ภาคผนวก ก.

โปรแกรมออกแบบวงจรครอสโอเวอร์เน็ตเวอร์ค

```
% 2 ways crossover design
close all; clear all; clc; format long;
FS = 48000; Fn = FS/2;
Wp = 3000/Fn; Ws = 13000/Fn; Rs = 35; Rp = 0.002;
[n, Wn] = ELLIPORD(Wp, Ws, Rp, Rs);
[b,a]= ellip(n,Rp,Rs,Wn);
zr = roots(b);
 pl = roots(a) ;

B0=[pl(1)*pl(2) -(pl(1)+pl(2)) 1];
A0=[1 -(pl(1)+pl(2)) pl(1)*pl(2)];

B1=[-(pl(3)) 1];
A1=[1 -(pl(3))];

N1 = 0.5*(conv(B0,A1));
N2 = 0.5*(conv(B1,A0));

D = conv(A0,A1);
NL = (N1+N2);
NH = (N1-N2);

[HL,T] = freqz(NL,D,1000);
[HH,T] = freqz(NH,D,1000);

t = 1/1000:1/1000:1;
semilogx(t,20*log10(abs(HL)),t,20*log10(abs(HH)))
axis([0 1 -80 5 ])
grid;
```

ภาคผนวก ข.

โปรแกรมสร้างไฟล์ตารางเปิดดู

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>

double a[50] ;
double su;
double Q,qq,b,c,ii,k;
int Bit,Do,Di,i,j,S;
int ind;
FILE *fp0,*pFile;
void main(void)
{
        int tab[20];
        char file1[10]="default.v";
        char file2[10]="tab.txt";
        char str[20] ;
        int K;
//*****************************************************************************
        printf("Generate Distribute Arithmetic\n");
        printf("\n Enter coefficient file name (tab.txt):");
        gets(file2);
        pFile = fopen (file2 , "r");
        if (pFile==NULL) perror ("Error opening file");
  else
  {
          for(i=0;i<100;i++)
          {
                  if(fscanf (pFile, "%s", str)==1)
                   a[i]=atof(str);
                   else
                    break;
           }
          fclose (pFile);
  }
  K = i-1;
  printf ("Scalling factor = %f \n",a[K]);
  for(i=0;i<K;i++)
  {
          a[i] =(double)(a[i] * a[K]);
          printf ("Coeficient of a(%d) = %f \n",i,a[i]);
  }
        printf("\n Enter output file name (default.v) :");
        gets(file1);
        if (pFile==NULL) perror ("Error opening file");
        else
        {
```

```
                fp0=fopen(file1,"w+");
                printf("\n Input Data Bus : %d\n",Di);
                Do = 16;
                Bit=Do;
//*************************************************************************
                Q  = pow((2),Bit);
                qq = floor(a[0]*(Q-1))/Q;
                b  = pow((2),Di);
                c  = pow((2),Do);
                printf("Q = %10.0f \nB = %f  \nC = %f \n",Q,b,c);
                Bit  = Bit-1;
                Di  = Di-1;
                Do = Do-1;
//*************************************************************************
                fprintf(fp0,"module coef1(Di,Do);\n");
                for(ind=0;ind<Di+1;ind++)
                {
                        fprintf(fp0,"// coef(%d) = %f  ;\n",ind,a[ind]);
                }
                fprintf(fp0,"\tinput[%d:0]Di;\n",Di);
                fprintf(fp0,"\toutput[%d:0]Do;\n",Do);
                fprintf(fp0,"\treg[%d:0]Do;\n",Do);
                fprintf(fp0,"\talways @(Di)\n\tbegin\n\t   case(Di)\n");
//*************************************************************************
                for(i=0;i<=9;i++)
                        tab[i]=0;
                for (i=0;i<=b-1;i++)
                {
                        ii=i;
                        k=0;
                        for(j=0;j<=Di;j++)
                        {
                                k  = pow((2),(Di-j));
                                if(k<=ii)
                                {       ii=ii-k;
                                        tab[j]=1;}
                                else
                                tab[j]=0;
                        }
                        su = 0;
                        for(ind=0;ind<Di+1;ind++)
                        {
                                printf("%d ",tab[ind]);
                                su = su+(a[ind]*tab[ind]);
                        }
                        printf("= %f\n ",su);
                        coef(su);
                }
//*************************************************************************
                fprintf(fp0,"\t      default : Do <= %d'b0;\n",Do+1);
                fprintf(fp0,"\t  endcase\n\t end\nendmodule");
                fclose(fp0);
        }
}
```

ภาคผนวก ค.

โปรแกรมวงจรครอสโอเวอร์เน็ตเวอร์คบน FPGA

```
module SW1(RL,Di,DT,DB);
   input RL;
   input [15:0] Di;
   output [15:0] DT;
   output [15:0] DB;

   assign DB[15:0]=Di[15:0] & {16{!RL}};
   assign DT[15:0]=Di[15:0] & {16{RL}};

endmodule

module sw_3(DT,DB,LR,A1,A2);
   input [15:0] DT;
   input [15:0] DB;
   input LR;
   output [15:0] A1;
   output [15:0] A2;
   wire [15:0]w1;
   wire [15:0]w2;
   wire [15:0]w3;
   wire [15:0]w4;

  assign w1[15:0]= DT[15:0] & {16{LR}};
  assign w2[15:0]= DB[15:0] & {16{! LR}};
  assign w3[15:0]= DT[15:0] & {16{! LR}};
  assign w4[15:0]= DB[15:0] & {16{LR}};
  assign A1 = w1 | w2 ;
  assign A2 = w3 | w4 ;
endmodule


module LIFO3(Di,Do,CLK,CLR,LR);
    parameter L=150;
    parameter width=16;
    input [width-1:0] Di;
    input  CLK;
    input  CLR;
    output  LR;
          output [width-1:0] Do;
          reg st;
          reg [width-1:0] ram0[L+3:0];
          reg [width-1:0] ram1[L+3:0];
          reg       [7:0]count;
          reg       [width-1:0] x;
          reg                 [7:0] ci;             // counter  up to L
          reg                 [7:0] co;             // counter up to L
```

