بسم الله الرحمن الرحیم

نام پروژه:

محاسبه FFT و ارتباط با ISA_BUS

گرداورندگان:

مسعود رضایی (۸۳۳۰۲۷۱۰۱۴)

محمد علی اکبری (۸۳۳۰۲۷۱۰۰۳)

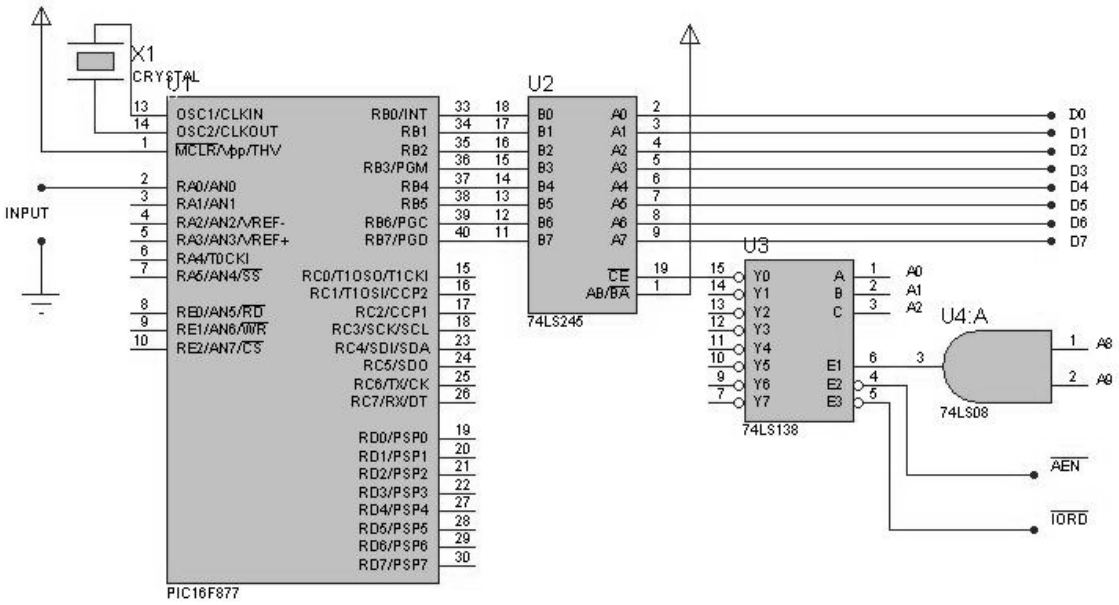
لیلی صدیق (۸۳۳۰۲۷۱۰۱۹)

استاد راهنما:

جناب اقای دکتر صدوقی

محاسبه FFT از سیگنال ورودی به ISA_BUS

چون کامپیوتر یک سیستم دیجیتال بوده سیگنال انالوگ ورودی باید به سطوح دیجیتال تبدیل شده و به کامپیوتر برای پردازش ارسال شود . تبدیل به سطوح دیجیتال یعنی نمونه گیری در حوزه ی زمان، ما برای این منظور احتیاج به یک HIGH SPEED ADC داریم , که از میکروکنترلرPIC بدیل منظور استفاده کرده ایم. میکروکنترلر سیگنال انالوگ را به دیجیتال تبدیل کرده و از طریق بافر74245 که با مدارات دیکودر پورت کنترل میشود اطلاعات را بر روی DATA_BUS قرار می دهدو از طریق نرم افزاری که به زبان C نوشته شده اطلاعات را پردازش وFFT را محاسبه و آن را رسم می کند .



آشنایی با ISA_BUS :

برخی از پایه های باس ISA در زیر شرح داده می شود .

$A_0 - A_{19}$:

پایه های $A_0 - A_{19}$ که$SA_0 - SA_{19}$ نیز نامیده می شود ۲۰ پایه آدرس برای کار با حافظه یا I / O فراهم می کند. لازم به ذکر است که برای I / O فقط پایه های $A_0 - A_{15}$ استفاده می شود.

AEN :

AEN (فعال کننده آدرس)تعیین می کند ریز پردازنده باسها را کنترل می کند یا DMA اگر AEN = 1 باشد. DMA باس آدرس ، داده ، MEMW ، MEMR ، IOW ، IOR را کنترل می کند و وقتی AEN = 0 باشد ریزپردازنده باسها راکنترل می کند .

