PCB Design With HDL Designer

- Motivation
  - Time savings
  - Money savings
  - Simplicity

- Approach
  - Use single tool for PCB and FPGA design
  - More FPGA designs than PCB designers
    - Use HDL designer for schematic capture
PCB Design With HDL Designer
Design Process

- PCB Design Process (Minimal):
  - Schematic Capture
  - Displaying Reference Designators and Component Information on Schematic
  - Netlist Creation and Conversion
PCB Design With HDL Designer
Schematic Capture - Symbols

- Part Symbols
  - HDL Symbol Editor
    - Part Name
    - Part Number
    - Package Type
    - Pin Name (or Port Name)
    - Pin Number
PCB Design With HDL Designer
Schematic Capture - Symbols
PCB Design With HDL Designer
Schematic Capture – Schematic Diagrams

- Schematic Diagrams
  - HDL Block Diagram Editor
    - Part Information from Part Symbol
      - Pin Number
      - Package Type
      - Part Number
    - Connection Information
      - Nets
      - Reference Designators
PCB Design With HDL Designer
Schematic Capture – Schematic Diagrams

Package List
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

Declarations
Ports
\( a \) : \text{end} logic
\( b \) : \text{end} logic
\( c \) : \text{end} logic
\( \text{pin}_a \) : \text{end} logic

Diagram Signals
\( \text{VCC} \) : \text{end} logic
\( \text{GND} \) : \text{end} logic

Component Declarations
\( \text{pin}_{-}0 \) = "1" (string)
\( \text{pin}_{-}1 \) = "2" (string)
\( \text{pin}_{-}2 \) = "3" (string)
\( \text{pin}_{-}3 \) = "4" (string)
\( \text{pin}_{-}4 \) = "5" (string)
\( \text{pin}_{-}5 \) = "6" (string)
\( \text{pin}_{-}6 \) = "7" (string)
\( \text{pin}_{-}7 \) = "8" (string)
\( \text{pin}_{-}8 \) = "9" (string)
\( \text{pin}_{-}9 \) = "10" (string)
\( \text{pin}_{-}10 \) = "11" (string)
\( \text{pin}_{-}11 \) = "12" (string)
\( \text{pin}_{-}12 \) = "13" (string)
\( \text{pin}_{-}13 \) = "14" (string)

ports
\( \text{pin}_{-}0 \) = "I" (string)
\( \text{pin}_{-}1 \) = "2" (string)
\( \text{pin}_{-}2 \) = "3" (string)
\( \text{pin}_{-}3 \) = "4" (string)
\( \text{pin}_{-}4 \) = "5" (string)
\( \text{pin}_{-}5 \) = "6" (string)
\( \text{pin}_{-}6 \) = "7" (string)
\( \text{pin}_{-}7 \) = "8" (string)
\( \text{pin}_{-}8 \) = "9" (string)
\( \text{pin}_{-}9 \) = "10" (string)
\( \text{pin}_{-}10 \) = "11" (string)
\( \text{pin}_{-}11 \) = "12" (string)
\( \text{pin}_{-}12 \) = "13" (string)
\( \text{pin}_{-}13 \) = "14" (string)

Part Numbers
\( \text{part}_{-}0 \) = "part_{-}008" (string)
\( \text{pkg}_{-}type \) = "dip44" (string)
PCB Design With HDL Designer
VHDL Coding

- Issue – Where to include Pin Numbers in VHDL?
  - Comments
    - Possible
  - VHDL Attributes
    - Good Approach but not displayed on Block Diagrams in HDL Designer
  - VHDL Generics
    - Chosen Approach but displays pin numbers as a block of text
PCB Design with HDL Designer
Netlist Conversion

Issue – How can a PADS netlist be produced?

Comments

Possible

VHDL Attributes

Good Approach but not displayed on Block Diagrams in HDL Designer

VHDL Generics

Chosen Approach but displays pin numbers as a block of text

```vhdl
GENERIC(
    pin_a0 : string := "1";
    pin_a0 : string := "2";
    pin_a1 : string := "3";
    pkg_type : string := "dip14";
    part_num : string := "part_ac04"
);
```
Conclusion

- Approach can be used PCB design
  - Would like Vendor to study modifications to HDL Designer
    - Schematic Display
    - PCB netlist output options
    - Design Rule Checking
    - Part List Generation