Implementation Technologies

- We can implement a design with many different implementation technologies - different implementation technologies offer different tradeoffs
  - VHDL Synthesis offers an easy way to target a model towards different implementations
  - There are also retargeting tools which will convert a netlist from one technology to another (from a standard cell implementation to a Field Programmable Gate Array implementation).

Available implementation technologies

- Full Custom
- Standard Cell
- Gate Array
- Field Programmable Gate Arrays (FPGAs)
- Complex PLDs (CPLDs)
- Programmable Logic Devices (PLDs)

Full Custom

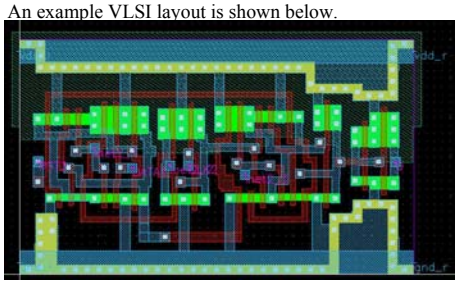
- Designer hand draws geometries which specify transistors and other devices for an integrated circuit. Designer must be an expert in VLSI (Very Large Scale Integration) design.
- Can achieve very high transistor density (transistors per square micron); unfortunately, design time can be very long (multiple months).
- Involves the creation of a a completely new chip, which consists of about a dozen masks (for the photolithographic manufacturing process). Mask creation is the expensive part.

Full Custom (cont)

- Offers the chance for optimum performance. Performance is based on available process technology, designer skill, and CAD tool assistance.
- Fabrication costs are high - all custom masks must be made so non-recurring engineering costs (NRE) is high (in the thousands of dollars). If required number of chips is high then can spread these NRE costs across the chips.
- The first custom chip costs you about $200,000, but each additional one is much cheaper.

Full Custom (cont)

- Fabrication time from geometry submission to returned chips is at least 6-8 weeks.
- Full custom is currently the only option for mixed Analog/Digital chips.
- An example VLSI layout is shown below.

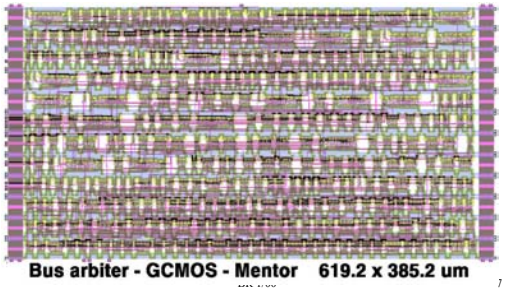


Standard Cell

- Designer uses a library of standard cells; an automatic place and route tool does the layout. Designer does not have to be a VLSI expert.
- Transistor density and performance degradation depends on type of design being done. Not bad for random logic, can be significant for datapath type designs.
  - Quality of available library and tools make a signficant difference.
- Design time can be much faster than full custom because layout is automatically generated.

Standard Cell (cont)

- Still involves creation of custom chip so all masks must still be made; manufacturing costs same as full custom.
- Fabrication time same as full custom.



Gate Array

- Designer uses a library of standard cells. The design is mapped onto an array of transistors which is already created on a wafer; wafers with transistor arrays can be created ahead of time. A routing tool creates the masks for the routing layers and "customizes" the pre-created gate array for the user's design.
- Transistor density can be almost as good as standard cell. Design time advantages are the same as for standard cell.
- Performance can be very good; again, depends on quality of available library and routing tools.

Gate Array (cont)

- Fabrication costs are cheaper than standard cell or full custom because the gate array wafers are mass produced; the non recurring engineering csts are lower because only a few (1-3) unique routing masks have to be created for each design.
- Fabrication time can be extremely short (1-2 weeks) because the wafers are already created and are only missing the routing layers. The more routing layers, the higher the cost, the longer the fabrication time, but the better usage of the available transistors on the gate array.
- Almost all high volume production of complex digital designs are done in either Standard Cell or Gate Array
  - Gate arrays used to be more popular, but recently Standard cells has shown a resurgence in use.

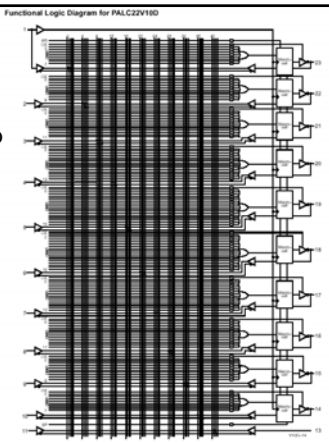
Programmable Logic

- Logic devices which can be programmed/configured on the desktop.
- Three families (in increasing density)
  - PALS (Programmable Array Logic), Programmable Logic Devices
  - Complex PLDs
  - Field Programmable Gate Arrays
- It should be noted that memories are the earliest type of programmable logic

PALs (Programmable Array Logic)

- An early type of programmable logic - still in common use today.
- Logic is represented in SOP form (Sum of Products)
- The number of PRODUCTs in an SOP form will be limited to a fixed number (usually 4-10 Product terms).
- The number of VARIABLEs in each product term limited by number of input pins on PLD (usually a LOT, minimum of 10 inputs
- The number of independent functions limited by number of OUTPUT pins.

