Introduction towards FPGA

More details on FPGA

- http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf

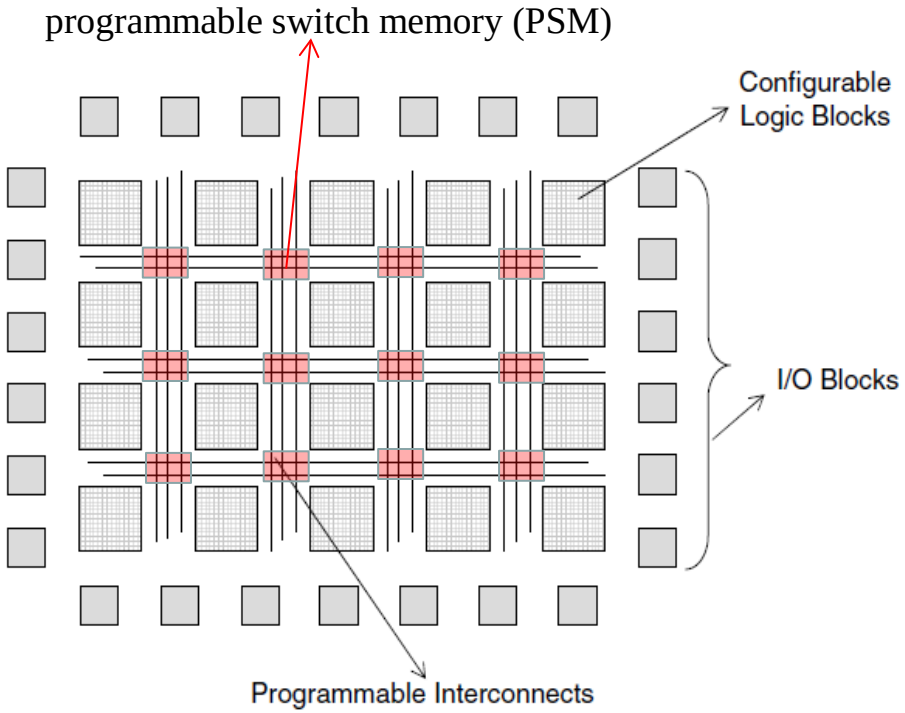
field-programmable gate array (FPGA)

core architecture of an FPGA consists of three main components:

- an array of configurable/programmable logic blocks: the combinational units
- a sea of programmable interconnects
- memories and specialized I/O blocks

other components in modern FPGAs are

- DSP blocks,
- hardwired IP cores
- soft cores



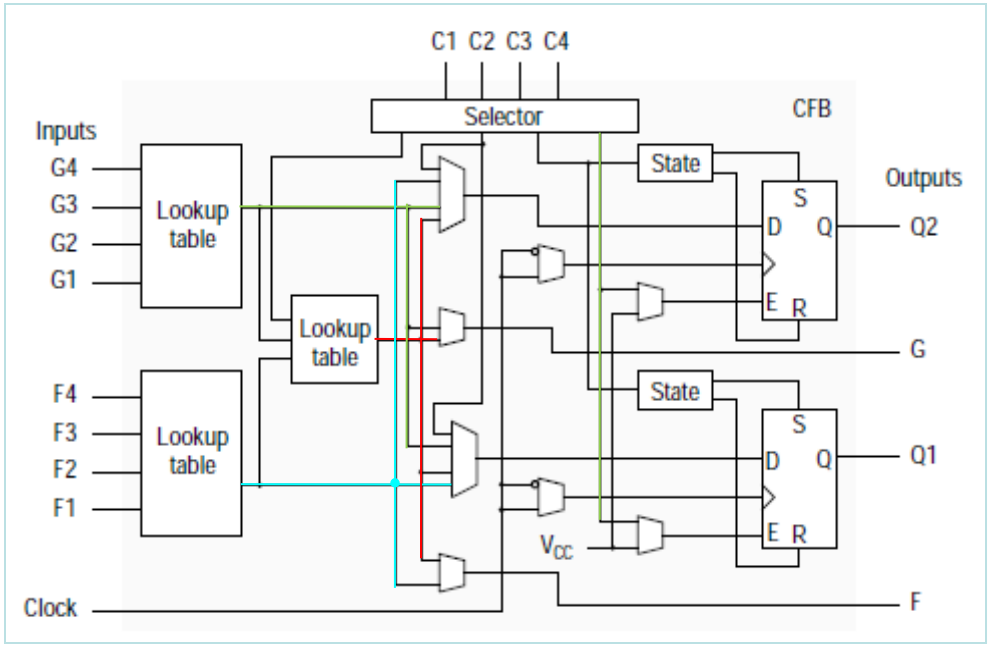
How are SRAM-based FPGAs different from Antifuse-programmed FPGAs?
Is there any 3-D FPGA?

