بحوث مستقبلية (١٦) ٢٠٠٦
ص ص [٢٣٥-٢٤١]

لغة فورث ومميزاتها كلغة تعريفية للمكونات المادية الافتراضية

شامل عبد الستار سليمان[١]

الملخص

تتيح لغة فورث للمبرمجين فرصة تعريف المكونات المادية باللغة نفسها التي تكتب بها البرمجيات. ويمكن أن تفحص المكونات المادية المعرفة في لغة فورث بواسطة تنفيذ كتابة (كلمات) خاصة بلغة فورث لاختبار العمليات التي تقوم بها تلك المكونات المادية.

Abstract

Forth language let programmers to define the hardware in the same language they write software in. The hardware that define in Forth can be verified by executing the hardware definition words, at the command line or by writing special words in Forth to test their operations.

(١) مدرس مساعد، قسم علوم الحاسبات، كلية الحدباء الجامعة.

تاريخ الاستـــلام: ٢٠٠٥/١٠/٢٤
تاريخ قبول النشر: ٢٠٠٦/٣/١٤

مقدمة في لغة فورث (Forth Language)

لغة فورث هي واحدة من اللغات الأكثر غرابة فهي تمثل طريقة جديدة في التفكير بالنسبة للبرمجة وقد تكون مشهورة في بعض الأماكن ومجهولة تماما في أماكن أخرى . المعجبون بهذه اللغة يحبونها لسرعتها في تنفيذ البرامج وللحيز الصغير الذي تحتاجه البرامج للخزن في الذاكرة ولاعتمادها على التركيب البنائي في البرمجة. وهي عبارة عن لغة غير تقليدية تعتمد على فلسفة اللغات التفسيرية المتسلسلة (Threaded interpreted language) إذ إنها تجمع معا مفهوم اللغة العليا واللغة الدنيا في آن واحد وتحوي لغة فورث على نظام تشغيل ومفسر(Interpreter) ومترجم (Compiler) و على الرغم من إن هذه المفاهيم متقاربة ولكنها عادة تعود إلى عناصر مستقلة في حين إن هذه العناصر تكاملت بصورة موجزة وكنظام فعال في لغة فورث من اجل تهيئة البيئة لتطوير البرامج وتنفيذها. (Mullen, J., 1987: 9-11,)

إن كل رمز يمثل إما برنامجاً تقليديا يدعى (كلمة)(Word) واما رقما. يشبه مفهوم (الكلمة) إلى حد بعيد مفهوم البرنامج المساعد في باقي اللغات. إذ إن سلسلة من (الكلمات) تشابه استدعاء سلسلة من البرامج المساعدة .

كذلك فان برامج بلغة فورث تمثل أهراماً من (الكلمات) إذ ينتج عن تنفيذ (كلمة) استدعاء (كلمات) فرعية التي تطلب بدورها (كلمات) أخرى حتى ينتهي التنفيذ في المستوى الأكثر انخفاضا والذي يتألف من الأوامر الأصلية فقط. يعرف هذا التركيب للبرنامج باسم الرمز السلسالي (Threaded code)والذي يتميز بالسرعة والقدرة العالية في التنفيذ. (Williams, G., 1980:14-17)

يعد تشارلز مور(Charles H. Moor) مخترع هذه اللغة في السبعينيات وهو فلكي استعملها عوضا عن لغة فورتران إذ استطاع زيادة الإنتاج لعدة أضعاف وقد شعر بان لغته الجديدة كانت تحسينا مهما بشكل يسمح تسميتها بلغة الجيل الرابع آنذاك ولهذا دعاها (Forth) الرابع. استخدمت اللغة للتحكم بالتلسكوب ((Kitt peak National observatory)) في اريزونا كما استخدمت في المسائل العلمية والصناعية وقد جعل لغته متوفرة للمستخدم وبالفعل أخذت اللغة تمد جذورها وفي مدة قصيرة بدأت مجموعات تتشكل من مستخدمي هذه اللغة ومن أهم التشكيلات هي (مجموعة المولعين بلغة فورث) ((Forth interest group (Fig) أما أهم المآخذ التي تكمن في برامج فورث هي صعوبة الفهم مقابل اللغات الأخرى فضلاً عن استخدامها التمثيل البولندي

المعكوس (Reverse Polish Notation) غير التقليدي والذي يتطلب تغيرا كاملا في التفكير والعمليات في البرمجة.(Alsulaiman, Shamil A., 1990:1-7)

مميزات لغة فورث

١- بدأت لغة فورث بمجموعة من الأوامر القياسية ذات قدرة عالية في التنفيذ ومن ثم أعطت هيكلا مفتوحاً من ناحية عدد مفردات رموزها وعدد قواعدها التركيبية بإذ أتاحت حرية تامة لاستحداث مفردات وقواعد جديدة من قبل المبرمج.

