Electrical and Computer Engineering

Unsigned Adder

By Prawat Nagvajara

Synopsis

Unsigned adder design using hardware description language with the IEEE std_logic_unsigned and std_logic_arith package '+' operator. The design extends the operand internally to n+1 bits (n-bit oerands). The most significant bit indicates the unsigned overflow condition.

Code

```
--------------------------------
-- Unsigned adder with overflow flag
-- Use 4-bit operands for the sake of
-- example.
--------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_arith.all,
ieee.std_logic_unsigned.all;

entity add_4bit is
port(
x, y: in  std_logic_vector(3 downto 0);
   z: out std_logic_vector(3 downto 0);
 ovf: out std_loGic
 );
end add_4bit;

architecture Behavioral of add_4bit is
-- declare wires with one additional bit
signal w1,w2,w3: std_logic_vector(4 downto 0);

begin

-- wiring (concurrent statements)
w1 <= '0'&x; -- concatenate '0' to the
w2 <= '0'&y; -- Most Significant Bit (MSB)
z <= w3(3 downto 0);--sum is the lower 4 bits
ovf <= w3(4); -- MSB is the overflow flag
-- For the sake of example, code addition
-- as a process sensitive to w1 and w2
process(w1, w2)
begin
w3 <= w1 + w2;
end process;

end Behavioral;
```

Exploration

Design and implement a signed adder with overflow flag. Add hardware to determine the overflow condition, that is, when the operands have the same sign and the sum is a different sign. The design will not require internal extension of the operand data width.