Source: S. Brown and J. Rose, "FPGA and CPLD Architectures: A Tutorial", IEEE Design & Test of Computers, Summer 1996, pp. 42-57

FPGA CLB: example Xilinx XC4000

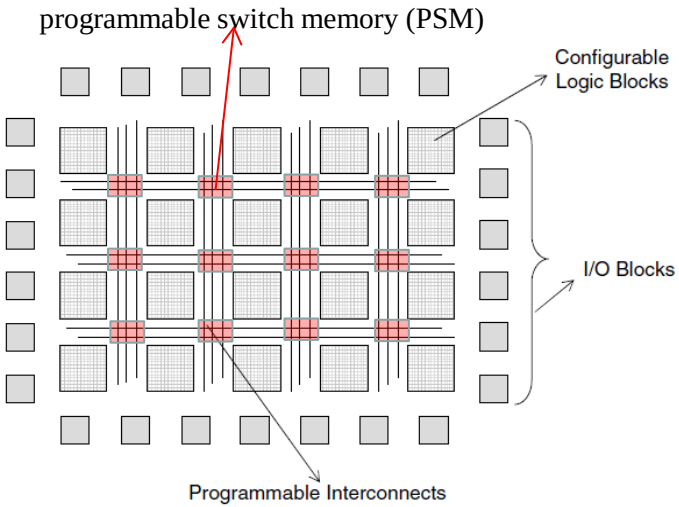
- contains 2 four-input LUTs fed by two pairs of 4-bit inputs to the CLB
- the third LUT is a 3-input LUT fed by the output of other two.
  - CLB can implement logic functions of up to nine bit inputs
  - or two separate 4-input functions
  - the output might be registered or not registered
  - registers could be set and reset and disabled

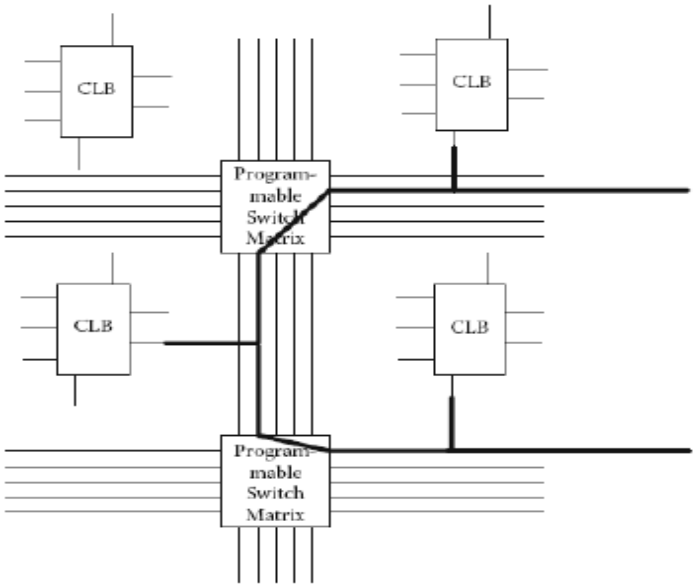
  - each LUT is 1-bit-wide memory array
  - address lines are CLB inputs
  - 1-bit memory output is LUT output
  - size of a K input LUT is ?
  - any desired K-input logic function can be realized this LUT by writing the truth table for the function directly into the memory.