CLOCK : این پایه پاس ساعتی برابر فرکانس سیستم فراهم می کند تا همه عملیات های خواند ن و نوشتن حافظه I / O باهم همزمان شوند .

$D_0 - D_7$:

باس داده دو طرفه ۸ بیتی است مسیر رد وبدل اطلاعات بین ریز پردازنده حافظه و I / O می باشد که توسط وسایل ۸ بیتی I / O که به SLOT متصل شده اند استفاده می شود .

DRQ1 وDRQ2 وDRQ3 وDACK1 وDACK2 و DACK3 سیگنالهای درخواست DMA و پذیرش DMAبرای پیاده سازی DMAدر انتقال داده استفاده می شود.

IOCHCHK :

سیگنال بررسی کننده کانال I / O یک سیگنال ورودی فعال با سطح صفر است که وقوع خطا در کارت افزون شونده در شکاف توسعه را مشخص می کند اصطلاح کانالI / O توسطIBM به منظور معرفی هرکارت افزون شونده به برد مادر از طریق شکاف توسعه به کار برده می شود . اگر یک کارت حافظه در شکاف توسعه قرار داده نشود دراین وقوع خطاءپریتی را آشکار می کند به طور داخلی سیگنال IOCHCHK به پایه NMI پردازنده متصل می شود تا خطاهای غیر قابل تصحیح را آشکار کند .

IOR و IOW

هردو سیگنال فعال با سطح صفرهستند IOW به وسیلهI / O دستور می دهد که داده رااز باس بگیرد وIO بوسیله I / O دستور می دهد که داده راروی BUS قرار دهد.

IRQ9 و IRQ3 – IRQ7

درخواست وقفه (IRQ)توسط وسائلI / O به کار برده می شودتا به پردازنده اطلاع دهد که وسیله نیاز به سرویس دارد اگر بیش ازیک درخواست فعال شود اولویت بهIRQ9 داده میشودسپس به IRQ3 تا IRQ7 پائین ترین اولویتIRQ0 وIRQ9 بالاترین اولویت رادارد.

OSC

پایه OSC (نوسان کننده) یک سیگنال خروجی بافرکانس MHZ ۱٤/۳۱۸ است که دارای چرخه کاری ۵۰ ٪ است.

REFRESH

REFRESH یک سیگنال فعال باسطح صفراست وقتی که سیگنال خروجی است مشخص می کند که چرخه بازنویسی حافظه پویادرحال اجرا است.این سیگنال همچنین میتواندبه عنوان یک ورودی توسط وسیله روی شکاف توسعه استفاده می شود.تاچرخه بازیابی رامشخص کند.

RESET DRV

یک سیگنال خروجی فعال با سطح یک است. بطورداخلی توسط بردمادربه منظورRESET یا مقدار دهی اولیه وسایل جانبی قبل ازبرنامه ریزی BIOS استفاده می شود.این سیگنال همچنین میتواندبرای همین منظوردروسایل نصب شده درشکاف توسعه استفاده شود.

SMEMR و SMEMW :

هردو سیگنال خروجی فعال با سطح صفراست SMEMR به حافظه دستور می دهدکه داده موردنظر راروی BUS قراردهد SMEMW به حافظه دستورمی دهد که داده را ازروی BUS بردارد.

Tc :

یک سیگنال خروجی فعال باسطح یک است وزمانی فعال میشودکه یکی ازکانالهای DMA آخرین بایت راانتقال دهد.

پایه های 5+ و۵- و۱۲+ و۱۲- و GND :

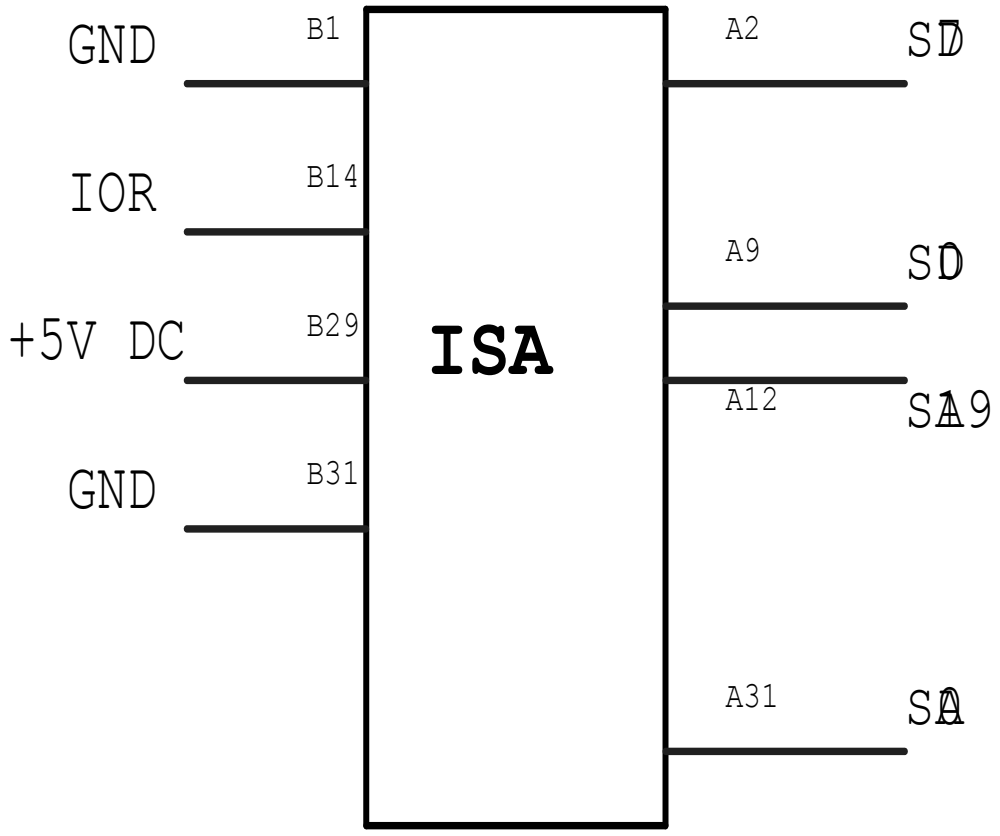
درمجموع ٦پایه برای ولتاژهای تغذیه وزمین تخصیص داده شده است . توجه شودکه فقط دوپایه زمین درتمام ٦۲ پایه باس ISA وجوددارداین امریکی ازموانع اصلی سرعت باس ISA به بیش از MHT ۸ است.

بامعرفی IBM PCIAT درسال ۱۹۸٤ برای سازگاری باباس ۱٦بیتی وباس آدرس ۲٤بیتی ریزپردازنده ۸۰۲۸٦ ، ۳٦ پایه دیگربه PC/XT اضافه شد. ۲٦پایه جدید PC/AT برای سیگنالهای داده $D_8 - D_{15}$ سیگنالهای

آدرس A_{21} -A_{23} ، سیگنالهای کنترل DMA جدید وبرخی دیگرازسیگنالهای کنترل سیستم بکاربرده شد. به دلیل عدم استفاده ازقسمت Extended دراین موردخاص ازتوضیح این قسمت پرهیزمی کنیم .

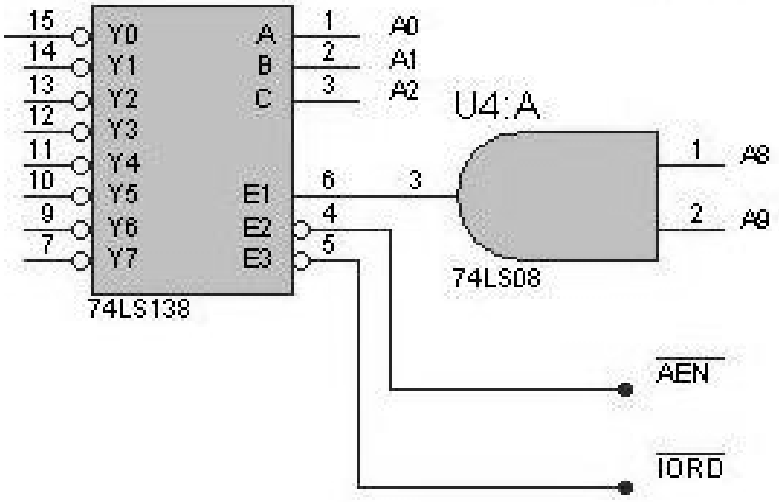
پایه های استفاده شده دراینترفیس مربوطه :

به منظورارتباط با Slot ISA دراینترفیس مربوطه از پایه D_0 - D_8 برای دریافت DATA و همچنین A_0 -A_{15} برای آدرس دهی و decoding وارتباط باپورتهای خارجی وهمچنین ازپایه I/O استفاده شده است . که توضیح هرکدام ازپایه هادرقسمت قبلی ذکرشده است .



دیکودر پورت 300H:

از یک۷۴۱۳۸ که یک دیکودر۸×۳ می باشد بدین منظور استفاده شده است . این دیکودر سه سیگنال کنترل دارد ، دو سیگنال فعال به ۰ و یک سیگنال فعال به۱ میبا شد. در این سیستم سیگنال های کنترلی ازIORD وAEN آمده و دیگر کنترل کهA8 و A9 با همAND شده و به 74138 متصل گردیده است . در این مدار از FULL DECODING استفاده نشده است چون هر گاه نرم افزار در حال اجرا است ، دیگر پورت های کامپیوتر استفاده ای نداشته و کامپیوتر آنها را صدا نمی زند و تداخل اطلاعات پیش نمی آید .



ADC:

یکی از مهمترین مزیتهای میکروکنترلر PIC دارا بودن مبدل های انالوگ به دیجیتال داخلی میباشد.تعداد این ورودی ها در میکرو کنترلرهای PIC بستگی به نوع انها دارد. بعنوان مثال PIC16F877 دارای ۸ کانال ورودی ADC میبا شد که ما در این پروژه از این IC استفاده کردهایم.
در این نوع میکرو کنترلر ها با تنظیم رجیسترهای مربوط به مبدل های انالوگ به دیجیتال میتوان عملکرد مبدل های انالوگ به دیجیتال را تعیین کرد.برای اطلاعات بیشتر راجع بهPIC به DATASHEET شرکت MICROCHIP مراجعه شود.

ADCON0:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit7							bit0

bit 7-6: **ADCS1:ADCS0: A/D Conversion Clock Select bits**
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock derived from an RC oscillation)

bit 5-3: **CHS2:CHS0**: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)(1)
110 = channel 6, (RE1/AN6)(1)
111 = channel 7, (RE2/AN7)(1)

bit 2: **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
If ADON = 1
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1: **Unimplemented**: Read as '0'

bit 0: **ADON**: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shutoff and consumes no operating current

که شرح بیت ها بدین صورت می باشد :

بیت های ۶و۷: تعیین فرکانس نمونه گیری

بیتهای ۵و۴و۳: تعیین منبع ورودی انالوگ

بیت ۲:با انتخاب ۱ ADC شروع به تبدیل سیگنال انالوگ به دیجیتال و با اتمام کار تبدیل این بیت به صورت خودکار ۰ میشود.

بیت ۰ :فعال ساز بلوک ADC

ADCON1:

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit7							bit0

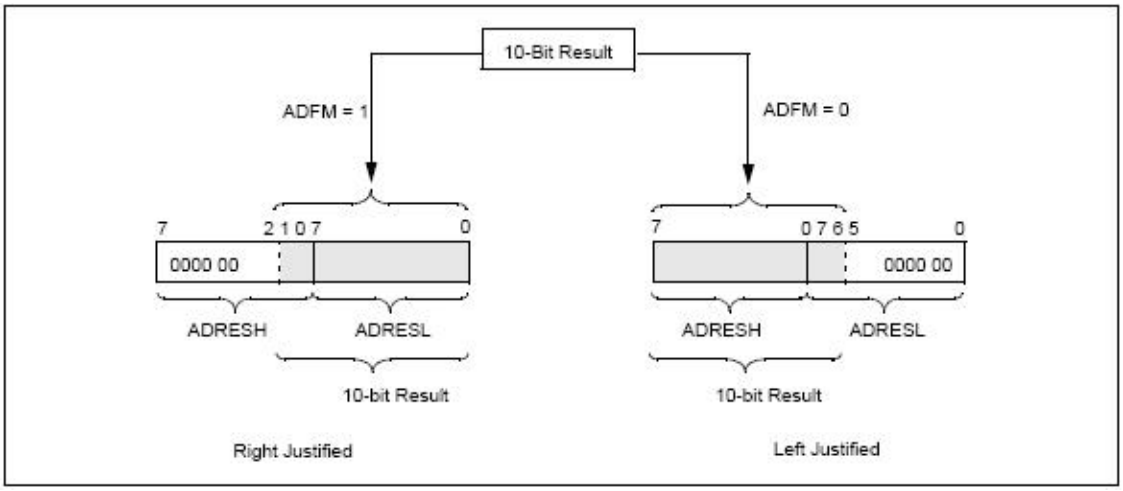
bit 7: **ADFM: A/D Result format select**
1 = Right Justified. 6 most significant bits of ADRESH are read as '0'.
0 = Left Justified. 6 least significant bits of ADRESL are read as '0'.

