التنفيذ المادي باستخدام FPGA لخوارزميتي كوردك وجدول المقارنة لحساب الدوال الرياضية الأولية

بلال شكر محمود إحسان عبد الستار علي

قسم هندسة الحاسبات / جامعة الموصل

خلاصـــــــة

تم في هذا البحث تصميم وتنفيذ وحدة حسابية مادية تقوم بحساب العديد من الدوال الرياضية الأولية (الجيب والجيب تمام ودالة معكوس الظل ودالتي جيب وجيب تمام القطع الزائد وكذلك الجذر التربيعي) والتي يتطلب حسابها باستخدام الأنظمة البرمجية الآلاف من نبضات الساعة كزمن تنفيذ. إن معمارية حساب كل دالة تم تصميمها باستخدام لغة وصف الكيان المادي "VHDL" وتركيبها على رقاقة XC3S500E FPGA في لوحة سبارتان 3E كتقنية هدف. في هذا البحث تم استخدام خوارزميتين لحساب الدوال الرياضية لملاءمة تنفيذهما باستخدام رقاقة (FPGA) الأولى خوارزمية الحاسب الرقمي لدوران الإحداثيات (CORDIC) التي ظهرت في عام 1959 ، والتي بإمكانها القيام بكل من عملية الضرب والقسمة وحل الدوال المثلثية والقطوع الزائدة واللوغاريتمات والجذور التربيعية، والثانية طريقة جدول المقارنة (Lookup Table). بسبب الاستفادة من خاصية التناظر في الدوال المثلثية واستخدام تقنية الأنابيب المتوازية في خوارزمية كوردك فقد تم الحصول على نسبة تسريع تتراوح بين (24.7 إلى 30.3)×100% بالمقارنة مع المعماريات المتوازية الأخرى. وكانت نسبة العطاء عملية واحدة \ لكل نبضة ساعة فيما عدا العملية الأولى التي تستغرق 32 نبضة.

FPGA Implementation Of Elementary Function Evaluation Unit Using CORDIC and Lookup tables

Basil Sh. Mahmood **Ehsan A. Ali**

Department Of Computer Engineering / University of Mosul

Abstract

In this paper, a hardware computing unit has been designed and implemented. This unit computes many elementary functions (such as sine, cosine, $\tan^{-1}$, sinh, cosh, and square root) that their computing by using software systems requires thousands of clock cycles as an execution time. The architecture of the function computation has been designed by using VHDL and placed on XC3S500E FPGA chip in Spartan 3E as a target technique. In this paper, two algorithms have been used in computing the mathematical functions, because they can be implemented using FPGA chip. The first is the Coordinate Rotation Digital Computer algorithm (CORDIC) which was introduced in 1959. It is a single unified algorithm for calculating many elementary functions including trigonometric, hyperbolic, logarithmic and exponential functions, multiplication, division and square root. The second one uses the lookup table. According to the self-similarity in the trigonometric functions, and using the techniques of parallel pipelining for the CORDIC algorithm, speedup of (24.7 - 30.3)×100% is obtained as compared with the other parallel architectures. The throughput became operation/clock pulse except the first operation whose latency was 32 clock pulse.

Keywords: CORDIC, lookup table, Elementary Function, FPGA

أستلم: **22 12 2009** قبل: **4 4** 2010

1. المقدمة:

إن حساب الدوال الرياضية القابلة للاشتقاق تشكل في أغلب الأحيان عنق الزجاجة (bottle neck) لأداء العديد من التطبيقات المحددة الحساب (compute-bound applications) لأنها تتطلب عمليات النقطة العائمة أو الثابتة المعقدة. وتتضمن هذه الدوال :الدوال الأولية (Elementary functions) مثل الدوال المثلثية (*tan(x), sin(x), cos(x)*) ودوال القطوع الزائدة (*tanh(x), sinh(x), cosh(x)*) ومعكوسات هذه الدوال ودالة الجذر التربيعي واللوغاريتم وكذلك الدوال المركبة مثل $(1-\sin^2(x))^{1/2}$ أو $(\tan^2(x)+1)$. وتعرف الدالة بأنها أولية إذا تم تركيبها أو بناؤها من مجموعة محددة من الدوال الثابتة (مثل e^n، x^n، $log_n(x)$ ومعكوسات هذه الدوال)[1] [2].

تعتمد العديد من التطبيقات الحالية والمستقبلية على دقة حساب الدوال الأولية ولاسيما الدوال المثلثية مثل الجيب والجيب تمام. ومن هذه التطبيقات حواسيب الجيب وتطبيقات الرسم باستخدام الحاسوب والاتصالات الرقمية والمرشحات التكيفية والتي تتطلب بشكل متكرر حساب الدوال المثلثية خلال سياق العمل. إن حساب القيم المثلثية عملية معقدة وتستغرق وقتا طويلا وغالبا ما يكون زمن الحساب للحصول على القيم المثلثية مهيمناً على زمن التنفيذ للخوارزمية. إن مستوى أداء أي نظام يُمكن أَن يضعف إذا كان تتابع الحسابات في النظام يحتاج إلى قيمة مرجعة من حساب دالة مثلثية فإذا تطلب ذلك، عندئذ سوف لن يكون النظام قادراً على المضي حتى يكتمل حساب الدالة المثلثية. لقد استخدمت عدة خوارزميات لحساب هذه الدوال برمجياً وتنفيذها مادياً، حيث توجد العديد من الخوارزميات المادية الكفوءة لحساب هذه الدوال إلا أنها لم تستخدم بشكل عام خلال الربع الأخير من القرن الماضي بسبب الاعتماد على الأنظمة البرمجية التي تستغرق الآلاف من نبضات الساعة كزمن تنفيذ في حساب مثل تلك الدوال الرياضية[1].

تعد مصفوفة البوابات القابلة للبرمجة حقلياً (FPGA) من الماديات القابلة لإعادة التهيئة حيث إنها تبدي مرونة تصميم عالية للبرامجيات وبزمن أداء يقترب من زمن أداء رقاقة (ASIC). وبسبب الكثافات القليلة لرقائق (FPGAs) البدائية فإنها كانت تستخدم بشكل رئيس للتطبيقات التي ليس فيها مطلب حسابي أو حسابات معقدة، إلا أنه مع التقدم السريع الحاصل في التقنية أصبحت (FPGAs) تحتوي على مصادر مطمورة (مثل الضارب المادي وكتل ذاكرة الوصول العشوائي Block RAMs) أكثر بكثير من تلك البدائية[3]. وهكذا استخدمت (FPGAs) في تسريع تطبيقات علمية مختلفة وحققت أداءً متفوقاً قياساً إلى معالجات الغرض العام((GPP) General-purpose processors)[4][5]. في التطبيقات العامة للدوال المثلثية، تستخدم متسلسلة تايلر في تقريب وحساب دوال الجيب والجيب تمام والظل. إلا إن هذه الخوارزمية ملائمة للتنفيذ باستخدام البرمجيات أكثر من الماديات كونها من الخوارزميات التكرارية التي تبنى باستخدام دوائر الجمع والضرب والقسمة[6].

هنالك العديد من البحوث والأعمال السابقة المتعلقة بمجمل الأفكار الأساسية التي تناولها البحث من الخوارزميات العلمية المستخدمة في حساب الدوال الرياضية المعقدة وتنفيذها باستخدام المكونات المادية مثل الـ FPGA. وفيما يأتي استعراضاً لمجموعة منها:

في العام 1959 اقترح الباحث J. E. Volder [7] خوارزمية الحاسب الرقمي لدوران الإحداثيات CORDIC (Coordinate Rotation Digital Computer) لتسهيل الحسابات العددية للدوال المثلثية لكونها تقتصر على عمليات الإزاحة والجمع فقط. هذه الخوارزمية تصف كيف أن دوران المتجه الممثل في المعادلات (1) يمكن أن ينفذ بعدد محدد من الدورات الأولية التي تسمى بالدورات الدقيقة (micro rotations).

$$x_{i+1} = x_i - \mu_i y_i 2^{-i}$$
$$y_{i+1} = y_i + \mu_i x_i 2^{-i} \quad \dots (1)$$
$$\theta_{i+1} = \theta_i - \mu_i a\tan 2^{-i}$$

حيث إن i : مؤشر التكرار. **x_i ، y_i : هما مركبتا المتجه المدور عند التكرار i.**

θ_i: زاوية الدوران عند التكرار i. **μ_i : إشارة زاوية الدوران عند التكرار i.**

وفي عام 1971 قام الباحث J. S. Walter [8] بتعميم خوارزمية CORDIC لتشمل حساب معظم الدوال الرياضية المعقدة مثل دوال القطوع الزائدة واللوغاريتم وكذلك عمليات الضرب والقسمة بعد أن كان مقتصراً على حساب الدوال المثلثية فقط. في عام 1989 قام الباحث H. A. Nienhaus [9] باقتراح معمارية لحساب الجذر التربيعي تجمع بين تقنية جدول المقارنة غير الكفوءة من ناحية الماديات وتقنيات الخوارزمية (Algorithmic techniques) غير الكفوءة من ناحية السرعة وذلك كحل وسط بين السرعة والتعقيد المادي. وفي العام 1998 قام الباحث R. Andraka [10] بدراسة عامة لخوارزمية CORDIC وما يشابهها أو يماثلها من الخوارزميات من خلال معاينة تنفيذها على رقائق الـــ(FPGAs). في العام 1999 قام الباحثان T. Vladimirova وH. Tiggeler [11] بدراسة مخططات خوارزمية CORDIC الملائمة للتنفيذ على رقائق الـــFPGA وذلك للحصول على مساحة سليكون قليلة وسرعة حسابات كفوءة لتوليد دالتي الجيب والجيب تمام. وفي عام 1999 أيضاً اقترح الباحث John N. Lygouras [12] طريقة جديدة استخدم فيها تصميماً مادياً للحصول على قيم الدوال المثلثية (الجيب والجيب تمام) وبدقة عالية، هذا التصميم يجمع بين استخدام الذاكرة والدوائر المنطقية كدوائر معالجة مسبقة. تقوم هذه الطريقة على الاستفادة من خاصية تناظر هذه الدوال حول المحورين y, x فضلاً عن إمكانية الحصول على الجيب تمام من دالة الجيب وذلك بتزحيفها بمقدار $\pi/2$ باتجاه عكسي. كما استخدم الاستكمال الخطي (linear interpolation) للحصول على دقة أعلى دون أن يزيد من حجم الذاكرة المستخدمة كجدول مقارنة.

