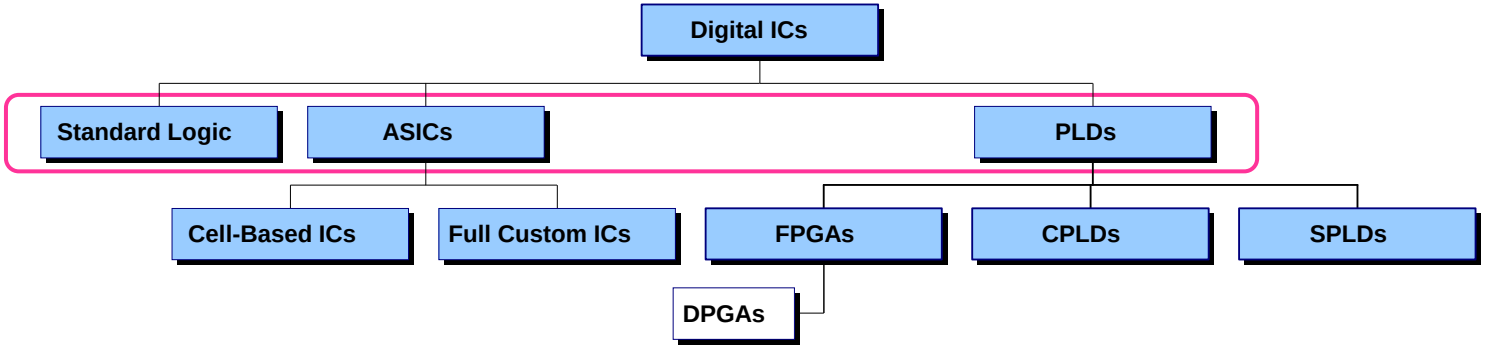
Digital Logic
Design with FPGA

Digital device implementation techniques

The way to FPGAs

Digital Integrated Circuits



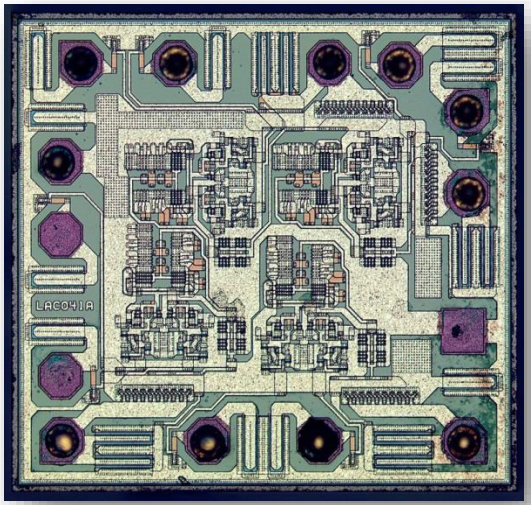
* Division according to function and design, manufacturing technology is not taken into account

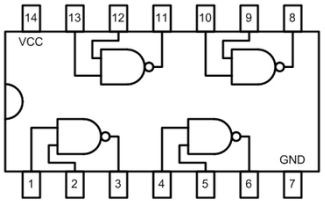
** The scheme does not include the ASSP group, the intermediary between Standard Logic and ASIC (of little importance from the point of view of further parts of the course)

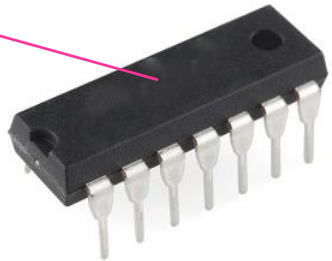
- Digital integrated circuits are currently the most common devices on the semiconductor market.
- The presented hierarchical structure reflects the subsequent stages of development of these systems.
- The first designs of simple programmable circuits (SPLD) appeared in the late 1970s.
- CPLD structures – early 1980s, FPGA – late 1980s, but their dynamic development began in the 2000s.
- FPGAs have replaced other types of chips in many applications and are still thriving.

Implementation techniques

- Small and medium scale integration systems
  - Complexity: up to approximately 200 logic gates per chip
  - Most often integrated circuits of the 74xx series
    - Gates, flip-flops, latches
    - Decoders, registers, counters, basic functional blocks



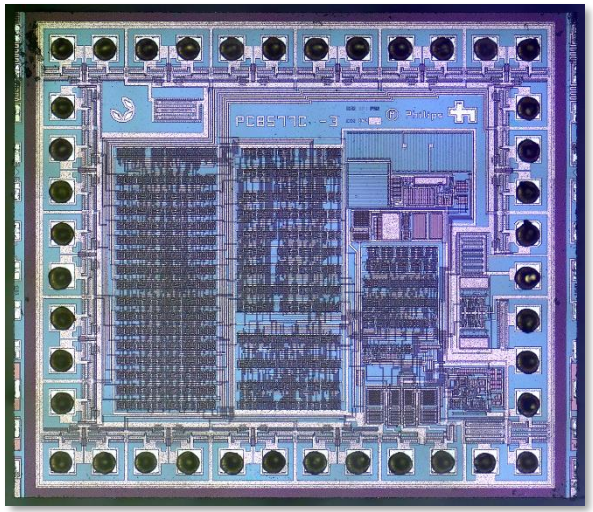




source: 74AHC00D NXP Semiconductors, https://pl.wikipedia.org/wiki/Seria_7400#/media/Plik:NXP-74AHC00D-HD.jpg

Implementation techniques

- Large Scale Integration systems
  - Complexity: from 200 to approximately 200k logic gates.
  - Small memories, programmable logic circuits, custom design circuits.
- Very Large-Scale Integration systems
  - Over 200k logic gates per chip.
  - Complexity is often defined as the number of transistors, especially when using integrated memories, or the number of configurable blocks in the case of PLD.



