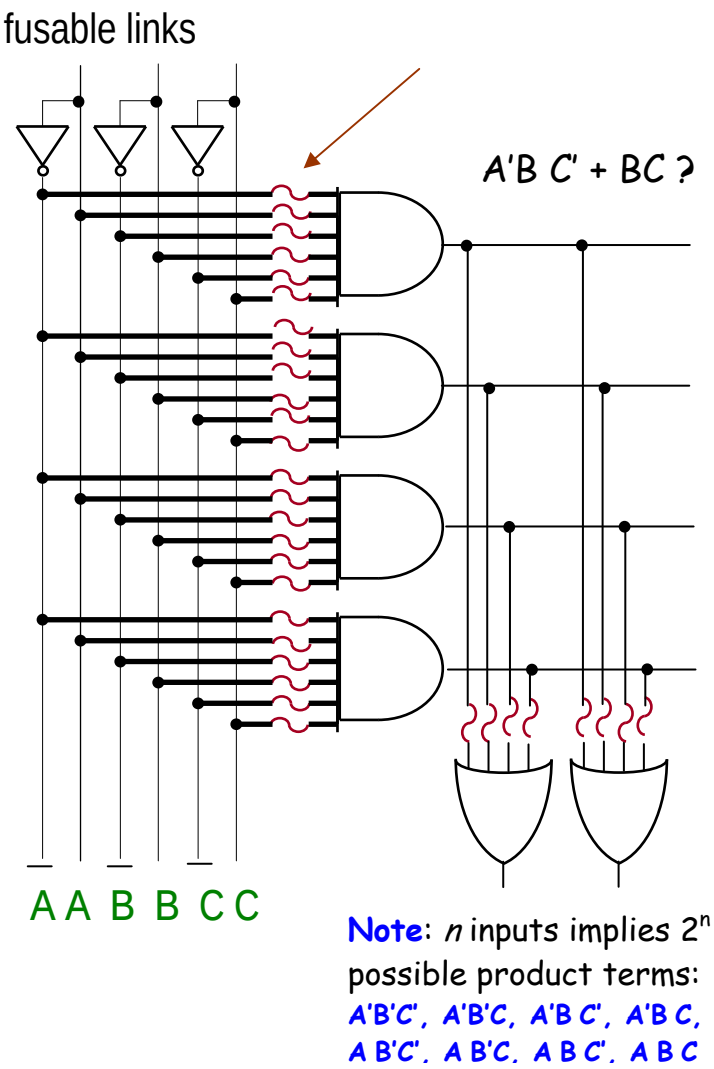


- **PLDs incorporate many AND-OR-INVERT and other structures with programmable interconnections.**

Programmable Logic Arrays (PLAs)

Programmable Logic Array (PLA): **AND & OR arrays are programmable** ⇒

AND plane produces product terms, OR plane realize sum of product terms.

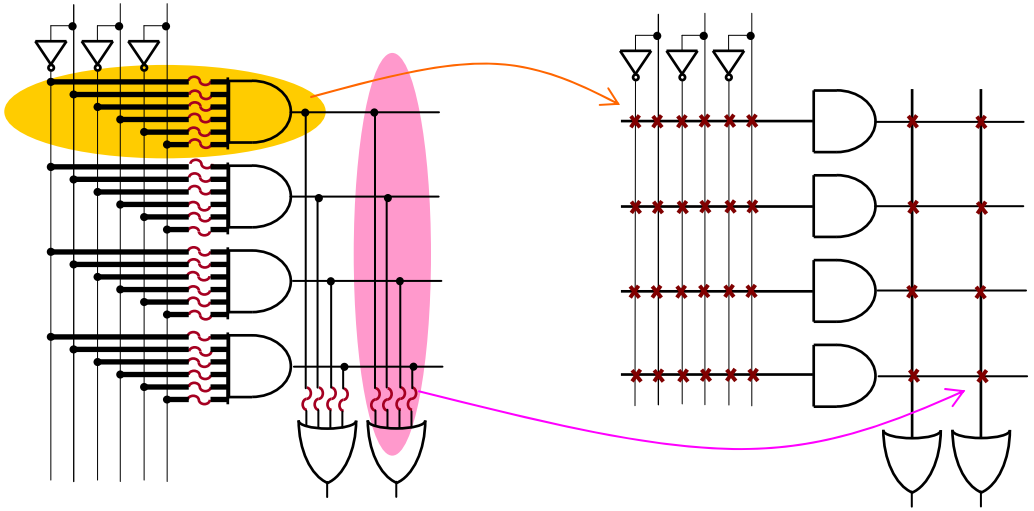


Commercial PLAs:

- **come in different sizes, with many more AND and OR terms.**
- **Either mask programmed by manufacturer or user programmed (Field Programmable PLAs – FPLAs).**

PLDs: Programming & Short hand notation …

A PLD comes with all fuses (~) intact and the fuses are selectively "blown" during programming. The intact fuses are shown by x's.



