

# 重定向编译器的研究进展和趋势<sup>\*</sup>)

吴圣宁 李思昆

(国防科技大学计算机学院 长沙 410073)

**摘要** 在软硬件协同设计中,常常需要改变嵌入式处理器的体系结构,并评价其对系统各种优化目标的影响,以便产生高效的目标代码。可重定向编译技术正好能满足这一需求。可重定向编译器和传统编译器之间的本质区别在于前者要求编译器代码尽可能重用,以便辅助体系结构设计者探索设计空间。本文综述了已有的主要可重定向编译技术,并指出了所遇到的问题和困难。

**关键词** 协同设计,可重定向,编译器

## Retargetable Compilers: Status and Trends

WU Sheng-Ning LI Shi-Kun

(Computer School, National University of Defence Technology, Changsha 410073)

**Abstract** In the software-hardware codesign, it is often necessary to change the architectures of embedded processors, and evaluate the effect on various system optimization goals in time, in order to achieve global system objectives at last. Retargetable compilation technologies are to meet this requirement. The essential difference between a retargetable compiler and a traditional one is that the former requires compiler code reusable as much as possible, in order to adapt to the need of design-space-exploration. This paper surveys the main retargetable compilation technologies achieved, and it points out the problems and difficulties encountered.

**Keywords** Codesign, Retargetable, Compiler

## 1 引言

半导体工艺的发展和 EDA 工具的进步促进了嵌入式系统的加速发展。System-on-Chip (SOC) 技术的应用在降低产品成本和提高性能的同时,也向嵌入式系统体系结构设计者提出了挑战。SOC 技术可以把通用处理器、协处理器、ASIP、DSP 和各种类型的存储器等集成在同一块芯片上,形成一个异构系统,为设计者实现针对特定应用的最优化系统提供了非常广阔的体系结构选择空间。

嵌入式系统越来越复杂,本身有着低功耗、低成本、实时性等要求,不断缩短的产品生命周期要求更快的上市时间。

这就需要有一个有力的开发环境来帮助设计者快速地评价各种候选的 SOC 体系结构,并能快速地并行完成系统硬件、软件的实现。这样,一方面,传统的嵌入式系统的软硬件串行设计流程演变成了软硬件协同设计的方法学。为提高生产率和可靠性,软硬件协同设计方法学中的一个重要趋势是在嵌入式 SOC 中采用可重用的 Intellectual Property (IP) 核。另一方面,为辅助体系结构设计者探索设计空间 (Design Space Exploration, DSE), 评价其设计的合理性,需要及时为新的平台开发软件工具 (例如编译器、汇编器、仿真器、调试器等)。这是一件耗时的工作。为适应设计者频繁修改体系结构特征并对其及时评价,需要有一个自动产生软件工具的环境。可重定向编译器就是其中的一个重要组成部分。

基于体系结构描述语言 ADL (Architecture Description Language) 的 SOC 协同设计方法学已成为进行嵌入式 SOC

体系结构设计空间探索和优化的主流设计方法。但目前的重定向编译器普遍离既能针对嵌入式体系结构特征产生高效目标代码,也方便重定向的目标还有较大差距。

本文第 2 节给出了重定向的三个层次;第 3 节总结了 ADL;第 4、5 节分别介绍了几个典型的基于 ADL 的和其它的重定向编译系统;第 6 节指出了目前重定向编译器存在的问题;最后是结论。

## 2 重定向编译的层次

可重定向具有不同的层次,这些层次的不同在于它们能支持多少机器相关的代码优化,也在于重定向过程中用户参与的多少。

### 2.1 参数重定向

一个自动重定向编译器含有一组开关,合理地配置这些开关就可说明目标体系结构。编译器支持的所有体系结构基本上都内建在编译器框架中了。如果机器相关的优化所依赖的目标机器信息不能由配置开关来说明,则参数重定向不能支持这些优化。

### 2.2 用户重定向

在用户重定向框架下,编译器的产生器接收目标机器的描述,决定适用于此处理器的所有优化,然后自动构造编译器。这种重定向技术已经成功地用在编译的代码产生阶段:LBURG 代码产生器的产生器接收目标机器指令集体系结构 (instruction set architecture, ISA) 描述,然后自动构造代码产生器,此代码产生器再为表达式树进行优化的指令选择。

