论文阅读报告

《IBM POWER7 MULTICORE PROCESSOR》

肖子原 xiaoziyuan@bit.edu.cn

目录

1. 总述

IBM POWER 处理器是当今世界上主流的精简指令集微处理器之一，在过去的 20 年中有着丰富的创造。在《IBM POWER7 multicore server processor》一文中，主要描述了 POWER7 处理器芯片的主要特性。

在过去几代处理器中，IBM 引入了精简指令集计算架构，高级分支预测，乱序执行，数据预取，多线程，SMP 等技术，在 POWER7 中 IBM 提出了八核芯片设计，eDRAM cache，等技术。 POWER7 的目标是在 POWER6 的基础上显着提高插槽级，核心级和线程级性能，为此，它实现的主要改进有：

- 降低单个核心能耗：降低核心频率以降低能耗，性能的改进通过体系结构来进行。
- 改进的总线技术，实现更高的吞吐。
- 将 L3 cache 挪至片内：由于使用了新的 eDRAM 技术，32MB 的 L3 cache 可以放在片上，这改善了 L3 延迟，同时节约了芯片的通信资源，在 cache 一节详细介绍。

POWER7 可以灵活的分配计算资源，每个核心拥有自己的锁相环，可以运行在各自的频率，同时可以选择性的关闭一些核心，将它的 L3 cache 等资源留给活动的核心使用。在核心内部，POWER7 有 ST，SMT2，SMT4 三种运行模式，一个核心内部资源可以全部分配给活动线程。

2. POWER7 core

POWER7 的一个核心由 6 个单元组成：指令获取单元（IFU），指令排序单元（ISU），LSU，FXU，VSU 和十进制 FPU（DFU），以及对应的 L2 cache。 在一个周期中，内核最多可以获取 8 条指令，最多可以解码和发射 6 条指令，并执行最多 8 条指令，核心内有 12 个执行单元，为了满足 HPC 应用的需求，与前几代产品相比，POWER7 内核具有两倍的 LS 带宽。

POWER7 具有先进的分支预测和预取功能，以及深度乱序执行功能，可显着提高 ST 模式下的计算性能。同时，它有足够的资源来有效地支持每个核心四个线程。

POWER7 核心采取一系列技术来降低功耗，除了前面提到过的降低核心频率，主要包括如下几项。

- 在 SMT4 模式采用了分区的设计方法，不是让每个线程拥有一套 GPR、FXU 和 LSU，而是让一对线程共享套 GPR、FXU 和 LSU，这样可以提供资源的利用。
- GPR,浮点寄存器（FPR）和向量寄存器（VR）都被合并为一个统一的重命名结构，可以降低芯片的面积和功率。
- 浮点单元（FPU）和矢量媒体扩展（VMX）单元在 POWER6 中是分开的,在 POWER7 中，

这两个单元被合并成一个称为 VSU 的单元。

- 单个 L1 cache bank 不能并行，但是整个 L1 cache 可以并行，这种设计减少了 cache 的面积和功耗。

3. 取指与分支预测

POWER7 中的取指工作由 IFU 负责，在 IFU 中实现高度精确的分支预测来提供最有可能的指令流，同时 IFU 还负责维持各个活动线程间指令执行的平衡。

3.1 取指

POWER7 的 I-cache 大小为 32KB，采用 4 路关联。配合其工作的部件有：指令有效地址目录（IEADIR）I-cache 目录（IDIR）指令有效到实际地址转换表（IERAT）等。IEADIR 可以无需进行地址变换，直接预测指令所在的路，先用 IERAT 完成第一级地址转换，再到 IDIR 中去检索，就可以判断所取的指令是否在 I-cache 中。

