بسمه تعالی

عنوان پروژه:

شناسایی سیستم با استفاده از پورت ISA

گردآورندگان:

مصطفی اسماعیلیان ، سید مصطفی عبداله پور

محمد عسکری

استاد راهنما :

دکتر صدوقی

بهمن ۸۴

عناوین گزارش پروژه :

مقدمه ای بر تابع تبدیل سیستم وکاربرد آن

وکاربرد آن در شناسایی سیستمهاLMSالگوریتم

آشنایی باISA SLOT

شرح کار

تابع تبدیل:

بی شک برای کنترل و بررسی هر سیستم فیزیکی به دانش در مورد خصوصیات ذاتی سیستم احتیاج داریم .

برای بررسی خصوصیات یک سیستم باید ابتدا آن سیستم را مدل سازی کنیم و سیستم مورد نظر را به یک رابطه ریاضی (معادله دیفرانسیل) تقریب بزنیم .

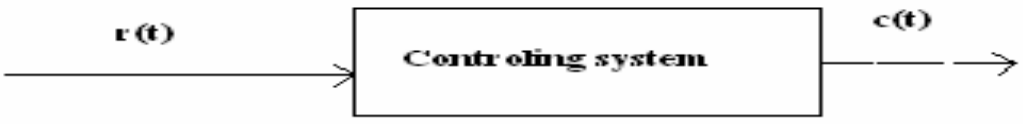
برای مدل سازی سیستم های فیزیکی روش های مختلفی وجود دارد که عبارتند از:

۱ : معدله دیفرانسیل ۲: تابع تبدیل ۳: معادلات حالت و...

ما در این قسمت هر چند به صورت مختصر به بررسی اهمیت تابع تبدیل و خصوصیات و کاربرد آن می پـردازیم .

اهمیت تابع تبدیل به قدری است که اگربگوییم علم کنترل بر مبنای آن بنا نهاده شده است چیزی به گزاف نگفتـه ایم . در نظریه ی کنترل توابعی که به عنوان تابع تبدیل شناخته میشوند معمولاً برای مـشخص کـردن روابـط بـین ورودی و خروجی عناصر یا سیستمی استفاده می شود که بتوان آن را به صورت معادلات خطی تغییـر ناپـذیر بـا زمان بیان کرد. بنا براین مطلب رابامعرفی معادله دیفرانسیل به عنوان پلی برای یافتن تابع تبدیل شروع می کنیم.

از لحاظ ریاضی و تئوری هر سیستم خطی تغییر ناپذیر با زمان (ال تی آی) با یک معادله دیفرانـسیل بـا ضـرایب ثابت و حقیقی نمایش می دهند اگر یک سیستم کنترلی به صورت زیر داشته باشـیم رابطـه ی ورودی و خروجـی آن را می توان به صورت زیر نوشت



$$d^n c(t)/t^n + a_1 d^{n-1} c(t)/t^{n-1} + ... + a_n c(t) = d^m r(t)/t^m + d^{m-1} r(t)/t^{m-1} + ... b_m r(t)$$

یک سیستم را به وسیله ی تابع تبدیل آن معرفی می کنند و تابع تبدیل یک معادله ی دیفرانسیل خطی مستقل از زمان را به صورت نسبت تبدیل لاپلاس خروج به تبدیل لاپلاس ورودی وقتی که تمام شرایط اولیه صفر باشند یعنی :

$$G(s)=\frac{C(s)}{R(s)}=\frac{b0\,s^{m}+b1\,s^{m+1}+....+bm}{a0\,s^{n}+a1\,s^{n+1}+....+am}$$

مفهوم تابع تبدیل وروال آن اهمیت بسزایی دارد ؛ زیرا مدل ریاضی عناصر سیستم را برای تحلیلگر و طراح فراهم می سازد . همچنین داشتن تابع تبد یل یک سیستم ، این امکان را به ما می دهد که سیستم فیزیکی را از حوزه عمل به حوزه ی کامپیوتر و نرم افزار برده و حالات و جایگاه های مختلفی را که ممکن است سیستم در آن واقع شود را بررسی کرده و عملکرد سیستم را مشخص کنیم .

