دراسة وتحليل خوارزميات التخمين لحساب الجذر التربيعي

الباحثة: د. لما الدروبي

كلية العلوم – جامعة البعث

الملخص

تعتبر عملية الجذر التربيعي من أصعب العمليات الحسابية و أكثرها تعقيداً ، نقدم في هذه المقالة دراسة لبعض خوارزميات ايجاد الجذر التربيعي مع مجموعة من الخوارزميات المعدلة وننفذها برمجياً لبيان فعالية كل منها .كما سنورد مقارنة ما بين هذه الخوارزميات من حيث عدد التكرارات في كل خوارزمية بالإضافة الى دقتها ، حيث تم تنفيذ خوارزمية نيوتن– رافسون معدلة و أظهرت فعالية أكبر بتقليل عدد التكرارات اللازمة للحصول على النتيجة ،كما تم إظهار بعض مساوئ الخوارزمية المعدلة. وستنتهي المقالة بمجموعة من التوصيات التي تفيد في الحصول على قيمة الجذر التربيعي بعدد محدود من التكرارات و بالدقة المطلوبة .

الكلمات المفتاحية

الجذر التربيعي ، خوارزمية نيوتن–رافسون ، خوارزمية عدم الاستعادة ، ماتلاب .

Study and Analysis of the Estimation Algorithms to Calculate the Square Root

Abstract

The square root of the process is one of the most challenging arithmetic operations, and most complex, in this article we offer a study of some algorithms to find the square root with a group of modified algorithms and implement them programmatically to demonstrate the effectiveness of each of them. As we will cite a comparison between these algorithms in terms of the number of iterations in every algorithm in addition to the accuracy, where implementation of Newton-Raphson algorithm modified and showed greater efficiency by reducing the number of iterations needed to get the result, as we have shown some disadvantages. and will end the article a set of recommendations which are useful for obtaining the value of the square root of a limited number of iterations and the required precision.

Key Words : Square Root, Non-restoring algorithm , Newton-Raphson Method, Matlab .

مقدمة

تعتبر عمليات الجمع والطرح والضرب والقسمة والجذر التربيعي هي العمليات الحسابية الخمس الأساسية في الحاسوب . الجذر التربيعي هو احدى أكثر العمليات استخداماً في المخططات البيانية الحاسوبية و الحسابات العلمية كخوارزميات معالجة الاشارة الرقمية DSP و التحكم بالبيانات ومعالجتها وكذلك تطبيقات الوسائط [3].

تعتبر هذه العملية من المسائل التقليدية في نظرية الأرقام الحسابية والمستخدمة في مختلف التطبيقات ، حيث يعتبر الحصول على نتيجة دقيقة لهذه العملية أمراً صعباً.

تمت دراسة العديد من خوارزميات الجذر التربيعي وتطويرها ثم كتابة الكود الرقمي لها وهي:

Newton–Raphson method ، SRT redundant method ، SRT non redundant method , sequential algorithm (digit–by–digit method).

قدم الباحث وانمينغ تنفيذاً لعملية حساب الجذر التربيعي بأداء عالي وكلفة منخفضة [5]، تتضمن البنية المقترحة جامع حفظ الحمل (CSA(carry–save Adder، ولحساب البت الحالي من الجذر التربيعي يستخدم جامع CSA مرة واحدة فقط ، مما سمح باستخدام البنية الأنبوبية في التصميم للحصول على أداءٍ عالي.

استخدم بعض الباحثون كوان وفان دير ميروي [7] وكذلك كوليكوفا [10] مرشحات كالمان لتخمين البارمترات التي تفضي بالنهاية الى حساب الجذر التربيعي. حيث أشار الباحث الى تفوق مرشح كالمان الخالي من المشتقات UKF(Unscented Kalman Filter) على مرشح كالمان الموسع (EKF(Extended Kalman Filter . كما نفذ البعض خوارزميات مختلفة لحساب الجذر التربيعي، مستخدماً بارامتر استهلاك الطاقة

وكذلك التأخير الزمني للمقارنة بين هذه الخوارزميات [8]، وذلك لأطوال مختلفة 4 بت و 8 بت و 16 بت و 32 بت و 64 بت.

