GAL Devices

- **Generic PAL** introduced by Lattice in 1980s.

- A more sophisticated output cell (**Macrocell**):
  - Included besides the flip-flop, several gates and multiplexers.
  - Macrocell itself was programmable.
  - a ‘return’ signal from the output of the Macrocell to the programmable array.
- **EEPROM** was employed instead of PROM or EPROM.

GAL Devices

- **GAL** is the only **SPLD** (Simple PLD) still manufactured in a standalone package.
- serves as the basic building block in the construction of most **CPLDs** (there are exceptions, however, like the CoolRunner **CPLD**, which employs **PLAs** instead).

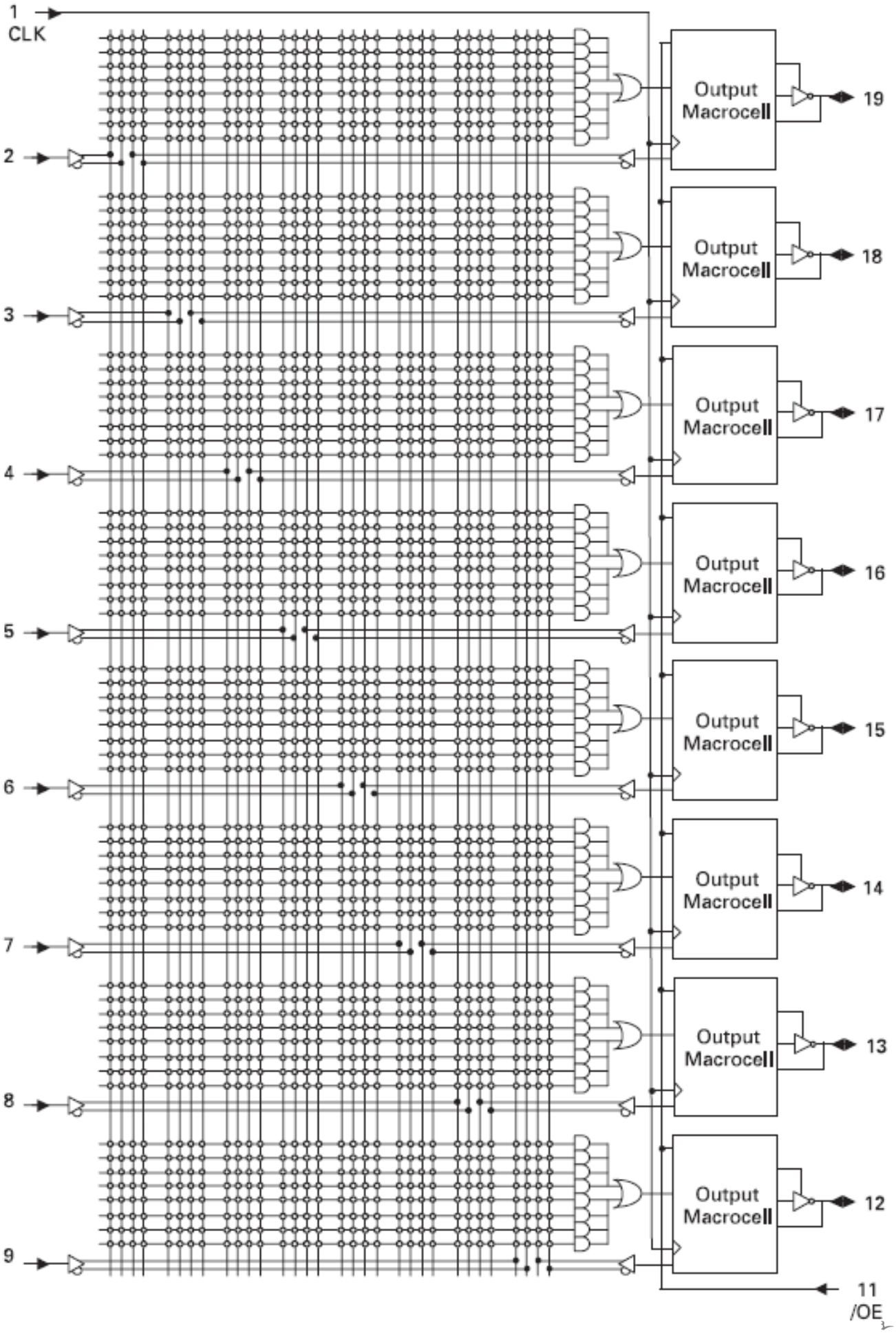


Figure A3
GAL 16V8 chip.

