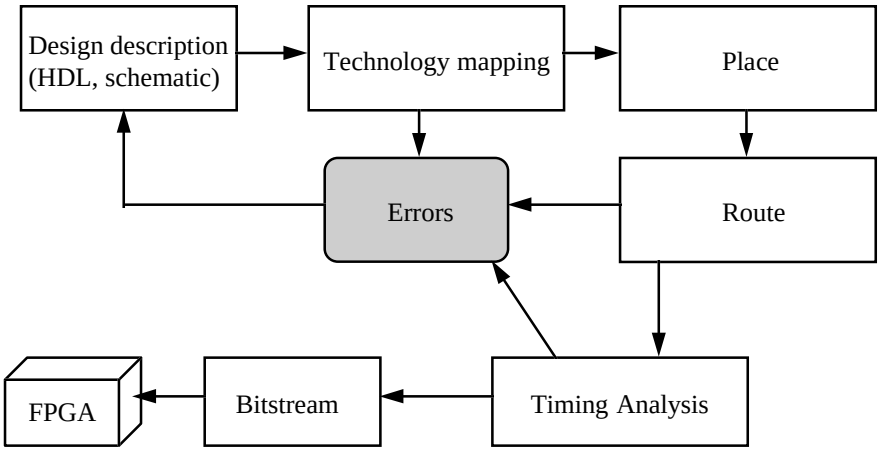
# REC FPGA Seminar IAP 1998

Session 2:

Design Tools, Configuration, and other Practical Considerations



# Design Tools: Process



# Design Tools: Design Entry

- HDL
  - Verilog, VHDL or proprietary language (AHDL, etc.)
    - verilog is like C with multithreading and strict typing
    - VHDL stands for VHSIC HDL; intended for detailed simulations; commissioned by the military; very complex
  - Ideal for large designs because of well-defined scoping and instantiation rules; top down design
  - Also ideal for state machines, decoders/encoders, and odd or awkward busses
  - Hardware mapping is difficult
    - very easy to make inefficient designs; subtle semantics choices can lead to drastic perfomance variations
    - hard to specify hardware-specific features such as carry chains
    - hard to specify placement and routing info



# Design Tools: Design Entry

- Schematic entry
  - more intuitive, easier to observe design flow
  - helpful when trying to optimize designs for speed or area
  - difficult when implementing large amounts of miscellaneous logic (state machines)
  - heirarchical schematic tools help make large designs more manageable
  - global changes difficult (hard to change global mistakes)
  - hardware mapping is much easier
    - schematic primitives for special hardware features
    - schematic attributes for routing info
    - WYSIWYG design entry

## Design Tools: Hardware Mapping

- Many options for HDL to hardware mapping
  - vendor-specific options
  - third party tools
  - EDIF is the most common intermediate language
- When using HDLs, good hardware mapping tools are critical for perfomance and device utilization
- Deep understanding of HDL is also useful
- Schematic hardware mapping is much easier - very close to WYSIWYG editing
- Hardware mappers often perform aggressive logic optimizations - watch your assumptions! (hazards)



## Design Tools: Place and Route

- Place and route tools are always vendor-specific
- Much progress remains in place and route tools
  - typical P&R times for a reasonably complex design is around 30 minutes to an hour
  - device utilization and performance still well below that of hand-placed and routed designs
- Many vendors offer hand-placement or tweaking tools for speed and area critical applications
- Partial compilation of macros in the works

## Design Tools: Timing Analysis

- Especially important for path-dependant delay devices
- Designs often iterate at this point - critical path is extracted and optimized
- Timing analysis tools also have a ways to go
  - difficult to analyze designs with multiple clock domains
  - impossile to analyze designs with combinational loops



## Design Tools: Bitstream management

- Bitstreams can be merged
  - daisy-chained devices

# Configuration

- Applies to ISP devices only
- Many options
  - serial ROM
  - master mode with standard ROM
  - slave of intelligent host or another FPGA in a daisy chain
- serial ROM
  - very popular in ASIC-style applications
  - low pin and parts count, but sometimes slower



# Configuration

- Master mode with parallel ROM device
  - FPGA drives a ROM's address bits and reads data from ROM
  - expensive in terms of pins, but pins can be reused in some designs
  - sometimes faster than serial methods
- Slave modes
  - intelligent host configures FPGA
    - host can be a PC or another FPGA in master mode
  - most flexible method
  - many FPGA architectures allow daisy chaining

# Other Considerations

- Design for the future
  - vertical migration
  - largest FPGA for your budget
- Be wary of logic interface levels and new low voltage devices
- Pin-locking
  - some FPGA architectures perform very poorly under pin-locking (Altera 8K in particular)
  - all architectures experience some performance loss under pin-locking
- I/O count
  - many designs are I/O limited, not logic limited
- System performance, not just logic performance
  - includes I/O times and routing times, clock skew
  - compare to FF toggle rates often quoted by vendors