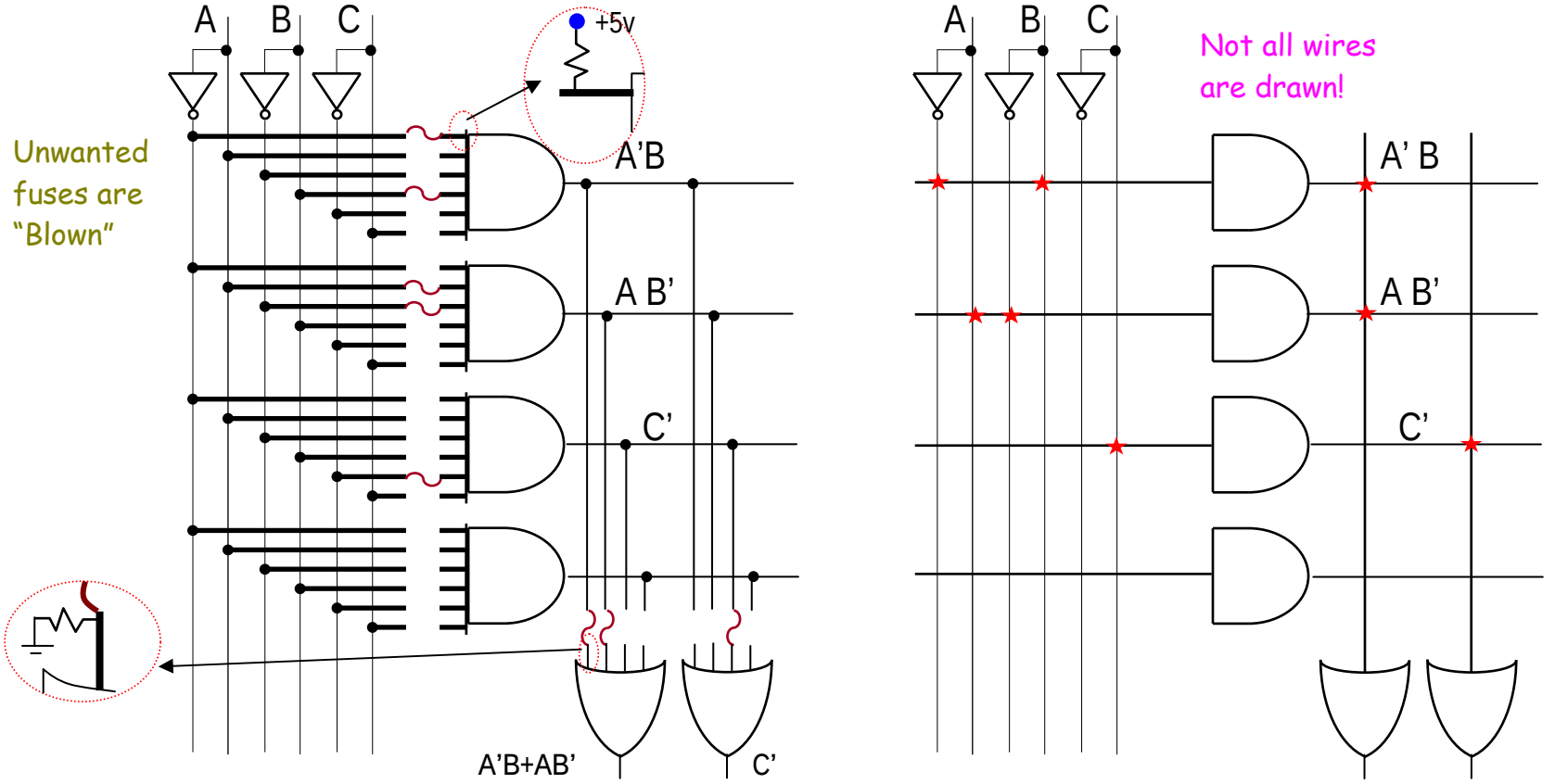
PLD is inserted into a *programming device* which is connected to a PC. User specifies logic functions to be realized on (e.g. VHDL based) software running on PC. The software generates a fuse map which specifies the fuses that are to be blown & those to be left intact to realize specified logic functions.

Fuse map is downloaded to *programming device*, which then blows the specified fuses and tests the programmed PLD for correct operation → *done*!