Complex PLD (**CPLD**)

- several PLDs (in general of **GAL** type) fabricated on a single chip.
- With programmable switch matrix.
- JTAG support
- interface to other logic standards (1.8 V, 2.5 V, 5 V, etc.).
- Altera, Xilinx, Lattice, Atmel, Cypress, etc.

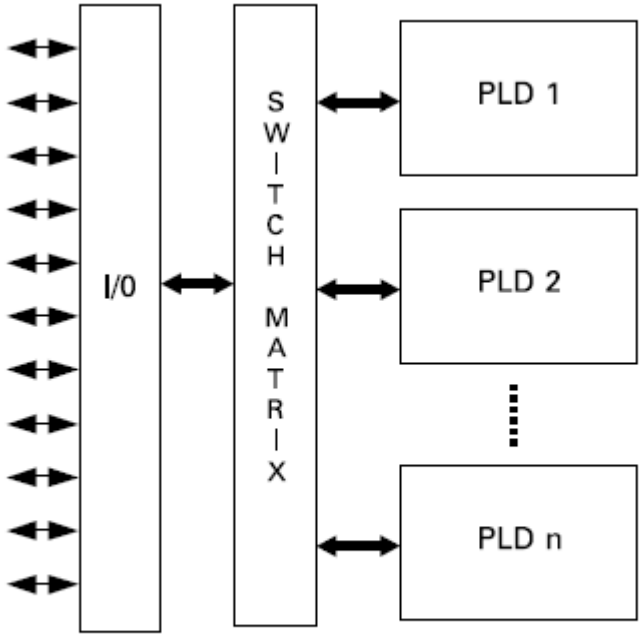


Figure A4
CPLD architecture.

Complex PLD (**CPLD**)