ناقش بعض الباحثون أفضل طريقة لحساب الجذر التربيعي باستخدام لوحات FPGA لتحقيق أفضل استخدام للموارد المتاحة [9] ، مبيناً أن الحصول على الدقة الأفضل يتم على حساب استخدام الموارد.

بشكلٍ عام، تصنف خوارزميات حساب الجذور التربيعية ضمن فئتين :

- ❖ الفئة الأولى تدعى مناهج التخمين مثل Newton–Raphson و Rough estimation .
- ❖ الفئة الثانية تدعى مناهج الحساب رقم–رقم digit–by–digit method .

من الضروري تصنيف منهج الحساب رقم–رقم إلى صنفين هما خوارزميات الاستعادة restoring وعدم الاستعادة non–restoring .

هدف البحث :

يتركز الهدف الأساسي للبحث في كيفية ايجاد قيمة الجذر التربيعي لعدد ما ، وذلك بخوارزميات مختلفة – وتحديد متطلبات كل خوارزمية من حيث عدد التكرارات و تسلسل التنفيذ متضمناً العمليات الحسابية اللازمة لكل خوارزمية للوصول الى النتيجة وبالدقة المطلوبة .

المواد وطرائق البحث :

التركيز الأكبر في البحث على الخوارزميات المعتمدة على التخمين حيث تم تنفيذها على برنامج ماتلاب ومنه تم استخلاص النتائج .

سنبين فيما يلي مجموعة الخوارزميات التي تمت دراستها في المقالة وخطوات كل منها.

1- التخمين التقريبي Rough Estimation

تتطلب العديد من خوارزميات الجذر التربيعي قيمة ابتدائية، اذا كانت هذه القيمة بعيدة عن القيمة الحقيقة للجذر التربيعي عندها سوف تتباطأ الخوارزمية ، لذلك سيكون من المفيد أن يكون هناك تقدير تقريبي والذي قد لا يكون دقيق بما فيه الكفاية ولكن يسهل من عملية الحساب.

بفرض s (العدد المطلوب جذره) معبر عنها بالعلاقة الرياضية $a \times 10^{2n}$ حيث $1 \le a < 100$ و n عدد صحيح يكون الجذر التربيعي $\sqrt{s} = \sqrt{a} \times 10^{n}$ يمكن تخمينه كالتالي:

$$\sqrt{s} \approx \begin{cases} 2 \times 10^{n} & if \quad a < 10 \\ 6 \times 10^{n} & if \quad a \ge 10 \end{cases}$$

المعاملات 2 و 6 تم اختيارها لأنها تقرب المتوسط الهندسي للقيم العليا والدنيا للعدد المعطى كما يلي:

$$\sqrt{\sqrt{1}\,.\sqrt{10}} = \sqrt[4]{10} \approx 2$$

$$\sqrt{\sqrt{10}\,.\sqrt{100}} = \sqrt[4]{1000} \approx 6$$

فمن أجل :

$$s = 125348 = 12.5348 \times 10^4 \Rightarrow \sqrt{s} \approx 6 \times 10^2 = 600$$

هذا التقريب مهم جداً في الخوارزميات التكرارية من أجل الحصول على نتيجة أسرع.

2- طريقة نيوتن–رافسون Newton–Raphson method

اقترح إسحاق نيوتن أول تفسير طبيعي لكفاءة الطريقة البابلية BABYLONIAN METHOD. أظهر نيوتن أن هذه الطريقة هي حالة خاصة لطريقة عامة لحل المعادلة غير الخطية [4].

منهج نيوتن–رافسون هو تقنية لحل المعادلات عددياً، وهذا المنهج يعتمد على فكرة بسيطة للتقريب الخطي. ليكن $f(x)$ تابع وليكن r جذر المعادلة $f(x) = 0$ ، نبدأ بتخمين القيمة x_0 لـ r ، ومن x_0 نوجد القيمة المخمنة الأفضل x_1 ومن x_1 نوجد x_2 وهكذا حتى نحصل على القيمة الأقرب لقيمة r ونطلق على هذه العملية "التكرار" *iterative* .