<sup>\*</sup>) 本文的研究得到国家自然科学基金项目 (90207019)、国家 863 计划课题 (2002AAIZ1480) 的资助。吴圣宁 博士研究生,主要研究领域为嵌入式系统编译器;李思昆 博士生导师,主要研究领域为微电子设计和工具。

用户重定向当前支持指令选择和非常有限的机器相关的优化。尤其是,不存在一个有效的机器描述形式,使得编译器的产生器能通过描述信息自动组织更复杂的机器相关优化算法,以充分利用目标机器的结构特征。

### 2.3 开发者重定向

开发者是指有编译器设计经验的人。允许开发者修改编译器,使其适合目标机器。显然,重定向和写一个新的编译器之间的界限有些模糊。开发者使用体系结构信息来改造优化算法,并为相应目标结构组织最有效的优化次序。

## 3 体系机构描述语言 ADL

ADL(Architecture Description Language)是描述软件和系统体系结构的语言<sup>[1]</sup>,用来驱动设计空间探索、硬件综合和编译器、汇编器、仿真器等软件工具的自动产生。就像基于 HDL 的 ASIC 设计流,基于语言的嵌入式 SOC 设计方法学带来不少方便,例如便于进行形式化验证和一致性检查,便于修改目标体系结构和存储系统组织以进行设计空间探索,可从单一的描述来自动产生软件工具包。

### 3.1 ADL 的分类

根据 ADL 的主要目的,可将结构描述语言分为五类:面向综合的、编译器产生的、仿真器产生的、验证的、其它的。

#### 3.1.1 面向综合的 ADL

MIMOLA<sup>[2]</sup>是德国 Dortmund 大学设计的 ADL,其实也是一种 HDL,用在 MSSQ 编译器和 RECORD 编译器中描述目标机器。处理器的结构用 MIMOLA 建模为 RTL 网表,MIMOLA 本身不含指令集和汇编语言信息。这种方法最大的优点是统一的 MIMOLA 描述可用于综合、仿真、测试产生、代码产生。

#### 3.1.2 面向编译器的 ADL

MADL(Mescal ADL)<sup>[3]</sup>是形式化的基于并发模型的体系结构描述语言,用以综合软件开发工具。它具有两层结构,即核心层和注释层。核心层基于一个灵活的操作状态机处理器模型,以便于在操作执行层描述并发性。注释层用来描述实现相关的或工具相关的信息。

Arche<sup>[4]</sup>是开放源码的基于 SystemC 的处理器体系结构描述语言。其关键特征是具有基于存储的协同验证机制,以检查模型的一致性、层次化的存储建模、可以和其它 SystemC IPs 集成,以及自动生成 SystemC 模拟器。

nML<sup>[5]</sup>是 TU Berlin 开发的 ADL,面向的主要处理器是 DSP 和 ASIP。处理器的行为(如指令集)用 nML 描述为属性文法。指令集的信息包括每条指令的行为(称为动作)、汇编格式、目标代码。nML 显式地支持若干寻址模式和指令集的层次化描述。nML 假定 FU 没有延迟,仅存储部件有,因此 nML 不支持多个时钟周期的功能部件(FU)和流水 FU。nML 没有显式地支持多字指令。指令间的资源冲突通过一种操作组合的语法隐式表示。

ISDL<sup>[6]</sup>是 MIT 开发的 ADL。ISDL 主要面向 VLIW,用于编译器、汇编器、仿真器的产生。ISDL 描述可翻译成可综合的 Verilog 代码。类似 nML,ISDL 主要描述处理器指令集。指令集描述包括每条指令的动作、汇编格式、代价(反映执行周期、代码大小、资源冲突等)和时序。没有显式描述流水线结构。资源冲突等造成的 ILP 约束通过一组布尔表达式规则来显式描述,当这些规则都满足时指令才有效。

MDES<sup>[7]</sup>是 UIUC 开发的 ADL,在 Trimaran 系统中支

持高性能处理器的设计空间探索和编译器的自动产生。MDES 描述处理器的结构和行为,描述信息基于高层的分类分为若干部分(例如格式、资源使用、延迟、操作、寄存器等)。但 MDES 只允许针对 HPL-PD 处理器系列的有限的可重定向。MDES 可描述存储系统,但只限于传统的层次,例如,寄存器文件、cache、主存。MDES 用层次化的形式通过显式的保留表来描述 ILP 约束。