في العام 2001 قام مجموعة من الباحثين Kharrat M. et al [13] باقتراح طريقة جديدة لتنفيذ خوارزمية CORDIC بشكل امثل، وعند مقارنتها مع الطريقة التقليدية كانت مساحة السليكون المستخدمة من رقاقة (FPGA) أقل وذات دقة أكثر في حساب الدوال المثلثية، وكذلك قللت من زمن الكُمون(latency time). إن الطريقة المقترحة تقوم على أساس خاصية تناظر الدوال المثلثية وتحديد فترة خوارزمية CORDIC بين $[\pi/2,0]$ بدلا من الفترة $[-\pi/2, \pi/2]$. أما في سنة 2003 فقد قام الباحثان S. Ravichandran و V. Asari [14] باقتراح طريقة جديدة تقوم على أساس حسابات مسبقة للبتات المدورة في خوارزمية CORDIC ذات الاتجاه الواحد، حيث إن الخوارزمية ذات الاتجاه الواحد تختلف عن الخوارزمية التقليدية في درجة الدوران. إن التصميم المقترح تم تنفيذه على الـــFPGA ومن خلال النتائج التجريبية لحساب قيم الدوال المثلثية ودوال القطوع المخروطية توصلا إلى أن الطريقة المقترحة قد حافظت على درجة الدقة، فضلا عن ذلك انه كلما قل عدد التكرارات (iterations) مع تعقيد أقل للدائرة فان سرعتها تزداد، ويمكن استخدامها في مختلف تطبيقات معالجة الإشارة الرقمية أو الشبكات العصبية التي تتطلب حسابات رياضية مكثفة. وفي العام 2005 قام مجموعة من الباحثين J. Valls et al [15] باقتراح معمارية CORDIC عامة للمحورين الدائري والخطي مشتقة من معماريات CORDIC ذوات المحور الواحد وتنفيذها بشكل كفوء على رقاقة (FPGA). وفي العام 2006 قام مجموعة من الباحثين R. Kirner et al [6] بدراسة خوارزميات وطرائق حساب الدوال المثلثية لتحليل متطلبات المصادر(resource demands) وتم ذلك باستخدام خوارزميات الحساب التكرارية مثل

متسلسلة تايلر وخوارزمية CORDIC ومقارنتها مع خوارزميات التنفيذ الأخرى مثل جدول المقارنة. وفي العام 2006 أيضا قام الباحثان Cheng-Yuan Lin وChuen-Yau Chen [16] باقتراح معمارية عالية الدقة لتنفيذ خوارزمية CORDIC وذلك بأخذ مزيد من التناظر لحساب الدوال المثلثية للمدى الكلي [0-2π] باستخدام مدى اصغر[0-π/8] بدلاً من المدى[0-π/4]. هذا التقليص في مدى دالتي الجيب والجيب تمام أدى إلى تقليل بنحو 50% من حجم الذاكرة المستخدمة كجدول مقارنة وتحسين دقة خوارزمية CORDIC بمقدار(1-bit). وفي العام2007 قام الباحثانD. R. Llamocca-Obregón وC.P. Agurto-Ríos [17] باستخدام تمثيل الفاصلة الثابتة في تنفيذ ثلاث معماريات لخوارزمية CORDIC لحساب دوال القطوع الزائدة المعدلة من قبل مجموعة من الباحثين X. Hue et al [18] (حيث قاموا في عام 1991 باقتراح تعديل على خوارزمية CORDIC الأصلية وذلك لغرض توسيع المدى المحدد للإدخال للخوارزمية الأصلية). إن المعماريات المنفذة هي:(1. نمط المعمارية التكرارية ذات الكلفة القليلة 2. نمط المعمارية بأسلوب خطوط الأنابيب 3. نمط المعمارية التكرارية ذات البتات المتسلسلة) هذه المعماريات تم وصفها بلغة (VHDL) وتنفيذها على تقنية (Stratix FPGA).

في هذا البحث تم تصميم واقتراح عدد من المعماريات المادية لتنفيذ وحساب بعض الدوال الرياضية الأولية (مثل الدوال المثلثية ودوال القطع الزائد والجذر التربيعي). اذ تم استخدام خوارزميتي CORDIC وجدول المقارنة في تنفيذ المعماريات المقترحة. إن كل دالة من هذه الدوال تمت برمجتها باستخدام لغة وصف الكيان المادي (VHDL) كأداة تصميم وتركيبها على رقاقة XC3S500E FPGA في لوحة سبارتان 3E كتقنية هدف.

2. خوارزمية كوردك ((Coordinate Rotation Digital Computer (CORDIC)):

ظهرت خوارزمية الحاسب الرقمي لدوران الإحداثيات CORDIC في عام 1959 من قبل العالم J. Volder [7]، وبعد ذلك عممت ووحدت من قبل العالم Walter في عام 1971 [8]، والتي بإمكانها القيام بكل من عملية الضرب والقسمة وحل الدوال المثلثية والقطوع الزائدة واللوغاريتمات والجذور التربيعية. ولذلك فان CORDIC تستخدم في العديد من التطبيقات العملية مثل الاذرعة المفصلية الآلية ومعالجة الإشارات الرقمية وحساب المصفوفات.

تتميز خوارزمية CORDIC بأنها بسيطة وكفوءة، وبصورة عامة فهي تستخدم عند عدم توفر الضارب المادي حيث إنها تتطلب فقط عمليات جمع وإزاحة مع جدول مقارنة صغير. لذلك تم تفضيلها على تقنيات التقريب متعددة الحدود الأخرى التي تستخدم الضاربات المادية المكلفة جدا. لقد وجدت هذه الخوارزمية الطريق إلى التطبيقات العملية عندما استخدمت في حواسيب الجيب (Pocket Calculators) مثل (Hewlett Packard's HP 35) ومساعدات المعالج الرياضية مثل (Intel 8087)[19] .

1.2 خوارزمية CORDIC الموحدة (The Unified CORDIC Algorithm)[20]:

لقد قام العالم Walter[8] بتوحيد الأنماط المختلفة لخوارزمية CORDIC إلى مجموعة واحدة من المعادلات ((2)و(3)و(4)):

$$R = (x^2 + my^2)^{1/2} \quad \ldots (2)$$

$$\phi = m^{-1/2} \tan^{-1}(m^{1/2} y / x) \quad \ldots (3)$$

$$m=\begin{cases}1 & \text{للدائرة}\\ 0 & \text{للخط المستقيم}\\ -1 & \text{للقطع الزائد}\end{cases} \qquad \dots (4)$$

حيث (R) شعاع المتجه و(ϕ) زاويته مع المحور الموجب لـــ x.

I. النمط الدائري (m=1) Circular Mode:

تتحرك نقطة النهاية للمتجه في هذا النمط على الدائرة الموصوفة في المعادلة الآتية:

$$x^2 + y^2 = R^2 \qquad \dots (5)$$

وكما في الشكل (1-a) حيث إن المركبتين(x,y) للمتجه R يتم حسابهما من المعادلتين الآتيتين:

$$x = R\cos\phi \qquad \dots(6)$$

$$y = R\sin\phi \qquad \dots(7)$$

وعند تدوير المتجه بزاوية θ عكس عقرب الساعة، فان قيم (x,y) الجديدة تصبح:

$$x' = R\cos(\phi+\theta) = x\cos\theta - y\sin\theta \qquad \dots(8)$$

$$y' = R\sin(\phi+\theta) = y\cos\theta + x\sin\theta \qquad \dots(9)$$

وبقسمة المعادلتين (8) و(9) على $\cos\theta$ نحصل على:

$$x'' = x - y\tan\theta \qquad \dots (10)$$

$$y'' = y + x\tan\theta \qquad \dots (11)$$

حيث إن

$$x'' = x'/\cos\theta \quad and \quad y'' = y'/\cos\theta$$

وباختيار $\tan\theta = 2^{-i}$ نحصل على مركبتي المتجه المدور من خلال عمليات بسيطة (تزحيف وجمع). وتمثل المعادلتان (10) و(11) معادلتي CORDIC الأساسيتين في النمط الدائري.

II. النمط الخطي (m=0) Linear Mode:

تتحرك نقطة النهاية للمتجه في هذا النمط على الخط الذي معادلته :

$$x = R \qquad \dots (12)$$

وكما في الشكل (1-b) حيث إن المركبتين (x,y) للمتجه R تحسبان من المعادلتين (13) و(14) الآتيتين:

$$x = R \qquad \dots (13)$$

$$y = \phi R \qquad \dots (14)$$

حيث $\phi = y/x$، إذ إنه في النمط الخطي تعرف قيمة ϕ على أنها نسبة y إلى x. وإن تدوير المتجه في هذا النمط بزاوية θ مكافئ لتغيير ϕ إلى $\phi+\theta$، وقيم (x,y) الجديدة تصبح:

$$x' = R \qquad \dots(15)$$

$$y' = (\phi+\theta)R = y + \theta x \qquad \dots(16)$$

وباختيار $\theta = 2^{-i}$ نحصل على مركبتي المتجه المدور من خلال عمليات بسيطة (تزحيف وجمع). وتمثل المعادلتان (15) و (16) معادلتي CORDIC الأساسيتين في النمط الخطي.