في بعض الأحيان نطلق على القيمة المخمنة *guess* حيث يكون منهج نيوتن جيداً في حال كانت x_0 المخمنة قريبة من r والعكس صحيح. لذلك عملية تخمين القيمة x_0 يجب أن تتم بعناية .

3- التكرار في منهج نيوتن–رافسون Newton–Raphson Iteration

لتكن x_0 القيمة المخمنة لـ r ولتكن $r = x_0 + h$ حيث هو الجذر الحقيقي و $h = r - x_0$ ، يبين الرقم h الفرق ما بين القيمة الحقيقية والقيمة المخمنة x_0 .

باستخدام التقريب الخطي نقول:

$$0 = f(r) = f(x_0 + h) \approx f(x_0) + h.\acute{f}(x_0)$$

$$h \approx -\frac{f(x_0)}{\acute{f}(x_0)}$$

$$r = x_0 + h \approx x_0 - \frac{f(x_0)}{\acute{f}(x_0)}$$

وتكون القيمة المحسنة لـ r هي x_1 وتعطى بالعلاقة :

$$x_1 = x_0 - \frac{f(x_0)}{\acute{f}(x_0)} \quad \Rightarrow \quad x_2 = x_1 - \frac{f(x_1)}{\acute{f}(x_1)}$$

ونستمر بهذه الطريقة، فإذا كانت القيمة المخمنة الحالية هي x_n عندها تكون القيمة المخمنة التالية :

$$x_{n+1} = x_n - \frac{f(x_n)}{\acute{f}(x_n)}$$

4- طريقة نيوتن رافسون لإيجاد الجذر التربيعي Newton–Raphson Method

وهي الخوارزمية الأولى التي استخدمت لحساب $\sqrt{s}$ ، الفكرة الرئيسية هي أنه اذا كانت x القيمة التقريبية للجذر التربيعي لعدد حقيقي غير سالب s عندها سوف تكون $\frac{s}{x}$ أقل من القيمة التقريبية للجذر.

بكلام آخر ، اذا كانت x هي القيمة المخمنة البدائية لـ $\sqrt{s}$ و e هو الخطأ في القيمة المخمنة $s = (x + e)^2$ عندها نقول :

$$e = \frac{s - x^2}{2x + e} \approx \frac{s - x^2}{2x} \;\; ; \;\; e \ll x$$

وبالتالي نستطيع تعويض الخطأ وتعديل القيمة المخمنة .

$$x := x + e = \frac{s + x^2}{2x} = \frac{x + \frac{s}{x}}{2}$$

مع كل عملية تكرار نحصل على تخمين أفضل ، وتستمر عملية التحديث حتى الحصول على الدقة المطلوبة ، أي تكون الخطوات كالتالي :

1− نبدأ بقيمة بدائية موجبة لـ $x0$ (القيمة الأقرب للجذر التربيعي لـ s) $x_0 \approx \sqrt{s}$.

2− نحسب قيمة x_{n+1} من العلاقة : $x_{n+1} = \frac{1}{2}(x_n + \frac{s}{x_n})$.

3− نكرر الخطوة الثانية حتى يتم الحصول على الدقة المطلوبة $\sqrt{s} = \lim_{n\to\infty} x_n$.

لحساب $\sqrt{s}$ حيث $s = 125348$ وباستخدام منهج التقدير التقريبي:

$$x_0 = 6 \times 10^2 = 600$$

$$x_1 = \frac{1}{2}\left(x_0 + \frac{s}{x_0}\right) = \frac{1}{2}\left(600.000 + \frac{125348}{600.000}\right) = 404.457$$

$$x_2 = \frac{1}{2}\left(x_1 + \frac{s}{x_1}\right) = \frac{1}{2}\left(404.457 + \frac{125348}{404.457}\right) = 357.187$$

$$x_3 = \frac{1}{2}\left(x_2 + \frac{s}{x_2}\right) = \frac{1}{2}\left(357.187 + \frac{125348}{357.187}\right) = 354.059$$

$$x_4 = \frac{1}{2}\left(x_3 + \frac{s}{x_3}\right) = \frac{1}{2}\left(354.059 + \frac{125348}{354.059}\right) = 354.045$$