CSDL<sup>[8]</sup>是由 Virginia 大学开发的用于 Zephyr 编译器系统的一组机器描述语言。CSDL 由四个语言组成:SLED、λ-RTL、CCL、PLUNGE。SLED 描述指令的汇编和二进制表示;λ-RTL 以寄存器传输的形式描述指令行为;CCL 描述函数调用规范;PLUNGE 是描述流水线结构的图形符号系统。

EXPRESSION<sup>[9]</sup>是由 UC Irvine 开发的 ADL,用于支持多种类型处理器体系结构的设计空间探索,从 RISC、DSP、ASIP 到 VLIW,其中包括各种类型和层次的存储系统组织。EXPRESSION 集成了处理器-存储器系统的结构和行为描述。结构描述包括各功能部件、存储部件、端口和它们之间的连接。流水线由组成流水线各段的部件及时序来描述。行为描述了指令集。每个操作以操作码、操作数和表示各操作域相对次序的格式来定义。每条指令可视为用操作来填充的空槽系列。用 EXPRESSION 可自动产生 ILP 编译器和仿真器。

#### 3.1.3 面向仿真的 ADL

LISA<sup>[10]</sup>是在 RWTH Aachen 开发的 ADL,支持 DSP 仿真器的产生。LISA 集成了结构和行为描述。LISA 的主要特征是对复杂的流水线的描述。每条指令由多个微操作组成,这些微操作定义为控制步中的寄存器传输。LISA 对一些流水线行为提供了显式支持。但 LISA 没有为 ILP 编译器提供显式的资源冲突描述。资源冲突用信号来模拟,这些信号捕获流水线上某些操作码的发生并触发相应的流水线控制。

#### 3.1.4 面向验证的 ADL

AIDL<sup>[21]</sup>是由 Tsukuba 大学开发的 ADL,用于高性能超标量处理器设计。AIDL 用时序逻辑描述流水线的时序行为,用于流水线行为的验证。AIDL 不支持软件工具包的产生。但 AIDL 描述可以用 AIDL 仿真器仿真,还可以翻译成可综合的 VHDL 代码。

### 3.2 ADL 的挑战和趋势

ADL 应既描述行为(指令集)也描述结构,以高效地自动产生高质量的软件工具包。

大多数 ADL 描述单处理器系统的处理器体系结构,并自动产生软件工具包。ADL 是 SOC 设计方法学的革新,但还有一些问题有待更深入的研究,例如形式化验证、综合、运行环境的产生和 IP 库的关系以及对更多类型 SOC 体系结构的支持等。

ADL 的形式化验证至今还没有广泛的研究。例如,随着处理器 ILP 的增加和更深流水线的采用,就很有必要在设计的前期阶段对流水线进行形式化验证。ADL 描述一致性的形式化验证也很重要。例如,ADL 常包含行为信息和结构信息,就需检查这两者间的一致性,而且应检查结构中没有冗余的成分。

ADL 描述应支持从中综合出处理器,因为设计处理器通常非常复杂且耗时。

一些 ADL 支持编译器和仿真器的自动产生,但没有 ADL 支持产生像 RTOS(Real-Time Operating System)这样

的软件运行环境。RTOS 对系统性能有显著的影响,RTOS 的自动产生和优化是将来的一个重要的研究方向。

为了实现 SOC 体系结构的有效的 DSE,ADL 描述应是可重用的。用 ADL 描述 SOC 结构的机制应标准化,封装设计属性(如性能、功耗、体积等),并把这些 SOC 结构的 ADL 描述存在 IP 库中。

将来的 SOC 可以包含多个异构的多线程处理器、ASIC 和各种类型的存储器,并采用不规则的互联。现在的大多数 ADL 只描述单处理器系统的处理器结构,应把它们加以扩展,以描述更广泛的 SOC 体系结构。