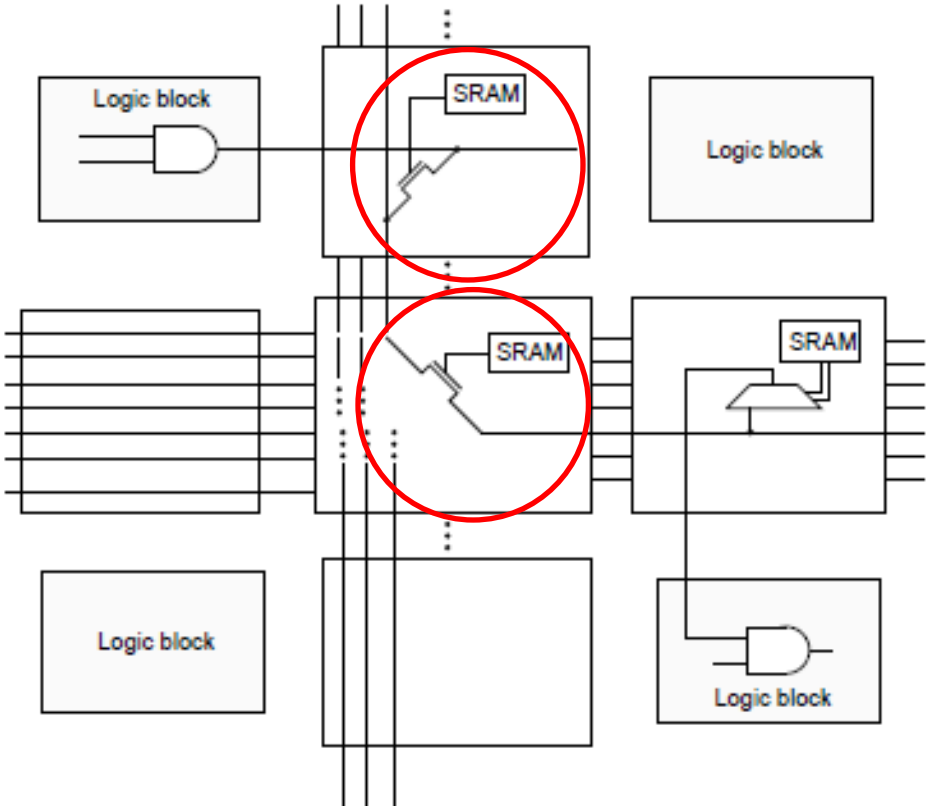


Source: S. Brown and J. Rose, “FPGA and CPLD Architectures: A Tutorial”, IEEE Design & Test of Computers, Summer 1996, pp. 42-57

