前导1预测算法的设计与实现

李 星 胡春媚 李 勇 李振涛
(国防科技大学计算机学院 长沙 410073)

摘 要 前导1预测(Leading One Prediction,LOP)算法常被用在浮点数的加减运算中,它能与尾数加法器并行工作,从而加快了尾数加法器计算结果的规格化过程,同时,这种方法会带来最多1位的误差。根据对误差的处理方式不同,将预测算法分成了3类,并详细介绍了其中的串行纠错前导1预测算法的具体结构,对其关键的组成部分在算法上进行了选择和优化。它与并行纠错LOP以及传统前导1检测(Leading One Detector,LOD)的逻辑综合的实验结果表明,该算法取得了面积、功耗和延时之间的较好均衡。在实际的应用中,该算法成功地运用在了工作频率为1GHz的三站式双通路(Two-Path)浮点加法器中。

关键词 前导1预测,前导1检测,纠错,规格化
中图法分类号 TP303 **文献标识码** A

Design and Implementation of Leading-One Prediction

LI Xing HU Chun-mei LI Yong LI Zhen-tao
(School of Computer Science, National University of Defence Technology, Changsha 410073, China)

Abstract Leading-one prediction(LOP), which is often used in floating-point addition/subtraction, can operate in parallel with the adder and reduce the delay in the normalization shift. However, this prediction might generate one-bit-error. Three different LOP architectures were classified by the methods handling the one-bit-error. Among that, the LOP architecture with serial correction was described in detail. At the same time, serial correction's key components in the algorithms were optimized. Through the synthesis experiments of LOP architecture with concurrent correction, serial correction and traditional leading one detector(LOD) method, we found that serial correction method has the best performance balancing area, power and delay. It is successfully used in two-path floating-point adder which is operated in 3-cycle pipeline with a 1Ghz clock frequency.

Keywords Leading one prediction, Leading one detection, Correction, Normalization

1 引言

现代数字信号处理器(DSP)对大量数据的处理能力以及实时性要求越来越高,目前的DSP已经能达到1GHz以上的频率,这就要求运算单元的运算速度很快,而浮点加法器在运算单元中占有非常重要的地位,其速度的高低直接影响运算单元的性能。因此设计高速的浮点加法器对DSP整体性能的提高至关重要。

在以IEEE754为标准的浮点数的运算中,一般包括5个步骤:指数相减、对阶移位、尾数加减、规格化移位以及舍入,如图1所示。

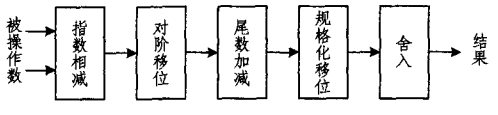

图1 浮点数的一般计算过程

其中,在进行规格化移位操作时,需要判断出尾数加/减法器计算结果中的第一个"1"的位置,即要计算出规格化操作所需要的移位量。本文的设计要求是3拍计算完成单/双精度浮点数的加减运算,并实现1Ghz工作频率的设计目标。本文对多种前导1算法进行了研究,实现了4种不同的算法结构,并对它们的延时、面积和功耗进行了评估。

本文首先对传统前导1检测算法和前导1预测算法分别进行了简单的介绍;其次,根据对1位误差的不同处理方法,将前导1预测算法的实现进行了分类,并介绍了4种不同的算法结构;然后,重点对串行纠错的前导1预测算法的实现进行了详细阐述;最后,本文采用逻辑综合实验的方法,对3种不同的典型算法进行了对比,确定了采用串行纠错的前导1预测算法作为最优的实现方案。

