CORDIC METODU KULLANILARAK TRİGONOMETRİK HESAP MAKİNESİ SİMÜLASYONU

Ahmet SERTBAŞ[1] Selçuk SEVGEN[2]
e-posta: [1]asertbas@istanbul.edu.tr *e-posta: [2]sevgens@istanbul.edu.tr*
[1,2]İstanbul Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34850, Avcılar, İstanbul

Anahtar Kelimeler: Bilgisayar Aritmetiği, CORDIC Algoritması, Trigonometrik Simülatör

ÖZ

Bu çalışmada, CORDIC algoritmaları kullanılarak trigonometrik fonksiyonların değerleri etkin ve hızlı bir şekilde hesaplanması amaçlanmıştır. Bu amaçla, C programlama dilinde trigonometrik bir hesap makinası simülatörü geliştirilmiştir. *Sin, cos, tan, cotan, sinh, cosh, tanh, cotanh* ve *ters fonksiyonlarının* simülasyonları elde edilmiştir.

1. GİRİŞ

CORDIC (**CO**ordinate, **R**otation **DI**gital **C**omputer) sinüs, cosinüs, üstel fonksiyonlar, logaritma gibi çeşitli gerçek zamanlı yüksek prezisyonlu aritmetik hesaplamalar için geliştirilen zeki ve etkin bir algoritma ailesidir. 1959 yılında Volder [1] tarafından ortaya konulan gerçekten önemli bu algoritmalar, n iterasyonda n bit doğrulukla sözkonusu fonksiyonları hesaplayabilmekte ve her iterasyonda az sayıda toplama ve öteleme işlemlerini gerektirmektedir. Algoritma, Walther [2] tarafından genişletilerek bölme, çarpma, karakök hesaplama, hiperbolik ve üstel fonksiyon hesaplama gibi diğer aritmetik işlemleri de yapabilecek yeteneğe kavuşturuldu. Son yıllarda, literatürde CORDIC algoritmalarını kullanarak değişik aritmetik hesaplamalar ve farklı gerçekleme metodlarını içeren çalışmaların arttığı görülmektedir [3-5].

Bilindiği gibi, klasik olarak, trigonometrik hesaplamalar için seriye açılım veya polinom yaklaşım metodları kullanılmaktadır. CORDIC algoritmaları kullanılarak trigonometrik fonksiyonların geliştirilmesi, hızlı ve etkin olarak hesaplama gücünün artmasına neden olmaktadır. Bu çalışmada, trigonometrik fonksiyonların değerlerini hızlı bir şekilde hesaplayabilen bir hesap makinası simülatörü gerçekleştirilmiştir. Bu amaçla, trigonometrik fonksiyonların değerlerini hesaplamak için CORDIC algoritmasının nasıl kullanıldığı üzerinde durulmuştur. Makalenin 2. bölümünde, genel CORDIC algoritması tanıtlanması, 3. bölümde geliştirilen CORDIC simülatörünün tüm fonksiyonları, 4. bölümde ise elde edilen sonuçlar verilecektir.

2. GENEL CORDIC ALGORİTMASI

Verilen bir θ açısı ($-\pi/2 \le \theta \le \pi/2$) için, genelleştirilmiş CORDIC denklemleri aşağıdaki gibi verilebilir[6].

$$x^{(i+1)} = x^{(i)} - \mu d_i\, y^{(i)}\, 2^{-i}$$

$$y^{(i+1)} = y^{(i)} + \mu d_i\, x^{(i)}\, 2^{-i} \qquad (1)$$

$$z^{(i+1)} = z^{(i)} - d_i\, e^{(i)}$$

Burada d_i katsayıları değeri ± 1 olan ve signum fonksiyonu ile tanımlanan katsayılardır.

$$d_i = sgn\,(z^{(i)}) = 1 \quad (z^{(i)} \ge 0), \qquad -1 \quad (z^{(i)} < 0)$$

(1) denklemlerinden de görüleceği gibi, $x^{(i+1)}$ ve $y^{(i+1)}$ hesaplamaları i bit sağa öteleme ve sadece 1 toplama/çıkarma işlemlerini gerektirmektedir. $e^{(i)}$ önceden hesaplanarak bir tablo halinde saklandığı gözönünde tutulursa, her CORDIC iterasyonu 2 öteleme, 3 toplama/çıkarma ve 1 tablo okuma (look-up table) işlemi ile ilgilidir.

