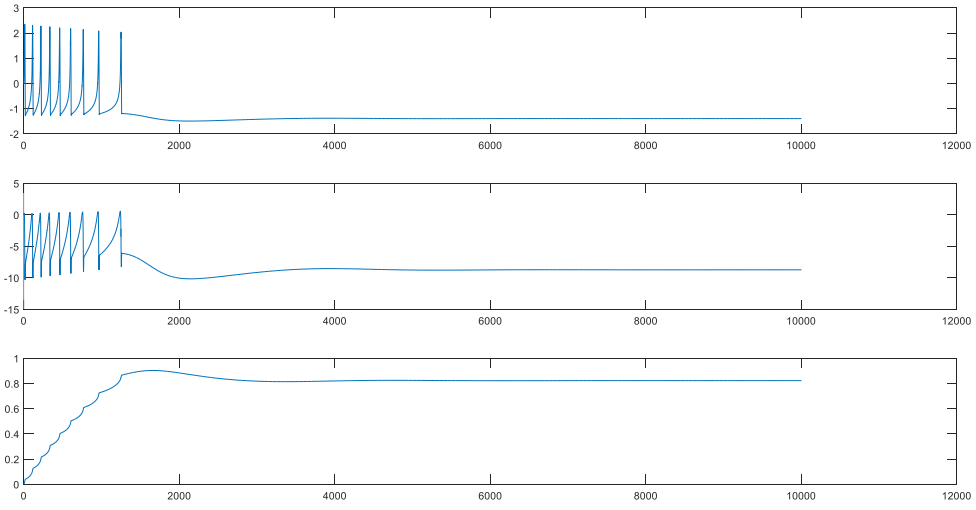
با استفاده از متلب، و کد زیر، x و y و z را میتوان تا ۱۰ هزار تکرار در شکل زیر رسم کرد:

```
clear all; clc; close all;
s1 = 1e-1;
s2 = 1e-1;
s3 = 1e-1;
Isimulous=1;
r=0.0021;
x(1)=0; y(1)=0; z(1)=0;

for n=1:10000;
x(n+1)=(y(n)+x(n)^2*(3-x(n))-z(n)+Isimulous)*s1+x(n);
y(n+1)=(1-5*x(n)^2-y(n))*s2+y(n);
z(n+1)=(r*(4*(x(n)+1.6)-z(n)))*s3+z(n);
end

subplot(311)
plot(x)
subplot(312)
plot(y)
subplot(313)
plot(z)
```

خروجی به این صورت است: (منحنی ها به ترتیب X و y و z هستند)



همانطور که مشخص است، به خاطر بازه y، باید ۵ بیت صحیح در نظر بگیریم.

تعیین تعداد بیت اعشار:

میخواهیم روابط زیر را پیاده سازی کنیم:

$$x[n+1] = (y[n] + x^2[n](3 - x[n]) - z[n] + I_{simulous})s_1 + x[n]$$
$$y[n+1] = (1 - 5x^2[n] - y[n])s_2 + y[n]$$
$$z[n+1] = (r(4(x[n] + 1.6) - z[n]))s_3 + z[n]$$

