From design to product

- Specification of design constraints/goals
- Investigation of concepts/feasibility
- Logic design
- Verification
- Static timing analysis
- Tape Release
- Implementation
- Electrical Debug
- Ship the part!
Programmable Logic

- Use a single chip (or a small number of chips)
- Program it for the circuit you want
- No reason for the circuit to be small

Programmable Logic Technologies

- Fuse and anti-fuse
  - Fuse makes or breaks link between two wires
  - Typical connections are 50-300 ohm
  - One-time programmable (testing before programming?)
  - Very high density
- EPROM and EEPROM
  - High power consumption
  - Typical connections are 2K-4K ohm
  - Fairly high density
- RAM-based
  - Memory bit controls a switch that connects/disconnects two wires
  - Typical connections are .5K-1K ohm
  - Can be programmed and re-programmed in the circuit
  - Low density
Field-Programmable Gate Arrays

- PALs, PLAs = 10 - 100 Gate Equivalents
- Field Programmable Gate Arrays = FPGAs
  - Altera MAX Family
  - Actel Programmable Gate Array
  - Xilinx Logical Cell Array
- 100 - 1000(s) of Gate Equivalents!

Field-Programmable Gate Arrays

- Logic blocks
  - To implement combinational and sequential logic
- Interconnect
  - Wires to connect inputs and outputs to logic blocks
- I/O blocks
  - Special logic blocks at periphery of device for external connections
- Key questions:
  - How to make logic blocks programmable?
  - How to connect the wires?
  - After the chip has been fabricated?
Tradeoffs in FPGAs

- Logic block - how are functions implemented: fixed functions (manipulate inputs) or programmable?
  - Support complex functions, need fewer blocks, but they are bigger so less of them on chip
  - Support simple functions, need more blocks, but they are smaller so more of them on chip
- Interconnect
  - How are logic blocks arranged?
  - How many wires will be needed between them?
  - Are wires evenly distributed across chip?
  - Programmability slows wires down – are some wires specialized to long distances?
  - How many inputs/outputs must be routed to/from each logic block?
  - What utilization are we willing to accept? 50%? 20%? 90%?

Xilinx Programmable Gate Arrays

- CLB - Configurable Logic Block
  - 5-input, 1 output function
  - or 2 4-input, 1 output functions
  - optional register on outputs
  - Built-in fast carry logic
  - Can be used as memory
  - Three types of routing
    - direct
    - general-purpose
    - long lines of various lengths
  - RAM-programmable
    - can be reconfigured
The Xilinx 4000 CLB

Two 4-in functions, registered out

Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)
5-in function, combinational out

Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)

Xilinx 4000 Interconnect

Figure 28: Single- and Double-Length Lines, with Programmable Switch Matrices (PSMs)
Switch Matrix

Figure 26: Programmable Switch Matrix (PSM)

Xilinx 4000 Interconnect Details
Combinational Logic Examples

- Key: General functions are generally limited to 5 inputs
  - (4 even better - 1/2 CLB)
  - No limitation on function complexity

- Example
  - 2-bit comparator:
    \[ A \cdot B = C \cdot D \text{ and } A \cdot B > C \cdot D \text{ implemented with 1 CLB} \]
    \[ (GT) \quad F = A \cdot C' + A \cdot B \cdot D' + B \cdot C \cdot D' \]
    \[ (EQ) \quad G = A' \cdot B' \cdot C' \cdot D' + A' \cdot B \cdot C' \cdot D + A \cdot B' \cdot C \cdot D' + A \cdot B \cdot C \cdot D \]

- 4-bit binary adder:
Can't design FPGAs by hand
- Way too much logic to manage, hard to make changes

Hardware description languages
- Specify functionality of logic at a high level

Validation: high-level simulation to catch specification errors
- Verify pin-outs and connections to other system components
- Low-level to verify mapping and check performance

Logic synthesis
- Process of compiling HDL program into logic gates and flip-flops

Technology mapping
- Map the logic onto elements available in the implementation technology (LUTs for Xilinx FPGAs)

Placement and routing
- Assign logic blocks to functions
- Make wiring connections

Timing analysis - verify paths
- Determine delays as routed
- Look at critical paths and ways to improve

Partitioning and constraining
- If design does not fit or is unroutable as placed split into multiple chips
- If design it too slow prioritize critical paths, fix placement of cells, etc.
- Few tools to help with these tasks exist today

Generate programming files - bits to be loaded into chip for configuration
Xilinx CAD Tools

- Verilog (or VHDL) use to specify logic at a high-level
  - Combine with schematics, library components
- Synopsys
  - Compiles Verilog to logic
  - Maps logic to the FPGA cells
  - Optimizes logic
- Xilinx APR - automatic place and route (simulated annealing)
  - Provides controllability through constraints
  - Handles global signals
- Xilinx Xdelay - measure delay properties of mapping and aid in iteration
- Xilinx XACT - design editor to view final mapping results