Ayrıca, μ parametresi farklı CORDIC hesaplamalarını belirleyen bir sabittir:

$\mu=1$	dairesel açı değişimleri	$e^{(i)}= \tan^{-1} 2^{-i}$
$\mu=0$	doğrusal değişimler	$e^{(i)}= 2^{-i}$
$\mu= -1$	hiperbolik açı değişimleri	$e^{(i)}= \tanh^{-1} 2^{-i}$

Bu çalışmada, özellikle $\mu=1$ seçilerek dairesel CORDIC ve $\mu= -1$ seçilerek hiperbolik CORDIC hesaplamaları yapılmıştır. Böylelikle trigonometrik hesap makinasının sağladığı sinüs (sin), cosinüs (cos), tanjant (tan), cotanjant (cot), sinüs hiperbolik (sinh), cosinüs hiperbolik (cosh), tanjant hiperbolik (tanh) ve cotanjant hiperbolik (coth) fonksiyonları ve ters fonksiyonları CORDIC metoduyla hesaplanmıştır.

♦ Dairesel CORDIC (μ=1) iterasyon sonuclarını karakterize eden denklemler (2) de verilmiştir. Bu denklemler, $z^{(i)}$ açısı sıfıra zorlandığı için rotasyonel mod adı ile anılmaktadır.

$$x^{(m)}= K(x \cos z - y \sin z)$$
$$y^{(m)}= K(y \cos z + x \sin z) \qquad (2)$$
$$z^{(m)} = 0$$

Bu modda, temel kural z'i sıfıra götürecek şekilde $d_i \in (-1,1)$ katsayılarını seçmektir. İterasyona başlangıç değeri ;

$x_0=1/K=\prod[\cos(\alpha^{(i)})] =\prod[1/(1+\tan(\alpha^{(i)})^2)]^{1/2}$, $y=0$

seçilirse, $x^{(m)}=\cos z$, $y^{(m)}=\sin z$ değerlerine yakınsayacaktır (tanz=sinz/cosz, cotanz=1/tanz).

Dairesel CORDIC iterasyonlarını kullanmanın ikinci bir yolu, vektörel mod olarak bilinen ve (2) denklemlerinin farklı bir şekilde yazılması ile elde edilen (3) denklemlerini referans almaktır.

$$x^{(m)}= K(x^2 + y^2)^{1/2}$$
$$y^{(m)}=0 \qquad (3)$$
$$z^{(m)} = z + \tan^{-1} (y/x)$$

Dairesel vektörel mod olarak da isimlendirilen bu yöntemde temel kural, y'i sıfıra götürecek şekilde $d_i \in (-1,1)$ katsayılarını seçmektir.
İterasyona başlangıç değeri $x_0=1$, $z_0=0$ seçilirse, $z^{(m)}=\tan^{-1}(y/x)$ değerine yakınsayacaktır. Algoritmaya giriş koşulu olarak ,

$x_0=w$, $y_0=(1-w^2)^{1/2}$ değerleri girilirse,

$z=\cos^{-1}(w)= \tan^{-1}[(1-w^2)^{1/2}/w]$;

$x_0=(1-w^2)^{1/2}$, $y_0=w$ değerleri girilirse,

$z=\sin^{-1}(w)= \tan^{-1}[w/(1-w^2)^{1/2}]$;

hesaplamaları yaptırılabilir.

