

DOI:10.19651/j.cnki.emt.2106910

# 基于FPGA的160 Gbit/s网络数据包过滤系统设计

周榕 翁天恒 陈天杨 杨鸣 张俊杰

(上海大学特种光纤与光接入网重点实验室 上海 200444)

**摘要:** 网络流量分析系统是保障网络空间安全的重要手段之一,但传统的分析处理系统难以应对海量的网络数据,亟需特定设备对网络数据包进行预过滤处理,以减轻分析处理系统的负担。针对现有需求,设计了一种基于FPGA的160 Gbit/s网络数据包过滤系统。系统首先对16通道输入的网络数据包进行解析,在提取关键字段后,采用哈希表的方式和轮询仲裁机制实现多通道查询,最终过滤输出符合规则的数据包。仿真与板级测试结果表明,系统平均处理延迟为1.5  $\mu\text{s}$ ,有效过滤比达99.5%以上,满足16通道、单通道10 Gbit/s线速过滤要求。在XC7VX485T芯片上满足40 960条关键字段查询时,存储资源占用不到80%,具有高吞吐、低延迟和高可靠性的特点,在主干网、数据中心等网络安全领域具有一定的实际应用价值。

**关键词:** FPGA;哈希表;包过滤

**中图分类号:** TN919 **文献标识码:** A **国家标准学科分类代码:** 510.50

## Design of 160 Gbit/s network packets filtering system based on FPGA

Zhou Rong Weng Tianheng Chen Tianyang Yang Ming Zhang Junjie

(Key Laboratory of Specialty Fiber and Optics Access Networks, Shanghai University, Shanghai 200444, China)

**Abstract:** The network traffic analysis system is an important technology that ensures the security of cyberspace. The traditional analysis system is difficult to deal with the massive amount of data. It is urgent for a specific technology to pre-filter the network data packets in order to reduce the burden of the analysis system. Considering the need, this paper designs a 160 Gbit/s network packets filtering system based on FPGA. The system first parses the input network data packets, extracts the keywords, and then uses a hash table to lookup. A round-robin arbiter is designed for the multi-channel query. Finally, filter and output the packets that meet the requirements. System-level simulation and board-level test results show that the average delay during processing is 1.5  $\mu\text{s}$ , and the effective filtering ratio is over 99.5%, which meets the requirement of 16 \* 10 Gbit/s line-rate filtering. On the XC7VX485T FPGA, RAM resources occupy less than 80% when supporting 40 960 keywords queries. The system is high-throughput, low-latency, and high-reliability. It is practical and has great value in network security fields such as backbone networks and data centers.

**Keywords:** FPGA; hash table; packet filter

## 0 引言

随着互联网规模不断增大,海量的网络数据给网络空间安全管理带来了极大地挑战,而网络流量分析系统是保障网络空间安全的重要手段之一<sup>[1-4]</sup>。网络流量分析系统一般由流量采集、分析和处理等部分组成,目前的处理分析系统难以应对繁多的网络数据,需要进行分析处理的数据往往只占海量数据的小部分,尤其在对主干网或者数据中心的网络数据流进行分析处理时,亟需特定设备对高速数据包进行预过滤,再交由后续网络分析系统处理,以

保障网络安全<sup>[5]</sup>。

为了解决网络数据包的过滤问题,需要根据特定的规则对网络数据包进行分类,而包分类技术一直以来是网络技术中的研究热点。传统的包分类算法主要包括穷尽查找、分解算法、决策树算法和元组空间算法,而根据实现方式不同又分为软件算法和硬件算法<sup>[6-7]</sup>。

在软件算法中,文献[8]利用多核CPU的并行运算特性,分别实现了基于哈希函数的线性查询匹配算法、基于分层查找的树形查询匹配算法和基于分层查找的树型匹配算法,这几种算法在小规模规则集时差异不大,而随着规则集

的增大,分层查找的树型匹配算法效果更优,同时并行规则匹配较串行规则提升了 10 倍性能,但这些算法仍存在有在规则集扩大后匹配时间线性增加的问题。

为进一步提升查询匹配性能,人们常利用硬件方式实现加速,文献[9]通过结合 GPU 异构并行计算设计了包分类算法,该算法利用对网络流的特性进行维度分解和对规则字段采用分类查找,从而避免扫描整个规则集以提高查找匹配效率,由于利用了 GPU 并行计算的特点,相较于纯软件实现进一步提升了性能,但算法匹配的时间复杂度仍然较大,无法满足高带宽的网络数据分类需求。

