

Chapter 1 Introduction to VHDL

VHDL

VHDL - Flaxer EII

Introduction

Ch 1 - 1

Course Objectives Affected

- **Write functionally correct and well-documented VHDL code**, intended for either simulation or synthesis, of any combinational or sequential logic design.
- **Define** and use **the three major styles** of writing VHDL code (structural, dataflow, behavioral).
- **Utilize Warp2, Active HDL (or Altera Max+Plus2) tools to simulate, synthesize, and implement**, in the appropriate technology, any combinational or sequential logic design expressed in VHDL.
- Using ISR method to program and test a real circuit in Ultra37K-EVB evaluation board.

VHDL - Flaxer EII

Introduction

Ch 1 - 2

Outline

- ➔ • **Purpose and Background of VHDL**
- Design Synthesis Process
- Design Tool Flow
- VHDL Example and Styles



VHDL - Flaxer EII

Introduction

Ch 1 - 3

Purpose and Background of VHDL

- **Problem**

- Need a method to quickly design, implement, test, and document increasingly complex digital systems
- Schematics and Boolean equations inadequate for million-gate IC.

- **Solution**

- A hardware description language (HDL) to express the design
- Associated computer-aided design (CAD) or electronic design automation (EDA) tools for synthesis and simulation
- Programmable logic devices for rapid implementation of hardware
- Custom VLSI application specific integrated circuit (ASIC) devices for low-cost mass production

Purpose and Background of VHDL

- Two widely-used HDLs today

- VHDL
- Verilog HDL (from Cadence, now IEEE standard)

- VHDL - Very High Speed Integrated Circuit (VHSIC) Hardware Description Language

- VHDL history

- Created by DOD to document military designs for portability
- IEEE standard 1076 (VHDL) in 1987
- Revised IEEE standard 1076 (VHDL) in 1993
- IEEE standard 1164 (object types standard) in 1993 (**std_logic**).
- IEEE standard 1076.3 (synthesis standard) in 1996 (**numeric_std**).
- IEEE standard 1076.4 (timing standard) in 1996 (**VITAL**).

Purpose and Background of VHDL

- Reasons to use VHDL

- Power and flexibility
- Device-independent design
- Portability among tools and devices
- Device and tool benchmarking capability
- VLSI ASIC migration
- Quick time-to-market and low cost (with programmable logic)

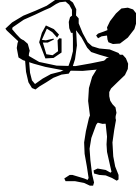


- Problems with VHDL (?)

- Loss of control with gate-level implementation (so what?)
- Inefficient logic implementations via synthesis (engineer-dependent)
- Variations in synthesis quality among tools (always improving)

Outline

- Purpose and Background of VHDL
- ➔ • **Design Synthesis Process**
- Design Tool Flow
- VHDL Example and Styles



