Using the Petri Network to Organize Resource Sharing between Two Processors

Firas Mohammad Al-Abadi
College of Computer Science and Mathematics
University of Mosul, Iraq

Received on: 12/10/2008 **Accepted on: 04/12/2008**

ABSTRACT

With the development of information technology and the use of its applications as an essential aspect in almost all patterns of life especially in fields of analysis and real time systems, a problem of controlling resources appeared and how to provide the system with the resources at a proper time to avoid the system to be failed or vanishes , a state called a deadlock. This research aims to use the modern concept of Petri Net to provide synchronization and coordination between systems having more than one processor , sharing a specific resources at a same time. The research concentrates on capability of applying feed backing concept of Petri Net to achieve a good synchronization between different systems to work efficiently without being in deadlock.

Keywords: information technology, real time systems, Petri Net, processor, deadlock.

استخدام مفهوم شبكة بتري لتنظيم مشاركة المصادر بين معالجين

فراس محمد العبادي

كلية علوم الحاسبات والرياضيات، جامعة الموصل

تاريخ استلام البحث: ٢٠٠٨/10/12 **تاريخ قبول البحث: ٢٠٠٨/١٢/04**

الملخص

مع التطور الحاصل في تقنية المعلومات واستخدامها كعامل أساسي وإشراكها في جميع أنماط الحياة وبشكل خاص في مجال التحليل وأنظمة الزمن الحقيقي ، ظهرت مشكلة السيطرة على المصادر وكيفية توفيرها للأنظمة في اللحظة المناسبة تلافيا لفشل النظام وتوقفه عن العمل عندما لا يتم توفير مصدر ما وهذا ما يسمى بمشكلة القفل المميت(Deadlock).

الهدف من هذا البحث هو استخدام مفهوم شبكة بتري (Petri Net) الحديث للعمل على توفير تزامن وتناسق بين الأنظمة التي تملك أكثر من معالج واحد وتشترك في العديد من المصادر فيما بينها بنفس الوقت.

يركز البحث على إمكانية تطبيق مفهوم التغذية الراجعة (Feedback) على شبكة بتري للحصول على تزامن جيد بين الأنظمة لتعمل بشكل صحيح دون الوقوع في مشكلة القفل المميت.

الكلمات المفتاحية: تقنية المعلومات، أنظمة الزمن الحقيقي، شبكة بتري، المعالج، القفل المميت.

1. المقدمة

تعتبر شبكة بتري أداة واعدة لشرح ووصف ودراسة أنظمة معالجة المعلومات وهي من الخصائص المتعلقة بالنظام (System) مثل التزامن (Synchronous) والتوزيع (Distributed) والتوازي (Parallel) والحالات الغير مستقرة (Nondeterministic) و (Stochastic) .[8][9]

تم اكتشاف شبكة بتري من قبل Carl Adam Petri حيث بدأت مفاهيمها الجزئية في 1962 وتعتبر أداة نمذجة (Modeling Tools) أو نمذجة رياضية (Mathematical) أو مخطط توضيحي (Graphical) ويمكن أن تطبق على عدة أنظمة وهي أداة موثقة .

يمكن استخدام شبكة بتري كاتصالات مرئية (Visual Communication) مشابه للمخطط الانسيابي (Flow Chart) أو كقالب رسم بياني (Block Diagram) أو في الشبكات [10] ، وتستخدم أيضا في وصفها الجزء العملي والنظري معا .

التعاريف العالمية لشبكة بتري عديدة فعلى سبيل المثال الجزء الأول من المقياس العالمي يقدم تعاريف رياضية لشبكة بتري تسمى النموذج اللفظي (Semantic Model) أما الجزء الثاني فهو يهتم بتعريف نموذج اللغة (High-level Petri Nets) بالاعتماد على الاختلاف بين الأدوات في اللغة وتسمى (Petri Net Markup Language (PNML)) حيث أن هناك عدة إصدارات من شبكة بتري سنتناول الجزء الأول في هذا البحث من الشبكة. [7]

هناك شبكة بتري يطلق عليها (Baukasten) والتي من شأنها دعم وتطوير البرامج والتطبيقات وهذا الدعم يشمل المهام التالية :-

1- إيجاد شبكة بتري ملائمة لتطوير البرامج .
2- استخدام تقنية شبكة بتري التي تخدم ناحية تطوير التطبيقات .
3- إيجاد نوع مناسب من أنواع شبكة بتري ملائم للتطبيقات .
4- إيجاد الأداة المناسبة .
5- توفير أمثلة محلولة تساعد مطوري التطبيقات . [3]