TCAM 技术常用于硬件查找<sup>[10-11]</sup>,而 FPGA 有并行流水的特点,越来越多的被应用于高速网络加速处理当中<sup>[12-13]</sup>。文献[14]提出并设计了基于 FPGA+TCAM 架构的网络分流系统,通过将网络数据包中的某些字段到 TCAM 查找从而确定该包是否进行过滤,相较于传统 CPU+网络处理芯片+TCAM 的结构<sup>[11]</sup>,极大提升了接入带宽和处理效率,可满足 40 Gbit/s 总带宽的网络分流需求;文献[15]在此基础上利用布隆姆过滤器对需要查询的字段进行预过滤,然后再通过 TCAM 进行查询,预过滤处理可极大地减少 TCAM 查询频次,从而进一步提升了系统的网络数据包分流效率。虽然采用 TCAM 可以带来极大的查询性能提升,但存在着高功耗和不易扩展的问题<sup>[16-17]</sup>。

以上网络数据流分类过滤方案均存在着各自的不足,为了克服上述方案存在的缺陷以及局限性,本文基于 FPGA 提出一种实现高速网络数据流分类过滤的方案。该方案首先分通道对接收的以太网数据帧进行解析,对解析得到的关键字段采用哈希表的方式进行查询匹配,最终根据查询匹配结果对以太网数据帧进行过滤输出。

本文的主要工作如下:1)设计了基于哈希表的关键字段查询匹配算法,并针对算法进行硬件结构优化,完成对高速网络数据包的分流;2)设计了多通道查询匹配仲裁机制,实现对 16 通道、单通道 10 Gbit/s 网络数据流的线速过滤处理;3)设计了数据包过滤系统的仿真和板级测试实验,并针对结果进行分析。

综上,该系统设计可有效过滤出特定网络数据包,易于 FPGA 并行实现以提高数据处理带宽,具有高带宽、低延迟和分类过滤性能好的特点,在主干网、数据中心等网络安全领域具有实际应用价值。

## 1 基于哈希表的关键字段查询匹配结构

### 1.1 查询匹配算法选择

在包分类过滤问题上,关键在于将数据帧解析得到的关键字段在已知的字段集合中进行匹配。常规的查询算法存在时间或空间复杂度大的弊端,而 TCAM 技术存在功耗高、插入更新慢和不易扩展的问题,如果采用 SRAM 实现 TCAM 又将带来较大的存储资源开销的问题,且结构复杂不易实现<sup>[17]</sup>。

哈希函数是一种将任意长度关键字段映射到固定长度数据的函数,而哈希表是一种可以根据映射后的关键字段直接查询存储内容的数据结构。采用哈希表方式进行字段的查询匹配可使时间复杂度降到最低,便于在 FPGA 上硬件实现,但可能存在哈希冲突,即存在经过哈希函数计算后的多个不同字段映射为同一个值的情况。可采用合适的哈希函数或预估措施来避免哈希冲突,一般采用开放定址、再哈希、链地址和建立公共溢出区等方法。常见利用哈希表实现字段查找的算法有布隆姆过滤器(bloom filter)和布谷鸟过滤器(cuckoo filter)等,其中布隆姆过滤器及其变体<sup>[18]</sup>通过对同一个字段进行多次哈希计算映射到数组的多个位置从而降低哈希冲突,而利用布谷鸟哈希算法的布谷鸟过滤器及其变体<sup>[19-20]</sup>通过两次哈希函数计算实现对字段的查询,相较于布隆姆过滤器性能更优。

综上,本文选择采用基于哈希表的关键字段查询匹配算法,对哈希表结构进行 FPGA 硬件化设计,从而实现高带宽数据包的快速查询精准匹配。

### 1.2 基于哈希表的关键字段查询匹配结构及其优化

由于布谷鸟哈希的插入时间复杂度较高,无法在确定的时钟周期内完成字段的插入<sup>[20]</sup>,算法也难以在 FPGA 上硬件实现,而布谷鸟过滤器在查询时又存在一定的误判几率。因此,为降低哈希表的插入、查询和删除的时间复杂度,满足动态更新和频繁精准查询的需求,本文在布谷鸟过滤器的基础上进行改进,提出如图 1 所示的三元冗余度哈希表,即每个哈希表地址对应的槽有 4 个。