از انجایی که تابع تبدیل ارتباط میان ورودی و خروجی یک سیستم را نشان می دهد با معلوم بودن تابع تبدیل می توان پاسخ سیستم را به ورودی های دلخواه به دست آورد . به عنوان مثال می توان ورودئی های مختلفی از قبیل ضربه، پله، شیب ، شتاب و ... را به سیستم اعمال نمود و رفتار سیستم را نسبت به این ورودی ها مشاهده کرد و با تغییر پارامتر های سیستم اصلی به سیستم مورد نظر و مطلوب نزدیک تر شد .

همچنین قابل ذکر است که با مشخص بودن تابع یک سیستم می توان دیگر مدل های یک سیستم فیزیکی مانند معادلات حالت ، معادله دیفرانسیل، نمودار حالت معادله گذر حالت و... را به دست آورد .

و همچنین می توان بر روی مفاهیمی چون پایداری ، کنترل پذیری و مشاهده پذیری که درعلم کنترل اهمیت بسزایی دارد بحث نمود .

شناسایی سیستم به روش معادلات بازگشتی :

در این روش ابتدا تابع تبدیلی بصورت زیر برای سیستم مورد نظر در نظر می گیریم

$$Y[z]/x[z] = (a_n z^{-1} + a_{n-1} z^{-2} + a_{n-2} z^{-3} + ...)$$

$$Y[n] = a_n x[n] + a_{n-1} x[n-1] + a_{n-2} x[n-2] + a_{n-3} x[n-3] + \ldots$$

برای مشخص شدن سیستم کافی است ضرایب معادله بازشگتی فوق معین شوند.

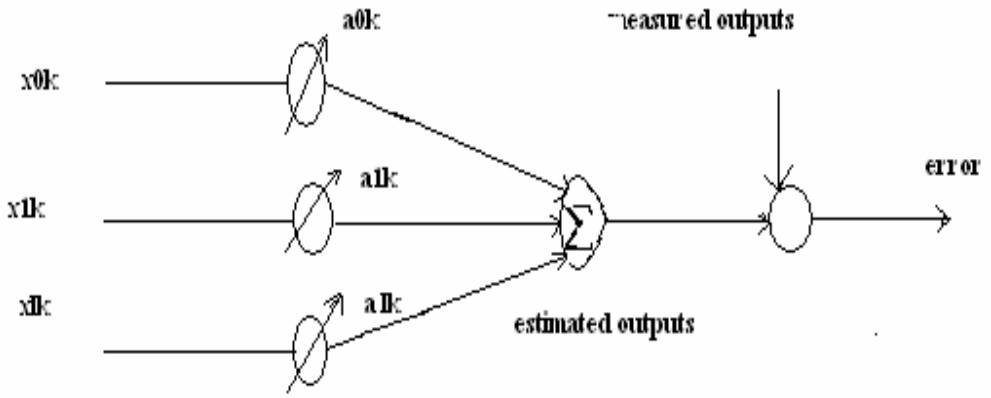
تعیین ضرایب معادله بازگشتی:

x[1] x[2],x[3],.....x[k], فرض کنید که ورودیهای زیر به سیستم اعمال شده است

y[1],y[2],y[3],.....y[k] و خروجیهای مقابل مشاهده شده است

$a_{n-3}, a_{n-2}, a_{n-1}, a_n$,... در ابتدا تخمینی برای ضرایب

برای سیستم تخمین می زنیم Y`[k] صورت تصادفی انتخاب می کنیم و به ازای ورودیها بدست آمده در این مرحله را حساب می کنیم که برابر است با $e_{(k)} = y_{(k)} - y`_{(k)}$ محاسبه شده در مراحل بعدی ما باید ضرایب چنان تغییر دهیم که خطا به سمت صفر میل کند.