- Applications:
  - Decoders
  - Encoders
  - Multiplexers
  - De-Multiplexers

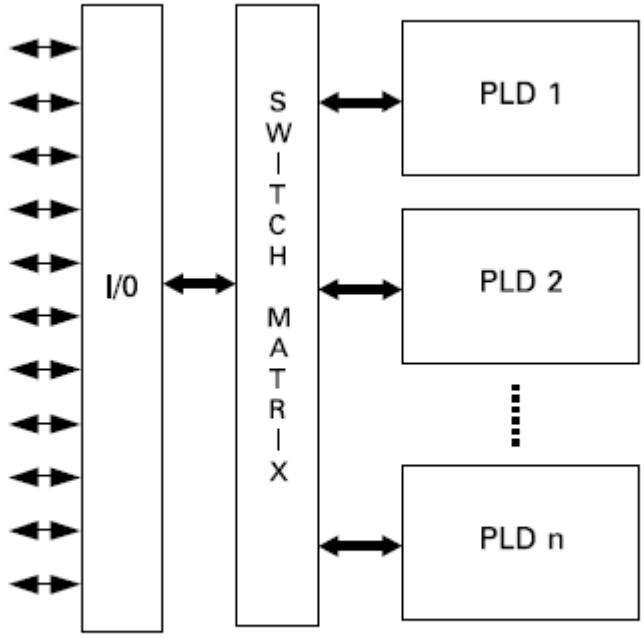
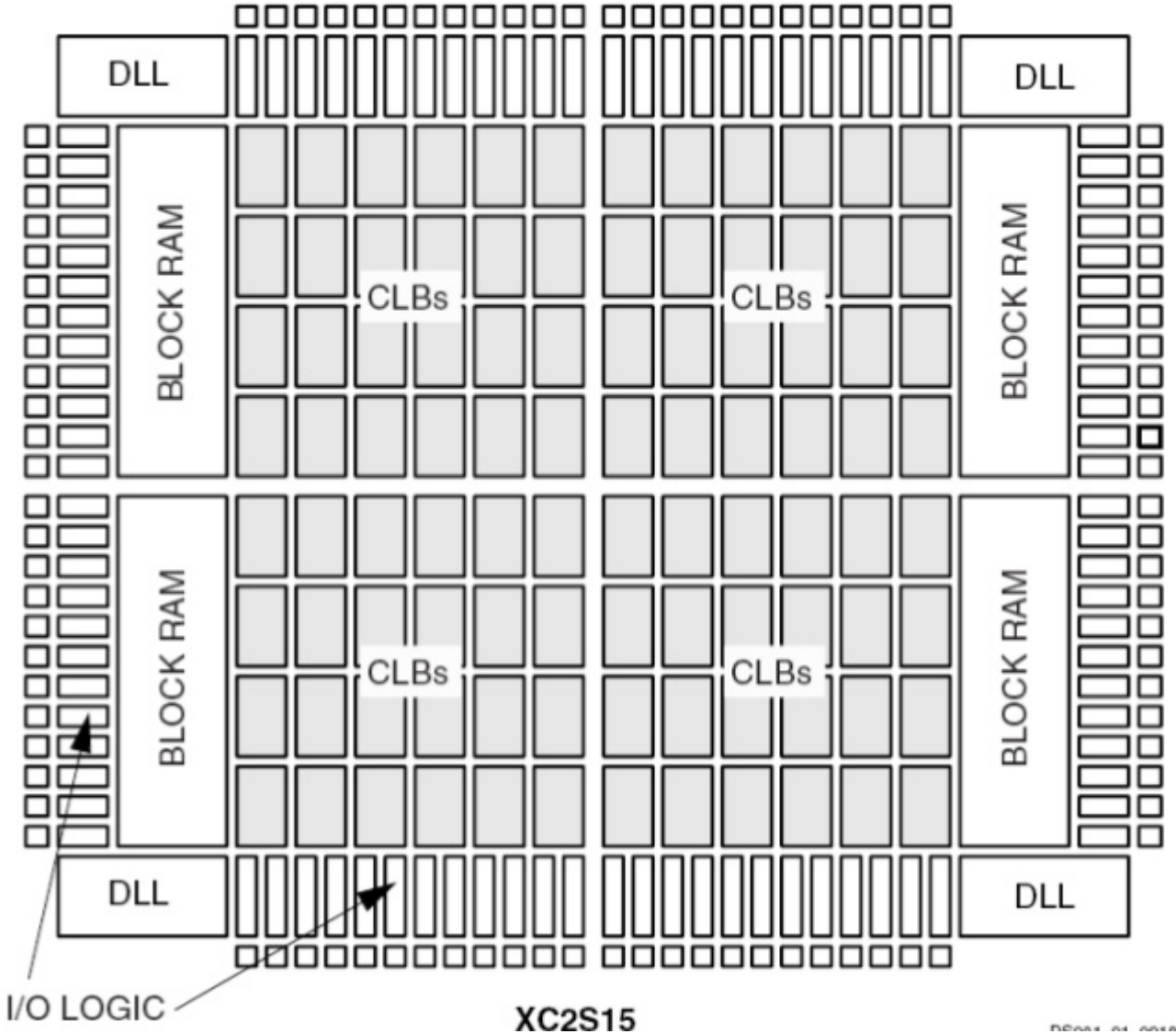


Figure A4
CPLD architecture.

Field Programmable Gate Array (**FPGA**)

- introduced by Xilinx in the mid 1980s.
- differ from CPLDs in architecture, storage technology, number of built-in features, and cost.
- Aimed at the implementation of high performance, large-size circuits.

Spartan-II FPGA



Field Programmable Gate Array (**FPGA**)

- It consists of a matrix of **CLBs** (Configurable Logic Blocks),
- interconnected by an array of **switch matrices**.
- Array Input/output Blocks.