♦ Diğer taraftan (μ=-1 seçilerek elde edilen) hiperbolik CORDIC iterasyon sonuclarını karakterize eden denklemler (4)'de görülmektedir. Yöntemde, Z sıfıra zorlandığı için hiperbolik rotasyonel mod olarak da isimlendirilir.

$$x^{(m)}= K'(x \cosh z + y \sinh z)$$
$$y^{(m)}= K(y \cosh z + x \sinh z) \qquad (4)$$
$$z^{(m)} = 0$$

Dairesel rotasyonel moda benzer şekilde, iterasyona başlangıç değeri olarak;

$x_0=1/K'=\prod[\cosh(\alpha^{(i)})] =\prod[1/(1-\tan(\alpha^{(i)})^2)]^{1/2}$, $y=0$

seçilirse, $x^{(m)}=\cosh z$, $y^{(m)}=\sinh z$ değerlerine yakınsayacaktır (tanhz= sinhz/coshz, cothz=1/tanhz).

Ters hiperbolik fonksiyonların hesaplanabilmesi için, hiperbolik vektörel mod kullanılmıştır:

$$x^{(m)}= K(x^2 - y^2)^{1/2}$$
$$y^{(m)}=0 \qquad (5)$$
$$z^{(m)} = z + \tanh^{-1} (y/x)$$

Benzer şekilde, iterasyona başlangıç değeri $x_0=1$, $z_0=0$ seçilirse, $z^{(m)}=\tanh^{-1}(y/x)$ değerine yakınsayacaktır. Algoritmaya giriş koşulu olarak,

$x_0=w+1$, $y_0=w-1$ değerleri için,

$z=(1/2)\ln(w)= \tan^{-1}[(1-w^2)^{1/2}/w]$;

$x_0=w+1+(1-w^2)^{1/2}$, $y_0=w-1+(1-w^2)^{1/2}$ değerleri için,

$z=\cosh^{-1}(w)= \ln(w+(1-w^2)^{1/2})$;

$x_0=w+1+(1+w^2)^{1/2}$, $y_0=w-1+(1+w^2)^{1/2}$ değerleri için,

$z=\sinh^{-1}(w)= \ln(w+(1+w^2)^{1/2})$;

hesaplamaları yaptırılabilir.

Elde edilen tüm CORDIC hesaplamaları toplu olarak Şekil 1 de görülmektedir.

3. TRİGONOMETRİK SİMÜLATÖR

Bu bölümde, *sin, cos, tan, cotan, sinh, cosh, tanh, cotanh* ve *ters fonksiyonlarını* hesaplamak için geliştirilen trigonometrik bir hesap makinası simülatörünün tüm fonksiyonları detaylı olarak incelenmiştir. Simülasyon sonuçları, hesaplama iterasyonlarının açık gösterimleri şeklinde verilmiştir. Simülatör, C programlama dilinde geliştirilmiştir.

Şekil 2'de, simülatörün kullanımı için tasarımlanan kullanıcı arayüzü modeli verilmiştir. (Arcsinh(0.5) hesaplamasının yaptırılması için arayüzün kullanımı görünmektedir / Şekil 2). Tablo 1-4'te, tüm trigonometrik CORDIC hesaplamaları için elde edilen iterasyon adımları verilmiştir. Bu hesaplamalar; 30^0 için temel trigonometrik fonksiyonlar (Tablo1), 0.414210 değeri için trigonometrik ters fonksiyonlar (Tablo 2), 1 değeri için hiberbolik fonksiyonlar (Tablo 3) , 0.5 için ters hiberbolik sinüs ve tanjant fonksiyonları vede 3 için ters hiperbolik kotanjant fonksiyonu (Tablo 4) hesaplamalarından oluşmaktadır.