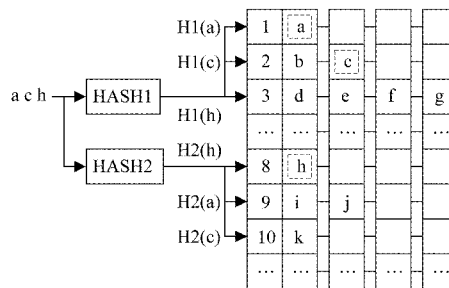


图 1 三元冗余度哈希表插入示意图

当需要插入字段时,采用两种不同哈希函数分别对待插入的字段进行计算,计算值作为哈希表的地址,可获得 2 个地址的 8 个槽位。当 2 个地址均有空余槽位时,将该字段插入到空余槽位较多的地址中,以保证插入均衡,如图 1 所示字段 a 的插入过程,a 经过 H1 和 H2 计算得到两个地址 1 和 9,其中地址 1 的槽位空余数量多于地址 2,故将 a 插入到地址 1 的空余槽位中。只有 1 个地址具有空余槽位时,则将待插入值直接插入到该地址的空余槽位中,如图 1 中插入字段 h 所示。若两个地址均无空余槽位时,为避免插入时间的不确定性,不再进行布谷鸟哈希的重定位操作,认为解决哈希冲突失败,不将字段插入,这样设计以便于 FPGA 硬件实现。

当进行查询时,将待查询的字段进行两次哈希计算,将2个地址对应的8个槽位中的字段全部取出,进行一一比较,若有相同的则认为查询成功,否则认为待查询的字段不存在。

当进行删除时,将待删除的字段进行两次哈希计算,得到2个地址中的8个字段与待删除字段进行比较,若一致则将对应槽位插入0值,如不存在一致的则认为待删除的字段不存在。

上述结构可以在降低哈希冲突概率的同时,保证哈希表的插入、删除和查询的时间复杂度均为 $O(1)$ ,可满足动态实时更新和快速查询的需要;哈希表的槽位中存放完整字段可保证精准查询;该结构算法复杂度较低可减少逻辑

资源开销,便于在FPGA中实现,仅使用一般的随机存取存储器资源,相较于基于TCAM具有更低功耗的优势。本文关键字段的查询匹配仲裁模块和插入与删除模块的设计就是围绕该结构展开的。

## 2 网络数据包过滤系统设计

图2是本文设计的基于FPGA的160 Gbit/s网络数据包过滤系统整体结构,该系统设计基于Xilinx的FPGA芯片,其具体型号为XC7VX485TFFG1761-2,FPGA开发软件采用Vivado Design Suite 2018.3,仿真软件采用Mentor Graphics的Modelsim 10.7。