III. نمط القطع الزائد (m=-1) Hyperbolic Mode:

تتحرك نقطة النهاية للمتجه في هذا النمط على القطع الزائد الذي معادلته:

$$x^2 - y^2 = R^2 \quad \ldots(17)$$

وكما في الشكل (1-c) حيث إن المركبتين (x,y) للمتجه R تحسبان من المعادلتين (18) و(19) الآتيتين:

$$x = R\cosh\phi \quad \ldots(18)$$

$$y = R\sinh\phi \quad \ldots(19)$$

إن تدوير المتجه بزاوية θ في هذا النمط مكافئ لتغيير ϕ إلى $\phi + \theta$، وقيم (x,y) الجديدة تصبح:

$$x' = R\cosh(\phi+\theta) = x\cosh\theta + y\sinh\theta \quad \ldots(20)$$

$$y' = R\sinh(\phi+\theta) = y\cosh\theta + x\sinh\theta \quad \ldots(21)$$

وبقسمة المعادلتين (20) و (21) على $\cosh\theta$ نحصل على:

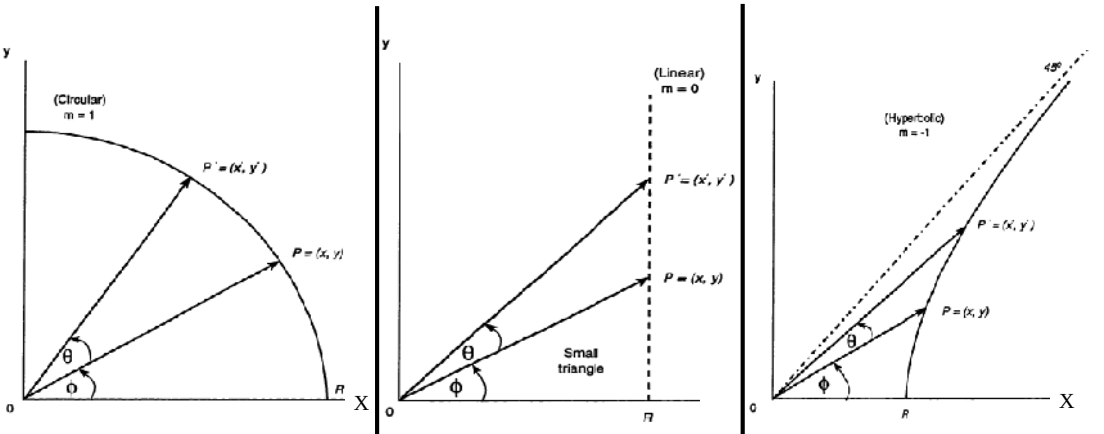
$$x'' = x + y\tanh\theta \quad \ldots(22)$$

$$y'' = y + x\tanh\theta \quad \ldots(23)$$

حيث إن

$$x'' = x'/\cosh\theta \quad and \quad y'' = y'/\cosh\theta$$

وباختيار $\tanh\theta = 2^{-i}$ نحصل على مركبتي المتجه المدور من خلال عمليات تزحيف وجمع فقط. وتمثل المعادلتان (22) و (23) معادلتي CORDIC الأساسيتين في نمط القطع الزائد.



الشكل (1-a) يوضح دوران متجه بزاوية θ في النمط الدائري

الشكل (1-b) يوضح دوران متجه بزاوية θ في النمط الخطي

الشكل (1-c) دوران متجه بزاوية θ في النمط القطع الزائد

الشكل (1) يوضح الأنماط الثلاثة لخوارزمية كوردك

2.2 تكرارات خوارزمية CORDIC (CORDIC Iterations) [20]:

في خوارزمية CORDIC تتحلل زاوية الدوران إلى مجموعة من الزوايا الأولية بغض النظر عن نمط عمل الخوارزمية، بحيث إن عمليات التكرار للخوارزمية تصبح مجرد عمليات تزحيف وجمع فقط. وهكذا في أي تكرار (i) فان قيم المركبتين (x, y) ستتجدد طبقا للمعادلات (24)و(25)و(26) الآتية:

$$x_{i+1} = x_i + m\,\mu_i y_i \delta_i \quad \ldots (24)$$

$$y_{i+1} = y_i - \mu_i x_i \delta_i \quad \ldots (25)$$

$$z_{i+1} = z_i + \mu_i \theta_i \quad \ldots (26)$$

حيث إن

$$i = 0, 1, \cdots, n-1 \qquad n \text{ is the number of bits in the mantisa} \quad \ldots (27)$$

$$\delta_i = 2^{-i} \quad \ldots (28)$$

$$\theta_i = \begin{cases} \tan^{-1}\delta_i & m = 1 \\ \delta_i & m = 0 \\ \tanh^{-1}\delta_i & m = -1 \end{cases} \quad \ldots (29)$$

$$\mu_i = \begin{cases} 1 & \text{دوران المتجه باتجاه عقرب الساعة} \\ -1 & \text{دوران المتجه بعكس اتجاه عقرب الساعة} \end{cases} \quad \ldots (30)$$

إن اختيار قيمة μ_i التي تحدد اتجاه الدوران يعتمد على نمط عمل خوارزمية CORDIC طبقاً للجدول(1).

الجدول (1) تعليمات اختيار اتجاه الدوران.

قيمة μ_i	نمط الدوران	النمط الاتجاهي
+1	$z_i < 0$	$y_i \geq 0$
-1	$z_i \geq 0$	$y_i < 0$

وبعد n من التكرارات، فان التغيّر الكلي في الزاوية هو مجموع التغييرات التراكمية:

$$\theta = \sum_{i=0}^{n-1} \mu_i \theta_i \quad \ldots (31)$$

حيث إن

$$\theta_i = m^{-1/2} \tan^{-1}(m^{1/2}\delta_i) \quad \ldots (32)$$

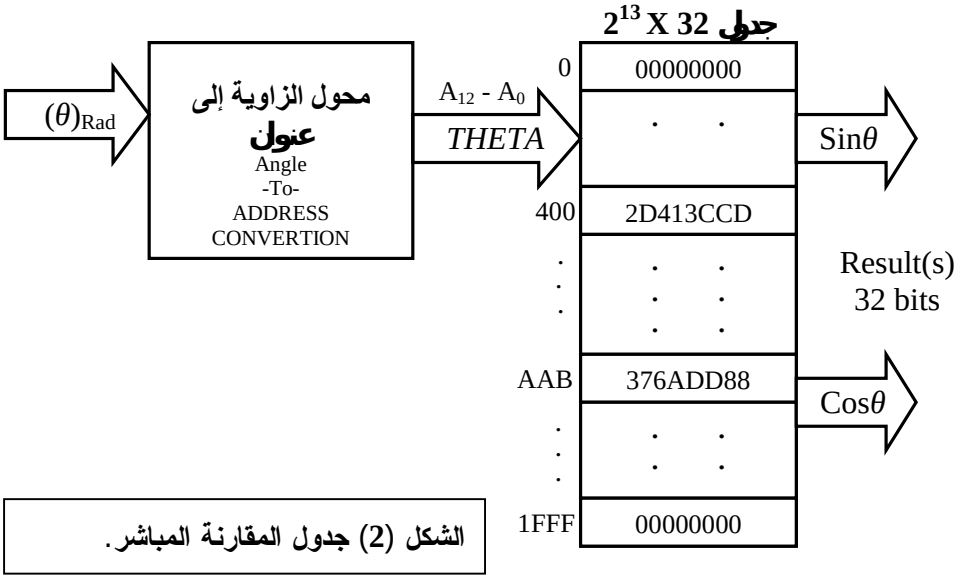
تتراكم تغييرات الزاوية في كل تكرار (i) في المتغير z_i وكما موضح في المعادلة (26).

3. المعماريات المقترحة لحساب الدوال الرياضية:

تناول هذا البحث اقتراح وتنفيذ عدة تصاميم منطقية لمعماريات تقوم بحساب العديد من الدوال الرياضية الأولية وذلك باستخدام خوارزميتي كوردك CORDIC وجداول المعاينة أو المقارنة Lookup Tables. أذ تم تنفيذ تصميمين مقترحين باستخدام جدول المقارنة لحساب دالتي الجيب والجيب تمام، حيث تم اختزال حجم الذاكرة المستخدمة فيهما إلى الربع فيما إذا تم تنفيذها بأسلوب جدول المعاينة المباشر. كذلك تم تنفيذ عدة تصاميم مادية لحساب دوال رياضية (مثل الجيب والجيب تمام ودالة معكوس الظل ودالتي جيب وجيب تمام القطع الزائد) باستخدام خوارزمية كوردك. وفيما يأتي عرضاً لتنفيذ الطرائق المذكورة آنفا مع نتائجها.

1.3 طريقة جدول المعاينة المباشر (Direct Lookup Table Method):

من أسهل الطرائق التي تم استخدامها لتصميم وتنفيذ دالتي الجيب والجيب تمام هي طريقة جدول المقارنة المباشر. إذ إنها لا تنفذ أية حسابات وإنما يتم خزن النتائج بشكل مباشر في الجدول. فقد تم إنشاء جدول مقارنة بحجم (32×2^{13}) باستخدام 15 من الذاكرات المطمورة (BRAM) في رقاقة الـــFPGA. كل موقع يحوي 32 بتاً كناتج محسوب مسبقا لكل قيمة من قيم الزاوية (θ) التي تحتاج إلى 13 بتاً لتمثيلها. وكما موضح في الشكل (2).



الشكل (2) جدول المقارنة المباشر.

المعادلة (32) تقوم بتحويل قيمة الزاوية النصف قطرية من تمثيل الفارزة الثابتة إلى قيمة صحيحة غير مؤشرة (unsigned integer) تمثل العنوان المكافئ للزاوية على الجدول. ومن ثم الوصول إلى قيمة الجيب والجيب تمام لتلك الزاوية[21].

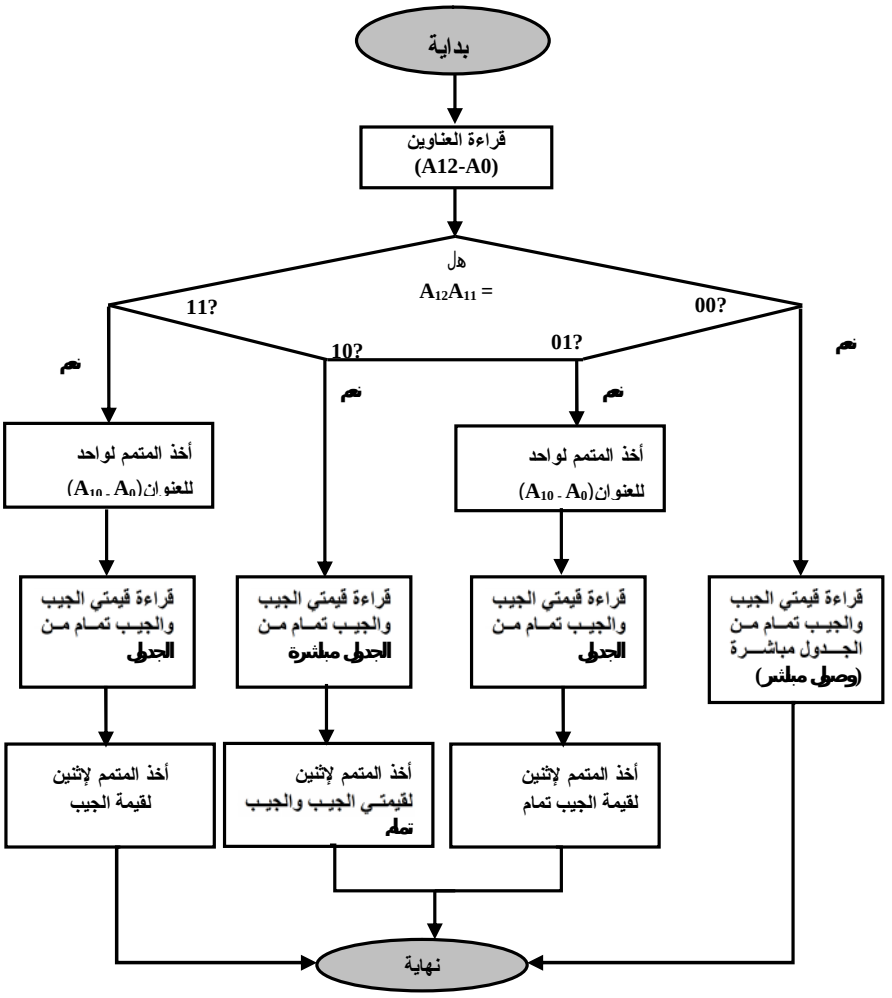
$$\theta = THETA \frac{2\pi}{2^{THETA_WIDTH}} \ radians \qquad \cdots (32)$$