Implementation techniques - SSI

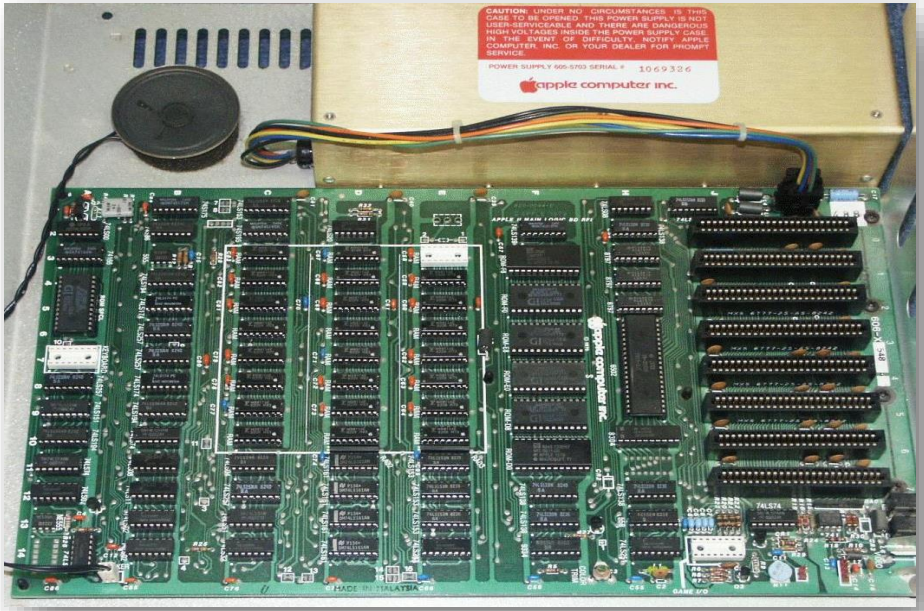
- Based on the overview of SSI and MSI components from catalogs (e.g. TI)
  - Different functional classes.
  - Most often used as 'glue logic' for systems with a larger scale of integration.
  - Formerly a basic implementation technique (even for large systems).

source: TANDY TRS-80 mainboard, http://dunfield.classiccmp.org/trs80/h/m4board.jpg

Implementation techniques - SSI

- Advantages of using the technique of implementation in SSI and MSI
  - Easy to understand features.
  - Access to internal signals of the implemented system.

- Disadvantages
  - Small packing of system components -> necessity to use large PCB
  - High power consumption
  - High implementation costs
  - Costly design mistakes



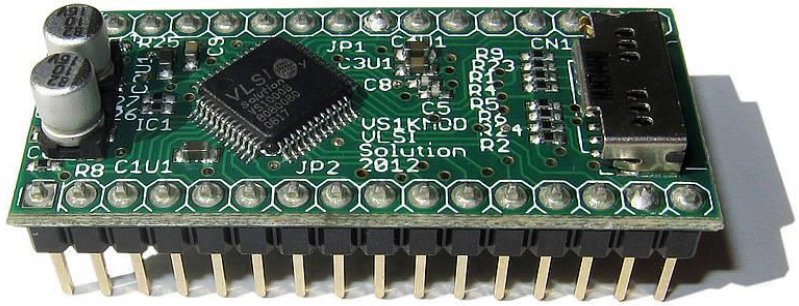
source: inside Apple II, http://dunfield.classiccmp.org/apple2/h/pboard.jpg

Implementation techniques - LSI

- To replace large and power-hungry circuits with SSI systems, significant parts of the designed system were placed within one integrated circuit
- Specialized systems (ASICs) - intended for specific applications, optimized to work in specific conditions
- Simple programmable systems (they allowed the designer to maintain the influence on the internal structure of the device and had some of the advantages of ASIC systems)

- Overview of LSI elements for digital systems (especially PLD in this complexity class)
  - ROM, PLA, PAL = SPLDs.
  - CPLDs

They can be used as 'glue logic'; thanks to high packing density, they can implement significant parts of the designed system



source: VSMD001 - VS1000 Audio Module, https://www.vlsi.fi/en/products/vsmd001.html

Implementation techniques - LSI

- Advantages of LSI implementation
  - Higher packing -> smaller PCBs or larger projects.
  - Easy to modify the design.
  - Thanks to reconfiguration, reduced costs of design error.

