```
library IEEE;                                                                    --IMPORTING LIBRARY
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.MATH_REAL.ALL;
library work;                                                                    --USING FILES FROM WORK DIRECTORY
use work.dif_ifft_pkg.ALL;                                                       --USING PACKAGE DIT_IFFT_PKG FROM WORK DIRECTORY
---------------------------------------------------------------------------------------------------------------------------
entity dif_ifft_8pt is                                                           --ENTITY DECLARATION
port(   s : in ar;                                                               --INPUT SIGNALS (TIME DOMAIN)
        y : out ar);                                                             --OUTPUT SIGNALS (FREQUENCY DOMAIN)
end dif_ifft_8pt;
---------------------------------------------------------------------------------------------------------------------------
architecture Behavioral of dif_ifft_8pt is --ARCHITECTURE DECLARATION
component butterfly is                                                           --COMPONENT DECLARATION
   port(
      b1,b2 : in complex;                                                        --INPUTS OF BUTTERFLY STRUCTURE
      w     :in complex;                                                         --PHASE FACTOR
      z1,z2 :out complex);                                                       --OUTPUTS OF BUTTERFLY STRUCTURE
end component;

signal z1,z2 : ar ;                                                              --DEFINING SIGNLAS Z1 AND Z2 WITH DEFAULT VALUE OF(0.0,0.0)

--PHASE FACTOR, w_N = e^(-j*(2*p/N)*n)
--w_N^n = cos((2*p/N)*n) - j*sin((2*p/N)*n);

--w_N VALUES FOR n = 0-4
constant w : ar2 := ( (1.0,0.0), (0.7071,0.7071), (0.0,1.0), (-0.7071,0.7071) );
signal p:ar;                                                                     --DEFINING SIGNAL P OF TYPE AR
begin
--FIRST STAGE OF BUTTERFLY, n = 0,2,4,6; N = 8
bf11 : butterfly port map(s(0),s(4),w(0),z1(0),z1(1));
bf12 : butterfly port map(s(2),s(6),w(2),z1(2),z1(3));                           --MAPPING INPUTS AND OUTPUTS
bf13 : butterfly port map(s(1),s(5),w(1),z1(4),z1(5));
bf14 : butterfly port map(s(3),s(7),w(3),z1(6),z1(7));

--SECOND STAGE OF BUTTERFLY, n = 0,2; N = 8
bf21 : butterfly port map(z1(0),z1(2),w(0),z2(0),z2(2));
bf22 : butterfly port map(z1(1),z1(3),w(0),z2(1),z2(3));                         --MAPPING INPUTS AND OUTPUTS
bf23 : butterfly port map(z1(4),z1(6),w(2),z2(4),z2(6));
bf24 : butterfly port map(z1(5),z1(7),w(2),z2(5),z2(7));

--THIRD STAGE OF BUTTERFLY, n = 0; N = 8
```

```
bf31 : butterfly port map(z2(0),z2(4),w(0),p(0),p(4));
bf32 : butterfly port map(z2(1),z2(5),w(0),p(1),p(5));                    --MAPPING INPUTS AND OUTPUTS
bf33 : butterfly port map(z2(2),z2(6),w(0),p(2),p(6));
bf34 : butterfly port map(z2(3),z2(7),w(0),p(3),p(7));

y(0)<= multi(p(0),(0.125,0.0));                                           --DIVIDING BY FACTOR OF 1/N
y(1)<= multi(p(7),(0.125,0.0));                                           --AND ASSIGNING TO OUTPUT SIGNAL
y(2)<= multi(p(6),(0.125,0.0));
y(3)<= multi(p(5),(0.125,0.0));
y(4)<= multi(p(4),(0.125,0.0));
y(5)<= multi(p(3),(0.125,0.0));
y(6)<= multi(p(2),(0.125,0.0));
y(7)<= multi(p(1),(0.125,0.0));
end Behavioral;
```