2 前导1算法概述及其结构

2.1 前导1算法概述

在传统的浮点算法中,前导1检测模块需要在尾数加法器计算出结果后,才能判定出规格化操作所需的移位量,这种

到稿日期:2012-07-20 返修日期:2012-10-22 本文受国家自然科学基金(60906014)资助。

李 星(1986—),男,硕士生,主要研究领域为高性能微处理器设计,E-mail:lixing.0425@163.com;胡春媚(1975—),女,副教授,主要研究领域为数字集成电路、高性能微处理器设计;李 勇(1969—),男,博士,副教授,主要研究领域为高性能DSP与SOC设计;李振涛(1976—),男,助理研究员,主要研究领域为高性能DSP与SOC设计。

加减法器与LOD串行化的操作大大增加了浮点加减运算的延时，如图2(a)所示。

为了能实现高性能的浮点加法器，大多数的浮点单元采用了前导1预测的算法(LOP)，如图2(b)所示，这种预测算法能直接从两个尾数中预测出规格化所需要的移位量，从而使前导1位置的判定能同尾数加减运算并行执行。

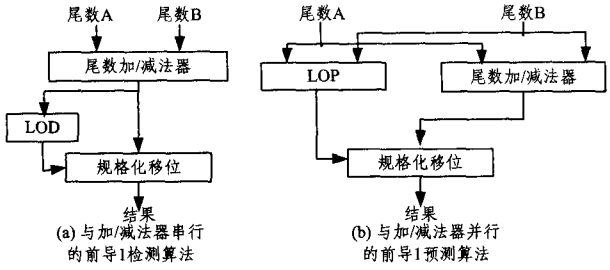

图2 浮点加法器中的前导1判定逻辑模块

2.2 前导1预测算法的不同结构

关于前导1预测算法已经有了很多的研究[1-10]。文献[3]提出了一种快速的前导1预测电路，但它是建立在尾数计算结果为正数的基础上的预测算法，所以该电路必须在计算之前比较两个尾数的大小并调换两个尾数的位置，这无疑增加了加法器的延时。除此之外，大多数的前导1预测算法在不比较两个尾数大小的前提下，无论计算结果是正值还是负值，均能对前导1的位置进行预测。文献[4]总结并比较了典型的不同算法结构的前导1预测算法，之后的关于前导1预测算法的大多数文章都是在这几种典型的算法结构上进行了优化或改进。前导1预测算法会带来最多1位的误差，根据预测后对可能的1位误差处理方式不同，本文将几种典型的算法结构分成了4类不同的前导1预测算法结构。

第一种结构如图3所示，其中的纠错检测树用来实现1位误差的判断，该模块与前导1检测和加法器并行工作。

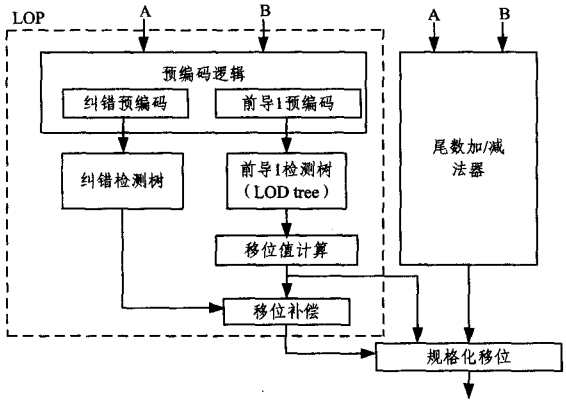

图3 基于纠错检测树的并行纠错检测前导1预测模块的总体结构

从算法结构上可看出，此前导1预测模块要同时产生出纠错预编码串和前导1预编码串，然后被送入纠错检测树进行误差的判断和送入前导1检测树进行前导1位置的判定。对于误差的判断，纠错检测树会产生一个纠错移位的标识信号，并用这个信号来控制在规格化移位时是否要再额外移一位。虽然这种方法消除了在规格化之后进行纠错检测所带来的延时，但由于其算法较复杂，电路的面积和功耗会变得很大，在文献[1]中已经提到，这种并行的误差纠错检测逻辑的大小约占整个前导1预测(LOP)逻辑的70%；文献[7,8]对纠错检测树和纠错预编码进行了优化，其电路面积分别比文献[1]中的算法结构要小，但它们的延时较长。

