

Introduction to PLDs and FPGAs

System implementation choices

- Standard part solution
- ASIC or custom IC solution
- FPGA based solution

	performance	NREs	Unit cost	TTM
↑	ASIC	ASIC	FPGA	ASIC
	FPGA	FPGA	MICRO	FPGA
	MICRO	MICRO	ASIC	MICRO

Evolution of semiconductor industry

- Electronic circuits that do data processing generally consist of
 - Data operators
 - Storage elements
 - Wires
- Semiconductor industry has evolved around effective delivery of these

Evolution of programmable devices (PDs)

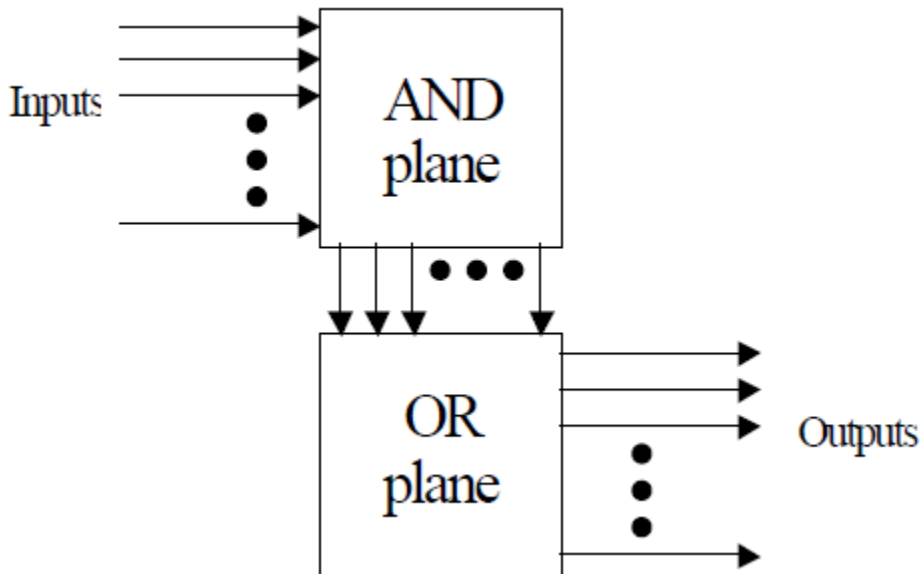
- Mask programmed read only memory (MPROM)
- PROM (OTPROM)
- Can we implement logic function using these?
 - Inefficient for the purpose
 - Better suited for storage

Evolution of programmable devices (PLDs)

- Programmable devices that implement logic functions are called programmable logic devices (PLDs/FPLDs)
- Programmable Logic Array (PLA)
 - Introduced in 1970's by Philips
 - First device specially developed for implementing logic functions
 - Has programmable wired AND plane followed by programmable wired OR plane
 - Versatile, easily produced functions in SOP form
 - Poor speed performance

Evolution of programmable devices (PLDs)

PAL Structure

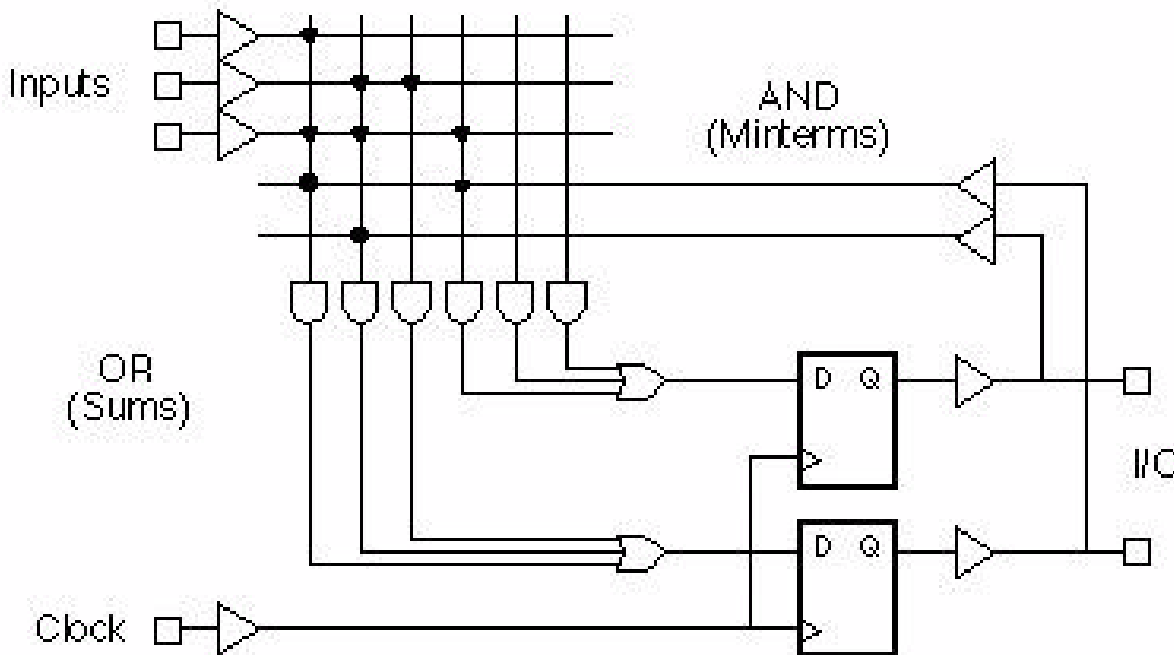


- This layout allows for a large number of logic functions to be synthesized in the SOP (and sometimes POS)
- Faster than PROMs
- Mostly used in control over a data path

Evolution of programmable devices (PLDs)

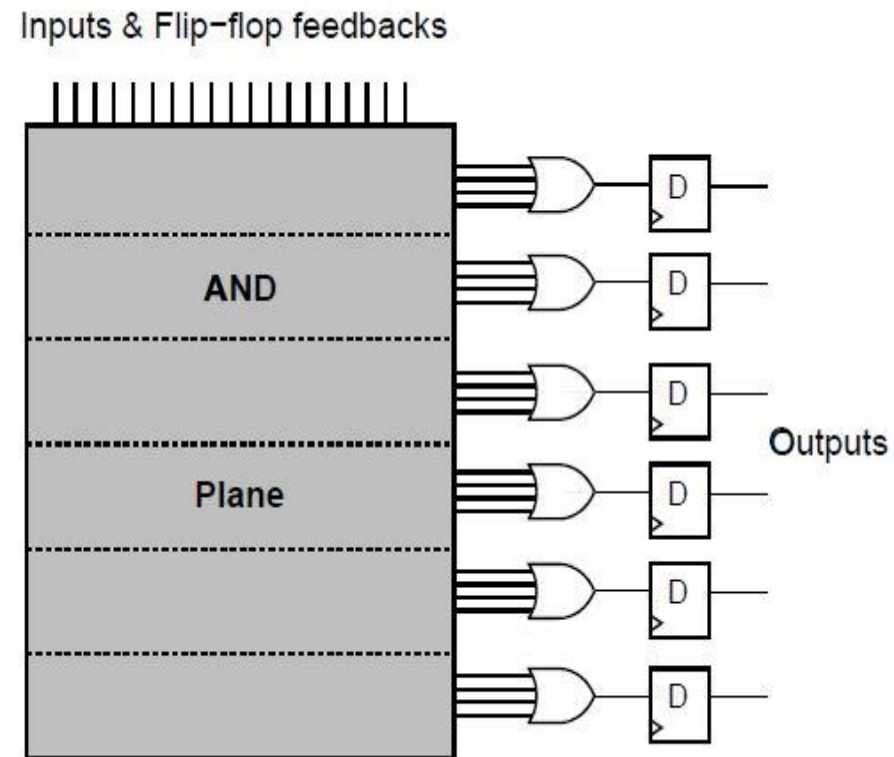
- PAL Structure

- Has programmable wired AND plane that feeds a fixed OR plane
- PALs usually contain flip-flops at the outputs to realize sequential circuits
- PAL devices consists of a small PROM core and additional output logic used to implement particular desired logic functions with few components.



Evolution of programmable devices (PLDs)

- All small PLDs are grouped into a category called Simple PLDs or SPLDs



Evolution of programmable devices (PLDs)

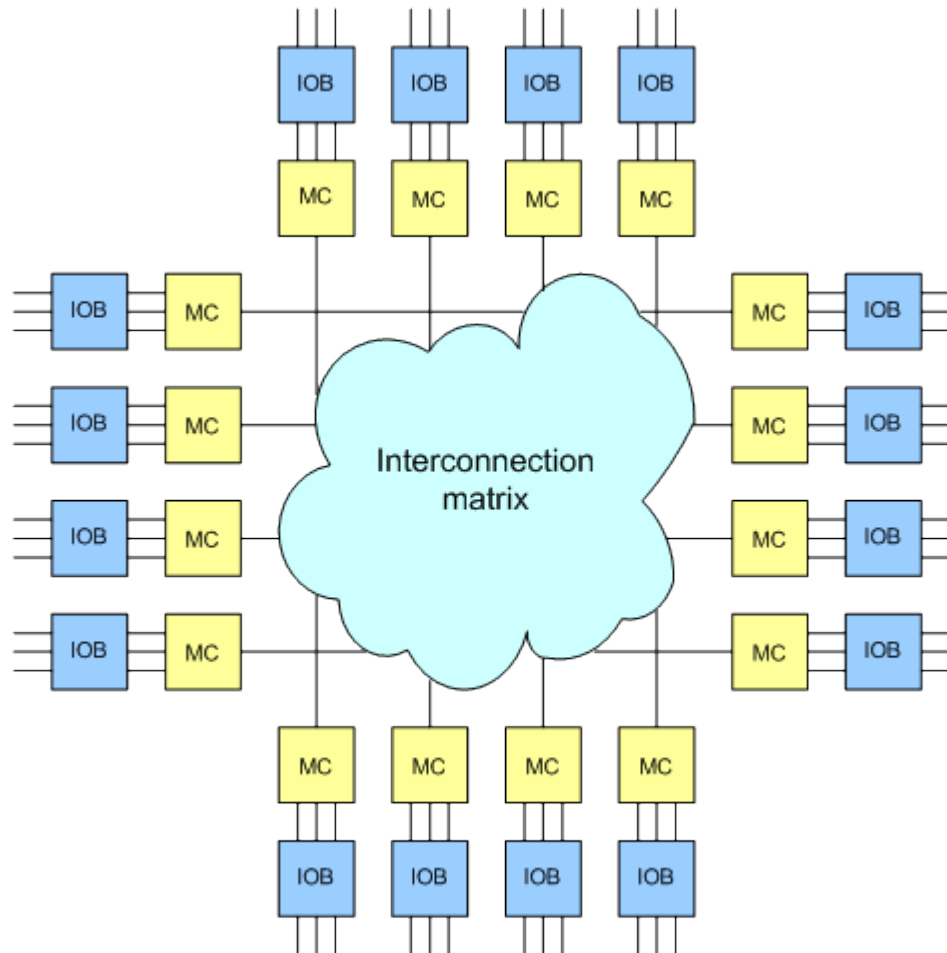
- There is a problem of capacity in SPLD architecture
 - The structure of the programmable logic-planes grow quickly in size as the number of inputs is increased
- The approach to provide higher capacity was to
 - integrate multiple SPLDs onto a single chip and
 - provide interconnect to which can be programmed to connect the SPLD blocks together
- A new category of FPLD's that evolved called CPLDs(Complex PLDs)

Evolution of programmable devices (PLDs)

- Pioneered by Altera
 - MAX 5000, MAX 7000 and MAX 9000 series
- CPLDs provide logic capacity of nearly 50 times of that of SPLDs
- Difficult to extend this architecture for higher logic capacity/density
- Gate Arrays was the way to go beyond CPLDs

Evolution of programmable devices (PLDs)

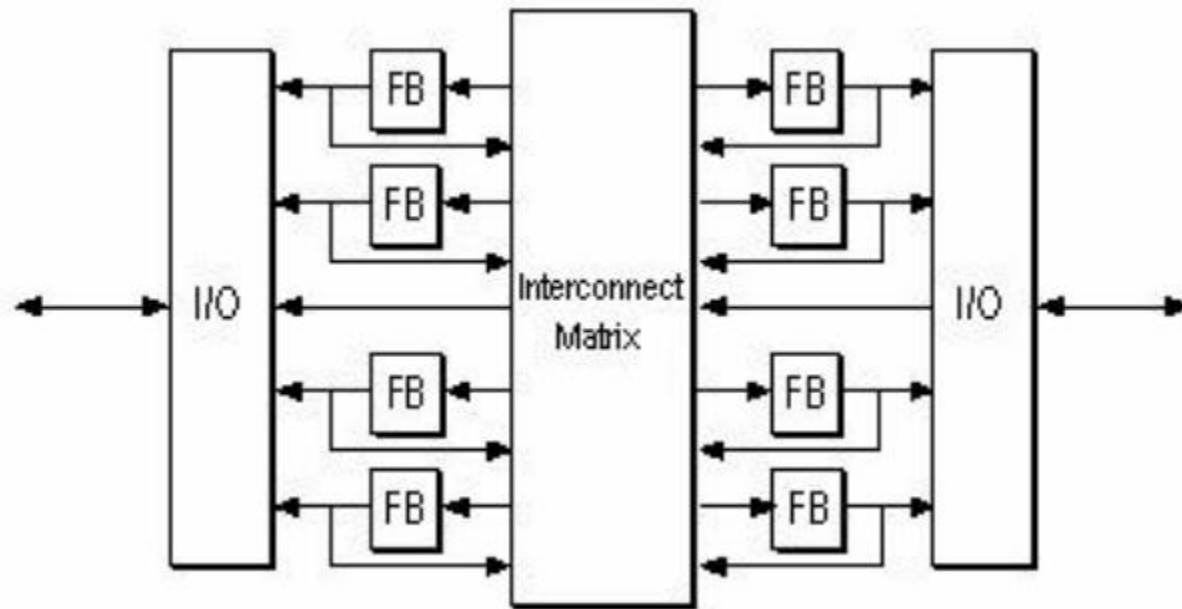
- CPLD Structure



COMPLEX PROGRAMMABLE LOGIC DEVICES (CPLD)