أهم المشاكل التي تحدث في الأنظمة والتي تعمل بالتزامن والتوازي هي مشكلة القفل المميت (Deadlock) وهي مشكلة شائعة وذلك نظرا لوجود عدد معين من المعالجات (Processors) وعدد محدد للمصادر (Recourses) مما يتطلب وجود مشاركة بالمصادر لغرض استمرار النظام في العمل بشكل صحيح وتام، هذه المشاركة بالمصادر من قبل المعالجات تتطلب استدعائها في زمن معين لاستخدامها وإرجاعها للاستفادة منها في معالجة أخرى، فماذا يحدث إذا ما تم طلب المصدر من قبل أكثر من معالج (Processor) في آن واحد؟ النتيجة بالتأكيد توقف النظام عن العمل بسبب عدم توفر المصدر في تلك اللحظة وهذا ما يسمى بمشكلة القفل المميت (Deadlock) [1].

هناك في الحقيقة عدة طرق واستراتيجيات من شانها أن تعمل على حل مشكلة القفل المميت في الأنظمة إما عن طريق :-

- المنع (Prevent)
- التجنب (Avoidance)
- التصحيح (Detection and Recovery) [11]

كل طريقة من هذه الطرق ممكن أن تعالج مشكلة القفل المميت ولكن حسب اعتبارات عديدة منها إن كان المصدر قابل للمشاركة أم لا إي هل يسمح بمشاركة أكثر من معالج في نفس الوقت أم لا، كمثال على ذلك الطابعة والتي لا يمكن استخدامها بنفس الوقت أما الذاكرة فيمكن أن تتم فيها المشاركة من قبل أكثر من معالج .

في هذا البحث سيتم طرح مشكلة القفل المميت (Deadlock) المتمثلة بوجود معالجين (Two Processor) مع مصدرين (Two Resources) تتم المشاركة فيما بينهما.

2. شبكة بتري Petri Net

هي واجهة رسومية وأداة نمذجة رياضية، وهي تتألف من أماكن (Places) وانتقالات (Transition) والتي ترتبط فيما بينها بالأقواس (Arcs)، الأماكن تحتوي على رموز (Tokens) وتمثل الحالة الراهنة للنظام، وتمثل الرموز بشكل أرقام . [6]

عملية التفعيل (Firing) تتم بانتقال الرمز من الحالة الابتدائية للشبكة وتسمى (Initial Marking(M0)) إلى الخطوط التحتية مباشرة لشبكة بتري وتكون موزونة. توجد عدة مفاهيم بالإضافة إلى الأماكن والانتقالات كمفهوم شبكة بتري المستقلة وغير المستقلة.[2]

تمثل الأماكن بدوائر والانتقالات بمربعات أو صناديق مرتبطة فيما بينها بأقواس وتعبر الأماكن عن الشرط (Conditions) والانتقالات هي الحدث (Event) [4] ، الجدول(1) يوضح بعض عمليات التغيير المثالية لعملية الانتقال.

الجدول(1) عمليات التغيير[4]

Input Place	Transition	Output Place
Precondition	Event	Post Condition
Input Data	Computation Step	Output Data
Input Signal	Signals Process	Output Signals
Resources Needed	Task or Job	Resources Released
Buffers	Processor	Buffers

2.1 الصيغة الرياضية التعريفية لشبكة بتري

شبكة بتري تتألف من خمسة رموز هي PN= (P, T, F, W, M0) حيث أن :-

P = {P1, P2, P3, ⋯⋯⋯⋯.,Pm} وهي مجموعة منتهية من الأماكن

T = {t1, t2, t3, ⋯⋯⋯⋯.,tn} مجموعة منتهية من الانتقالات

F = ⊆ (P x T) ∪ (T x P) مجموعة جزئية

W = F → {1, 2, 3, ⋯⋯⋯,n} وزن الدالة

M0 = P → {1, 2, 3,⋯⋯⋯.,m} بداية الوسم (Marked)