第二种结构如图4所示，它是利用来自加法器的进位来检测是否要补偿1位。

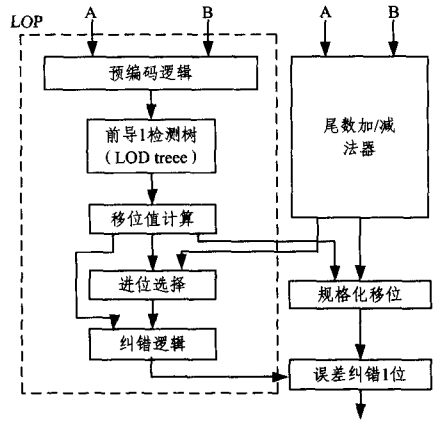

图4 基于进位判定的并行纠错检测前导1预测模块的总体结构

由于进位选择逻辑需要尾数加减法器产生的进位信号和LOD计算的移位值，使得纠错逻辑产生的纠错信号比较慢，相比第一种结构，该算法带来的延时将会增大。

第三种结构如图5所示，文献[5,10]采用了类似的算法结构。由于预测最多能产生1位误差，因此可先通过加法器的计算结果计算出实际移位量的最低有效，然后与预测移位值的最低有效位进行比较，来判断是否需要补偿1位。此算法结构与第二种结构很相似，它同样因误差判定信号产生得比较慢而增大了LOP的延时。

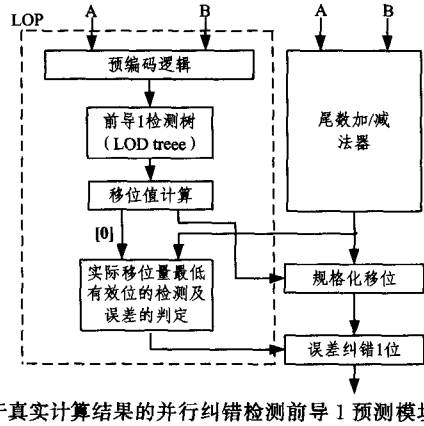

图5 基于真实计算结果的并行纠错检测前导1预测模块的总体结构

第四种结构如图6所示，它在规格化操作之后再判断是否需要纠错1位，这种算法结构虽然将误差纠错的判定放到预测算法的尾部，但这使得尾数加减法器和LOP的延时相当，从整体效果上路径延时比较均衡，而且相比前3种结构，此算法结构的电路面积最小。

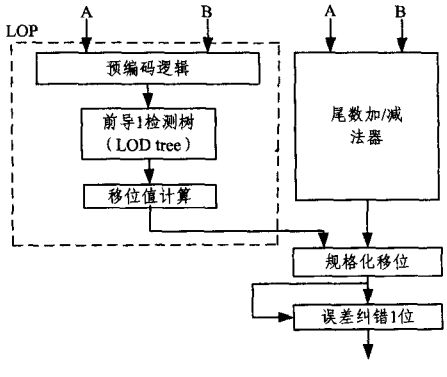

图6 串行纠错检测前导1预测模块的总体结构

本文在第 3 节中重点描述了串行纠错前导 1 预测算法的具体结构，并对其关键的部分在算法上进行了比较和优化，而且在第 4 节中对比了它与其它算法结构的实现效果。

3 串行纠错前导 1 预测算法设计实现

3.1 前导 1 预编码逻辑

由于尾数加法器的输出结果可以是正值也可以是负值，因此在文献[6]介绍的预编码逻辑使用了两类编码串：正编码串和负编码串，然后用前导 1 检测逻辑分别对这两个编码串进行检测，从而产生两个前导 1 位置的编码，最后用实际结果的符号位来选择正确的前导 1 位置的编码。而文献[1]是将上述的两类编码串进行合并，产生一个统一的编码串，然后只用一个前导 1 检测模块对其进行检测并得到前导 1 位置的编码。前者的方法需要加法器计算结果的符号位来产生正确的编码串，因此增加了加法器的延时。为了使 LOP 达到与尾数加减法器完全的并行，本文采用了统一的编码方式。