- Large number of PALs in a single chip, connected to each other through a cross-point switch
- They can handle much more complex logic

CPLD ARCHITECTURES

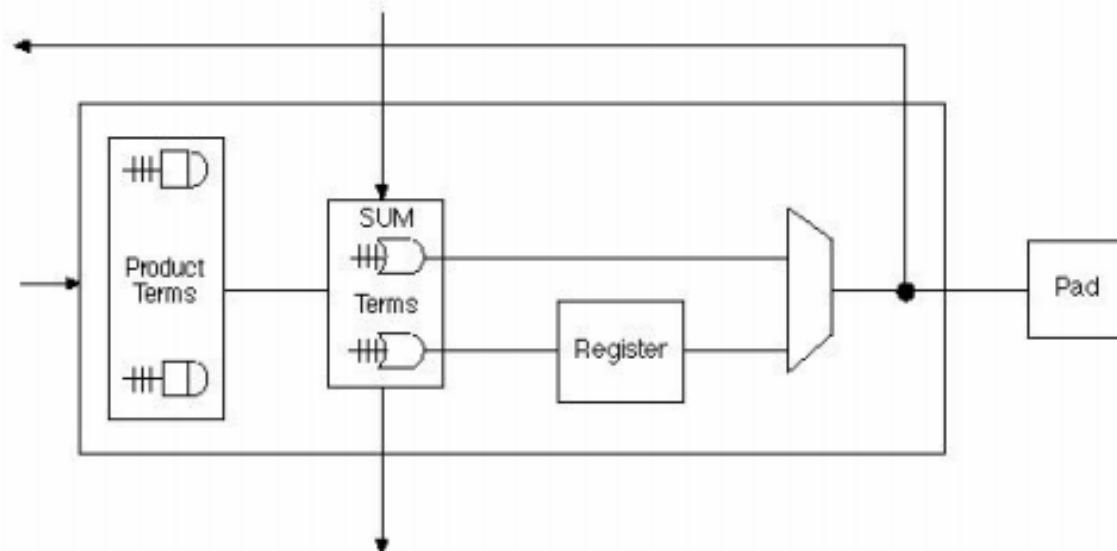


Function Blocks

- The AND plane can accept inputs from the I/O blocks, other function blocks, or feedback from the same function block
- The terms are then OR'd together using a fixed number of OR gates, and terms are selected via a large multiplexer
- The outputs of the mux can then be sent straight out of the block, or through a clocked flip-flop
- This particular block includes additional logic such as a selectable exclusive OR and a master reset signal, in addition to being able to program the polarity at different stages

I/O Blocks

- The I/O block is used to drive signals to the pins of the CPLD device
- Usually a flip-flop is included
- Also, some small amount of logic is included in the I/O block simply to add some more resources to the device

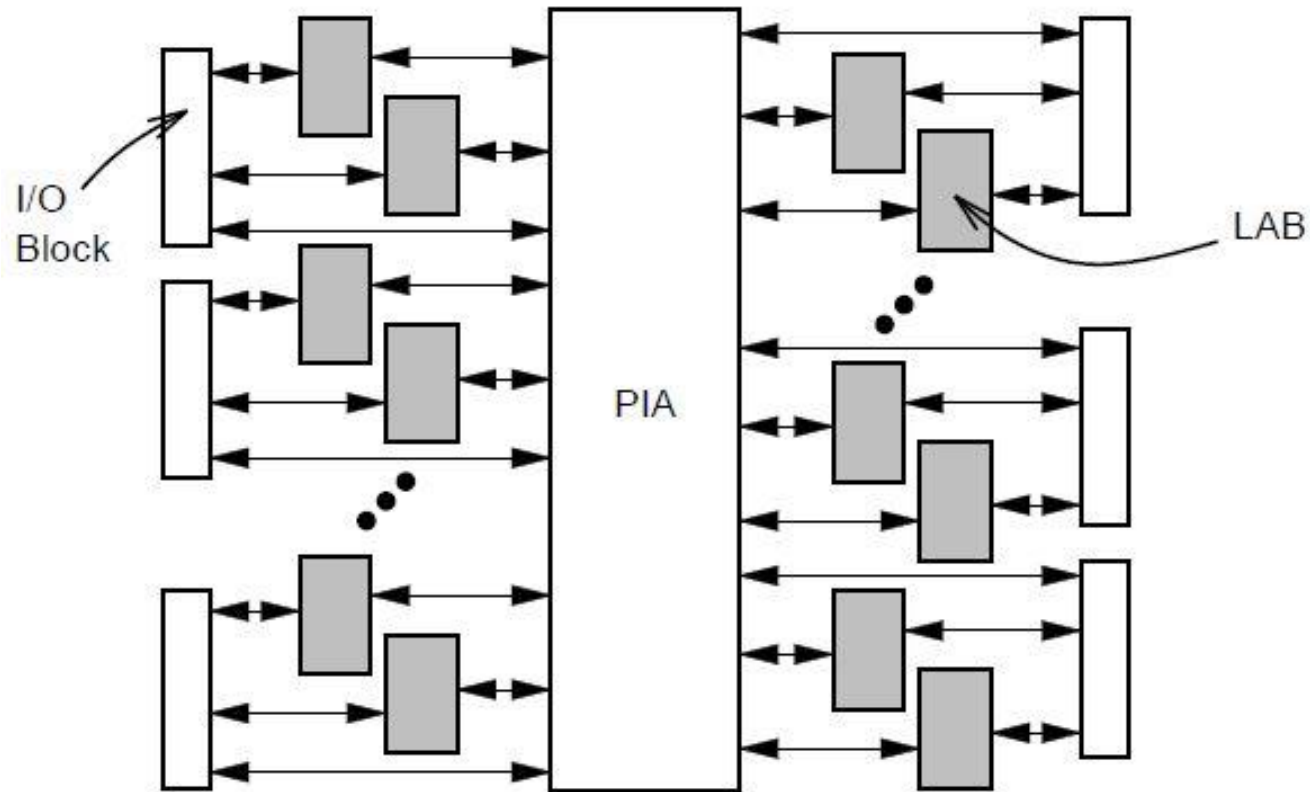


Interconnection Matrix

- The CPLD interconnect is a very large programmable switch matrix that allows signals from all parts of the device go to all other parts of the device.
- Programmable Elements
 - EPROM, EEPROM, Flash EPROM