$$x_5 = \frac{1}{2}\left(x_4 + \frac{s}{x_4}\right) = \frac{1}{2}\left(354.045 + \frac{125348}{354.045}\right) = 354.045$$

$$\sqrt{125348} \approx 354.045$$

لحساب $\sqrt{s}$ حيث $s = 127$

$$s = 127 = 1.27 \times 10^2 \quad => \quad \sqrt{s} \approx 2 \times 10^1 = 20$$

$$x_0 = 2 \times 10^1 = 20$$

$$x_1 = \frac{1}{2}\left(x_0 + \frac{s}{x_0}\right) = \frac{1}{2}\left(20.000 + \frac{127}{20.000}\right) = 13.175$$

$$x_2 = \frac{1}{2}\left(x_1 + \frac{s}{x_1}\right) = \frac{1}{2}\left(13.175 + \frac{127}{13.175}\right) = 11.407$$

$$x_3 = \frac{1}{2}\left(x_2 + \frac{s}{x_2}\right) = \frac{1}{2}\left(11.407 + \frac{127}{11.407}\right) = 11.270$$

$$x_4 = \frac{1}{2}\left(x_3 + \frac{s}{x_3}\right) = \frac{1}{2}\left(11.270 + \frac{127}{11.270}\right) = 11.269$$

$$x_5 = \frac{1}{2}\left(x_4 + \frac{s}{x_4}\right) = \frac{1}{2}\left(11.269 + \frac{127}{11.269}\right) = 11.269$$

$$\sqrt{127} \approx 11.269$$

5- خوارزمية نيوتن-رافسون المعدلة Modified Newton-Raphson Method

كما تبين لنا أن التخمين الأولي يؤثر وبشكل كبير على سرعة الحصول على النتيجة الدقيقة ، ولحل هذه المشكلة تم اقتراح حيز من الذاكرة [6] ، نخزن فيه مجموعة من القيم $2^0, 2^1, 2^2 \ldots \ldots 2^{31}, 2^{32}$. عندها نقارن العدد المطلوب الحصول على الجذر

التربيعي له وليكن m ، مع القيم السابقة ، فإذا كان m بين $2^2, 2^0$ عندها تكون القيمة المخمنة $x_0 = \frac{m}{2}$ ، واذا كانت m بين القيمتين $2^4, 2^2$ عندها تكون القيمة المخمنة $x_0 = \frac{m}{4}$ وهكذا .

الجدول –1– الذاكرة المستخدمة لاختيار القيمة المخمنة

رقم المجال	الحد الأدنى	الحد الأعلى	القيمة المخمنة x_0
1	2^0	2^2	$m/2$
2	2^2	2^4	$m/4$
3	2^4	2^6	$m/8$
4	2^6	2^8	$m/16$
5	2^8	2^{10}	$m/32$
6	2^{10}	2^{12}	$m/64$
7	2^{12}	2^{14}	$m/128$
8	2^{14}	2^{16}	$m/256$
9	2^{16}	2^{18}	$m/512$
10	2^{18}	2^{20}	$m/2^{10}$
11	2^{20}	2^{22}	$m/2^{11}$
12	2^{22}	2^{24}	$m/2^{12}$
13	2^{24}	2^{26}	$m/2^{13}$
14	2^{26}	2^{28}	$m/2^{14}$
15	2^{28}	2^{30}	$m/2^{15}$
16	2^{30}	2^{32}	$m/2^{16}$
17	$> 2^{32}$		$m/2^{17}$

6– خوارزمية حساب الجذر التربيعي بالاستعادة وبدون استعادة restoring and non–restoring square root algorithm

تعتبر هاتين الخوارزميتين من طرق الحساب رقم–رقم (digit by digit) ، وفيها يتم توليد رقم واحد في الجذر التربيعي في كل تكرار وبشكل متسلسل ، ومن مميزات هذه الطرق أن كل رقم يتم إيجاده في الجذر التربيعي لا يتغير فيما بعد .