```
integer i;
initial
begin
 for (i = 0; i < L+3; i = i+1)
            ram0[i] = 0;
            ram1[i] = 0;
  end
assign LR=st;
assign Do=x;

always @(posedge CLK)
begin
  if(CLR) begin
            co = L;
            ci = 1;
            st = 0;
            count = 255;
            x = 0;
  end
 else  begin
     count = count+1;
     if(count == L) begin
              count = 0;
              st=!st;
              if(st) begin
                 ci = ci - 1;
                co = co + 1;
              end
              else begin
                 ci = ci + 1;
                 co = co - 1;
              end
             end
             if(st) begin
                      ram0[co] <= Di;
                      x = ram1[ci] ;
                      co = co + 1;
                      ci = ci - 1;
            end
            else begin
                      ram1[ci] <= Di;
                      x = ram0[co] ;
                      ci = ci + 1;
                      co = co - 1;
            end
  end
end

endmodule
```

```
module Z_Del(Di,Do,Clk,CLR);
   parameter      L =300;
   input CLR;
   input  [15:0] Di;
   output [15:0] Do;
   input  Clk;
   reg    [15:0] ram[L+1:0];
   reg    [15:0] x;
   reg    [8:0]  ci;
   reg    [8:0]  co;
  integer i;

   initial
  begin
         for (i = 0; i < L+1; i = i+1)
          ram[i] = 0;
   end
          assign Do=x;
           always @(posedge Clk)
           if(CLR) begin
           x <= 0;  ci = 0;    co = 0;
           end
           else begin
                  ram[ci] <= Di;
                  x <=ram[co];
                  co = co+1; ci = ci+1;
                  if(ci == L-1)
                  co = 0;
                  else if(ci == L)
                  ci = 0;
           end
endmodule

module acc(A,B,Y,Co,Ci);
   parameter width = 16 ;
   input [width-1:0] A;
   input [width-1:0] B;
   output [width:0] Y;
   input Ci;
   output Co;
   wire [width-1:0]Do;
    reg   [width:0]result;
    wire t1;
         assign Do=B^{16{Ci}};
         assign t1 = A[15]^Do[15];
         assign Co = result[width]^t1;
         assign Y[16]=Co;
         assign Y[15:0] =  result[width-1:0];
         always @(A or Do or Ci)
                  result = A + Do + Ci;
endmodule
```

```
module PISO(Di,Do,CK,RST,LD);
   parameter width = 16;
   input [width-1:0] Di;
   output Do;
   input CK,RST;
   input LD;
          reg      [width:0]dat;
          assign Do = dat[0];
          always @ (posedge CK or posedge RST or negedge LD )
          begin
                   if (RST)
                            dat[16:0] = 0  ;
             else begin
                   if(!LD) begin
                            dat[15:0] = Di;
                            dat[16] = 0;
                   end
                   else begin
                            dat[15:0]  = dat[16:1];
                            dat[16] = 0;
                   end
         end
  end
endmodule

module PISO(Di,Do,CK,RST,LD);
  parameter width = 16;
   input [width-1:0] Di;
   output Do;
   input CK,RST;
   input LD;
          reg [width:0]dat;
          assign Do = dat[0];
          always @ (posedge CK or posedge RST or negedge LD )
          begin
                   if (RST)
                            dat[16:0] = 0  ;
             else begin
                   if(!LD) begin
                            dat[15:0] = Di;
                            dat[16] = 0;
                   end
                   else begin
                            dat[15:0]  = dat[16:1];
                            dat[16] = 0;
                   end
         end
 end
endmodule
```

ผลงานวิจัยที่ได้รับการตีพิมพ์

[1] J.Orsawat, D. Sookcharoenphol and K. Janjitrapongvej, “An Implementation of Digital Crossover Network by using Perfect Linear Phase IIR Filters”, The Third International Symposium on Communications and Information Technologies (ISCIT 2003)

[2] C. Kanna, J.Orsawat, D. Sookcharoenphol and K. Janjitrapongvej,“Finite Word Length Effect of Digital Crossover Network using Perfect Linear Phase IIR Filters” , The 4 th Information and Computer Engineering Postgraduate Workshop (ICEP 2004)

[3] เจตน์ ออสวัสดิ์, ดลชัย สุขเจริญผล และ กนก เจนจิระพงศ์เวช ,"การออกแบบการลดทอนในช่วงหยุดความถี่ผ่านอย่างสูงของวงจรกรองความถี่ IIR ที่มีเฟสเป็นเชิงเส้น ” , วารสารพระจอมเกล้าลาดกระบัง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เดือน เมษายน 2547 ฉบับที่ 2 ปีที่ 16

[4] ชาญณรงค์ การนา, เจตน์ ออสวัสดิ์, ดลชัย สุขเจริญผล และ กนก เจนจิระพงศ์เวช, " วงจรดิจิตอลครอสโอเวอร์เน็ตเวิร์คแบบ ใหม่ที่ให้กรุ๊ปดีเลย์ ราบเรียบ", จัดการประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 27 "EECON-27 (27th Electrical Engineering Conference)"