```
--------------------------------
-- Name:     Roger Grayson
-- Date:     9-10-02
-- Purpose:  Generic Nx4x1 mux
--
--------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

entity muxNx4x1 is
    generic(N:integer:=32);
    port(y0,y1,y2,y3:in std_logic_vector(N-1 downto 0);
         sel         :in std_logic_vector(1 downto 0);
         f           :out std_logic_vector(N-1 downto 0));
end muxNx4x1;

architecture process_arch of muxNx4x1 is
begin
    process(y0,y1,y2,y3,sel)
    begin
        case sel is
            when "00" => f <= y0;
            when "01" => f <= y1;
            when "10" => f <= y2;
            when others => f <= y3;
        end case;
    end process;
end process_arch;
--------------------------------
-- Name:     Roger Grayson
-- Date:     9-10-02
-- Purpose:  Generic Nx2x1 mux
--------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

entity muxNx2x1 is
    generic(N:integer:=32);
    port(y0,y1:in std_logic_vector(N-1 downto 0);
         sel         :in std_logic;
         f           :out std_logic_vector(N-1 downto 0));
end muxNx2x1;

architecture process_arch of muxNx2x1 is
begin
    process(y0,y1,sel)
    begin
        case sel is
            when '0' => f <= y0;
            when others => f <= y1;
        end case;
    end process;
```

```
end process_arch;

----------------------------------------
-- Name: Roger Grayson
-- Date: Feb 6, 2003
-- Pupr: 32x32x1 multiplexor for register file
----------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
entity mux is
    port(   y0,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21,
y22,y23,y24,y25,y26,y27,y28,y29,y30,y31:in std_logic_vector(31 downto 0);
            sel     :in std_logic_vector(4 downto 0);
            f       :out std_logic_vector(31 downto 0));
end entity mux;

architecture process_arch of mux is
begin
    process(y0,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21,
y22,y23,y24,y25,y26,y27,y28,y29,y30,y31,sel)
    begin
        case sel is
            when "00000" => f <= y0;
            when "00001" => f <= y1;
            when "00010" => f <= y2;
            when "00011" => f <= y3;
            when "00100" => f <= y4;
            when "00101" => f <= y5;
            when "00110" => f <= y6;
            when "00111" => f <= y7;
            when "01000" => f <= y8;
            when "01001" => f <= y9;
            when "01010" => f <= y10;
            when "01011" => f <= y11;
            when "01100" => f <= y12;
            when "01101" => f <= y13;
            when "01110" => f <= y14;
            when "01111" => f <= y15;
            when "10000" => f <= y16;
            when "10001" => f <= y17;
            when "10010" => f <= y18;
            when "10011" => f <= y19;
            when "10100" => f <= y20;
            when "10101" => f <= y21;
            when "10110" => f <= y22;
            when "10111" => f <= y23;
            when "11000" => f <= y24;
            when "11001" => f <= y25;
            when "11010" => f <= y26;
            when "11011" => f <= y27;
            when "11100" => f <= y28;
            when "11101" => f <= y29;
```

```
            when "11110" => f <= y30;
            when others  => f <= y31;
        end case;
    end process;
end process_arch;
```