٢- أعطت لغة فورث القابلية على تجزئة المشكلة إلى أجزاء صغيرة كل جزء يمكن حله بكتابة (كلمة) ويمكن بسهولة تجميع هذه (الكلمات) في(كلمات) أخرى وبالتالي فان حل المشكلة تكون بتنفيذ (كلمة) واحدة فقط وهي تطابق الطريقة المثالية التي يعمل بها العقل البشري.

٣- إن لغة فورث هي لغة فعالة إذ يمكنها أن تترجم (الكلمات) الجديدة آنيا وبذلك يستطيع المبرمج أن يفحص (الكلمات) الجديدة بصورة آنيا.

٤- إن لغة فورث تتيح زيادة إنتاجنا في البرمجة وتجعل الحاسوب يعمل بكفاءة اكبر كذلك فإنها تعمل أسرع بكثير من معظم اللغات العليا وأسرع بالضعف من البرامج المكتوبة بلغة التجميع المرادفة لها(إذ إن أي تطبيق مكتوب بلغة فورث يحتاج إلى مساحة اقل من مرادفه المكتوب بلغة التجميع).

مكونات لغة فورث الأساسية

أهم مكونات لغة فورث هي:

١. المعجم (Dictionary): إن لغة فورث مرتبة في معجم وهو عبارة عن سلسلة من (الكلمات) مرتبطة مع بعضها البعض عبر عناوين تسمى عنوان الاتصال (Link address) الموجود في حقل الاتصال إذ كل (كلمة) في المعجم تتكون من أربعة أقسام (حقل الاسم – حقل الاتصال – حقل الرمز– حقل العوامل) كما في الشكل (١).

شكل (١)

إذ إن **حقل الاسم** يحوي على اسم (الكلمة).

حقل الاتصال يحوي على عنوان حقل الاسم (للكلمة) السابقة.

حقل الرمز يحوي إما عنوان روتين يدعى (DOCOL)

أو عنوان حقل العوامل(2 + $)

حقل العوامل يحوي إما عناوين (الكلمات) التي تعرف وظيفة هذه (الكلمة)

أو الايعازات بلغة التجميع التي تعرف وظيفة هذه (الكلمة)

٢. **المكدس (Stack)**

تعتمد لغة فورث اعتمادا كبيرا على مكدسين من نوع (LIFO) يستخدم أحدهما لخزن الأعداد والعناوين المستخدمة من قبل (الكلمات) وآخر لخزن عناوين الرجوع.

٣. **المفسر (Interpreter)**

يقوم بتنفيذ (الكلمات) في حالة وجود تعريف لها في المعجم وإلا تعد رقما وتعامل كعدد فإذا فشلت هذه الحالة سيرجع خطا.

٤. **المترجم (Compiler)**

ما يميز لغة فورث هو وجود المترجم إلى جانب المفسر إذ يمكن تعريف (كلمة) جديدة وخزنها في المعجم عبر استخدام المترجم. والمترجم هو عبارة عن (كلمة)

مخزونة في المعجم أيضا وتدعى (:) يمكن استدعائها لكي تترجم أية (كلمة) وخزنها في المعجم ومن ثم استخدام (الكلمة) بعد ذلك كأحدى (كلمات) لغة فورث الأصلية. (Derick, M., Baker, L.,1982:5-12)

مقدمة في اللغات التعريفية للمكونات المادية الافتراضية

إن استخدام التصميم بواسطة الحاسوب اصبح من الأمور المهمة والأساسية لتطوير إنتاج المكونات المادية وهنالك عدة لغات تعريفية للمكونات المادية الافتراضية ولكن عندما نريد أن نعمل في بيئة وحيدة على المكونات المادية والبرمجيات فان لغة فورث تكون هي الأفضل.

استخدام لغة فورث لمحاكاة المكونات المادية

هنالك عدة أسباب لاستخدام لغة فورث لمحاكاة المكونات المادية :

١- إن الموديل البرمجي يمكن أن يكتمل أسرع بكثير من المكونات المادية.
٢- يمكن اختبار التطبيق قبل تصميم المكونات المادية.
٣- يمكن بسهولة عرض أو تغيير أية حالة داخلية في التصميم.
٤- يمكن للموارد الخاصة بالتصميم أن تقنن بصورة مبكرة في عملية التصميم.

استخدام لغة فورث كلغة تعريفية للمكونات المادية الافتراضية

إن استخدام لغة فورث كلغة تعريفية للمكونات المادية الافتراضية توفر الفوائد التالية:

١- تقليل الزمن المستغرق لإنشاء النظام.
٢- يمكن أن تكون لغة فورث لغة وصفية للمكونات المادية.
٣- يمكن للمشروع أن يستخدم لغة وحيدة.
٤- إمكانية القدرة على تصميم مكونات مادية فعالة.

