Concurrent VHDL

Announcements

- Homework #2 due Wednesday
- Reading Assignment
  - Ch. 3 section 6, Ch. 5 sections 1-4
- Free homework grade for critical thinking lecture
  - Don’t forget to scan in

RIT | Division of Academic Affairs
Eugene H. Fram Chair in Applied Critical Thinking

2019 Fram Signature Lecture

"POWERFUL STUFF: An Entrepreneurial Mindset Built Upon Critical Thinking" with Doug Melton of KEEN

Date: Tuesday, September 17, 2019
Time: 3:30 pm – 4:45 pm
Place: Ingle Auditorium
(Reception immediately following in Fireside Lounge)

Access Services will be providing interpreters

Proudly cosponsored by:

RIT | College of Engineering Technology

RIT | Kate Gleason College of Engineering

A Note about Libraries

- We will learn more about libraries later
- For now we will use the following Libraries
  - library IEEE;
  - use IEEE.STD_LOGIC_1164.ALL;
  - use IEEE.NUMERIC_STD.ALL;
  - use IEEE.STD_LOGIC_UNSIGNED.ALL
- These libraries permit use of predefined logic values, logic operations like AND, OR, and arithmetic operations like + (add) etc.

Things to know

- Comments
  - --this is a comment
  - Use lots of comments – good habit to get into
  - Use descriptive signal and port names
- Indenting
  - Use indenting to make code more ‘readable’
  - SPACES (2, 3 or 4) not TABS
- Editor
  - I recommend Notepad++
    - Free online
    - Recognizes VHDL and formats accordingly
    - Color codes comments, reserved words, etc…

Port and Signal Types

- STD_LOGIC
  - **Represents a single wire**
  - Can have 9 values
    - 'U‘ : uninitialized. This signal hasn't been set yet.
    - 'X‘ : unknown. Impossible to determine this value/result.
    - '0‘ : logic 0
    - '1‘ : logic 1
    - 'Z‘ : High Impedance
    - 'W‘ : Weak signal, can't tell if it should be 0 or 1.
    - 'L‘ : Weak signal that should probably go to 0
    - 'H‘ : Weak signal that should probably go to 1
    - '-‘ : Don't care
  - Useful in simulation
  - We will use STD_LOGIC in DSD

Port and Signal Types

- Bit
  - Also represents a single wire
  - Can only have a value of 0 or 1
  - Not as useful as STD_LOGIC
    - In the real world there are many more possible values on a wire than 0 and 1

Port and Signal Types (con’t)

- STD_LOGIC_VECTOR
  - **Represents a bus or bundle of wires**
  - Must include a range
    - Ex:
      - port (a : STD_LOGIC_VECTOR(7 DOWNTO 0);
      - This is an 8 bit bus
    - It could also be specified with STD_LOGIC_VECTOR(0 TO 7)
    - ‘DOWNTO’ is more common and preferred in DSD

Port and Signal Types (con't)

- STD_LOGIC vs. STD_LOGIC_VECTOR
  - When assigning a value to STD_LOGIC, use single quotes
  - Ex
    - Var1 <= '0'; tri_state <= 'Z';
  - When assigning a value to STD_LOGIC_VECTOR use double quotes
  - Ex
    - a <= "11110000"
    - The value on the right must contain the same number of elements as the width of the vector

Selected Signal Assignment

- Describing schematic with gates and wires is not very efficient
- Still have to design schematic first
- In a selected signal assignment you work from the truth table, not the schematic

X	Y	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Selected Signal Assignment

```
ENTITY fulladd IS
    PORT( Cin, X,Y  :  IN STD_LOGIC;
                S, Cout     :  OUT STD_LOGIC);
    END fulladd;

ARCHITECTURE behavioral OF fulladd IS
    SIGNAL inputs : STD_LOGIC_VECTOR(2 DOWNTO 0);
    BEGIN
              inputs <= X & Y & Cin;
              WITH inputs SELECT
                          S <= '1' WHEN "001"| "010" | "100" | "111",
                                 '0' WHEN OTHERS;
              WITH inputs SELECT
                          Cout <= '1' WHEN "011" | "101" | "110" | "111",
                                       '0' WHEN OTHERS;
   END behavioral;
```