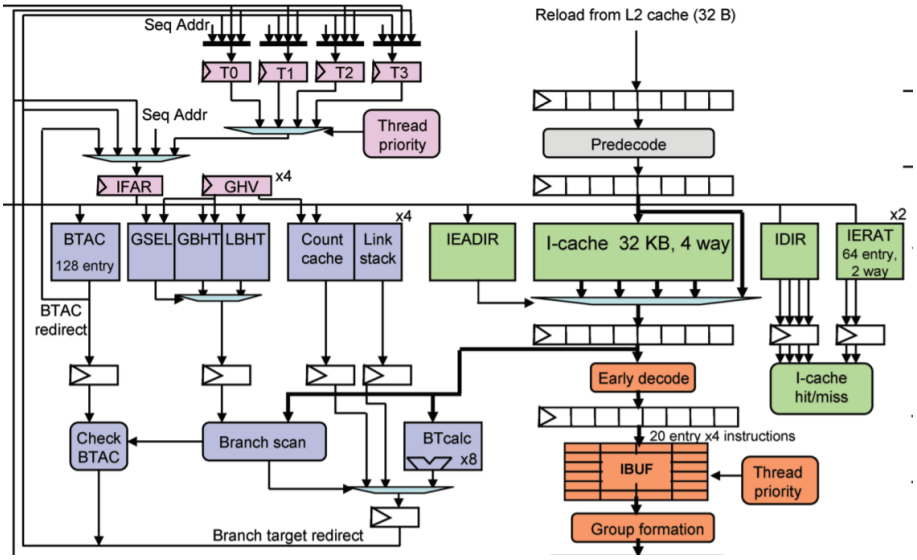

图 3-1 POWE7 取指过程

获取地址寄存器（IFAR）用于指出需要取的指令地址，它的值可以来自 4 个线程中某一个线程的程序计数器，也可以来自分支扫描部件所确定的分支预测地址结果。每当 I-cache miss 发生时，IFU 从 L2 cache 中调入指令，每当发生这样的调入时，预取引擎可以启动最多两个额外的预取。在 ST 模式下允许预取最多，在 SMT2 模式下每个线程允许预取一个，SMT4 模式不允许预取。

3.2 分支预测

分支预测中的预测有两层含义 1)是预测跳转是否发生，2)是预测跳转的地址。对于“是否发生”，POWER7 采用分支预测表预测跳转是否发生，采用 BTAC，link stack 等部件预测跳转的目的地。

分支预测表包括局部表(LBHT)和全局表(GBHT)两部分，LBHT 的大小为 8K，GBHT 的大小为 16K，用 GSEL 表来在 LBHT 和 GBHT 之间进行选择。LBHT 用于记录一个位置上的指令最近两次的跳转状态，其使用地址的低 10 位作为索引，GBHT 根据最近几次的跳转结果预测跳转状态，在 GHV 寄存器中记录了最近 21 次的跳转结果，每次用 1bit 记录，将它压缩到 11 位再连接上指令地址，作为哈希值，hash 到 GBHT 中。对于一条确定的跳转指令，GSEL 表中的条目将决定使用哪一个表来进行预测。

POWER7 使用以下两种机制预测跳转目标地址：

1）使用 link stack 预测过程

调用返回，每个线程一个。每当扫描搭到 branch and link 指令时，下一条指令的地址在该线程的 link stack 中被压栈。每当扫描到 branch to link 指令时，link stack 就“弹栈”。 在 ST 和 SMT2 模式下，每个线程使用 16 个条目的 link stack。在 SMT4 模式下，每个线程使用八项 link stack。

2）使用 count cache 预测其他分支指令，其由所有活动线程共享。将指令地址和 GHV 连接得到值进行 7 bit XOR 操作，XOR 获得的值用来来索引 count cache。计数高速缓存中的每个条目包含 62 位预测地址和两个置信位。

POWER7 对分支预测采取了一些优化措施，主要包括以下 2 点。1)在 ST 模式中，使用 BTAC 来预测两个周期后才能得到的跳转目标地址。如果预测正确，将无缝地获取下一块指令，避免了两个周期的等待。2)如果分支指令的作用仅是为了有条件地跳过后面的 FX 或 LS 指令，POWER7 通常可以检测到这样的分支指令，将其从指令流水线中移除，并有条件地执行 FX 或 LS 指令。

4. 指令排序

POWER7 使用 IFU 来取指，IFU 取指完成后，将指令以组为单位发送给 ISU，ISU 在完成寄存器重命名后负责发射指令，并监视指令的完成情况。一个组最多 6 条指令，其中最多 2 条分支指令，ISU 一个周期内最多可以发射 8 条指令。

在指令进入发射队列前，使用 mapper logic 完成寄存器的重命名，在 POWER7 中可以重命名以下寄存器：GPR，VSR，CR，XER 等，其中 GPR 和 VSR 被映射到 80 个物理寄存器，CR