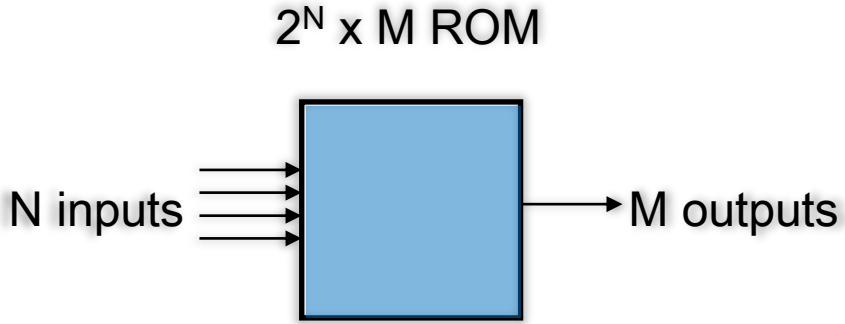
- Disadvantages
  - Specific implementation tools.
  - The need to have a good understanding of the hardware layer.
  - Limited access to internal signals of the project.

Implementation techniques – LSI programmable

- What is ROM?
- What is PLA?
- What is PAL?
- What are the relationships between these systems?
- What does the abbreviation CPLD stand for?

Implementation techniques - ROM

- Read Only Memory (ROM) is a sum-of-products (SOP) logic system with a fixed AND matrix and a programmable OR matrix.
- The ROM element can implement M logical functions of N-variables.



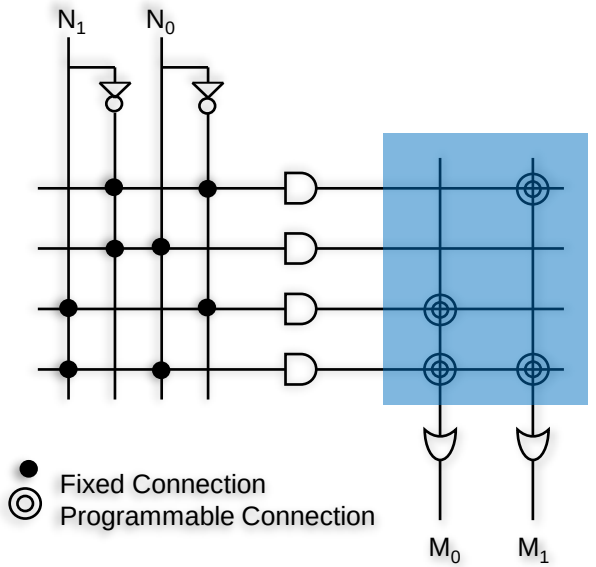
Implementation techniques - ROM

- The simplest way to implement function:
  specifying the truth table for the designed functions and writing the contents to ROM.
- There is no need to simplify the function - implementation based on the entire list of minterms.

Example: ROM memory, 2**N M-bit words;

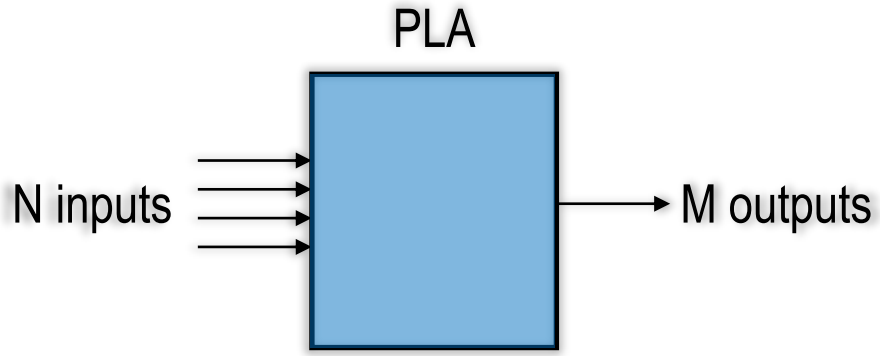
N = 2, M = 2

- M0 = N1•N0 + N1•N0’
- M1 = N1•N0 + N1’•N0’



Implementation techniques - PLA

- Programmable Logic Array (PLA) is a logic system implementing the sum of products (SOP) with a programmable OR matrix and a programmable AND matrix (however, fewer programming points than ROM).
- The function can be programmed based on the minterm list; minterms can be shared between different functions.

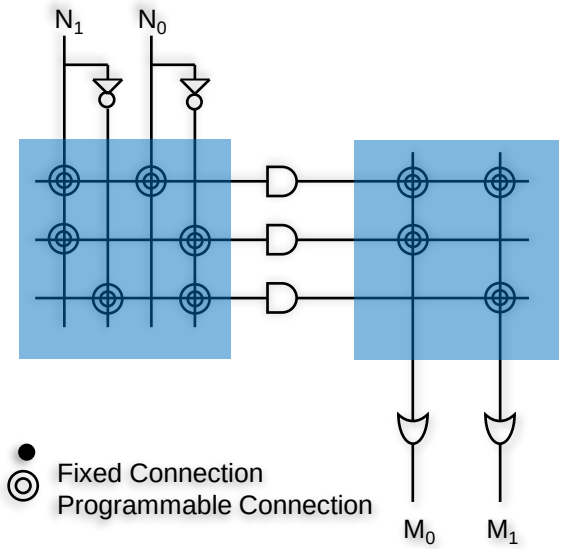


Implementation techniques - PLA

- The designer must reduce the function to the SOP form with the hope that it will be possible to implement it in the available PLA element (limited number of AND lines!).
- CAD/EDA tools needed to optimize minterm sharing.