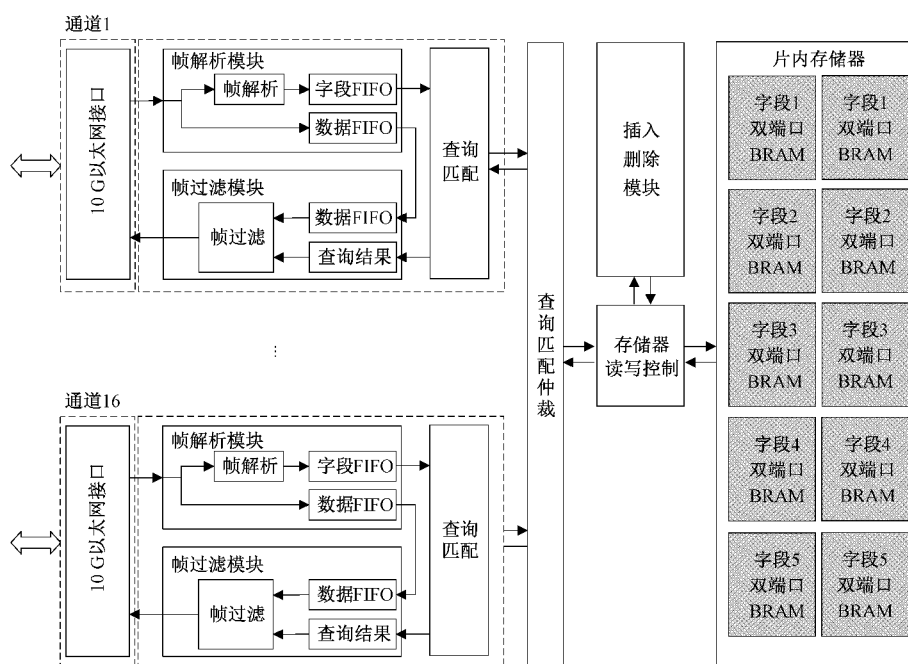


图2 160 Gbit/s网络数据包过滤系统整体结构

硬件系统结构包括16个单通道数据处理模块、多通道查询匹配仲裁模块、存储器读写控制模块、规则字段插入删除模块和片内存储器等部分。其中单通道的数据处理模块由10 G以太网接口模块、数据帧解析模块、查询匹配模块和数据帧过滤模块等部分组成,片内存储器可满足5种字段的插入删除和查询的需求。

各通道接收到以太网数据帧后进行跨时钟域转换,完成后将完整数据传输到帧解析模块进行缓存,并将解析得到的关键字段也进行缓存,交由查询模块读取并申请查询。查询匹配仲裁模块按照轮询优先级对各通道的申请进行仲裁,获得申请的通道被允许查询片内存储器,并将查询结果传给帧过滤模块,同时将该数据帧缓存在帧过滤模块中,并根据匹配结果决定是否需要丢弃。过滤后的数据帧根据配置的通道进行发送,此外,片内存储器存有的字段集合通过插入删除模块进行更新和删除。

### 2.1 帧解析模块设计

帧解析模块是将接收到的以太网数据帧进行解析,获取关键字段,以便后续模块进行处理。以太网帧结构如图3所示,除去前导码、帧起始符和帧校验序列外,其余部分由以太网首部和以太网负荷组成。在本文设计中,所接收到的以太网帧会存在由前级设备为每帧数据添加的标签字段,置于以太网负荷的尾部,标签中包含3个位宽为56 bit的关键字段,故本模块需将位于IP首部的源IP地址和目的IP地址,以及位于帧尾的数据标签中关键字段进行解析提取。该模块设计采用流水线处理方式,接收数据帧存入缓冲FIFO的同时对数据进行解析,解析字段的信息在FIFO存储完毕的同时获得,并被存入到解析字段缓存FIFO中,以待后续模块读取。

在以太网数据传输标准中,帧与帧之间需存在帧间隙。此外,以太网帧存在前导码、帧起始符和帧校验序列等部分,故在万兆线速传输时,仅包含以太网首部和以太

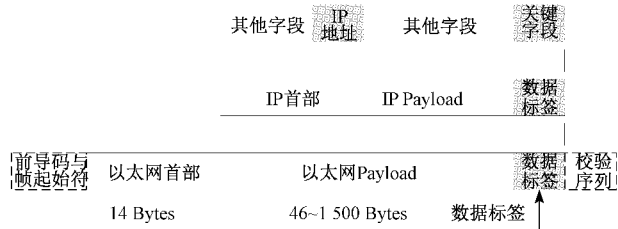


图 3 帧结构示意图

网负荷的以太网帧之间将存在 2~3 个时钟周期的间隙，而帧解析模块处理的时序如图 4 所示，在帧间隙设置为 1 个时钟周期时，可以进行正常解析处理，说明在 156.25 MHz 时钟频率、数据位宽为 64 bit 的情况下，可完全满足线速解析处理要求。



图 4 帧解析模块线速处理时序图

### 2.2 插入删除模块设计

本文设计利用哈希表来实现关键字段查询匹配，哈希表采用双端口块随机存取存储器(BRAM)进行存储，该存储器具有两个读写端口，两个端口可同时读或者写，单个哈希表的深度为 8 192；哈希表单个地址采用三冗余设计，可存储 4 个关键字段，用以避免哈希冲突；单张表满足 8 192 个以上关键字段的插入、查询与删除。以 IP 地址字段哈希表为例，对哈希表进行插入和删除操作的硬件实现结构如图 5 所示。