را بدست می آوریم و مشتق آنرامساوی صفر قرار می دهیمerrorبرای اینکار توان دوم بعد از ساده سازی روابط معادله زیربدست می آید

$$a_{k+1} = a_k + 2u^*x^*e$$

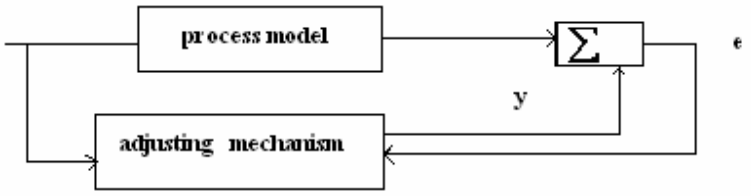
$$\begin{bmatrix} a_{0\,k+1} \\ a_{1\,k+1} \\ \cdots \\ a_{n\,k+1} \end{bmatrix} = \begin{bmatrix} a_{0k} \\ a_{1k} \\ \cdots \\ a_{n1} \end{bmatrix} + 2^* \mathbf{u}^* \begin{bmatrix} x_{0k} \\ x_{1k} \\ \cdots \\ x_{nk} \quad 1 \end{bmatrix}$$

ضرایب جدیدی تولید می شود که این ضرایب توسط [illegible] این رابطه با استفاده از سری کوچکتری ایجاد می کند و با ادامه این روند خطا به سمت صفر میل خواهد کرد

می گویند. LMS این روش را روش

یک سیستم را شناسایی کنیم.onlineما میتوانیم بصورت

بلوک دیاگرام کلی فرایند را می توانیم بصورت زیر رسم کرد.



آشنایی با ISA SLOT

باس ISA باس IBM PC/AT است به همین دلیل بعضی ها آنرا باس AT می نامند هریک 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 = 1باشد. 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 توسط IBM 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.

IRQ3 - IRQ7 و IRQ9

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 اولویت 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 اجرااست.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 :

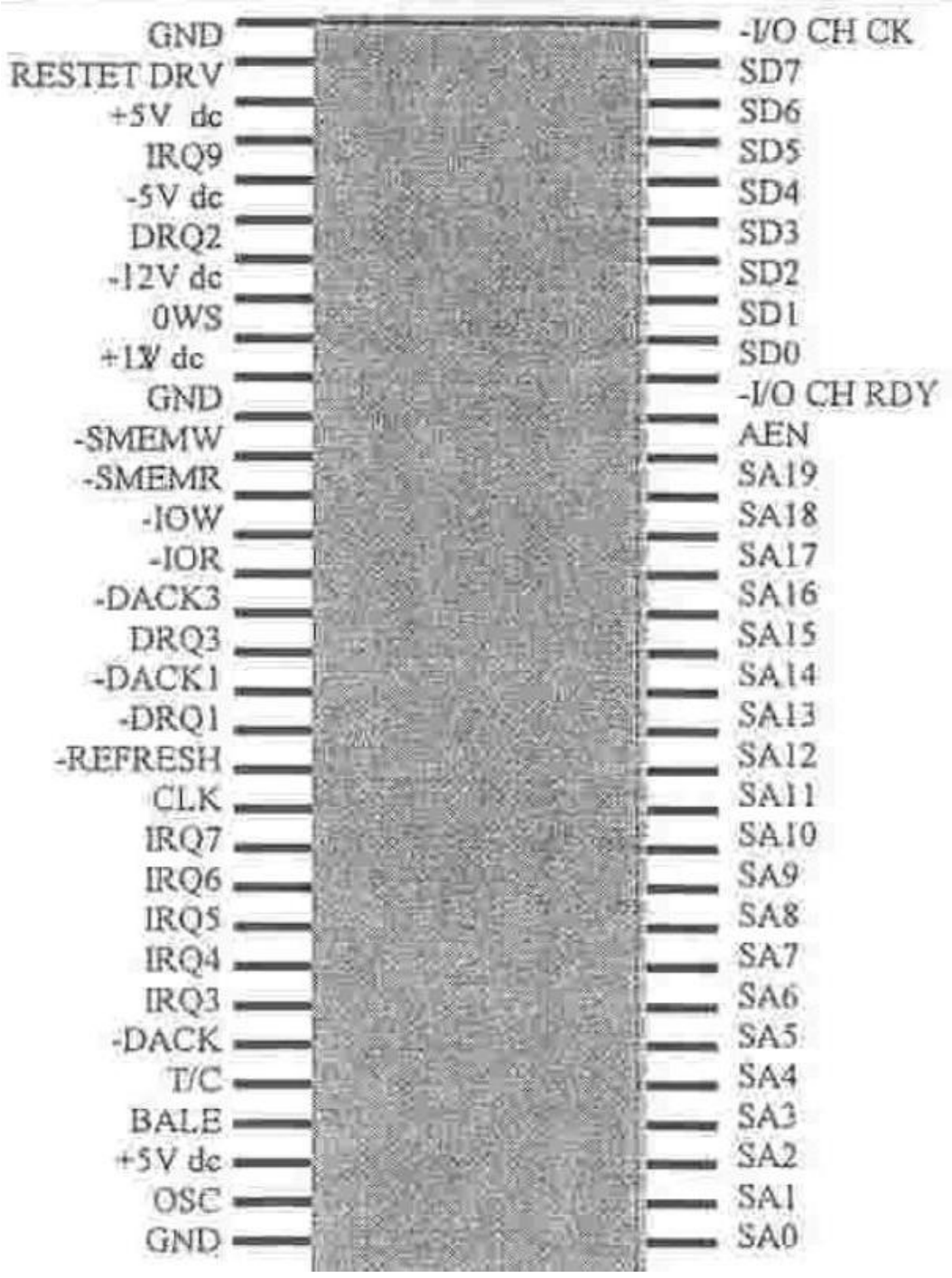
هردو سیگنال خروجی فعال با سطح صفراست SMEMR به حافظه دستور می دهدکه داده موردنظر راروی BUS قراردهد SMEMW به حافظه دستورمی دهد که داده را ازروی BUS بردارد.