Programmable Logic Arrays (PLAs)

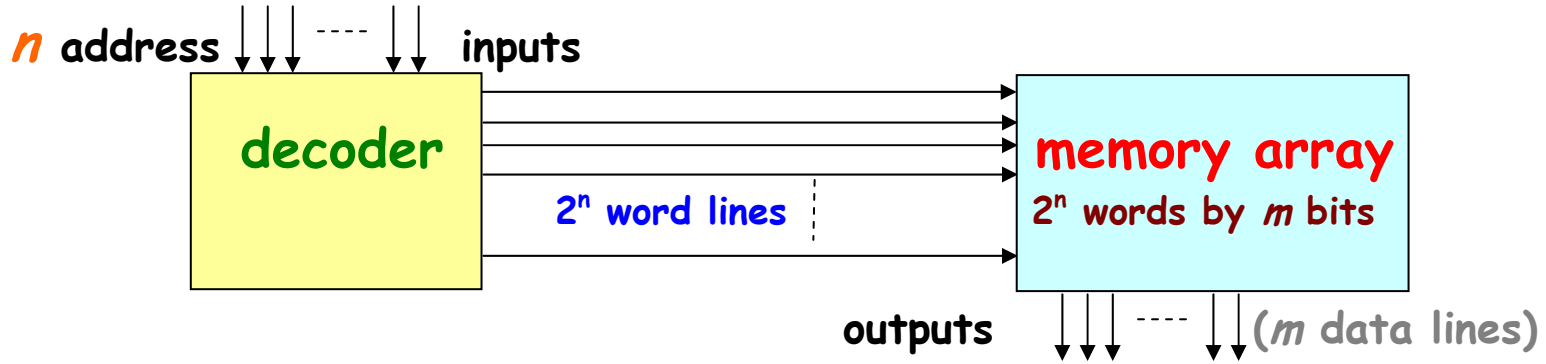
AND inputs are "pulled-up" high while OR inputs are "tied" low through resistors.

PLDs: AND, OR inputs gate inputs are depicted as single lines:



Programmable Read Only Memory (PROM)

A programmable read only memory or PROM has a fully decoded AND array (all AND terms decoded) & a completely flexible OR array (unlike PAL).



Combinational logic implementation using a PROM:

#1

$F0 = A' B' C + A B' C' + A B' C$

$F1 = A' B' C + A' B C' + A B C$

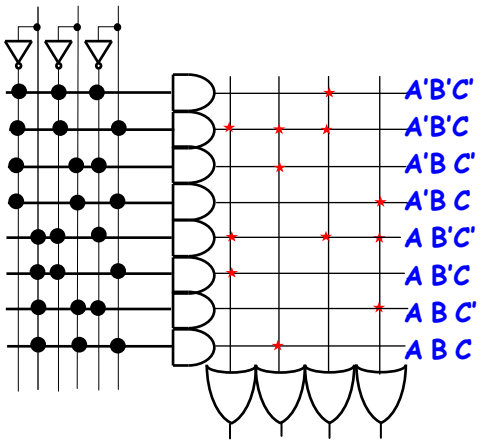
$F2 = A' B' C' + A' B' C + A B' C'$

$F3 = A' B C + A B' C' + A B C'$

#2

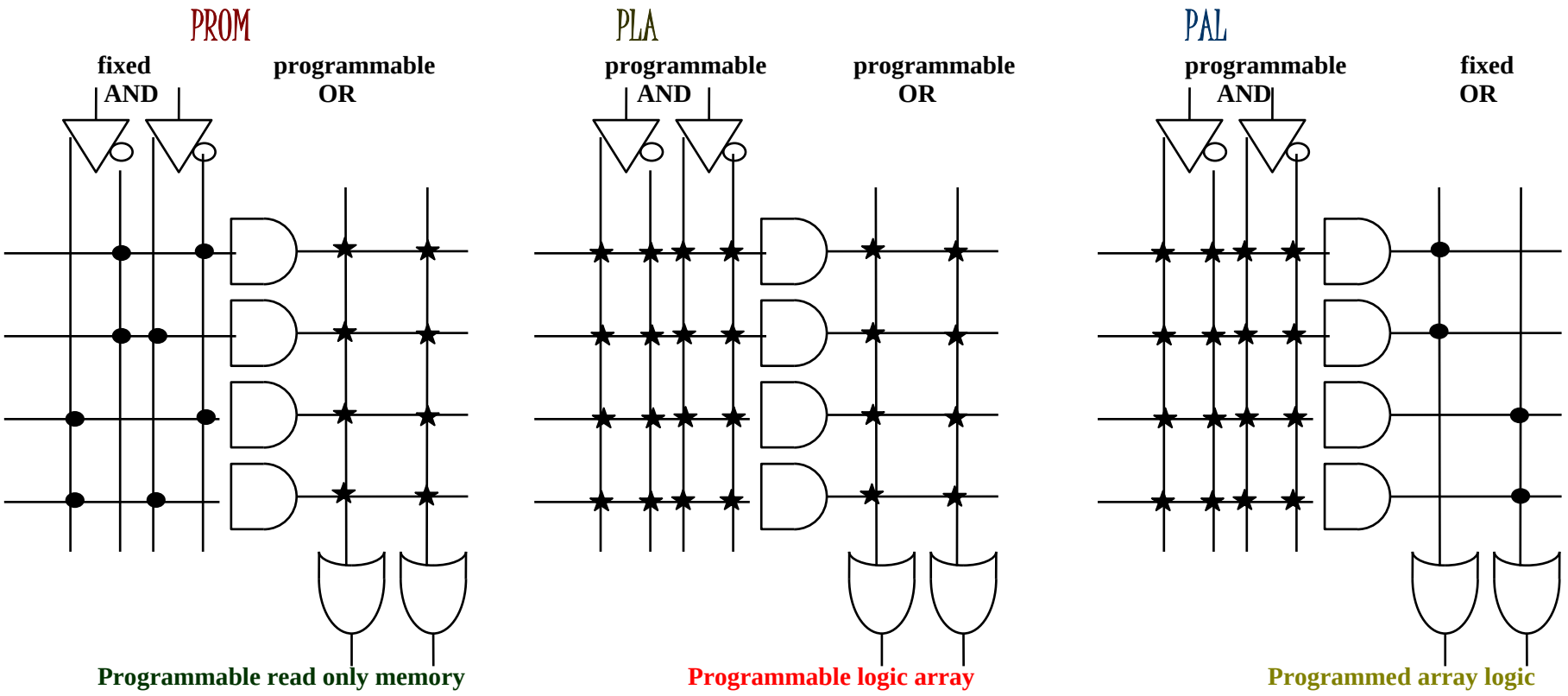
A	B	C	FO	F1	F2	F3
0	0	0	0	0	1	0
0	0	1	1	1	1	0
0	1	0	0	1	0	0
0	1	1	0	0	0	1
1	0	0	1	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	0	1
1	1	1	0	1	0	0