预编码逻辑的实现可以分为两个步骤：首先对两个输入数据 A、B 进行编码。将 A、B 相应的每一位进行不带借位的减法，其结果会出现 1、0、−1 这 3 种情况中的一种情况，可以令 g_i、e_i、s_i 分别代表 1、0、−1，则：

$$g_i = a_i \cdot \overline{b_i}, \text{如果 } a_i > b_i \tag{1}$$

$$e_i = \overline{a_i \oplus b_i}, \text{如果 } a_i = b_i \tag{2}$$

$$s_i = \overline{a_i} \cdot b_i, \text{如果 } a_i < b_i \tag{3}$$

于是可以得到关于 g_i、e_i、s_i 的编码串，接着，对此编码串中的每一位再进行编码，定义为预编码串 F，F 中的第一个“1”所在的位置便是前导 1 的位置。预编码串 F 的每一位 f_i 定义如下：

$$f_i = e_{i+1} \cdot (g_i \cdot \overline{s_{i-1}} + s_i \cdot \overline{g_{i-1}}) + \overline{e_{i+1}} \cdot (s_i \cdot \overline{s_{i-1}} + g_i \cdot \overline{g_{i-1}}) \tag{4}$$

式(4)中的下标为边界值时，规定如下：若数据位宽为 54 位，则当 $i=53$ 时，$e[54]=1$；当 $i=0$ 时，$s[-1]=g[-1]=0$。

前导 1 预编码逻辑的电路结构如图 7 所示。

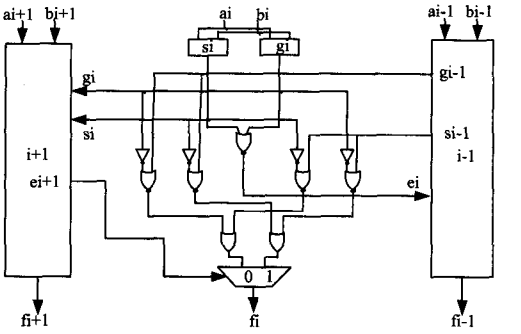

图 7 前导 1 预编码逻辑的电路结构图

3.2 前导 1 检测模块

常见的前导 1 检测算法有两种。第一种算法是文献[1]和文献[9]中介绍的树形结构的前导 1 检测算法，它和文献[3]所提算法相似。这种前导 1 检测模块的基本单元是具有 4 个输入和 2 个输出的 LOD 树，如图 8 所示。