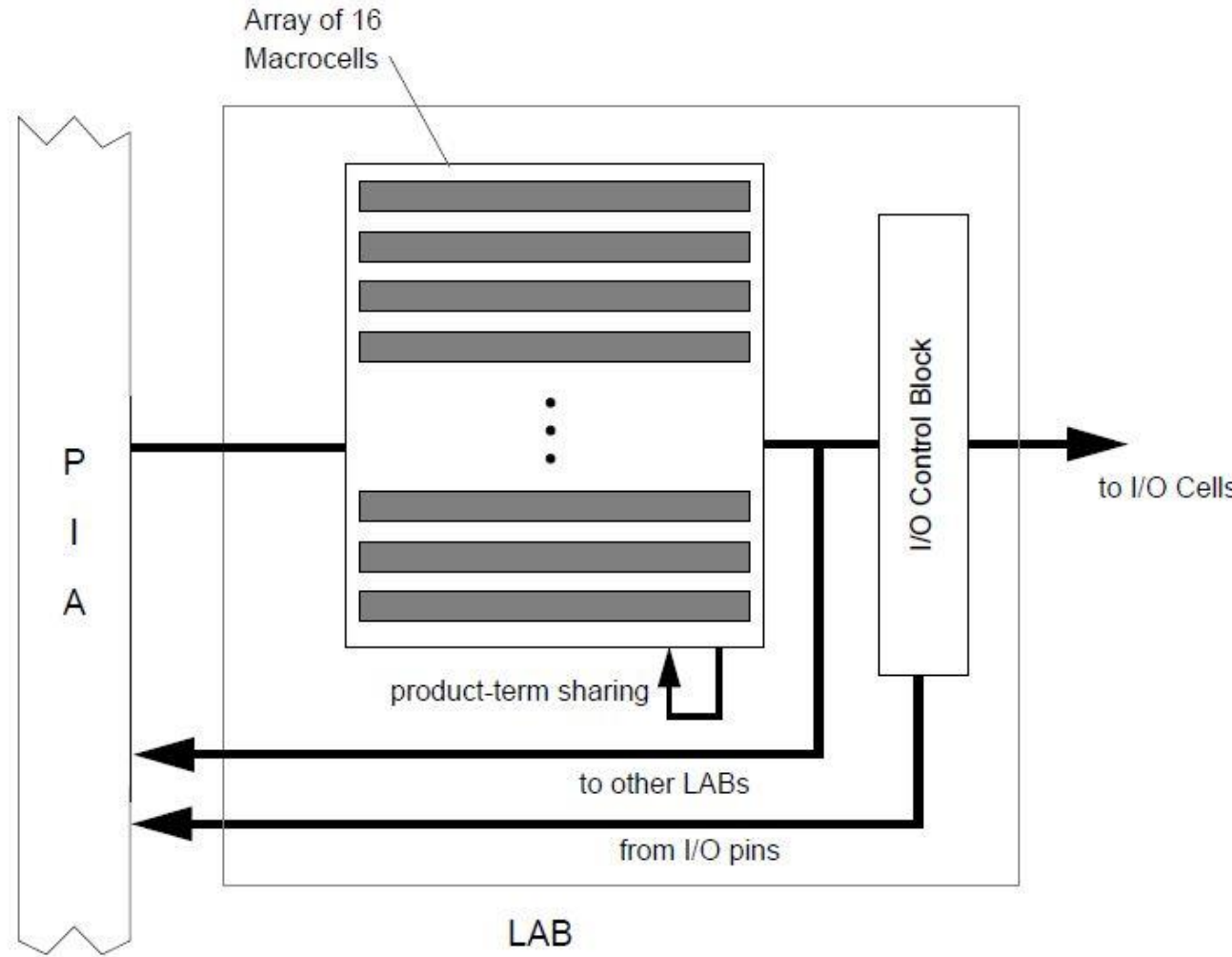
Overview of some CPLDs

- Altera MAX 7000
- EEPROM based



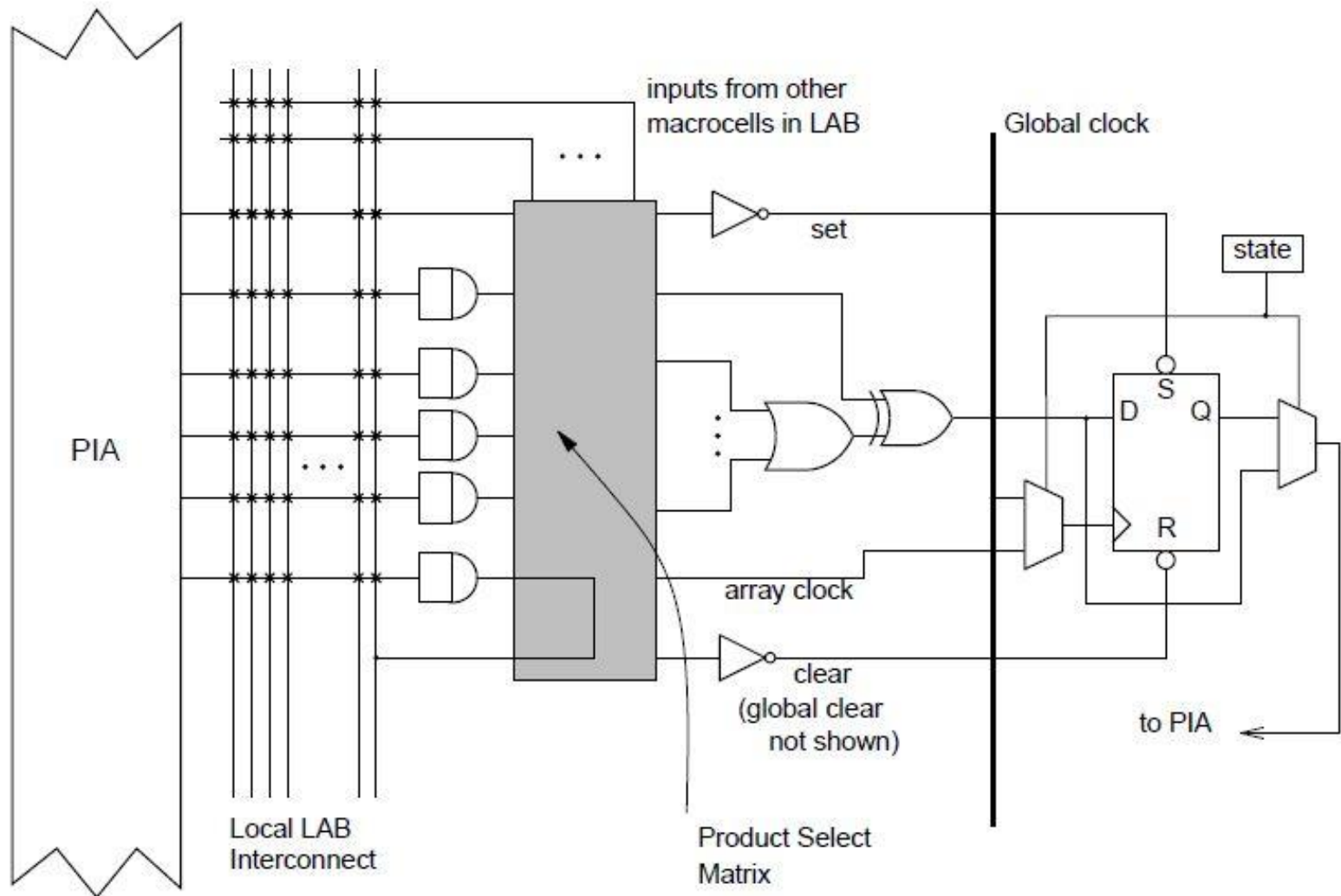
Overview of some CPLDs

- Altera MAX 7000 LAB



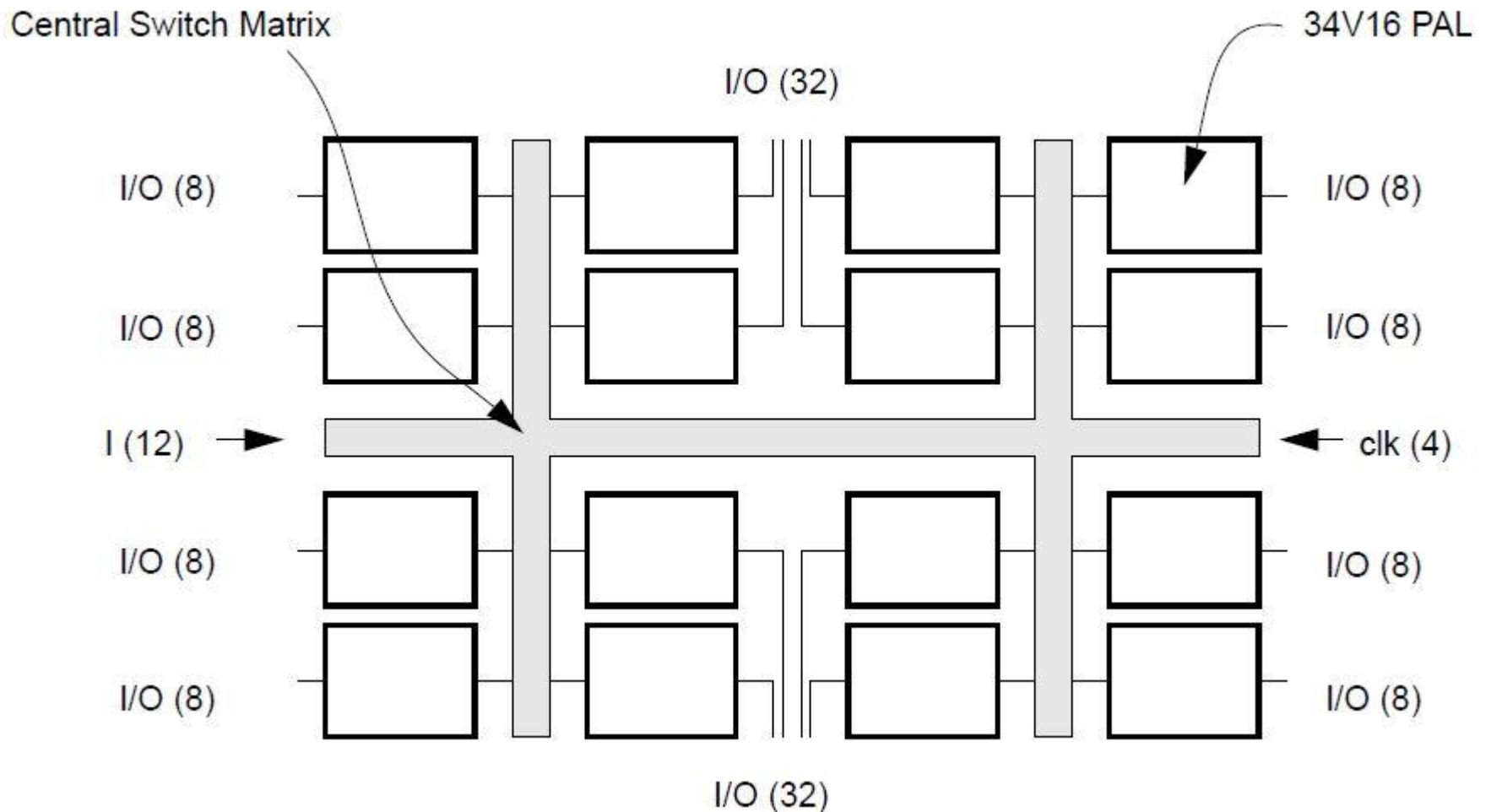
Overview of some CPLDs

- Altera MAX 7000 macrocell



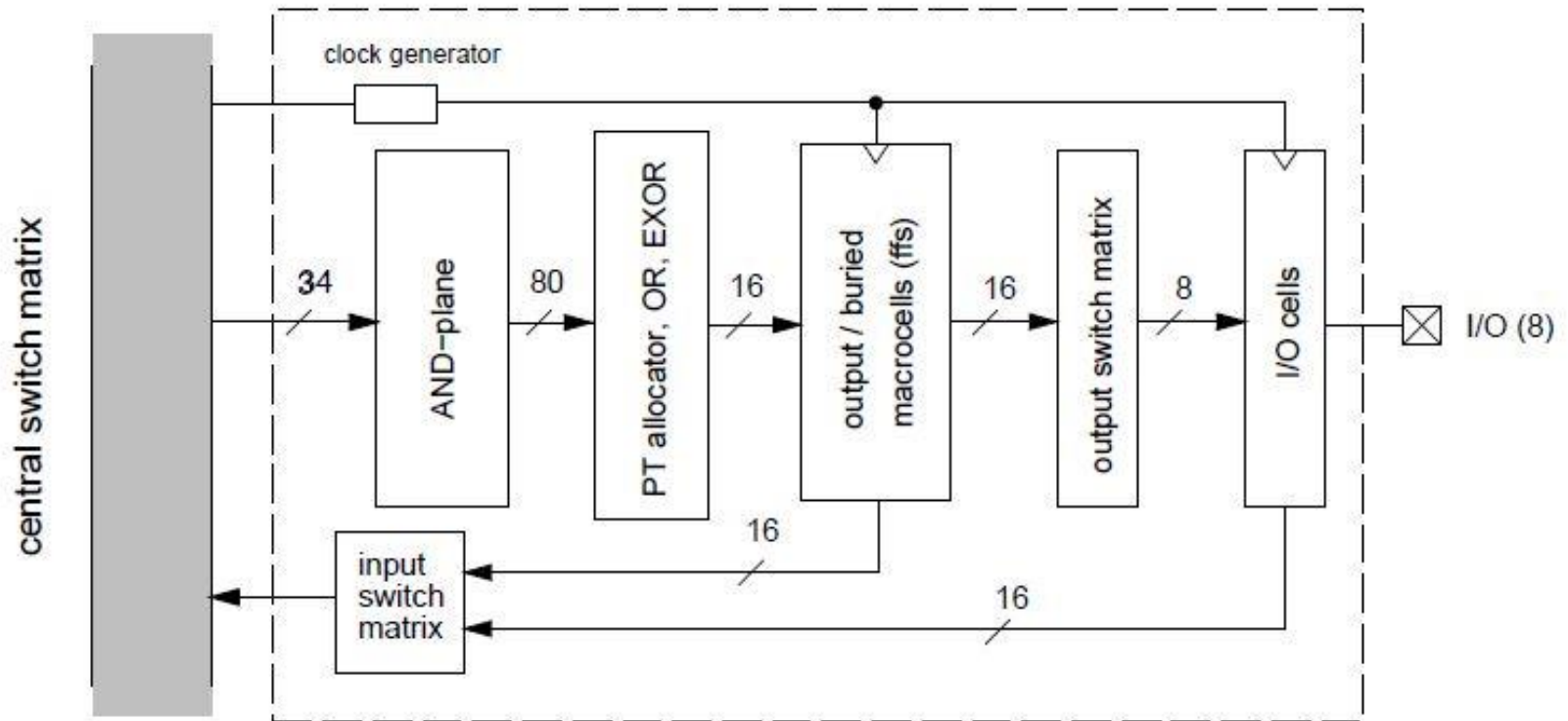
Overview of some CPLDs

- AMD Mach 4 CPLD, EEPROM based



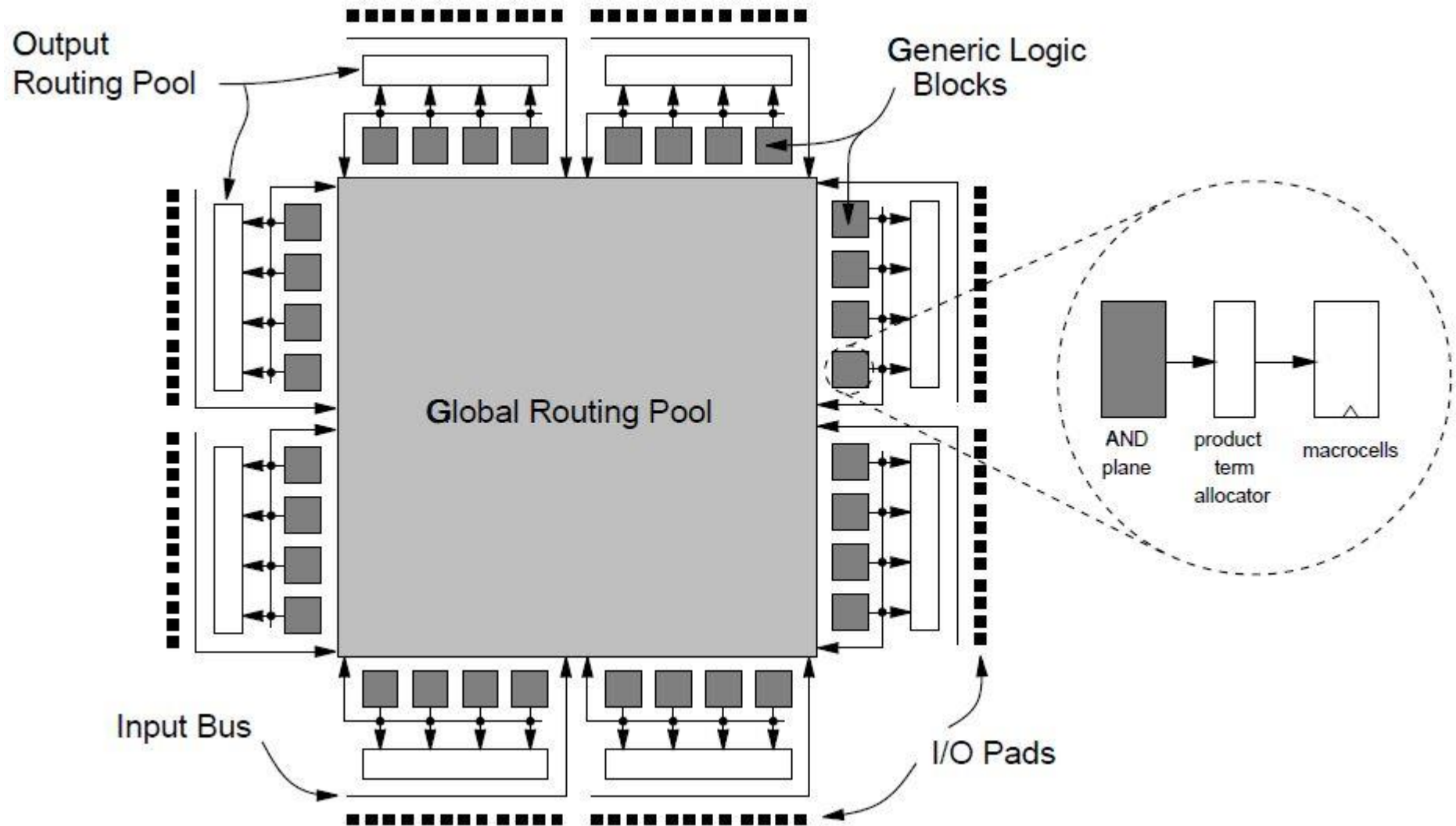
Overview of some CPLDs

- AMD Mach 4 PAL



Overview of some CPLDs

- Lattice PLSI

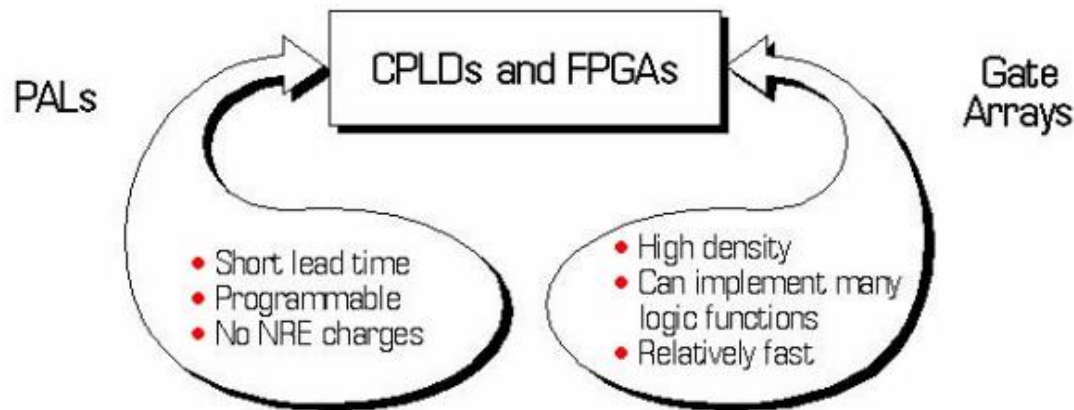


Applications of CPLDs

- CPLDs can realize reasonably complex designs, such as graphics controller, LAN controllers, UARTs, cache control, and many others
- As a general rule-of-thumb, circuits that can exploit wide AND/OR gates, and do not need a very large number of flip-flops are good candidates for implementation in CPLDs
- A significant advantage of CPLDs is that they provide simple design changes through re-programming (all commercial CPLD products are re-programmable)

CPLDs and FPGAs

- Provide flexibility and complexity of an ASIC but with the shorter turn-around time of a programmable device
- CPLDs and FPGAs bridge the gap between PALs and Masked Gate Arrays(MPGAs)
- CPLDs are as fast as PALs but more complex
- FPGAs approach the complexity of MPGAs but are still Programmable
- Difference between a large CPLD and a small FPGA is the presence of on-chip non-volatile memory in the CPLD