Example: PLA with N inputs and M outputs;
N = 2, M = 2

- M0 = N1•N0 + N1•N0'
- M1 = N1•N0 + N1'•N0'



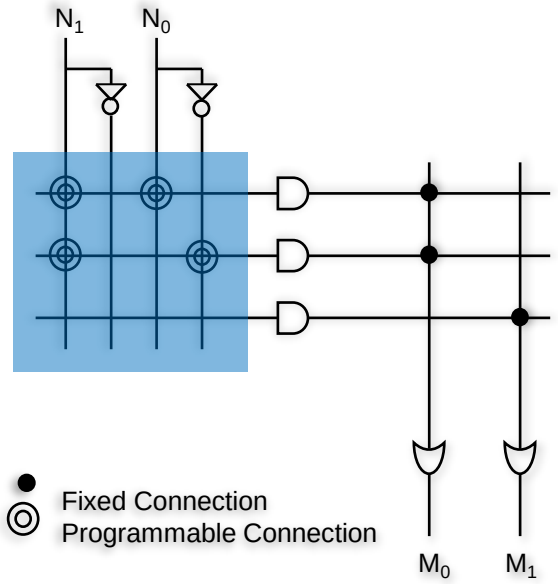
Implementation techniques - PAL

- Programmable Array of Logic (PAL) is a sum-of-products (SOP) logic system with a programmable AND matrix and a fixed OR matrix.
- The function can be programmed based on the minterm list; no sharing of points.

Example: PAL device with N inputs and M outputs;

