Microelectronic Design

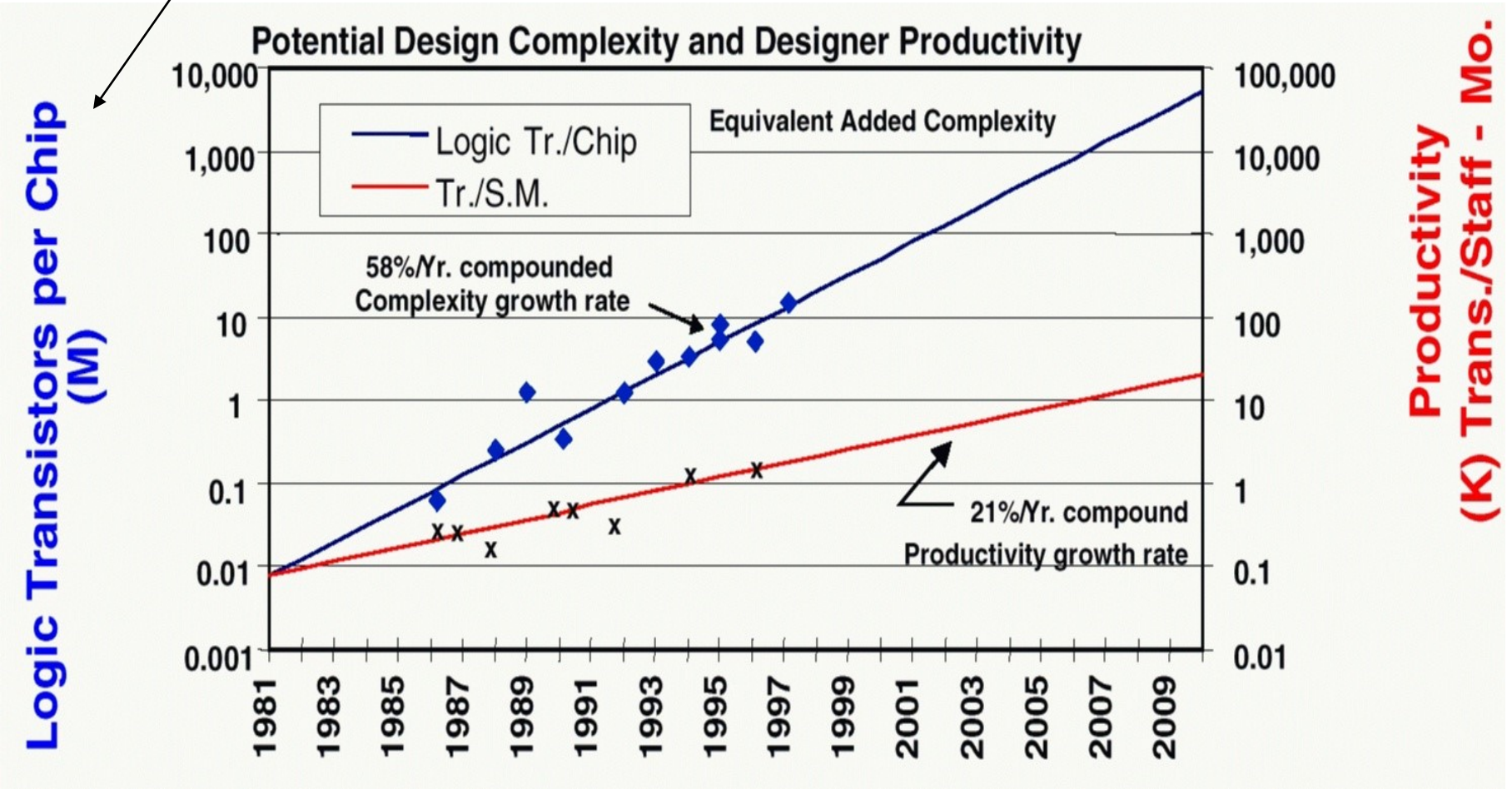
EE2222 Computer Interfacing and Microprocessors

Partially based on
Introduction to Computer Design by Prof. Eli Bozorgzadeh
Digital Design Principles and Practices by John F. Wakerly