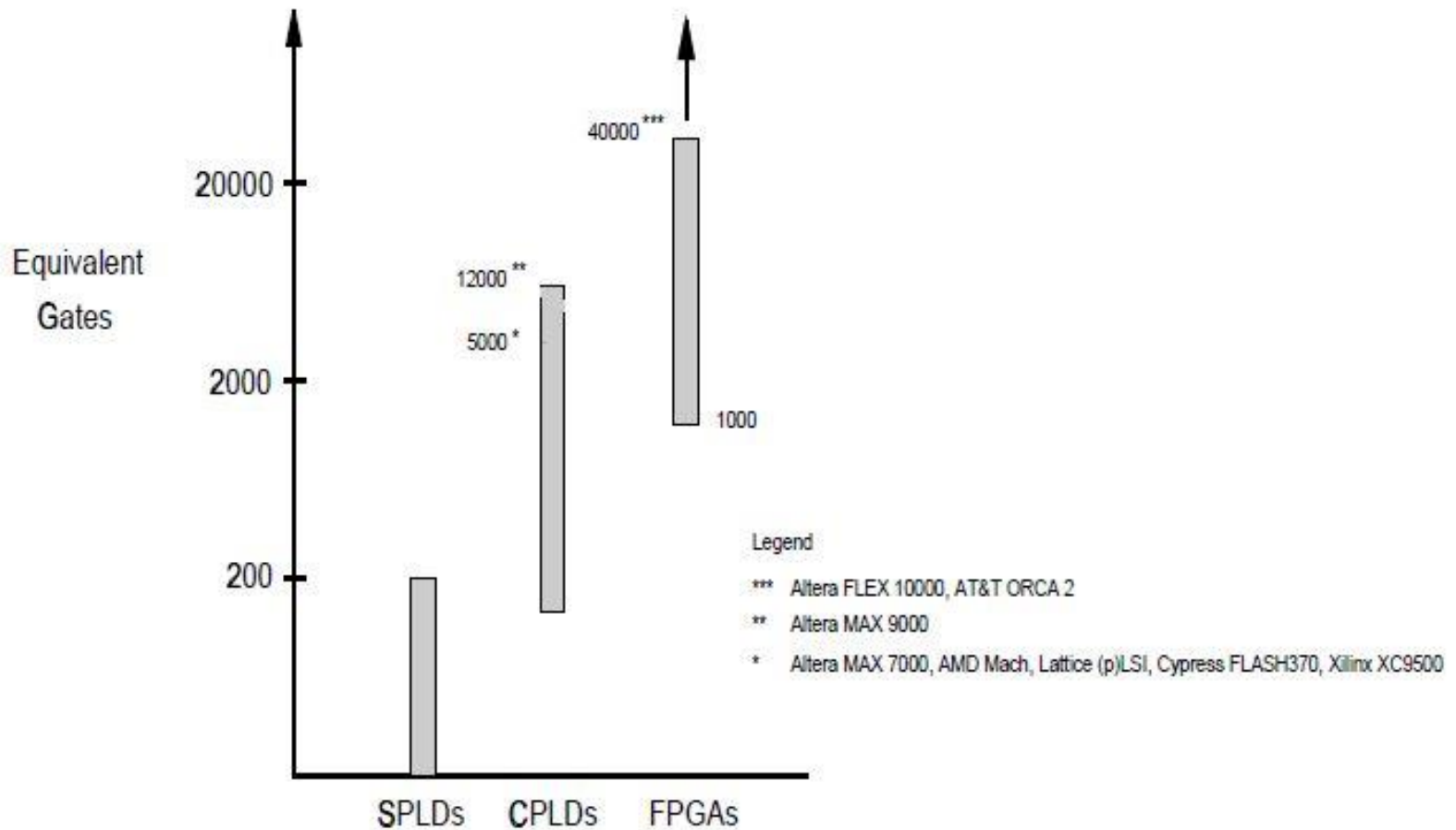
Gate Arrays

- Mask Programmable Gate Arrays (MPGAs)
 - Traditional gate arrays
 - Provide highest logic densities
 - Consist of array of pre-fabricated transistors provided by the manufacturer
 - Customization is done by performing final metallization steps as specified by the designer ->custom wiring
 - Disadvantages:
 - Programmed by the manufacturer
 - Longer design time as it involves manufacturing steps
 - Cost

FPGAs – why?

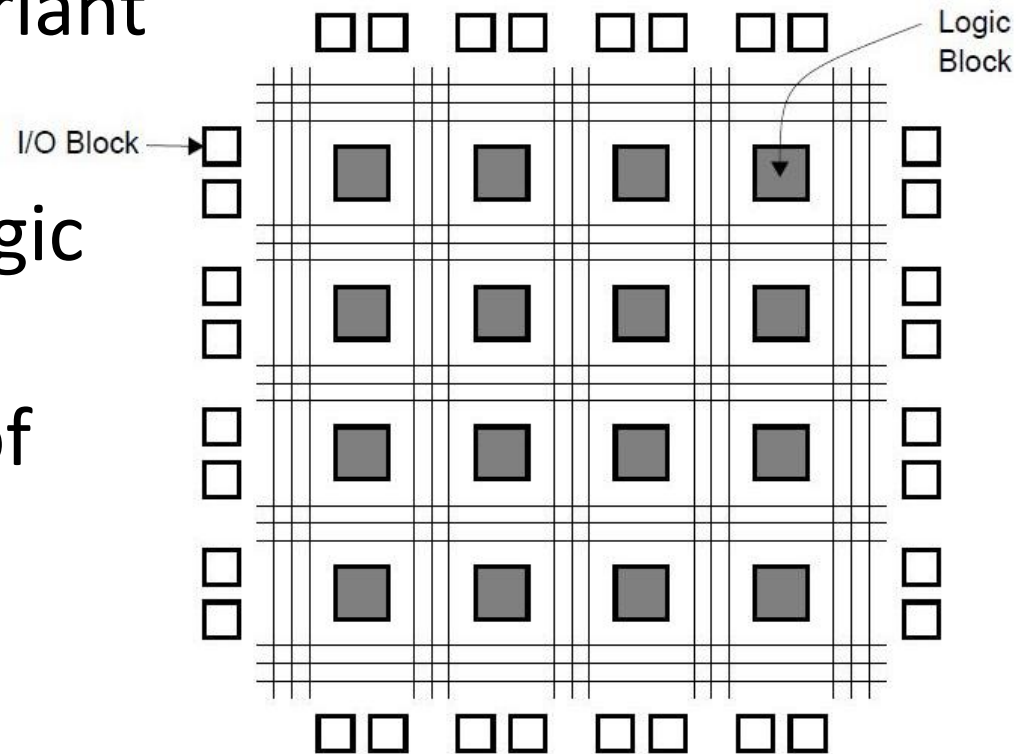
- FPGA stands for **F**ield **P**rogrammable **G**ate **A**rray
- Increasing number of designs use FPGAs for implementation
- Pro's
 - Readily available as standard parts
 - Faster implementation time
 - User/in circuit/field programmable
 - Simple tools/Design flow
 - Low risk, Design security
- Con's
 - Power, Performance, Cost, Size
- Traditional role – prototyping/glue logic

Logic capacity comparison



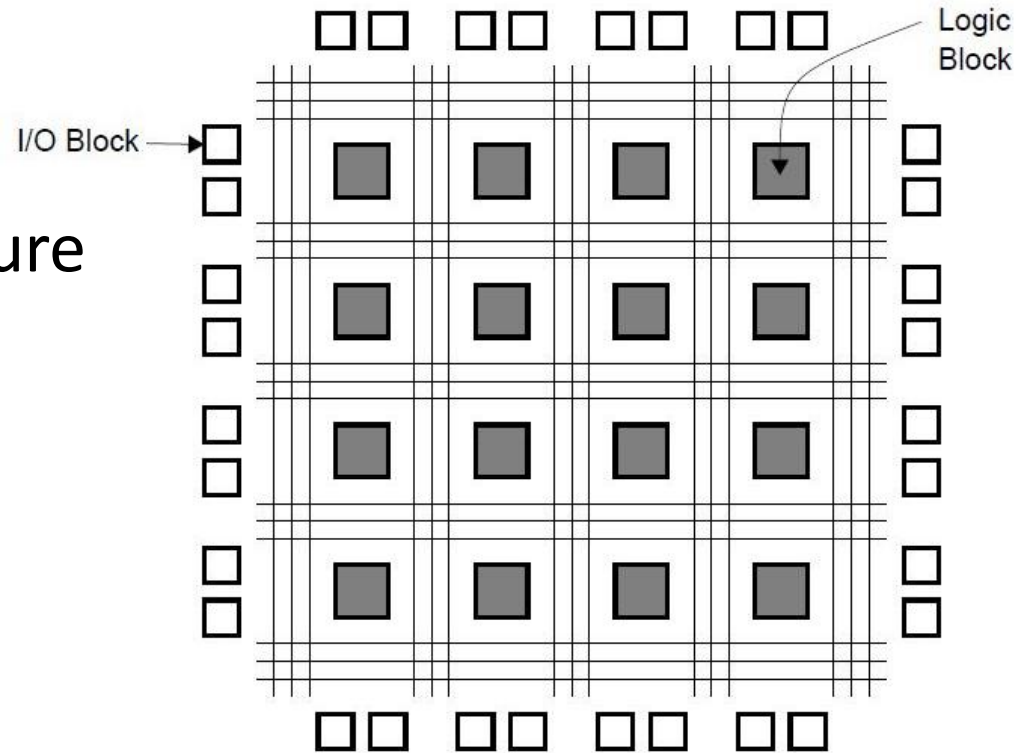
FPGAs

- User programmable variant of MPGAs
- FPGAs combine high logic capacity of MPGAs and user programmability of PLD's