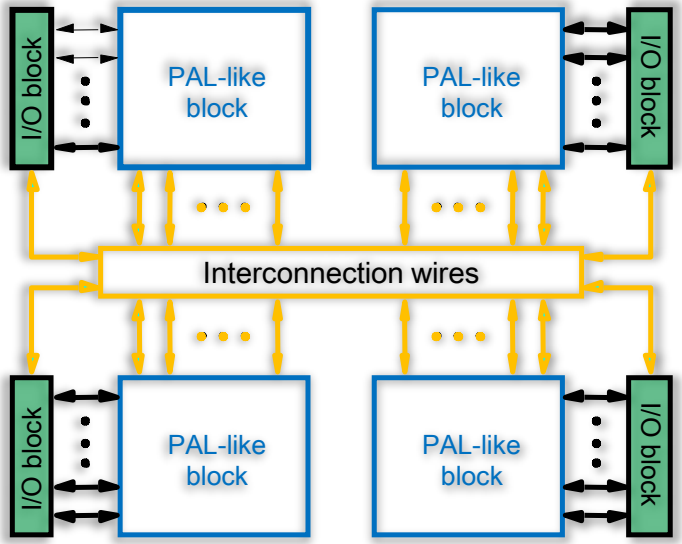
N = 2, M = 2

- M0 = N1•N0 + N1•N0'
- M1 = N1•N0 + N1'•N0'
  – implementation impossible

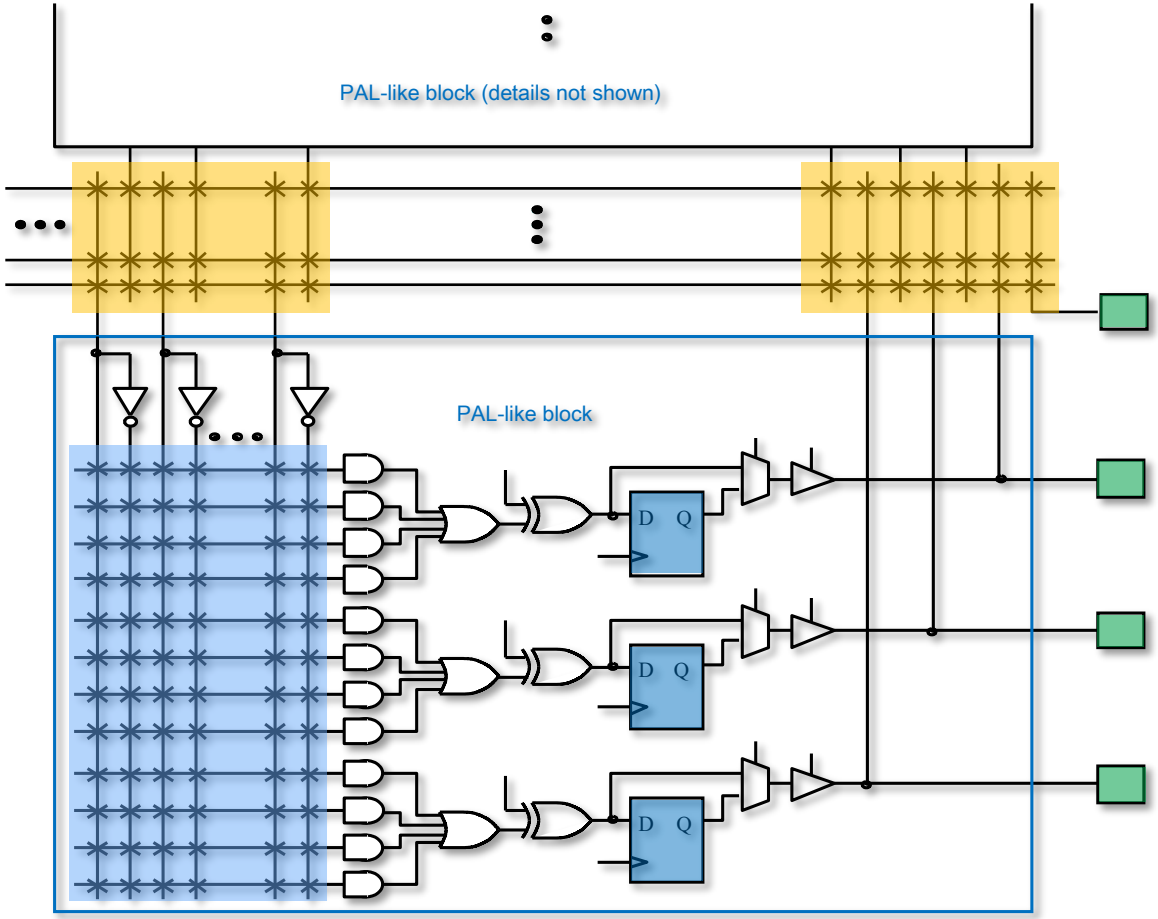


Implementation techniques - CPLD

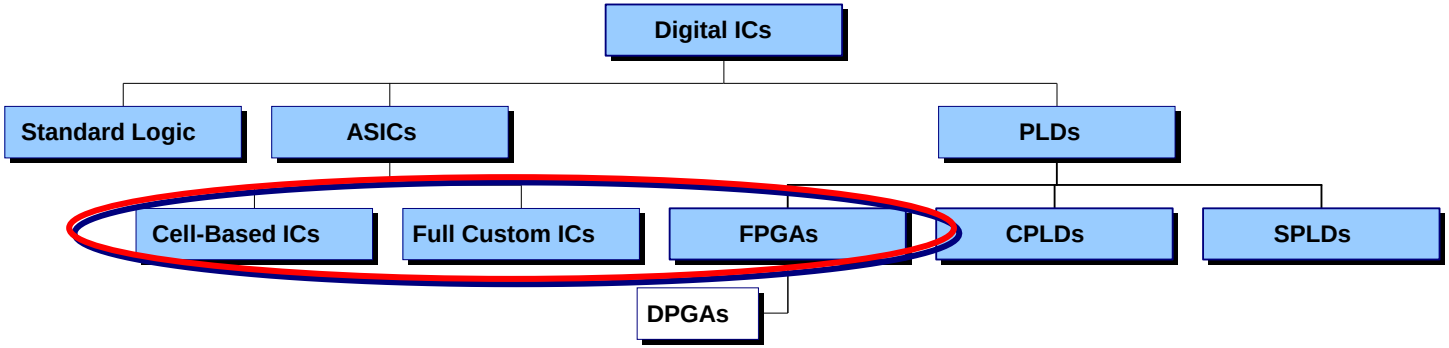
- CPLD is a complex programmable system (complex PLD) composed of many simple programmable blocks (such as PLA, PAL, and less often ROM).
- These blocks are connected through a programmable connection matrix.
- The problem of increasing complexity of CPLD systems related to the method of implementing interblock connections.



CPLD – internal structure



Digital Integrated Circuits



- Methods of designing integrated circuits of LSI and VLSI: 'cell-based' and 'full custom'.
- Field Programmeble Gate Array is a natural filling of the space between the methods of implementing ASIC systems and designing using standard elements.
- The first attempt to develop this area was CPLD systems, but it quickly turned out that they were not suitable for implementing large (highly complex) projects. The main problem here was the shortage of memory elements.

Implementation techniques

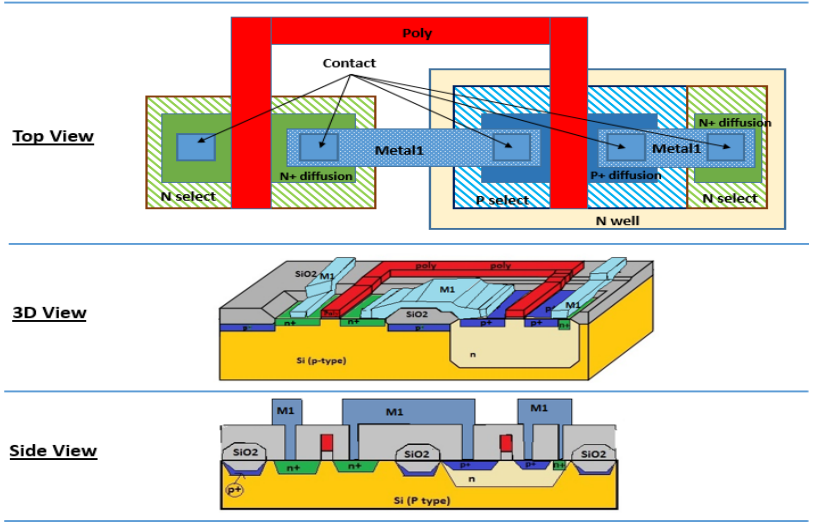
- Full Custom Logic
  - designing the system from the level of masks, individual transistors, …
- Standard Cell Design (semi-custom)
  - the use of libraries of standard elements prepared, e.g. by the manufacturer
- Gate Array Design (semi-custom)
  - in order to reduce production costs, an implementation technique based on gate arrays was created

In all these techniques, even if programming was possible, it was one-time programming and on the technological line at the manufacturer of the system.