## 4 基于 ADL 的重定向编译器

下面列举几个具有代表性的重定向编译系统。

### 4.1 CHES

CHES<sup>[11]</sup>是定点 DSP 的重定向代码产生环境,它建立在行为/结构混合处理器模型上。

CHES 编译器的关键特征是:1)重定向性。CHES 使用单一的处理器模型表示处理器的所有行为,这个模型称为 ISG(Instruction Set Graph),可从高层处理器描述语言 nML 自动构造。2)代码质量。具有针对 DSP 特征的优化技术,首先代码选择和寄存器分配使用了 ISG 模型中的结构信息,其次使用全局优化,最后对 phase coupling 采用了有效的机制。

### 4.2 RECORD

目标处理器模型用 MIMOLA HDL 描述,MIMOLA 相当于 VHDL 的子集。RECORD 系统<sup>[12]</sup>接受处理器的行为和结构模型,RECORD 的一个重要特征是:目标机器的描述可以有不同抽象层次;可以从 RT 层网表到指令集描述。接受 RT 网表是因为一些 ASIP 就是在这一层设计的。这种方法提供了 ECAD(网表)和编译器(指令集)之间的桥梁。

### 4.3 AVIV

MIT 的 AVIV<sup>[13]</sup>是为不同指令集体系结构的处理器产生优化机器码的重定向代码产生器。机器描述采用了 ISDL HDL,包括指令集描述和一些结构信息。AVIV 为目标处理器产生优化的体积最小的目标代码,它主要面向 ROM 容量要求苛刻的 SOC 和具备 ILP(Instruction Level Parallelism)特征的处理器,包括 VLIW 体系结构。

### 4.4 IMPACT

多指令发射处理器性能的发挥受到为并发硬件产生有效代码的编译器的能力的约束。IMPACT (Illinois Microarchitecture Project utilizing Advanced Compiler Technology)主要面向指令级并行<sup>[14]</sup>。IMPACT 是一个复杂的编译器系统,有许多优化、仿真、profile、代码产生的工具,支持多种体系结构(MIPS,SPARC,AMD,x86 等)。通过用 IMPACT C 编译器试验,来分析多指令发射处理器性能,分析体系结构设计问题。

### 4.5 SPAM

SPAM<sup>[15]</sup>是面向 DSP 的重定向编译器,采用了开发者重定向。SUIF(Stanford University Intermediate Format)编译器实现 SPAM 编译器的前端,并产生没有优化的中间表示形式 IR。然后,对此 IR 进行一系列优化,以减小代码体积。SPAM 编译器的后端,称为 SWIF,由两部分组成:一部分是一组数据结构,用以封装各种源程序表示;另一部分是一组算法,通过分析这些数据结构来产生代码和进行机器相关的优化。

## 4.6 V-SAT 系统

关键组成部分包括 EXPRESSION 体系结构描述语言、分析和评价体系结构的仿真器以及一个图形用户接口以方便用户的描述和分析。用户通过图形界面对体系结构进行可视化描述,然后转换成 EXPRESSION<sup>[16~18]</sup>描述。EXPRESSION 支持复杂存储系统的描述。产生的重定向编译器可为并发的流水结构的处理器产生优化代码。

## 4.7 GCC

GCC<sup>[22]</sup>是一个支持多种平台的编译系统,主要由三部分组成:前端、后端及与机器相关的机器描述。机器描述由两个部分组成:一个为描述目标机各种参数的宏定义头文件 machine.h;另一个为描述目标机指令集的 machine.md。GCC 系统支持的体系结构特征非常丰富,以致于机器描述非常复杂,X86 的 machine.md 文件有两三万行之多,对 ARM 的描述也有七千多行。

## 5 其它的重定向编译系统

### 5.1 NCI

NCI(The National Compiler Infrastructure project)<sup>[19]</sup>是要为编译器研究者提供一个共同的合作平台,由 SUIF 和 Zephyr 组成。SUIF 编译器框架建立在 SUIF(Stanford University Intermediate Format)并行编译器的基础上,其主要目标是开发一个可扩展的系统,以支持并行、面向对象程序语言、优化等编译问题的研究。

### 5.2 Machine SUIF

Harvard University 的 Machine SUIF<sup>[20]</sup>建立在 Stanford SUIF 编译器框架之上,是一个灵活、可扩展的框架,用以构造编译器后端。它通过 OPI(The Optimization Programming Interface)隔离了机器相关的信息,因此编译优化代码可以方便地重用。