في خوارزمية الاستعادة restoring لنعرف $r_0 = D \times 2^{-32}$ ، الجذر التربيعي الجزئي (partial square root) $qi = Q1Q2...Qi$ مع $q_0 = 0$ ، لتحديد بت الجذر التربيعي Q_{i+1} ،$(i = 0,1,2, \dots \dots \dots .,15)$ يحسب الباقي المؤقت $4r_i - (4q_i + 1)$ حيث r_i هو الباقي الجزئي (partial remainder) الذي تم الحصول عليه في التكرار i . إذا كان الباقي المؤقت غير سالب عندها :

$$\begin{gathered} Q_{i+1} = 1 , \\ q_{i+1} = 2q_i + 1 , \\ r_{i+1} = 4r_i - (4q_i + 1). \end{gathered}$$

في حالات أخرى:

$$\begin{gathered} Q_{i+1} = 0 , \\ q_{i+1} = 2q_i , \\ r_{i+1} = 4r_i . \end{gathered}$$

حيث q للدلالة على الجذر التربيعي الجزئي (حتى التكرار الحالي) و Q للدلالة على الجذر التربيعي النهائي.

معنى الاستعادة restoring هو أنه عندما يكون الباقي المؤقت سالباً ، نحن نستعيد الباقي الجزئي عن طريق إضافة $(4q_i + 1)$ إلى الباقي المؤقت أو اختيار الباقي الجزئي القديم $4ri$.

من أجل:

$$D = D_{31}D_{30}D_{29} \ldots\ldots\ldots\ldots . D_1 D_0$$

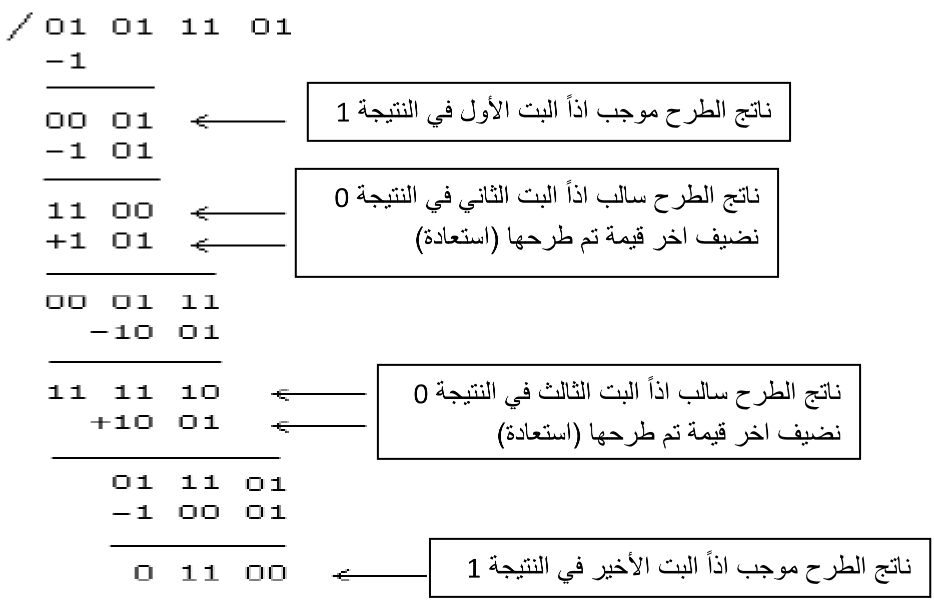
و
$$Q = \sqrt{D} = Q_{15}Q_{14}Q_{13} \ldots\ldots\ldots Q_1 Q_0$$

نبدأ الحساب مع الخانتين الأكثر أهمية $(D_{31}D_{30})$ من D ، حيث يوجد أربع قيم محتملة 00,01,10,11 . يمكن تحديد قيمة Q_{15} بطرح 01 من $D_{31}D_{30}$. اذا كانت النتيجة موجبة $Q_{15} = 1$ ، واذا كانت النتيجة سالبة ، عندها $Q_{15} = 0$ ونستعيد القيمة التي تم طرحها .