映射到 56 个物理寄存器，XER 映射到 24 个物理寄存器。

POWER7 使用三个独立的发射队列：一个 48 项的 UQ，一个 12 项分支指令队列（BRQ）和一个 8 项 CR 队列（CRQ）。由于 UQ 的条目数较多，没有采用 shifting queue，而是采用指针来维护队首，这样显著降低了其能耗。UQ 队列分为两半，每半有 24 个条目。它包含 FXU，LSU，VSU 或 DFU 执行的所有指令。队列的上半部分包含 FX0，LS0 和 VS0 流水线的指令，下半部分包含 FX1，LS1 和 VS1 流水线的指令。

ISU 还要负责跟踪指令的资源依赖。对于 BRQ 和 CRQ，通过将重命名的资源的目标物理指针与所有未完成的资源指针进行比较来检查指令依赖性。对于 UQ，通过依赖矩阵来跟踪依赖性。 POWER7 采用全局完成表（GCT）来跟踪所有需要发射的指令。它以指令组的形式进行跟踪，因此将指令作为一个组进行调度和完成。GCT 有 20 个条目，由所有活动线程动态共享。每个 GCT 条目对应于一组指令，每条指令对应一个“完成位”当组中的所有指令都标记为“已完成”，则整个组可以被认为是完成。

5. 数据存取

5.1 LS 执行和排序

在 ST 和 SMT2 模式下，给定的 LS 指令可以在任一流水线中执行。在 SMT4 模式中，来自线程 0 和 1 的指令在流水线 0 中执行，而来自线程 2 和 3 的指令在流水线 1 中执行。进出 LSU 的主数据流包括来自 L2 cache 的 32 字节 load 数据和 16 字节 store 数据，来自 VSU 的 16 字节 load/store 数据，来自 FXU 的 8 字节 load/store 数据。

LSU 必须确保加载和存储指令的架构程序执行顺序的效果，为此，LSU 使用两个队列：SRQ 和 LRQ。SRQ 是一个 32 项的 CAM 结构，每个线程有 64 个可用的虚拟条目，允许分派 64 个未完成的存储指令。对于每个 SRQ 条目，与之对应的是一个 16 字节的 SDQ 条目，它被用来保存需要存储的数据。LRQ 与 SRQ 有着相似的结构。

5.2 地址转换

在 POWER7 中，编写程序使用的是 64 位有效地址（EA），而在 cache 和主存中使用的则是 46 位的真实地址，这个转换过程需要操作系统的配合。在程序执行期间，有多级转换可以将 EA 转换为 RA，在 L1 cache 处完成第一级快速转换，第一级转换所使用的器件包括 64 项的 D-ERAT 和 64 项的 IERAT。在 ERAT 未命中的情况下，调用第二级转换，第二级转换使用的器件为 32 条目的 SLB 和 512 条目的 TLB。

首先使用段表将有效地址转换为 68 位虚拟地址，然后用页表虚拟地址转换为 46 位真实地址。所有缓存器件（ERAT，TLB，SLB）中的数据都是完整页表或者段表中的副本。D-ERAT 是一个 64 项 CAM 缓存。在物理上，有两个相同的 D-ERAT 副本，在 SMT4 模式下，两个副本具有不同的内容。

SLB 是一个 32 条目的 CAM 缓存，由操作系统管理。每个 SLB 条目可以支持 256 MB 或 1 TB 的段大小。POWER7 支持 multiple page size support（MPSS）扩展。使用 MPSS，基页大小为 4 KB 的段可以在段中同时存在 4 KB，64 KB 和 16 MB 页。TLB 是一个 512 条目的 CA 缓存，TLB 由硬件管理并采用 LRU 替换策略。与 POWER6 不同，一个分区被换入时，不需要显式使 TLB 条目无效。如果分区先前在同一核心上运行，则其某些 TLB 条目仍有可能存在，这可以减少 TLB 未命中并提高性能。

5.3 D-cache 组织

POWER7 包含一个专用的 32KB 八路关联 L1 D-cache,cache 行大小为 128 字节，由四个扇区组成，每个扇区 32 字节。 L2 缓存有一个专用的 32 字节取数接口，可在每个周期内提供 32 字节的数据。 L1 D-cache 有三个端口 - 两个读端口和一个写端口。写入具有比读取更高的优先级，并且用于从 L2 读数的写入具有比 store 指令的写入更高的优先级。