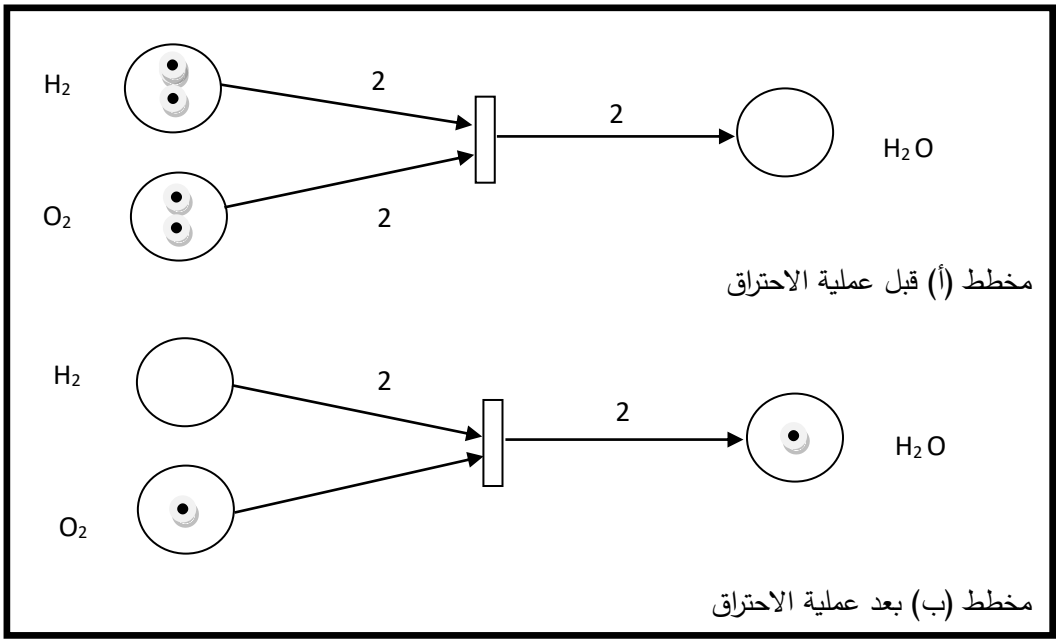
$P \cap T = \emptyset$ and $P \cap T \neq \emptyset$

لوصف عملية الوسم في شبكة بتري هناك قواعد للانتقال أو التفعيل وهي :-

1- الانتقال (T) قابل للتفعيل إذا كان إدخال (P) of (T) على الأقل W(P,T) token حيث أن W(P,T) هو وزن القوس (Arc) من (P) إلى (T) .

2- قابلية التفعيل (T) ممكن أن تحدث أو لا تحدث اعتمادا على ما مطلوب من النظام.

3- عملية التفعيل هي قابلية تفعيل (T) وتحريك W(P,T) token جزيئا من كل مدخل (P) لـ (T) وإضافة W(P,T)رمز لكل مخرج (P) لـ (T) بحيث أن W(P,T) token هو الوزن لعملية الانتقال من (T) إلى (P)، يمكن توضيح القواعد بأخذ المعادلة الكيميائية المعروفة للتفاعل المكونة للماء كمثال حيث يوجد رمزان في كل واحد من الأماكن كما في المخطط التوضيحي(1)(أ) حيث تظهر جزيئتان من كل من الهدروجين والأوكسجين وبعد عملية التفعيل الحالة الابتدائية سوف تتغير حسب القواعد الثلاثة المذكورة كما في المخطط التوضيحي (1)(ب)



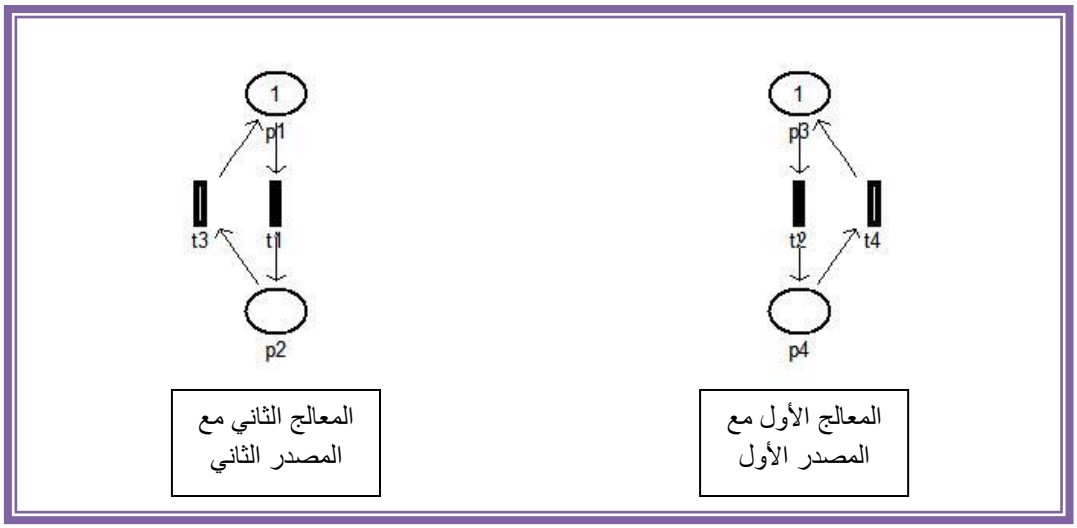
مخطط (1) توضيح معادلة الماء الكيميائية باستخدام Petri Net