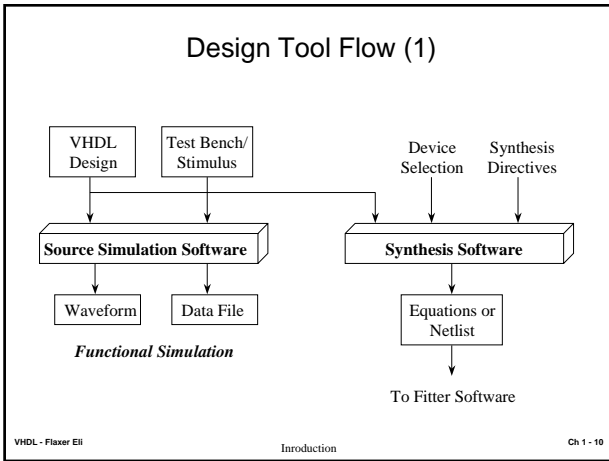
Design Synthesis Process

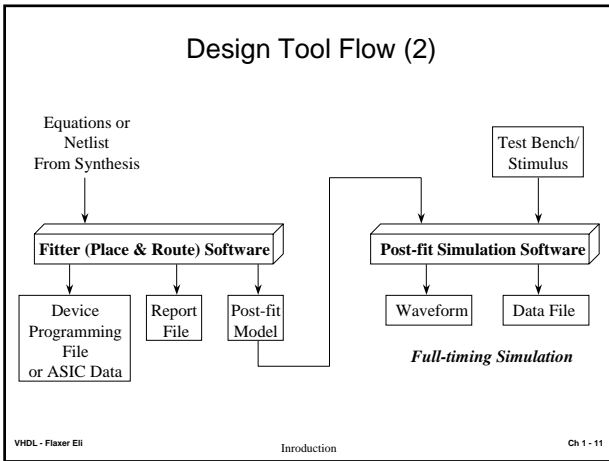
- Define the design requirements
- Describe the design in VHDL
 - Top-down, hierarchical design approach
 - Code optimized for synthesis or simulation
- Simulate the VHDL source code
 - Early problem detection before synthesis
- Synthesize, optimize, and fit (place and route) the design for a device
 - Synthesize to equations and/or netlist
 - Optimize equations and logic blocks subject to constraints
 - Fit into the components blocks of a given device
- Simulate the post-layout design model
 - Check final functionality and worst-case timing
- Program the device (if PLD) or send data to ASIC vendor

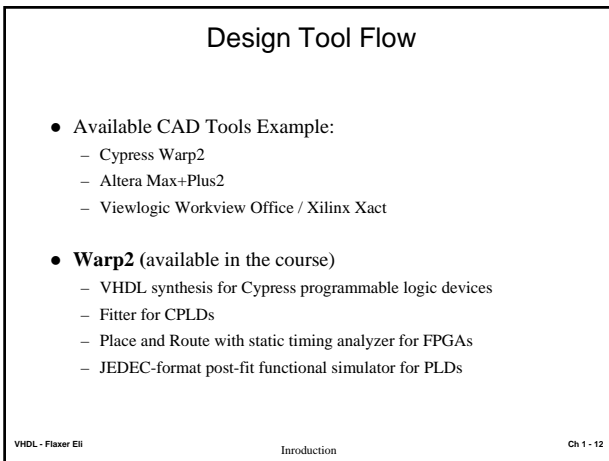
Outline

- Purpose and Background of VHDL
- Design Synthesis Process
- ➔ • **Design Tool Flow**
- VHDL Example and Styles

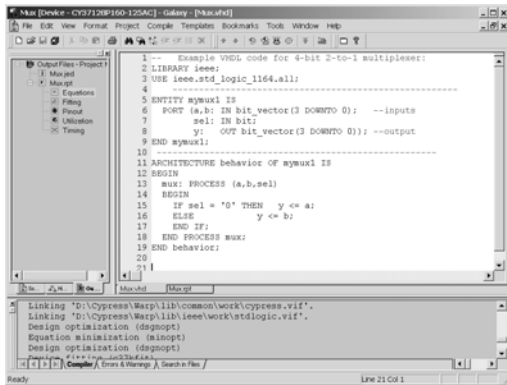








Warp2 Desktop



```
1 -- Example VHDL code for 4-bit 2-to-1 multiplexer:
2 LIBRARY ieee;
3 USE ieee_std_logic_1164.all;
4
5 ENTITY mux1 IS
6 PORT (a,b: IN bit_vector(3 DOWNTO 0); --inputs
7       sel: IN bit2;
8       y:  OUT bit_vector(3 DOWNTO 0)); --output
9 END mux1;
10
11 ARCHITECTURE behavior OF mux1 IS
12 BEGIN
13   mux: PROCESS (a,b,sel)
14   BEGIN
15     IF sel = "0" THEN y <= a;
16     ELSE y <= b;
17   END IF;
18 END PROCESS mux;
19 END behavior;
20
```

