

文章编号: 1001- 2486(2009) 06- 0006- 06

YHFT-DX 高性能 DSP 中 Cache 失效流水设计*

郭 阳, 傅 晖, 刘 胜, 李 勇

(国防科技大学 计算机学院, 湖南 长沙 410073)

摘要: YHFT-DX 是国防科技大学自主研制的一款高性能 DSP。以提升 YHFT-DX 的 Cache 性能为目标, 研究了降低 Cache 失效延迟的优化策略, 设计并实现了一种针对高频高性能 DSP 的一级数据 Cache 优化策略——失效流水。与传统优化策略相比, 该策略将连续访问 Cache 的失效请求并进行流水化处理, 使多个 Cache 失效延迟重叠, 从而达到降低平均 Cache 失效代价的目的。将该策略应用到 YHFT-DX 芯片的一级数据 Cache 控制器的设计与优化中, 使访问 Cache 失效引起的流水线停顿从 8 拍降为 2 拍, 显著提升了系统性能。

关键词: DSP; 失效流水; 非阻塞 Cache; 数据预取

中图分类号: TP368.1 **文献标识码:** A

Design of Cache Miss Pipelining in YHFT-DX High Performance DSP

GUO Yang, FU Yi-hui, LIU Sheng, LI Yong

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: YHFT-DX is a high performance DSP designed by national university of defense technology. This paper focuses on improving Cache performance, investigates optimization methods to reduce Cache miss stall penalties, designs and implements an optimization method focusing on one level data Cache controller in high frequency and high performance DSP-miss pipelining. Compared with traditional optimization methods, this method can deal with continual cache misses in pipeline, which overlaps multi Cache miss stalls, and then it can achieve the goal of reducing Cache miss stall penalties. Applying the method to the design and optimization in one level data Cache controller in YHFT-DX DSP, the Cache miss stall is reduced from 8 cycles to 2 cycles, and the system performance is evidently improved.

Key words: DSP(Digital Signal Processor); miss pipelining; unblocking Cache; data prefetching

数字信号处理器(Digital Signal Processor, DSP)广泛应用于计算机、网络、移动电话、3G 通信等领域。DSP 的性能以每年 60% 的速度在增长, 而 SRAM 访问时间每年的改善还不到 10%, 这就是常说的“存储墙”问题^[1-2]。

为了满足 DSP 实时性要求,“两级 Cache+ RAM”的存储结构成为高性能 DSP 片内存储结构的重要方向, 如 TI 公司的高性能 DSP TMS320C621x、C671x 和 C64x 均采用两级 Cache 结构, 其二级 Cache 也可配置为 RAM 使用^[3]。面向该结构的 Cache 优化研究多集中在减小失效率、降低命中时间、减小失效延迟三个方面。增大 Cache 容量是常见的一种 Cache 优化方法, 可以有效降低 Cache 失效率, 但也会增加 Cache 的失效代价^[4]。

2004 年, 我们研制成功“银河飞腾”浮点 DSP——YHFT-DSP/700^[5], 它采用八流出超长指令字(VLIW)体系结构, 拥有直接外存通路和高效外部接口, 其一级指令 Cache(以下简称 L1P)和一级数据 Cache(以下简称 L1D)容量都为 4KB, 二级 Cache(以下简称 L2)大小为 64KB。YHFT-DX 是我们自主研制的新一代高性能定点 DSP, 同样采用八流出 VLIW 结构, 其 L1P 和 L1D 容量都增大为 16KB, L2 增大为 1MB, Cache 总容量比 YHFT-DSP/700 增加了近 16 倍, 使得 YHFT-DX 的 Cache 访问失效率比 YHFT-

* 收稿日期: 2009- 07- 03

基金项目: 国家自然科学基金资助项目(60573173); 新世纪优秀人才计划项目(NCET); 教育部“高性能微处理器技术”创新团队资助项目(IRT0614)

作者简介: 郭阳(1971-), 男, 副研究员, 博士。

DSP/700 小很多, 但 L1D 的访存失效在 L2 命中所需阻塞流水线的时间也比 YHFT-DSP/700 多了 4 拍。针对这种情况, 本文提出并设计实现了一种降低 L1D 平均失效延迟的 Cache 优化策略——失效流水, 使访问 Cache 失效引起的流水线停顿从 8 拍降低为 2 拍。