2.3 طريقة جدول المعاينة غير المباشر (InDirect Lookup Table Method):

من اجل تفادي الزيادة الأُسية في حجم جدول المقارنة المباشر للحلول التي تتطلب معاملات إدخال أكثر من 16 بتاً (كعنوان للجدول)، وكذلك من اجل زيادة الدقة للحسابات المنفذة تم اقتراح وتنفيذ صيغتين أو مخططين لجدول المقارنة تقومان على أساس تنفيذ خطوات معالجة مسبقة للمعاملات وخطوات معالجة لاحقة للقيمة المقروءة من الجدول وهذا ما يسمى بجدول المقارنة غير المباشر. إن خطوات المعالجة هذه تتطلب دوائر منطقية لتنفيذها. لذا فان هاتين الصيغتين تجمعان بين الشرائح المنطقية واستخدام الذاكرة في حل الحسابات الرياضية.

1.2.3 الطريقة المقترحة الأولى:

إن الفكرة الأساسية في تنفيذ معمارية هذه الطريقة هي الاستفادة من خاصية التناظر الموجودة في الدوال المثلثية[12]. أما الهدف منها فهو تقليل حجم الذاكرة المستخدمة كجدول مقارنة، وذلك بخزن قيم ربع الموجة لدالة الجيب بدلا من قيم موجة كاملة وبذلك نقلل حجم الذاكرة المستخدمة إلى ربع حجمها في الطريقة السابقة (جدول المقارنة المباشر). إذ من الممكن الحصول على قيمتي الجيب والجيب تمام لبقية الموجة من خلال عمليات المعالجة المسبقة واللاحقة ويوضح المخطط الانسيابي في الشكل (3) سير تنفيذ هذه العمليات.



الشكل (3): مخطط انسيابي يوضح عمليات المعالجة المسبقة واللاحقة لجدول المقارنة غير المباشر.

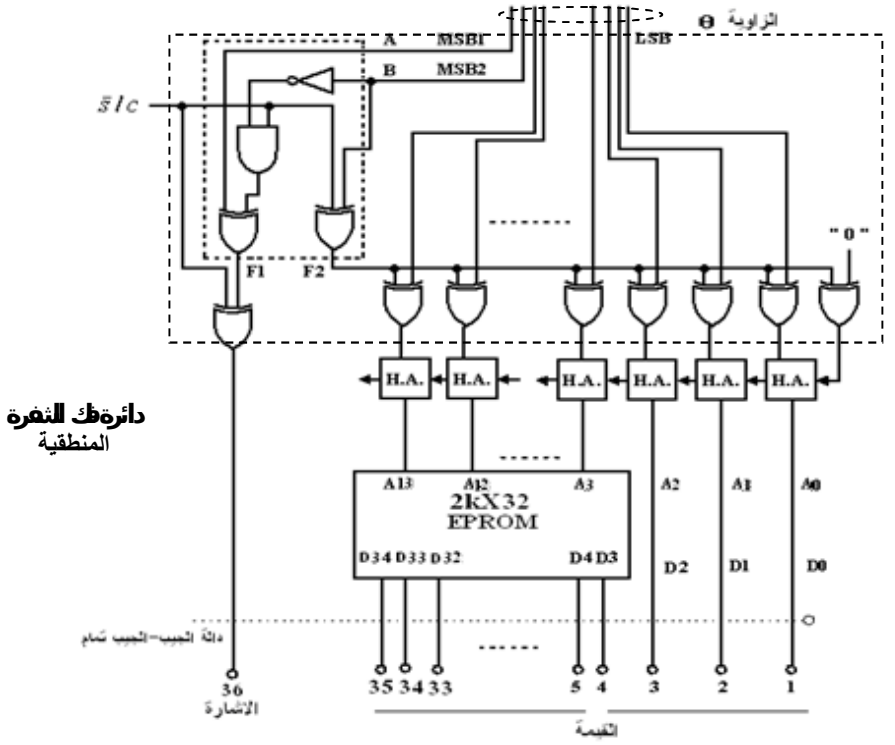
2.2.3 الطريقة المقترحة الثانية:

وهي طريقة مقترحة من قبل الباحث J. N. Lygouras [12]، وتم تنفيذها في هذا البحث مع اختلاف وحيد وهو أن الباحث استخدم ذاكرة قراءة فقط (EPROM) بحجم (2K X 16 bit)، أما الذاكرة المستخدمة في هذا البحث فهي بحجم (2K X 32 bit). وهي مشابهة للطريقة الأولى من حيث تقليل حجم الذاكرة المستخدمة كجدول مقارنة بمقدار 75% وبالاستفادة أيضا من خاصية التناظر للدوال المثلثية وتختلف عنها بأنها:

- تستخدم ثلاثة بتات إضافية لزيادة الدقة: وتتمثل بالبتات الثلاثة الأولى ذات المرتبة الأقل (LSBs) من الناتج. تضيف هذه البتات سبع قيم إضافية بين كل قيمتين متتاليتين في ذاكرة القراءة فقط PROM وتسمى إضافة القيم بهذه الطريقة بالاستكمال أو الاستيفاء الخطي(Linear interpolation).
- يكون الناتج (θ Sin أو θ Cos) ممثلا بصيغة القيمة المؤشرة.

ويمثل الشكل (4) الدائرة المادية لحساب دالتي الجيب والجيب تمام لهذه الطريقة، ويتبين أن اختيار نوع الدالة المطلوب حسابها (جيب أو جيب تمام) يتم عن طريق إشارة ($\overline{s}/c$) دون الحاجة إلى عمليات معالجة لاحقة، أما عمليات

المعالجة المسبقة فتتمثل بدائرة فك الشفرة المنطقية والتي الغرض منها تمكين قراءة قيم دالتي الجيب والجيب تمام للفترة الكاملة $(0 \le \theta < 2\pi)$ من قيم دالة الجيب المخزونة فقط للفترة $(0 \le \theta < \pi/2)$ مستندا على تناظر هذه الدوال حول المحورين السيني والصادي.



الشكل (4): الدائرة المادية المنفذة لحساب دالتي الجيب والجيب تمام.

يوضح الجدول (2) النتائج التي تم الحصول عليها لمجموعة من الزوايا النصف قطرية، حيث إن ناتج الجيب أو الجيب تمام يتمثل بالبتات من(D34-D0) حيث أن البتات من (D34-D3) تمثل القيمة المقرؤة من الذاكرة أما البتات (D2D1D0) فتمثل البتات الثلاثة الأولى من الناتج والتي بتغييرها يتم الحصول على سبع قيم إضافية بين كل قيمتين متتاليتين في ذاكرة القراءة فقط PROM.

الجدول (2) نتائج دالتي الجيب والجيب تمام باستخدام جدول المقارنة غير المباشر (الطريقة الثانية).

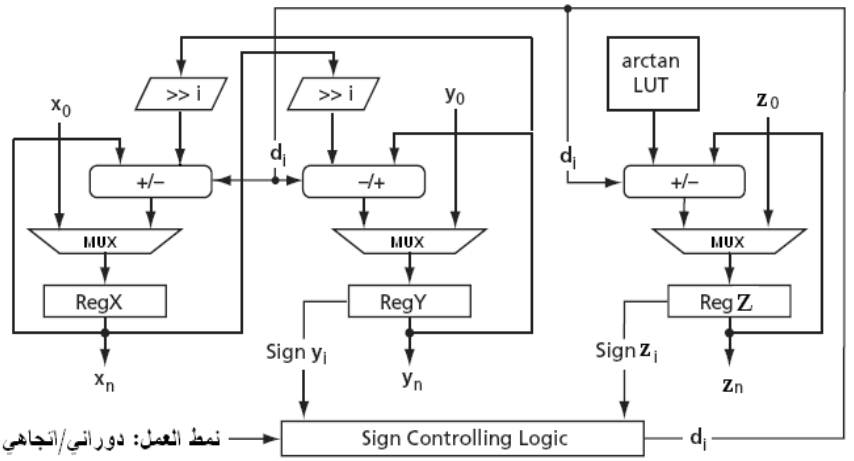
Theta (rad)	THETA (address)	pre- processing		Result	
		$A_{15}A_{14}$ MSB1,MSB2	THETA(A_{13}-A_3) 1's comp.	Data Accessed From Table (D34-D3)	Linear Interpolation Bits D_2 D_1D_0
pi/4	400	00	10000000000	Sin=2D413CCD	000 : 111
			10000000000	Cos=2D413CCD	
2pi/3	AAB	01	01010101011	Sin=376ADD88	000 : 111
			10101010100	Cos= 2003A099	
-5pi/6	2555	10	10101010101	Sin=2003A099	000 : 111
			10101010101	Cos= 376ADD88	
-pi/4	1C00	11	10000000000	Sin=2D413CCD	000 : 111
			01111111111	Cos=2D413CCD	

3.3 تنفيذ الطرز المختلفة لمعماريات CORDIC على رقاقة (FPGA):

هناك عدة طرائق لتنفيذ معمارية معالج CORDIC باستخدام رقاقة (FPGA). إن المعمارية المثالية تعتمد على المقايضة بين سرعة الأداء والمساحة السيليكونية المستخدمة في التطبيق المقصود. سيتم في البدء فحص معمارية CORDIC التكرارية ذات البتات المتوازية والتي هي عبارة عن الوصف المادي المباشر لمعادلاتها الأساسية ومنها نقيم المعماريات الأخرى على أساس استخدام حل الماديات الأقل وحل الأداء الأعظم.

1.3.3 معمارية CORDIC التكرارية ذات البتات المتوازية (المعالج الملفوف):

إن الوصف المادي المباشر للمعادلات الثلاثة الأساسية لخوارزمية CORDIC (24)و(25)و(26) هو أساس تصميم وتنفيذ المعمارية التكرارية الموحدة لمعالج CORDIC وكما موضح في الشكل (5).