22V10 PLD



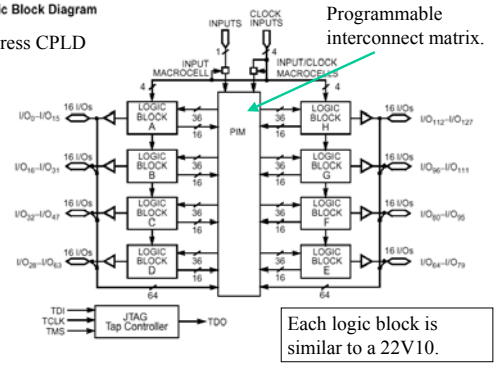
Complex PLDs

- What is the next step in the evolution of programmable logic?
  - More gates!
- How do we get more gates? We could put several PALs on one chip and put an interconnection matrix between them!!
  - This is called a *Complex PLD (CPLD).*

Logic Block Diagram

Cypress CPLD

Programmable interconnect matrix.

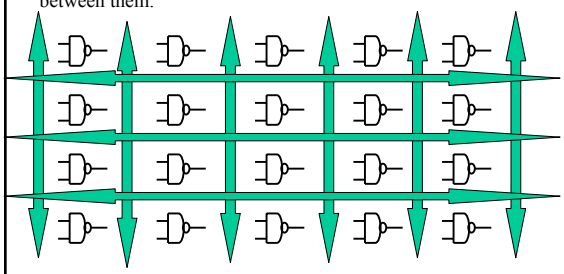


Each logic block is similar to a 22V10.

Figure 1. Ultra37128 Block Diagram

Any other approaches?

Another approach to building a "better" PLD is place a lot of primitive gates on a die, and then place programmable interconnect between them:



Field Programmable Gate Arrays

The FPGA approach to arrange primitive logic elements (logic cells) arrange in rows/columns with programmable routing between them.

What constitutes a primitive logic element? Lots of different choices can be made! Primitive element must be classified as a "complete logic family".

- A primitive gate like a NAND gate
- A 2/1 mux (this happens to be a complete logic family)
- A Lookup table (I.e, 16x1 lookup table can implement any 4 input logic function).

Often combine one of the above with a DFF to form the primitive logic element.

Other FPGA features

- Besides primitive logic elements and programmable routing, some FPGA families add other features
- Embedded memory
  - Many hardware applications need memory for data storage. Many FPGAs include blocks of RAM for this purpose
- Dedicated logic for carry generation, or other arithmetic functions
- Phase locked loops for clock synchronization, division, multiplication.

Other FPGA Comments

- Performance is usually several factors to an order of magnitude lower than standard cell. Performance depends heavily on quality of FPGA technology.
- Design time advantages are the same as for standard cell (use same type of cell/macro library).
- Densities are an order of magnitude lower than standard cell but an order of magnitude higher than normal PLDs.
- Very good for prototype design because many FPGAs are re-usable. Can be used to prototype and verify designs before investing in technologies with high start-up costs (e.g. full custom).

Programmability Options

- PLDs, CPLDs, and FPGAs have different types of programmability.
- One time programmable: Part is programmed once and holds its programming "forever". Not reusable, but usually the cheapest.
- UV-Erasable: Erasable with UV light. Needs a ceramic package with window; package adds expense to part. Programming retained after power down. Programming/Erasing limited to 1000s of cycles.
- Electrically Erasable: Both reprogramming and erasing is electrical. Part can programmed/erased on circuit board, no special packaging needed. Erase time much faster than UV erase. Programming retained after power down. Programming/Erasing limited to 1000s of cycles.

Programmability Options (cont.)

- Static Random Access Memory (SRAM) Programming:
  - Configuration bits are stored in SRAM. Can be reprogrammed infinite number of times.
  - Programming contents NOT retained after power down; FPGA must be 'configured' every time on power up.
  - External non-volatile memory device required to hold device programming; on power up contents of external device transferred to FPGA to configure the device.
  - Altera, Xilinx corporations offer this type of FPGAs.
- Highest density FPGAs use SRAM for configuration bits.

Comparing Technologies - Density (gates per chip)

- Highest to lowest density: Full Custom, Standard Cell, Gate Array, FPGAs, CPLD, PLD
- Full Custom, Standard Cell, Gate Array are called ASIC technologies (Application Specific Integrated Circuit). Large Density gap between ASIC technologies and Programmable logic technologies (FPGAs, CPLD, PLD).
- Highest end FPGA density is now equal to low-end ASIC density (i.e., hundreds of thousands of gates with embedded SRAMs).

Comparing Technologies - Speed

- Highest to lowest performance: Full Custom, Standard Cell, Gate Array, PLDs, CPLDs, FPGAs.
- Again, large performance gap between ASIC technologies and programmable technologies.
- Performance of programmable technologies is in reverse order of their densities.

Comparing Technologies - Cost

- Depends heavily on volume. If only need a few hundred, then FPGAs can be cheaper. If need thousands, then ASIC technologies are cheaper.
- NRE cost (non-recurring engineering costs) are higher for ASIC techologies than FPGAs
- Per-unit-cost (chip cost) higher for FPGAs

Summary

- Full custom can give best density and performance
- Faster design time and ease of design are principle advantages of gate array and standard cell over full custom.
- Fast fabrication time and lower cost are principle advantages of gate arrays over standard cell.
- Gate arrays offer much higher density over FPGAs and are cheaper than FPGAs in volume production.

Summary (cont.)

- FPGAs principle advantage over gate arrays is 'instant' fabrication time (programmed on desktop). FPGAs are also cheaper than gate arrays in low volume. Densities are reaching 100's of thousands of gates/chip. Can be used to prototype full custom/standard cell designs.
- PLDs still hold a speed advantage over most FPGAs and are useful primarily for high speed decoding and speed critical glue logic.