```
% - Fast Fourier Transform [64FFT]
% Univ. of the Ryukyus, Okinawa, Japan

function [y]=myfft64(x);
% internal buffer memory and output memory
x1 = zeros(64,1);
x2 = zeros(64,1);
x3 = zeros(64,1);
y  = zeros(64,1);
% stage 1
for mm = 0:1:15
   %radix-4
   twiddle1=exp(-2*pi*j*mm*1/64);
   twiddle2=exp(-2*pi*j*mm*2/64);
   twiddle3=exp(-2*pi*j*mm*3/64);
   x1(mm+1)  =x(mm+1)  +x(mm+17)+x(mm+33)  +x(mm+49);
   x1(mm+17)=(x(mm+1)-j*x(mm+17)-x(mm+33)+j*x(mm+49))*twiddle1;
   x1(mm+33)=(x(mm+1)  -x(mm+17)+x(mm+33)  -x(mm+49))*twiddle2;
   x1(mm+49)=(x(mm+1)+j*x(mm+17)-x(mm+33)-j*x(mm+49))*twiddle3;
end;
% stage 2
for nn = 0:16:48
   for mm = 0:1:3
      %radix-4
      twiddle1=exp(-2*pi*j*mm*4*1/64);
      twiddle2=exp(-2*pi*j*mm*4*2/64);
      twiddle3=exp(-2*pi*j*mm*4*3/64);
      x2(mm+nn+1)  =x1(mm+nn+1)  +x1(mm+nn+5)+x1(mm+nn+9)  +x1(mm+nn+13);
      x2(mm+nn+5) =(x1(mm+nn+1)-j*x1(mm+nn+5)-x1(mm+nn+9)+j*x1(mm+nn+13))
*twiddle1;
      x2(mm+nn+9) =(x1(mm+nn+1)  -x1(mm+nn+5)+x1(mm+nn+9)  -x1(mm+nn+13))
*twiddle2;
      x2(mm+nn+13)=(x1(mm+nn+1)+j*x1(mm+nn+5)-x1(mm+nn+9)-j*x1(mm+nn+13))
*twiddle3;
   end;
end;
% stage 3
for nn = 0:4:60
   x3(nn+1) = x2(nn+1)  +x2(nn+2)+x2(nn+3)  +x2(nn+4);
   x3(nn+2) = x2(nn+1)-j*x2(nn+2)-x2(nn+3)+j*x2(nn+4);
   x3(nn+3) = x2(nn+1)  -x2(nn+2)+x2(nn+3)  -x2(nn+4);
   x3(nn+4) = x2(nn+1)+j*x2(nn+2)-x2(nn+3)-j*x2(nn+4);
end;
% reorder
for n2 = 0:3
   for n1 = 0:3
      for n0 = 0:3
         y(16*n0+4*n1+n2+1)=x3(16*n2+4*n1+n0+1);
      end;
   end;
end;
```