FPGA: programmable interconnects

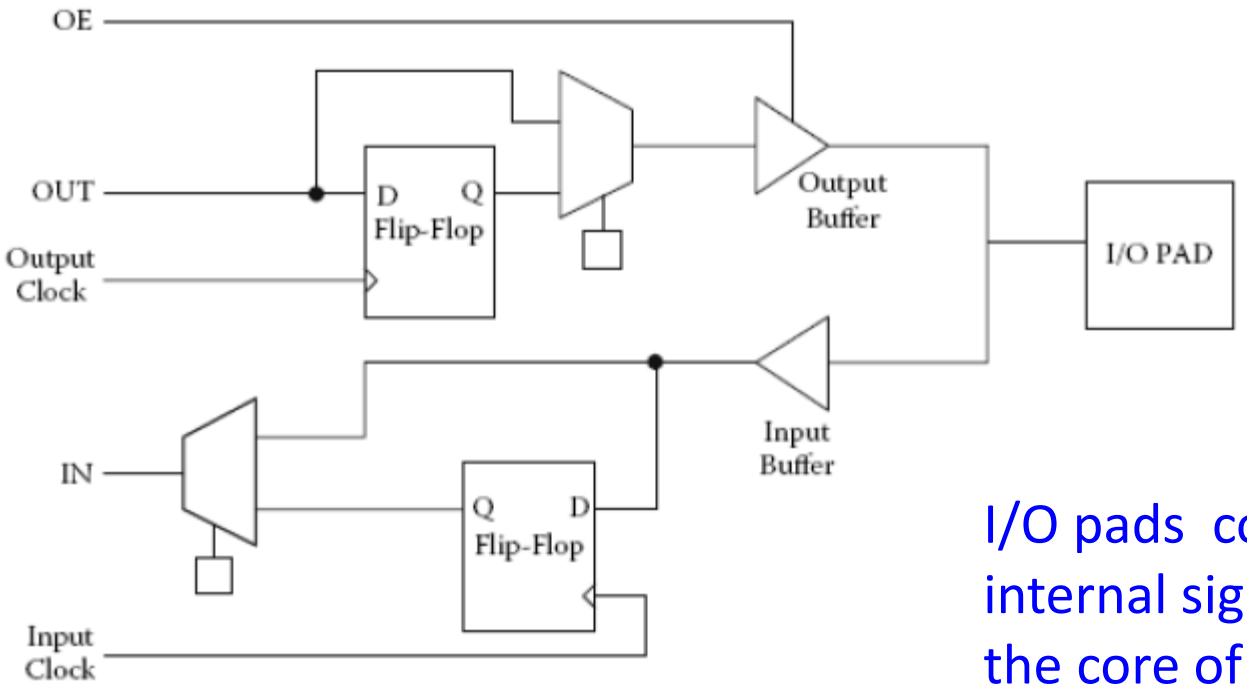






Source: S. Brown and J. Rose, “FPGA and CPLD Architectures: A Tutorial”, IEEE Design & Test of Computers, Summer 1996, pp. 42-57

FPGA: I/O blocks



I/O pads connect internal signals from the core of IC to the external pins of the chip.

Source: S. Brown and J. Rose, "FPGA and CPLD Architectures: A Tutorial", IEEE Design & Test of Computers, Summer 1996, pp. 42-57

advantage and disadvantages of FPGA

- reconfigurable and reprogrammable: allows design reuse
- shorter development time: allows fast time-to-market but more difficult to debug compared to a software approach
- change in design can be made quickly: easy to rectify mistakes
- inexpensive development tools
- supports parallel design, allows SoC design, offers greater performance compared to software–based solutions with even lower clock speed
- designs made for FPGA can be modified for ASIC-based implementation
- excellent for prototyping
- virtually no processor obsolescence
- usually higher unit cost and less performance than a ASIC
- **typically have much higher power dissipation than ASIC**