#3



PLDs: PROM, PLA & PAL

PLDs **incorporate many AND-OR-INVERT and other structures with programmable interconnections. Programming the interconnections in a PLD results in different logic implementations.**



Commercial PLDs

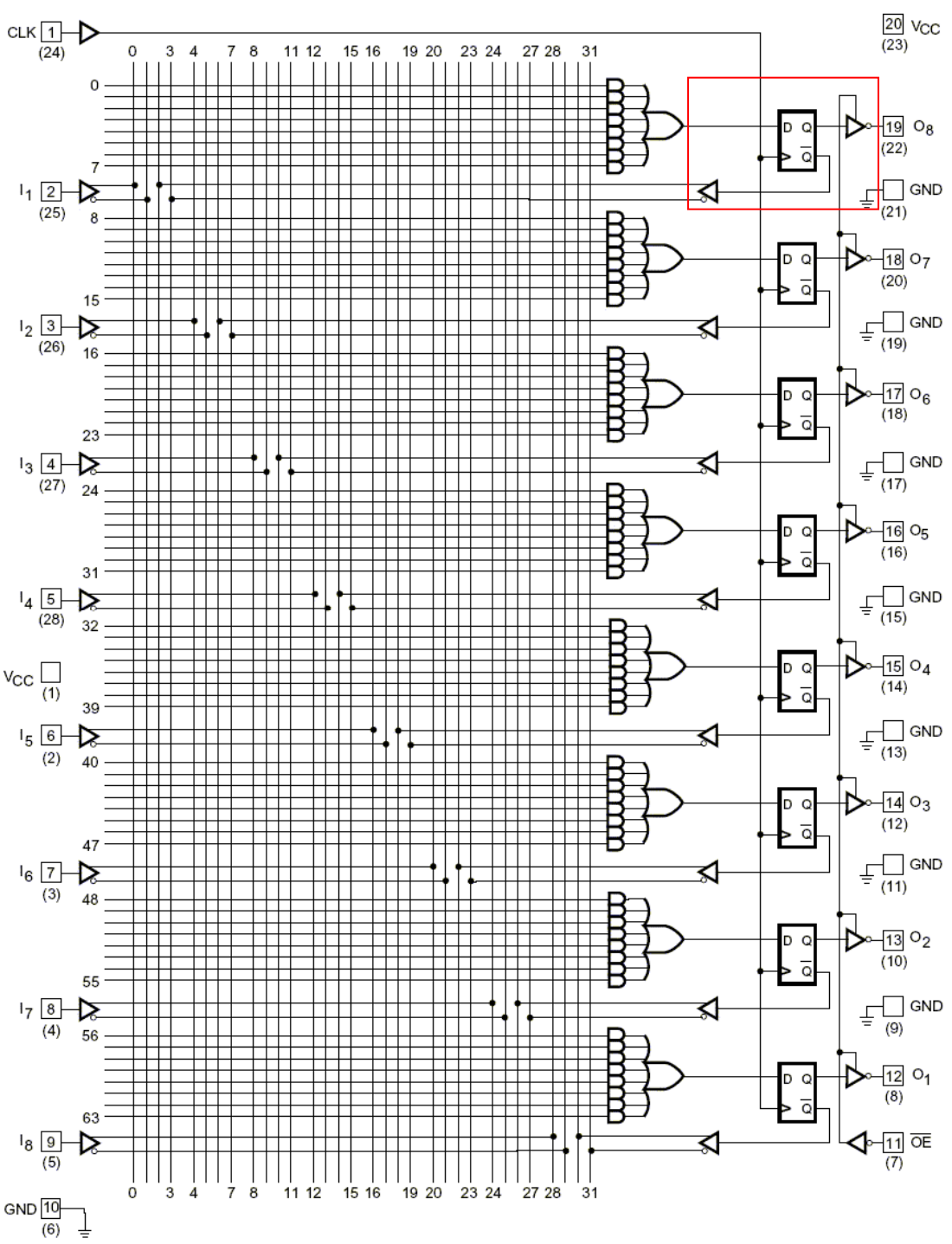
A PAL16R8 PLD ⇒

- 8 pdt terms to D input of FFs
- common clock (+ve edge) and tri-state enable for all
- Q output is fedback to AND array
- *applications*: counters SMs, etc