D_{31}	D_{30}	D_{29}	D_{29}	………	D_1	D_0

Q_{15}	Q_{14}	Q_{13}	………	Q_0

الشكل -1- صيغة المعامل (المجذور) D والجذر التربيعي Q



الشكل –2– مثال على خوارزمية الاستعادة

في خوارزمية عدم الاستعادة non–restoring نعدل كل بت في النتيجة Q مرة واحدة وليس مرتين ، حيث نبدأ بتخمين أولي $Q = Q_{15}Q_{14}Q_{13} \dots \dots \dots Q_1Q_0 =$ $0 \dots .1000$ ومن ثم نكرر من $k = 14\ to\ 0$ ، وفي كل تكرار ننفذ العملية $D - Q \times Q$ ، وبناءً على اشارة النتيجة نضيف أو نطرح 1 في البت الموافق في Q_k.

```
D = 01,11,11,11
Q = 1000  D - Q x Q  = 00,11,11,11  is nonnegtive
  + 0100
Q = 1100  D - Q x Q  = 11,10,11,11  is negtive
  - 0010
Q = 1010  D - Q x Q  = 00,01,10,11  is nonnegtive
  + 0001
Q = 1011  D - Q x Q  = 00,00,01,10
```

الشكل -3- مثال على خوارزمية عدم الاستعادة

يتضح لنا مما سبق أن الخوارزمية لها عدة مساوئ ، فهي تتطلب عملية جمع أو طرح للحصول على كل بت في النتيجة ، كما أنه من المحتمل حدوث خطأ في البت الأخير ، وتتطلب الخوارزمية اجراء العملية $D - Q \times Q$ في كل تكرار ، وعلى الرغم من امكانية تبديل العملية $Q \times Q$ الا أنها ستبقى معقدة .

قدم كل من Yamin Li و Wanming Chu خوارزمية عدم استعادة جديدة [1] ، ركزت هذه الخوارزمية على الباقي الجزئي في كل تكرار ، بعكس خوارزميتي الاستعادة وعدم الاستعادة السابقة التي ركزت على كل بت من الجذر التربيعي في كل تكرار .

الخوارزمية الجديدة تولد بت صحيح في النتيجة عند كل تكرار ، بما فيها التكرار الأخير ، وتكون العملية في كل تكرار بسيطة جداً ، نجمع أو نطرح بناءً على اشارة النتيجة في التكرار السابق ، الباقي الجزئي الذي يولد في كل تكرار يتم استخدامه في التكرار التالي حتى اذا كان سالباً (هذا هو معنى عدم الاستعادة في الخوارزمية). في التكرار الأخير ،

اذا كان الباقي الجزئي ليس سالباً ، فيكون هذا الباقي دقيق ، وفي حالات أخرى يمكن الحصول على الباقي الدقيق من خلال عملية جمع مع الباقي الجزئي.

```
D = 01,11,11,11    R=0      Q=0000  R: nonnegative
R = 01
  - 01            (-Q01)
R = 00,11                   Q=0001  R: nonnegative
  - 01,01         (-Q01)
R = 11,10,11                Q=0010  R: negative
  + 00,10,11      (+Q11)
R = 00,01,10,11             Q=0101  R: nonnegative
  - 00,01,01,01   (-Q01)
R = 00,00,01,10             Q=1011  R: nonnegative
```

الشكل -4- مثال على خوارزمية عدم الاستعادة الجديدة

النتائج والمناقشة :

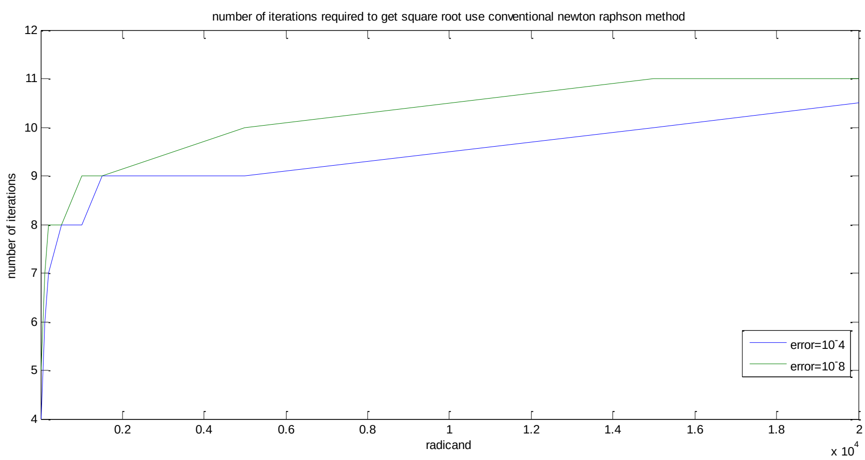
تعتمد فعالية الخوارزمية على عاملين أساسين ، عدد المتحولات المستخدمة ضمن الخوارزمية ، وعدد العمليات الحسابية والمنطقية الموجودة ضمن الخوارزمية ، فكلما كانت العمليات أقل كلما كانت الخوارزمية أكثر فعالية ، والعكس صحيح.