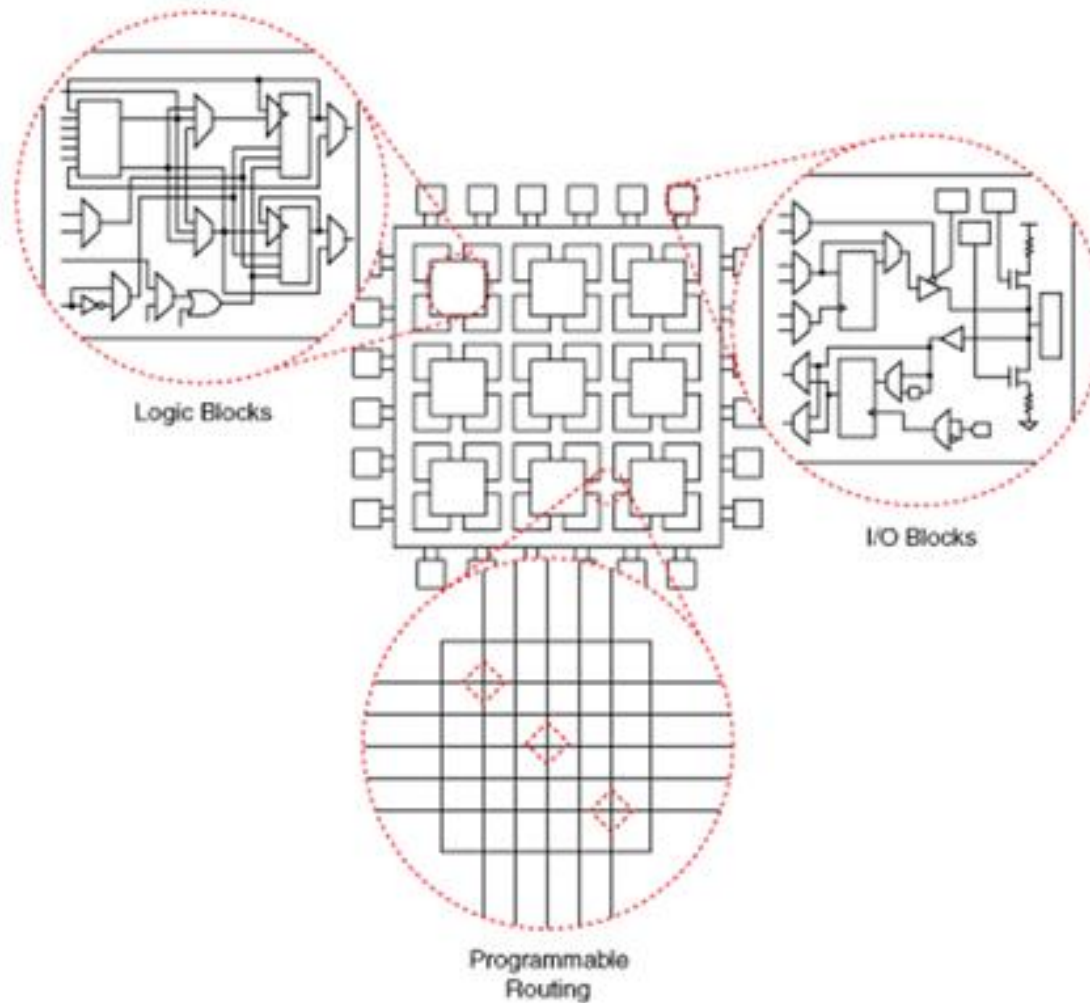


FPGA Architecture

- Has programmable
 - Logic resources
 - Interconnection structure
 - I/O blocks



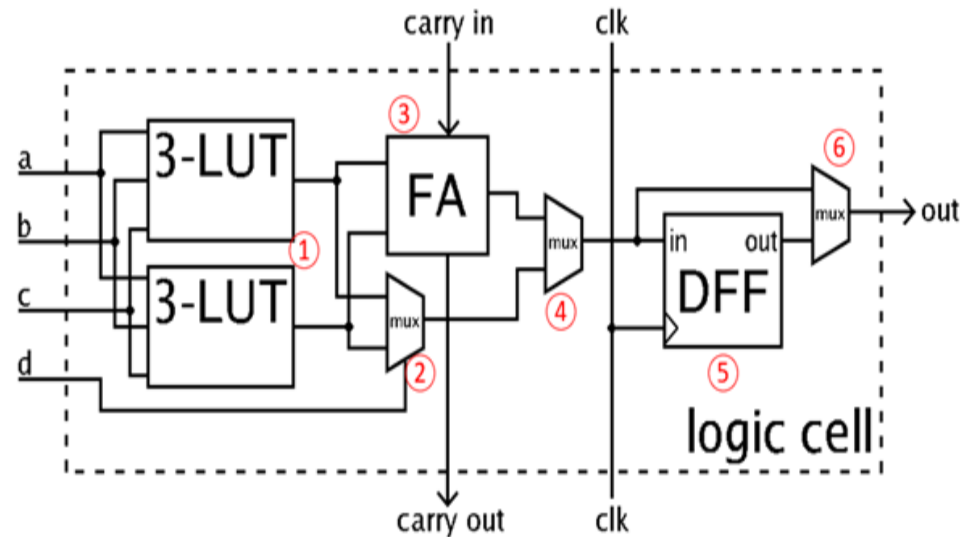
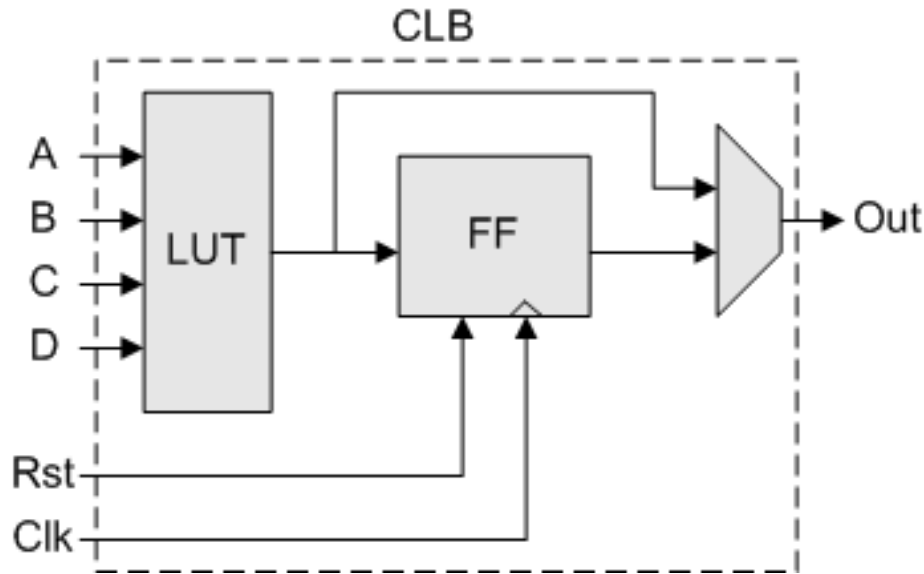
FPGA Architecture



Logic Resources

- Are organized as multiple copies of Configurable Logic Blocks (CLBs)
- Called differently depending on the vendor : for e.g. CLB/PLB/LE..
- Logic blocks typically consist of combination of
 - Look-up tables (LUTs) which can store a predefined list of logic outputs for any combination of inputs: LUTs with four to six input bits are widely used
 - Combinational gates like basic NAND gates or XOR gates
 - Multiplexers, Full Adders, Wide fan-in AND-OR structure
 - Flip-flops, Block RAMs
- Additionally
 - Clock generation circuits, Clock circuitry for driving the clock signals to each logic block
 - High speed transceivers
 - DSP modules, CPU cores
 - ALUs, memory, and decoders may be available.

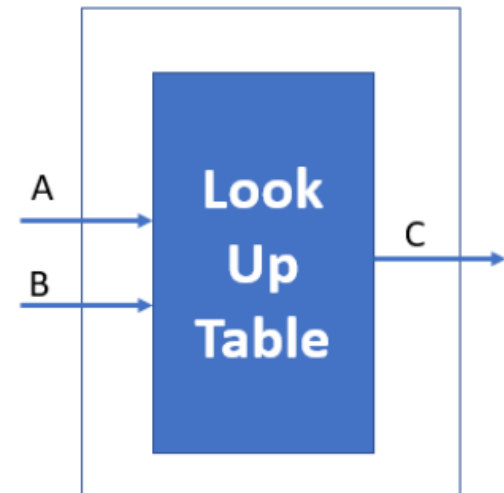
Configurable logic blocks(CLBs)



Lookup Table (LUT)

- A Lookup Table, as the name suggests, is an actual table that generates an output based on the inputs.
- Consider an AND gate
- This table is stored in a small RAM.
- Inputs A and B are the address pins and C is the data pin.
- Every time your address pins are changing they are pointing at a different address entry and they are “reading out” the result which is 0 or 1 based on the inputs.

Input A	Input B	Output C
0	0	0
0	1	0
1	0	0
1	1	1

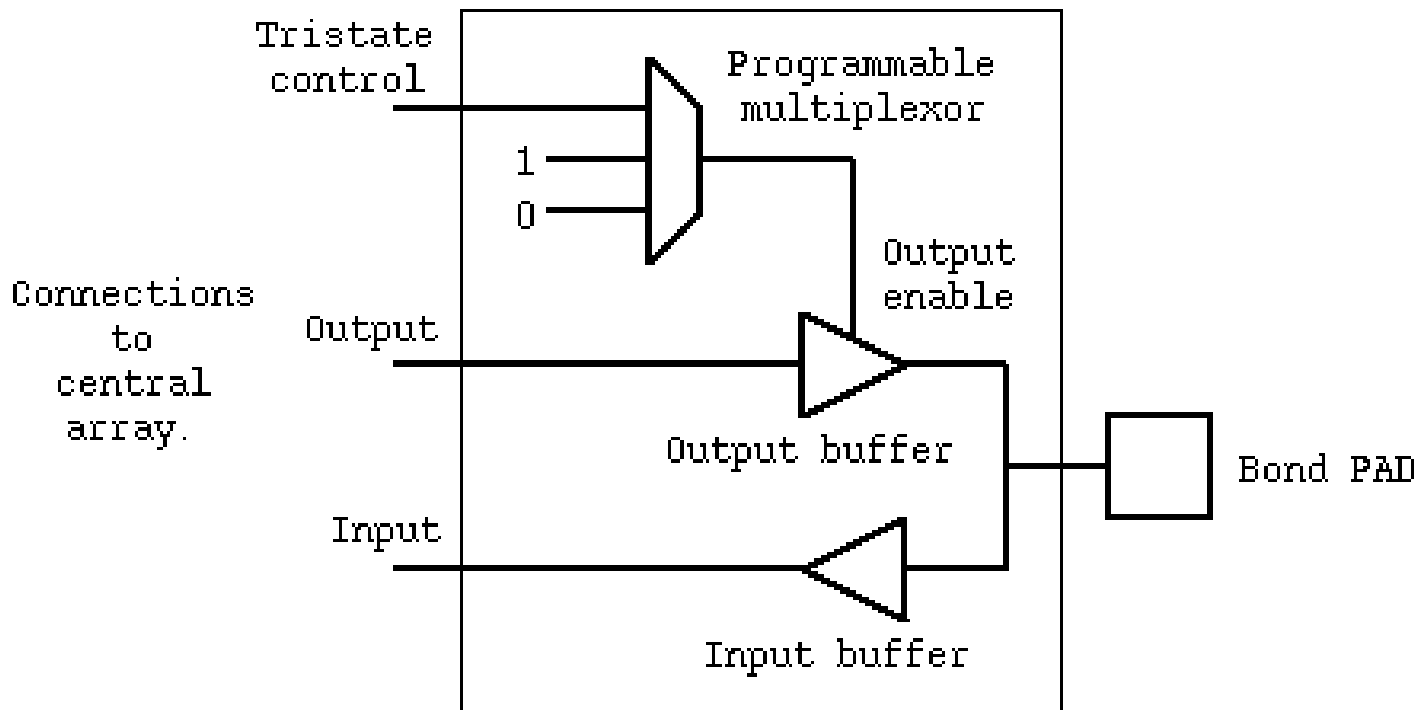


Clock Circuitry

- Special I/O blocks with special high drive clock buffers (clock drivers) are distributed
- These buffers are connected to clock input pads and drive the clock signals onto the global clock lines

Configurable I/O BLOCKS

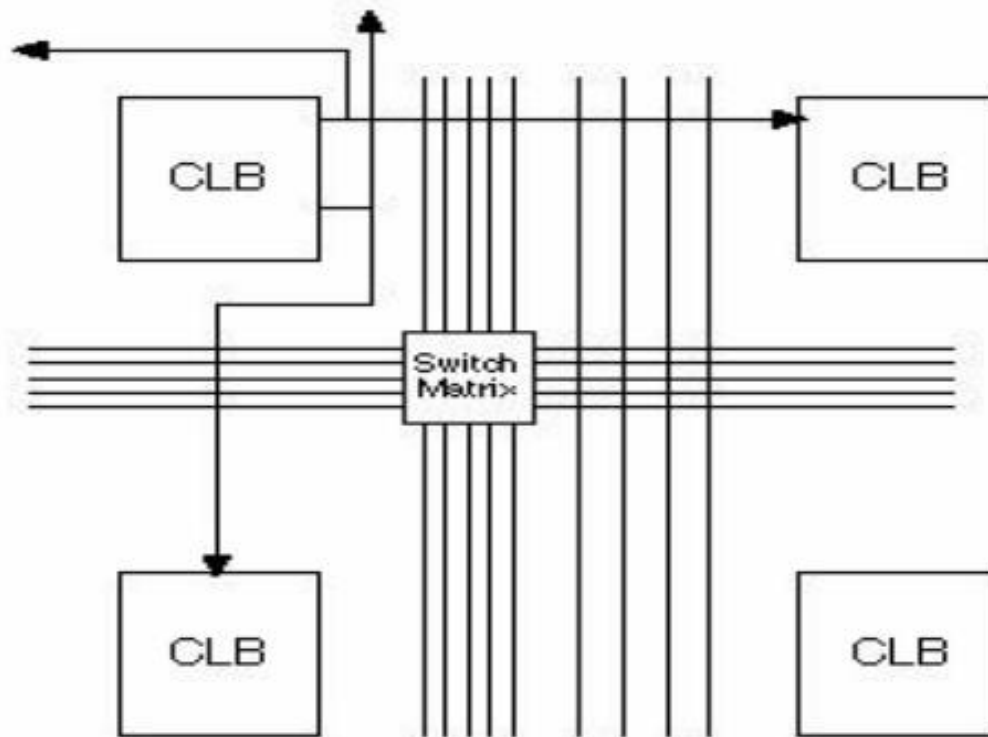
- Used to bring signals onto the chip and send them back again
- It consists of an input buffer and an output buffer with three state and open collector output controls.
- Typically there are pull up resistors on the outputs and sometimes pull down resistors.
- The polarity of the output can usually be programmed for active high or active low output



Programmable Interconnection structure

- Provides connection between wires, logic blocks, I/O blocks..
- We pay for this when we buy an FPGA !
- Wires
 - Long lines to connect distant logic blocks
 - Short lines connect neighboring blocks with each other
 - Dedicated clock trees are used to distribute synchronization signals
 - these lines have large fanout and little skew and jitter
 - Dedicated set/reset lines are used to reset all flip-flops in the FPGA.
 - Also has tri-state buffers
- Switches
 - Switch matrix connects long and short lines in a specific way

Programmable Interconnect



Small vs. Large granularity

- Small grain FPGAs resemble ASIC gate arrays
- Small elements can be connected to make larger functions
- Large grain FPGA – CLB can contain two or more flip-flops

Small Granularity

better utilization

direct conversion to ASIC

Large Granularity

fewer levels of logic

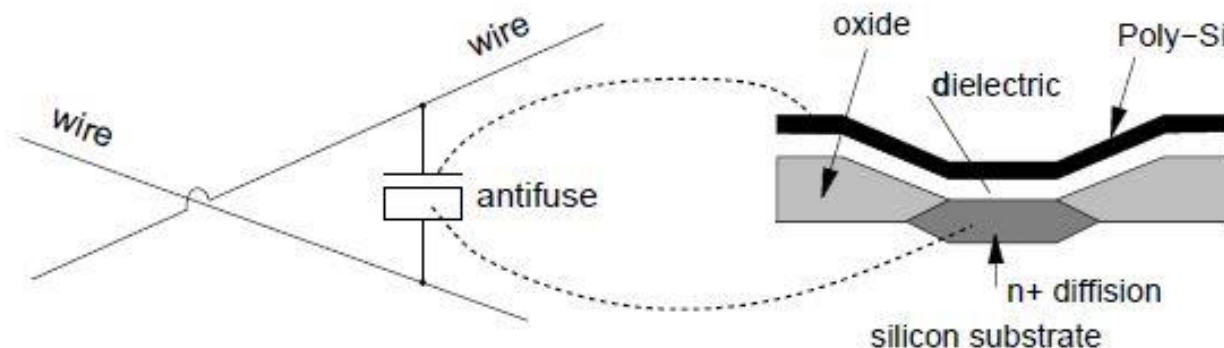
less interconnect delay

Programming Technologies

- Programming technologies enable user programmability
- Conventional programming technologies used in memory devices are generally used
- Few of these are
 - Fuse (Used in PLA's/PAL's)
 - Anti-fuse
 - EPROM, EEPROM, FLASH memory switches
 - SRAM based switches

Programming Technologies – Anti-fuse

- Anti-fuses consist of microscopic structures that normally make no connection.
- When current is passed during device programming, the two sides of anti-fuse form a low resistance connection
- Actel antifuse – PLICE ,18V-10mA, 100M \rightarrow 1K Ω