Conventional PLDs require a programmer & are either OTP (*one-time programmable*) or UV Erasable:

Electrically Erasable PLDs: can be programmed & erased in place:

- wire connection to a computer is needed
- once programmed, it will retain program
- does not need to be taken out of the circuit



Complex Programmable Logic Devices CPLD

Complex PLDs (CPLDs) combine PAL combinational logic with FFs:

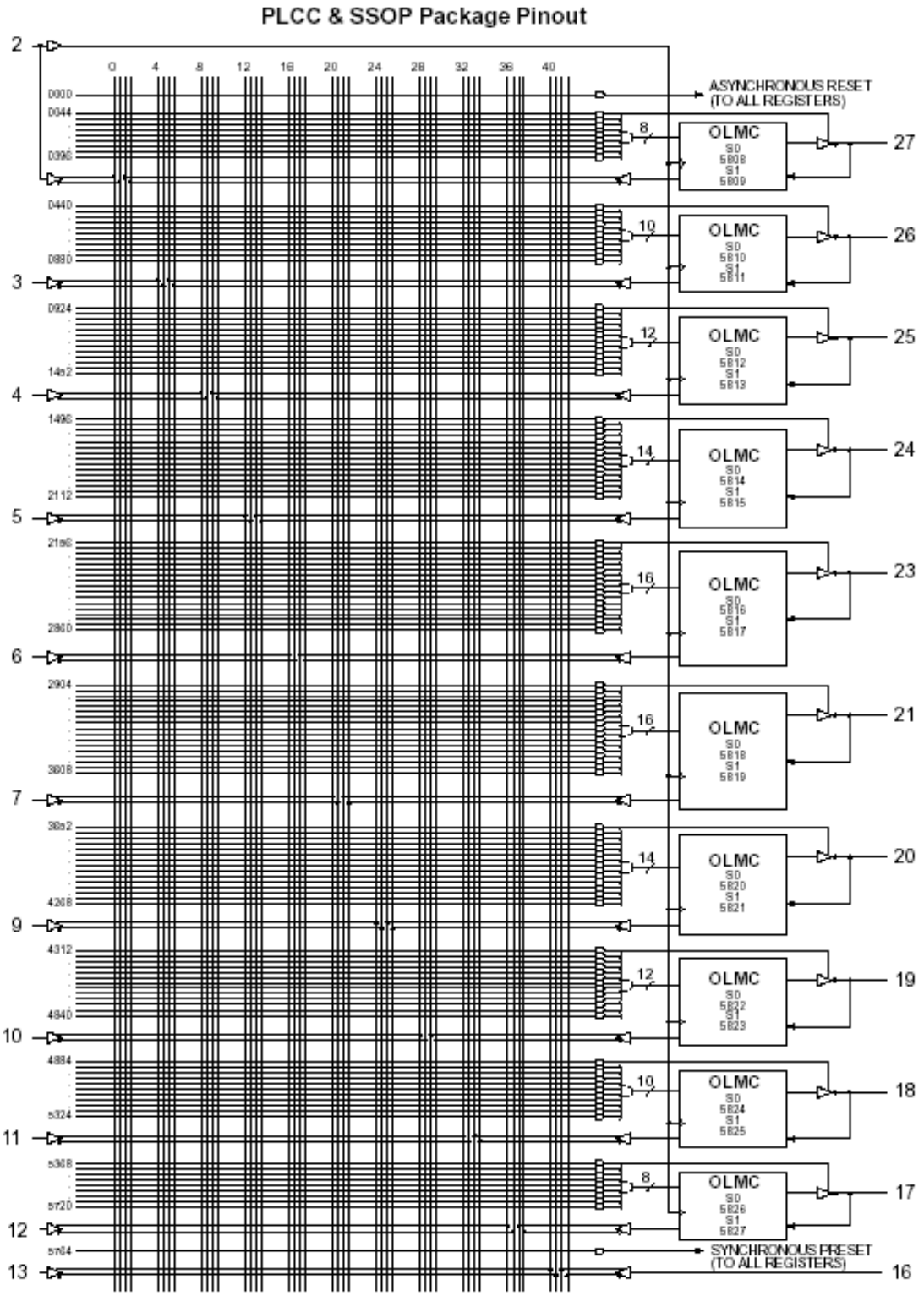
- **organized as logic blocks connected via programmable connections**
- **fixed OR array size**
- **combinational or registered outputs**
- **enough logic for simple counters, state machines, decoders, etc**
- **while PAL/PLA are generally *One-time programmable* (OTP), many CPLDs are *electrically erasable* and *in-system reprogrammable* (ISR).**
  - **ISR CPLD devices placed in printed circuit boards (PCBs) can be fed it with a serial data stream from a personal computer.**
  - **ISR CPLD is able to decode the data stream and configure the CPLD to perform its specified logic function.**
- **Examples: 22G10, 20V8 (check out the datasheets!)**

