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

Details of One Virtex Slice

Implements any Two 4-input Functions

4-input function
3-input function; registered

Implements any 5-input Function

5-input function
Implement Some Larger Functions

- e.g. 9-input parity

Two Slices: Any 6-input Function

- 6-input function

from other slice

Two Slices: Implement some larger functions

- e.g. 19-input parity

from other slice

Fast Carry Chain: Add two bits per slice

- Carry(a, b, cin)
- Sum(a, b, cin)

Lookup Tables used as memory (16 x 2)
- Distributed Memory

Lookup Tables used as memory (32 x 1)
Block RAM

Non-Local Routing

- Hex wires
- Extend 6 CLBs in one direction
- Connections of 3 and 6 CLBs
- "Express busses"
- Take advantage of many metal layers
- Long wires
- Extend the length/height of the chip
- Global signals
e.g. clk, reset
- Tri-state busses
- Extend across the chip
- Use for datapath bit-slice

Virtex Routing

Using the DLL to De-Skew the Clock

Using the DLL to De-Skew the Clock

Virtex Routing

Virtex IOB

Figure 6: Dual-Port Block SelectRAM

Figure 7: Virtex-E Local Routing

Figure 8: DLL De-skew of Board Level Clock

Figure 9: Virtex-E Input/Output Block (IOB)
Computer-aided Design

- 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)

Applications of FPGAs

- Implementation of random logic
- easier changes at system-level (one device is modified)
- can eliminate need for full-custom chips
- Prototyping
  - ensemble of gate arrays used to emulate a circuit to be manufactured
  - get more/better/faster debugging done than possible with simulation
- Reconfigurable hardware
  - one hardware block used to implement more than one function
  - functions must be mutually-exclusive in time
  - can greatly reduce cost while enhancing flexibility
  - RAM-based option
- Special-purpose computation engines
  - hardware dedicated to solving one problem (or class of problems)
  - accelerators attached to general-purpose computers

CAD Tool Path (cont’d)

- 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 unrouteable as placed split into multiple chips
  - if design 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
- Synplify
  - compiles Verilog to logic
  - maps logic to the FPGA cells
  - optimizes logic
- Xilinx APEX - automatic place and route (simulated annealing)
  - provides controllability through constraints
  - handles global signals
- Xilinx Xdsay - measure delay properties of mapping and aid in iteration
- Xilinx XACT - design editor to view final mapping results