More RTOS/VHDL Intro

John C. Shovic

Design and Simulate

- Design environment/design flow
  - Circuit Definition → Verification (simulation)
  - Synthesis → Simulation
    - Place and Route → Verification
      Simulation (with back annotated delays)
- Implementation (FPGAs)
  - Design Review
  - Submit

Basic Concepts

- Entity/Architecture
  - Entity vs. schematic symbol
  - Port statement/syntax
    (STD_LOGIC_VECTOR/UNSIGNED)
  - In, out, inout
  - Entity name as design unit
  - Architecture vs. simulation model in schematic capture
  - Architecture syntax
  - Architecture/entity binding (configuration)
  - Concurrent area
More Basic Concepts

• Signals/Signal assignments
  • Y <= '1'; Y <= A; etc.
  • "Standard" functions
    (and, or, nand, nor, xor, xnor, not, =, <, >, /=)
  • Y <= A and B;
  • VHDL time queue

More Basic Concepts

• Concurrent vs. Sequential operation
  – Signals model wires -- concurrent
  – Possible to check explicit order of occurrences; must have sequential statements (if - then).
  – Sequential statements allow definition of memory (more later)

• Structural vs. Behavioral modeling

VHDL for Combinational Circuits

• Signal assignment statements
  – Conditional
  – Selected
  – Examples
    • Mux, decoder, ssd, truth table
  – Arithmetic/data circuits (adders, comparators, shifters, multipliers)
### A Mux

Architecture mux1 of mux is

```vhdl
begin
    y <= A when (Sel = "00") else
         B when (Sel = "01") else
         C when (Sel = "10") else
         D when (Sel = "11")
end Mux2;
```

---

### 8 Bit Adder

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity adder_8 is
    port (a,b: in UNSIGNED (7 downto 0);
         cin: in STD_LOGIC;
         sum: out STD_LOGIC_VECTOR (7 downto 0));
end adder_8;

architecture behav of adder_8 is
begin
    sum <= a + b + cin;
end behav;
```

---

### Libraries and Packages

- **Library contents**
  - Pre-analyzed entities and architectures, configurations, packages
    - IEEE "standard" libraries
    - Packages
      - std_logic, std_logic_arith
    - Visibility (library and use statements)
Sequential Stuff

- VHDL for Sequential Circuits
- DFFs
- State machines
- Digital systems

DFF Register

library ieee;
use ieee.std_logic_1164.all;

entity DFFREG is
    port (D : in STD_LOGIC_VECTOR(7 downto 0);
          clk, rst : in STD_LOGIC;
          Q : out STD_LOGIC_VECTOR(7 downto 0));
end DFFREG;

architecture dffreg_arch of DFFREG is
begin
    process(clk, rst)
    begin
        if rst='1' then Q <= "00000000";
        elsif (CLK'event and CLK='1') then Q <= D;
    end if;
end process;
end dffreg_arch;

Finite State Machine

- Look at Handout

architecture example of FSM_EX is
    signal sreg : STD_LOGIC_VECTOR(1 downto 0);
begin
    process (CLK, rst) begin
        if RST='1' then
            sreg <= "00";
        elsif (CLK'event and CLK = '1') then
            case sreg is
            when "00" =>
                if X='1' then
                    sreg <= "01";
                elsif X='0' then
                    sreg <= "00";
                end if;
            end case;
        end if;
    end process;
end example;
Homework Assignment

• Assignment #1 -
  Due on Tuesday,
  September 26,
  2000

• Read Chapter 4
  and 5 in VHDL