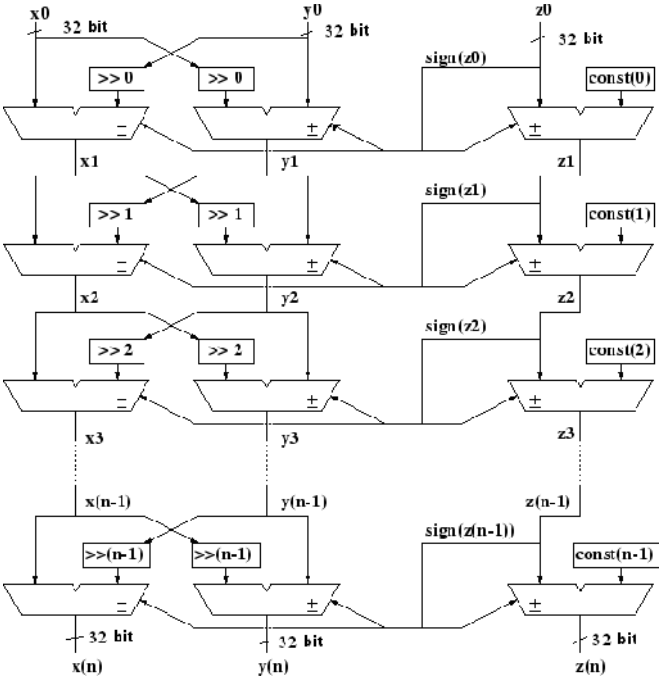
الشكل (5): المخطط الكتلي لمعالج CORDIC التكراري_الملفوف.

حيث إن (d_i): هي إشارة المتغير z_i في السجل z إذا كان نمط العمل هو النمط الدوراني أو هي إشارة المتغير y_i في السجل y ، إذا كان نمط العمل هو النمط الاتجاهي. في بداية عمل المعمارية يتم تحميل سجلات x, y, z بالقيم الابتدائية (x_0, y_0, z_0) عن طريق دوائر المزج (MUX). بعد ذلك، تمر قيم هذه السجلات خلال سجلات الإزاحة ودوائر (الجامع الطارح) ثم تعاد النواتج مرة أخرى إلى سجلات x, y, z. وفي كل مرة تتغير قيم سجلات الإزاحة نتيجة إجراء التزحيف المقرر لكل تكرار. وكذلك الحال في جدول المقارنة (arctan LUT) الذي يحوي مجموعة الزوايا الأولية، ففي كل تكرار تتم زيادة عنوان الذاكرة لتحميل القيمة المناسبة إلى دائرة (الجامع الطارح) للمتغيرz. تستمر عملية التكرار مستغرقةً n من دورات الساعة وصولاً إلى التكرار الأخير، حيث تتم قراءة النتائج من سجلات x, y, z بشكل مباشر. إن التصميم الموضح في الشكل (5) يستخدم مسار بيانات بعرض الكلمة ولذلك يدعى التصميم بـ (المعمارية ذات البتات المتوازية).

2.3.3 معمارية CORDIC المتوازية (المعالج غير الملفوف):

من الملاحظ أن المعمارية التكرارية تتطلب n من نبضات الساعة لحساب قيمة الدالة لكل إدخال. أي أن نسبة العطاء لهذه المعمارية هي أُنموذج لكل n من نبضات الساعة. ومن اجل الحصول على نسبة عطاء بواقع أُنموذج لكل نبضة يجب تنفيذ جميع التكرارات أو الدورات الدقيقة في نبضة واحدة. يتم ذلك بمضاعفة معمارية CORDIC

الأساسية بعدد n من المراحل أي أن كل مرحلة تتضمن معالجاً أساسياً، عندئذ لا نحتاج إلى جدول مقارنة لخزن قيم الزوايا الأولية. إذ إن كل قيمة من قيم الزوايا الأولية ستكون كثابت إدخال لكل مرحلة من مراحل المعمارية المتوازية وكما مبين في الشكل (6). هذه المعمارية تتطلب فقط مصفوفة من دوائر (الجامع الطارح) المرتبطة فيما بينها حيث تنتفي الحاجة إلى سجلات الإزاحة جاعلا من المعالج غير الملفوف معمارية ترابطية صرفة (أي بدون سجلات).



الشكل (6): معمارية CORDIC المتوازية المنفذة على رقاقة FPGA.

في معظم الأحيان، ولاسيما في رقائق (FPGA) فان استعمال دائرة ترابطية كبيرة غير محبذ من الناحية العملية[10].

3.3.3 معمارية CORDIC المتوازية بأسلوب خطوط الأنابيب:

من السهولة تنفيذ خطوط العمل الأنبوبية على المعالج المتوازي، وذلك بإضافة سجلات خزن بين دوائر (الجامع الطارح). هذه السجلات لا تزيد من كلفة التصميم المادي وذلك لكون معظم معماريات FPGA تحتوي على سجلات في كل خلية منطقية. كان الغرض من تنفيذ هذه المعمارية هو الحصول على أداء عالٍ وذلك بزيادة نسبة العطاء حيث إن نسبة العطاء للمعمارية تزداد بزيادة مراحل الخطوط الأنبوبية. وللحصول على نتيجة دقيقة باستخدام CORDIC لعدد ثنائي مكون من n بت يجب تنفيذ (n) من التكرارات للمعادلات ولو أدى ذلك إلى استهلاك في الزمن (time consuming)، وذلك بسبب التحسن المهم في الأداء الذي يطرأ على الخوارزمية والذي تتطلبه نتائج الدوال المثلثية[19]. لذلك تم تحديد عدد التكرارات بـــ32 والذي يساوي عدد بتات الإخراج.

4. المحاكاة ومناقشة النتائج:

في هذه الفقرة سيتم عرض النتائج التي تم التوصل إليها من خلال هذا البحث. إذ تتضمن هذه النتائج، أداء المعماريات التي تم تنفيذها باستخدام خوارزميتي كوردك وجدول المقارنة، وكذلك يتضمن عرضا لنتائج المحاكاة باستخدام الأداة Xilinx ISE Simulator (Test Bench).

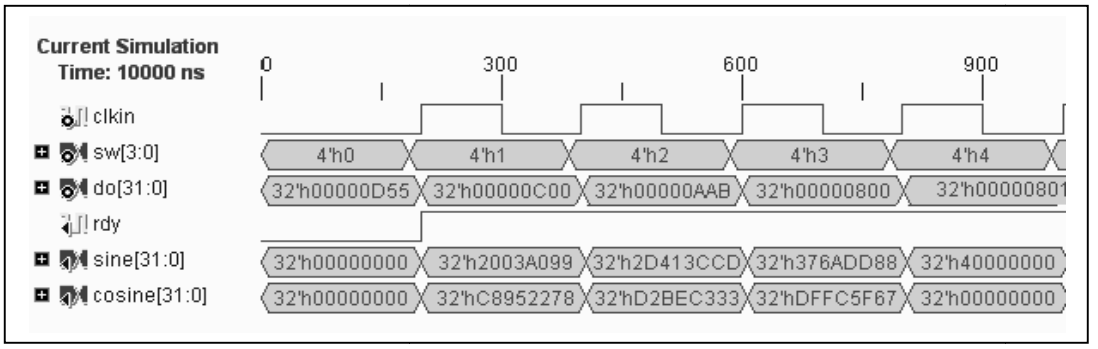
1.4 أداء المعماريات:

إن قياس الأداء للمعماريات المنفذة على رقاقة (FPGA) يعتمد على المقايضة بين المساحة السيليكونية والسرعة(أقصى تردد Maximum Frequency) لمعماريات CORDIC، أما قياس أداء طرائق جدول المقارنة فيعتمد على المقايضة بين حجم الذاكرة المستخدمة والدقة. وفيما يأتي شرحا موجزا لأداء كل معمارية:

1.1.4 جدول المقارنة المباشر:

من خلال ما تم تنفيذه، يمكن القول بأن جدول المقارنة المباشر يعد طريقة مرنة إلى حد كبير في حل الدوال والمعادلات الرياضية. إذ إنها لا تنفذ أية حسابات وإنما يتم خزن النواتج المحسوبة مسبقا في الذاكرة بشكل مباشر.

إن نسبة العطاء (Throughput) لجدول المقارنة المباشر هي (1) أي نتيجة واحدة في كل دورة نبضة وكما مبين في الشكل (7)، ولذلك تستخدم في التطبيقات التي تتطلب سرعة تنفيذ عالية.



الشكل (7) المخطط الزمني لنتائج دالتي الجيب والجيب تمام باستخدام جدول المقارنة المباشر.

2.1.4 جدول المقارنة غير المباشر:

لقد تم تنفيذ مخططين مقترحين لحساب دالتي الجيب والجيب تمام على أساس تنفيذ معالجات مسبقة لمعاملات الإدخال ومعالجات لاحقة للقيمة المقروءة من جدول المقارنة. هذا الإجراء هو للحد من الزيادة الأُسية في حجم الذاكرة المستخدمة، فقد تم تقليل حجم الذاكرة المستخدمة بمقدار 75% عن الحجم المستخدم وللدقة نفسها في جدول المقارنة المباشر. فضلاً عن ذلك، يتبين من الجدولين (3) و(4) أن عدد كتل الذاكرة المستخدمة في تنفيذ الطريقتين قد قل إلى ربع عددها المستخدم في جدول المقارنة المباشر، وذلك على حساب زيادة بسيطة في كمية الخلايا المنطقية.

الجدول (3) كمية الموارد المادية المستخدمة لإنشاء جدول المقارنة غير المباشر(الطريقة المقترحة الأولى).

Type Resources (or Frequency)	Utilized Resources	Total Resources	Ratio
Number of Slices	70	4656	1%
Number of Slices Flip flops	66	9312	0%
Number of Block RAMS	4	20	20%
Maximum Frequency	181.244MHz		

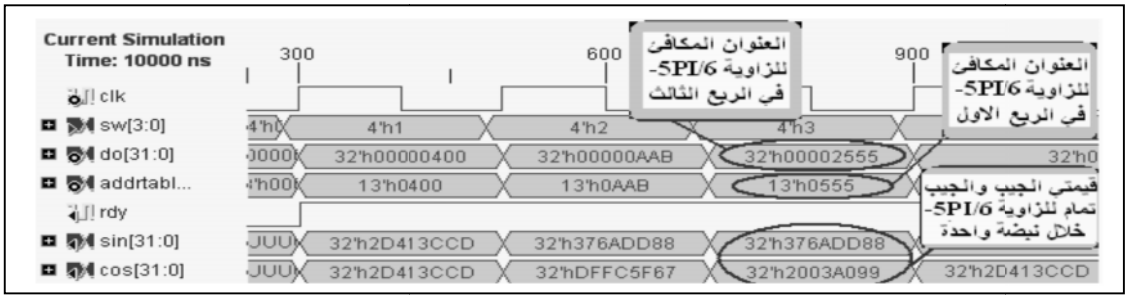
الجدول (4) كمية الموارد المادية المستخدمة في تنفيذ الطريقة المقترحة الثانية لإنشاء جدول المقارنة غير المباشر.