6. FXU 和 VSU

6.1 FXU

POWER7 的 FXU 有两条相同的流水线(FX0 和 FX1)组成，每条流水线的核心是通用寄存器 GPR，围绕它工作的有：算术和逻辑单元（ALU），用于执行加，减和比较指令;rotator，用于执行翻转，移位和选择指令;除法器;乘法器等，其上执行的大部分指令都能 1 个时钟周期内执行完成。两个 GPR 文件的内容由 SMT 模式决定，在 ST 和 SMT2 模式下相同，但在 SMT4 模式下不同。每个 GPR 文件包含 112 个项，可以为两个线程提供寄存器重命名，每个 GPR 有 4 个读取端口，两个为 FXU 提供操作数，两个为 LSU 提供写入数据，有 4 个写入端个，两个接收 FX 的计算结果，两个从 LSU 读取数据。

POWER7 的 FXU 对一些指令做了优化，主要优化如下所述。

- 实现了 64 位完全流水的乘法指令，具有 4 个周期的延迟和每周期一条的吞吐率，其采用 Radix-4 改进的 Booth 编码，可以为计算密集型应用提供较大的加速。
- 增加了一种位操作，可以将来自 64 位源寄存器的任何 8 位多路复用到目标寄存器的低位字节。

6.2 VSU

POWER7 的 VSU 实现了新的 VSX 架构，引入了 64 个寄存器，对于某些特定的问题，其浮点性能提升了一倍，此外，它将先前独立的 VMX 单元和二进制 FPU(BFU)合并为一个单元。由于引入了新的寄存器 VSR，需要将原来的 FPR 和 VR 映射到 VSR 上，32 个 64 位的 FPR 寄存器映射到 VSR 0 到 31 号条目的 0 到 63 位，32 个 128 位的 VR 寄存器映射到 VSR 32 到 63 号条目的第 0 到 127 位。

7. 缓存层次

Power7 的缓存结构针对上一代进行了优化，以适应芯片上核心数量的剧增，其中最为重要改进有两点：

- 以空间换时间。减少了 L1 D-cache 和 L2 cache 的容量，但缩短了访问延迟，并且提升了带宽。
- 将 L3 cache 移到片上。采用了新的 eDRAM 片上 L3 cache，来替换原有的 SRAM 片外缓存，同时，针对 L3 cache 设计了一套启发式的调度替换方法。

7.1 L1 和 L2 的容量减少

L1 D-cache 的容量由 64KB 减少到了 32KB，L2 cache 由 4MB 减少到了 256KB，但同时，他们的访问延迟分别由 0.8 纳秒和 5 纳秒，减少到了 0.5 纳秒和 2 纳秒，面积减少带来的正面效应不仅仅在于延迟，缓存的能量消耗也随着面积的减小而减小。各级缓存容量和大小的变化如表 7-1 所示。

表 7-1

POWER6 (assuming 5-GHz core)	*POWER7 (assuming 4-GHz core)*
64 KB store-through L1 D-cache 0.8ns latency, 80 GB/s private	32 KB store-through L1 D-cache 0.5ns latency, 192 GB/s private
4 MB store-in L2 cache ~5.0-ns latency, 160 GB/s private	256 KB store-in L2 cache 2.0-ns latency, 256 GB/s private
	4 MB partial victim local L3 region ~6.0-ns latency, 128 GB/s private
32 MB victim L3 cache ~35-ns latency, 80 GB/s shared by 2	32 MB adaptive victim L3 cache ~30-ns latency, 512 GB/s shared by 8

POWER7 的 L2cache 对应于核心，每个核心拥有自己的一个 256KB 私有 L2 cache，这个 cache 每行 128 字节，采用 8 路关联组成。在每 4 个核心时钟周期中 L2 cache 可以完成一次下列操作 1)core fetches 2) cast-out reads 3) snoop intervention reads 4) write-backs of miss data 。

POWER7 采取了一些列降低缓存能耗和延迟的操作，包括

- L2 高速缓存中的一些结构以核心频率运行，而其他结构以核心频率的一半运行。连接核心的接口，缓存目录单元等靠近核心一侧的部件以核心频率工作，而控制逻辑和缓存数据阵列以核心频率的一半工作。这种两个时钟频率的独特组合可优化数据带宽和延迟，同时减少能量消耗和缓存区域的面积。
- 对于读取-更改-写回 操作，如果操作的对象是一个 8 字节长的双字，则仅需要在缓存阵列中更新这 8 个字节即可，而不用更新整行。
- 核心对于 L2 cache 的读取多是以预测性的方式进行的，并且读取操作需要两个周期，如果执行完第一个周期就发现错误，则不再执行第二个周期的操作。
- read/claim（RC）machine 被分为两类：a）一组 8 个功能齐全的 RC machines，它们被整合到 L2 缓存控制器中;b）一组 24 个更节能的“轻量级”RC machines，它们被整合到 L3 缓存控制器中。两种 RC machine 分别应对不同类型的操作，从而减少能源使用。