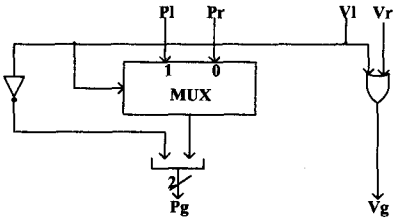

图 8 LOD 树的基本单元的逻辑结构图

输入数据位宽为 64 位的前导 1 检测树的结构如图 9 所示，它有 6 级逻辑，除了第一级外，其它的每级逻辑由一个 OR 门和 2 输入的多路选择器组成。

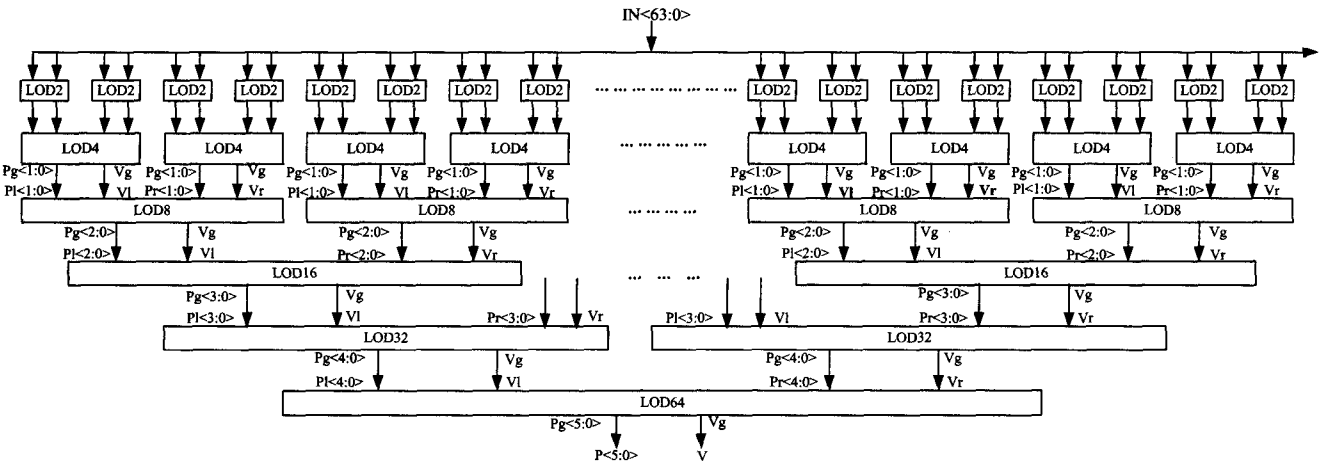

图 9 检测树结构的前导 1 检测逻辑的结构示意图

另一种前导 1 检测算法采用分组的方式对前导 1 进行检测。对于 64 位的数据，按从高到低的顺序，每 8 位一组分成 8 组，每组组内进行全或，然后组间和组内同时进行编码，并由组间控制信号对组内编码进行选择，组间编码对应前导 1 位置编码的高位，而组内编码对应前导 1 位置编码的低位，如图 10 所示。

由于组内编码逻辑不在整个前导 1 检测模块的关键路径上，因此本文摒弃了利用并行的多路选择器进行前导 1 位置低位的编码，而是直接对 8 位数据进行组合逻辑运算来进行编码，以此来减小电路的面积。

分组方式的前导 1 检测逻辑在对前导 1 的位置进行检测编码时，组间编码(高位结果)和组内编码(低位结果)可以并行执行，而且它与下一节中涉及的分级规格化移位(见图 11)能更好地进行时序配合(组间编码先完成，在组间编码规格化移位的同时进行组内编码)，从而提高了前导 1 检测的速度，因而本设计选择了分组的方式对前导 1

进行检测。

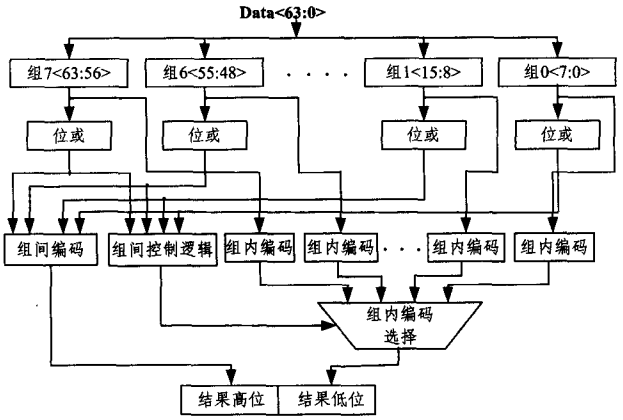

图 10　分组方式的前导 1 检测逻辑的结构示意图

3.3　误差纠错逻辑

预测算法可能会带来误差，但最多误差 1 位。本文采用的误差纠错方法是对经过规格化移位后的结果的最高有效位进行检测，判断是否为“1”，若为“0”，说明前导 1 预测的结果存在 1 位误差，规格化移位的结果还需要再规格化左移 1 位，如图 6 所示。从算法结构上看，这种串行的纠错方法会增加整个规格化移位操作的延时，然而，由于纠错移位最多需要移 1 位，因此其在逻辑上不复杂，时序上也较快，不会对整个流水站周期产生过大的影响。如果该路径出现在关键路径上，还可结合分级的规格化移位，方便地对流水站进行微调来消除关键路径，如图 11 所示。