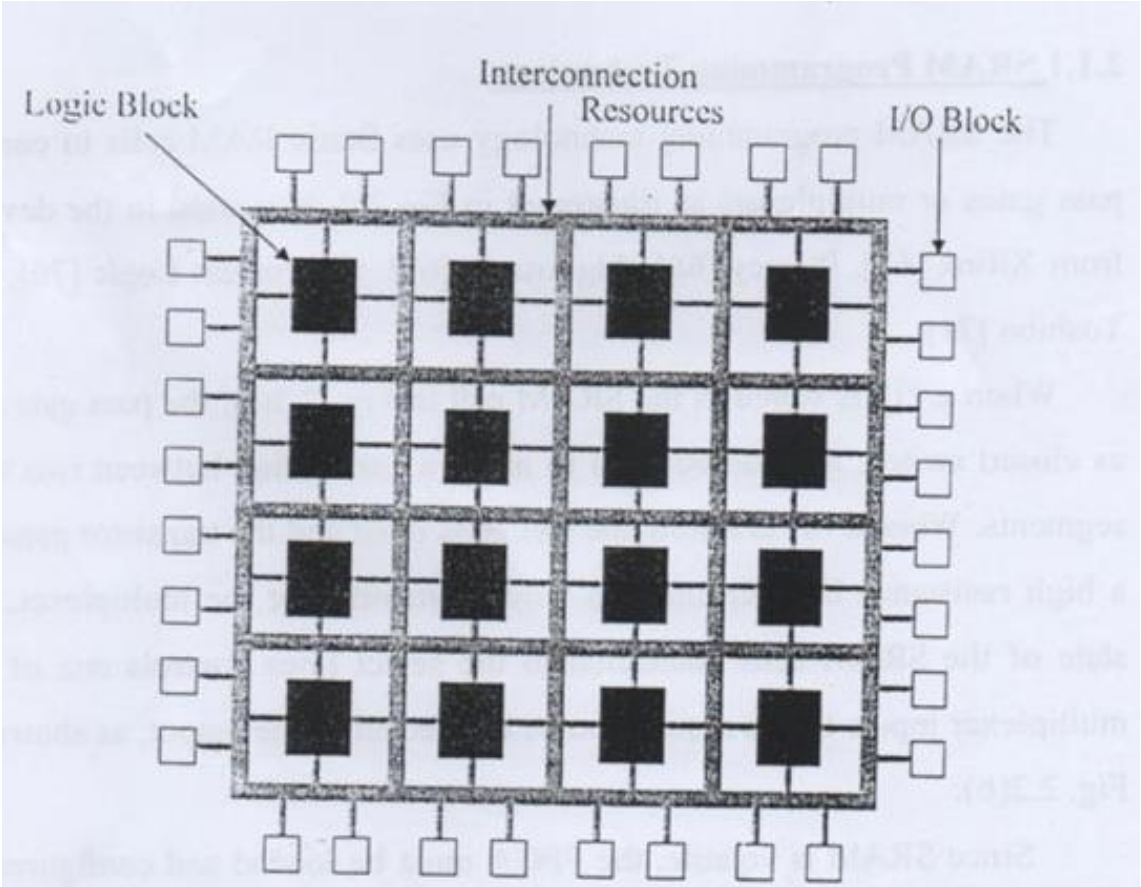


Fig. 2.1 FPGA architecture.

Field Programmable Gate Array (**FPGA**)

- First, instead PAL (in SPLDs), its operation is normally based on a LUT (lookup table).
- The number of flip-flops is much more abundant (more sophisticated sequential Circuits).
- JTAG support.
- interface to diverse logic levels.
- SRAM memory
- clock multiplication (PLL or DLL).
- PCI interface, etc.
- Some chips: multipliers, DSPs, and microprocessors.

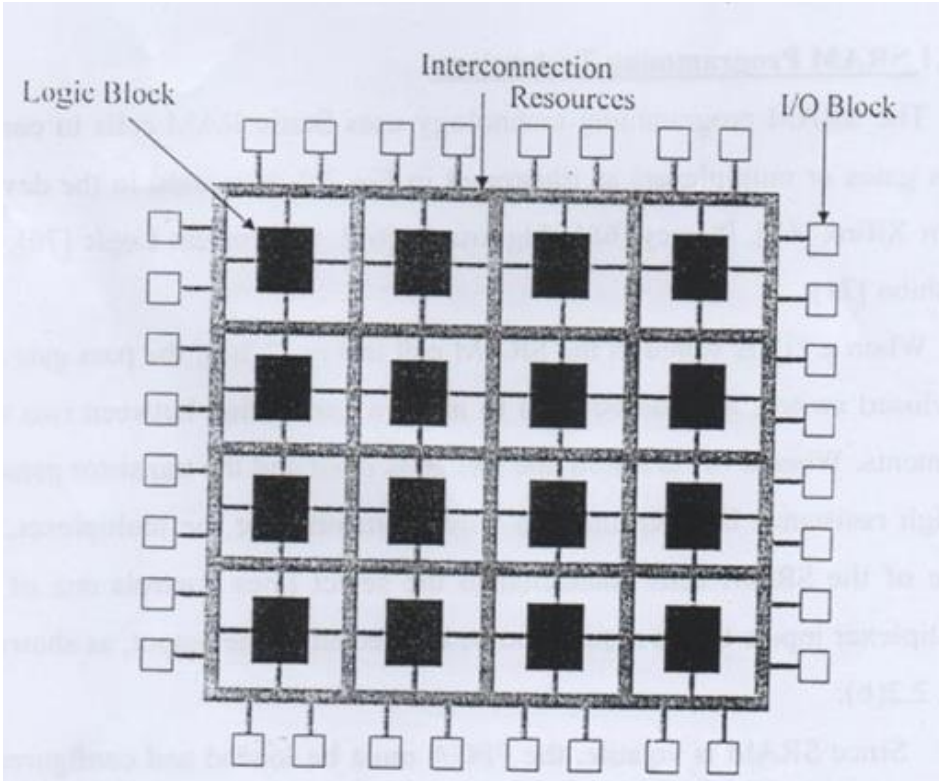


Fig. 2.1 FPGA architecture.