7.2 片上 L3 cache

POWER7 使用 IBM 的高密度低功耗 eDRAM 技术来实现 L3 cache，每一个核心拥有一个 4MB 的本地 L3 区域，它紧密耦合到与每个核心的私有 L2。本地 L3 区域中的每个存储元件仅需要一个晶体管，而传统密集 SRAM 技术需要六个晶体管。 与传统 SRAM 形成鲜明对比的是，eDRAM 只需要三分之一的面积，并且能量消耗只有 SRAM 的五分之一，包含 32 MB L3 cache 的 eDRAM 占据的 POWER7 芯片面积不到 15％。由此产生的面积和能量减少是能够将八个核心集成到单个芯片中的主要因素。

由于 POWER7 32-MB L3 缓存位于处理器芯片上，因此与 POWER6 片外 32 MB L3 缓存相比，它提供了更低的延迟访问，同时也减少了芯片所需的通信资源，这使通信资源能够被部署给存储器和 I / O 子系统接口。

一套启发式的调度替换方法被设计用于管理 POWER7 的 L3 cache，其中最重要的算法就是替换算法，替换算法的核心思想为：

将 L3 中所有数据行分为两类：

- 从本地 L2 中换出的数据，如图中 1 所示，来自于图中的 A 操作。
- 从其他 L3 中调入的数据，如图中 2 所示，来自于图中的 B 操作。

第“1”类数据被第“2”类数据的权利更高，当本地 L3 空间不足时，优先将第“2”类数据换出。具体过程如图 7-1 所示。

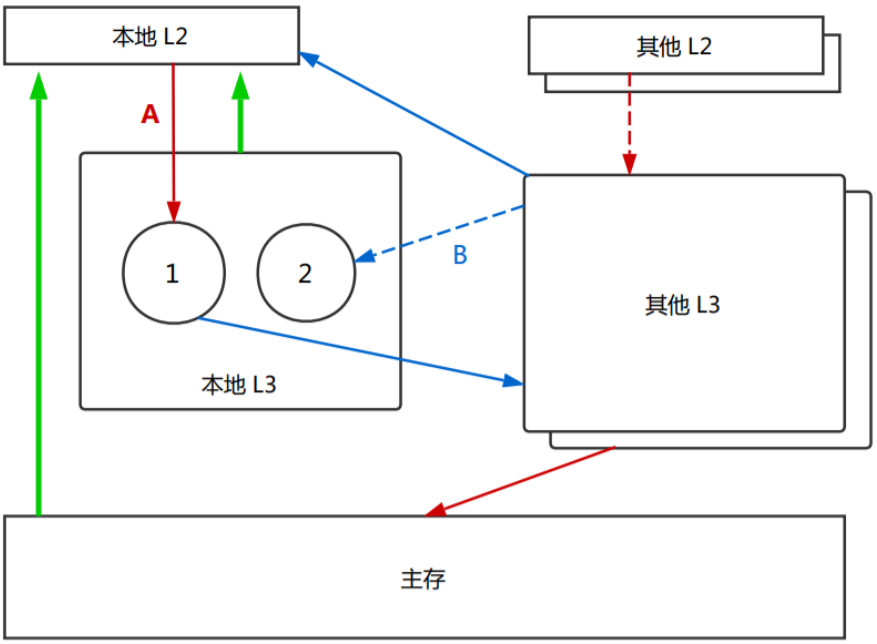

图 7-1

一个核心除了可以利用本地的 4MB L3 cache 外，在某些情况下，它还可以其他 7 个 L3 区域作为 L3 cache，从而，一个核心可以利用最多 32MB 的自适应 L3 cache。当一个数据行被从本地 L3 cache 换出时，如果被换出的数据行是第 2 类，则它被写回主存中。如果被换出的是第 1 类，则一组启发式方法权衡缓存状态和缓存容量，以确定是否将数据行横向换出到另一个 L3 区域。如果要换出，则另一组启发式算法要确定发送到其他 7 个 L3 区域中的哪一个。