Type Resources (or Frequency)	Utilized Resources	Total Resources	Ratio
Number of Slices	14	4656	0%
Number of Slices Flip flops	3	9312	0%
Number of Block RAMS	4	20	20%
Maximum Frequency	163.1321MHz		

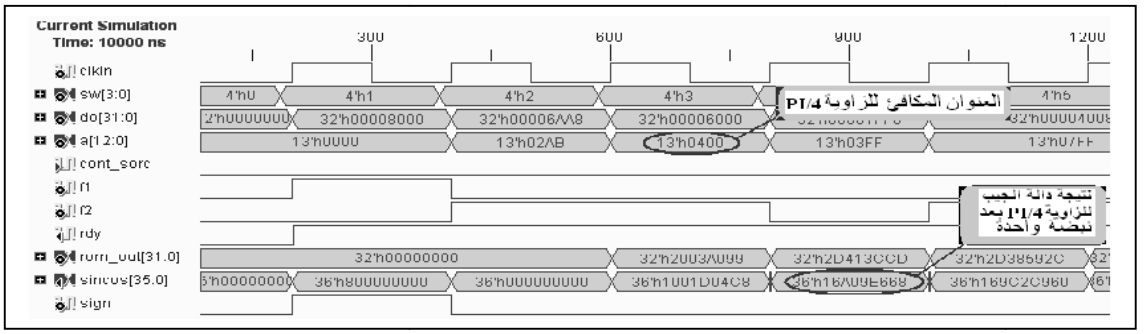
إن سرعة الأداء لطريقة جدول المقارنة غير المباشر اقل من سرعة أداء جدول المقارنة المباشر بسبب الزمن المطلوب للإعداد والحجز (setup time and hold time) للبوابات المنطقية لدائرتي المعالجة، والتي تزيد من طول مسار التأخير، وبذلك تحد من السرعة القصوى التي يمكن أن تعمل بها المعمارية.

إن العطاء الناتج من طريقتي جدول المقارنة غير المباشر هو أيضا أُنموذج واحد لكل دورة نبضة وكما موضح في الشكلين (8) و (9)، ولكنه في كلتا الحالتين اقل من عطاء طريقة جدول المقارنة المباشر. ففي الطريقة المقترحة الأولى تم الحصول على 181.244MHz كأقصى تردد للتشغيل وكما موضح في الجدول (3)، عليه فان:

العطاء = 181 ميكا أُنموذج\ثانية. أما في معمارية الطريقة الثانية وكما موضح في الجدول (4)، فان أقصى زمن تأخير تم الحصول عليه هو 1/163.13MHz = 6.130ns (وهذا الزمن اقل بكثير من الزمن 60 نانو ثانية الذي تم الحصول عليه في [12])، حيث أن أعلى تردد يمكن أن تعمل به هو 163.1321MHz، عندئذ فان العطاء يساوي (163 ميكا أُنموذج\ثانية).



الشكل (8) المخطط الزمني لنتائج دالتي الجيب والجيب تمام باستخدام جدول المقارنة غير المباشر (الطريقة الأولى).



الشكل (9) المخطط الزمني لنتائج دالتي الجيب والجيب تمام باستخدام جدول المقارنة غير المباشر (الطريقة الثانية).

3.1.4معماريات CORDIC:

في هذا البحث تم التطرق إلى ثلاثة أنماط مختلفة من المعماريات لتنفيذ خوارزمية CORDIC لحساب الدوال الرياضية. إن المعمارية المثالية تعتمد المقايضة بين حل الماديات الأقل وحل الأداء الأعظم. وفيما يأتي عرض لأداء كل معمارية من خلال مناقشة محاسن ومساوئ كل منها:

1.3.1.4 معمارية البتات المتوازية (المعمارية الملفوفة):

المحاسن:

- تتميز هذه المعمارية بصغر حجمها حيث إنها تستهلك 6% فقط من الشرائح المنطقية للـــFPGA . كذلك فان السرعة القصوى التي يمكن أن تعمل بها عالية (111.441MHZ) وكما مبين في الجدول (5).

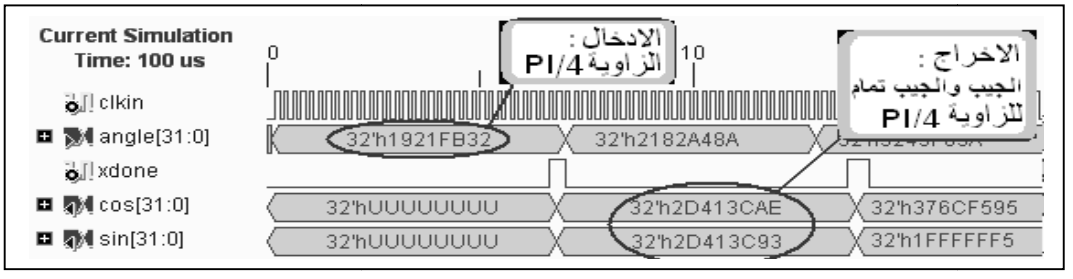
المساوئ:

- تستغرق n من نبضات الساعة لحساب قيمة الدالة لكل إدخال (حيث n عدد تكرارات خوارزمية CORDIC وفي هذا البحث تم تحديد n=32). أي أن عطاءها هو أُنموذج لكل 32 نبضة من نبضات الساعة. إذ إن العطاء= (التردد الأقصى) \ n =(111.441)MHz \ (32)=3.48 ميكا أُنموذج\ ثانية.
- وجود عدد كبير من التوصيلات المتقاطعة بعرض كلمة (32 بت) وهذا غير محبذ عمليا في التنفيذ باستخدام رقاقة (FPGA).
-

الجدول(5) الموارد المستخدمة في تنفيذ الكيان المادي لمعالج CORDIC الملفوف.

Type Resources (or Frequency)	Utilized Resources	Total Resources	Ratio
Number of Slices	284	4656	6%
Number of Slices Flip flops	108	9312	1%
Number of Block RAMS	0	20	0%
Maximum Operating Frequency	111.441MHz		

والشكل (10) يوضح المخطط الزمني لمحاكاة معمارية البتات المتوازية لدالتي الجيب والجيب تمام كمثال لتوضيح أداء هذه المعمارية.



الشكل (10) المخطط الزمني لنتائج حساب دالتي الجيب والجيب تمام باستخدام معمارية CORDIC للبتات المتوازية.

2.3.1.4 المعمارية المتوازية (غير الملفوفة):

المحاسن:

- تتميز بنسبة عطاء بواقع أُنموذج واحد لكل نبضة ساعة. ومن ملاحظة السرعات القصوى لمعماريات الدوال الرياضية المنفذة بأسلوب التوازي المبينة في الجدول (6)، يمكن حساب عطاء كل دالة، حيث إن عطاء معمارية حساب الجيب والجيب تمام هو (4.752) ميكا أُنموذج لكل ثانية، وعطاء معمارية جيب وجيب تمام القطع الزائد هو (4.435) ميكا أُنموذج لكل ثانية، أما عطاء معمارية حساب دالة "معكوس الظل" فهو (4.657) ميكا أُنموذج لكل ثانية.
- في هذه المعمارية تنتفي الحاجة إلى جدول المقارنة اللازم لخزن قيم الزوايا الأولية، ودائرة آلة القرار المسؤولة عن التحكم بعناوين جدول المقارنة وتنفيذ التكرارات، فضلاً عن سجلات التزحيف.

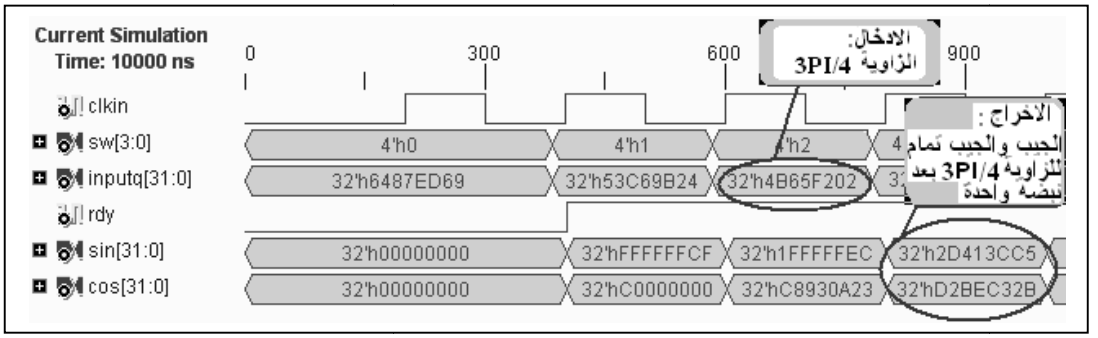
المساوئ:

- يتبين من ملاحظة الجدول (6) أن هذا النمط من المعماريات يستهلك كمية كبيرة من الشرائح المنطقية مقارنة بنمط المعمارية الملفوفة. كذلك فإن الترددات القصوى التي يمكن أن تعمل بها هذه المعماريات واطئة مقارنة بالأنماط الأخرى، وذلك بسبب طول مسار التأخير (Delay Path) الناتج عن أزمان الإعداد والحجز لعدد كبير من البوابات المنطقية التي تتألف منها المعمارية.
-

الجدول (6) الشرائح المنطقية المستخدمة في تركيب معماريات CORDIC المتوازية غير الملفوفة.

التردد الأقصى للتشغيل	No. of Bounded IOBs Total=232	No. of 4 input LUTs Total=9312	No. of Slices Flip flops Total=9312	No. of Slices Total=4656	نوع المصادر / الدالة
4.752MHz	70 (30%)	3996 (42%)	113 (1%)	2054 (44%)	الجيب والجيب تمام
4.657MHz	20 (8%)	4332 (46%)	178 (1%)	2235 (48%)	معكوس الظل
4.435MHz	70 (30%)	4169 (44%)	116 (1%)	2105 (45%)	جيب وجيب تمام القطع الزائد

ويوضح الشكل (11) أداء المعمارية المتوازية من خلال المخطط الزمني لمحاكاة معمارية حساب دالتي الجيب والجيب تمام.



