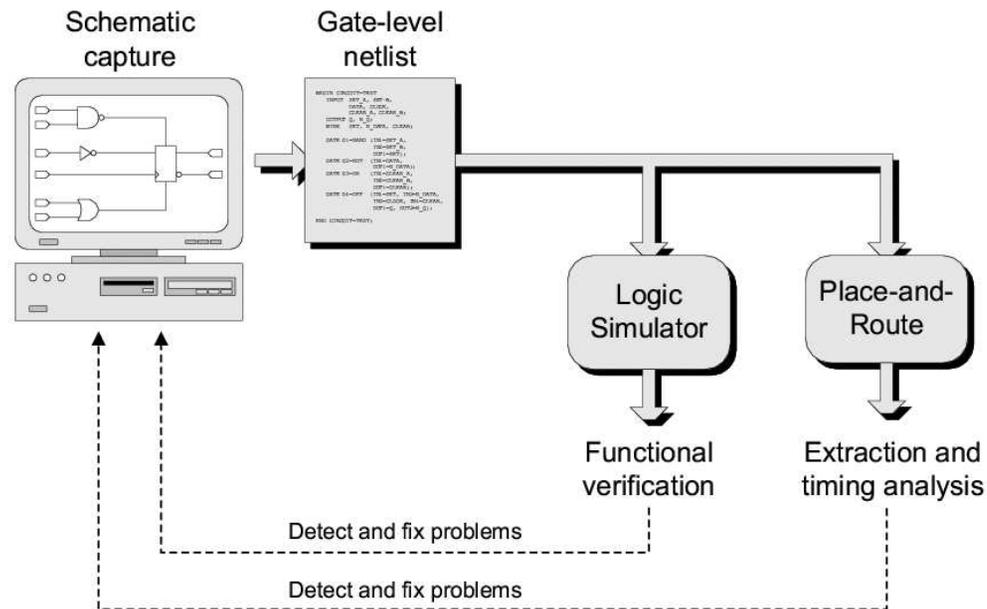


Schematic-Based Design Flows

Early schematic-driven ASIC flow



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

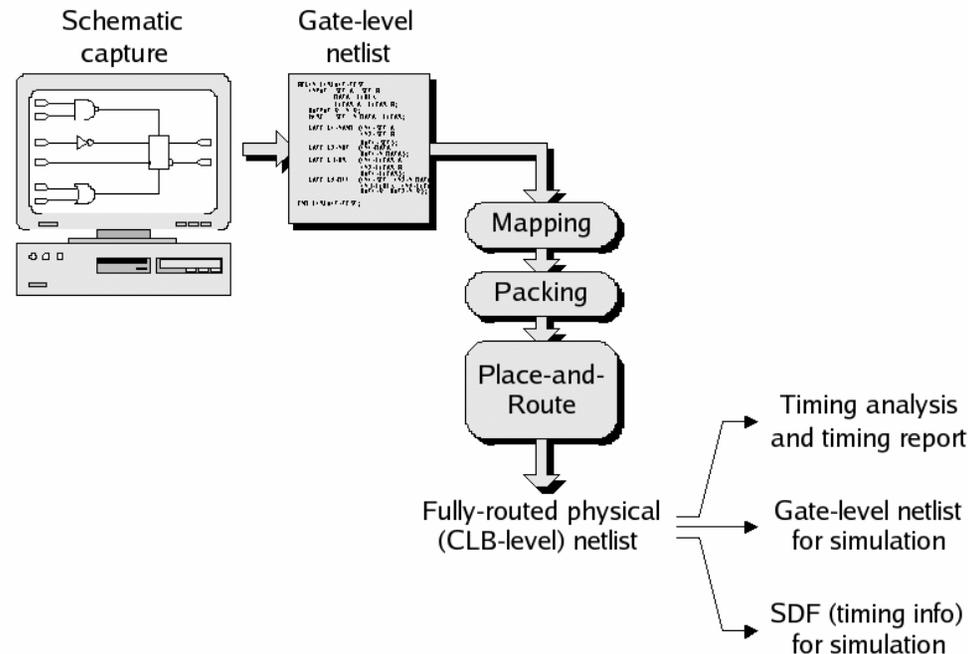
Designer could access symbols for logic gates and functions from a library.

Simulator would use a corresponding library with logic functionality defined as well as timing information.

End result: Place and route software used the netlist to implement the design on the physical silicon chip.