Tc :

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

پایه های s+ و۵- و۱۲+ و۱۲- GNP وGNP :

درمجموع ٦پایه برای ولتاژهای تغذیه وزمین تخصیص داده شده است . توجه شودکه فقط دوپایه زمین درتمام ٦٢ پایه باس ISA وجودداردراین امریکی ازموانع اصلی سرعت باس ISA به بیش از MHT ٨ است. بامعرفی IBM PCIAT درسال ١٩٨٤ برای سازگاری باباس ١٦بیتی وباس آدرس ٢٤بیتی ریزپردازنده ٨٠٢٨٦ ، ٣٦ پایه دیگربه PC/XT اضافه شد. ٢٦پایه 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 .


GND
RESTET DRV
+5V dc
IRQ9
-5V dc
DRQ2
-12V dc
0WS
+12V dc
GND
-SMEMW
-SMEMR
-IOW
-IOR
-DACK3
DRQ3
-DACK1
-DRQ1
-REFRESH
CLK
IRQ7
IRQ6
IRQ5
IRQ4
IRQ3
-DACK
T/C
BALE
+5V dc
OSC
GND
-I/O CH CK
SD7
SD6
SD5
SD4
SD3
SD2
SD1
SD0
-I/O CH RDY
AEN
SA19
SA18
SA17
SA16
SA15
SA14
SA13
SA12
SA11
SA10
SA9
SA8
SA7
SA6
SA5
SA4
SA3
SA2
SA1
SA0

پایه های استفاده شده دراینترفیس مربوطه :

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

شرح چگونگی شناسایی سیستم در عمل:

منظوراز شناسایی سیستم پیدا کردن تابع تبدیل یک سیستم است که ما برای اینکاراز الگوریتم واسطی است بین رایانه وسیستم برای اعمال ورودی به سیستم ISAاستفاده کرده ایم. کارت LMS و ثبت خروجیهای آن در رایانه می باشد.