والعملية سوف لن تحدث مجددا لعدم إمكانية تفعيل (T) مرة أخرى، وهناك مفهوم متعلق أيضا بشبكة وهو أن لكل مكان (Place) سعة محددة من الرموز أي أنها لابد من أن تكون ذات عدد ثابت k(p) وهو الحد الأعلى من أرقام الرموز التي تستطيع الأماكن أن تخزنها في إي وقت.

ومن هذا المنطلق يمكن أن نصغ قاعدة رابعة لقواعد التفعيل وهي أن عدد الرموز في كل انتقال من (P) إلى (T) يجب أن لا يتجاوز سعة k(p) بعد عملية التفعيل.

3. تمثيل المسألة

باستخدام شبكة بتري يمكن تمثيل مسألة النظام الذي يحوي على أكثر من معالج ويتطلب المشاركة في المصادر في آن واحد، فإذا أخذنا على سبيل المثال وجود كل واحد منهم على حدا يكون التمثيل بالمخطط التوضيحي(2) التالي

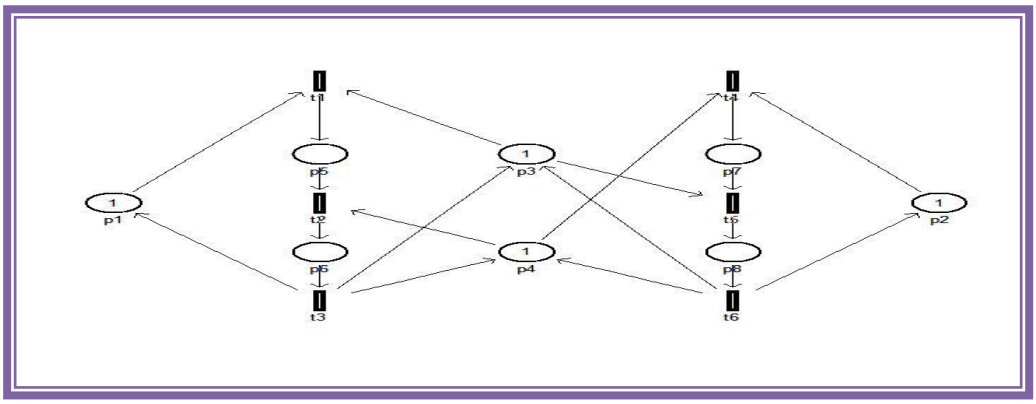


مخطط (2) يمثل معالجين مع مصدرين باستخدام Petri Net

تم تمثيل المعالجين بـ {P1,P3} وتمثيل المصدرين {P2,P4} والرقم واحد داخل الدائرة هو الرمز (Token) والذي يمثل عملية البدء بطلب المصدر من قبل المعالج وأما {t1,t2,t3,t4} فتمثل الانتقالات (Transition) ، وهو ضروري ضمن مفهوم شبكة بتري لإتمام عملية التطبيق [5]، المخطط التوضيحي أعلاه مناسب لتمثيل استخدام المصدر من قبل المعالج في الأنظمة المستقلة والتي لا تحتوي على معالجات ومصادر عدة مما يجنبها الوقوع في القفل المميت (Deadlock).