Tablo 1. Trigonometrik CORDIC hesaplamaları

n	sin(30)	cos(30)	tan(30)	cotan(30)
1	0.607252935009	0.607252935009	1.000000000000	1.000000000000
2	0.303626467504	0.910879402513	0.333333333333	3.000000000000
3	0.531346318133	0.834972785637	0.636363636364	1.571428571429
4	0.426974719928	0.901391075404	0.473684210526	2.111111111111
5	0.483311662141	0.874705155408	0.552542372881	1.809815950920
6	0.510646198247	0.859601665966	0.594049800582	1.683360551624
7	0.497214922217	0.867580512814	0.573105221790	1.744880280234
8	0.503992894973	0.863696021234	0.583530411837	1.713706740410
9	0.500619082390	0.865664743480	0.578305962164	1.729188466703
10	0.498928330938	0.866642515126	0.575702578895	1.737008025774
11	0.499774661519	0.866155280427	0.577003538295	1.733091625323
12	0.500197588902	0.865911249831	0.577654567948	1.731138392193
13	0.499986184788	0.866033368383	0.577329007221	1.732114595825
14	0.500091901752	0.865972334913	0.577491776111	1.731626390827
15	0.500039046996	0.866002858100	0.577410388798	1.731870467524
16	0.500012618686	0.866018118081	0.577369697292	1.731992525224
17	0.499999404298	0.866025747668	0.577349352077	1.732053558912
18	0.500006011550	0.866021932975	0.577359524639	1.732023041665
19	0.500002707939	0.866023840347	0.577354438347	1.732038300188
20	0.500001056129	0.866024794026	0.577351895209	1.732045929525
21	0.500000230224	0.866025270865	0.577350623642	1.732049744212
22	0.499999817270	0.866025509283	0.577349987859	1.732051651561
23	0.500000023747	0.866025390074	0.577350305751	1.732050697886
24	0.499999920509	0.866025449679	0.577350146805	1.732051174723
25	0.499999972128	0.866025419876	0.577350226278	1.732050936305
26	0.499999997938	0.866025404975	0.577350266014	1.732050817095
27	0.499999985033	0.866025412426	0.577350246146	1.732050876700
28	0.499999991485	0.866025408701	0.577350256080	1.732050846898
29	0.499999994711	0.866025406838	0.577350261047	1.732050831996
30	0.499999996324	0.866025405907	0.577350263531	1.732050824546
31	0.499999997131	0.866025405441	0.577350264772	1.732050820821
32	0.499999996728	0.866025405674	0.577350264152	1.732050822683
33	0.499999996929	0.866025405557	0.577350264462	1.732050821752
34	0.499999997030	0.866025405499	0.577350264617	1.732050821286
35	0.499999997081	0.866025405470	0.577350264695	1.732050821053

Tablo3. Hiperbolik CORDIC hesaplamaları

n	sinh(1)	cosh(1)	tanh(1)	cotanh(1)
1	0.603748559952	1.207497119904	0.500000000000	2.000000000000
2	0.905622839928	1.358434200287	0.666666695918	1.499999934184
3	1.075427055359	1.471637010574	0.730769237000	1.368421040963
4	1.167404413223	1.538851261139	0.758620694998	1.318181808899
6	1.167404413223	1.538851261139	0.758620694998	1.318181808899
6	1.213213443756	1.572327017784	0.771603763106	1.296001973829
7	1.188645839691	1.553370594978	0.765204287717	1.306840560164
8	1.176510095596	1.544084310532	0.761946797576	1.312427590984
9	1.170478463173	1.539488554001	0.760303452813	1.315264314926
10	1.173485279083	1.541774630547	0.761126338333	1.313842327661
11	1.174990892410	1.542920589447	0.761536854487	1.313134084199
12	1.175744295120	1.543494343758	0.761741887734	1.312780636201
13	1.175367474556	1.543207287788	0.761639401172	1.312957284590
14	1.175179123878	1.543063759804	0.761588182220	1.313045584669
16	1.175179123878	1.543063759804	0.761588182220	1.313045584669
16	1.175273299217	1.543135404587	0.761613851723	1.313001329661
17	1.175226211548	1.543099522591	0.761601046687	1.313023405561
18	1.175202608109	1.543081641197	0.761594575901	1.313034561488
19	1.175190806389	1.543072700500	0.761591340452	1.313040139620
20	1.175196647644	1.543077230453	0.761592890136	1.313037467854
21	1.175199627876	1.543079495430	0.761593703602	1.313036065386
22	1.175201058388	1.543080568314	0.761594101125	1.313035380032
23	1.175200343132	1.543079972267	0.761593931781	1.313035671990
24	1.175200700760	1.543080210686	0.761594045871	1.313035475292
25	1.175200939178	1.543080329895	0.761594141543	1.313035310348
26	1.175201058388	1.543080449104	0.761594159961	1.313035278594