ورودیهایی را برای سیستم خود در نظر میگیریم C or Matlabابتدا توسط زبان برنامه نویسی می فرستیم اطلاعات در آی سی ۳۷۳ که در ISAبعد هر یک از ورودیهای خود را به پورت به ولتازآنالوگ تبدیل می شود چون این ولتازDAC قرار داده ایم لچ شده وتوسط ISAخروجی آپ امپ قرار داده ایم تا آنرا معکوس کند .DACمنفی می باشد در خروجی که در خروجی آن قرار دارد تبدیل به دیجیتال شدهADCدر مرحله بعد خروجی سیستم توسط و در رایانه ثبت می شود .

واین عمل تا زمانیکه به ازای هریک از ورودی ها خروجی سیستم اندازه گیری ودر حافظه سیستم ذخیره شود ادامه دارد.

بعد از این مرحله یافتن تابع تبدیل سیستم بر عهده زبان برنامه نویسی می باشد که برنامه آن درزیر آورده شده است. C,Matlabبه دو زبان

```
#include <dos.h>
#include <conio.h>
#include <iostream.h>
```

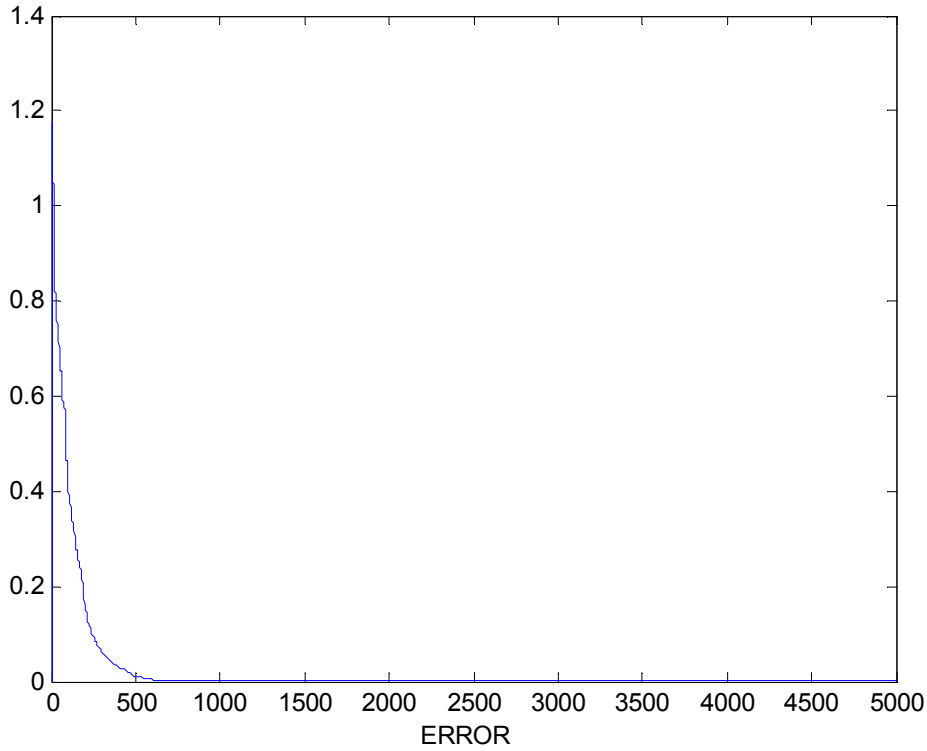
```
#include <stdlib.H>
void main(void)
{
clrscr();
int i,k,a1,n,j,g;
float u=1;
float y[500],X[50],x[500],a[100],Y,e;
for(i=0;i<500;i++)
        {
        x[i]=random(256);
        outp(0x300,x[i]);
        delay(.1);
        outp(0x303,256);  //      writing in adc's enable
        a1=0;
        while(a1&1!=1)
                a1=inp(0x302);
        y[i]=inp(0x301);
        y[i]=y[i]/100;
        cout<<y[i]<<"   ";
        } /*
for(i=1;i<n;i++)
        {a[i]=random(100);a[i]=a[i]/100;}
for(i=n;i<500;i++)
        {
        g=1;
        for(j=i;j>(i-n+1);j--)
                {
                X[g]=x[j]/100;
                g++;}
        Y=0;
        for(j=1;j<n;j++)
```

```
            Y=Y+a[j]*X[j];
            e=y[i]-Y;
            cout<<e<<"  ";
        for(k=1;k<n;k++)
            a[k]=a[k]+X[k]*u*e;
        }
//cout<<endl;
for(i=1;i<n;i++)
        cout<<"a["<<i<<"]"<<a[i]<<endl; */
getch();

            }
```

.....**TEST THE ALGORITM WITH MATLAB** .....

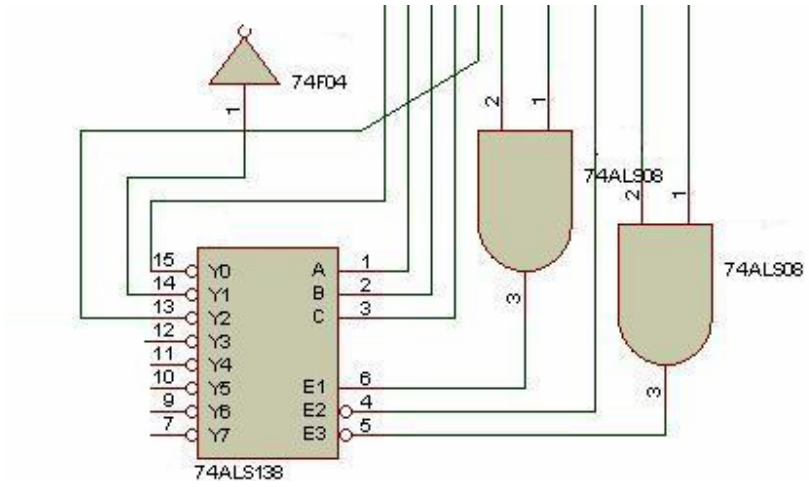
```
N=5000;
x=rand(1,N);
h=rand(1,8);%[1 2 -1 3 5];
d=filter(h,1,x);
figure(1);plot(x)
Order=8;
w=rand(1,Order);
Mu=0.1;%Step Size


for i=Order:length(x)
    X=x(i:-1:i-Order+1);

    y=X*w';
    e(i)=d(i)-y;

    w=w+Mu*X*e(i);
    WeightUpdateError(i)=sqrt(sum((h-w).^2));
```

```
end
warning off
figure(2);plot(20*log(abs(e+eps)));
figure(3);plot(WeightUpdateError);
warning on
```



شرحی بر مدار دیکد:

برای دیکد کردنI/O از آی سی ۱۳۸استفاده شده است پایه های ۱و۲و۳توسط a0,a1,a2آدرس دهی می شـود هر یک از فعال سازهای ۱۳۸ بصورت زیر دیکد شده اند.