1 Cache 优化相关策略分析

目前, 降低失效延迟的传统优化策略有非阻塞 Cache 和数据预取^[6]两种。文献[7]提出了一种在关键字优先基础上一次缺失下保持继续命中功能的非阻塞 Cache 的策略。其主要思想是: 当出现一次 Cache 失效时, 还可以继续让后续的命中访存指令进行访问 Cache 的操作, 如果出现两次访问 Cache 失效时, 则阻塞流水线。根据程序局部性原理, 随后的访存操作继续命中的概率很小, 并且很可能是访问上一个或者下一个 Cache 行的地址。而该策略采用了请求字优先的技术后, 可以较好地处理随后的访存失效在同一个 Cache 行的情况, 但是对地址步长增减的连续 Cache 失效却没有给出好的解决方案。

文献[8]提出了一种结合访存失效队列状态的预取策略, 通过流过滤表分析访存失效的状态, 并选择合理时机发出预取请求, 预取地址通过分析访存失效队列并进行统计, 更好地利用了程序局部性原理, 解决地址步长增减的连续 Cache 失效问题; 同时, 比文献[9]中 Jouppi 最早提出的 StreamBuffer 预取技术的效率要高, 而且预取与正常的访存操作由于总线占用而冲突的情形也较少。但文献[8]提出的预取策略最终无法摆脱预取到的数据有可能不被使用, 或后续失效需要的数据由于预取请求不能发出而使预取 Buffer 无法发挥作用的局面。

YHFT-DX 是一款对功耗和实时性要求很高的数字信号处理器, 其低功耗的特点要求当出现访问 Cache 失效时, 预取来的数据不能是无效的。其实时性的特点要求程序通过编译器事先编译好再执行, 不支持指令乱序流出, 因此基于流预取 Buffer 和非阻塞 Cache 的传统优化策略对于 YHFT-DX 的一级数据 Cache 控制器的优化来说都不可行。

2 YHFT-DX 的失效流水机制

YHFT-DX 的 L1D 支持两条指令同时访存, 在 1 拍内最多可以产生 2 个读失效。如果按照串行原则, 即先向 L2 发一个读失效请求, 等待 L2 响应, 得到 L2 分配的新数据行到来后, 再向 L2 发送另外一个读失效请求。为了不破坏指令调度的先后顺序, 先完成的访存指令要等待后发出的访存指令的完成, 然后两条指令并行进入下一站, 此后才允许后继访存失效指令向 L2 发数据请求。如果两个读失效均命中 L2 的 Cache, 则这两条并发的读请求将导致 CPU 停顿 16 个周期, 即每个失效都阻塞 8 个时钟周期。

实际上, L1D 在向 L2 发出数据请求并等待响应的时间内, 没有做任何事情。因此可以把时间间隙利用起来, 即在等待 L2 返回数据的时间里, 不等数据到达而继续对后续访存指令进行失效或命中判断, 如果有新的失效, 立即向 L2 发请求, 如此流水处理失效, 把单条指令的访存失效延迟重叠起来。如图 1 所示, 描述了 L1 与 L2 失效流水线整体结构。失效流水线可划分为 6 站:

第一站: L1D 发出失效请求地址、宽度和数据, L2 接收并做分类处理;

第二站: L2 读 Tag;

第三站: L2 比较 Tag, 判断 L1D 的失效是否在 L2 的 Cache 中命中;

第四站: 如果命中 L2 的 Cache, 则向 L2 的 Cache 数据体读数据;

第五站: 命中数据从 L2 的 Cache 体读出, L2 将 L1D 所需要的数据返回;