$S_1 = 1e-1$

$S_2 = 1e-1$

$S_3 = 1e-1$

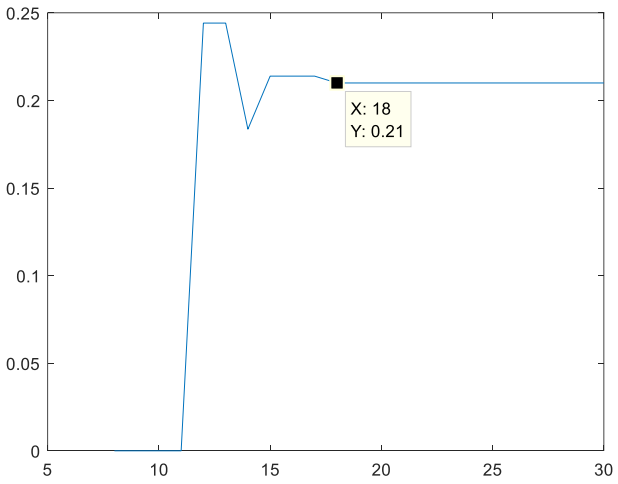
Isimulous=1

r=0.0021

همانطور که با دایره قرمز مشخص کرده ام، در این الگوریتم لازم است عدد 0.0021*0.1=0.00021 **در عددی ضرب شود. اگر دقت اعشار کم باشد، این عدد به کلی کار را خراب میکند. با کد متلب زیر، خطای تولید عدد** 0.00021 **به صورت اعشار ممیز ثابت را برای تعداد بیت اعشار از ۸ تا ۳۰ در شکل زیر رسم کرده ایم.**

```
clear all; close all; clc;

s3 = 1e-1;

r=0.0021;

for i=8:30

a(i-7)=1000*fi(r/10,1,i+5,i);

end

plot(8:30,a)
```

بعد از ۱۹ بیت اعشار، جواب دارای دقت خوبی می شود.



پس در مجموع ۲۴ بیت داریم. الگوریتم های کوردیک را هم با این تعداد بیت هماهنگ کرده ایم.

پیاده سازی ضرب ها:

$$x[n+1] = (y[n] + x^2[n](3 - x[n]) - z[n] + I_{simulous})s_1 + x[n]$$
$$y[n+1] = (1 - 5x^2[n] - y[n])s_2 + y[n]$$
$$z[n+1] = \big(r(4(x[n\,] + 1.6) - z[n])\big)s_3 + z[n]$$

همانطور که مشخص است، در الگوریتم فوق، ضرب در اعداد زیر صورت میگیرد:

3 – 0.1 – 5 – 0.0021 – 4

برای هر کدام، یک ماژول جداگانه طراحی میکنیم. برای پیاده سازی ضرب، از الگوریتم زیر استفاده میکنیم:

Algorithm 2.4: Optimal CSD Coding

1) Starting with the LSB substitute all 1 sequences larger than two with $10\ldots0\overline{1}$. Also substitute 1011 with $110\overline{1}$.
2) Starting with the MSB, substitute $10\overline{1}$ with 011.

این الگوریتم میگوید که میتوان شیفت ها و جمع ها (یا تفریق ها) را بهینه سازی کرد. مثلا اگر بخواهیم عددی را در ۱۵ (معادل 1111) ضرب کنیم، به جای اینکه ضرایب ۱ و ۲ و ۴ و ۸ آن را با هم جمع کنیم (این ضرب ها با شیفت ساخته می شود پس منابع fpga را مصرف نمیکند) بهتر است ابتدا ضریب ۱۶ را ساخته و از ضریب ۱ کم کنیم. در این صورت، منابع کم تری از fpga مصرف می شود.

برای مشخص کردن -1 از عبارت 1bar استفاده می شود. (در الگوریتم فوق توضیح داده شده است چگونه یک عدد باینری را به این صورت بهینه کنیم. یکی که بالایش خط دارد منفی یک است یعنی باید منها شود)

پیاده سازی ضرب در ۰/۱:

0.1= 00000/0001100110011001101

پیاده سازی ضرب در 3:

3= 00011/0000000000000000000

پیاده سازی ضرب در 5:

5= 00101/0000000000000000000

پیاده سازی ضرب در 0.0021:

0.0021= 00000/0000000010001001101

پیاده سازی ضرب در 4:

4= 00100/0000000000000000000

نیازی به استفاده از الگوریتم گفته شده نیست.

الگوریتم کوردیک:

ورودی: ۲۲ بیت با ۱۹ بیت اعشار و ۳ بیت صحیح و ورودی بین -۴ تا ۴

خروجی اول (توان دو): ۲۳ بیت با ۱۹ بیت اعشار و ۴ بیت صحیح و رنج بین -۸ تا ۸

خروجی دوم (توان ۳): ۲۴ بیت با ۱۹ بیت اعشار و ۵ بیت صحیح و رنج بین -۱۶ تا ۱۶

$$x[n+1] = (y[n] + x^2[n](3 - x[n]) - z[n] + I_{simulous})s_1 + x[n]$$
$$y[n+1] = (1 - 5x^2[n] - y[n])s_2 + y[n]$$
$$z[n+1] = \big(r(4(x[n\,] + 1.6) - z[n])\big)s_3 + z[n]$$

در محاسبات، با بلوک کوردیک توان دو و سه را حساب کرده ایم. برای محاسبات عبارت هایلایت شده، ضرب را باز کرده ایم یعنی توان دو را در سه ضرب کرده ایم و سپس منهای توان سه کرده ایم.