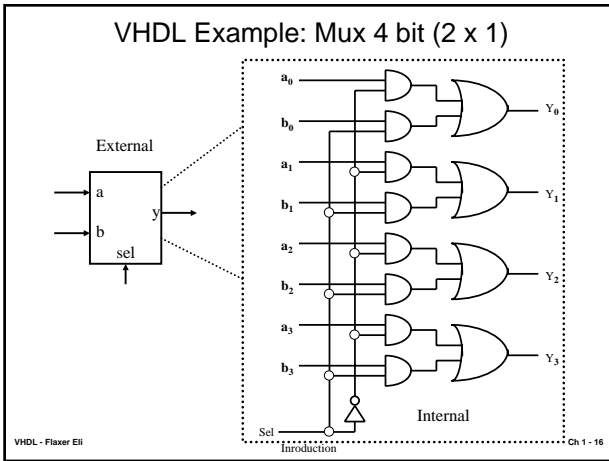
Design Tool Flow

- Altera Max+Plus2
 - VHDL, AHDL, schematic entry
 - Synthesis and fitter for Altera PLDs and FPGAs
 - VHDL functional simulator and post-fit full-timing simulator
- Viewlogic Workview Office / Xilinx Xact
 - VHDL, Verilog, schematic entry
 - Synthesis for many vendors PLDs, FPGAs, and ASICs
 - VHDL functional simulator and post-fit full-timing simulator
 - Xilinx Xact place and route for Xilinx FPGAs

Outline

- Purpose and Background of VHDL
- Design Synthesis Process
- Design Tool Flow
- ➔ • VHDL Example and Styles





VHDL Example and Styles

```

-- Example VHDL code for 4-bit 2-to-1 multiplexer:
LIBRARY ieee;
USE ieee.std_logic_1164.all;

-----
ENTITY mymux1 IS
  PORT (a,b: IN bit_vector(3 DOWNTO 0); --inputs
        sel: IN bit;
        y:  OUT bit_vector(3 DOWNTO 0)); --output
END mymux1;
-----
ARCHITECTURE behavior OF mymux1 IS
BEGIN
  mux: PROCESS (a,b,sel)
  BEGIN
    IF sel = '0' THEN y <= a;
    ELSE y <= b;
    END IF;
  END PROCESS mux;
END behavior;
  
```

VHDL - Flaxer Eli Introduction Ch 1 - 17

VHDL Example: Synthesis Result

```

-----
PLD Compiler Software: C37KFIT.EXE 22/DEC/2000 [v4.02] 6.2 IR 27
DESIGN EQUATIONS (10:47:48)
y(0) =
  b(0) * sel
  + a(0) * /sel

y(1) =
  b(1) * sel
  + a(1) * /sel

y(2) =
  b(2) * sel
  + a(2) * /sel

y(3) =
  b(3) * sel
  + a(3) * /sel

Completed Successfully
-----
  
```

VHDL - Flaxer Eli Introduction Ch 1 - 18

VHDL Example and Styles

- Levels of Abstraction (Architectural Styles):
- **Behavioral**
 - High level, algorithmic, sequential execution
 - Hard to synthesize well
 - Easy to write and understand (like high-level language code)
- **Dataflow**
 - Medium level, register-to-register transfers, concurrent execution
 - Easy to synthesize well
 - Harder to write and understand (like assembly code)
- **Structural**
 - Low level, netlist, component instantiations and wiring
 - Trivial to synthesize
 - Hardest to write and understand (very detailed and low level)

Summary

- VHDL and programmable logic are the best current solution for rapid design, implementation, testing, and documenting of complex digital systems.
- A standard 6-step design synthesis process is used with VHDL.
- The general flow of information through standard VHDL synthesis CAD tools was described.
- Features of the three VHDL CAD tools available (Warp2, Max+Plus2, and Workview) were presented.
- A VHDL code example was discussed and the three architectural styles of VHDL were defined.