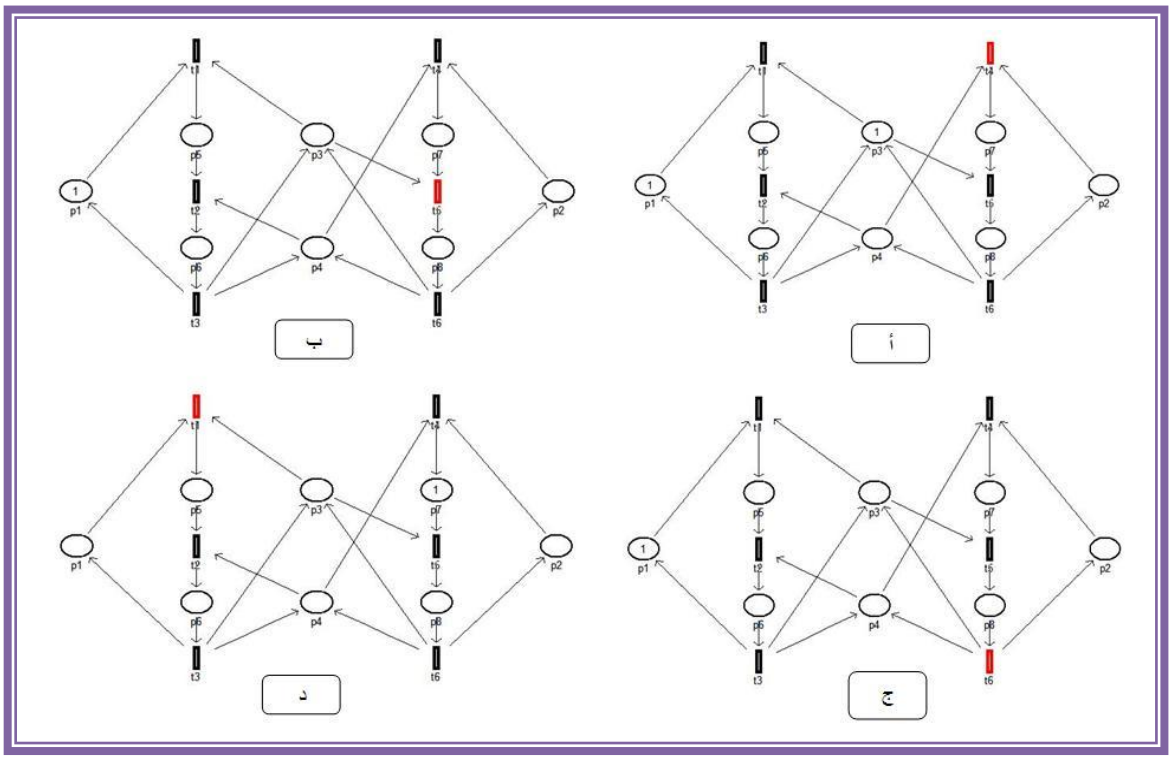
3.1 التطبيق العملي باستخدام Petri Net

تمثيل العملية نفسها على أنظمة متعددة المعالجات فستكون كما في المخطط التوضيحي (3) التالي :-



مخطط (3) عملية مشاركة المصدرين لكلا المعالجين

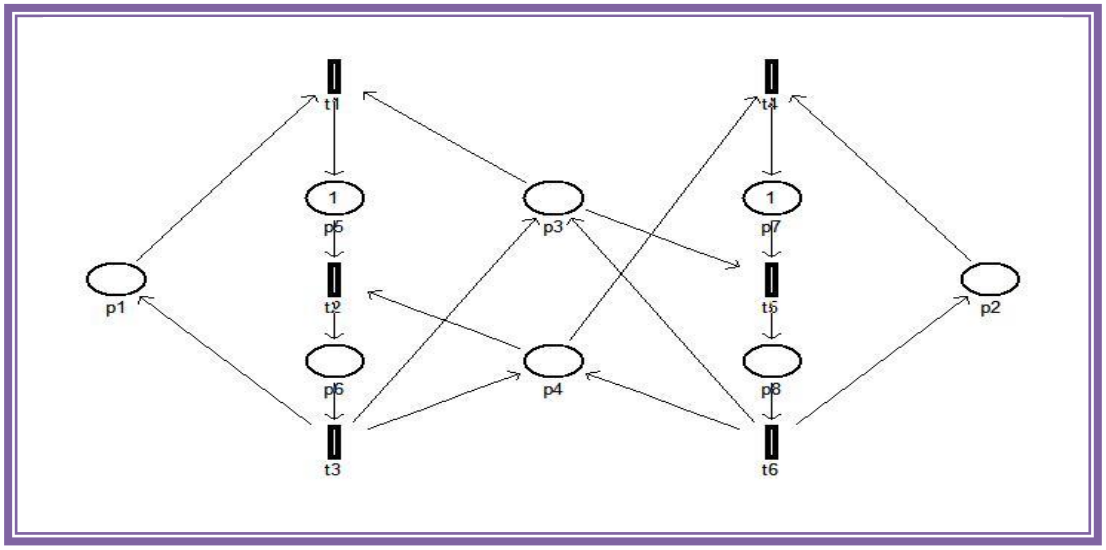
تم تمثيل النظام بمجموعة من الأماكن وهي {P1,P2,P3,P4,P5,P6,P7,P8} والانتقالات {t1,t2,t3,t4,t5,t6} ولبدء عملية طلب المصدرين من قبل المعالجين في آن واحد تم إضافة رمز(Token) في كل من {P1,P2,P3} حيث ستتم سلسلة من عمليات التفعيل (firing) كما في المخطط (4) التالي.