الشكل (11) المخطط الزمني لنتائج حساب دالتي الجيب والجيب تمام باستخدام معمارية CORDIC المتوازية.

3.3.1.4 المعمارية المتوازية بأسلوب خطوط الأنابيب:

المحاسن:

- تتميز هذه المعمارية بسرعة أداء عالية إذ من المعروف إن أسلوب خطوط الأنابيب يسرع عمل المعمارية من خلال تقسيم المعمارية المتوازية إلى مراحل مفصولة. كل مرحلة تعمل بصورة مستقلة عن المرحلة التي تسبقها والمرحلة التي تليها. وهذا يؤدي إلى تجزئة مسار التأخير الطويل للمعمارية المتوازية إلى مسارات تأخير قصيرة ومن ثم زيادة السرعة التي يمكن أن تعمل بها المعمارية. إن نسب الزيادة في السرعة (Speed up) للمعماريات المنفذة بأسلوب خطوط الأنابيب تتراوح بين (24.7 إلى 30.3)×100% بالمقارنة مع المعماريات المتوازية. ويمكن ملاحظة السرعة القصوى لمعمارية كل دالة في الجدول (7).
- نسبة العطاء لهذه المعمارية عالية جدا، حيث إن أول إخراج (ناتج) يكون بعد تأخير (latency) يساوي 32 نبضة ساعة ومن ثم هنالك إخراج في كل دورة نبضة. والتأخير(latency): هو عدد النبضات المطلوبة للحصول على أول إخراج.

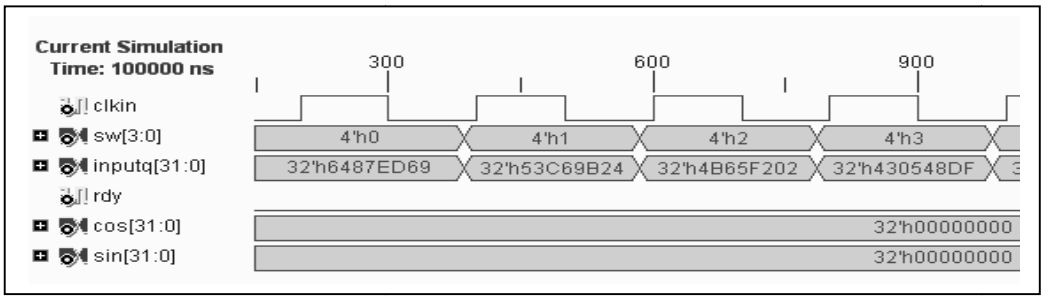
الجدول (7) أعداد الخلايا المنطقية المستخدمة في بناء معماريات CORDIC المتوازية بأسلوب خطوط الأنابيب.

نوع المصادر / الدالة	No. of Slices Total=4656	No. of Slices Flip flops Total=9312	No. of 4 input LUTs Total=9312	No. of Bounded IOBs Total=232	التردد الأقصى للتشغيل
الجيب والجيب تمام	2187 (46%)	3970 (42%)	3996 (42%)	70 (30%)	135.719 MHz
معكوس الظل	2136 (45%)	3935 (42%)	3857 (41%)	38 (16%)	135.512 MHz
جيب وجيب تمام القطع الزائد	2173 (46%)	4094 (43%)	4183 (44%)	70 (30%)	134.762 MHz

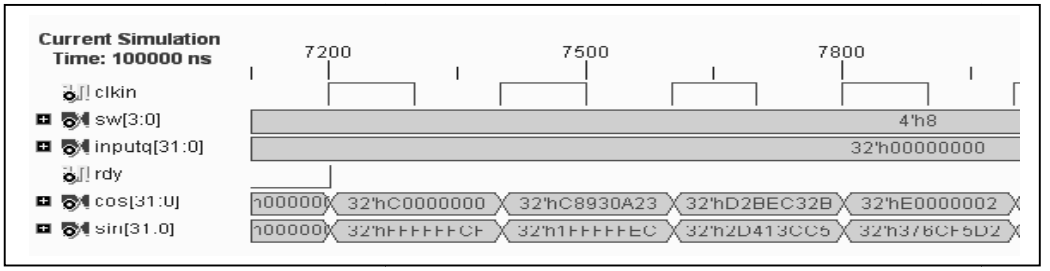
المساوئ:

- إن العدد الكبير للإدخالات(fan-in) والإخراجات (fan-out) والمتمثل بـــ(32 بت) لكل إدخال أو إخراج يؤدي إلى تعقيد في التصميم من حيث زيادة حجم المعمارية، إذ إن المعالجة تكون على مستوى الكلمة المؤلفة من 32 بتاً.
- إضافة سجلات بعدد المراحل تؤدي أيضا إلى زيادة الشرائح المنطقية من نوع المرجاحات (Flip-Flops) المستخدمة من رقاقة (FPGA)، إذ إن نسب (42%، 42%، 43%) من هذه الشرائح تم استخدامها في تنفيذ المعماريات الثلاث وكما ملاحظ في الجدول (7).

توضح الأشكال (12)و(13)و(14) النتائج العملية لمحاكاة التصميم للدوال الرياضية المنفذة (الجيب والجيب تمام، معكوس الظل، جيب وجيب تمام القطع الزائد) بشكل متعاقب، وبوساطة برنامج المحاكاة Xilinx ISE Simulator (Test Bench). من الشكل (12) عندما تكون قيمة الإدخال inputq التي تمثل الزاوية تساوي ($3\pi/4$=4B65F202$_h$) والتي تعادل (2.3561945$_d$) فان قيمة الإخراج sin التي تمثل جيب الزاوية تساوي (2D413CC5$_h$) والتي تعادل (0.707107$_d$) تقريباً، بينما قيمة الإخراج cos التي تمثل جيب تمام الزاوية تساوي (D2BEC32B$_h$) والتي تعادل تقريباً (-0.707107$_d$).



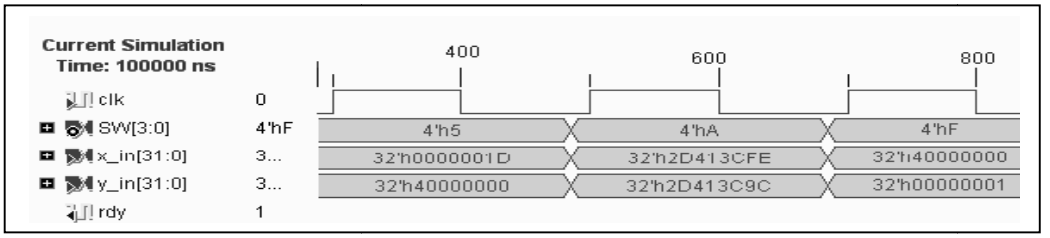
(a) قيم الإدخال (θ).



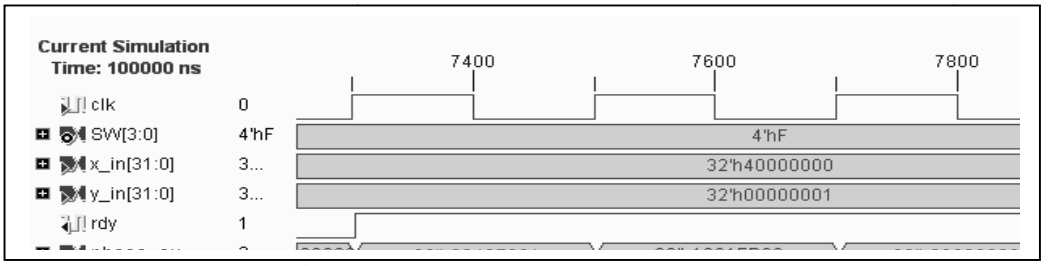
(b) النواتج (sinθ, cosθ) بعد n من دورات الساعة.

الشكل (12) المخطط الزمني لنتائج حساب دالتي الجيب والجيب تمام باستخدام معماريةCORDIC المتوازية بأسلوب خطوط الأنابيب.

من الشكل (13) عندما تكون قيمتا الادخالين ($x_in=0000001D_h$) و($y_in=40000000_h$) فان قيمة الاخراج التي تمثل زاوية نصف قطرية تساوي ($3243F69A_h$) والتي تعادل ($(\pi/2)$ 1.5707963_d). من ملاحظة الشكل (14)، حيث عندما تكون قيمة الادخال inputq التي تمثل الزاوية تساوي ($\pi/4=2.3561945$) والتي تعادل ($1921fb32_h$) فان قيمة الاخراج sinh التي تمثل جيب القطع الزائد تساوي ($379844DBB_h$) والتي تعادل (0.868671_d)، بينما قيمة الاخراج cosh التي تمثل جيب تمام القطع الزائد تساوي ($54C664F7_h$) والتي تعادل (1.32461_d).

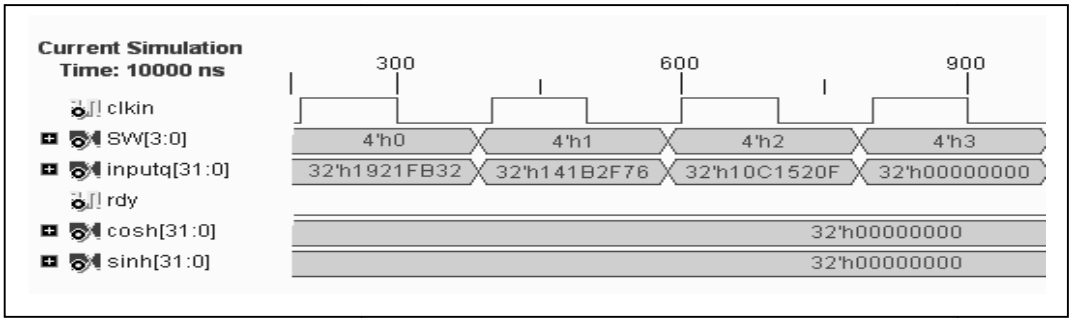


(a) قيم الإدخال (y_in=sinθ,x_in=cosθ).

