

```
-----  
--  
-- Purpose:  
--  
--   Barrel Shifter  
--  
-- Discussion:  
--  
--  
-- Licensing:  
--  
--   This code is distributed under the GNU LGPL license.  
--  
-- Modified:  
--  
--   2012.03.23  
--  
-- Author:  
--  
--   Young W. Lim  
--  
-- Parameters:  
--  
--   Input:  
--  
--   Output:  
-----
```

```
library STD;  
use STD.textio.all;
```

```
library IEEE;  
use IEEE.std_logic_1164.all;  
use IEEE.numeric_std.all;
```

```
entity bshift is
```

```
  port (  
    din      : in  std_logic_vector (31 downto 0) := X"0000_0000";  
    nbit     : in  std_logic_vector (4  downto 0) := X"00";  
    dout     : out std_logic_vector (31 downto 0) := X"0000_0000");
```

```
end bshift;
```

```
architecture rtl_bshift of bshift is
```

```
begin
```

```
  bshft: process (din, nbit)  
    variable dinX : std_logic_vector (63 downto 0);  
  begin -- process bshft
```

```
    s := din(31);  
    for i in 63 downto 32 loop  
      dinX(i) := s;  
    end loop; -- i  
    dinX := dinX(63 downto 32) & din (31 downto 0);
```

```
    case nbit is  
      when "00000" => result := dinX(31 downto 0);  
      when "00001" => result := dinX(32 downto 1);  
      when "00010" => result := dinX(33 downto 2);  
      when "00011" => result := dinX(34 downto 3);  
      when "00100" => result := dinX(35 downto 4);  
      when "00101" => result := dinX(36 downto 5);  
      when "00110" => result := dinX(37 downto 6);  
      when "00111" => result := dinX(38 downto 7);  
      when "01000" => result := dinX(39 downto 8);
```

```
when "01001" => result := dinX(40 downto 9);
when "01010" => result := dinX(41 downto 10);
when "01011" => result := dinX(42 downto 11);
when "01100" => result := dinX(43 downto 12);
when "01101" => result := dinX(44 downto 13);
when "01110" => result := dinX(45 downto 14);
when "01111" => result := dinX(46 downto 15);
when "10000" => result := dinX(47 downto 16);
when "10001" => result := dinX(48 downto 17);
when "10010" => result := dinX(49 downto 18);
when "10011" => result := dinX(50 downto 19);
when "10100" => result := dinX(51 downto 20);
when "10101" => result := dinX(52 downto 21);
when "10110" => result := dinX(53 downto 22);
when "10111" => result := dinX(54 downto 23);
when "11000" => result := dinX(55 downto 24);
when "11001" => result := dinX(56 downto 25);
when "11010" => result := dinX(57 downto 26);
when "11011" => result := dinX(58 downto 27);
when "11100" => result := dinX(59 downto 28);
when "11101" => result := dinX(60 downto 29);
when "11110" => result := dinX(61 downto 30);
when "11111" => result := dinX(62 downto 31);
when others => result := dinX(31 downto 0);
end case;

dout <= result;

end process bshft;

end rtl_bshfit;
```