Function (3A)

Resolution Function

Young Won Lim
07/08/2012

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Resolved Signals (1)

```
entity mux is
  port( i1, i2, a : in fourval; q : out fourval );
end mux2;


architecture arch of mux is



  signal nota : fourval;


begin
  u1 : inv    port map (a, nota);
  u2 : and2 port map (i1, a, intq);
  u3 : and2 port map (i2, nota, intq);
  q <= intq;
end arch
```

```
component and
   port( a, b : in fourval; c : out fourval);
end component;
compoent inv
   port( a : in fourval; b : out fourval);
end component;
```

```
  signal intq : resolve fourval := x;

  signal intq :  resfour := x;
```

Resolution Function Declaration

```
package fourpack is
  type fourval is (X, L, H, Z);
  type fourvalvector is array(natural range <>) of fourval;
  function resolve( s: fourvalvector ) return fourval;
  subtype resfour is resolve fourval;
end fourpack;
```

Resolution Function Definition (1)

```
package body fourpack is
 function resolve( s: fourvalvector) return fourval is
  variable result : fourval :=Z;
 begin
  for i in s'range loop
   case result is
      when Z =>   ●   ●   ●
      when L =>   ●   ●   ●
      when H =>   ●   ●   ●
      when X =>   ●   ●   ●
   end case;
  end loop;
  return result;
 end resolve;
end fourpack;
```

Resolution Function Definition (2)

```
when Z =>
  case s(i) is
    when H => result := H;
    when L => result := L;
    when X => result := X;
    when others => null;
  end case;
```

```
when L =>
  case s(i) is
    when H => result := X;
    when X => result := X;
    when others => null;
  end case;
```

```
when H =>
  case s(i) is
    when L => result := X;
    when X => result := X;
    when others => null;
  end case;
```

```
when X =>
   result := X;
  end case;
```

	Z	L	H	X
Z	Z	L	H	X
L	L	L	X	X
H	H	X	H	X
X	X	X	X	X

Example (1)

```
package body mvl4_pkg is



  function tristate_rf( v: logic4_vector) return logic4  is
    variable result : logic4 :='Z';
  begin
    for i in v'range loop
       result := tristate_rf_table(result, v(i));
       exit when result = 'X';
    end loop;
    return result;
  end resolve;


end mvl4_pkg;
```

```
type logic4_table is
       array (logic4, logic4) of logic4;

Constant tristate_rf_table
       : logic4_table :=   (('X', 'X', 'X', 'X'),
                            ('X', '0', 'X', '0'),
                            ('X', 'X', '1', '1'),
                            ('X', '0', '1', 'Z') );
```

Example (2)

```
package body mvl4_pkg is



  function wireand_rf( v: logic4_vector) return logic4  is
    variable result : logic4 :='Z';
  begin
    for i in v'range loop
      result := wireand_rf_table(result, v(i));
      exit when result = 'X';
    end loop;
    return result;
  end resolve;


end mvl4_pkg;
```

```
type logic4_table is
       array (logic4, logic4) of logic4;

Constant wireand_rf_table
       : logic4_table :=   (('X', 'X', 'X', 'X'),
                            ('X', '0', 'X', '0'),
                            ('X', 'X', '1', '1'),
                            ('X', '0', '1', 'Z') );
```

References

[1] http://en.wikipedia.org/
[2] J. V. Spiegel, VHDL Tutorial,
http://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html
[3] J. R. Armstrong, F. G. Gray, Structured Logic Design with VHDL
[4] Z. Navabi, VHDL Analysis and Modeling of Digital Systems
[5] D. Smith, HDL Chip Design
[6] http://www.csee.umbc.edu/portal/help/VHDL/stdpkg.html
[7] VHDL Tutorial - VHDL onlinewww.vhdl-online.de/tutorial/
[8] compgroups.net
[9] J. Pick, VHDL Techniques, Experiments, and Caveats
[10] D. L. Perry, VHDL 2nd Ed