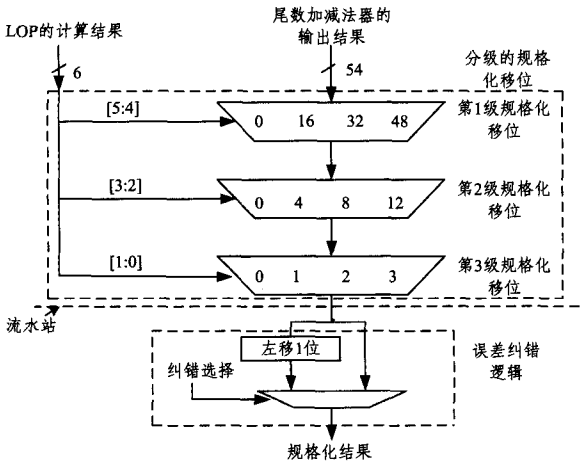

图 11　串行方式的误差纠错逻辑结构示意图

4　性能评估

本文利用逻辑综合工具，使用 40nm 的工艺库，分别对双精度浮点数的并行纠错前导 1 预测模块、基于真实计算结果的并行纠错前导 1 预测模块、串行纠错前导 1 预测模块(包括加法器和移位器，如图 3、图 5、图 6 所示)，以及与加法器串行的前导 1 检测模块(见图 2)进行了逻辑综合，其综合结果如表 1 所列，其中这 4 种结构中的前导 1 检测逻辑均采用了分组方式的前导 1 检测算法，加法器均采用超前进位加法器。4 种算法结构的综合结果显示，LOP 均出现在了关键路径上。

表 1　前导 1 位置判定逻辑的 4 种不同算法结构的综合结果

性能指标	面积/um^2	动态功耗/uW	静态功耗/uW	关键路径延时/ns
传统前导 1 检测模块	4430.93	2023.4	144.70	0.71
并行纠错的前导 1 预测模块	10440.53	3336.2	341.85	0.55
基于真实计算结果的并行纠错前导 1 预测模块	7718.32	2880.9	266.70	0.57
串行纠错的前导 1 预测模块	6640.17	2505.8	233.85	0.57

从表 1 中的综合结果可以看出，虽然传统前导 1 检测算法比预测的算法简单，且其面积和功耗最小，但传统前导 1 检测模块的延时最大，分别比并行纠错前导 1 预测模块增加了 29%，比基于真实计算结果的并行纠错前导 1 预测模块和串行纠错前导 1 预测模块均增加了 26%。从表 1 中还可以看到，与并行纠错前导 1 预测模块相比，虽然串行纠错前导 1 预测模块的延时大了 20ps(增加了 3.6%)，但其面积却减小了 36%，功耗降低了 26%；同时，串行纠错前导 1 预测模块的延时与基于真实计算结果的并行纠错前导 1 预测模块的延时几乎相同，但其面积和功耗都相对地较小(面积小了 14%，功耗降低了 13%)。

根据以上实验结果可知，虽然串行纠错前导 1 预测算法所实现的逻辑电路在速度上并不是最快的，但是在面积、功耗和延时之间却能得到很好的均衡，实现的综合性能最高。采用此算法结构的 LOP 与双通路(Two-Path)[11-13]的双精度浮点加减算法相结合，达到了 1GHz 的时钟频率、三拍完成的设计目标。