- Field Programmable Logic
  - an implementation technique that is a development of the 'gate array design' approach

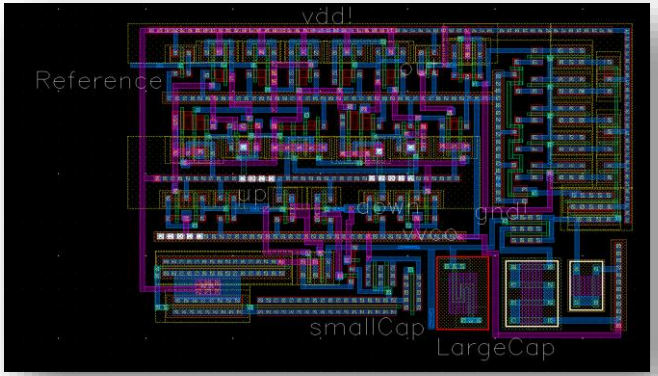
Implementation techniques

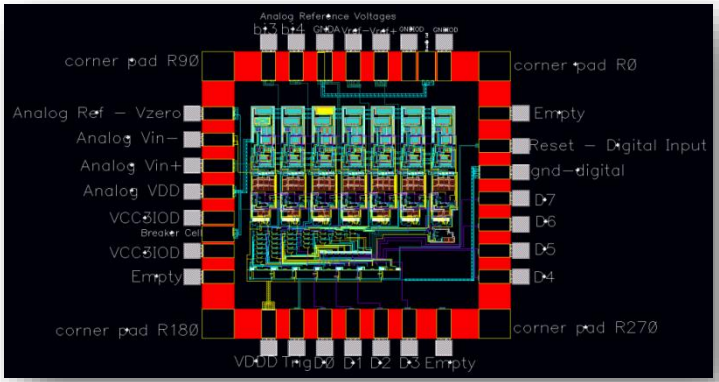
- Full Custom Logic
  - Each logical element of the system (or transistor) is manually designed and optimized.
  - The structure with the highest density, the highest possible operating speed, the lowest power consumption.
  - The NRE (non recurring engineering cost) indicator is the highest possible.
  - Currently rarely used due to very high engineering costs and low efficiency.



Implementation techniques

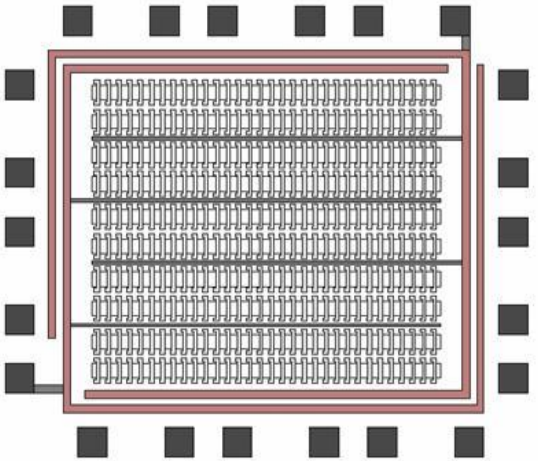
- Standard Cell Design
  - It is based on predefined blocks (a la 74xx) available to the designer in the cell library
  - The entire project is built from previously defined components
  - Performed using a schema editor or HDL
  - Automatic tools for arranging and combining cells
  - Cell sizes are usually standard sizes (facilitating P&R tools)
  - Significant reduction in design time compared to "full custom design"





Implementation techniques

- Gate Array Design
  - "Full custom" and "standard cell" projects require dedicated masks for the production of integrated structures (dedicated means expensive)
  - "Gate array" implementations use prefabricated structures composed of unconnected gates
  - The designer specifies the connections and, based on them, the metal layers for the gate array are implemented
  - Low cost of producing masks
  - Rapid design iterations
  - Poor use of the system area (redundant gates)



Implementation techniques

- Field Programmable Logic
  - SPLDs
    - ROM, PLA, PAL
  - CPLD

  - FPGA – Field Programmable Gate Array
    - The main family of programmable (configurable) circuits for implementing digital systems
    - Improvement of the "gate array" technique, offer an improved time-to-market ratio and reduced costs of prototype implementation.
    - 'field programmable' means direct programmability! 'in the field' - at the user's place
    - Types of FPGA devices (due to configuration method):
      - Non-volatile configuration, one-time programming (anti-fuse)
      - Non-volatile configuration, multiple programming (flash)
      - Volatile configuration (sram)

