Lab 05: Design A

- Dataflow style VHDL implementation of a four-bit ripple-carry adder
- VHDL code already provided
- You need to perform the following
  - Create a project and compile
  - Create test vectors and perform function simulation
  - Download the code to DE1 board
  - Test the DE1 board with provided test vectors

Lab 05: Design B

- Structural VHDL implementation of a four-bit ripple-carry adder
  - Component: Full adder (one bits), which can be written in dataflow style
  - Instantiate the full adder four times and wire them (four bits): See VHDL code in next slides
- Sample VHDL codes already provided in slides
  - Shown in next two slides
- You need to perform the following
  - Create a project, create VHDL codes, and compile
  - Create test vectors and perform function simulation
  - Download the code to DE1 board
  - Test the DE1 board with provided test vectors

Full adder in VHDL

- New data objective type: STD_LOGIC
- New package declaration: ieee.std_logic_1164.all
- New logic operator: XOR

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY fulladd IS
        PORT ( Cin, x, y : IN STD_LOGIC ;
                        s, Cout : OUT STD_LOGIC ) ;
END fulladd ;

ARCHITECTURE LogicFunc OF fulladd IS
BEGIN
        s <= x XOR y XOR Cin ;
        Cout <= (x AND y) OR (Cin AND x) OR (Cin AND y) ;
END LogicFunc ;
```

Ripple-carry adder in VHDL

- Structural style
- Declaration of Component
- Instantiation statement
- Named or positional association

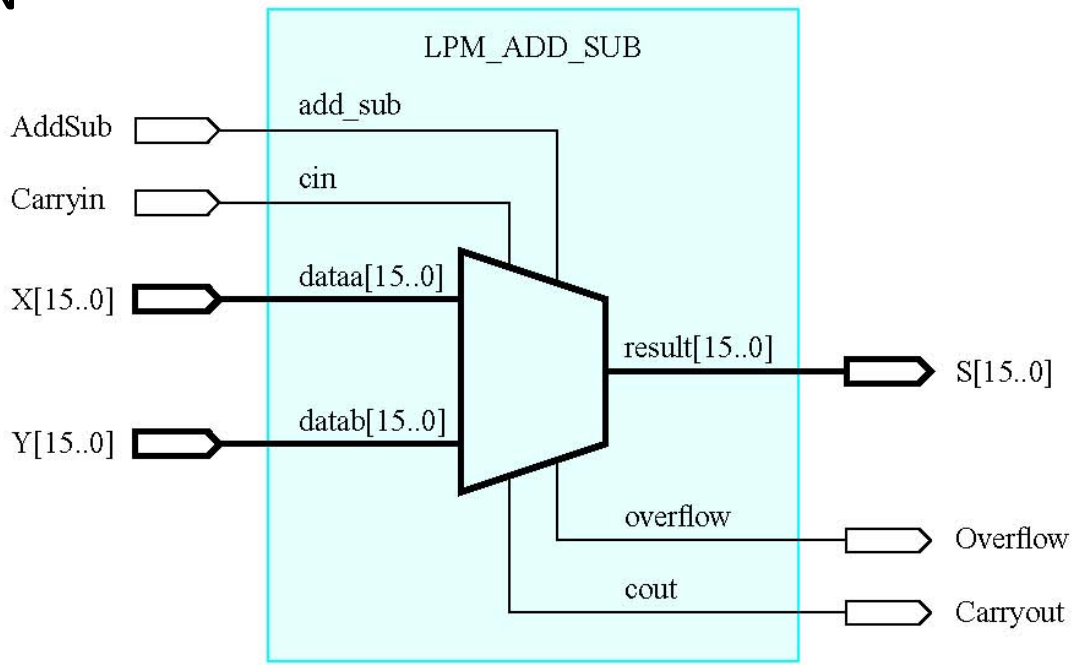
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY adder4 IS
        PORT ( Cin : IN STD LOGIC ;
                        x3, x2, x1, x0        : IN STD_LOGIC ;
                        y3, y2, y1, y0        : IN STD_LOGIC ;
                        s3, s2, s1, s0        : OUT STD_OGIC ;
                        Cout                  : OUT STD_LOGIC ) ;
END adder4 ;

ARCHITECTURE Structure OF adder4 IS
        SIGNAL c1, c2, c3 : STD_LOGIC ;
        COMPONENT fulladd
                PORT ( Cin, x, y        : IN STD_LOGIC ;
                                s, Cout       : OUT STD_LOGIC ) ;
        END COMPONENT ;
BEGIN
        stage0: fulladd PORT MAP ( Cin, x0, y0, s0, c1 ) ;
        stage1: fulladd PORT MAP ( c1, x1, y1, s1, c2 ) ;
        stage2: fulladd PORT MAP ( c2, x2, y2, s2, c3 ) ;
        stage3: fulladd PORT MAP (
                Cin => c3, Cout => Cout, x => x3, y => y3, s => s3 ) ;
END Structure ;
```

Lab 05: Design C

- Schematic design of a four-bit ripple-carry adder through use of LPM module: LPM_ADD_SUB
- You need to perform the following
  - Create a project, create your schematic, and compile
  - Create test vectors and perform function simulation
  - Download the code to DE1 board
  - Test the DE1 board with provided test vectors

LPM_ADD_SUB

- Library parameterized module (LPM)
  - LPM_WIDTH
  - LPM_REPRESENTATION



LPM_ADD_SUB

- Addition/subtraction: add_sub (0-sub, 1-add)
- Adder: $cin = 0, 1$
- Subtractor: $cin = 1$ or leave *cin* open