أولاً : في خوارزميات حساب الجذر التربيعي رقم-رقم يتم توليد بت صحيح في النتيجة في كل تكرار ، وبالتالي لا حاجة لتغيير هذا البت فيما بعد . حيث في خوارزمية الاستعادة restoring يتم طرح 01 من المجذور ، وفي حال كانت النتيجة (الباقي) سالبة يجب استعادة القيمة السابقة و ذلك عن طريق جمع القيمة التي طرحت. بينما في خوارزمية عدم الاستعادة non-restoring لا يوجد استعادة ، وبالتالي تم تقليل عدد العمليات الحسابية .

في خوارزمية عدم الاستعادة الجديدة ، لا يتم التركيز على كل بت من الجذر التربيعي ، وإنما تم التركيز على الباقي الجزئي في كل تكرار .

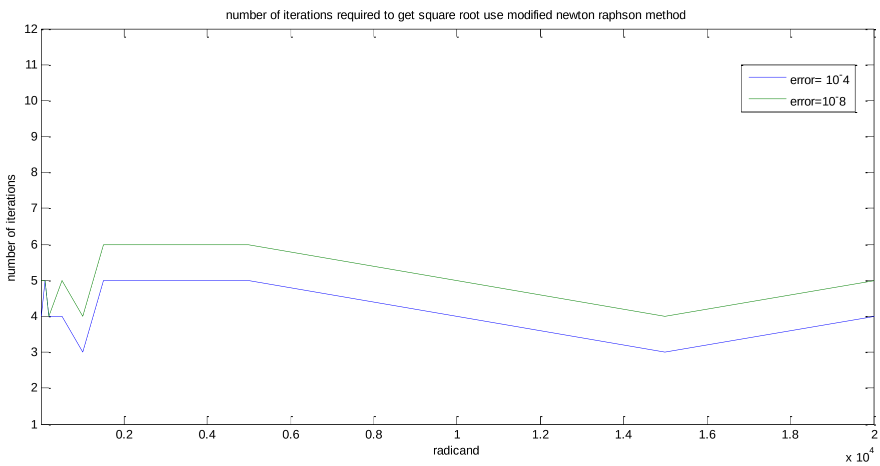
ما يميز الخوارزميات السابقة ، أنه وبما أنها تتعامل مع بتات المجذور يمكن تنفيذ الدارة التي تقوم بعملية الجذر التربيعي بشكل أنبوبي، وبالتالي الاستفادة من مزايا المعالجة التفرعية ، ما يؤدي الى تسريع عملية المعالجة وخصوصاً اذا تم تنفيذ الدارة بشكل أمثل[2][3].

ثانياً : في الخوارزميات المعتمدة على التخمين ، نجد أنه كلما كانت القيمة المخمنة أقرب الى القيمة الحقيقية كلما كانت الخوارزمية أسرع ، وكان عدد التكرارات أقل .ففي خوارزمية نيوتن رافسون التقليدية نجد ومن العلاقة ما بين المجذور وعدد التكرارات اللازمة للحصول على النتيجة بنسبة خطأ محددة، أنه كلما كانت نسبة الخطأ أقل (الدقة أكبر) كلما ازداد عدد التكرارات ، أي ازداد زمن التنفيذ نتيجة زيادة العمليات الحسابية المتطلبة في كل تكرار.



الشكل -5- التكرارات اللازمة للحصول على النتيجة باستخدام خوارزمية نيوتن التقليدية

في خوارزمية نيوتن رافسون المعدلة أصبحت القيمة المخمنة أقرب الى القيمة الحقيقية وذلك بسبب استخدام الذاكرة ، مما أدى الى تقليل عدد التكرارات بشكلٍ واضح . وكما في الخوارزمية التقليدية فإن عدد التكرارات يزداد مع زيادة الدقة المطلوبة في النتيجة.