GAL22V10 CPLD

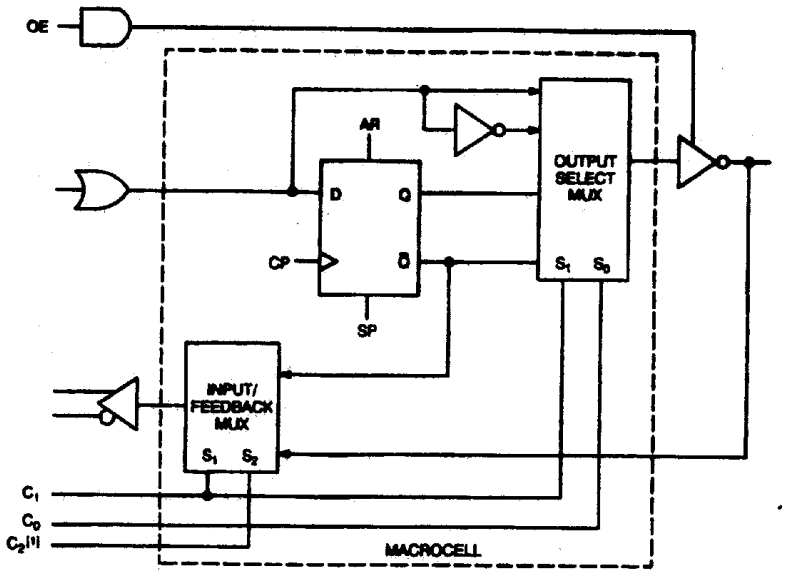
GAL: Generic Array Logic

Output Logic Macrocell (OLMC) is user configurable

- **compatible with standard bipolar & CMOS devices**
- **OLMC / FF outputs can be fed back**
- **dedicated inputs**
- ***programmable* I/O lines**

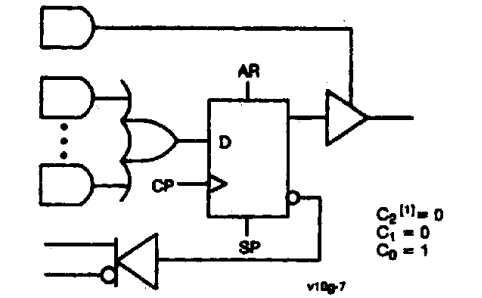


GAL22V10 OLMC

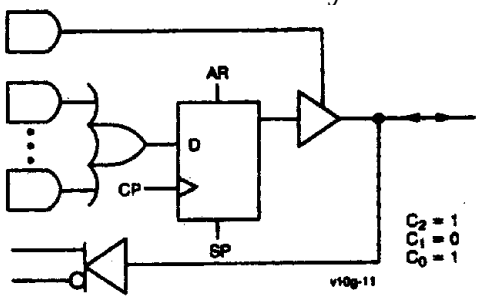


For each OLMC :

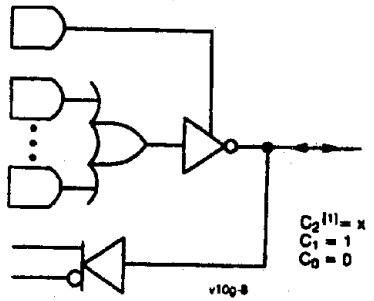
- **variable product terms per OLMC**
- **product-term for output enable control**
- **output polarity can be *active high/ low*, in *combinatorial* or *registered* mode**
- **product term for Asynchronous Reset (AR) (sets registers to *zero*) and for Synchronous Preset (SP) (sets registers to *one*) on *rising edge* of the next clock pulse.**
- **two modes: *registered* & *combinatorial* *I/O***
- ***REGISTERED MODE*: OLMC output is driven by Q output of its D flip-flop**
- ***COMBINATORIAL I/O MODE*: OLMC output is driven by output of sum term gate. Feedback into the AND array is from the side of output enable buffer**