结束语　为了能满足 1GHz 频率、三站流水的双通路浮点加法器设计目标的要求，本文介绍了一种用在浮点加减运算中的串行纠错前导 1 预测算法的设计及实现。通过逻辑综合实验的结果表明，采用串行纠错前导 1 预测算法所实现的电路在速度、面积和功耗三者之间能取得很好的均衡，并且达到了设计的 1GHz 的频率目标，它很适合应用在基于 Two-Path 算法的三站式快速浮点加法器的实现上。在使用中，规格化移位的同时要进行指数的调整，之后还要进行异常结果的判断，因此路径将会较长。使用串行纠错前导 1 预测算法的另一个优越性是可以层次清晰地将部分级数低的规格化移位和误差纠错逻辑进行后移，以均衡各级流水站的长度，使整体效果达到最佳。

参 考 文 献

[1] Bruguera J D, Lang T. Leading-One Prediction with Concurrent Position Correction[J]. IEEE Transactions on Computers, 1999, 48(10): 298-305

[2] Ji Rong, Ling Zhi-qiang, Zeng Xian-jun, et al. Comments on Leading-One Prediction with Concurrent Position Correction [J]. IEEE Transaction on Computer, 2009, 58(12)

[3] Suzuki H, Morinaka H, Makino H, et al. Leading Zero Anticipatory Logic for High Speed Floating-Point Addition[J]. IEEE Journal of Solid State Circuits, 1996, 31(8): 1157-1164

[4] Schmookler M S, Nowka K J. Leading zero anticipation and detection: A comparision of methods. Colorado[C]// Proc. of the 15th IEEE Symposium on Computer Arithmetic. July 2001: 7-12

(下转第 50 页)

4.2 FECN 和 BECN 的观测分析

对 FECN 和 BECN 的观测分析主要就是使用 wireshark 对 ibdump 捕捉的数据包进行分析，各个数据流的 FECN 标记情况是均匀分布的(见图 6)。由于 ibdump 程序只捕捉接收的数据包，不捕捉发送的数据包，在 H4 捕捉到的数据包可以观察到 FECN 对照，而 BECN 只能在各个发送端才能观察到。通过分析可知，当拥塞刚刚发生时，占据带宽大的数据流会在很短的时间(330μs 左右)接收到比占据带宽小的数据流相对多的 CNP，更快地降低发送速率，而占据带宽小的数据流收到相对少的 CNP，较慢地降低发送速率，当二者的发送速率接近一致时，FECN 和 BECN 都出现了稳定的分布，从而实现带宽的公平分配。实验中，H1 首先发送数据流，H2 在 1 秒后开始发送数据流，当拥塞发生时，在 330μs 左右的时间内，H1 收到了 25 个 CNP，而 H2 只收到了 10 个 CNP。

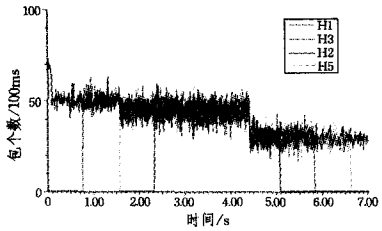

图 6 FECN 观测

4.3 ibdump 程序参数设置对 IB 网络性能的影响

表 3 no ibdump、ibdump -b、ibdump --mem 对照

参数		平均流量(Mb/s)
-b	2	6629.984256
	3	**6346.9363**
	4	6739.66387
	5	6573.76563
	6	6493.04474
	7	6692.145152
	8	**6806.667264**
	9	6721.32301
	10	6702.158848
	11	6686.116864
	12	6580.53632
--mem 5GB		**6083.919457**
no ibdump		**7327.796224**