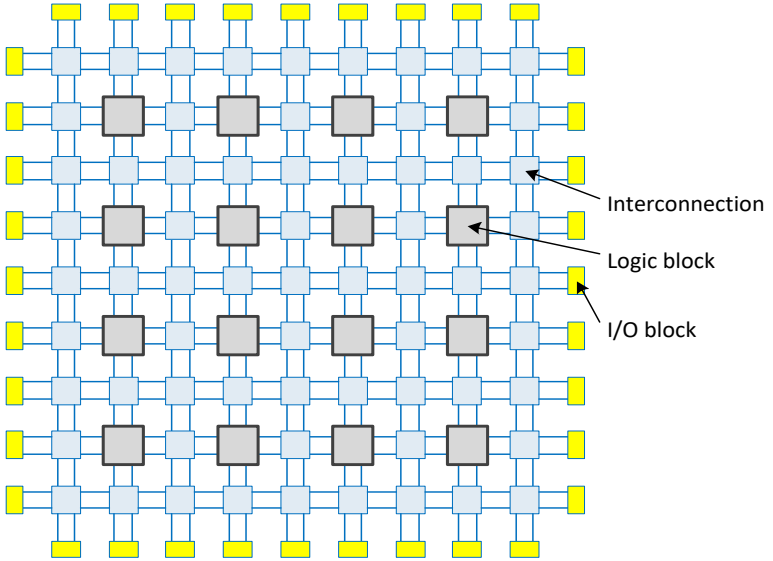
FPGA devices

- An FPGA is more than just a gate array...

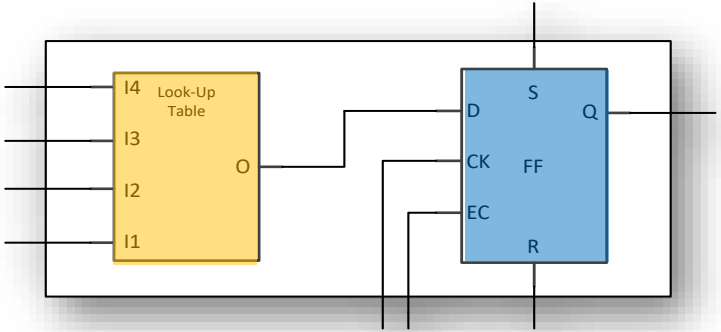
Components placed in modern FPGA systems

- Programmable logic blocks
- Programmable I/O blocks
- Programmable connection lines

- Clock management blocks
- Memory blocks
- Microprocessors
- Fast serial interfaces
- …



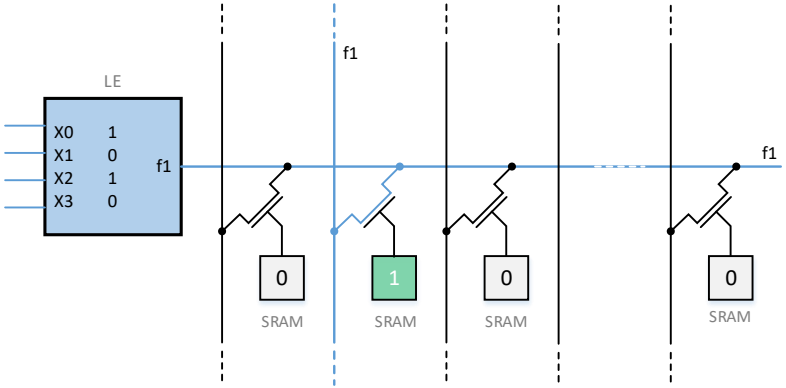
Programmable logic



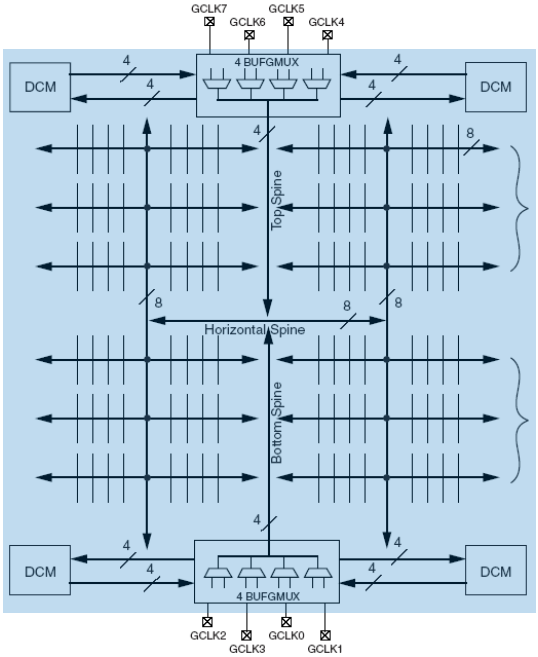
- Simplified programmable logic block (SRAM version)
- A block of static memory for implementing combinational functions
- Parameterizable flip-flop for implementing sequential systems
- The logic block is called CLB (Configurable Logic Block) by Xilinx/AMD, LE (Logic Element) by Intel/Altera, and LE (Logic Element) by Microsemi.

*Elements of connections inside the configurable block have been omitted

Programmable connections



- Simplified programmable connection (SRAM version)
- Using MOS transistors as switches
- Static memory cells for maintaining connection state

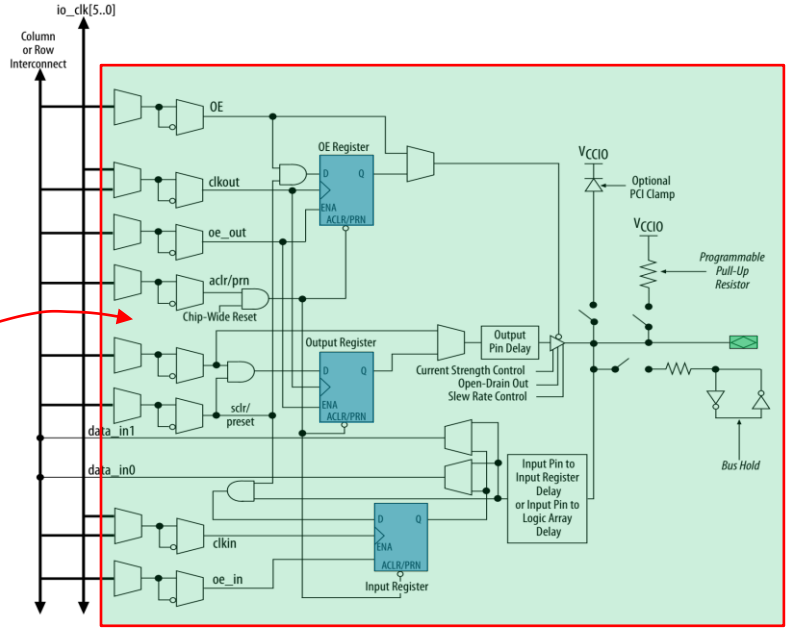


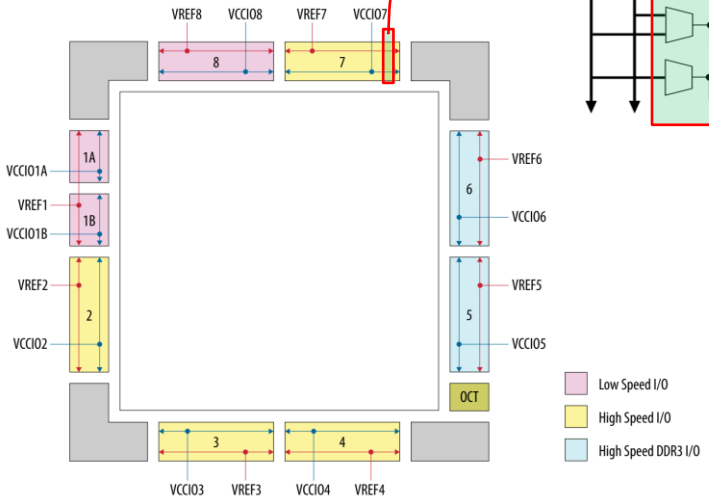
FPGA clock tree [1]

source: [1] Xilinx; Spartan III FPGA Family, DS099 April 2006

Programmable I/O

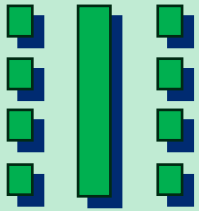
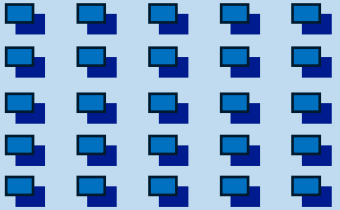
- Programmable port direction
  - in, out, bidirectional, tri-stated
- Programmable signaling standard
  - LVTTL, LVCMOS, LVDS …
- Programmable data transmission rate
  - SDR, DDR
- Programmable delay
- Programmable signal strength
- Programmable slope





source: Intel® MAX® 10 General Purpose I/O User Guide, https://www.intel.com/content/www/us/en/docs/programmable/683751/22-1/i-o-overview.html

CPLD vs. FPGA

	Complex Programmable Logic Device	Field-Programmable Gate Array
		
Architecture	heterogeneous - "PAL-like" mainly combinational circuits	homogeneous - "Gate array-like" lots of registers + RAM dedicated specialized blocks
Amount of resources	low to medium 0.5-10K	medium to very high 1K to 30M
Efficiency	specified time below 1 GHz	application dependent below 1 GHz
Connections	crossbar	incremented

Fields of application

- CPLD
  - glue-logic"
  - control systems
  - automotive industry
  - portable devices

- FPGA
  - DSP
  - CAD systems (emulation, prototyping)
  - CCM – custom computing machines
    (RC – reconfigurable computing, CSoC, APSoC)
  - …

FPGAs - summary

- "a bit of programmable logic with a huge number of programmable connection paths“

- The introduction of programmed connections is of particular importance - a fundamental improvement over the classic "gate array" technique
- Manufacturers offer a whole range of systems of various sizes (from hundreds to millions of conversion gates) and various properties