Programming Technologies – Anti-fuse

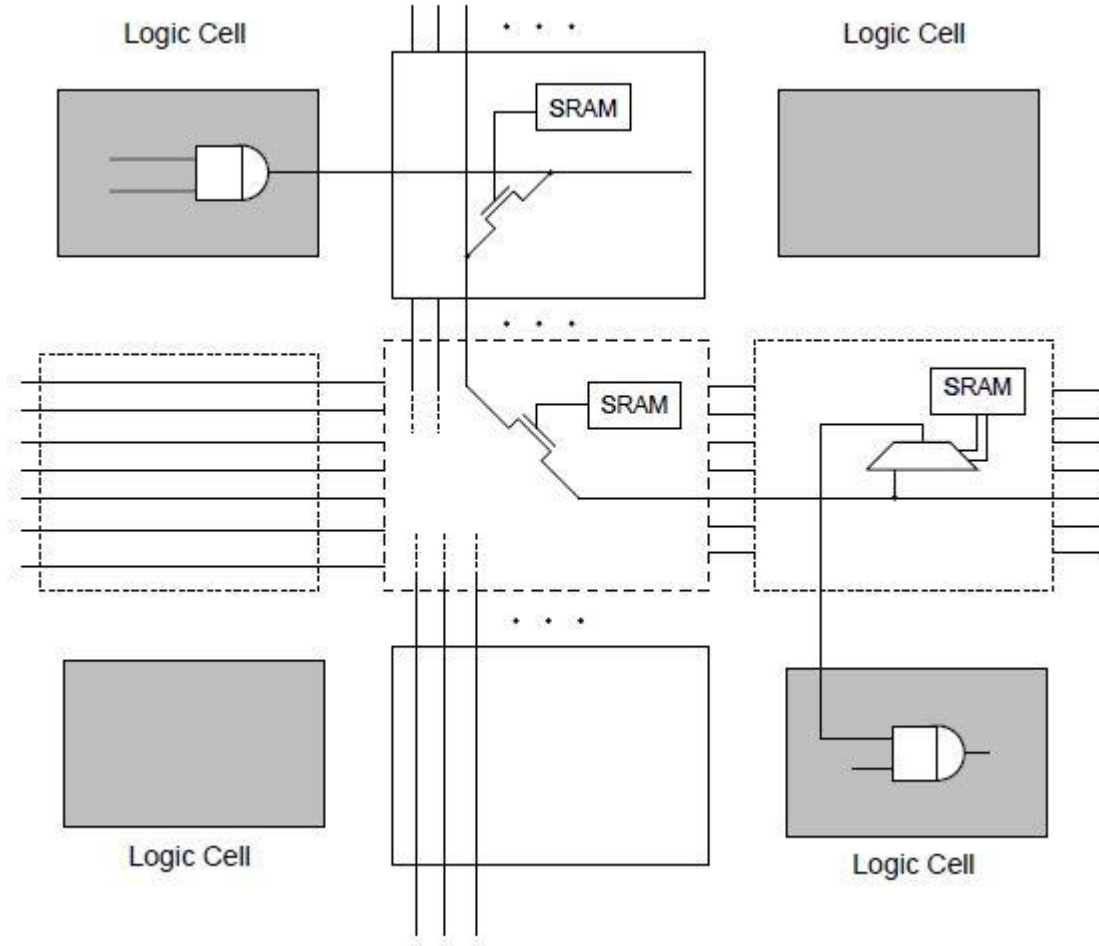
- Suitable for FPGAs because they can be built using modified CMOS technology
- Advantages:
 - Non-volatile
 - Delays due to routing are very small – they tend to be faster
- Disadvantages:
 - Complex fabrication process
 - Require an external programmer to program them
 - Once programmed, they cannot be changed

Programming Technologies – SRAM

- FPGA connections are achieved using pass-transistors, transmission gates, or multiplexers that are controlled by SRAM cells - Writing the bit with a zero turns off a switch, while writing with a one turns on a switch.
- Advantages:
 - They use a standard fabrication process that chip fabrication plants are familiar with
 - Can be optimized for better performance as fabrication technology evolves
 - Can be reprogrammed any number of times
- Disadvantages:
 - Volatile, which means a power glitch could potentially change it.
 - Large routing delays

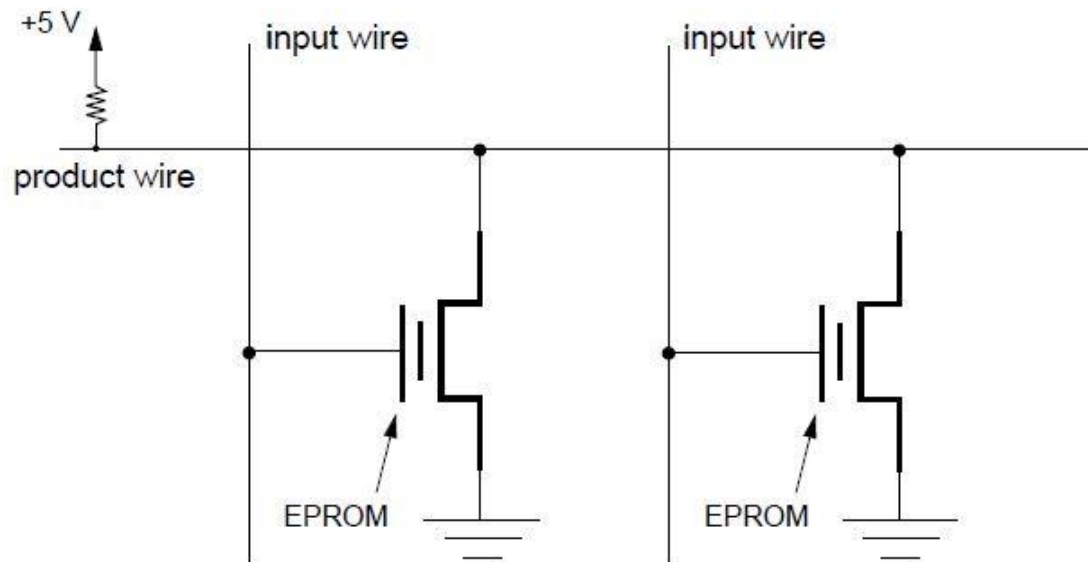
Programming Technologies – SRAM

- SRAM switch
 - Popular in FPGA's
 - Xilinx
- SRAM+ internal FLASH



Programming Technologies – EEPROM/FLASH

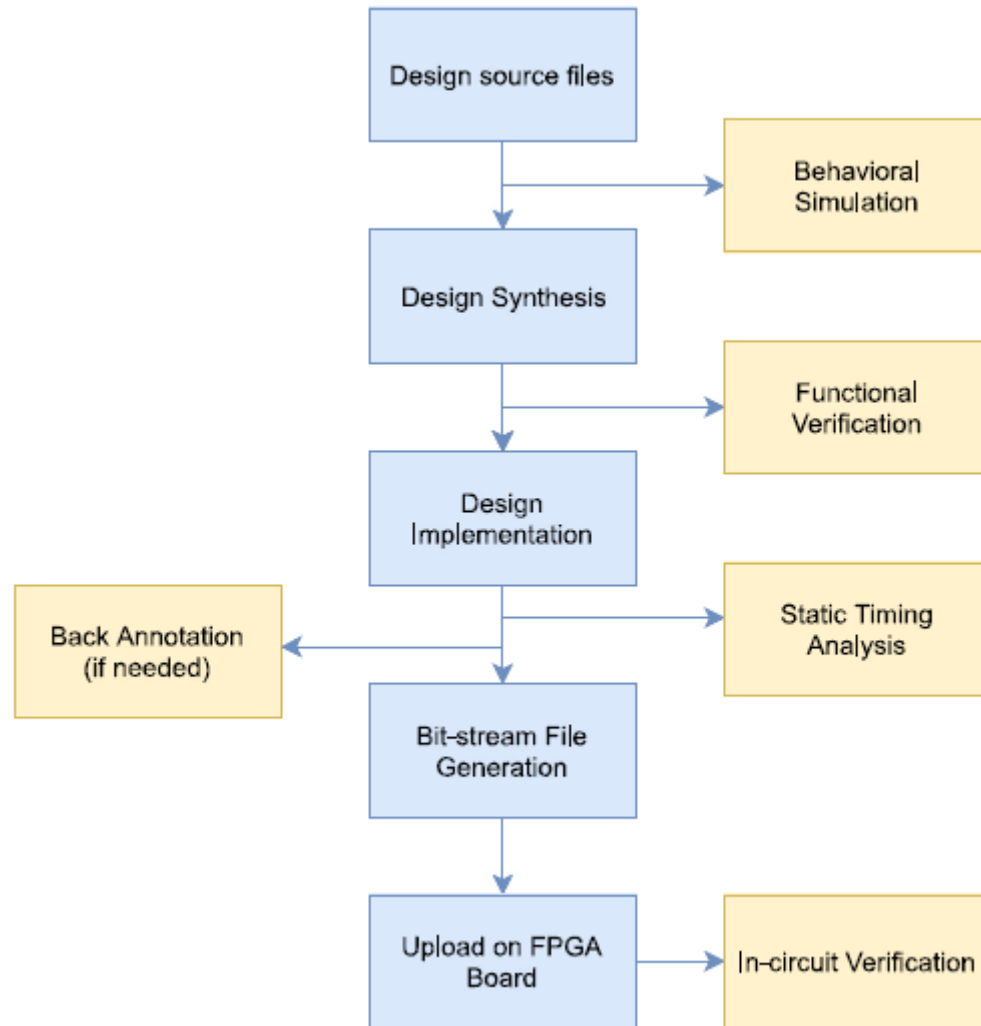
- EPROM, EEPROM, FLASH memory switches
 - Popular in CPLD's also in some SPLD's
 - Actel, Altera



Programming technologies - Summary

Type	Re - Programmability	Volatility	Technology
Fuse	No	No	Bipolar
EPROM	Yes, UV erasable	No	UVC MOS
EEPROM (Flash)	Yes, In-circuit	No	EECMOS
Antifuse	No	No	Modified CMOS
SRAM	Yes, In-circuit	Yes	CMOS

FPGA DESIGN FLOW



Design Entry

- Choosing a design entry method
- For smaller chips - schematic entry
- For larger designs – HDL : portability, flexibility, and readability

Modelling the design

- Top-down design
- Use logic that fits well with the architecture of the device chosen
- Macros
- Synchronous design
- Care must be taken to avoid
 - meta-stability
 - Floating nodes
 - bus contention

Simulation

- Small sections of design simulated separately before connecting them up to larger section
- Many iterations of design and simulation needed to get correct functionality
- Design review after design and simulation
- Design verification techniques can be used

Synthesis

- The design is taken to synthesis step post functional verification
- Synthesis software translates HDL design to a level which can mapped to the logic blocs in an FPGA
- Place and Route
- Chip layout – resulting in a real physical design for real chip

Post layout simulation

- Post layout, the design is re-simulated with new timing numbers produced by actual layout
- For any deviations from the specifications, the following strategies can be employed
 - For significant problems, sections of FPGA need to be redesigned
 - For marginal timing violations, perform another synthesis with better constraints
 - For design slightly larger than FPGA, another place and route with better constraints

Testing

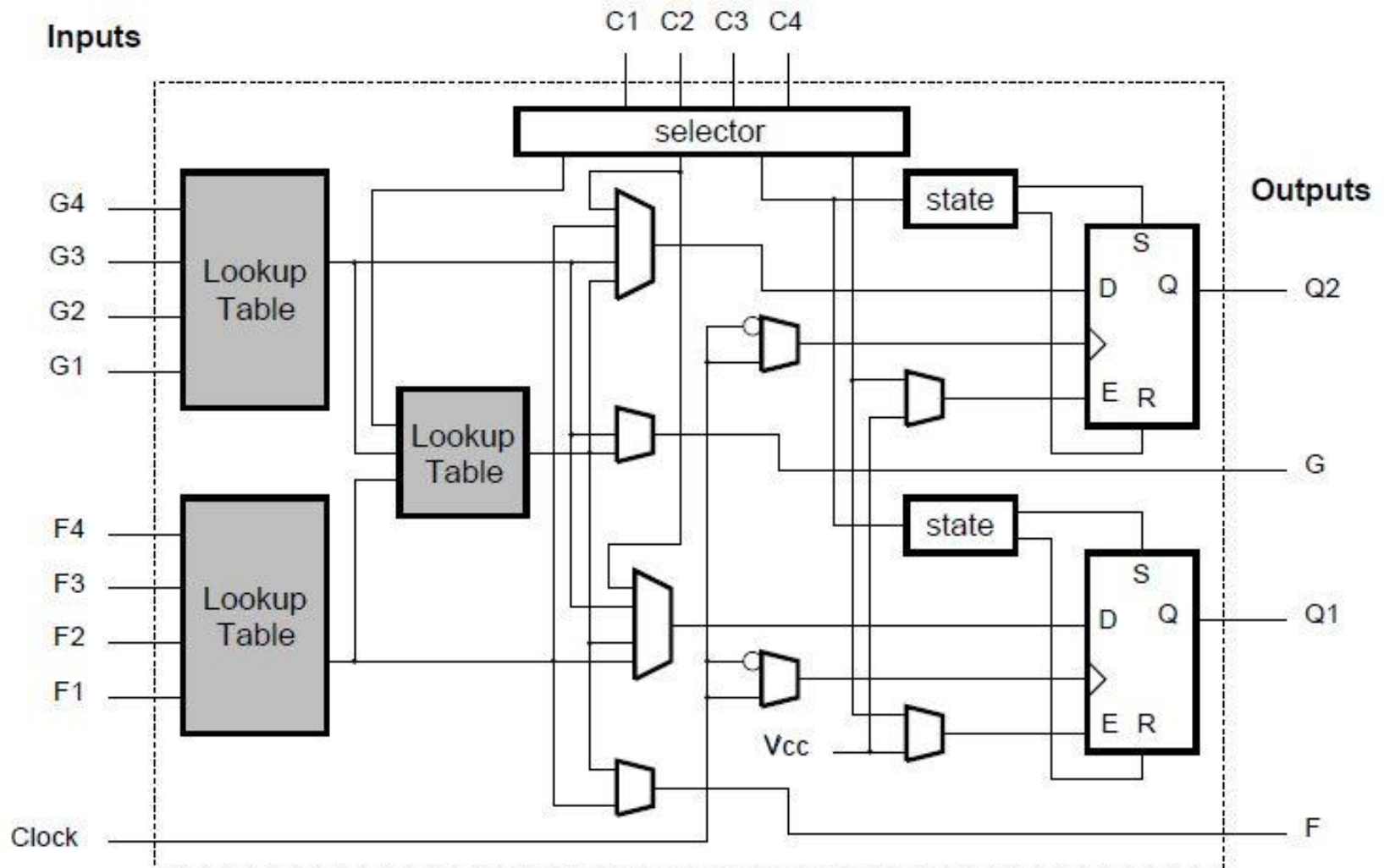
- Program the device by downloading the bit-stream generated by the FPGA software onto the device -> Prototypes
- Verify the prototypes either by board level or system level testing

