

並列論理型言語による 1F-8 計算機性能評価支援システム

蛭田 規之 野田 泰徳 平野 達郎 植村 昌俊
【沖電気工業(株)】

1. はじめに

計算機の高性能化への要求は益々強まるばかりであり、それに伴い高度な制御機構を持つハードウェアの設計が多くなってきている。そこでは、設計の難度が高くその負担も大きくなっているため、高性能のハードウェアを実現するための設計支援が望まれている。

筆者らは、パイプライン処理方式のハードウェアアーキテクチャ及びその上での命令機能の実現方式に関して、性能面からの評価を行う際に有用な性能評価データの導出を行うシステムについて検討を行ってきた。¹⁾ 今回、並列論理型言語を用いてこのシステムの試作を行ったので、その概要をここに報告する。

2. 計算機の性能評価

パイプライン処理方式計算機は、各命令がより小さな処理単位であるステージに分解され、複数命令のステージの処理を並列に実行することにより処理効率の向上が図られている。そして、この計算機の処理性能は、命令の並列実行度に大きく依存しており、パイプライン動作の乱れにより低下する。

この動作の乱れは、パイプラインを構成する各演算器における、命令の実行待ち時間の発生やジャンプ等の特殊な命令の実行によって生じ、その原因を手で解析するのは困難になりつつある。そこで、アーキテクチャ設計段階の設計結果である対象計算機のデータバス構造や命令機能の実現法をもとに、パイプライン動作の乱れが生じる原因を解析し、設計の改善を行う作業に対する設計支援ツールが望まれることになる。

3. システム概要

本システムは、評価対象計算機のデータバス構造及びその上での命令機能の実現方法をもとに、対象計算機内のデータの流れに着目したシミュレーションを行い、命令の並列実行状況を示した動的評価データとそれを統計的に加工した静的評価データを出力する。

3.1 入力

(1) テストプログラム

対象計算機の評価に用いるテストプログラムで、命令のリストで表現される。この命令はリスト順に処理される。

(2) 命令定義

対象計算機の命令の動作を実現するためのデータバス上のファシリティの利用状況および他の命令実行に対する制御動作が、レジスタ転送レベルでステージ単位に記述される。

(3) ファシリティ宣言

対象計算機のデータバスを構成するファシリティの宣言を行う。

(4) アーキテクチャパラメータ

対象計算機のパイプライン段数、マシンサイクルの周期等を設定する。

3.2 機能

(1) 命令の実行

パイプラインの各演算器は命令ストリームからそれぞれ命令をフェッチし、その命令に対応する命令定義を参照してステージ毎の動作を行う。ここで、命令定義に他の命令の実行を制御する記述がある時、その制御情報は制御先の演算器へ知らされる。

(2) ファシリティ利用競合の解消

同一のマシンサイクルにおいて、異なる命

令から同じファシリティの利用要求が発生した時、システムは優先されるべき命令を認識し、その命令のステージのみの実行を行い、競合した他の命令のステージの実行を延期する。ここで、ファシリティの競合条件は、各ファシリティのタイプ毎に設定されている。

(3) 確率的条件の判定

コンディショニング命令の条件部の成功/不成功は、予め設定された割合に従って確率的に決定される。また、キャッシュアクセスのヒット/アンヒットも同様の方式で決定され、各々のケースの所要アクセスタイムの時間中、他のファシリティへの利用要求は継続される。

3.3 出力

(1) 動的評価データ

テストプログラム実行時の、演算器単位の命令実行状況が図1に示すようにタイムチャート形式で出力される。この時、以下の情報が付加される。

- ・競合により命令実行に乱れが生じた時場合
その対象となったファシリティ名
- ・他の命令からの実行制御により乱れが生じた場合、その制御を発した命令及びステージ名
- ・コンディショニングの条件判定結果、
キャッシュアクセスの状態等

図1に動的評価データの例を示す。

(2) 静的評価データ

対象計算機の平均MIPS値、平均並列実行度、各ファシリティの競合度合等の定量的評価データを出力する。

4. 並列論理型言語による実現

本システムは、並列論理型言語GHC²⁾によって記述されている。GHCは並列に存在するプロセスを動的に生成し、各プロセス間で通信や同期の制御を行うことが可能であるここでは、この特徴を利用し、パイプラインの各演算器をプロセスに対応させてモデル化を行い、演算器間の通信を実現した。

<<<< SIMULATION RESULT >>>>			
T :	PIPE_1	PIPE_2	PIPE_3
	\$\$\$ unhit \$\$\$	\$\$\$ hit \$\$\$	\$\$\$ hit \$\$\$
001:loadMA_fet1	f_wait(1)	f_wait(1)	
:	[ic,mu]	[ic,mu]	
002:loadMA_fet2	f_wait(1)	f_wait(1)	
:	[ic,mu]	[ic,mu]	
003:loadMA_fet3	f_wait(1)	f_wait(1)	
:	[ic,mu]	[ic,mu]	
004:loadMA_fet4	loadMB_fet	f_wait(2)	
:		[ic,mu]	
005:loadMA_dec	f_wait(1)	addABA_fet	
:	[de]		
	\$\$\$ unhit \$\$\$		
006:loadMA_ex1	loadMB_dec	f_wait(2)	
:		[de]	
	\$\$\$ hit \$\$\$		
007:loadMA_ex2	f_wait(1)	addABA_dec	
:	[mu, sb, dc]		
008:loadMA_ex3	f_wait(1)	f_wait(1)	
:	[dc, bl, rf-pl]	[rf-a, rf-b]	
	\$\$\$ hit \$\$\$		
009:moveAC_fet	loadMB_ex1	f_wait(2)	
:		[rf-b]	
	\$\$\$ hit \$\$\$		
010:moveAC_dec	jump_fet	addABA_ex1	
011: f_wait(3)	jump_reset	addABA_ex2	
:	[rf-a, a]		
		\$\$\$ hit \$\$\$	
012:moveAC_ex1	f_wait(3)	addCAC_fet	
:	[ic,iu]		
	\$\$\$ hit \$\$\$		
013:moveAC_ex2	addCAC_fet	addCAC_dec	
:			
014:addCAC_fet	addCAC_dec	addCAC_ex1	
015:addCAC_dec	addCAC_ex1	addCAC_ex2	
:		\$\$\$ hit \$\$\$	
addCAC	1	AC_ex2	AC_f

図1 動的評価データの例

5. おわりに

アーキテクチャの性能評価データの導出を行なうシステムを開発したことにより、設計者の経験と知識により設計改善を行なう手法を獲得することが可能となった。従って、導出された評価データをもとに、性能向上を図るための、データベース構造や命令制御方式に対する設計改善方法の指摘を行うエキスパートシステムについて、今後検討を行ってきたい。

尚、本研究は第5世代コンピュータプロジェクトの一環として行われた。

【参考文献】

- 1) 蛭田 他: 並列論理型言語による性能評価システムの検討, 情報処理学会第32回全国大会(1986)
- 2) Kazunori Ueda: "Guarded Horn Clauses", ICOT Technical Report, TR-103 (1985)