برای پیاده سازی اعداد ثابت از constant **استفاده کرده ایم.**

1.6 = 000011001100110011001100110

1 = 000010000000000000000000

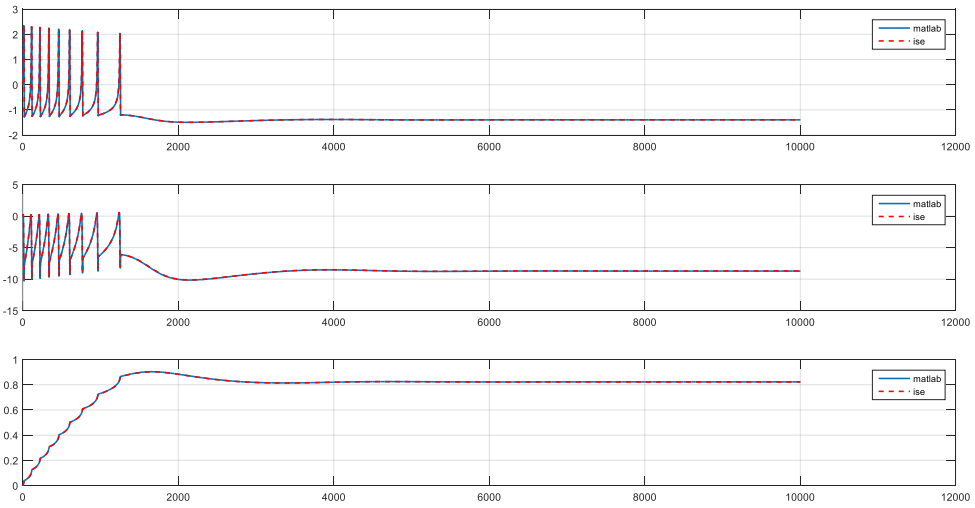
همچنین به جای اینکه 5x^2 **را حساب کنیم و بعد در** 0.1 **ضرب کنیم مستقیم** 0.5x^2 **را حساب کرده ایم که به سادگی با شیفت به دست آمده است.**

کد و تست بنچ در فایل های ارسال شده موجود است. در نهایت، نتایج ise را در فایل های txt ذخیره میکنیم و سپس فایل txt را توسط کد متلب زیر خوانده و با نتایج حاصل از متلب مقایسه میکنیم:

```
clear all; clc; close all;
s1 = 1e-1;
s2 = 1e-1;
s3 = 1e-1;
Isimulous=1;
r=0.0021;
x(1)=0; y(1)=0; z(1)=0;

for n=1:10000;
    a=y(n)+x(n)^2*(3-x(n))-z(n)+Isimulous;
    b=(1-5*x(n)^2-y(n));
    c=(r*(4*(x(n)+1.6)-z(n)));
x(n+1)=a*s1+x(n);
y(n+1)=b*s2+y(n);
z(n+1)=c*s3+z(n);
end

fid=fopen('textout.txt');
A=textscan(fid, '%s', 'delimiter', '\n');
new_x_=A{1};
for i=1:10000
        xx=cell2mat(new_x_(i));
        xx=str2num(xx);
        new_x(i)=xx;
end
new_x=new_x/512/1024;

fid=fopen('teytout.txt');
A=textscan(fid, '%s', 'delimiter', '\n');
new_y_=A{1};
for i=1:10000
        yy=cell2mat(new_y_(i));
        yy=str2num(yy);
        new_y(i)=yy;
end
new_y=new_y/512/1024;


fid=fopen('teztout.txt');
A=textscan(fid, '%s', 'delimiter', '\n');
new_z_=A{1};
for i=1:10000
        zz=cell2mat(new_z_(i));
        zz=str2num(zz);
        new_z(i)=zz;
end
new_z=new_z/512/1024;
```

```
subplot(311)
plot(x,'linewidth',1.5)
hold on
plot(new_x,'r--','linewidth',1.5)
grid on
legend('matlab','ise')
subplot(312)
plot(y,'linewidth',1.5)
hold on
plot(new_y,'r--','linewidth',1.5)
grid on
legend('matlab','ise')
subplot(313)
plot(z,'linewidth',1.5)
hold on
plot(new_z,'r--','linewidth',1.5)
grid on
legend('matlab','ise')
```