Field Programmable Gate Array (**FPGA**)

- Storage of the interconnects:
  - **CPLDs** are non-volatile (antifuse, EEPROM, Flash, etc.).
  - most **FPGAs** use SRAM, and are therefore volatile.
    - saves space and lowers the cost but requires an external ROM.
  - non-volatile FPGAs (with antifuse), which might be advantageous when reprogramming is not necessary.

Field Programmable Gate Array (**FPGA**)

- Several companies manufacture FPGAs, like Xilinx, Actel, Altera, QuickLogic, Atmel, etc.

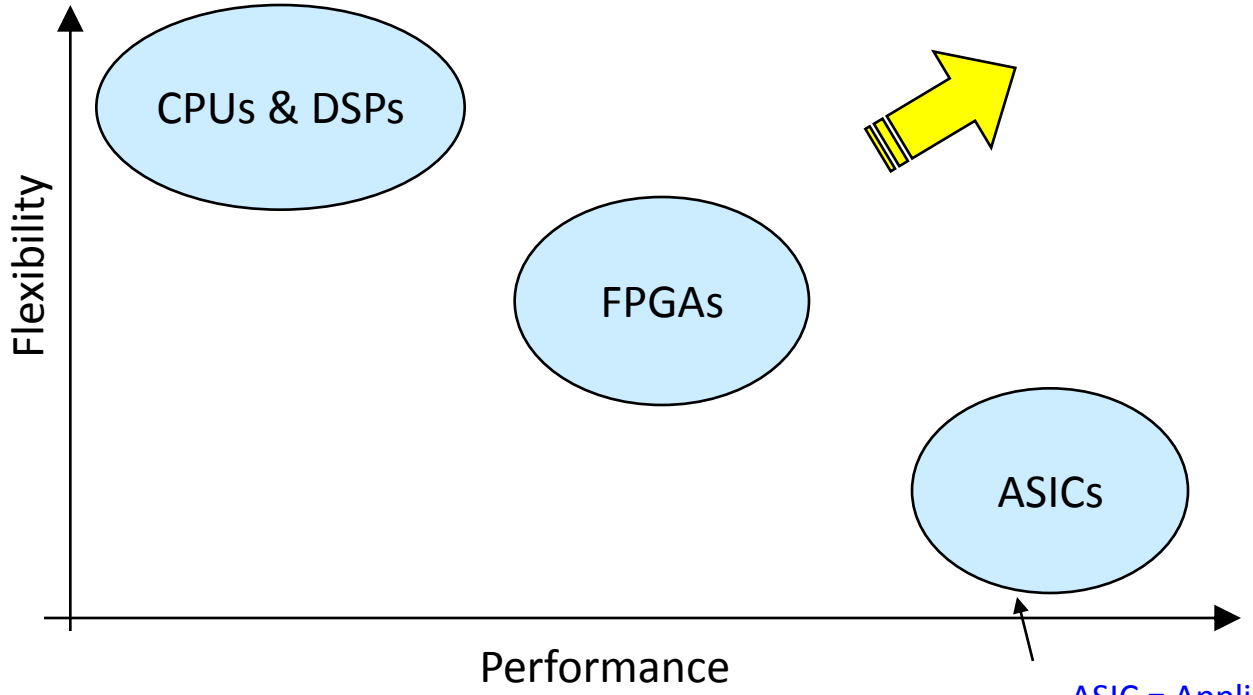
- Interconnects:
  - all Xilinx FPGAs use **SRAM**, **reprogrammable**, but **volatile** (thus requiring external ROM).
  - Actel FPGAs use **antifuse** , **non-volatile** (they), but are **non-reprogrammable** (except one family, which uses Flash memory).

Field Programmable Gate Array (**FPGA**)

- the actual **application** will dictate which chip architecture is most **appropriate**.

- Applications:
  - Aerospace and Defense.
  - Medical Elctronics.
  - Wired Communications.
  - Wireless Communications.
  - High performance computing.

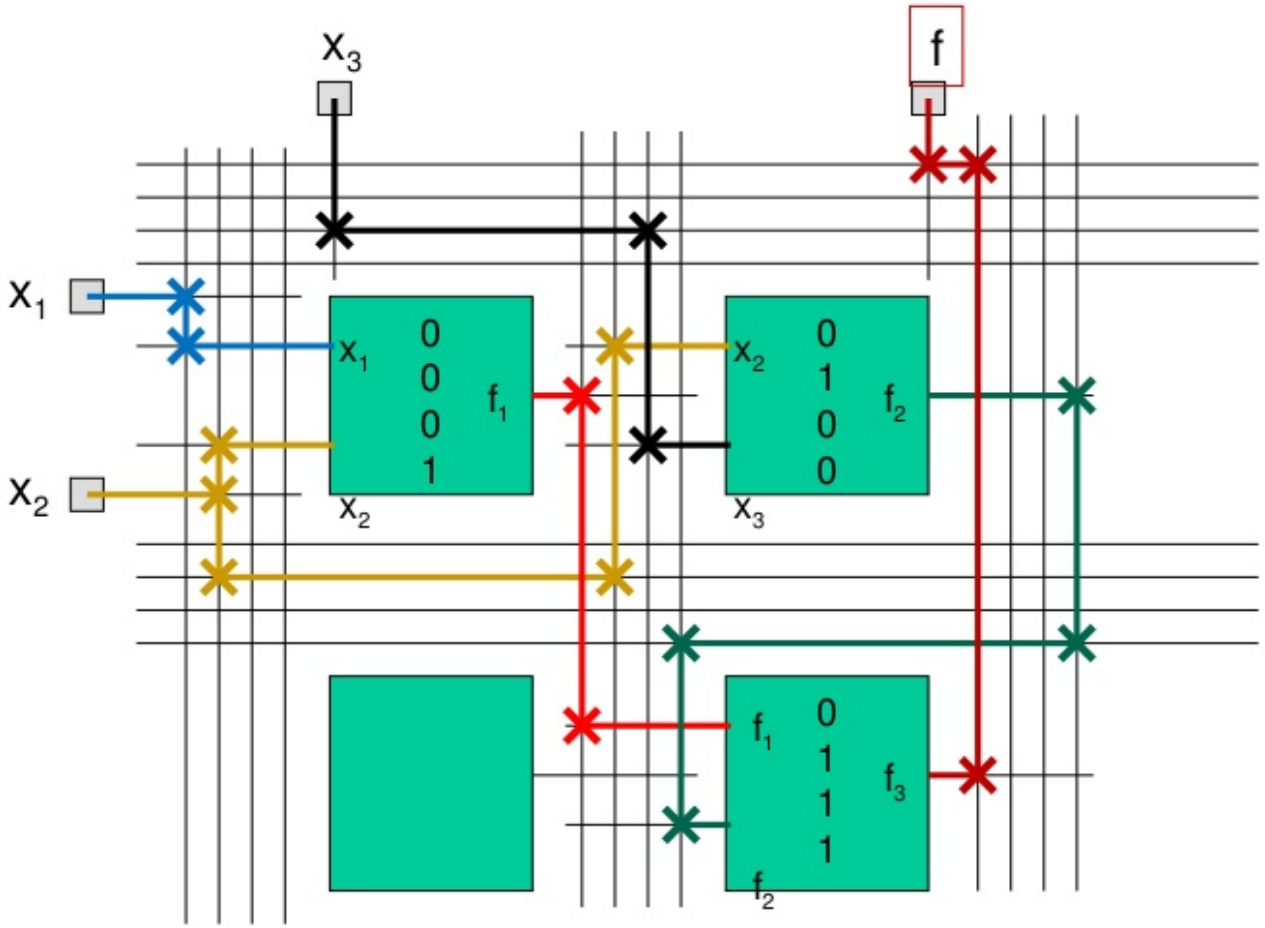
Performance vs. Flexibility



Goal: the performance of ASIC's with the flexibility of programmable processors.

FPGAs

- An example of programming an FPGA



$$f_1 = x_1 x_2$$

$$f_2 = \overline{x_2} x_3$$

$$f = x_1 x_2 + \overline{x_2} x_3$$