Microelectronic systems

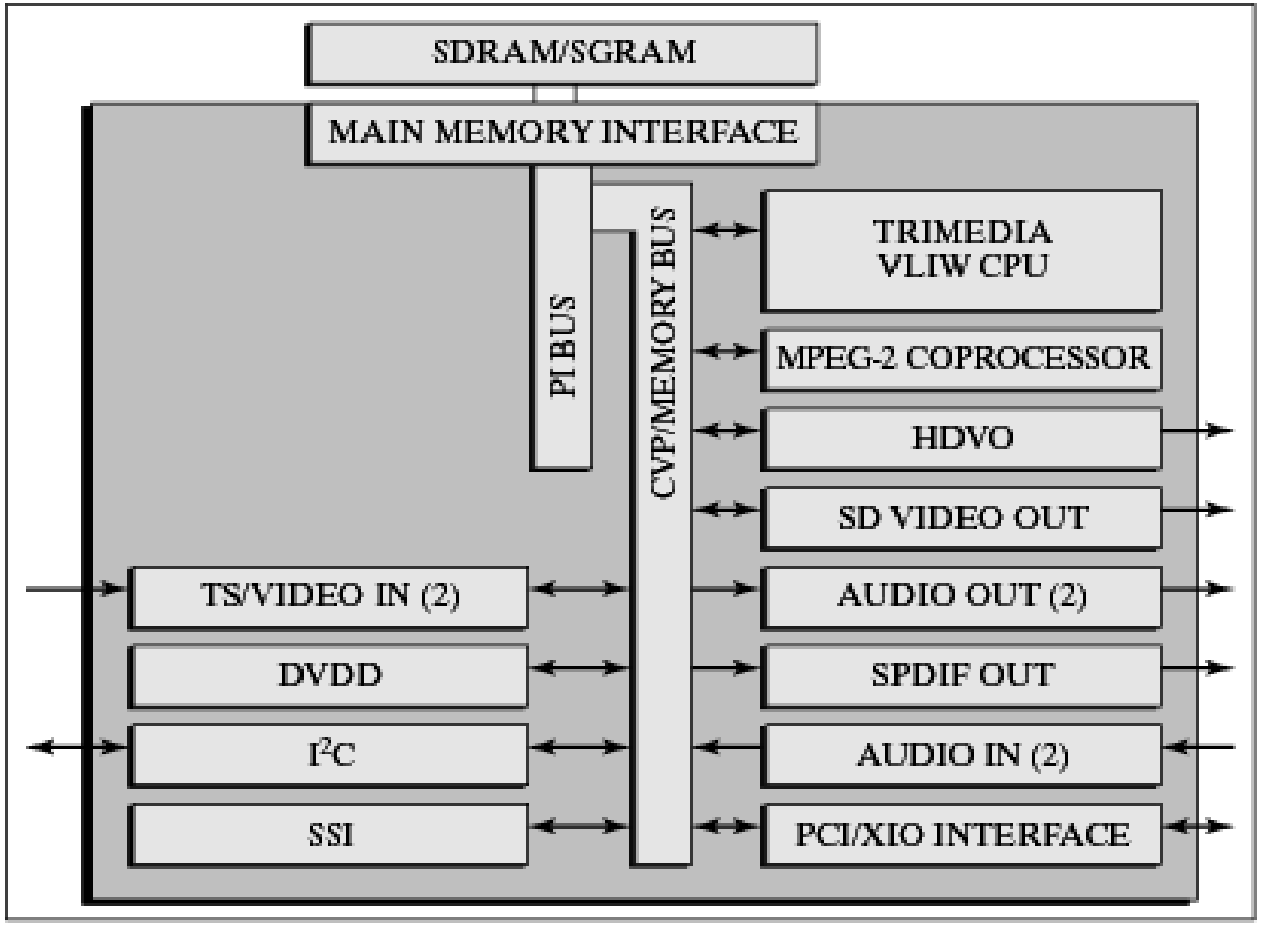
- Computers, mobile phones, embedded systems, etc.
- Composed of increasingly integrated and complex circuit design > 10^7 transistors
- Integrated circuits are called VLSI or microelectronic circuits
- Integrated circuits exploit semiconductor materials