أسلوب التصميم بواسطة لغة فورث كلغة تعريفية للمكونات المادية الافتراضية

١- كتابة المحاكاة البرمجية للتصميم.
٢- اختبار التصميم.
٣- تحويل المحاكاة البرمجية إلى تعاريف للمكونات المادية.
٤- ترجمة التعاريف للمكونات المادية إلى معادلات منطقية.

٥- ضبط المعادلات المنطقية في الأجهزة.

٦- فحص والتأكد من أن المعادلات المنطقية تعمل بشكل صحيح.

٧- تسيير الإشارات وتحديد مسامير الإدخال والإخراج.

٨- تحويل التصميم الذي تم تسييره إلي خارطة.

الموديل البرمجي

إن الموديل البرمجي يشبه الصندوق الأسود لا يهم كيف يعمل مادام يعمل بشكل صحيح. أهم المميزات في الموديل البرمجي هو إمكانية إغفال التفاصيل عندما تكون الأفكار قد قيمت مبكرا في مرحلة التصميم .

التعريف للمكونات المادية

بعد أن يتم تقييم الموديل البرمجي يحول إلى تعريف للمكونات المادية . التعريف هو نموذج موسع للموديل البرمجي. البرامج سوف تعمل بشكل أبطأ قليلا عليه ، لكن يجب أن تعمل بالطريقة نفسها

لكي يتطابق التصميم مع هيكل الجهاز ويجب أن يتضمن التعريف المعلومات المتعلقة بالجهاز.

إن عملية التحويل إلى التعريف يتضمن تقسيم الوظائف المعقدة إلى أقسام صغيرة. في هذه المرحلة يمكن بناء مكتبة من (الكلمات) لكي تكون الوظائف المعقدة. (Johan, R., 2000: 1-6).

الاستنتاجات

إن لغة فورث وفرت أساس جيد للغات التعريفية للمكونات المادية الافتراضية وذلك لقابليتها على توسيع مفرداتها ومن يعمل مع فورث يعلم بإمكانيتها على زيادة قيمة الإنتاج البرمجي لخواصها الفريدة. وهي مفيدة جدا عندما تعمل مع مكونات مادية مختلفة أيضا.

إن المفسر في فورث بسيط جدا إذ يحتاج فقط إلى ثلاثة مؤشرات ومسجلين ووحدة الحساب والمنطق (ALU) لكي يعمل بكفاءة ولهذا السبب يمكن أن يكتمل الموديل البرمجي بسرعة كبيرة.ويمكن ببساطة تغييره عند تغيير (مجموعة الايعازات) في التصميم.ومن يعمل مع فورث يدرك أنها لغة جيدة في كتابة التطبيقات وكذلك عند استخدامها كلغة تعريفية للمكونات المادية الافتراضية.

المصادر

1) Alsulaiman, Shamil A., "Implementation of the Forth system on the IBM PC/XT microcomputer, Msc., Thesis collage of science, Computer Dept., Mosul University, April, 1990.
2) Brenton, s., phillips, s., "Dr. Dobb's tool book of Forth", volume 1&2, M&T Publishing Inc, 1987.
3) Brian Dipert, Getting a handle on VHDLs, EDN, May, 1998.
4) Clive "Max" Maxfield, Hug an XOR gate today: An introduction to Reed-Muller Logic., EDN, March, 1996.
5) Derick, M., Baker L., "Forth encyclopedia", Mountain View press Inc, 1982.
6) Douglas J Smith, VHDL and Verilog fundamentals- expressions, operands and perators, EDN, April, 1997.
7) Feierbach, G., Thomas, P., "forth tools and applications", Reston Publishing Company Inc., Prentice-Hall Company, Reston, Virginia, 1985.
8) Williams, Gregg, "Threads of a FORTH Tapestry"", Copyright BYTE Publications Inc., Distributed by Forth Interest Group, 1980.
9) Harris, K. "Forth Extensibility", Copyright BYTE Publications Inc., Distributed by Forth Interest Group, 1980.
10) Johan Sandstrom, VHDL & Verilog Syntax & Semantics Handbook. Integrated System Design Magazine, January, 1996.
11) Johan R., Hart, Forth as a VHDL, Testra Corporation, EDN, May, 2000.
12) Loeliger, P., "Threaded interpretive language", Byte Publication Inc., McGraw-Hill, 1981.
13) Mullen, J., "Flexible test environment", Forth Dimension, vol. 9, no. 2, pp. 9-11, 1987.
14) Steven H. Leibson, Vhsic Hardware Description Language, EDN, March, 1989.
15) Troy Scott, Adopting VHDL for PLD design and simulation. EDN, April, 1998.