FPGA vs DSP

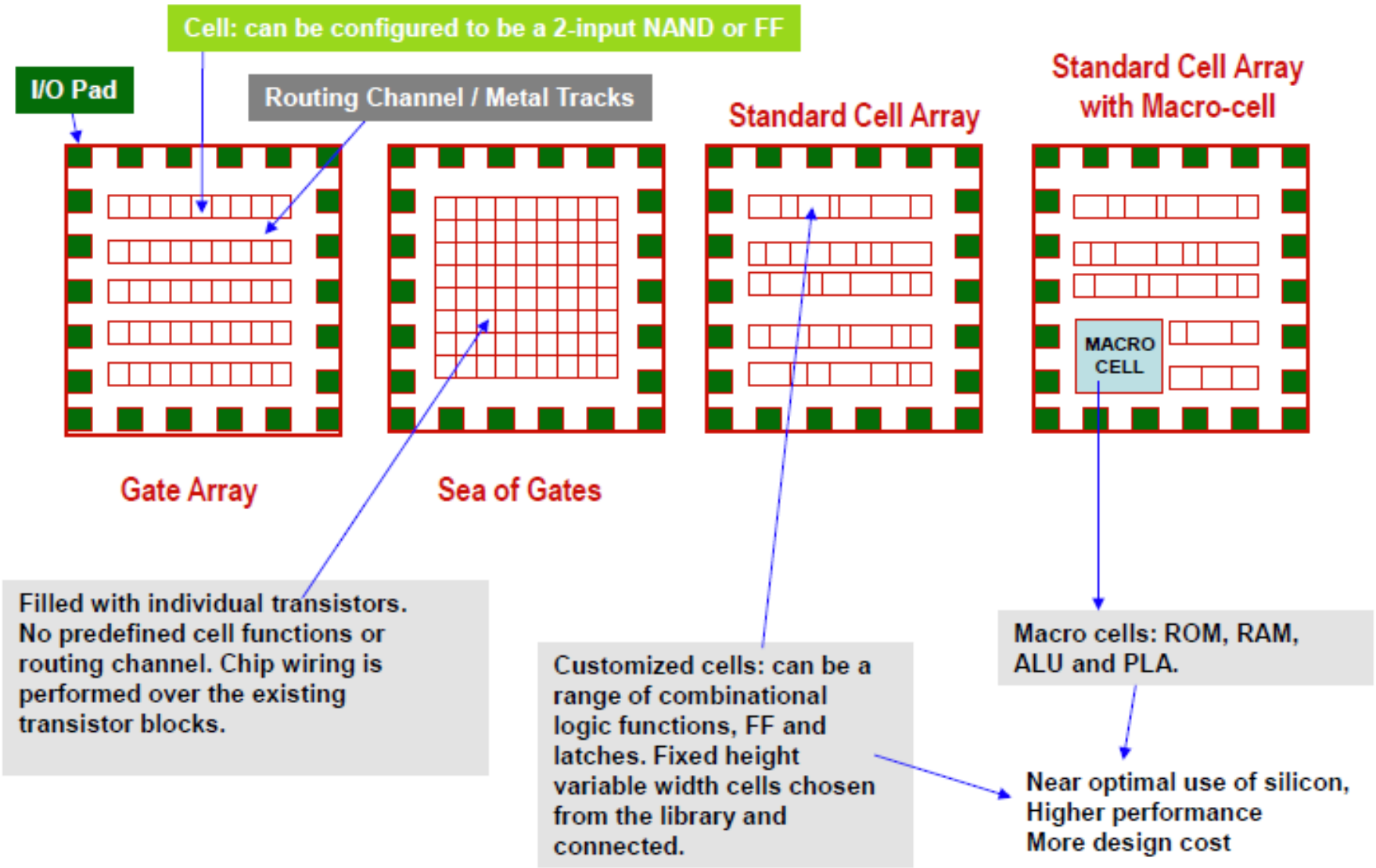
- Both FPGA and DSP offer flexibility for reuse and programmability
- FPGA is more expensive than DSP
- Power consumption of FPGA is higher than DSP
- Development time for FPGA is longer than DSP
- Development of FPGA is considered more complicated than DSP
- DSP has advantage in time-to-market
- FPGA supports parallel design and offers greater performance compared to DSP
- FPGA can provide relatively much higher performance for highly parallelizable tasks.
- FPGA has offers higher performance per dollar compared to DSP although it is more expensive.
- Heterogeneous architectures, consisting of both DSP and FPGA components is an emerging trend

source: http://www.bdti.com/MyBDTI/pubs/info_stanford02_fpgas.pdf

application specific integrated circuits (ASIC)

- provides customized tailored hardware solutions for specific applications or problems
  - not reusable and inflexible: not configurable
  - cannot be upgraded, updates requires redesign
  - involves high NRE cost of several millions for development & testing:
  - expensive CAD tools
  - need to have large volume market to reduce cost
  - longer design and development time: longer time-to-market
  - mistakes in product development are costly
  - maximum performance per watt, per unit area
  - mixed-analog and digital design
  - design optimization by and IC manufacturing
- semi-custom design: gate array implementation: standard cell implementation
- full-custom design: cell design, cell library development and use: simulation and testing for design verification
- silicon compilation: automatic layout generation: cell compiler: fast turn around implementation

ASIC and gate arrays



FPGA vs ASIC

- Both FPGA and ASIC can be used to support highly parallel tasks to provide greater performance
- ASIC can provide higher clock period than FPGA
- ASIC gives higher speed performance than FPGA
- ASIC has a smaller form factor than FPGA
- FPGA consumes more power and energy than FPGA
- FPGA is more expensive than ASIC for large volume market
- FPGA is reconfigurable, while ASIC is not flexible at all.
- FPGA allows design reuse and it is easy to upgrade
- FPGA has shorter development time and design changes and upgrades can be made quickly
- Easy to rectify mistakes in FPGA
- FPGA has faster time-to-market
- ASIC development tools are expensive