

Joseph A. Fisher : Very Long Instruction Word Architectures and The ELI-512

Proc. of the 10th Annual International Symposium on Computer Architecture, pp.140-150 (June 1983)

本論文は、「科学技術計算では、トレーススケジューリング (trace scheduling) という広域コード最適化手法によって、従来のマシン命令機能レベルに存在する並列処理可能性 (後に、「命令レベル並列性 (ILP: Instruction Level Parallelism)」という) を見つけ、それらを VLIW (Very Long Instruction Word) と呼ぶ 500 ビット以上のマシン命令に埋め込んで同時実行すれば、既存の逐次マシンの 10 ~ 30 倍の性能向上が得られる」と主張している。ILP は 1980 年代から 1990 年代にかけての RISC (Reduced Instruction Set Computer) の出現およびその発展に中心的な役割を果たすが、本論文は「ILP をアーキテクチャの設計で活用する」ことを先導した。

本論文は、コンピュータアーキテクチャに関する年次の国際会議 ISCA'83 で発表され、異彩を放った。それは、「ELI-512」という VLIW アーキテクチャのデビューのせいではなく、「トレーススケジューリング」というコンパイル (コード最適化) 手法がアーキテクチャ論文の主役だったからである。

ISCA は当時は唯一といえる「コンピュータアーキテクチャに関する国際会議」であった。毎年発表されるほとんどの論文が主として「ハードウェアアーキテクチャ」すなわち「ソフトウェアから見たハードウェア構成方式」の有効性の主張を中心に据えている。ハードウェアアーキテクチャと不可分のコンパイラ (特にコード最適化を代表とするマシン依存機能) へ言及する論文はごく少数で、言及があったとしても、それはハードウェアアーキテクチャを中心に論述する論文ではおまけの感が否めなかった。

それらに比して、本論文は、コンパイラが行うコード最適化手法としてのトレーススケジューリングの有効性を論文の中心に据えた、トレーススケジューリングで最適化したコードが有効に機能するハードウェアアーキテクチャ (VLIW) を (逆) 提案した、の 2 点で異色だっ

た。当時はもとより現代でも、ハードウェアアーキテクチャつまりマシンの設計・開発が先行し、コンパイラの実装は後続する、あるいは、マシンの設計過程とは切り離す。本論文で、Fisher はこれに異を唱え、「アーキテクチャ設計とは、ソフトウェア (コンパイラ) とハードウェアを一体として設計する」こととしたのである。

トレーススケジューリングそのものは、水平型マイクロ命令列 (マイクロコード) の広域コード最適化手法として Fisher 自身がすでに (1981 年) 発表済みであった。また、VLIW アーキテクチャも、マイクロアーキテクチャとしては QA-2 (Fisher が本論文を発表した ISCA'83 のセッション: Multiple Functional Unit Processors で筆者らが発表) が採用していた。しかし、Fisher は「トレーススケジューリングのターゲットを (マイクロ命令コードではなく) マシン命令コードと (見な) し、VLIW を (マイクロアーキテクチャではなく) マシンアーキテクチャと (見な) し、さらにそれらをトータルアーキテクチャとした」点で、後に隆盛する ILP プロセッサアーキテクトの先駆者となった。

本論文によって、コンピュータアーキテクトたちは、「ハードウェアとソフトウェアとの機能分担方式」というコンピュータアーキテクチャの定義 (原点) を思い出すことになる。アーキテクチャ史上のトレードオフ論争のうち、RISC 対 CISC 論争に引き続いて 1990 年代に起こり現代も続くスーパスカラ (superscalar) 対 VLIW 論争は、コンピュータシステムでのソフトウェアで実現する機能とハードウェアで実現する機能とのトレードオフ論争が原点である。この論争では、本論文では「コンパイラが主導すべき」とする ILP の検出機能の実現ウェア (ware) が直接の論点となっている。

(平成 17 年 3 月 13 日受付)

柴山 潔 / 京都工芸繊維大学工芸学部電子情報工学科
shibayam@kit.jp