Moore's Law: capacity doubles every 18 months



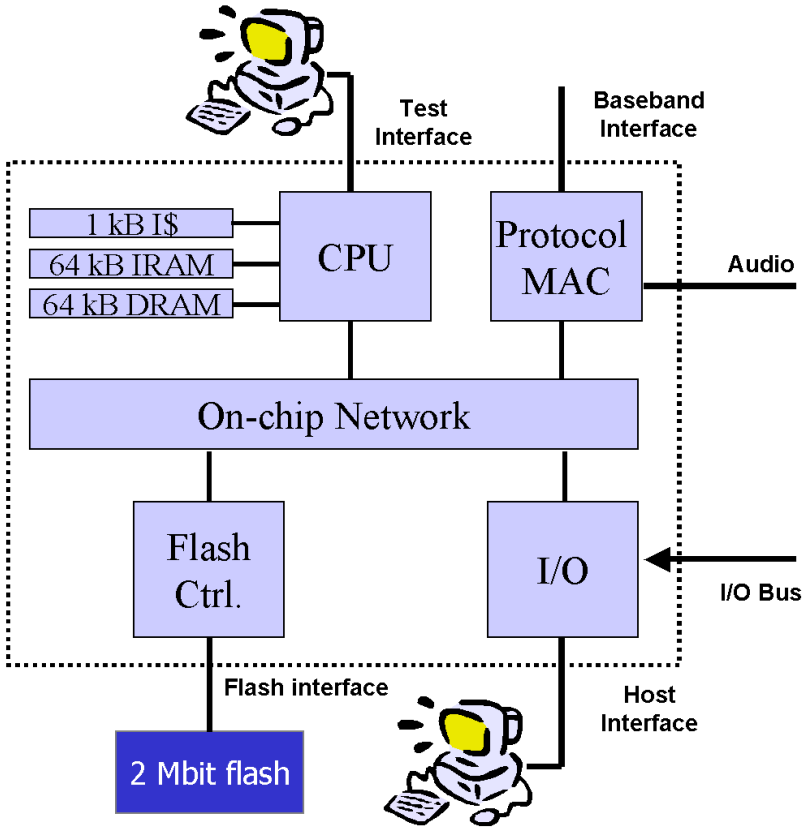
ITRS 1999

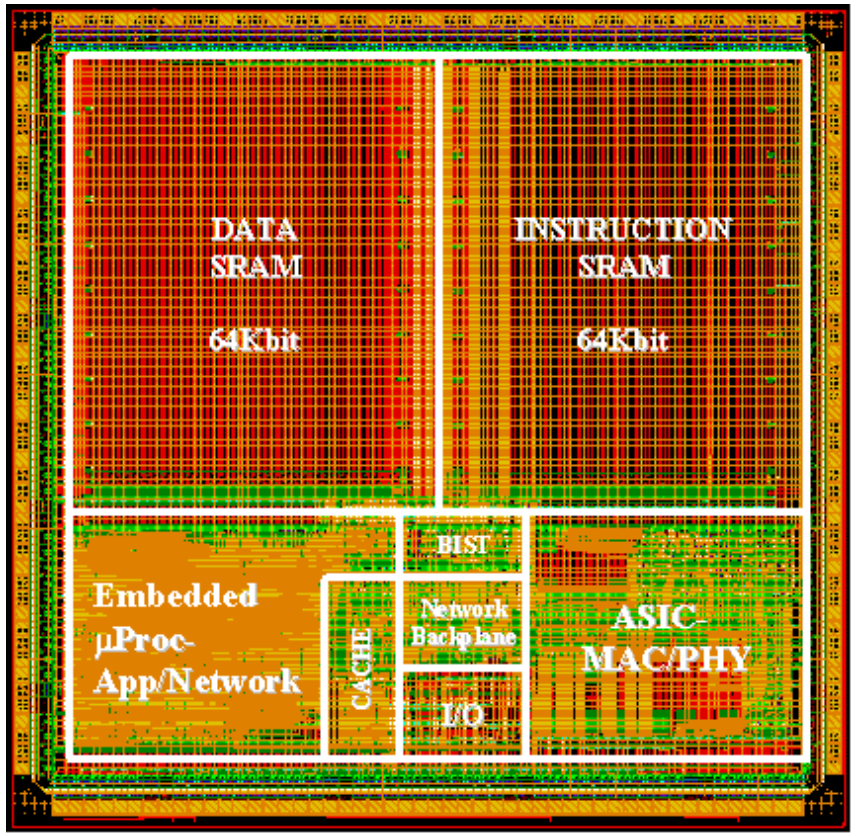
A System-on-a-Chip: Example



Courtesy: Philips

Floorplanning





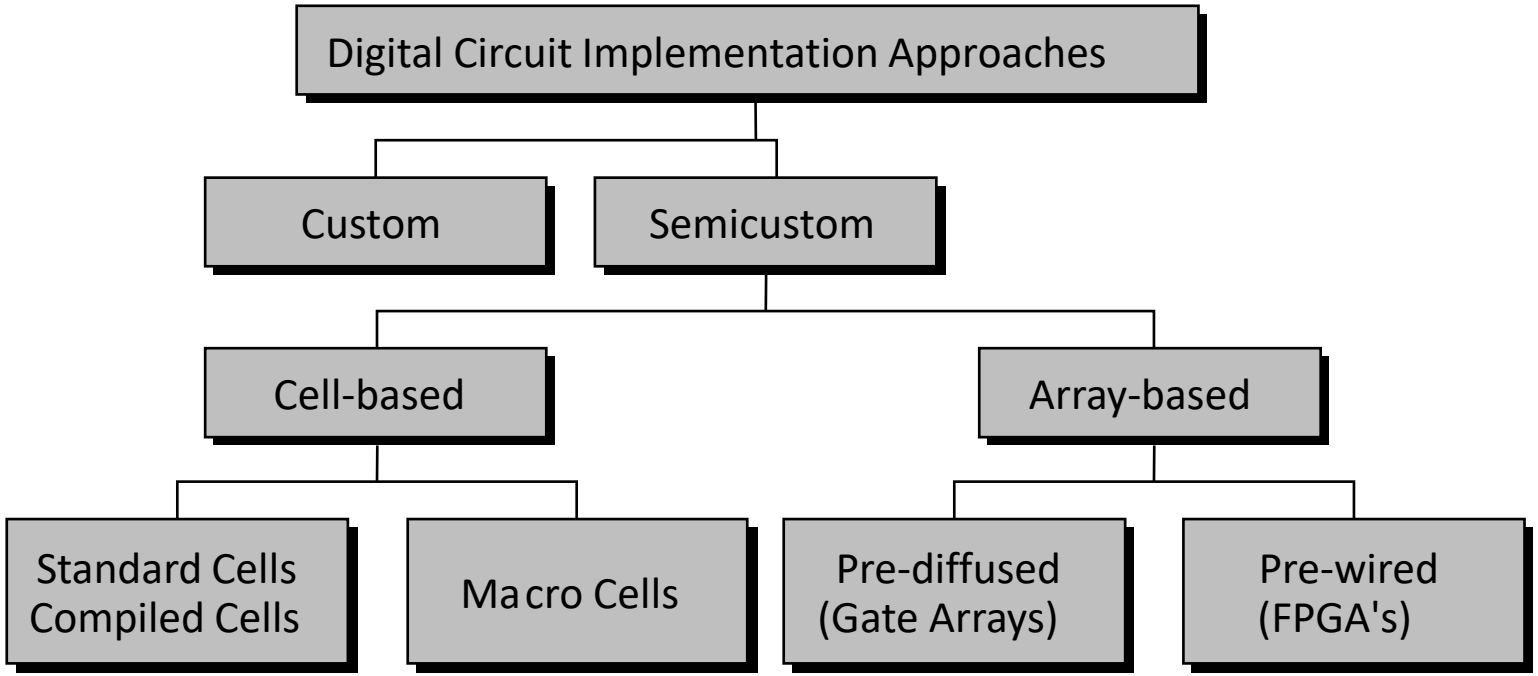
A Protocol Processor for Wireless

Microelectronic design styles

- General-purpose processors:
  - High-volume sales
  - High performance
- Application-Specific Integrated Circuits (ASICs):
  - Varying volumes and performances
- Prototypes
- Special applications (e.g. space)

Microelectronic design styles

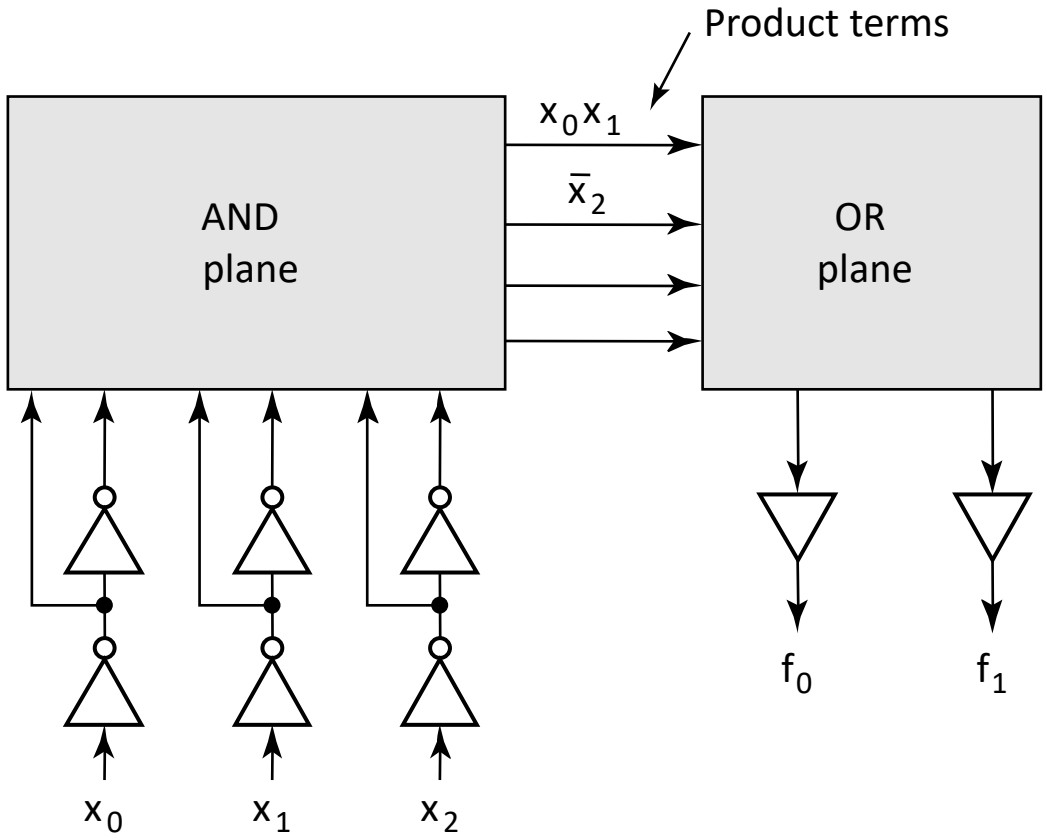
- Custom and semi-custom designs



Implementation Strategies

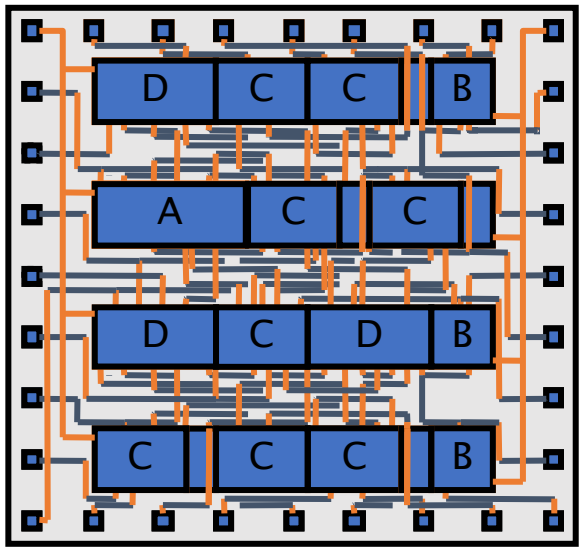
- PLA
  - Technology confined in cell macros (tiling)
- Cell based logic
  - Technology confined to cells (area)
  - Both 1-d and 2-d solutions
- Transistor Arrays (Gate arrays)
  - Technology confined to layers

Programmable Logic Array (PLA)



Standard Cells

- Cell library:
  - Cells are designed once
  - Cells are highly optimized
- Layout style:
  - Cells are placed in rows
  - Channels are used for wiring
  - Over the cell routing
- Compatible with macro-cells (e.g. RAMs)



Macro Cells

- Module generators:
  - Synthesized layout
  - Variable area and aspect-ratio
- Examples:
  - RAMs, ROMs, PLAs, general logic blocks
- Features:
  - Layout can be highly optimized
  - Structured-custom design

Macro Cell Design Style

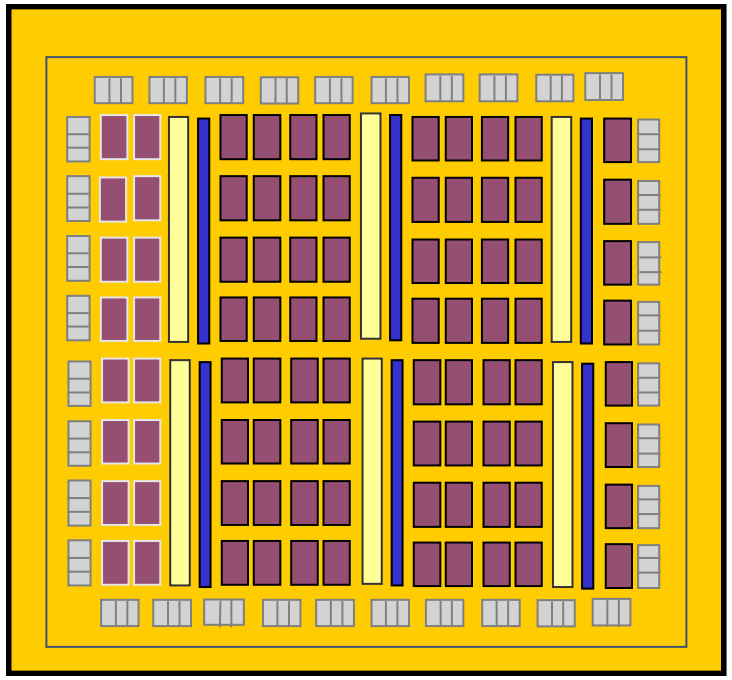
- Cells are of varying sizes (generally rectangular) and widely varying functional complexity (gates to register files to arithmetic units like adders & multipliers)
- Standard cells can be part of the design as well
- More flexibility than standard cells but the resulting placement and routing problems are more complex

Array-based Design

- Pre-diffused arrays:
  - Personalization by metallization/contacts
  - Mask-Programmable Gate-Arrays
- Pre-wired arrays:
  - Personalization on the end
  - Field-Programmable Gate-Arrays

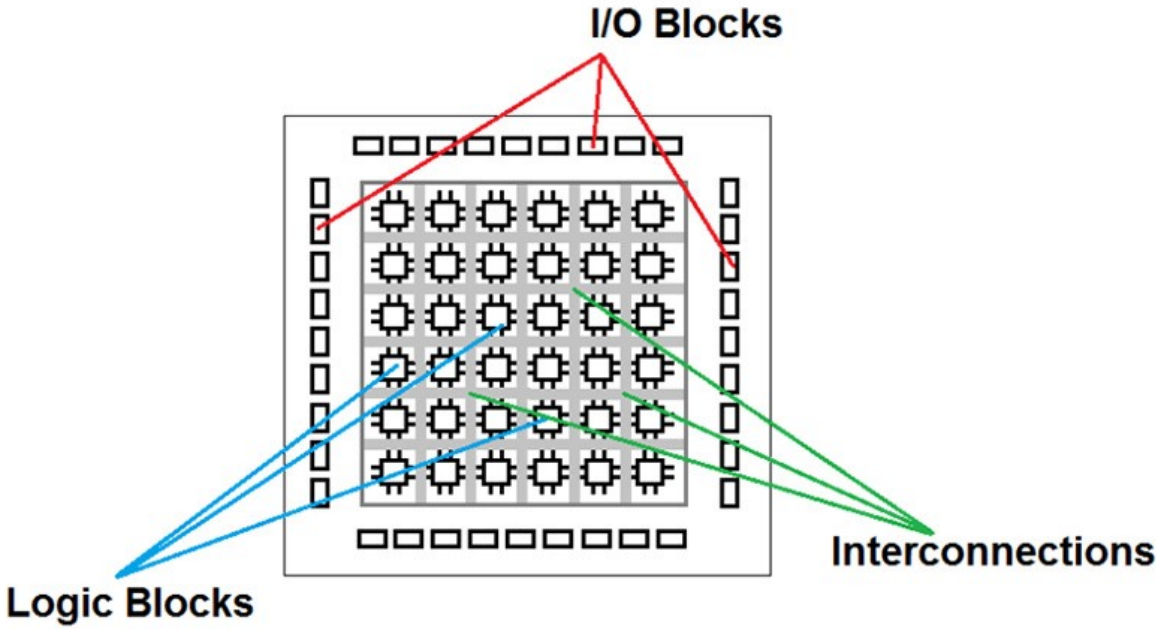
Field-Programmable Gate-Arrays (FPGA)

- Array of cells:
  - Each cell performs a logic function
- Personalization:
  - Soft: memory cell (e.g. Xilinx).
  - Hard: Anti-fuse (e.g. Actel).
- Immediate turn-around (for low volumes)
- Inferior performances and density
- Good for prototyping and re-customization



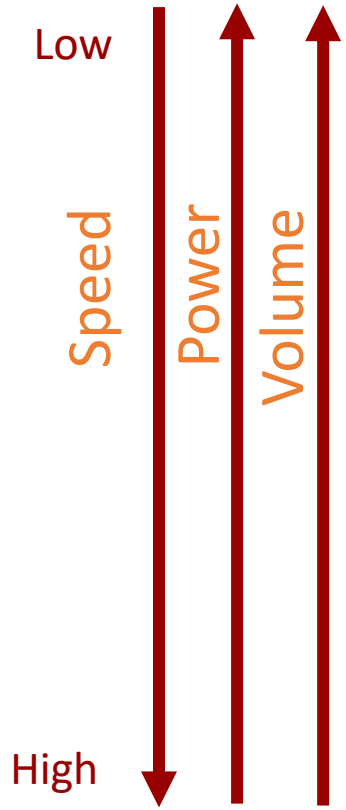
FPGA Design Style

- Both logic and routing are programmable
- Least flexibility in routing: needs to be done along pre-fabricated routing tracks going along horizontal & vertical channels
- Programmable switchboxes at the intersection of routing channels for interconnecting horizontal & vertical tracks in different channels



Compare choices

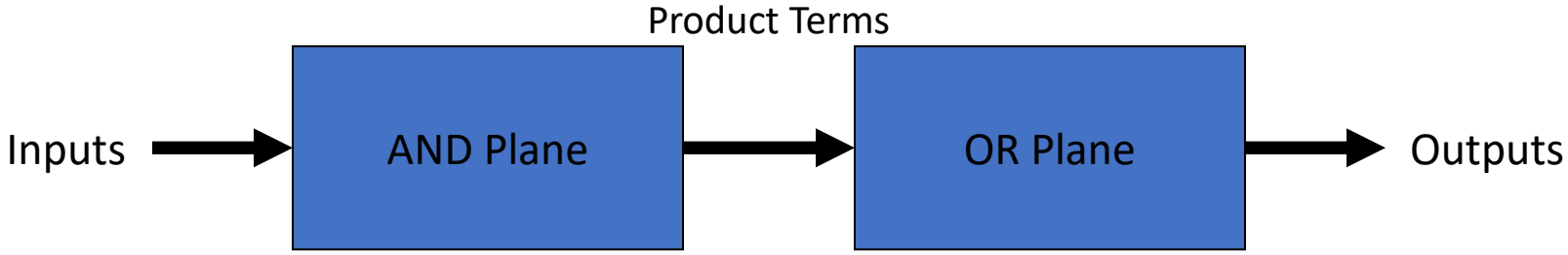
- Microprocessors
- Domain-specific processors
  - DSP
  - Network processors
  - Microcontrollers
- Reconfigurable SoC
- FPGA
- Gatearray
- ASIC



Memory Chips

Programmable Logic Devices

- The first type of PLDs considered has the AND-OR plane structure shown in the figure.
  - This type of architecture is used to implement ROMs, PLAs, and PALs.
  - It implements Boolean expressions in Sum of Products (SOP) form:
    - AND plane forms product terms selectively from the inputs, and
    - OR plane forms outputs from sums of selected product terms.
  - A programmable interconnect fabric joins the two planes, so that the outputs implement sum-of-product expressions of the inputs.
- Whether and how a plane can be programmed determines the particular type of PLD that is implemented by the overall structure.



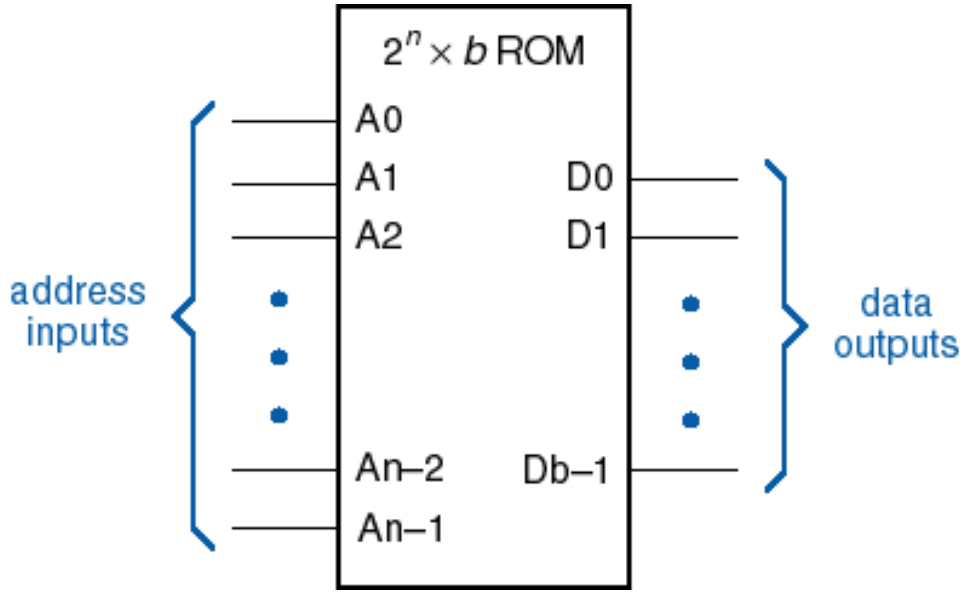
AND-OR plane structure of a programmable logic device

Storage Devices

- The architecture used to implement PLDs lends itself to implementation of storage devices.
- Storage Devices can be:
  - Read-Only or Random Access, depending on whether the contents of a memory cell can be written during normal operation of the device.
- ROM (read-only memory) is a device programmed to hold certain contents, which remain unchanged during operation and after power is removed from the device.
- RAM (random-access memory) in contrast its contents can be changed during operation, and they vanish when the power is removed.

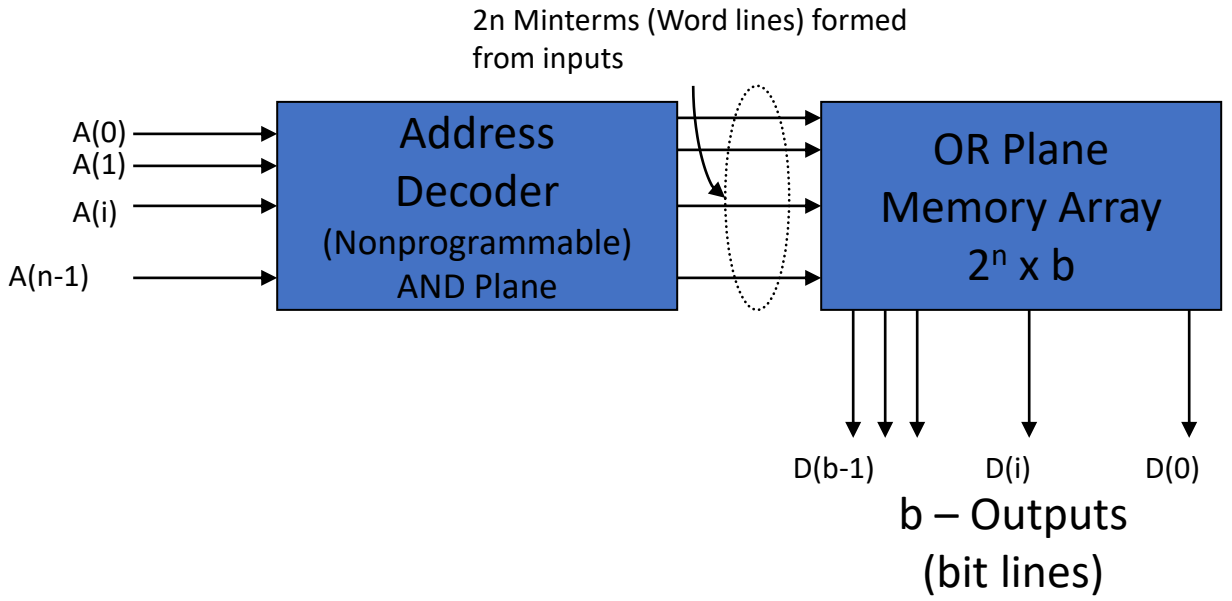
Read-Only Memory (ROM)

- Read-Only Memory (ROM)
  - A 2n x b ROM consists of an addressable array of semiconductor memory cells organized as 2n words of b bits each.
- ROM Interface:
  - n – inputs defining address lines.
  - b – outputs called bit lines.
- ROM is non-volatile memory. It's content is preserved even if no power is applied.



Read-Only Memory (ROM)

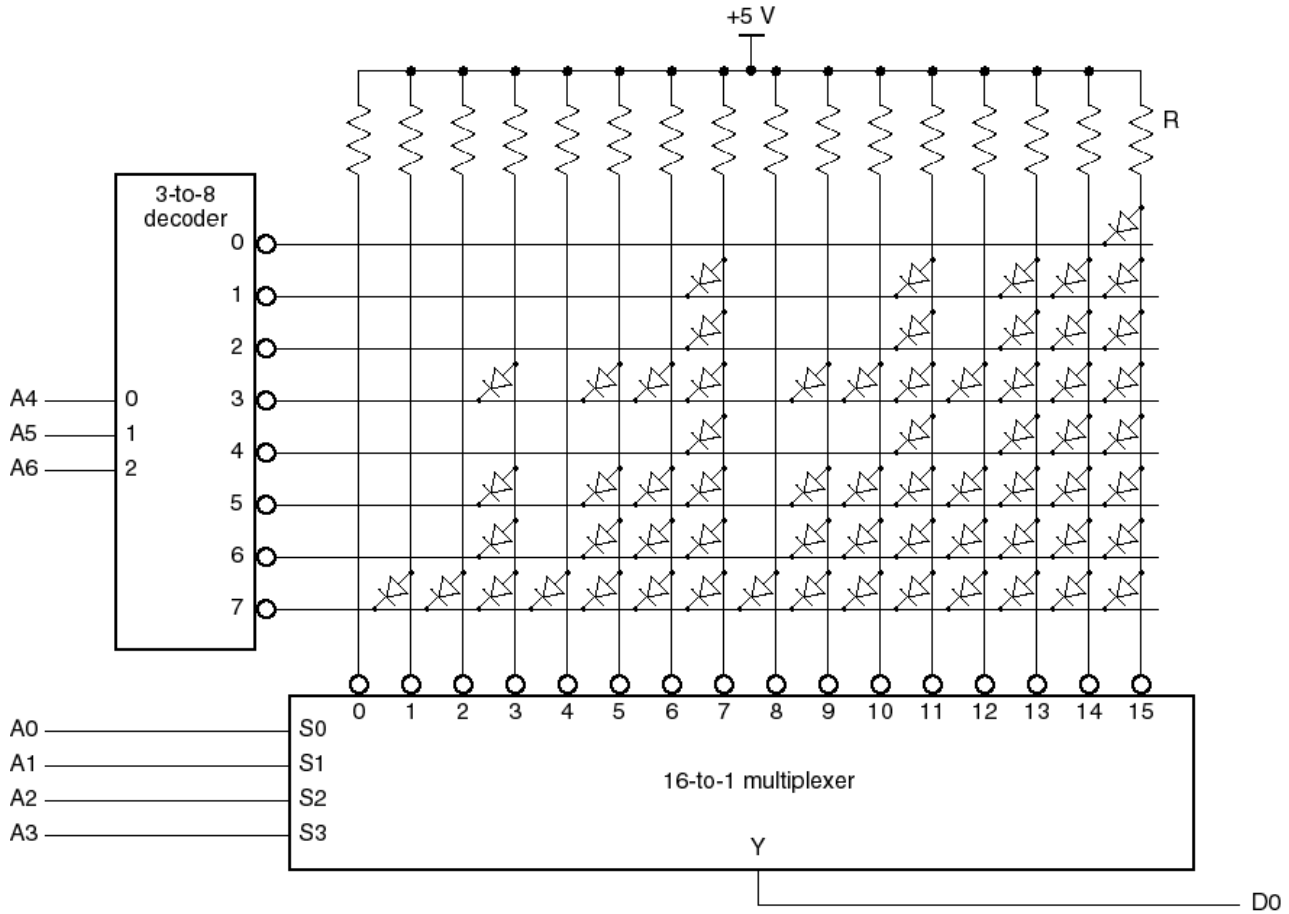
- AND-OR planes for a ROM:



Two-dimensional decoding

- Suppose that one wants to build a 128 x 1 ROM.
  - Straight forward solution will require a 7-to-128 decoder:
    - 128 7-input NAND gates,
    - 14 buffers and inverters with a fanout of 64 each.
  - ROMs with a 1M bits or more are available commercially and they do not use linear structure for decoder – which would require a 20-to-1,048,576 decoders.
- The structure used is called *two-dimensional decoding*. This structure enables reduction of the decoder size to something on the order of the square root of the number of addresses.
  - The basic idea in two-dimensional decoding is to arrange the ROM cells in an array that is as close as possible to square.
  - In the next illustration a possible internal structure for a 128x1 ROM is depicted.