第六站: L1D 接收数据, 完成一个失效请求的处理。

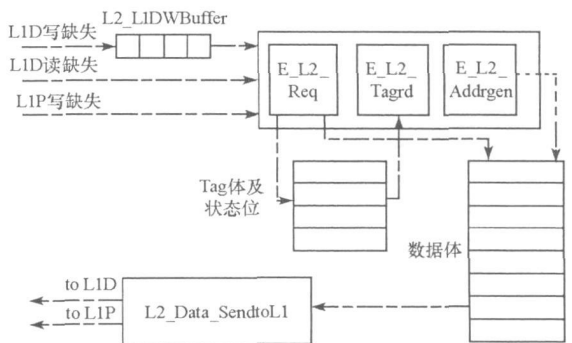


图 1 失效流水示意图

Fig. 1 Miss pipelining meanings

失效流水线始于 L1D, 终于 L1D, 而中间却要跨越整个 L2。只有从整体上综合考虑这两个部分才能保证两条流水线的正常工作。

3 L1D 失效流水的设计

3.1 失效请求处理

设计 L1D 失效流水线, 关键在于能连续发出失效请求, 即在等待 L2 响应数据到达的时间空隙中, 需要继续处理后续存取指令, 如果有失效指令就继续向 L2 发出请求。

YHFT-DX 所有指令间的相关性问题在派发前都已由编译器解决完成。一旦 L1D 判断出访存失效, 下一拍流水线马上阻塞, 从根本上限定了不会有太多新的存取指令进入数据 Cache 中, 即没有复杂的数据相关性和顺序性问题。图 2 是本文设计的 L1D 流水线结构, 可以分为 5 站:

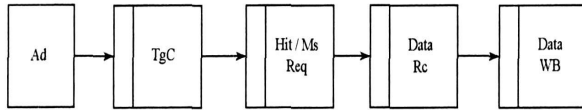


图 2 L1D 的流水线结构

Fig. 2 Pipeline structure of L1D

第一站: CPU 内核产生访问内存的地址;

第二站: L1D 从 Tag 体中读出数据, 并进行命中与失效判断, 将判断结果送入站间寄存器;

第三站: 如果命中就向 L1D 的 Cache RAM 读数据, 如果失效就向 L2 发出失效请求, 并停顿流水线;

第四站: 接收从 Cache 返回的数据;

第五站: 将数据写回 CPU。

从流水线结构可以看出, 当第二站判断出失效后, 还可以再流水一拍, 到第三站再发出失效请求, 而此时第二站可以判断下一拍指令的失效命中。因为 L1D 支持两个通路的并行访问, 所以 L1D 流水线在第三站处理两个并行的失效指令时, 还可以在第二站判断出两条并行指令失效或命中的情况, 因此 L1D 失效流水最多可能判断并处理四个失效请求。

3.2 L1D 失效流水的实现

图 3 是 L1D 失效流水具体实现的结构, 其包含的模块和功能如下:

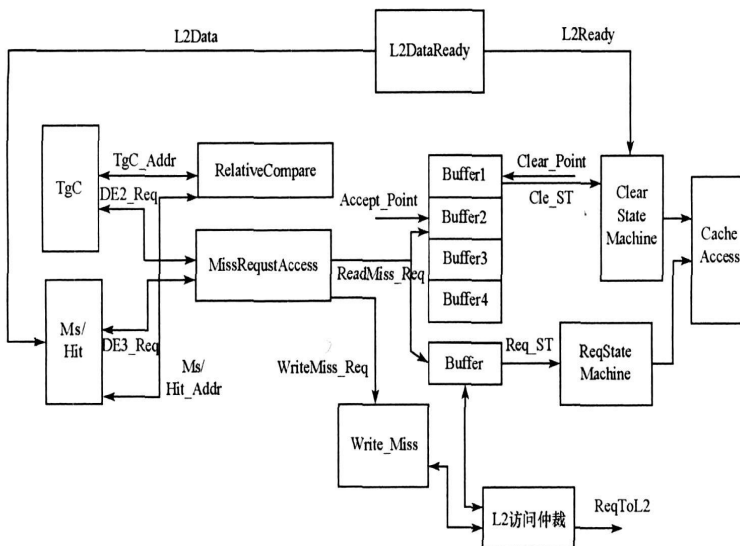


图 3 L1D 失效流水的实现结构

Fig. 3 Implementation structure of L1D miss pipelining

- TgC 模块: L1D 流水线的第二站, 发送第二拍失效请求。
- Ms/Hit 模块: L1D 流水线的第三站, 发送第一拍失效请求。
- RelativeCompare 模块: 比较 TgC 站和 Ms/Hit 站的地址, 排除前后两拍指令的相关性。
- MissRequestAccess 模块: 接收流水线失效请求, 分类后将请求送往读和写失效处理模块。
- WriteMiss 模块: 向 L2 发送写失效请求。
- Buffer、Buffer1~ Buffer4、ClearStateMachine、ReqStateMachine、CacheAccess 五部分构成读失效处理模块: Buffer 和 ReqStateMachine 向 L2 发送读失效请求, Buffer1~ 4 和 ClearStateMachine 负责 L2 返回的数据接收, CacheAccess 负责 L2 返回的数据向 L1D 数据体写回。
- L2 访问仲裁模块: 选择有效的 L1D 的请求向 L2 发出。
- L2DataReady: L2 返回数据时, 对返回的数据进行缓冲。