REGISTER FEEDBACK, REGISTERED, ACTIVE-HIGH OUTPUT



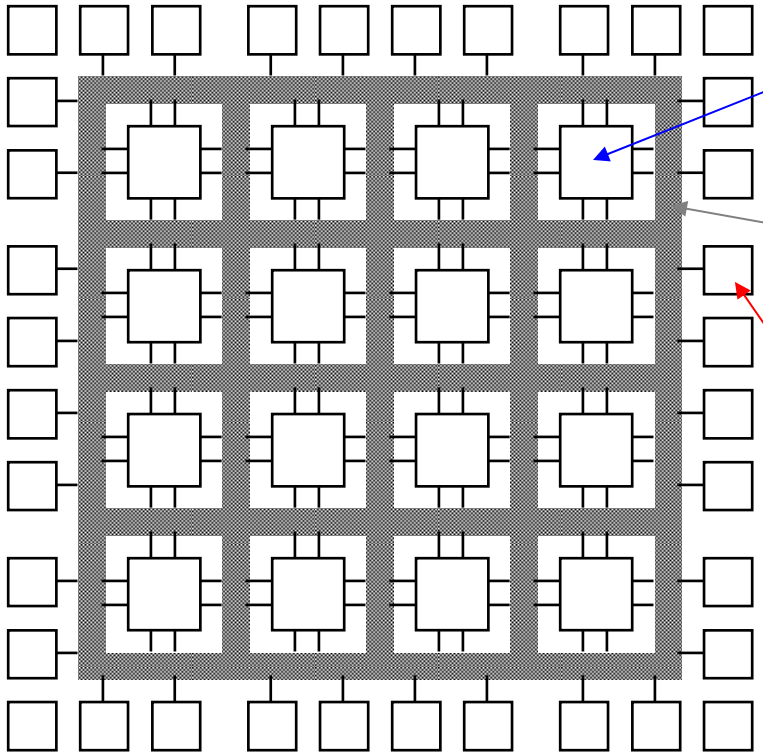
I/O FEEDBACK, REGISTERED, ACTIVE-HIGH OUTPUT



) FEEDBACK, COMBINATORIAL, ACTIVE-LOW OUTPU

Field Programmable Gate Array (*FPGA*)

Field Programmable Gate Array FPGAs: logic modules/blocks that are interconnected through programmable connections.



***Logic blocks*:**
To implement combinational & sequential logic

Interconnect
Wires to connect inputs and outputs to logic blocks

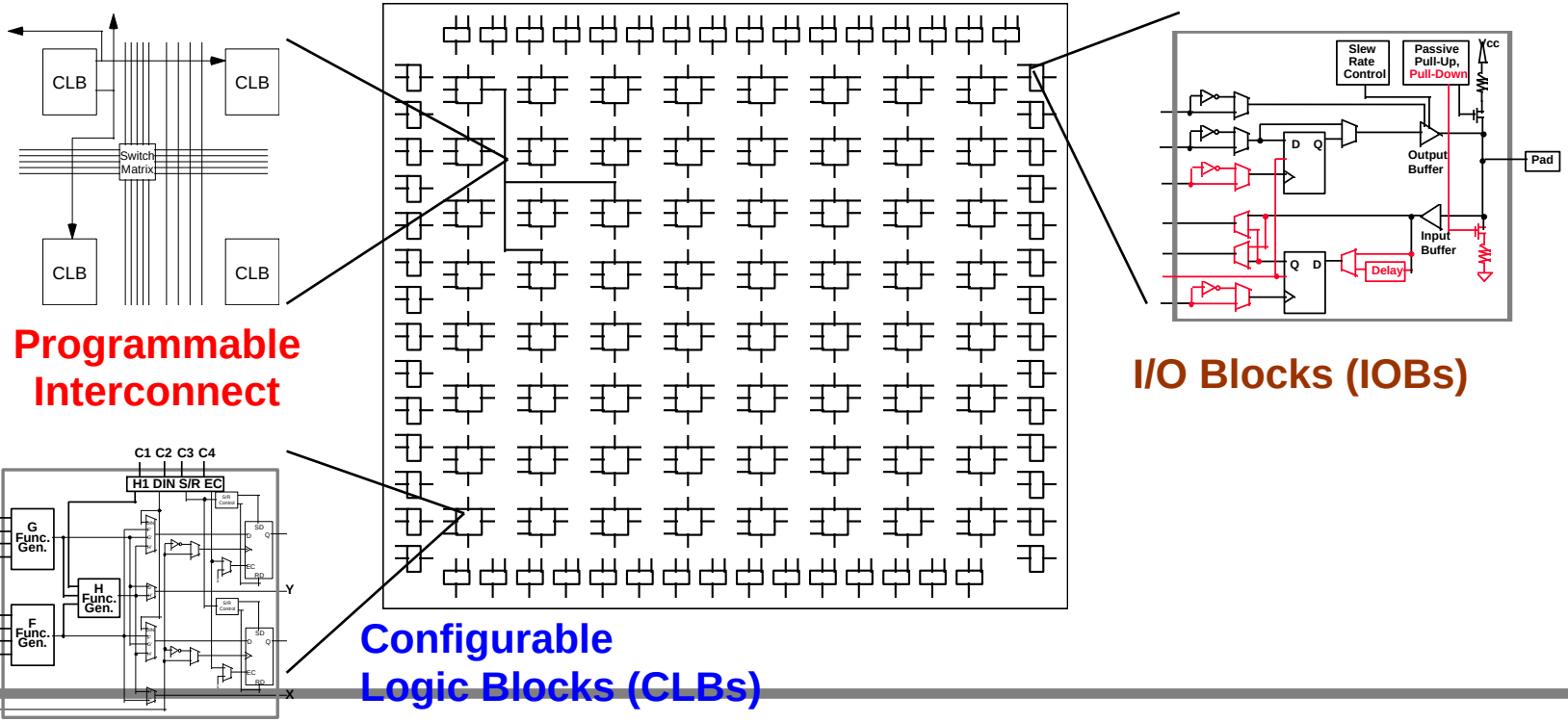
I/O blocks
Special logic blocks at periphery of device for external connections