Selected Signal Assignment

```
ARCHITECTURE behavioral OF fulladd IS
    SIGNAL inputs : STD_LOGIC_VECTOR(2 DOWNTO 0);
    BEGIN
            inputs <= X & Y & Cin;
            WITH inputs SELECT
                    S <= '1' WHEN "001"| "010" | "100" | "111",
                        '0' WHEN OTHERS;
            WITH inputs SELECT
                    Cout <= '1' WHEN "011" | "101" | "110" | "111",
                            '0' WHEN OTHERS;
   END behavioral;
```

Concatenation: takes three separate wires and combines them into 3-bit Bus. **Order is important.** (X, Y, Cin)

Selected Signal Assignment (con't)

```
WITH expression SELECT
  signal_name <= signal_value WHEN choices,
                 signal_value WHEN choices,
                  :
                  :
                  signal_value WHEN OTHERS;
```

Conditional Signal Assignment

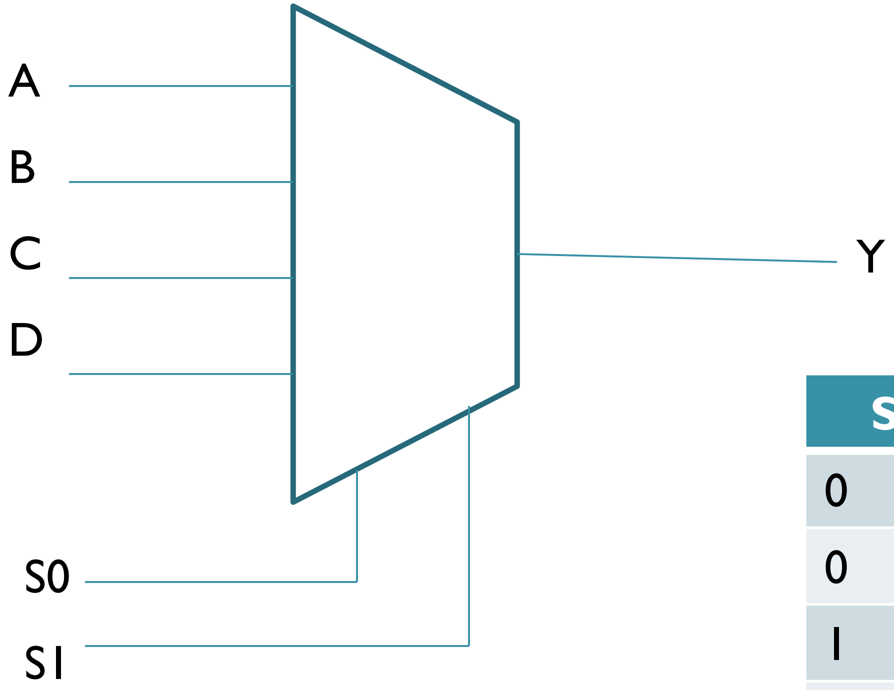
```
Target <= value_expression1 WHEN condition1 ELSE
          value_expression2 WHEN condition2 ELSE
          value_expression3 WHEN condition3 ELSE
          value_expression;
```

- Condition must evaluate to true or false

```
ARCHITECTURE behavioral OF fulladd IS
  signal inputs : STD_LOGIC_VECTOR(2 DOWNTO 0);
  BEGIN

      inputs <= X & Y & Cin;
      S <= '1' WHEN inputs = "001" or inputs = "010" or inputs = "100" or inputs = "111" else '0';

      Cout <= '1' WHEN inputs = "011" or inputs = "101" or inputs = "110" or inputs = "111" else '0';

  END behavioral;
```

Example: Mux

- Consider the following Mux



S1	S0	Y
0	0	A
0	1	B
1	0	C
1	1	D

- Write the architecture
  - Using selected signal assignment
  - Using conditional signal assignment

Concurrent Statements

- Concurrent Statements include
  - Simple assignment statements
    - a <= ‘1’;
    - b <= “11001100”;
    - c <= not (a AND d);
  - Selected assignment statements
  - Conditional assignment statements
- **Describe COMBINATORIAL logic**
- **They all evaluate at the same time**