



ECE380 Digital Logic

Introduction to Logic Circuits:
CAD Tools and VHDL



Introduction to CAD tools

- A CAD system usually includes the following tools
 - Design entry
 - Synthesis and optimization
 - Simulation
 - Physical design



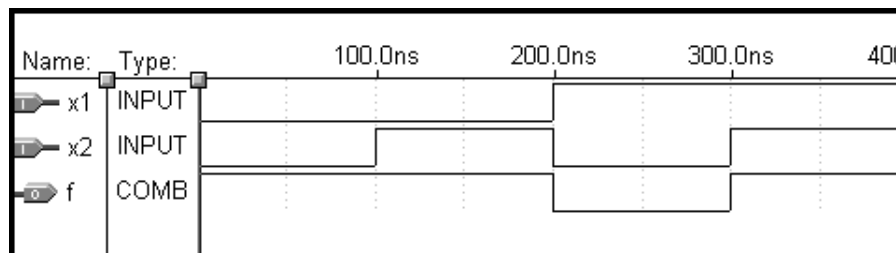
Design entry

- The process of entering into the CAD system a description of a circuit being designed is called ***design entry***
- Three common design entry methods
 - Using truth tables
 - User enters a truth table in plain text format or draws a waveform that represents the desired functional behavior
 - Schematic capture
 - User graphically enters a desired logic circuit
 - Hardware description languages
 - User enters a programming language-like description of a desired logic circuit



Design entry with truth tables

- Commonly use a ***waveform editor*** to enter a timing diagram that describes a desired functionality for a logic circuit
 - CAD system transforms this into equivalent logic gates
 - Not appropriate for large circuits, but can be used for a small logic function that is to be part of a larger circuit



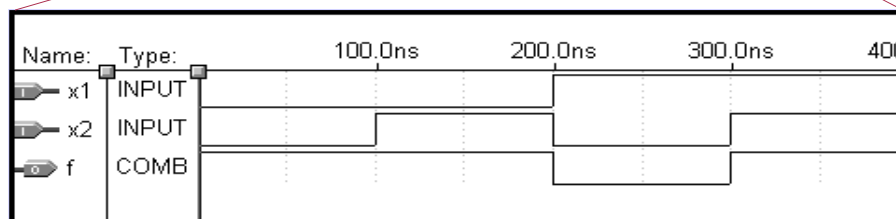
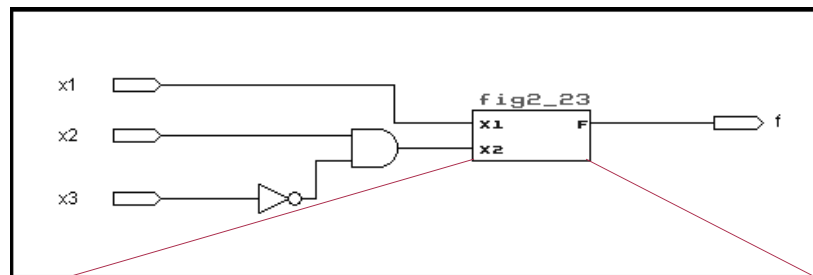


Schematic capture

- A common type of CAD tool
- **Schematic**: refers to a diagram of a circuit in which circuit elements (logic gates) are shown as graphical symbols and connections between them are drawn as lines
- Tool provides a collection of symbols that represent gates of various types with different inputs and outputs. A **library**.
- Previously designed circuits can be represented with a graphical symbol and used in larger circuits. Known as **hierarchical design** and provides a way of dealing with complexities of large circuits



Schematic capture





Hardware description languages

- A hardware description language (HDL) is similar to a computer program except that it is used to describe hardware
- Common HDLs
 - VHDL (VHSIC Hardware Description Language)
 - Verilog
 - Many others (vendor specific)
- VHDL and Verilog are standards
 - Offer portability across different CAD tools and different types of programmable chips



Synthesis

- Synthesis CAD tools perform the process of generating a logic circuit from some stated functional behavior
- **Translating (compiling)** VHDL code into a network of logic gates is a part of synthesis
- Not only will the CAD tool produce a logic circuit, but it can also optimize that circuit
 - In terms of speed and/or size (*logic optimization*)
 - Called **logic synthesis** or **logic optimization**
- Finally, **technology mapping** and **layout synthesis** (physical design) complete the synthesis process



Simulation

- Once designed, it is necessary to verify that the design circuit functions as expected
- In a **functional simulation** the user specifies valuations of the circuits inputs and the CAD tool generates the outputs (commonly in the form of a timing diagram)
 - User verifies generated outputs against expected outputs
- Functional simulators assume the time needed for signals to propagate through the logic gates is negligible
 - For a real implementation this is not sufficient
 - Use a **timing simulator** to obtain accurate (complete) simulation



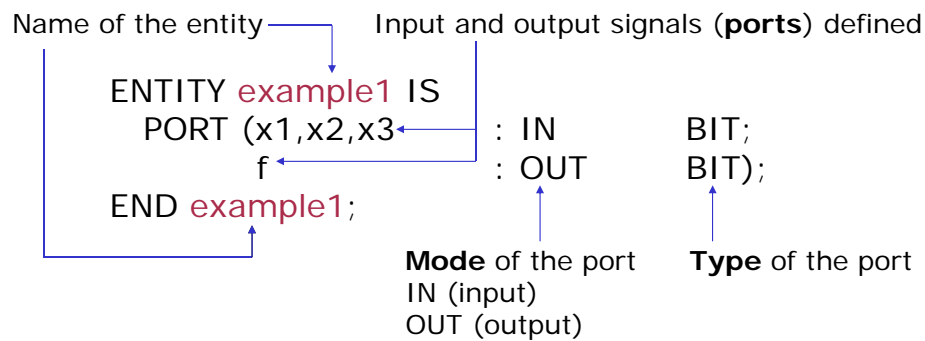
Introduction to VHDL

- Designer writes a logic circuit description in VHDL source code
- VHDL compiler translates this code into a logic circuit
- Representation of digital signals in VHDL
 - Logic signals in VHDL are represented as a data object
 - VHDL includes a data type called **BIT**
 - BIT objects can assume only two values: 0 and 1



Writing simple VHDL code

- First step in writing VHDL code is to declare the input and output signals
- Done using a construct called an **entity**

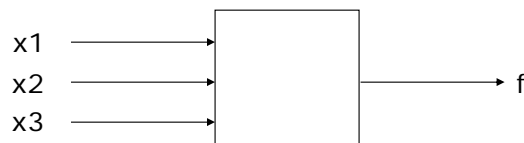
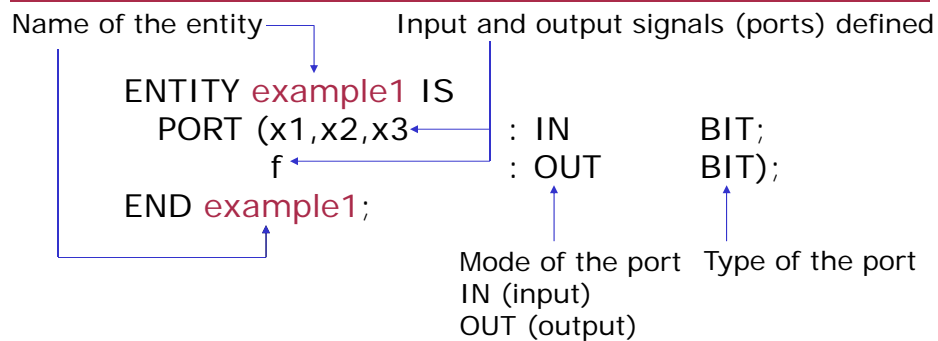


Electrical & Computer Engineering

Dr. D. J. Jackson Lecture 6-11



Writing simple VHDL code



Electrical & Computer Engineering

Dr. D. J. Jackson Lecture 6-12



Writing simple VHDL code

- The entity specifies the inputs and outputs for a circuit, but does not describe the circuit function
- Circuit functionality is specified using a VHDL construct called an **architecture**

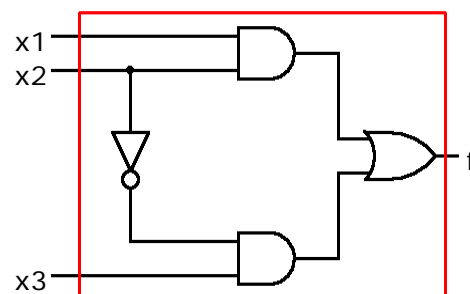
Architecture name → Entity used by LogicFunc

```
ARCHITECTURE LogicFunc OF example1 IS
BEGIN
    f <= (x1 AND x2) OR (NOT x2 AND x3);
END LogicFunc;
```

↑
VHDL statement that describes the circuit functionality



Complete VHDL code example



```
ENTITY example1 IS
    PORT (x1,x2,x3 : IN BIT;
          f       : OUT BIT);
END example1;

ARCHITECTURE LogicFunc OF example1 IS
BEGIN
    f <= (x1 AND x2) OR (NOT x2 AND x3);
END LogicFunc;
```



Boolean operators in VHDL

- VHDL has built-in support for the following operators
 - AND logical AND
 - OR logical OR
 - NOT logical NOT
 - NAND, NOR, XOR, XNOR (covered later)
- Assignment operator `<=`
 - A variable (usually an output) should be assigned the result of the logic expression on the right hand side of the operator
- VHDL does not assume any precedence of logic operators. Use parentheses in expressions to determine precedence
- In VHDL, a logic expression is called a ***simple assignment statement***. There are other types that will be introduced that are useful for more complex circuits.



Example VHDL code

- Write the VHDL code (entity and architecture constructs) for the adder circuit
 - Name the entity ***Add*** and name the architecture ***AddFunc***
- Write the VHDL code for the majority circuit
 - Name the entity ***Majority*** and name the architecture ***MajorityFunc***