Overview of some FPGAs

- In general FPGAs have the following classes of architecture
 - Symmetrical array
 - Row based
 - Hierarchical PLDs

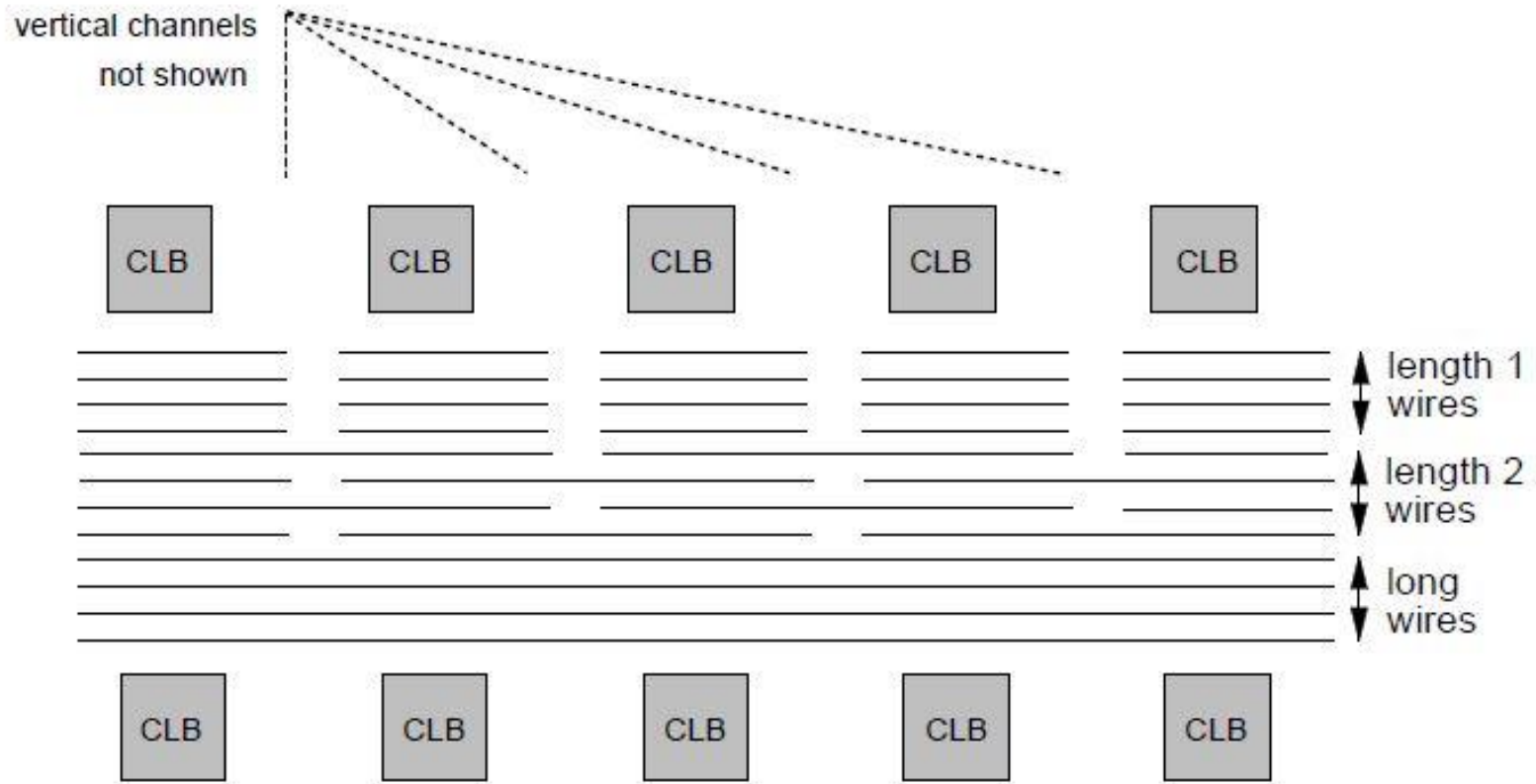
Overview of some FPGAs

- Xilinx XC4000 CLB, SRAM based



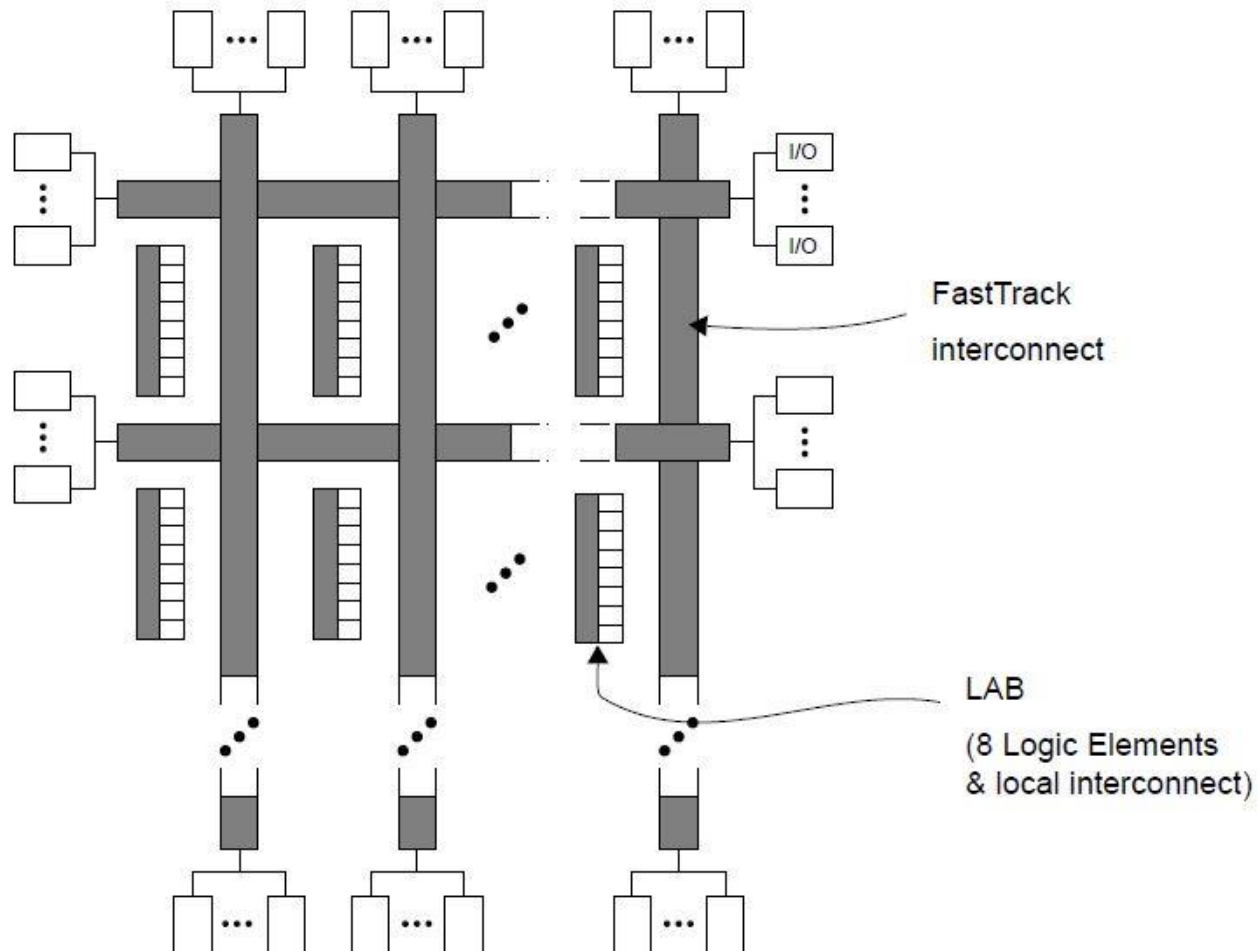
Overview of some FPGAs

- Xilinx XC4000 Interconnect



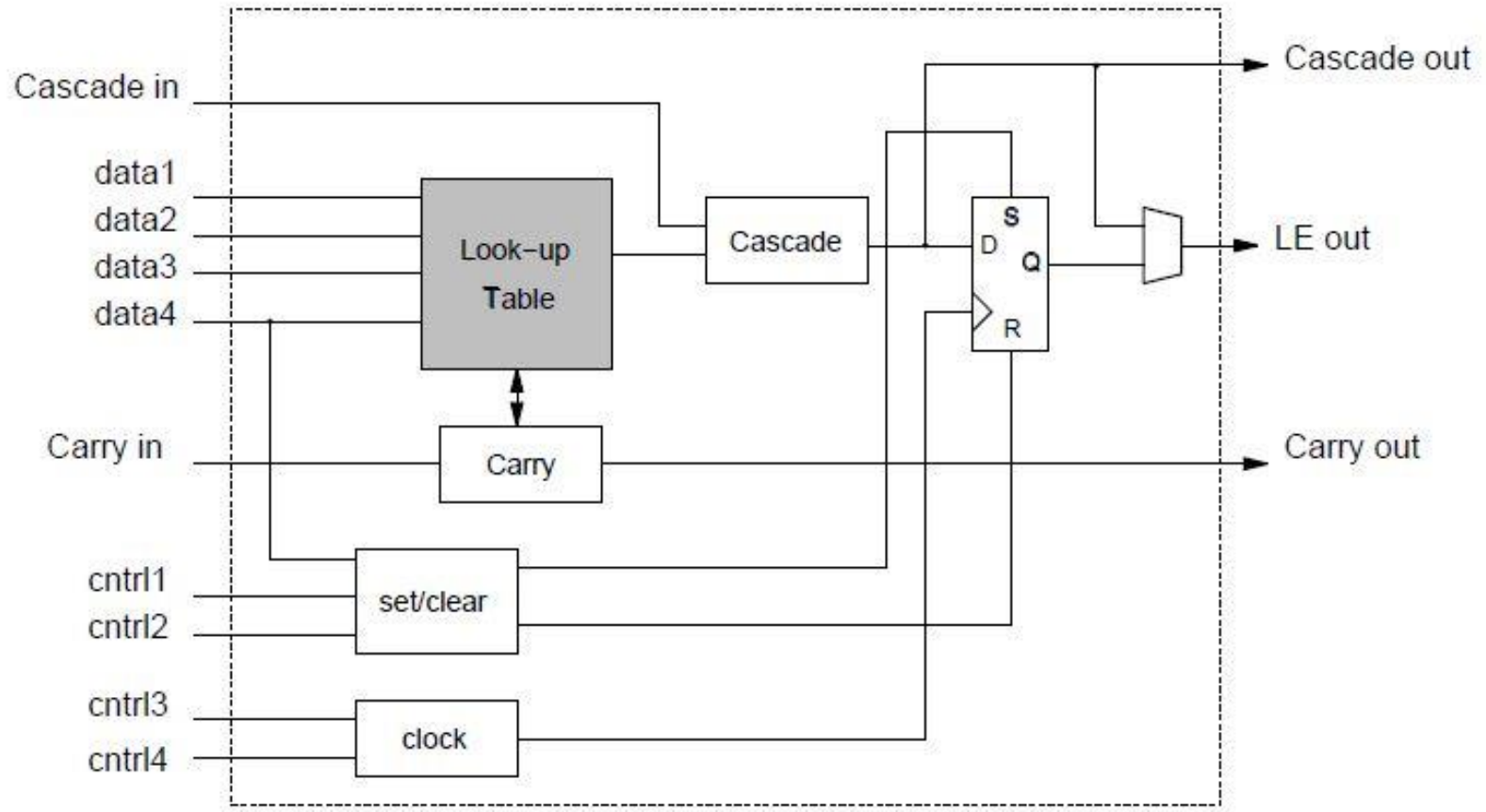
Overview of some FPGAs

- Altera Flex8K FPGA



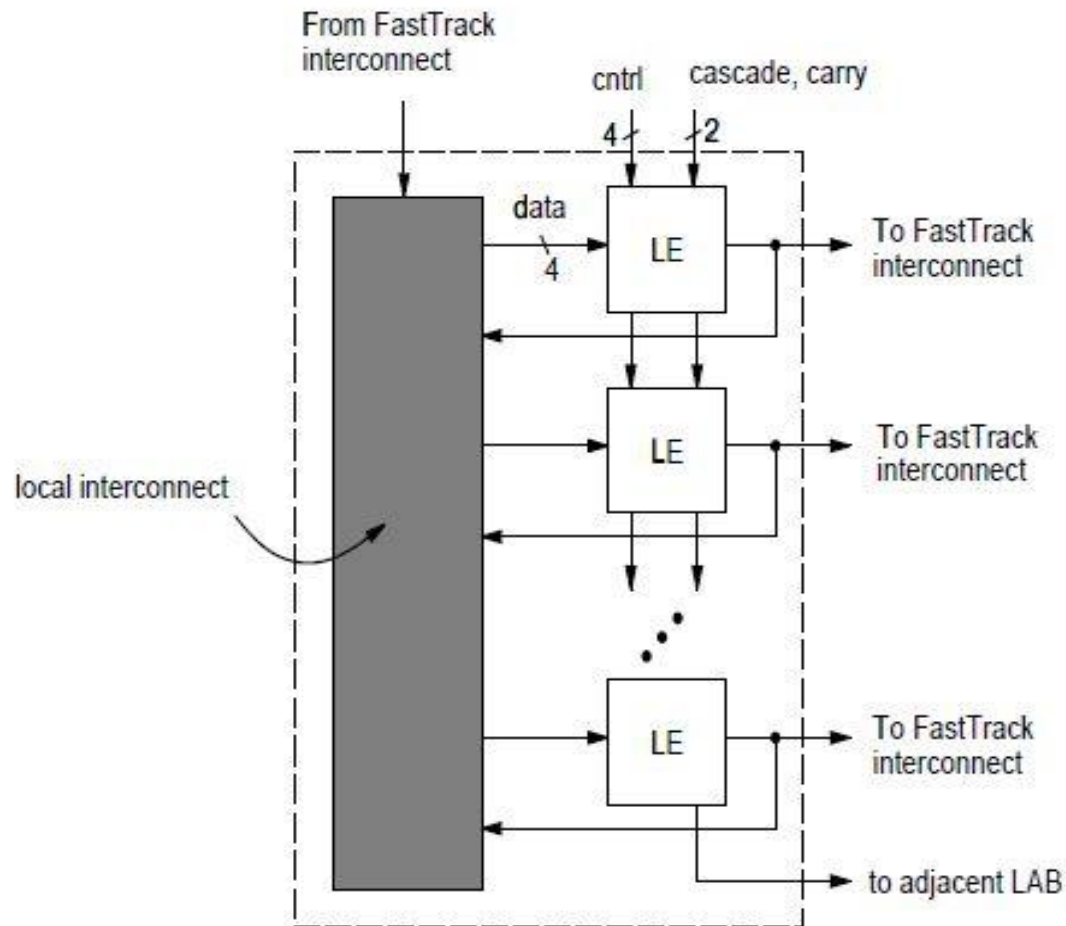
Overview of some FPGAs

- Altera Flex8K LE



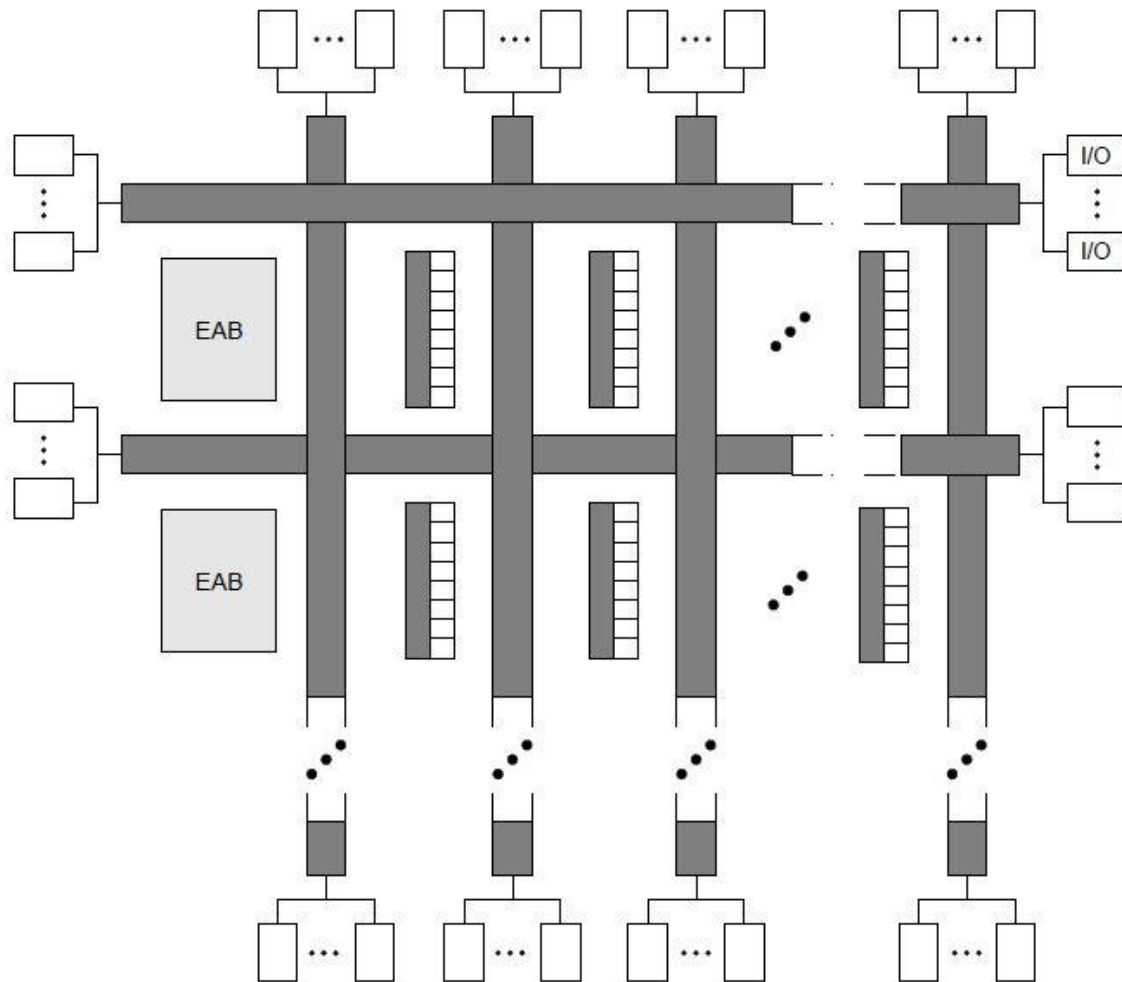
Overview of some FPGAs

- Altera Flex8K LAB



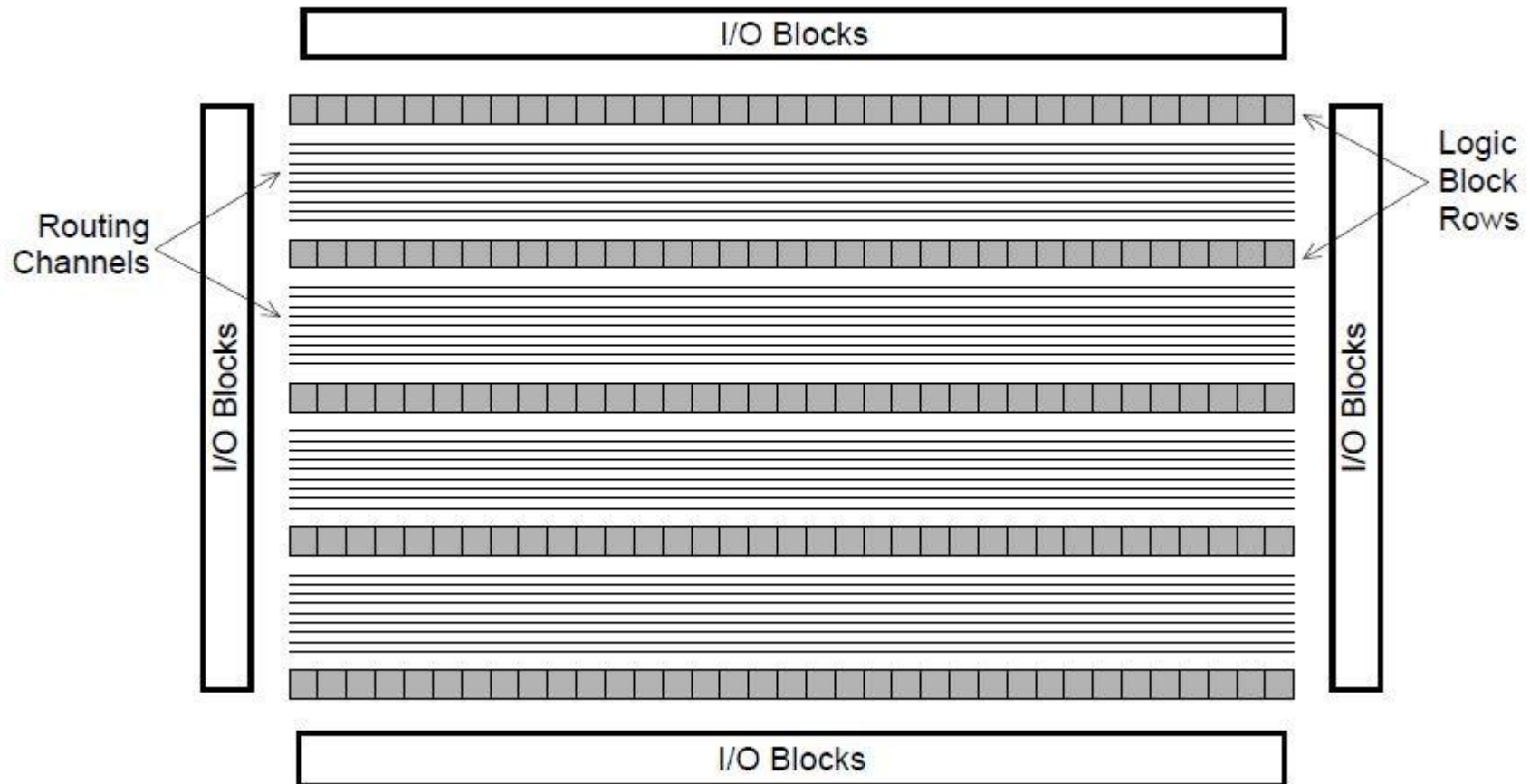
Overview of some FPGAs

- Altera Flex10K FPGA with EABs