Tablo 2. Ters Trigonometrik CORDIC hesaplamaları

n	arcsin 0.414210	arccos 0.414210	arctan 0.414210	arccot 0.414210
1	0.000000000000	0.000000000000	0.000000000000	1.570796327000
2	45.000000000000	45.000000000000	45.000000000000	-43.429203673000
3	18.434947967529	71.565048217773	18.434947967529	-16.864151640529
4	32.471191406250	57.528804779053	32.471191406250	-30.900395079250
5	25.346174240112	64.653823852539	25.346174240112	-23.775377913112
6	21.769840240479	68.230155944824	21.769840240479	-20.199043913479
7	23.559751510620	66.440246582031	23.559751510620	-21.988955183620
8	24.454925537109	65.545074462891	22.664577484131	-21.093781157131
9	24.902540206909	65.097457885742	22.664577484131	-20.646166487331
10	24.678730010986	65.321266174316	22.440773010254	-20.869976683254
11	24.566823959351	65.433174133301	22.552679061890	-20.981882734890
12	24.510871887207	65.489128112793	22.496726989746	-20.925930662746
13	24.482894897461	65.517105102539	22.524703979492	-20.953907652492
14	24.468906402588	65.531089782715	22.510715484619	-20.939919157619
15	24.475900650024	65.524093627930	22.503721237183	-20.932924910183
16	24.472404479980	65.527587890625	22.500225067139	-20.929428740139
17	24.470655441284	65.529335021973	22.498476028442	-20.927679701442
18	24.469781875610	65.530212402344	22.499349594116	-20.928553267116
19	24.469345092773	65.530647277832	22.499786376953	-20.928990049953
20	24.469564437866	65.530426025391	22.500005722046	-20.929209395046
21	24.469673156738	65.530319213867	22.499897003174	-20.929100676174
22	24.469617843628	65.530372619629	22.499841690063	-20.929045363063
23	24.469591140747	65.530403137207	22.499814987183	-20.929018660183
24	24.469577789307	65.530418395996	22.499828338623	-20.929032011623
25	24.469585418701	65.530410766602	22.499820709229	-20.929024382229
26	24.469581604004	65.530410766602	22.499824523926	-20.929028196926
27	24.469579696655		22.499822616577	-20.929026289577

Tablo 4. Ters Hiperbolik CORDIC hesaplamaları

n	arcsinh(0.5)	arctanh(0.5)	arccotanh(3)
1	0.000000000000	0.000000000000	0.000000000000
2	1.098612308502	0.549306154251	0.549306154251
3	0.587786674500	0.804718971252	0.293893337250
4	0.336472243071	0.679061770439	0.419550538063
5	0.230817690492	0.616480231285	0.356968969107
6	0.586798548698	0.553898692131	0.294387400150
7	0.524278163910	0.522638499737	0.325647592545
8	0.493025630713	0.538264751434	0.341273874044
9	0.477400302887	0.546077430248	0.349086523056
10	0.485212832689	0.549983680248	0.345180243254
11	0.481306582689	0.548030555248	0.347133368254
12	0.479353457689	0.549007117748	0.346156805754
13	0.480330020189	0.549495398998	0.346645087004
14	0.480818301439	0.549251258373	0.346400946379
15	0.240531221032	0.549373328686	0.346523016691
16	0.481306582689	0.549251258373	0.346645087004
17	0.481184512377	0.549312293530	0.346584051847
18	0.481245547533	0.549281775951	0.346553534269
19	0.481215029955	0.549297034740	0.346568793058
20	0.481199771166	0.549304664135	0.346576422453
21	0.481207400560	0.549308478832	0.346572607756
22	0.481211215258	0.549306571484	0.346574515104
23	0.481213122606	0.549305617809	0.346573561430
24	0.481212168932	0.549306094646	0.346574038267
25	0.481211692095	0.549306333065	0.346573799849
26	0.481211930513	0.549306213856	0.346573680639
27	0.481211811304	0.549306154251	0.346573621035

Tablo 1'de, trigonometrik fonksiyonların yakınsaması 32 iterasyonda (8 dijit doğruluk), Tablo 2-4' te verilen hiperbolik fonksiyon ve ters fonksiyonlarının hesaplanması 26 iterasyonda (6 dijit doğruluk) gerçeklenmiştir. Algoritma, hiperbolik sinüs ve cosinüs fonksiyonlarının hesaplanmasında $|z| < 1.13$ kısıtı ve ters hiperbolik tanjant hesaplamasında ise $|y| < 0.81$ kısıtı ile istenilen doğrulukta yakınsama sağlayabilmektedir. Ters hiperbolik kosinüs fonksiyonu $|z| < 1$ için tanımlı olmadığından bu algoritma ile hesaplanması direk olarak yapılamamıştır, ancak ters hiperbolik sinüs fonksiyonu üzerinden hesaplama yaptırılabilir.

4. SONUÇLAR

Bu çalışmada, trigonometrik fonksiyonların hesaplanmasında klasik yaklaşımlara göre daha etkin ve hızlı hesaplama gücüne sahip bulunan CORDIC Algoritmaları kullanılarak bir trigonometrik hesap makinesi simülatörü gerçekleştirilmiştir. Algoritmanın yakınsama hızının yüksekliği, N iterasyonda sonuçların N-bit doğrulukla elde edilmesini sağlayabilmektedir. CORDIC algoritmaları üstel ve logaritmik fonksiyonların hesaplanmasında da etkin olarak kullanılabilir. Bu nedenle, simülatör diğer fonksiyonları da içerecek şekilde zenginleştirilerek daha yüksek hesaplama yeteneğine kavuşturulmalıdır.

5. KAYNAKLAR

[1] Volder, J.E., 'The CORDIC Trigonometric Computing Technique',IRE Trans. Electronic Computers, Vol.8, September 1959, pp.330-334.

[2] Walther, J.S., 'A Unified Algorithm for Elementary Functions', Proc.Spring Joint Computer Conf., 1971, pp. 379-385.

[3] Duprat, J. and J.M. Muller,'The CORDIC Algorithm: New Results for Fast VLSI Implementation', IEEE Trans. Computers, Vol.42, No.2, 1993, pp.168-178.

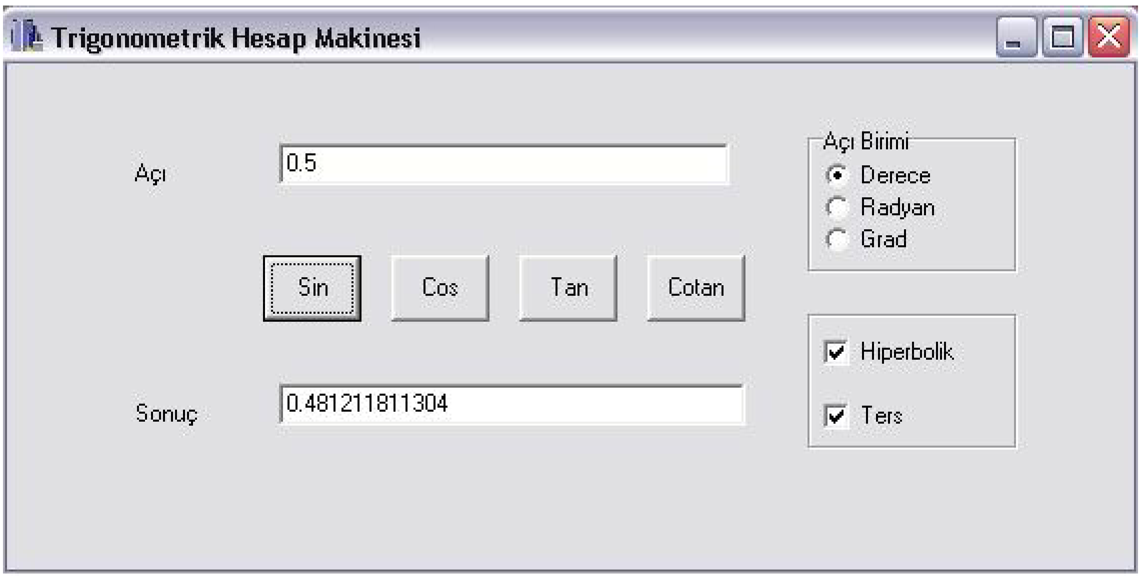
[4] Mazenc, C., X. Merrheim, and J.M. Muller, 'Computing Functions $\cos^{-1}$ and $\sin^{-1}$ Using CORDIC, 'IEEE Trans. Computers, Vol.42, No.1, 1993, pp.118-122.

[5] Phatak, D.S., 'Double Step Branching CORDIC: A New Algorithm for Fast Sive and Cosine Genaration,', IEEE Trans. Computers, Vol. 47, May 1998, pp.587-603.

[6] Parhami B., 'Computer Arithmetic/ Algorithms and Hardware Designs', Oxford University Press, 2000.

	Rotasyonel Mod	Vektörel Mod
DAİRESEL **CORDIC** $\mu=1$	**İlk koşullar:** $x_0=1/K$, $y=0$ $x, y, z \rightarrow$ CORDIC $\rightarrow$ $K(x\cos z - y\sin z)$, $K(y\cos z + x\sin z)$, 0 **x= cosz, y=sinz, tanz=y/x, cotz=x/y**	**İlk koşullar:** $x_0=1$, $z=0x$ $x, y, z \rightarrow$ CORDIC $\rightarrow$ $\mathbf{K(x^2+y^2)^{1/2}}$, $\mathbf{0}$, $\mathbf{z+\tan^{-1}(y/x)}$ $\mathbf{\cos^{-1}w=\tan^{-1}[(1-w^2)/w]^{1/2}}$ $\mathbf{\sin^{-1}w=\tan^{-1}[w/(1-w^2)]^{1/2}}$
HİPERBOLİK **CORDIC** $\mu=-1$	**İlk koşullar:** $x_0=1/K'$, $y=0$ $x, y, z \rightarrow$ CORDIC $\rightarrow$ $K'(x\cosh z - y\sinh z)$, $K'(y\cosh z + x\sinh z)$, 0 **x= coshz, y=sinhz, tanhz=y/x, cothz=x/y**	**İlk koşullar:** $x_0=1$, $z=0$ $x, y, z \rightarrow$ CORDIC $\rightarrow$ $\mathbf{K'(x^2-y^2)^{1/2}}$, $\mathbf{0}$, $\mathbf{z+\tanh^{-1}(y/x)}$ $\mathbf{\ln(w)= 2\tan^{-1}[(1-w^2)^{1/2}/w];}$ $\mathbf{\sinh^{-1}(w)= \ln(w+(1+w^2)^{1/2});}$ $\mathbf{\cosh^{-1}(w)= \ln(w+(1-w^2)^{1/2});}$

Şekil 1. Trigonometrik CORDIC Hesaplamaları



Şekil 2. Trigonometrik Hesap Makinası Kullanıcı Arayüzü