图 4 是 L1D 处理失效指令状态转换图, L1D 失效指令的处理通过下面 7 个步骤来完成:

第一步: 在 TgC 站比较 Tag 后判断出指令命中或失效, 将命中或失效信息送入下一站。

第二步: Ms/Hit 站将产生的失效指令进行顺序化处理, 选择合适的地址、宽度和数据后, 再向 MissRequestAccess 模块发出失效请求。然后由 MissRequestAccess 模块将流水栈发送来的失效请求分类, 并将写失效请求送往 Write_Miss 模块, 将读失效请求送往读失效请求处理 Buffer。如果是读失效请求则进行第四步; 如果是写失效请求则进行第五步。

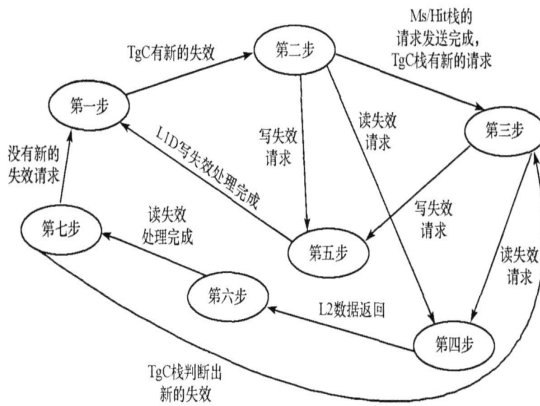


图 4 L1D 处理失效指令状态转换图

Fig. 4 State transition diagram for L1D handle miss instruction

第三步: 如果 Ms/Hit 站的失效请求发送完毕, 就将失效请求发送权限交给 TgC 站, 如果在 TgC 站中有新的失效请求, 则先通过 RelativeCompare 模块排除前后访存指令间的相关性, 然后进行顺序化处理, 选择合适的地址、宽度和数据后, 再向 MissRequestAccess 模块发出请求。如果是读失效则进行第四步; 如果是写失效则进行第五步。

第四步: 读失效处理 Buffer 向 L2 发出读失效请求, 并将发送成功的失效请求地址送入空闲的循环队列 Buffer 中保存, 等待 L2 处理完失效请求并返回数据时进行第六步。

第五步: 写失效处理模块向 L2 发送写失效处理请求, 并向流水站发出写失效处理完毕信号。因为 YHFT-DX 的 Cache 采用非写分配策略, 所以只要写失效请求被 L2 成功接收后, 就算处理完成, 后续写失效处理交由 L2 来完成。

第六步: L2 的数据返回时, 通过 ClearStateMachine 失效请求清空状态机对进入循环队列的读失效请求, 按照请求进队列的先后顺序进行数据接收处理。如果 L2 返回的数据可以进 Cache, 则还要通过 CacheAccess 模块将数据写回 L1D 的 Cache RAM 中, 同时通知流水站对失效指令所需数据进行接收。当一个失效请求的数据接收完成, 就将该失效请求的有效信号置为无效, 同时给流水站发送失效请求处理完毕的信号, 进行第七步。

第七步: 当 Ms/Hit 站发现站内所有的失效指令的请求都已经处理完成, 启动流水线。让 TgC 站的

失效请求进入 Ms/Hit 站。同时 TgC 站开始新一拍的访存指令的失效或命中判断。如果没有失效指令,那么流水线继续流动;否则,再转入第三步,进行失效请求处理。

4 失效流水优化策略的评测

与传统 Cache 优化策略相比,本文设计实现的 L1D 失效流水有以下优点:

(1) 硬件实现简单。失效流水的相关性判断模块只需要将第二站的指令地址与第三站的指令地址进行比较,就能判断出第二站的指令失效是否与正在处理的失效指令相关。

(2) 效率高。L1D 失效流水发出的失效请求从 L2 读出来的 Cache 数据块一定是可用的,在一定程度上减少了 L1D 对 L2 访问带宽的占用,提高了访存效率。

(3) 应用面广。“失效流水”策略兼顾了嵌入式高性能 DSP 实时性和低功耗的特点,对于其他高频高性能嵌入式微处理器的设计与优化,都有很好的借鉴作用。

(4) 优化效果明显。表 1 是没有使用失效流水进行优化的 L1D 处理失效指令的情况,2 个连续的读失效需要阻塞 16 拍流水线。表 2 是使用了失效流水进行优化后的 L1D 的情况,4 个连续的读失效只需要阻塞 14 拍流水线,平均每个读失效阻塞 3.5 拍流水线。如果连续多拍出现读失效指令,理想情况下可以做到每个读失效只阻塞 2 拍流水线。

表 1 没有实现缺失流水的 L1D

Tab. 1 L1D not realizing miss pipelining

周期	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
R1	Ms	Rq							Da	Db								
R2	Ms									Rq						Da	Db	
CPUStall	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

表 2 实现缺失流水的 L1D

Tab. 2 L1D realizing miss pipelining

周期	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
R1	Ms	Rq							Da	Db							
R2	Ms		Rq							Da	Db						
R3	Tg	Ms		Rq								Da	Db				
R4	Tg	Ms			Rq										Da	Db	
CPUStall	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0

本文选取 6 个常用的 Benchmark 来测试优化效果,如表 3 所示。其中,FFT 是对 64×64 矩阵进行二维傅立叶变换;Float_EXP_LOG 是计算 700 个浮点数的指数结果和 700 个数的对数结果的程序;Ti_3in1 是包含多组 FIR、FFT、向量运算、图像压缩等典型 Benchmark 的程序库;MPEG4 Decoder 是参数为“尺寸 20×10 , 帧数 1, 帧率 10 帧/s”的视频解码程序;MPEG4 Encoder1 是参数为:“尺寸 20×10 , 帧数 1, 帧率 10 帧/s”的视频编码程序;MPEG4 Encoder2 是针对 YHFF-DX 硬件优化的视频编码程序,参数与 MPEG4 Encoder1 相同。

表 3 L1D 缺失流水模拟结果

Tab. 3 The simulation result of L1D miss pipelining

BenchMark	优化前缺失 周期数	优化后缺失 周期数	隐藏延迟 拍数	提升 Cache 性能	提升系统性能
FFT	1 140 728	864 509	276 219	24%	0.8%
Float_EXP_LOG	251 280	86 526	164 754	66%	3.3%
Ti_3in1	7672	5416	2256	29%	0.2%
MPEG4 Decoder	9 094 984	6 746 320	2 348 664	26%	2%
MPEG4 Encoder1	3 883 288	1 866 708	2 016 580	52%	3.9%
MPEG4 Encoder2	3 865 736	1 856 787	2 008 949	51%	10%

表 3 中的 Cache 性能评价公式是:

$$\text{Cache 效率提升} = \text{隐藏延迟拍数} \div \text{优化前失效周期数} \times 100\%$$

隐藏延迟拍数就是系统运行 Benchmark 所减少命中时间 L2 的总周期数。隐藏延迟拍数越多, 优化前缺失周期数越少, 那么优化效果就越明显。由表 3 可以看出, L1D 实现失效流水优化后, 系统运行不同 Benchmark 的 Cache 性能都得到提升, 最高达到 66%, 最低是 24%。通过分析硬件的结构与模拟程序的关系, 发现实质上失效流水优化一级数据 Cache 取得的性能提升与程序可优化程度、程序数据相关程度两个因素有关: 程序连续失效率越大, 优化的效果就会越明显; 程序中失效指令的相关性越小, 硬件优化性能提升的效果越明显。

5 结论

在 YHFT-DX 芯片的 L1DCache 控制器的设计与优化中, 本文采用了一种降低失效延迟的 Cache 优化策略——失效流水, 使平均访存失效代价降低了 25%, 使整个系统平均能提升 1%。下一步, 我们将该方法用于芯片的 L1PCache 控制器的优化中, 进一步提高整个失效流水线的效率。