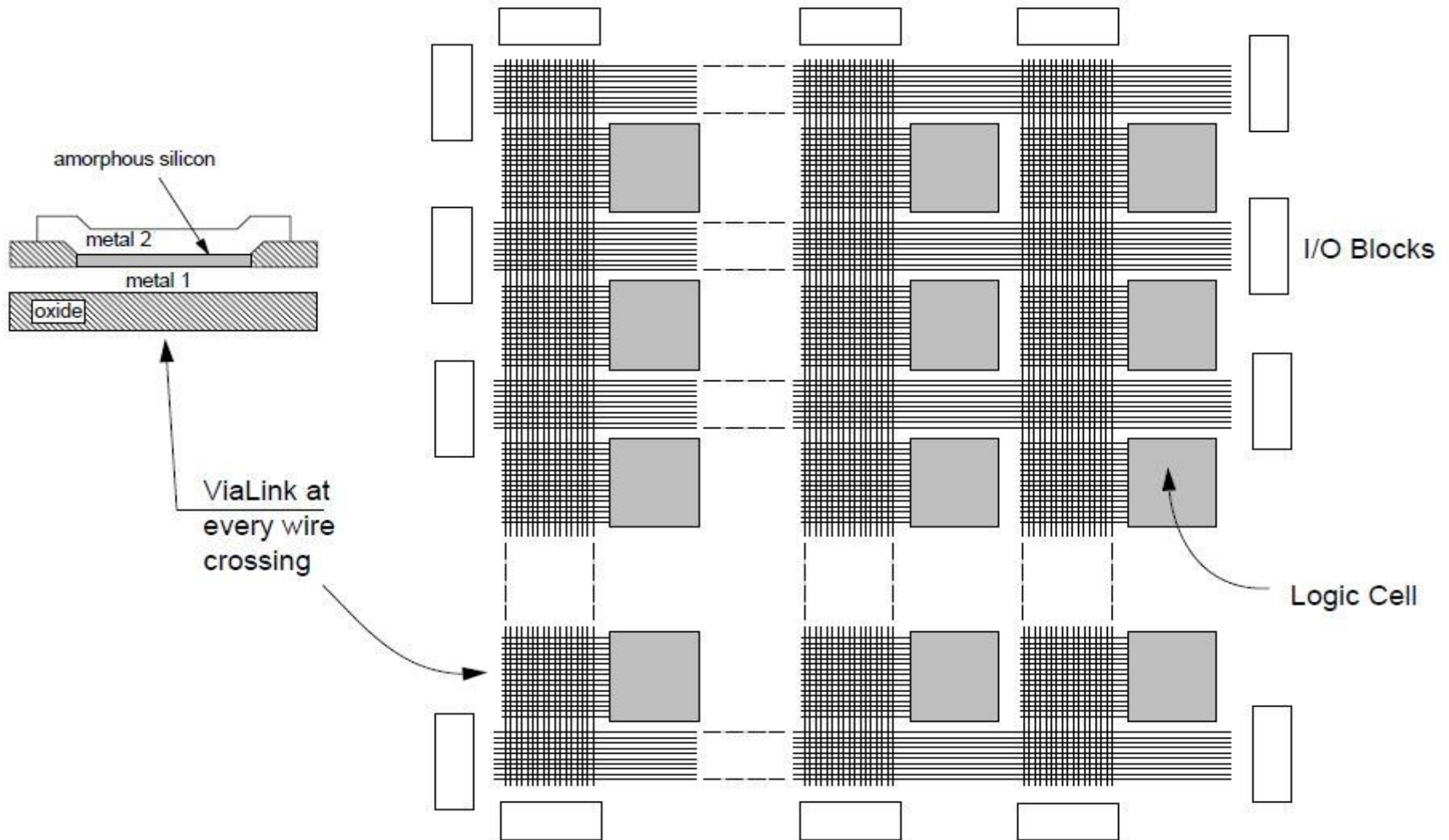
Overview of some FPGAs

- Actel FPGA – channeled gate array



Overview of some FPGAs

- Quicklogic – pASIC FPGA, Antifuse based



Applications of FPGAs

- Typical applications includes:
 - Random logic
 - Integrating multiple SPLDs
 - Device controllers
 - Communication encoding
 - Filtering
 - Small to medium sized systems with SRAM blocks
 - Rapid prototyping
 - In verification, Emulation
 - And many more

References

- Field programmable gate arrays
 - John V. Oldfield, Richard C. Dorf
- FPGA based system design
 - Wayne Wolf
- Fpgacentral.com
- Engineersgarage.com
- 1-core.com
- <http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf>