Digital Design using VHDL

Andy Tyrrell
FPGAs – all about

Space  Vs  Time
What is VHDL?

VHSIC
Hardware
Description
Language

VHSIC – Very High Speed Integrated Circuit
- Developed in the early 1980’s
  - Method of describing electronic systems for the American Department of Defense
- Use for digital design only
- Synthesis tools used to create and optimise the implementation
Electronic Design Process

System Specification

HW/SW partition

Hardware Spec

ASIC
PLD/FPGA
Std Parts

Boards and Systems

Software Spec

Software
In general VHDL is used with synthesis tools for ASIC and PLD / FPGA design.
Limitations of VHDL

- Digital only, not analogue...yet
- Dependent upon synthesis tools
  - Slightly different syntax
  - Limited control on physical hardware layout
Each level of abstraction defines how much detail about the design is specified in its description.
**Levels of Abstraction in Digital Design**

- **Layout** is the lowest level of abstraction
- Specifies the actual layout of the design on silicon, and may also specify detailed timing information or more “analogue” effects
Levels of Abstraction in Digital Design

- Logic level deals with the interconnection of logic gates and registers
- Layout information and “analogue” effects are ignored
- Deals with function, architecture, and technology
Levels of Abstraction in Digital Design

- Register Transfer Language – VHDL used to define every register in the design
- Still contains architectural information
Levels of Abstraction in Digital Design

- Behavioural level uses VHDL to describe function of a design, without specifying the architecture of registers.
- Contains as much timing information as a designer requires to represent the function.
Behavioral vs. RTL (Structural)

- Behavioural Code – describes the functionality and behaviour of the function
- Structural Code - describes the actual gate and register level of the function
Typical Synthesis Design Flow
Concurrency – VHDL is able to describe activities that happen in parallel

Hierarchy – the ability to describe the structure. Also, the ability to mix descriptions of structure with descriptions of behaviour

Sequential – VHDL has statements that execute one after the other in sequence, such as a traditional software language like ‘C’

Time – VHDL allows time to be modeled
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY dff IS
PORT (d, clk, rst: IN STD_LOGIC;
       q: OUT STD_LOGIC);
END dff;

ARCHITECTURE behaviour OF dff IS
BEGIN
    PROCESS (rst, clk)
    BEGIN
        IF (rst = '1') THEN
            q <= '0';
        ELSIF (clk'EVENT AND clk='1') THEN
            q <= d;
        END IF;
    END PROCESS;
END behaviour;
LIBRARY ieee; -- library declaration
USE ieee.std_logic_1164.all; -- std and work visible by default

ENTITY dff IS
  PORT (d, clk, rst: IN STD_LOGIC; -- IN STD_LOGIC
        q: OUT STD_LOGIC); -- OUT STD_LOGIC
END dff;

ARCHITECTURE behaviour OF dff IS
BEGIN
  PROCESS (rst, clk) -- executed every time a signal declared
  BEGIN -- in its list changes. Here rst & clk
    IF (rst = '1') THEN -- rst goes to '1'
      q <= '0'; -- output reset to '0'
    ELSIF (clk'EVENT AND clk='1') THEN -- no rst, and
      q <= d; -- clock change to 1 (rising edge ↑)
    END IF;
    -- output = d
  END PROCESS;
END behaviour;

VHDL Example - see again later
Basic VHDL Structure

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY entity_name IS
    PORT (......);
END entity_name;

ARCHITECTURE arch_name OF entity_name IS
    [declarations]
BEGIN
    code ......
END arch_name;
LIBRARY ieee;
-- library declaration
USE ieee.std_logic_1164.all; -- std and work visible by default

ENTITY dff IS
PORT (d, clk, rst: IN STD_LOGIC; -- IN STD_LOGIC
q: OUT STD_LOGIC);
-- OUT STD_LOGIC

END dff;

ARCHITECTURE behaviour OF dff IS
BEGIN
PROCESS (rst, clk) -- executed every time a signal declared
BEGIN
-- in its list changes. Here rst & clk
IF (rst = '1') THEN
-- rst goes to '1'
qu <= '0';
-- output reset to '0'
ELSIF (clk'EVENT AND clk='1') THEN -- no rst, and
q <= d;
-- clock change to 1 (rising edge)
END IF;
-- output = d
END PROCESS;

END behaviour;
**Libraries**

- Contains a package or a collection of packages
- Resource Libraries
  - Standard Package
  - IEEE developed packages
  - Xilinx (FPGA) Component packages
  - Any library of design units that are referenced in a design
- Working Library
  - Library into which the unit is being compiled
Model Referencing of Libraries/Packages

- All packages must be compiled
- Implicit Libraries
  - Work
  - STD
  Note: Items in these packages do not need to be referenced, they are implied
- LIBRARY Clause
  - Defines the library name that can be reference
  - It’s a symbolic name to path/directory
  - Defined by the compiler
- USE Clause
  - Specifies the package and object in the library that you have specified in the library clause
LIBRARY std;

- Contains the following packages
  - Standard (Types: Bit, Boolean, Integer, Real, and Time - and operator functions to support these types)
  - Textio (File operations)
- An implicit library (built-in)
  - Does not need to be referenced
LIBRARY IEEE;

Contains the following packages:

- std_logic_1164 (std_logic types & related functions)
- std_logic_arith (arithmetic functions)
- std_logic_signed (signed arithmetic functions)
- std_logic_unsigned (unsigned arithmetic functions)
LIBRARY <name>;
USE <name>.<package_name>.all;
LIBRARY ieee;
USE ieee.std_logic_1164.all;
----------------
ENTITY tri-state IS
  PORT (ena :IN STD_LOGIC;
         input :IN STD_LOGIC_VECTOR (7 DOWNTO 0);
         output :OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
END tri_state;
---------------
ARCHITECTURE tri_state OF tri_state IS
BEGIN
  output <= input WHEN (ena = '0') ELSE (OTHERS => 'Z');
END tri_state;
LIBRARY ieee; -- semi-colon indicates the end of a statement
USE ieee.std_logic_1164.all; -- ieee multi-level logic system

LIBRARY std; -- Resource library (data types, text I/O etc)
USE std.standard.all; -- for VHDL design environment

LIBRARY work; -- library is where we save our design
USE work.all; -- (.vhd file & compiler & simulator etc files)

-- NOTE: std and work visible by default.
End of lecture 1

History of VHDL
Background to the design process and level of abstraction
Libraries
Simple examples of VHDL descriptions