Electrical and Computer Engineering

Radix 10 Adder

By Prawat Nagvajara

Synopsis

This note covers a design of radix 10 adder using arithmetical operator '+' on vectors. The design suggests a straightforward generalization to include

generic (r : natural;
n: natural);

where r is the radix and n is the least whole number greater than or equal to the logarithm base 2 of r.

Code

The VHDL code below uses addition, greater than or equal and subtraction of vectors and numbers. The radix r is equal 10 and n = 4 in this design. The inputs x and y are extended to n+1. The design assigns the lower n-bit of the internal variable to the output ports.

```
----------------------------------
-- radix 10 2-digit adder with
-- carry in and carry out
-- To generalize the design include
-- generic (r : natural;
--          n : natural);
-- where r is the radix r and
-- n is the ceiling of log base 2 of r
--------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity radix_10_add is
port(
x, y : in std_logic_vector(3 downto 0);
 c_in: in std_logic;
   z : out std_logic_vector(3 downto 0);
c_out: out std_logic
);
end radix_10_add;

architecture beh of radix_10_add is
  begin

process(x, y, c_in)
variable temp_x, temp_y, temp_z :
std_logic_vector(4 downto 0);
begin
  temp_x := '0'&x;
  temp_y := '0'&y;
  temp_z := unsigned(temp_x) +
            unsigned(temp_y);
  if c_in = '1' then
    temp_z := unsigned(temp_z) + 1;
  end if;
  if temp_z >= 10 then
    temp_z := unsigned(temp_z) - 10;
    c_out <= '1';
  else c_out <= '0';
  end if;
  z <= temp_z(3 downto 0);
end process;
end beh;
```

Simulation

Figure 1 show a simulation of the code (modelsim) where vector x, y and z are displayed as radix unsigned (right click on the signals select Radix and select unsigned).

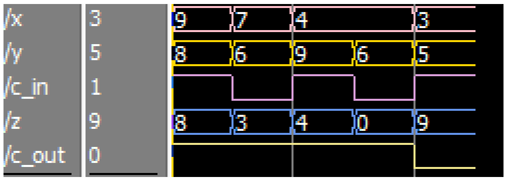

Fig. 1 Simulation Wave