

## **ABSTRACT**

The concept of portable code or machine independence has always had the interest of software developers. The current machine independent languages suffer from the low performance and high system resources consumption making them poor candidates for high performance applications. This thesis proposes a new model of machine independence. It aims to developing a new machine independence model whose performance and resource consumption is comparable to that of machine dependent machine languages.

The current solutions of machine independence are surveyed. The survey involved a study of existing software and hardware solutions. The advantages and disadvantages of the languages such as Java and .NET are discussed.

The new mode of portability involves the implementation of a hardware translation unit (HTU) that is installed on the user's machine. The HTU performs the conversion from an intermediate machine language (IML) to the native code of the resident CPU. The IML is a newly designed machine language that is easy to translate at runtime that resembles the machine code of an actual processor.

The HTU is designed as a memory controller and consists of four sub modules. Memory Interface Module (MIM), Processor Interface Module (PIM), Controller Module (CM) and Decoder and Translation Module (DTM) are the four main components of the interpreter.

The design has been simulated on ModelSim simulator for an x86 processor ZET. The synthesis and timing results are generated for a Xilinx Virtex 6 with best time of 5.4ns (184MHz). The number of instructions per second with translation is 2.9 Mega.

## المستخلص

مفهوم سهولة إنتقال البرمجيات بين الأنواع المختلفة من اجهزة الحاسبات وعدم تبعية البرمجيات للأنواع المختلفة من الآلات كان دائماً في مقدة إهتمامات مطوري البرمجيات. لغات البرمجة المستقلة المستخدمة حالياً تعاني من انخفاض الأداء وارتفاع استهلاك موارد النظام مما يجعلها اغير مرشحة للتطبيقات عالية الأداء.

هذه الرسالة تقترح نموذجاً جديداً للإستقلال عن نوعية الأجهزة في ذات الوقت الذي يحافظ فيه على استهلاك الموارد و يحصل على أداء مشابه للغات التابعة لنوع الآلات.

للوصول إلى أفضل شكل للنموذج المقترح، تم عمل مسح شامل لكل الحلول الحالية للغات الغير تابعة لنوع الجهاز. تضمن المسح دراسة عن البرمجيات والمكونات الصلبة المستخدمة في الحلول وتم مناقشة مزايا وعيوب لغات مثل ومايكروسوفت .NET و Java.

الطريقة جديدة من عدم الإعتمادية تنطوي على تنفيذ وحدة الترجمة (HTU) المثبتة على جهاز المستخدم. تقوم HTU بتنفيذ التحويل من لغة الآلة الوسيطة (IML) إلى اللغة الأصلية الخاصة بوحدة المعالجة المركزية المستخدمة. IML هي لغة آلة مصممة بواسطة الباحث بحيث أنه من السهل أن تترجم في وقت التشغيل التي لغة التجميع الخاصة بالمعالج الفعلي المستخدم. تم تصميم HTU كوحدة تحكم بالذاكرة وتتألف من أربع وحدات فرعية: وحدة واجهة الذاكرة (MIM)، وحدة واجهة المعالج (PIM)، وحدة التحكم (CM) ووحدة فك التشفير والترجمة (DTM). تم إستخدام لغة تصميم المكونات الصلبة عالية الكثافة (VHDL) لتصميم جهاز وحدة البرمجة.

تم محاكاة النظام بإستخدام برنامج ModelSim لمعالج x86-ZET وتم توليد كمية الموارد والسرعة الخاصة بالنظام عند إستهداف مصفوفة البوابات المبرمجة حقيقياً (FPGA) من نوع Virtex-6 من شركة Xilinx وتم الحصول على أفضل وقت يساوي 5.4ns أي ما يوازي 184MHz. عدد الأوامر المنفذة مع الترجمة هي 2.9 مليون أمر في الثانية.