表 3 是 no ibdump、ibdump -b、ibdump --mem 3 种情况下的流量对照，从中可以发现 ibdump 程序参数对 IB 网络性能的影响都很大。对于-b 参数，参数值为 3 时网络性能最差，平均流量只有 no ibdump 情况下的 86.61%；参数值为 8 时网络性能最好，但也只有 no ibdump 情况下的 92.89%。对于--mem 参数，因为实验中突发流量很大，5 秒左右的发送会产生约 4.4GB 的数据，实验中设定的参数为 5GB，ibdump 直接占用 5GB 的内存用于存储捕捉到的数据包。由于需要频繁地捕捉数据包，并储存到内存中，虽然减少了数据包丢失，但是系统的开销过大，对网络性能造成了很大的影响，平均流量只有 no ibdump 情况下的 83.04%，比 ibdump -b 3 的性能还要差。

结束语 实验分析表明，基于 ibdump 的 IB 网络拥塞控制机制和拥塞行为观测实验方法能很好地验证 IB 网络的拥塞控制机制，有效地解决拥塞问题，实现良好的公平性。同时对实验中 ibdump 各项参数对网络的性能影响也进行了分析，进行了 ibdump 参数设置的优化。

下步将继续研究分析 IB 网络拥塞控制各项参数的影响和拥塞控制机制的 ECN 处理算法。

参 考 文 献

[1] Top 500 supercomputer sites[OL]. http://top500.org/

[2] Santos J R, Turner Y, Janakiraman G J, et al. End-to-end congestion control for InfiniBand[C]// INFOCOM. 2003

[3] Pfister G, Gusat M, Craddock D, et al. Solving Hot Spot Contention Using InfiniBand Architecture Congestion Control[J]. Invited paper in High Performance Interconnects for Distributed Computing, July 2005

[4] Gran E G, Eimot M, Reinemo S-A, et al. First Experiences with Congestion Control in InfiniBand Hardware[C]// IPDPS' 10. 2010

[5] Gran E G, Zahavi E, Reinemo S-A, et al. On the Relation between Congestion Control, Switch Arbitration and Fairness[C]// CCGRID'11. 2011

[6] InfiniBand Architecture Specification[S]. Release 1.2.1, InfiniBand Association, 2007. http://www.InfiniBand.org, 2007

[7] Mellanox OFED for Linux User Manual Rev 1.5.3[M]. Mellanox Technologies. Ltd, 2011. http://www.mellanox.com,

[8] 吕高峰，苏金树，孙志刚，等. IBS216 交换机设计与实现[J]. 计算机研究与发展，2011，48：1-9

(上接第 34 页)

[5] Dimitrakopoulos G, Galanopoulos K, Mavrokefalidis C, et al. Low-Power Leading-Zero Counting and Anticipation Logic for High-Speed Floating Point Units[J]. IEEE Transactions on Very Large Scale Integration System, 2008, 16(7)

[6] Lee K T, Nowkda K J. 1GHz leading-zero anticipator using independent sign-bit determination logic[C]// Symposium on VLSI Circuit Digest of Technical Papers. 2000

[7] Zhang Ge, Hu Wei-wu, Qi ZI-chu. Parallel Error Detection for Leading Zero Anticipation[J]. J. Comput. Sci. &Technol, 2006, 21(6): 901-906

[8] Yao Tao, Gao De-yuan. A Novel Concurrent Error Detection Circuit for Leading Zero Anticipator[C]// 2nd International Conference on Computer Engineering and Technology. 2010

[9] Oklobdzija V. An Algorithmic and Novel Design of a Leading Zero Detector Circuit: Comparison with Logic Synthesis[J]. IEEE Transactions on VLSI System, 1993, 2(1): 124-128

[10] Hinds C N, Lutz D R. A Small and Fast Leading One Predictor Corrector Circuit[C]// Asilomar Conference on Signals, Systems and Computers. 2005: 1181-1185

[11] Oberman S F, Flynn M J. A variable Latency Pipelined Floating-point Adder[R]. CSL-TR-96-689. Stanford University, 1996

[12] 黄迟. 64 位高速浮点加法器的 VLSI 实现和结构研究[D]. 上海：复旦大学，2004

[13] 王颖. 高性能 CPU 中浮点加法器的设计与实现[D]. 上海：同济大学，2005