مخطط (4) تنفيذ سلسلة من عمليات الـfiring

عملية التفعيل ستتم إما في الـ t1 or t4 على حد سواء، وفيما يلي سلسلة من الخطوات باستخدام نموذج شبكة بتري :-

1- تفعيل t4 كما في (أ) سيتم اختزال الرمز من {p2 , p4} وإضافة رمز إلى (p7) .
2- تفعيل t5 كما في (ب) سينتقل الرمز إلى (p8) .
3- تفعيل t6 كما في (ج) ستنتقل الرموز إلى كل من {p2 , p3, p4}.
4- تفعيل t1 كما في (د) ستنتقل الرموز إلى (p5) و تفعيل t4 مرة أخرى كما في النقطة رقم(1) عندها سنلاحظ وجود رمزان في كلا {p5 , p7} مما يعني فشل استمرارية النظام ووقوعه في مشكلة القفل المميت (Deadlock) وذلك لعدم إمكانية تفعيل t2 , t5 بمعنى أقصى حد لتفعيل الـ t هو (5) مرات فقط، كما في المخطط التوضيحي(5) التالي.



مخطط (5) يبين وقوع النظام في حالة الـDeadlock

فيما يلي جدول(2) إحصائية تمثل عملية اختزال وإضافة الرموز من الأماكن حيث :-

Event :- تمثل عدد مرات وقوع الحدث أي عدد مرات تفعيل الـ t .

Arrival Sum :- ويمثل عدد الرموز التي تم إضافتها إلى الأماكن.

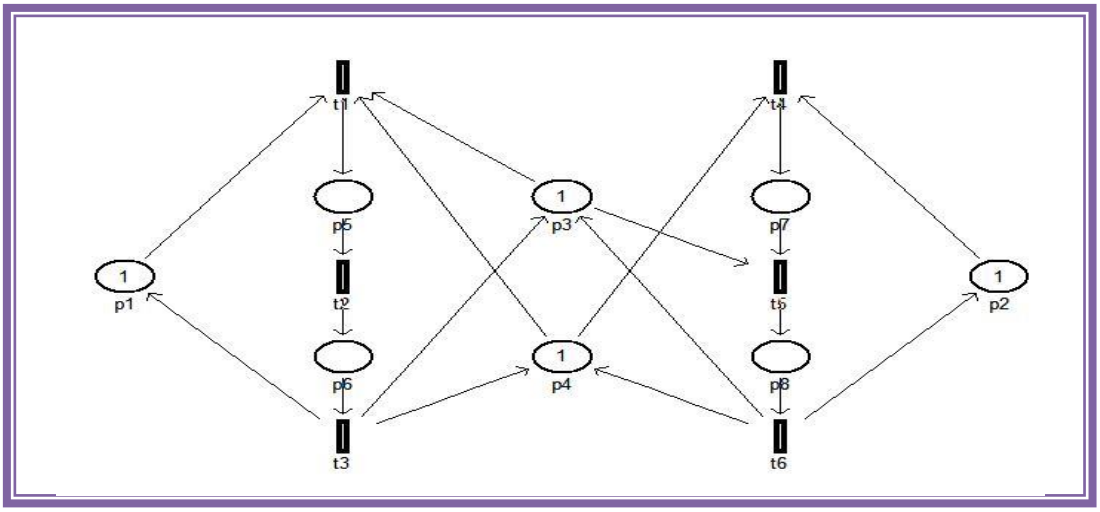
Throughput Sum : ويمثل عدد الرموز المختزلة من الأماكن.

جدول (2)إحصائية الـ Places خلال Event 5

Place Name	Arrival Sum	Throughput Sum
p1	0	1
p2	1	2
p3	1	2
p4	1	2
p5	1	0
p6	0	0
p7	2	1
p8	1	1

3.2 استخدام أسلوب الـ Lookhead Feedback للتخلص من مشكلة Deadlock

يمكن الاعتماد على شبكة بتري في حل وتنظيم العديد من المسائل وذلك باستخدامها ومزاوجتها مع التقنيات والأساليب المختلفة، وباستخدام أسلوب التغذية الراجعة وتطبيقها على شبكة بتري ستوفر إمكانية التخلص من عقبة الـ Deadlock وذلك عن طريق توفير مسار واحد للرموز (Tokens) يتيح للنظام استمرارية العمل ويجنبه الفشل، هذا يعني استمرار النظام إلى مالا نهاية أو توقفه حسب برمجة النظام من قبل المستخدم كاستخدام عدد مرات الحدث لـ (t) أو استخدام وقت محدد له، فيما يلي إعادة صياغة المسئلة بالمخطط التوضيحي(5) مع أسلوب التغذية الراجعة .

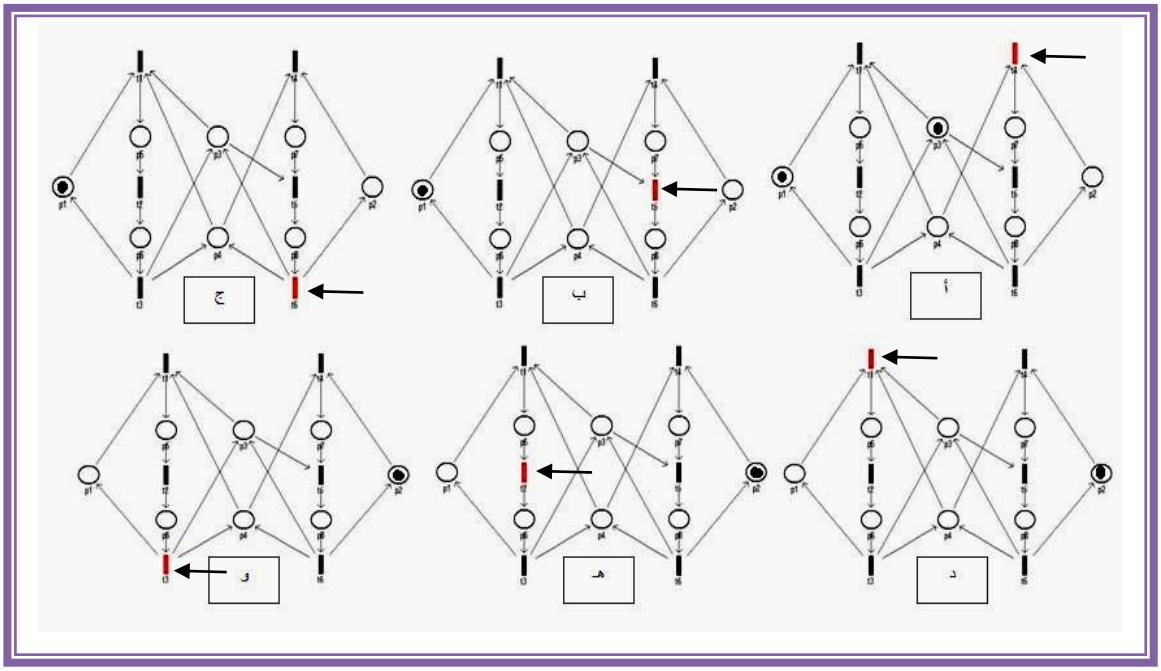


مخطط (5) استخدام أسلوب التغذية الراجعة للتخلص من الـ Deadlock

عملية التغذية الراجعة تمت عن طريق ربط قوس (arc) بين الـ p4 and t1 بدلا من الصيغة السابقة والتي كانت p4 and t2 وبهذا نجحنا في خلق توازن فعلي وعملي بين الطلبات من قبل المعالجين وتحديد بدقة عملية مشاركة المصدرين من قبل أي معالج دون الوقوع في حالة القفل المميت وهذه سلسلة من الأحداث تمت فيها استخدام المصدرين دون عائق ممثلة بمخطط توضيحي (6) أدناه (الأسهم المفردة تمثل الانتقالات المفعلة) .