bit 6-4: **Unimplemented: Read as '0'**

bit 3-0: **PCFG3:PCFG0**: A/D Port Configuration Control bits

که شرح بیت ها بدین صورت می باشد .

بیت ۷ : خروجی ADC که ۱۰ بیت میباشد در دو ثبات ADRESH , ADRESL که به ترتیب بایت کم ارزشتر و بایت پرارزشتر میبا شند. حال میتوان انتخاب کرد که ۶ بیت اضافی در کدام ثبات باشد.



بیت های ۰و۱و۲و۳ : برای تنظیم بیت های پورت A به عنوان ورودی انالوگ یا ورودی وخروجی دیجیتال ویا به عنوان ANALOG REFF. می باشد. نحوه تنظیم این چهار بیت به صورت زیر میباشد:

PCFG3: PCFG0	AN7(1) RE2	AN6(1) RE1	AN5(1) RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN / Refs(2)
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input
D = Digital I/O

برنامه میکرکنترلربه زبان بیسیک در محیط micro code stdio نوشته شده که به صورت زیر می باشد :

ابتدا بیت ADCON0.2 را یک کرده سپس منظر می ماند تا کار تبدیل به اتمام برسد (بیت ADCON0.2 صفر شود) با اتمام کار مقدار بدست آمده را که 10 BIT می باشد در دو ثبات ADRESH وADRESL ذخیره می شود .چون ما برنامه را به صورتLEFT JUSTIFIED نوشته ایم مقدار ADRESH را بر روی پورت B ریخته که این پورت بهBUS BUFFER متصل بوده و اطلاعات را روی BUS قرارمی دهد.

```
DEFINE OSC 12
TRISB     = %00000000          ' Set PORTB to all output
TRISA     = %11111111          ' Set PORTA to all input
ADCON1 = %00000010             'Set PORTA analog and LEFT ‘justify
                               result
ADCON0 = %00000001             'Configure and turn on A/D Module
‘////////////////////////////////////////////
loop:  ADCON0.2 = 1            ' Start Conversion
NotDone:
If ADCON0.2 = 1 Then NotDone   ' Wait for low on bit-2 of ADCON0,
                               'conversion finished

      PORTB = ADRESH           ' Move high byte of result to adval
```

```
        Goto loop                    ' Do it forever
End
```

برای محاسبه ی FFT از فرمول :

$$F(u)=\sum_{x=0}^{x=n-1}\left[f(x)\times e^{\left(\frac{-j\times 2\pi\times x\times u}{n}\right)} \right]$$

استفاده می شود که می توان آن را به صورت زیر بسط داد

$$e\hat{}u = cos(u)\text{-}jsin(u)$$

حال ضرایب REAL وIMAG را جدا جدا حساب کرده و در فرمول زاویه و اندازه قرارمی دهیم و نهایتاً آنها را رسم می کنیم . حال مشاهده میشود که سیگنال ورودی ثابت است ولی شکل DFT متغیر است. ما برای اینکه شکل DFT که از سیگنال نمونه گیری شده گرفته میشود ثابت وقابل مشاهده باشد در برنامه تکه ای خاص از سیگنال را رسم میکنیم یعنی نقطه ماکزیمم را پیدا کرده و از ان نقطه به بعد به تعداد NP نمونه گیری میشود بنا بر این به نظر میرسد که سیگنال ورودی و کامپیوتر همزمان شده اند. ما برای اینکه این نکته را نشان دهیم سیگنال نمونه گیری شده را نیز رسم کرده ایم.

برنامه نوشته شده به قرار زیر است که نتیجه ان با نرم افزار MATLAB مقایسه شده است:

```
#include<dos.h>
#include<stdio.h>
#include<math.h>
#include<graphics.h>
#include<conio.h>
#include<complex.h>
void main(void){
double p=1,teta;
FILE *stream,*bin;
int Fl,n,w,N,W,NP=100,poly2[8];
poly2[0]=0;poly2[1]=0;poly2[2]=640;poly2[3]=0;
poly2[4]=640;poly2[5]=480;poly2[6]=0;poly2[7]=480;
unsigned char    a[100];
float tM,A1[ 600],A2[600];
char sw,w0;
int gdriver = DETECT, gmode;
initgraph(&gdriver, &gmode, "");
        setfillstyle(9,8);
        fillpoly(4,poly2);
        outtextxy(109,10,"ABS");
        outtextxy(308,10,"IMAG");
        outtextxy(508,10,"REAL");
M=200/NP;
for(n=0;n<(NP/4);n++)a[n]=5;
for(n=25;n<(NP/2);n++)a[n]=0;
for(n=50;n<(NP*3/4);n++)a[n]=5;
for(n=75;n<NP;n++)a[n]=0;
//for(n=0;n<NP;n++)a[n]=n;
stream = fopen("c:/dft.txt", "a+");
bin=fopen("c:/dft.hex","w+b+");
Fl=0;
while(1){
        W=0;
        for(n=0;n<NP;n++)if(a[n]>W)W=a[n];
here:       if(inportb(0x300)!=W)goto here;
here1:              if(inportb(0x300)==W)goto here1;
    for(n=0;n<NP;n++){a[n]=inportb(0x0300);a[n]=inportb(0x0300);}
again:
```

```
        if(Fl<2) Fl++;
         for(w=0;w<NP;w++){
                A1[w]=0;
                A2[w]=0;
                for(n=0;n<NP;n++){
                        A1[w]=A1[w]+cos(n*w*2*M_PI/NP)*a[n];
                        A2[w]=A2[w]-sin(n*w*2*M_PI/NP)*a[n];
                        }
                          if(Fl<2){
                          fprintf(stream,"%d:      real=   %f          image=   %f
a[%d]=%d\n",w,A1[w],A2[w],w,a[w]);
                          fwrite(&a[w],sizeof(a[w]),1,bin);}
               t=(sqrt((A1[w]*A1[w])+(A2[w]*A2[w])))*p;
               W=w*M;
               for(N=W;N<((w+1)*M);N++){
                        setcolor(0);
                        line(15+N,460,15+N,20);
                        if(t>219)t=219;
                        if(t<-219)t=-219;
                        setcolor(9);
                        line(15+N,240,15+N,240-t);}
               if(A1[W]!=0){
                        teta=A2[W]/A1[W];
                        teta=atan(teta);
                        teta=teta*90/M_PI;}
               putpixel(15+W,330-teta,14);
               t=W/M;
               t=a[t]*p;
               if(t>219)t=219;
               if(t<-219)t=-219;
               N=(W+1)/M;
               N=a[N]*p;
               if(N>219)N=219;
               if(N<-219)N=-219;
               setcolor(11);
               line(15+W,240-N,16+W,240-t);
               ////////////////////////
               setcolor(8);
               line(220,460,220,20);
               t=A2[w]*p;
```

```
W=w*M;
for(N=W;N<((w+1)*M);N++){
        setcolor(0);
        line(220+N,460,220+N,20);
        if(t>219)t=219;
        if(t<-219)t=-219;
        setcolor(9);
        line(220+N,240,220+N,240-t);}
t=W/M;
t=a[t]*p;
if(t>219)t=219;
if(t<-219)t=-219;
N=(W+1)/M;
N=a[N]*p;
if(N>219)N=219;
if(N<-219)N=-219;
setcolor(11);
line(220+W,240-N,221+W,240-t);
//////////////////////////
setcolor(8);
line(425,460,425,20);
t=A1[w]*p;
W=w*M;
for(N=W;N<((w+1)*M);N++){
        setcolor(0);
        line(425+N,460,425+N,20);
        if(t>219)t=219;
        if(t<-219)t=-219;
        setcolor(9);
        line(425+N,240,425+N,240-t);}
t=W/M;
t=a[t]*p;
if(t>219)t=219;
if(t<-219)t=-219;
N=(W+1)/M;
N=a[N]*p;
if(N>219)N=219;
if(N<-219)N=-219;
setcolor(11);
line(425+W,240-N,426+W,240-t);
```

```
}//////////////////////////////////////////
if(kbhit()){
        sw=getch();
        switch(sw){
                case 'H': p=p*2;
                        goto again;
                case 'P': p=p/2;
                        goto again;
                case 's': Fl=0;
                        goto again;

                case 'q': fclose(stream);
                        fclose(bin);
                        return;
                case 'Q': fclose(stream);
                        fclose(bin);
                        return;
                default :
                goto again;
        }       }
}
}
```