参考文献:

- [1] Henessy J L, Patterson D A. Computer Architecture: A Quantitative Approach[M]. Third Edition. 北京: 电子工业出版社, 2004: 257- 258.
- [2] Sule A M. Design of Pipeline Fast Fourier Transform Processors Using 3 Dimensional Integrated Circuit Technology[D]. PHD Thesis, NCSU, 2007.
- [3] 田黎育, 何佩琨, 朱梦宇. TMS320C6000 系列 DSP 编程工具与指南[M]. 北京: 清华大学出版社, 2006.
- [4] Henessy J L, Patterson D A. Computer Architecture: A Quantitative Approach[M]. Fourth Edition. 北京: 电子工业出版社, 2007: 412- 413.
- [5] 陈书明, 李振涛, 万江华, 等. “银河飞腾”高性能数字信号处理器研究进展[J]. 计算机研究与发展, 2006, 43(6): 993- 1000.
- [6] Guo Y, Chheda S, Koren I, et al. Energy-aware Data Prefetching for General-purpose Programs[C]//Proceedings of Power-aware Computer Systems, IEEE CS Press, Portland, USA, 2004: 78- 94.
- [7] 黄海林, 等. 嵌入式处理器中降低 Cache 缺失代价设计方法研究[J]. 小型微型计算机系统, 2006, 27(11): 2077- 2081.
- [8] 郇丹丹, 等. 结合访存失效队列状态的预期策略[J]. 计算机学报, 2007, 30(7): 1104- 1114.
- [9] Jouppi N P. Improving Direct-mapped Cache Performance by the Addition of a Small Fully-associative Cache and Prefetch Buffer[C]//Proc. of 17th Annual Intl Symposium on Computer Architecture, 1990: 364- 373.

(上接第 5 页)

参考文献:

- [1] Nicolaidis M. Design for Soft Error Mitigation[J]. IEEE Transactions on Device and Materials Reliability, 2005, 5(3): 405- 418.
- [2] Mavis D G, Eaton P H. Soft Error Rate Mitigation Techniques for Modern Microcircuits[C]//IEEE 40th Annual Int. Reliability Physics Symp. Dallas, USA, 2002: 216- 225.
- [3] Boulghassoul Y, Massengill L W, Sternberg A L, et al. Towards SET Mitigation in RF Digital PLLs: From Error Characterization to Radiation Hardening Considerations[J]. IEEE Trans. Nucl. Sci., 2006, 53(4): 2047- 2053.
- [4] Loveless T D, Massengill L W, Bhuvu B L, et al. A Hardened-by-design Technique for RF Digital Phase-locked Loops[J]. IEEE Transactions on Nuclear Science, 2006, 53(6): 3432- 3438.
- [5] Koga R, Pinkerton S D, Moss S C, et al. Observation of Single Event Upsets in Analog Microcircuits[J]. IEEE Trans. on Nucl. Sci., 1993, 40(6): 1838- 1844.
- [6] Savage M W, Turflinger T L, Titus J L, et al. Characterization of Set Response of the LM124A, the LM111, and LM6144[C]//IEEE NSREC 2003 Radiation Effects Data Workshop Record, 2003: 121- 126.
- [7] Pease R L, Sternberg A L, Boulghassoul Y, et al. Comparison of SETs in Bipolar Linear Circuits Generated with an Ion Microbeam, Laser Light, and Circuit Simulation[J]. IEEE Trans. on Nucl. Sci., 2002, 49(6): 3163- 3170.
- [8] Loveless T D, Massengill L W, Holman W T, et al. Modeling and Mitigating Single event Transients in Voltage-controlled Oscillators[J]. IEEE Transactions on Nuclear Science, 2007, 54(6): 2561- 2567.
- [9] Tang Y H, Geiger R L. A Non-sequential Phase Detector for PLL-based High-speed Data/Clock Recovery[Z]. Electronics Letters 7th, 2002, 38(23).
- [10] Calin T, Velazco R, Nicolaidis M, et al. Topology-related Upset Mechanisms in Design Hardened Storage Cell. Radiation and Its Effects on Components and Systems[C]//RADECS 97, Fourth European Conference, 1997: 484- 488.