عملية التفعيل تتم أما في الـ t1 or t4 ، فيما يلي سلسلة خطوات باستخدام التغذية الراجعة في نموذج لشبكة بتري :-

1- تفعيل t4 كما في (أ) سيتم اختزال الرموز من {p2 , p4} وإضافة رمز إلى (p7) .
2- تفعيل t5 كما في (ب) سينتقل الرمز إلى (p8) .
3- تفعيل t6 كما في (ج) ستنتقل الرمز إلى كل من {p2 , p3 , p4}.
4- تفعيل t1 كما في (د) سينتقل الرمز إلى (p5) .
5- تفعيل t2 كما في (هـ) سينتقل الرمز إلى (p6) .
6- تفعيل t3 كما في (و) ستنتقل الرموز إلى كل من {p1 , p 3 ,p4} .



مخطط (6) سلسلة من عمليات الـ Firing باستخدام التغذية الراجعة

وهكذا سوف تعاد العملية من جديد بدأ من نقطة (1) إلى (6) دون حدوث تقاطع في عملية التنفيذ، فيما يلي جدول(3) إحصائية تمثل عملية اختزال وإضافة الرموز من الأماكن حيث أن عدد الأحداث (Event) هو (10):-

جدول(3)إحصائية الـ Places خلال Event 10

Place Name	Arrival Sum	Throughput Sum
p1	1	2
p2	2	2
p3	3	4
p4	3	4
p5	2	1
p6	1	1
p7	2	2
p8	2	2

4. الخاتمة والاستنتاجات

إن الأنظمة التي تعاني من مشكلة القفل المميت (Deadlock) بسبب سوء إدارة استخدام المصادر الواجب توفرها لتنفيذ البرامج سواء أكانت مجموعة ملفات أو أجهزة إدخال وإخراج وغيرها، يمكن معالجتها بأسلوب يجعل من الاستحالة للنظام أن يقع في هذه المشكلة.

نتيجة لتطبيق البحث تم التوصل إلى أن استخدام شبكة بتري في معالجة القفل المميت عن طريق دمج الشبكة بأسلوب التغذية الراجعة (Lookhead Feedback) أدى إلى نتائج ايجابية جعلت من استخدام مفهوم الشبكة الهجين مع أنظمة التشغيل يحقق نتائج على مستوى عالي من الدقة وهذا بالضبط ما تحتاجه أنظمة الزمن الحقيقي (Real Time System) من حساسية استخدام المصادر ودقة الوقت المطلوبة في هذه الأنظمة.

شبكة بتري مقارنة مع الطرق المعروفة سابقا تقدم حلول تجعل من المستحيل الوقوع في القفل المميت خلال أي لحظة من لحظات عمل النظام نظرا إلى طريقة تصميم الشبكة وجعلها تعمل وفق نمط دقيق لا يقبل الخطأ كما هو موضح في المخططين السابقين المرقمين (6,5) ضمن الفقرة (3.2) .

المصادر

[1] Commoner F. 'Deadlock in Petri Nets' Weak field Applied Data Research Inc, 1972.

[2] David R. , Alla H. , Discrete, 'Continuous, and Hybrid Petri Nets' publisher by Springer, 2005.

[3] Ehrig H. , Reisig W. , Rozenberg G., Weber H. , 'Petri Net Technology for Communication-based Systems', publisher by Springer, 2003.

[4] Galbo B. and et al., "Introductory Tutorial on Petri Nets", Petri Nets Conference, June 2000, Aarhus, Denmark.

[5] Holt A. W. and Commoner F., 'Event and Conditions', Applied Data Research Inc, 1970.

[6] Home Page 'http://www.daimi.au.dk/PetriNets.

[7] International Standard ISO/IEC 15909-2 WD Version 0.9.0, 'Software and Systems Engineering – High-level Petri Nets Part 2: Transfer Format', June 23, 2005.

[8] Magott J. 'Performance evaluation of concurrent systems using Petri Net', Inform. Processing Lett, 1984.

[9] MURATA T. 'Petri Nets Properties, Analysis and Applications' ,FELLOW, IEEE, Invited Paper APRIL 1989.

[10] Petri C.A. , 'Kcommunication with Automata' , English translation NewYork, 1966.

[11] Silberschatz A., Galvin P. , Gagne, 'Operating System Concepts, fifth Edition', Addison Wesley Longman, 1998.