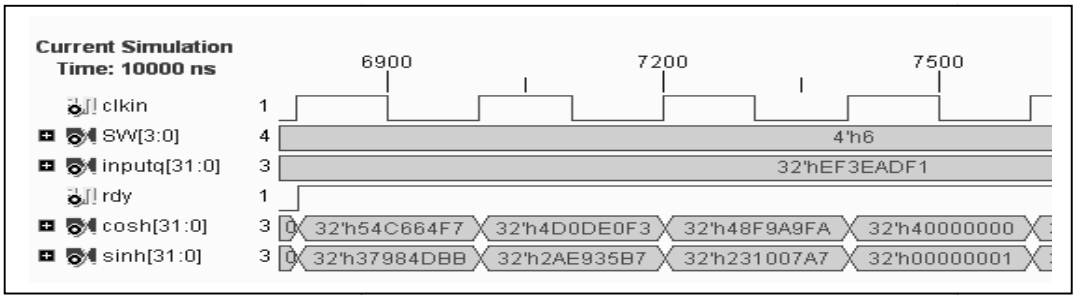


(b) النواتج (θ) بعد n من دورات الساعة.

الشكل (13) المخطط الزمني لنتائج حساب دالة معكوس الظل باستخدام معماريةCORDIC المتوازية بأسلوب خطوط الأنابيب.



(a) قيم الإدخال (θ).



(b) النواتج (sinhθ, coshθ) بعد n من دورات الساعة.

الشكل (14) المخطط الزمني لنتائج حساب دالتي جيب وجيب تمام القطع الزائد باستخدام معماريةCORDIC المتوازية بأسلوب خطوط الأنابيب.

5.الاستنتاجات:

من خلال ما سبق ذكره من الأفكار المنجزة في هذا البحث فقد تم استنتاج ما يأتي:

1. تستخدم طريقة جدول المقارنة للدوال الرياضية المتناظرة والتي لها مدى محدد من القيم مثل الجيب والجيب تمام.
2. تستخدم طريقة جدول المقارنة المباشر في التطبيقات التي تحتاج إلى سرعة أداء عالية ولا تحتاج إلى دقة عالية.
3. توفر طريقة جدول المقارنة غير المباشر لدالتي الجيب والجيب تمام حصراً 75% من حجم الذاكرة المستخدمة بالمقارنة مع تنفيذ الدالة باستخدام جدول المقارنة المباشر.
4. إن الطريقة الثانية لجدول المقارنة غير المباشر تحد من الزيادة الأُسية لحجم جدول المقارنة، حيث تم زيادة الدقة بإضافة سبع قيم للدالة بين كل قيمتين في الجدول وذلك من خلال البتات الثلاثة الأولى من الإدخال ذات المرتبة الأقل.
5. إن طريقة جدول المقارنة ليست طريقة خوارزمية حسابية لتنتج عنها أخطاء حسابية، وإنما هي طريقة لخزن نتائج محسوبة مسبقا وبذلك فان الأخطاء تنتج عن الطريقة التي تمت بها الحسابات قبل الخزن، وكذلك التقريب الذي استخدم في جعل القيمة المحسوبة تلائم سعة كل موقع من الذاكرة المستخدمة كجدول مقارنة، فكلما كان التمثيل أكثر دقة كلما قلت الأخطاء.
6. إن توفر عدد كبير من الخلايا المنطقية ومصادر الربط والتوجيه في رقائق (FPGA) الحديثة تسهل من تنفيذ الخوارزميات التكرارية (مثل خوارزمية CORDIC) بأسلوبي المعالجة المتوازية وخطوط الأنابيب.
7. لقد تم التوصل من النتائج التي تم الحصول عليها في هذا البحث، أن أفضل طريقة لتنفيذ خوارزمية CORDIC هي المعمارية المتوازية بأسلوب خطوط الأنابيب.

8. لقد تم فحص واختبار صحة عمل معماريات CORDIC الحسابية للدوال الرياضية المنفذة بعد تسقيطها على رقاقة XC3S500E FPGA، وذلك من خلال عرض نتائج حساباتها على شاشة العرض البلورية LCD الموجودة على لوحة سبارتان 3E وإجراء مقارنة مع نتائج الحاسبة العلمية لشركة مايكروسوفت (Microsoft Calculator Version 5.1) ® الموجودة في بيئة ويندوز XP-SP2 والتي تتميز بدقة نتائج عملياتها، حيث إن النتيجة تمثل بـــ32 رقماً بعد الفاصلة العشرية. والجدول (8) يوضح نتائج المقارنات من خلال حساب مقدار أعظم خطأ لكل دالة من الدوال المنفذة.

الجدول (8) مقدار أعظم خطأ لعدد من الدوال الرياضية المنفذة.

الدالة	مقدار أعظم خطأ (Max. Error)	عند الزاوية
دالة الجيب	6.7055225E-8	$\pi/6$
دالة الجيب تمام	-4.5640382E-08	$\pi/4$
دالة جيب القطع الزائد	8.6927116E-08	$\pi/6$
دالة جيب تمام القطع الزائد	5.6419920E-08	$\pi/4$

كذلك تمت مقارنة نتائج الجذر التربيعي لعدة قيم مع نتائج الحاسبة العلمية، فوجدت مقاربةً جداً حيث تراوحت قيم الأخطاء بين (0.0) و (4.34E-06). أما عند مقارنة عدد من نتائج دالة معكوس الظل، فكانت قيم الاخطاء التقريبية صغيرة جداً فمثلاً قيمة الخطأ الناتجة عن حساب $(\tan^{-1}\pi/6)$ هي (7.5598E-8) وقيمة الخطأ عند حساب$(\tan^{-1}\pi/4)$ هي(6.3397E-8)، بينما قيمتها عند حساب $(\tan^{-1}0)$ هي (0.0). هذه النسبة الضئيلة من قيمة الخطأ هي نسبية وليست مطلقة، إذ أن جميع الطرائق العددية تمتلك مثل هذه النسب من الخطأ.

References المصـــادر

[1] O. Mencer, N. Boullis, W. Luk and H. Styles, "Parameterized function evaluation for FPGAs", Field-Programmable Logic and Applications, LNCS 2147, pp. 544-554, 2001.

[2] Chow, Timothy Y., "What is a Closed Form Number?," The American Mathematical Monthly, Vol. 106, No. 5, pp. 440-448, May, 1999.

[3] Xilinx Incorporated. http://www.xilinx.com.

[4] K. D. Underwood and K. S. Hemmert. "Closing the Gap: CPU and FPGA Trends in Sustainable Floating-Point BLAS Performance,". In Proc. of IEEE Symposium on Field-Programmable Custom Computing Machines, California, USA, ISBN: 0-7695-2230-0, page(s): 219- 228 April 2004.

[5] L. Zhuo and V. K. Prasanna." Scalable and Modular Algorithms for Floating-Point Matrix Multiplication on FPGAs".. Proc. IEEE Conf. 18th International Parallel and Distributed Processing Symposium, New Mexico, USA, ISBN: 0-7695-2132-0, page(s): 92-, April 2004.

[6] Raimund Kirner, Markus Grössing, Peter P. Puschner "Comparing WCET and Resource Demands of Trigonometric Functions Implemented as Iterative Calculations vs. Table-Lookup", Institut for Technische Informatik Technische University at Wien, Austria, Dagstuhl Seminar Proceedings 06902 July 4, 2006 Germany.

[7] J.E.Volder, "The CORDIC trigonometric computing technique", IRE Trans.on Electronic Computers. Vol. EC-8. No.3, pp.330-334, Sep 1959.

[8] J. S. Walter. "The unified algorithm for elementary functions". In. Proc. AFIAPS Spring joint Computing Conf. Vol. 38, pp 379-385, 1971.

[9] Nienhaus, H.A., "A Fast Square Rooter Combining Algorithmic and Lookup Table Techniques," Proc. IEEE Southeastcon '89, Vol.3 on page(s): 1103-1105, 9-12 Apr 1989.

[10] Ray Andraka, "A Survey of CORDIC algorithms for FPGA based computers," International Symposium on Field Programmable Gate Arrays," Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, Pages: 191-, 200,Year 1998.

[11] Vladimirova, T. and Tiggler, H. "FPGA Implementation of Sine and Cosine Generators Using the CORDIC Algorithm,", Proc. of Military and Aerospace Application of Programmable Devices and Technologies Conference (MAPLD 99), Sep. 1999, Laurel, MA, A-2, pp. 28-30.

[12] John N. Lygouras " Memory Reduction in Look-Up Tables for Fast Symmetric Function Generators" IEEE Trans. on Instrumentation and Measurement, Vol. 48, No. 6, December 1999.

[13] M.W.Kharrat, M. Loulou, and N. Masmoudi, "A New Method to Implement CORDIC Algorithm," in Proc. IEEE Int. Conf. Electronics, Circuits and Systems, Malta, Vol.2, pp. 715-718, Sept.2001.

[14] S. Rvichandran, V. Asari "Pre-computation of Rotation Bits in Unidirectional CORDIC for Trigonometric and Hyperbolic Computations," Old Dominion University Norfolk, Virginia 23529, USA, Proc. IEEE Computer Society Annual Symposium on VLSI, ISBN: 0-7695-1904-0, On page(s): 215- 216, 20-21 Feb. 2003.

[15] T. Sansaloni , F. Angarita, A. Perez-Pascual " Efficient FPGA Implementation of CORDIC Algorithm For Circuular And Linear Coordinates," IEEE International Conference, Field Programmable Logic and Applications, ISBN: 0-7803-9362-7, Page(s):535 – 538, 24-26 Aug. 2005.

[16] Cheng-Yuan Lin, Chuen-Yau Chen, "High-Resolution Architecture for CORDIC Algorithm Realization," Proc. IEEE Int. Conf. on Communications, Circuits and Systems, Vol: 1, page(s): 579-582, ISBN: 0-7803-9585-9, June 2006.

[17] D. R. Llamocca-Obregón, C. P. Agurto-Ríos "A Fixed-Point Implementation of The Expanded Hyperbolic CORDIC Algorithm," Latin American Applied Research. artic_v3701/vol_37_1_pag:83-91 (2007).

[18] Hu, X., R.G Harber and S. C Bass, "Expanding the range of convergence of the CORDIC algorithm", IEEE Trans. on Computers, vol 40, No.1, pp13-21, Jan, 1991.

[19] Dong-U Lee, "Reconfigurable Hardware for Function Evaluation and LDPC Coding," MPhil/ PhD Transfer Report, Department of Computing, Imperial College London, July,2003.

[20] Ahmad Nour Al-Islam Al-Sawi,"HCORDIC: A High-Performance Cordic Algorithm," Master thesis, college of Graduate Studies King Fahd University Of Petroleum & Minerals, anuary, Dhahran Saudi Arbia 1997.

[21] "Sine/Cosine Look-Up Table v5.0," May 21, 2004 Xilinx Inc.
URL: www.xilinx.com/ipcenter.

تم اجراء البحث في كلية ألهندسة = جامعة ألموصل