الشكل -6- التكرارات اللازمة للحصول على النتيجة باستخدام خوارزمية نيوتن المعدلة

الاستنتاجات والتوصيات :

أولاً : مما سبق تبين لنا أنه وكلما كانت العمليات الحسابية في كل تكرار أبسط ، كلما كانت الخوارزمية أقل تعقيداً و أسهل وبالتالي تكون الدارة التي تنجز العملية أبسط(التكلفة أقل) .

ثانياً : تعتبر خوارزميات حساب الجذر التربيعي رقم- رقم ذات فعالية أكبر مع أنظمة المعالجة التفرعية التي تقلل من زمن التنفيذ.

ثالثاً : قدمت خوارزمية نيوتن رافسون المعدلة نتائجاً جيدة بالمقارنة مع الخوارزمية التقليدية ، حيث عدد التكرارات أصبح أقل بشكلٍ واضح ، ولكنها تتطلب عمليات مقارنة

بالإضافة الى حيز من الذاكرة ، ما يجعل هذه الخوارزمية ذات كلفة وتعقيد أكبر ، وبالتالي تستخدم هذه الخوارزمية عندما يراد تقليل عدد التكرارات التي تعطينا النتيجة بغض النظر عن الكلفة المادية.

رابعاً : من غير العملي تطبيق الاصدار الأنبوبي على طريقة نيوتن (حتى ولو كان عملياً فإنه مكلف للغاية).

خامساً : مما سبق يمكن طرح المواضيع التالية للدراسة ، والتي تؤدي الى زيادة فعالية الخوارزميات :

1- في الخوارزميات المعتمدة على التخمين يكون توجه الباحث الى تحسين القيمة المخمنة ،والتي تفضي بالنهاية الى تقليل عدد التكرارات ، وبالتالي تقليل عدد العمليات الحسابية و تقليل الزمن.
2- يجب التركيز على محددات مثل هذه الخوارزميات (التكلفة ، تعقيد الخوارزمية) وبالتالي يتم مراعاة هذه العوامل بحسب متطلبات المسألة.

المراجع

[1]- LI.Y & CHU.W, 1996 - A New Non-Restoring Square Root Algorithm and Its VLSI Implementations. IEEE International Conference on Computer Design, VLSI in Computers and Processors, Austin ,Texas, USA , pp538-544.

[2]- LI.Y & CHU.W, 1997 - Parallel-Array Implementations of A Non-Restoring Square Root Algorithm. IEEE International Conference on Computer Design (ICCD'97), Austin, Texas, USA, pp690–695.

[3]- SUTIKNO.T , 2011- An Efficient Implementation of the Non Restoring Square Root Algorithm in Gate Level. IACSIT International Journal of Computer Theory and Engineering, Vol.3, No.1.

[4]- KOSHELEVA.O, 2006 - Babylonian Method of Computing the Square Root :Justifications Based on Fuzzy Techniques and on Computational Complexity. Texas, USA.

[5]- LI.Y & CHU.W, 1997 -Cost/Performance Tradeoff of n-Select Square Root Implementations .

[6]- KHAN.A & ANAND.D & CHATURVEDI.D , 1997 -A Low Complexity Square Root Implementation.

[7]- WAN.E & Merwe.R , 2001 - THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION.IEEE, Beaverton,USA.

[8]- Franke.M & Th.Schwarzbacher.A & Brutscheck.M & ecker.St, 2007 - Implementation of Different Square Root Algorithms. CIITU

China–Ireland International Conference on Information and Communications Technologies.

[9]– De Dinechin.F & Joldes.M & Pasca.B, 2010 – Multiplicative Square Root Algorithms for FPGAs. ensl–00475779, version 1, France.

[10]– Kulikova.M, 2017 – Square–Root Algorithms for Maximum Correntropy Estimation of Linear Discrete–Time Systems in Presence of non–Gaussian Noise. Systems & Control Letters Volume 108, October 2017, Pages 8–15, ELSEVIER.