

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_arith.all;
4 use ieee.std_logic_unsigned.all;
5 -----
6 entity mux_tree is
7     generic (width : integer := 5);
8     port
9     (
10        din  : in  std_logic_vector(((2**width) -1) downto 0);
11        sel  : in  std_logic_vector((width-1) downto 0);
12        dout : out std_logic
13    );
14 end entity mux_tree;
15
16 architecture arc_mux_tree of mux_tree is
17 component mux_tree is
18     generic (width : integer := 5);
19     port
20     (
21        din  : in  std_logic_vector(((2**width) -1) downto 0);
22        sel  : in  std_logic_vector((width-1) downto 0);
23        dout : out std_logic
24    );
25 end component mux_tree;
26 begin
27     mux_2x1: if (sel'length=1) generate
28         dout<=din(conv_integer(sel));
29     end generate mux_2x1;
30
31     mux_tree_gen: if (sel'length/=1) generate
32         signal temp : std_logic_vector(1 downto 0);
33     begin
34         left_subtree: mux_tree
35             generic map (width => (sel'length - 1))
36             port map
37             (
38                 din => din(din'high downto (din'length/2)),
39                 sel => sel((sel'high -1) downto sel'low) ,
40                 dout => temp(temp'high)
41             );
42
43         right_subtree: mux_tree
44             generic map (width => (sel'length - 1))
45             port map
46             (
47                 din => din(((din'length/2) - 1) downto din'low),
48                 sel => sel((sel'high -1) downto sel'low) ,
49                 dout => temp(temp'low)
50             );
51
52         dout<=temp(conv_integer(sel(sel'high)));
53     end generate mux_tree_gen;
54 end architecture arc_mux_tree;

```