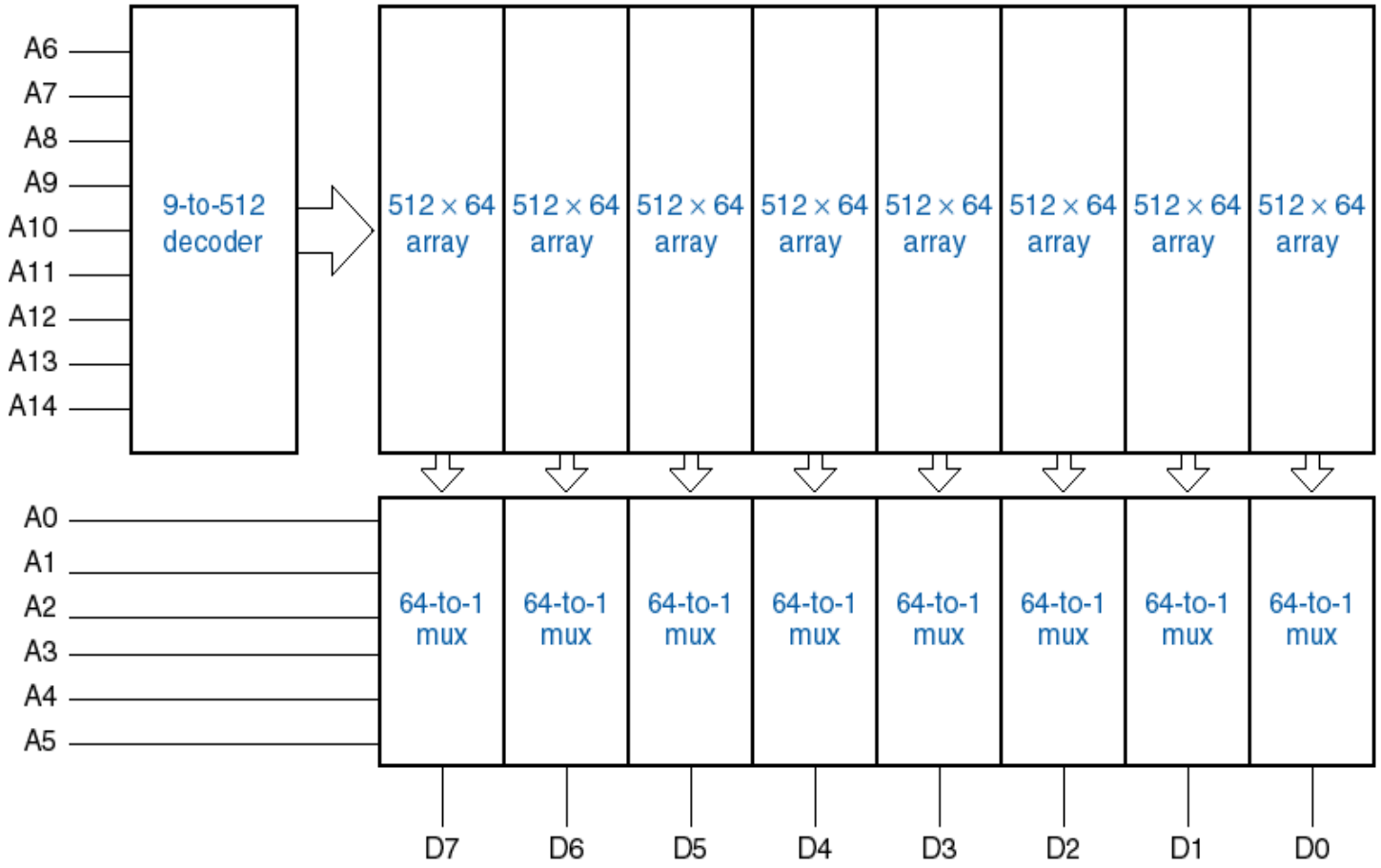
Two-dimensional decoding



Two-dimensional decoding

- As can be seen, two-dimensional decoding allows a 128x1 ROM to be built with a 3-to-8 decoder and a 16-input multiplexer (whose complexity is comparable to that of a 4-to-16 decoder).
- A 1Mx1 Rom could be built with a 10-to-1024 decoder and 1024-input multiplexer. A lot simpler than the one dimensional alternative.
- Additional benefit to reduction of decoding complexity is that two-dimensional decoding has one other benefit –- it leads to a chip whose physical dimensions are close to square -- important for chip fabrication and packaging.
- In ROMs with multiple data outputs the storage arrays corresponding to each data output may be made narrower in order to achieve an overall chip layout that is closer to square. For example, the next figure shows the possible layout of a 32K x 8 ROM.

Possible layout of a 32K x 8 ROM



Commercial ROM Types

- A modern ROM is fabricated as a single IC chip; one that stores 4M bits with a price under $5.
- Various methods are used to “program” the information stored in a ROM:
  - Mask Programmable ROMs.
    - Manufacturer has to be provided with the ROM content in order to create one or more customized masks to manufacture ROMs with the required pattern.
    - ROM manufacturers impose a mask charge of several thousand dollars for the customized aspects of mask-ROM production. Because of mask charges and the four-week delay typically required to obtain programmed chips, mask ROMs are normally used today only in very high-volume applications.
    - For a low-volume applications there are more cost-effective choices, discussed next.

Commercial ROM Types

- Programmable read-only memory (PROM)
  - Similar to a mask ROM, except that the customer may store data values (program the PROM) in just a few minutes.
  - PROM is manufactured with all of its diodes or transistors “connected”. This corresponds to having all desired bits at a particular value (typically “1”). The PROM programmer can be used to set desired bits to the opposite value.
  - In bipolar PROMs this is done by vaporizing tiny fusible links inside the PROM corresponding to each bit.
  - A link is vaporized by selecting it using the PROM’s address and data lines, and then applying a high-voltage pulse (10-30V) to the device through a special input pin.
  - Early reliability problems with vaporized links technology were solved and reliable fusible-link technology is used now days not only in bipolar PROMs but also in the bipolar PLD circuits.

Commercial ROM Types

- Erasable programmable read-only memory (EPROM):
  - EPROM is programmable just like PROM.
  - In addition it also can be “erased” to all 1s-state by exposing it to ultra-violet light.
  - EPROM uses a different technology called “floating-gate MOS”.
  - EPROM manufacturers guarantee that a properly programmed bit will retain 70% of its charge for at least 10 years even if the part is stored at 125o C.
  - Insulating material surrounding the floating gate becomes slightly conductive if it is exposed to ultraviolet light with a certain wavelength which provides for the EPROM content to be erased.
  - Most common application of EPROMs is to store programs in microprocessor systems.
  - EPROMs are typically used during development. ROMs and PROMs are used once the program is finalized because usually they cost less than EPROMs of similar capacity.

Commercial ROM Types

- Electrically Erasable Programmable Read-Only Memory (EEPROM).
  - It is like and EPROM except that individual stored bits may be erased electrically.
  - Floating gates in an EEPROM are surrounded by a much thinner insulating layer and can be erased by applying a voltage of the opposite polarity as the charging voltage to the non-floating gate.
  - Large EEPROMs (1M bit and larger) allow erasing only in fixed-size blocks, typically 128-512 Kbits (16-64 Kbytes) at a time. These memories are typically called flash EPROMs or flash memories.
  - EEPROM can be reprogrammed only a limited number of times (Insulating layer wares off).

ROM-based Design Disadvantages

- For a simple to moderately complex functions, a ROM-based circuit
  - may cost more,
  - consume more power, or
  - run slower than a circuit using a few SSI/MSI devices and PLDs or small FPGA.
- For functions more than 20 inputs, a ROM-based circuit is impractical because of the limit on ROM sizes that are available. For example, one wouldn’t build a 16-bit adder in ROM – it would require billions and billions of bits.