Schematic-Based Design Flows

Early schematic-driven FPGA flow



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

Front end of FPGA flow is the same, w.r.t. library and simulation.

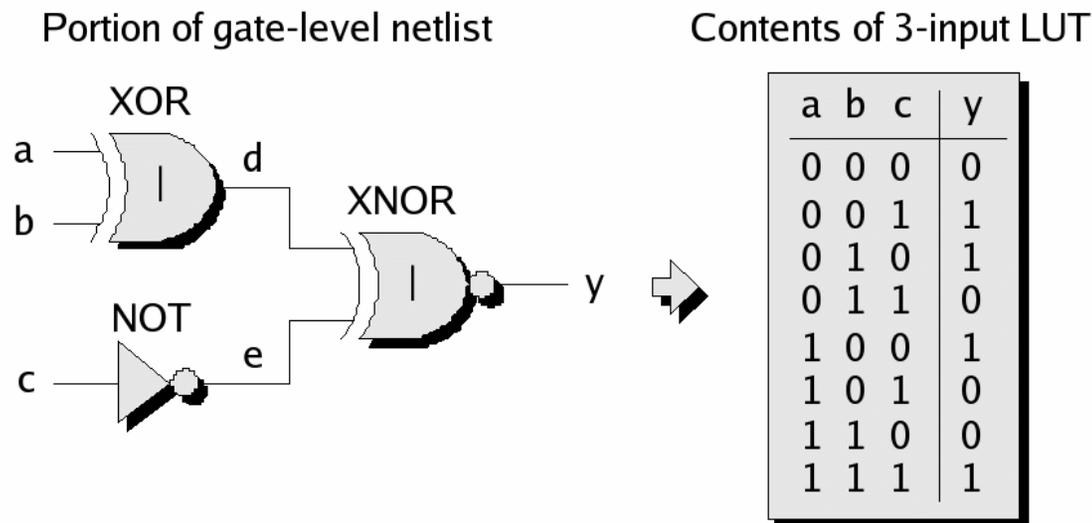
However, implementation portion of the flow is different because the netlist is mapped onto the CLBs.

Mapping and *packing* steps are added to the flow to handle this.

Schematic-Driven FPGA Flow

Mapping refers to the process of associating entities such as the gate-level functions in the netlist with the LUT-level functions on the FPGA.

The mapping is obviously *not* one-to-one because a LUT can be used to represent several logic gates.



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

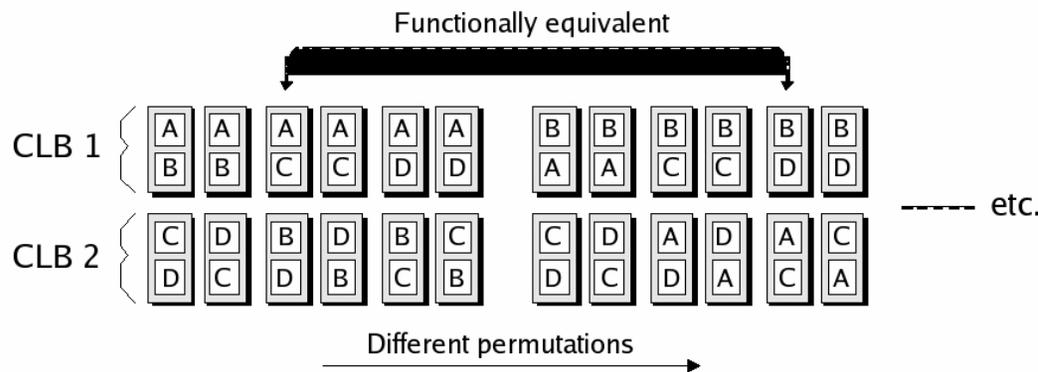
Mapping is a non-trivial problem b/c there are many ways in which a netlist can be partitioned and mapped into a LUT.

Schematic-Driven FPGA Flow

Packing follows the *mapping* step in which the LUTs and registers are packed into CLBs.

This step is also non-trivial b/c there are many ways to combine and permute the LUTs to CLBs.

For example, assume we have a netlist that maps to 4 LUTs and the FPGA can contain 2 LUTs per CLB, then there are 4! different ways to pack the LUTs.