در محیط MATLAB برای محاسبه FFT میتوان از برنامه زیر استفاده کرد.

```
clear;
%t=0:pi/20:2*pi;
%x=sin(t);
x=[ones(1,25),zeros(1,25),ones(1,25),zeros(1,25)];
X=fft(x);
Len=length(x);
%x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N
N=Len;
for i=1: Len
    Yr(i)=0;Yi(i)=0;
   for n=1:N
       Yr(i)=Yr(i)+x(n).*cos(2*pi*(i-1)*(n-1)/N);
       Yi(i)=Yi(i)-x(n).*sin(2*pi*(i-1)*(n-1)/N);
   end
end
figure(1 ),bar(x);
figure(2 ),bar(abs(Yr-i*Yi));
figure(3 ),bar(Yi);
figure(4 ),bar(Yr);
```

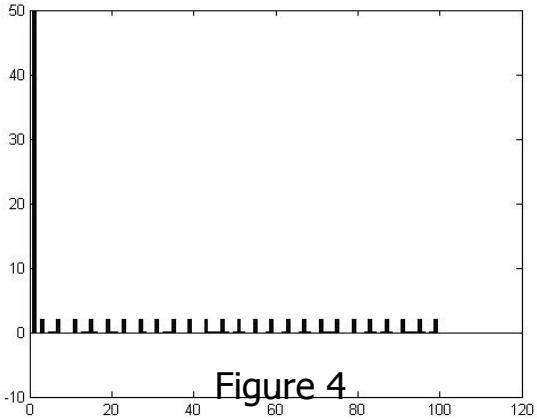
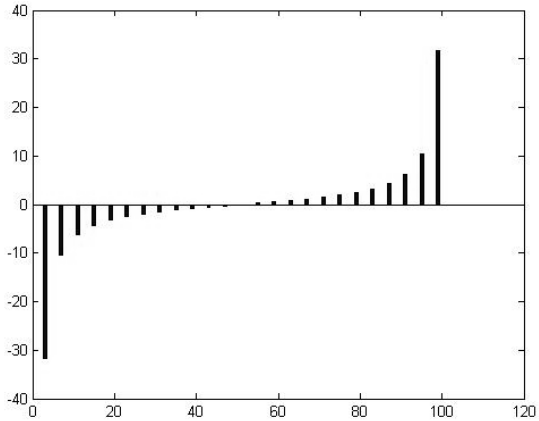


Figure 4



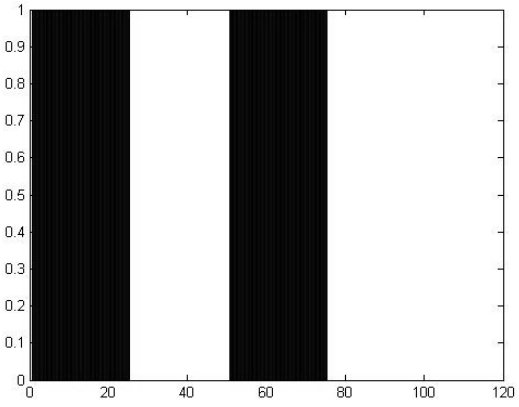


Figure 1

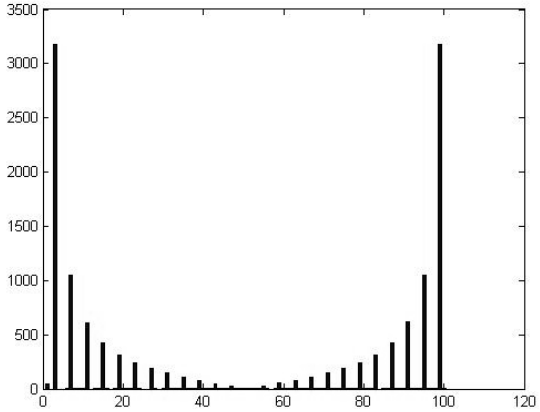


Figure 2