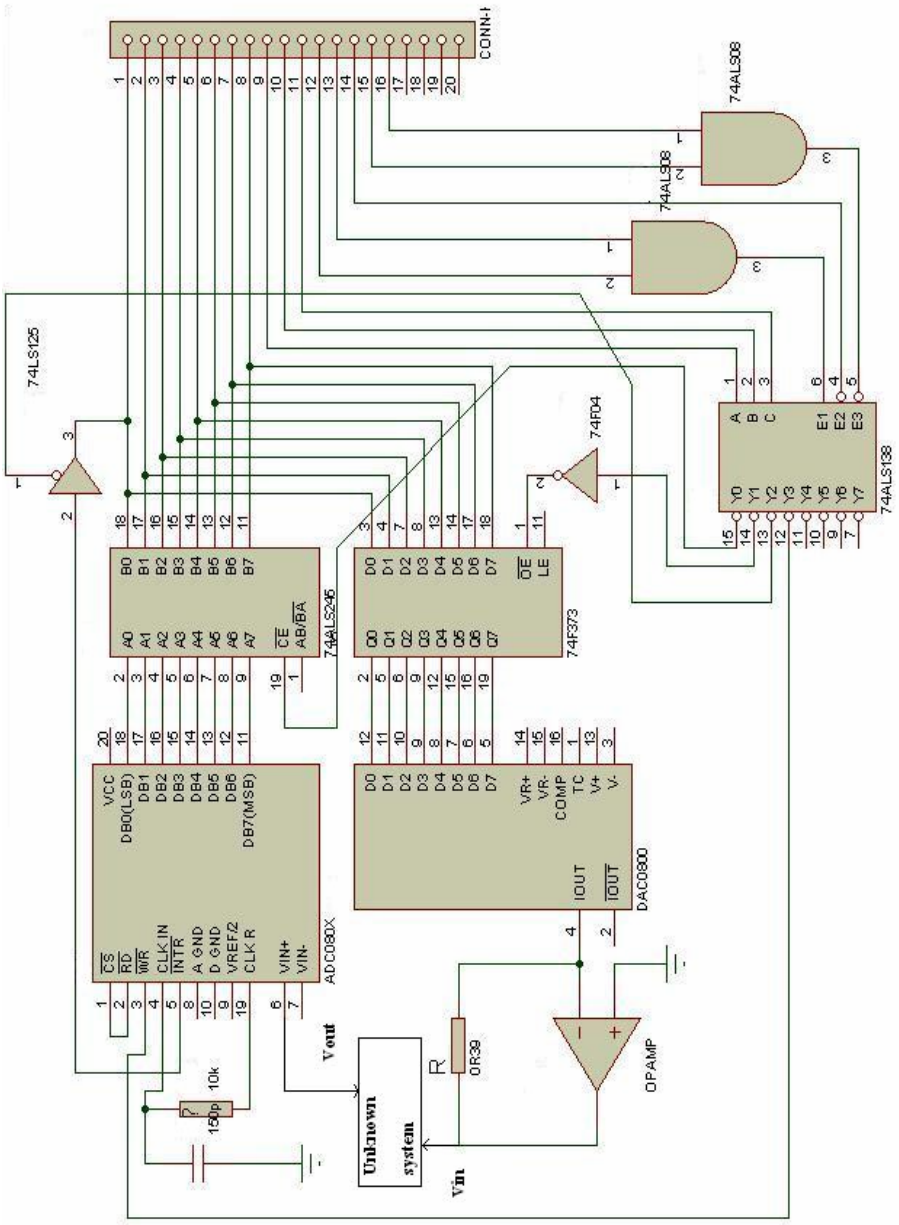
خروجی y0 برای نوشتن اطلاعات در ۳۷۳و خروجی y1برای خواندن اطلاعات توسط کامپیوتردر نظر گرفته شده وخروجیهای y2 y3برای فعال کردن ADCوخواندن interrupt بکار رفته اند که شرح آن داده خواهد شد.

فعال کردن ADCوخواندن interrupt:

خروجی y3 ۱۳۸مستقیم به adc wrمتصل شده است وadc, y2,int به ورودیهای ۱۲۵رفته و خروجی آن به d0متصل شده است که عملکرد این مدار به این صورت می باشد که هر گاه اطلاعاتی در خروجی فرستاده می شود بوسیله برنامه زیر adc wrفعال شده وadcشروع به تبدیل اطلاعات به دیجیتال می کند.

```
outp(0x303,256);  //     writing in adc's enable
        a1=0;
        while(a1&1!=1)
              a1=inp(0x302);
        y[i]=inp(0x301);
```

وصل است مرتبا چک می شود 125که به خروجی D0فعال می شودوY3در این هنگام خروجی نیز فعال می شود و در نتیجهD0خود را فعال می کند و به دنبال آن INTبعد ازاتمام تبدیل adc توسط کا مپیوتر ذخیره می گردد.ADCاطلاعات


CONN-I
74ALS08
74ALS08
74LS125
74HC04
74ALS138
74ALS245
74F373
ADC0800
DAC0800
OPAMP
10k
150p
0R39
R
Vout
Vin
Unknown
system