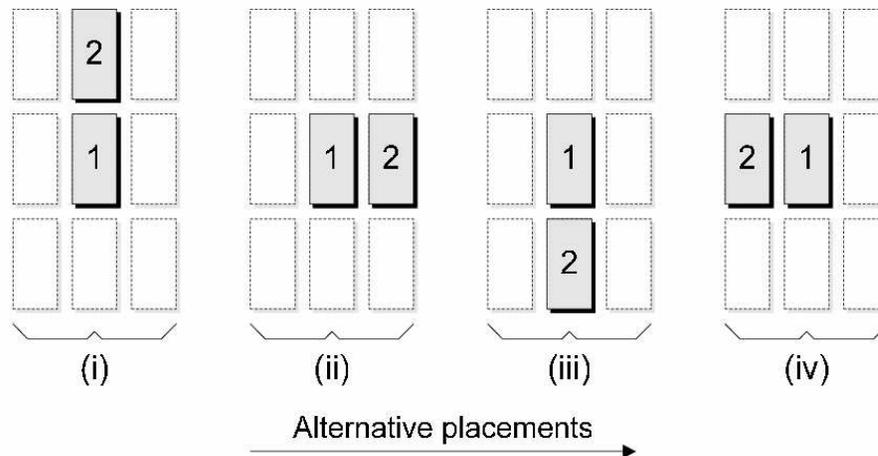
The Design Warrior's Guide to FPGAs,
 ISBN 0750676043,
 Copyright(C) 2004 Mentor Graphics Corp

Although some of these are equivalent, e.g., AC-BD and BD-AC, the number of possibilities is still very large.

Schematic-Driven FPGA Flow

Place-and-Route refers to selecting the CLBs and configuring the interconnect.

Assume there are 2 CLBs that need to be connected together and, further, that they need to be adjacent to each other.



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

Since there are only 2 LUTs and 4 possibilities here, this task is fairly simple.

However, with hundreds of thousands of CLBs, each of which may connect to one or more CLBs, finding optimal placement is non-trivial.

Schematic-Driven FPGA Flow

Following place-and-route, *static timing analysis* (STA) can be run on the physical representation to calculate I/O and internal path delays.

Timing violations are also checked here, i.e., setup and hold times.

At this point, designers may wish to re-run their timing simulations with accurate, *post-place-and-route*, timing information.

The tools need to generate a *new netlist*, that reflects the mapping, packing and place-and-route, and has timing information.

The format used has been standardized and is called *standard delay format* (SDF).

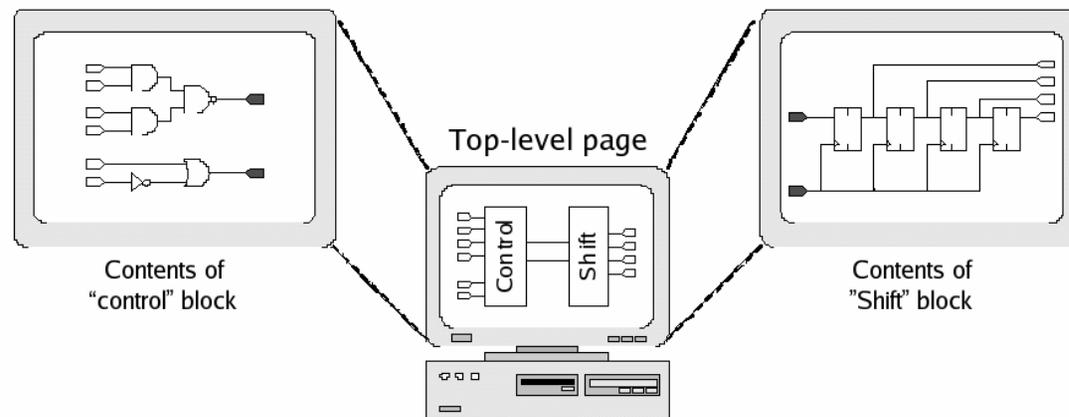
The original schematic editors forced the users to create their designs *as a single flat schematic*.

Pages were used to partition the design.

This created problems with reusability and modularity (multiple instantiations of some portion of the design), as well as visualization.

Schematic-Driven FPGA Flow

Today's packages support the notion of *hierarchy*, that allowed a design to be decomposed and implemented as sub-blocks.



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

FPGA vendors offered (still offer) these tools at a low cost, since their focus is on building/selling chips.

Over time, external EDA vendors started to support portions of the flow including schematic capture, mapping and packing, and did a better job.

FPGA companies still maintain control of the place-and-route software however.