## 6 嵌入式系统存在的编译问题

### 6.1 编译器重定向

很多针对嵌入式处理器特征的编译系统,虽然能产生高效率的目标代码,但只能适用狭窄的处理器类型,且不便重定向。目前的重定向编译器往往试图以一种统一的 ADL 来描述机器的所有体系结构特征,在系统特征复杂化时很难描述清楚,因此难以产生高效目标代码。机器描述难学难用。

### 6.2 编译器的约束代码产生

现有的约束代码产生往往只考虑一种约束,而嵌入式系统常有性能、体积、功耗、成本等多种约束,编译器应能在各种约束目标同时作用下产生优化代码。

**结论** 本文介绍了重定向编译器的发展现状,给出了编译器重定向的三个层次,总结了一些典型的体系结构描述语言和重定向编译系统的重要特征,指出了目前重定向编译器领域的研究所遇到的问题。

基于体系结构描述语言的设计方法学有着良好的应用前景。从开发者重定向层次开始,需要研究新的体系结构描述语言和重定向编译器结构。

## 参考文献

- 1 Avionics Avionics Architecture Description Language (AADL) Seminar, Toulouse, France, October 2002
- 2 Bashford S, Bieker U, Harking B, et al. The MIMOLA Language Version 4.1. University of Dortmund, 1994

- 3 Wei Qin, Rajagopalan S, Malik S. A Formal Concurrency Model Based Architecture Description Language for Synthesis of Software Development Tools. In: LCTES'04, 2004
- 4 Rigo S, Araujo G, Bartholomeu M, et al. ArchC: A SystemC-Based Architecture Description Language. In: Proceedings of the 16th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'04), 2004
- 5 Freericks M. The nML machine description formalism. Fachbereich Informatik, TU Berlin, 1991
- 6 Hadjyiaanis G, Hanono S, Devadas S. ISDL: An instruction set description language for retargetability. In: Proc. of 34th DAC, 1997
- 7 Gyllenhaal J C. A machine description language for compilation. [Master's thesis], Dept. of ECE, UIUC, 1994
- 8 <http://www.RCS.virginia.edu/zephyr>, The Zephyr compiler infrastructure
- 9 Halambi A, Grun P, Ganesh V, et al. EXPRESSION: A language for architecture exploration through compiler/simulator retargetability. In: Proc. of DATE, 1999
- 10 Pees S, Hoffmann A, zivjnovic V, et al. LISA: Machine description language for cycle-accurate models of programmable DSP architectures. In: Proc. of 35th DAC, 1999
- 11 Marwedel R, Goossens G, et al. Code Generation for Embedded Processors, Kluwer Academic Publishers, 1995
- 12 Leupers R. Retargetable generator of code selector from HDL processor models. In: European Design and Test Conference (ED&TC), 1997
- 13 Hanono S Z. Aviv: A Retargetable Code Generator for Embedded

- Processors; [Ph. D thesis], Department of Electrical Engineering and Computer Science, MIT, 1999
- 14 Chang P P, Mahlke S A, Chen W Y, et al. IMPACT: An Architectural Framework for Multiple-Instruction-Issue Processors. In: Proceedings of the 18th Annual Int'l Symposium on Computer Architecture, Toronto, 1991. 266~275
- 15 Sudarsanam A. Code Optimization Libraries for Retargetable Compilation for Embedded Digital Signal Processors; [PhD Thesis], Princeton University Department of EE, 1998
- 16 Mishra P, Mamidipaka M, Dutt N. Processor-Memory Coexploration Using an Architecture Description Language. ACM Transactions on Embedded Computing Systems, 2004, 3(1): 140~162
- 17 Kejariwal A, Mishra P, Astrom J, et al. HDLGen: Architecture Description Language driven HDL Generation for Pipelined Processors; [Technical Report #03-04]. Center for Embedded Computer Systems, University of California, Irvine, CA, USA, February, 2003
- 18 Mishra P, Kejariwal A, Dutt N. Rapid Exploration of Pipelined Processors through Automatic Generation of Synthesizable RTL Models, Rapid System Prototyping, San Diego, 2003
- 19 <http://www.cs.virginia.edu/nci/>, The National Compiler Infrastructure Project
- 20 <http://www.eecs.harvard.edu/machsui/index.html>, Machine SUIF
- 21 Morimoto T, Yamazaki K, Nakamura H, et al. Superscalar processor design with hardware description language AIDL. In: Proc. of APCHDL, 1994
- 22 <http://gcc.gnu.org/index.html>