خروجی به این صورت است:



نتایج دارای دقت خوبی است.

نتایج سنتز: با ۱۹ بیت اعشار دقت

final Project Status (08/01/2023 - 08:02:36)			
Project File:	cordic.xise	**Parser Errors:**	No Errors
Module Name:	final	**Implementation State:**	Placed and Routed
Target Device:	xc6slx9-3tqg144	• **Errors:**	
Product Version:	ISE 14.7	• **Warnings:**	
Design Goal:	Balanced	• **Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	All Constraints Met
Environment:	System Settings	• **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	515	11,440	4%	
Number used as Flip Flops	217			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	298			
Number of Slice LUTs	1,079	5,720	18%	
Number used as logic	1,063	5,720	18%	
Number using O6 output only	514			
Number using O5 output only	38			
Number using O5 and O6	511			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number used exclusively as route-thrus	16			
Number with same-slice register load	0			
Number with same-slice carry load	16			
Number with other load	0			
Number of occupied Slices	331	1,430	23%	
Number of MUXCYs used	932	2,860	32%	
Number of LUT Flip Flop pairs used	1,135			
Number with an unused Flip Flop	622	1,135	54%	
Number with an unused LUT	56	1,135	4%	
Number of fully used LUT-FF pairs	457	1,135	40%	
Number of unique control sets	15			
Number of slice register sites lost to control set restrictions	63	11,440	1%	
Number of bonded IOBs	75	102	73%	

IOB Flip Flops	73			
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	1	16	6%	
Number used as BUFGs	1			
Number used as BUFGMUX	0			
Number of DCM/DCM_CLKGENs	0	4	0%	
Number of ILOGIC2/ISERDES2s	0	200	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	200	0%	
Number of OLOGIC2/OSERDES2s	73	200	36%	
Number used as OLOGIC2s	73			
Number used as OSERDES2s	0			
Number of BSCANs	0	4	0%	
Number of BUFHs	0	128	0%	
Number of BUFPLLs	0	8	0%	
Number of BUFPLL_MCBs	0	4	0%	
Number of DSP48A1s	0	16	0%	
Number of ICAPs	0	1	0%	
Number of MCBs	0	2	0%	
Number of PCILOGICSEs	0	2	0%	
Number of PLL_ADVs	0	2	0%	
Number of PMVs	0	1	0%	
Number of STARTUPs	0	1	0%	
Number of SUSPEND_SYNCs	0	1	0%	
Average Fanout of Non-Clock Nets	2.95			

Performance Summary			[-]
Final Timing Score:	0 (Setup: 0, Hold: 0)	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Tue Aug 1 08:01:46 2023			
Translation Report	Current	Tue Aug 1 08:01:53 2023	0	0	0
Map Report	Current	Tue Aug 1 08:02:17 2023	0	0	6 Infos (0 new)
Place and Route Report	Current	Tue Aug 1 08:02:30 2023	0	0	3 Infos (0 new)
Power Report					
Post-PAR Static Timing Report	Current	Tue Aug 1 08:02:35 2023	0	0	4 Infos (0 new)
Bitgen Report					

نتایج:

Number of Slice LUTs: 1,079 out of 5,720 18%

Number of Slice Registers: 515 out of 11,440 4%

Number used as Flip Flops: 217

Number used as Latches: 0

Number used as Latch-thrus: 0

Number used as AND/OR logics: 298

Maximum Frequency: 54.462MHz

بدون استفاده از قیود