THE ECE 554 XILINX DESIGN PROCESS

• Design process overview
• Design references
• Xilinx libraries
• Design tutorial
• What’s next

Design Process Steps

• Definition of system requirements.
  – Example: ISA (instruction set architecture) for CPU.
  – Includes software and hardware interfaces including timing.
  – May also include cost, speed, reliability and maintainability specifications.

• Definition of system architecture.
  – Example: high-level HDL (hardware description language) representation - this is not required in ECE 554 but is done in the real world).
  – Useful for system validation and verification and as a basis for lower level design execution and validation or verification.
Design Process Steps (continued)

- Refinement of system architecture
  - In manual design, descent in hierarchy, designing increasingly lower-level components
  - In synthesized design, transformation of high-level HDL to “synthesizable” register transfer level (RTL) HDL

- Logic design or synthesis
  - In manual or synthesized design, development of logic design in terms of library components
  - Result is logic level schematic or netlist representation or combinations of both.
  - Both manual design or synthesis typically involve optimization of cost, area, or delay.

Design Process Steps (Continued)

- Implementation
  - Conversion of the logic design to physical implementation
  - Involves the processes of:
    - Mapping of logic to physical elements,
    - Placing of resulting physical elements,
    - And routing of interconnections between the elements.
  - In case of SRAM-based FPGAs, represented by the programming bitstream which generates the physical implementation in the form of CLBs, IOBs and the interconnections between them
Design Process Steps (Continued)

- Validation (used at number of steps in the process)
  - At architecture level - functional simulation of HDL
  - At RTL level - functional simulation of RTL HDL
  - At logic design or synthesis - functional simulation of gate-level circuit - not usually done in ECE 554
  - At implementation - timing simulation of schematic, netlist or HDL with implementation based timing information (functional simulation can also be useful here)
  - At programmed FPGA level - in-circuit test of function and timing

Xilinx HDL/Core Design Flow
Xilinx HDL/Core Design Flow
- HDL Editing

**DESIGN WIZARD** ← Accessed within HDL Editor → **LANGUAGE ASSISTANT**

- **HDL Module Frameworks**
- **Language Construct Templates**

**HDL EDITOR**

**RTL HDL Files**

---

Xilinx HDL/core Design Flow
- Core Generation

Select core and specify input parameters

**CORE GENERATOR**

- **EDIF netlist for core_name**
- **HDL instantiation module for core_name**
- **Other core_name files**
Xilinx HDL/core Design Flow
- HDL Functional Simulation

1. HDL instantiation module for core_name
2. Set Up and Map work library
3. RTL HDL Files
4. Testbench HDL Files
5. Compile HDL Files
6. EDIF netlists for core_names
7. Test Inputs or Force Files
8. Functional Simulate
9. Waveforms or List Files

MODELSIM

Xilinx HDL Design Flow
- Synthesis

1. All HDL Files
2. Edit FPGA Express Synthesis Constraints
3. Select Top Level
4. EDIF netlists for core_names
5. Select Target Device
6. Synthesis/Implementation Constraints
7. Synthesize
8. Gate/Primitive Netlist Files (EDIF or XNF)
9. Synthesis Report Files
10. FPGA EXPRESS
Xilinx HDL/core Design Flow - Implementation

1. **Model Extraction**
   - HDL or EDIF for Implemented Design

2. **Netlist Translation**
   - Gate/Primitive Netlist Files (XNF or EDN)

3. **Map**

4. **Place & Route**

5. **Create Bitstream**

6. **Timing Model Gen**

7. **Standard Delay Format File**

8. **XILINX DESIGN MANAGER**

9. **BIT File**

---

Xilinx HDL/core Design Flow - Timing Simulation

1. **Set Up and Map work Directory**

2. **Compile HDL Files**

3. **Compiled HDL**

4. **HDL Simulate**

5. **Waveforms or List Files**

6. **Testbench HDL Files**

7. **Test Inputs, Force Files**

8. **HDL or EDIF for Implemented Design**

9. **Standard Delay Format File**

---

1/21/2003
Xilinx HDL Design Flow
- Programming and In-circuit Verification

- Bit File
  - Input Byte
    - GXSPORT
    - Other Inputs
  - GXSLOAD
  - ECE 554 FPGA Board
  - Outputs

Design References - 1

- There are two Xilinx 4.2i releases
  - 4.2i: uses Synopsys FPGA Express synthesis tool (we use this one)
  - ISE 4.2i: uses Xilinx XST synthesis tool
- The manuals are a bit mixed – Do not use material related to XST
- Manuals (will be provided on website)
  - FPGA compiler II/FPGA express Verilog HDL reference manual - essential guide to writing Verilog for FPGA express - suggest you download and print a copy for your use 2 pages/page
  - Synthesis and simulation design guide - lots of useful information on writing HDL code
  - CORE generator guide - you will use cores lots, so can be useful.
Design References - 2

- Libraries guide - useful only if you want to instantiate parts
- Constraints guide – in particular, useful if you want to use timing constraints
- Foundation series 4.2i installation guide and release notes - good for finding bugs, but always out-of-date - use on-line answers database instead

- The following guides are occasionally useful, but far less frequently:
  - Design manager/flow engine guide
  - Development system reference guide
  - Foundation series 4 user guide
  - FPGA compiler II/FPGA express VHDL reference manual
  - Global Glossary

- Databook, app. Notes, and answers database on-line at:
  http://support.xilinx.com/support/support.htm

Simulation References

- Most useful:
  - ModelSim SE user’s manual

- Occasionally referenced:
  - ModelSim SE command reference
The Xilinx Libraries

- Useful only if you have to instantiate (in your HDL) Xilinx primitives or macros (not all can be instantiated) from the *Libraries guide*.
- Note selection guide includes CLB counts and section at front on notation used to describe macros.

Design Practices

- Use synchronous design.
  - CLBs are actually reading functions from SRAM!
  - Avoid clock gating.
  - Avoid ripple counters.
  - Avoid use of direct sets and resets except for initialization.
  - Synchronize asynchronous signals as needed.
  - Study timing issues handout.
What’s Next

• Verilog HDL – introductory lecture next week will give an overview of Verilog, our HDL language of choice
• HDL/core design flow – design tutorial next week will employ the flow described for a Verilog HDL/core example