***Key issues*:**
How to make logic blocks programmable?
How to connect the wires?
After the chip has been fabricated

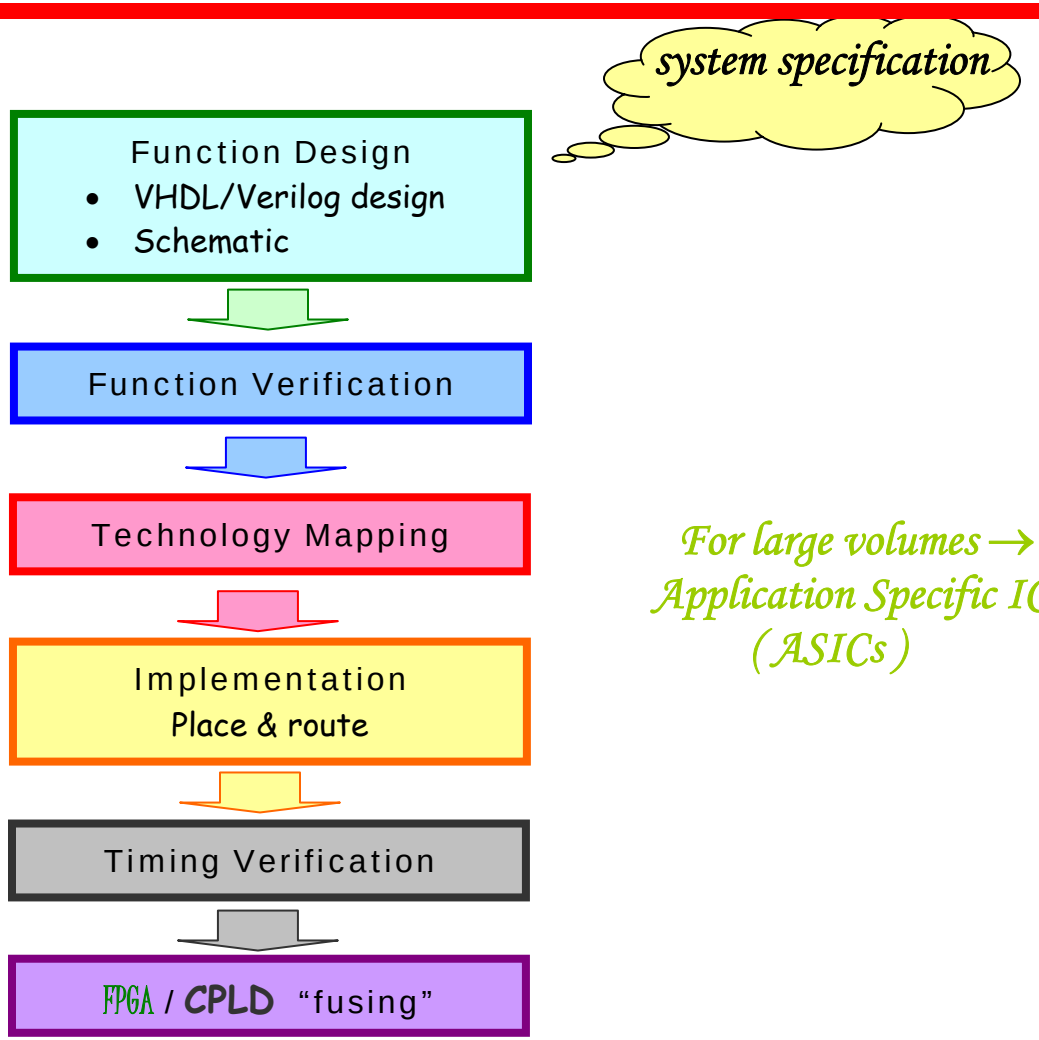
Xilinx *FPGA*...

Xilinx FPGAs consist of three major segments

- ***Configurable I/O Blocks*** **(IOBs): provide a programmable interface between internal logic array & I/O pins**
- ***Configurable Logic Blocks*** **(CLBs): perform user-specified logic functions**
- ***Interconnects*****: programmed to form networks on the chip**



FPGA & CPLD Design Flow…



For large volumes → Application Specific ICs (ASICs)