8. I/O 子系统

8.1 片内互连

POWER7 的 8 个核心间主要通过一致性互连和数据互连来沟通，片上互连工作在一个频率（称为片上总线频率），片上总线频率是静态的。

核心间的数据一致性由片上一致性互连维护，它被分为两个部分，一个用于处理奇数缓存行，另一个处理偶数缓存行。当一个核心想要发送一致性请求时，首先要向仲裁逻辑请求，被授权之后，才能将一致性请求广播出去，发送给其他芯片的请求将会交由 SMP 互连部件处理，同样，SMP 互连部件也会接收来自其他芯片的请求。因为 POWER7 采用的是一个非阻塞的一致性广播协议，所以整个系统的一致性处理速率取决于处理速率最慢的那个设备，

为了让所有设备都正确接收到一致性请求，POWER7 采用系统固件来协商下限频率。

片上数据互连由 8 根 16 字节的总线组成，它们被分成多个段，每个存储器控制器有两个 16 字节的 on-ramp 和两个 16 字节的 off-ramp，每个核心对应的缓存区域有 1 个 on-ramp 和一个 off-ramp。

8.2 多芯片互联

POWER7 芯片间的互连采用了两级连接，最多可连接 32 个 POWER7 芯片，形成一个 256 核 SMP 系统。第一级连接最多可以结合 4 个芯片，每一个芯片都和其他所有芯片相连。第 2 级连接最多可以结合 8 个一级节点，每一个一级节点都和其他 7 个相连接，所以最多支持 32 个芯片互联。

在整个部分中，有一个功能特别引起了我的注意，POWER6 中引入了局部一致性广播功能：将一致性请求分为两部分，一部分仅在节点内广播，而另一部分广播到整个系统。利用一种范围预测法可以确定广播的范围，我观察了 POWER7 的 cache state 之后觉得，cache 状态中的 IG 和 IN 可能与这个预测法有关，IG 对应的有效数据行可能存在于整个系统，而 IN 对应的有效数据可能仅存在于本地。一致性流量与核心数目的平方成正比，这种局部广播的一致性协议，节省了很多的一致性流量，使得 POWER7 系统可以容纳更多的核心。[2]

8.3 集群互联

在配合专用的集群互联芯片之后，POWER7 可以扩展为一个大型的集群系统，4 块 POWER7 芯片连接到 1 个集群互联芯片，通过多级连接之后，整个系统最多可以容纳 512-K 个处理器。

9. 总结与阅读感想

IBM 在 POWER7 处理器引入了诸多技术改进，其中一项技术改进是技术驱动的：将片外 L3 cache 改为片内 L3 cache，其余的改进基本都是由需求驱动的，POWER7 要实现由双核过度到 8 核；由双线程扩展到 4 线程，其他众多的细节技术改进大多都是为了实现这个而提出的。

这次我从十月末开始阅读《IBM POWER7 multicore server processor》一文，一直到 11 月 19 日结束。这是我第一次细读一篇专业论文，拿到论文之后，我第一遍采用的是粗读，看了各个章节标题和前几段。第一遍读来并不是很流畅，虽然大概知道了论文有哪些部分，但是很多作业词汇不认识，很多基本概念不熟悉，阻碍了我继续精读下去的计划。

于是我把不知道的概念记录下来，又把《计算机体系结构——量化研究方法》找了出来对照着去看，这一看大概花了一周多的时间，强化理解了一些概念，解决了一些问题，之后才开始了为其一周的细读。这次细读感觉就要比第一次好得多，解决了一些困惑之后就此顺推又解决了其他一些困惑，当读到一致性协议这个部分时，我对于 POWER7 的局部一致性广播产生了探究的兴趣，它是在 POWER6 文章中引入的概念，我又把 POWER6 的论文找来去看 cache 一致性这部分。

由于是第一次详细阅读专业论文，还没有掌握论文的定式，我把阅读的目标定为了初步理解论文中描述的系统，所以我的阅读思路是“来者通吃”，而没有特别去阅读论文中的创新点，阅读报告也是基于这个思路。

参考文献

[1] IBM POWER7 multicore server processor[J]. IBM Journal of Research & Development, 2011, 55(3):1-1.
[2] Power I . IBM POWER6 microarchitecture[J]. IBM Journal of Research & Development, 2007, 51(6):639-662.
[3] John L. Hennessy. 计算机体系结构 量化研究方法(第 5 版)[M].北京:人民邮电出版社,2013.1