(上接第 169 页)

增加而增加。图 9 使用 ROC 曲线来反映检测的精度,从图上可以看出在不同的 TP 值下,FP 始终处于一个很低的值,这就证明了使用阴性选择算子充当过滤器能够很好地抑制 FP 值的增加。

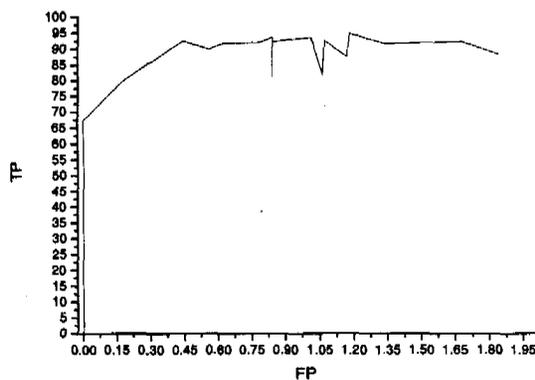


图 9 反映检测性能的 ROC 曲线

**结论和展望** 本文吸收了生物免疫学中免疫识别的灵感,组合了阴性选择算子和遗传算法来实现性能异常检测。在算法中多次使用阴性选择算子充当过滤器,尤其是把它添加到遗传算法中,有效地抑制了 FP 值的增加;采用了新的部分匹配规则来度量两个个体之间的距离;使用模糊逻辑产生模糊集,更好地反映正常和异常的界线,最后使用两组数据进行了仿真实验。实验的结果表明,算法具有较高的检测精度。本文的工作只是实现了性能异常的检测,并不能对于异常的程度给予量化并且确定软件衰退的原因。未来的工作将研究这些问题,为制定性能恢复策略提供依据。

### 参考文献

- 1 Garg S, Moorsel A V. A Methodology for Detection and Estima-

- tion of Software Aging[A]. In: Proc. of 9th Intl Symposium on Software Reliability Engineering[C], Paderborn, Germany, Nov. 1998. 282 ~ 292
- 2 Hansen J P, Siewiorek D P. Models for time coalescence in event logs[A]. In: Proc. of 22nd IEEE Intl Symposium on Fault-Tolerant Computing[C], 1992. 221 ~ 227
- 3 Iyer R K, Young L T. Automatic recognition of intermittent failures; An experimental study of field data[J]. IEEE Transactions on Computers, 1990, 39(4): 525~537
- 4 Ye Nong, Chen Qiang. An Anomaly Detection Technique Based on A Chi-square Statistic for Detecting Intrusions into Information Systems[J]. Quality and Reliability Engineering International, 2001, 17(2): 105 ~ 112
- 5 Lane T, Brodley C E. An Application of Machine Learning to Anomaly Detection[A]. In: Proceedings of the 20th National Information Systems Security Conference [C], Baltimore, MD. Oct. 1997. 366~377
- 6 Forrest S. Self-nonsel self discrimination in a computer[A]. In: Proceedings of IEEE Symp on Research in Security and Privacy[C], 1994. 202~212
- 7 McCoy D F, Devarajan V. Artificial immune systems and aerial image segmentation[A]. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics[C]. Orlando, Florida, 1997. 867~872
- 8 Kim J, Bentley P. Towards an Artificial Immune System for Network Intrusion Detection; An Investigation of Clonal Selection with a Negative Selection Operator[A]. In: Proceedings of on Evolutionary Computation (CEC-2001) [C], Seoul, Korea, 2001. 1244~1252
- 9 Kim J, Bentley P. An evaluation of negative selection in an artificial immune system for network intrusion detection[A]. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)[C], San Francisco, California, USA, 2001. 1330 ~ 1337
- 10 Caudell T, Newman D. An adaptive resonance architecture to define normality and detect novelties in time series and databascs [A]. In: Proceedings of (Portland, Oregon)[C], 1993. 166 ~ 176
- 11 Murphy P M, Aha D W. UCI Repository of machine learning databases, 1992
- 12 孙瑞祥. 进化计算与智能诊断[D]. 西安交通大学, 2000