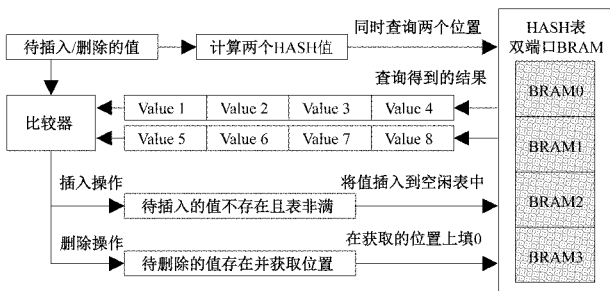


图 5 哈希表插入删除操作的硬件实现结构

当需要插入或删除某一字段时，先对待插入或者待删除的值同时进行两次哈希计算，第 1 次计算采用 CRC32 硬件实现算法，计算值根据表的深度取低位作为哈希表的第 1 个地址，第 2 次计算是对第 1 个地址循环左移 1 位，保证之后插入的均衡，得到了哈希表的第 2 个地址，该过程可单个时钟周期完成。当需要进行插入时，利用双端口 BRAM 的特点，将计算得到的 2 个地址的 8 个值在 1 个时钟周期内读出，并与待插入的值进行比较，将存在 4 种情

况：1)待插入值已经存在，则将待插入值丢弃；2)8 个位置均已占满，无法解决哈希冲突，将待插入值丢弃；3)1 个地址均有空位且待插入值不存在，则将待插入值写入到两个地址中空位较多的地址去，保证插入均衡，可进一步降低哈希冲突概率；4)只有 1 个地址有空位且待插入值不存在，则将待插入值插入到该地址的空位位置上。当需要进行删除时，同样将得到的 8 个值与待删除值进行比较，若待删除的值在哈希表中存在，则将对应位置的值写 0，不存在则不写入值。该结构完成一次对哈希表进行插入或者删除操作时，需要消耗 3 个时钟周期。

为进一步验证哈希表的抗碰撞性能，本文设计仿真实验进行验证，以 IP 地址哈希表为例，设置表的深度为 8 192，三冗余结构，理论最多可插入 32 768 个 IP 地址。先连续插入不同数量的 IP 地址到哈希表中，再统计哈希表中各地址的插入情况，重复测试多次。其中每次测试中插入的 IP 地址分为随机值和顺序值，随机值采用 3 种不同的伪随机序列产生，对应随机值 A、B、C，而顺序值即每次测试的初始 IP 地址设置为随机值，后续的 IP 地址在初始值的基础上累加产生，累加步进值分别为 3、5、7，对应顺序值 A、B、C，多次测试的结果统计如图 6 所示，有效插入率由式(1)计算得到。

$$\text{有效插入率} = \frac{\text{插入成功个数}}{\text{插入总个数}} \times 100\% \quad (1)$$

平均有效插入率为对插入 6 种不同值的有效插入率计算平均值得到。

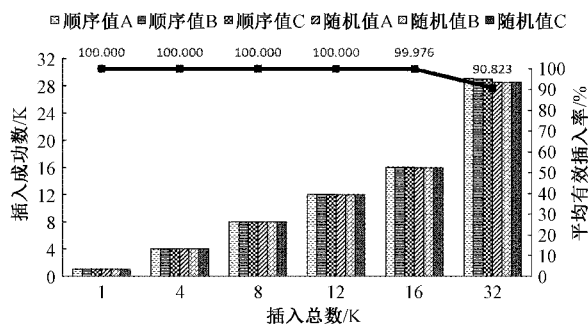


图 6 哈希表插入仿真测试结果统计

对结果进行分析发现，在表的深度为 8 192 的情况下，插入 12 288 个字段的插入有效插入率为 100%，而当插入 16 384 个字段仅有 0.03% 的概率因无法解决哈希冲突而导致插入失败。由此可知，本文设计的单个哈希表可满足至少 8 192 个字段的插入需要，同时该哈希表可支持扩展，深度由存储器大小决定，可通过使用片外存储器实现更大规模的规则集，以供查询匹配使用。

### 2.3 查询匹配仲裁模块设计

查询匹配仲裁模块主要完成对单通道接收解析的关键字段进行查询申请和字段匹配，并对多个通道的查询申请进行仲裁，本文采用的硬件实现结构如图 7 所示。

当某个通道接收到数据帧并完成解析后，即帧解析模



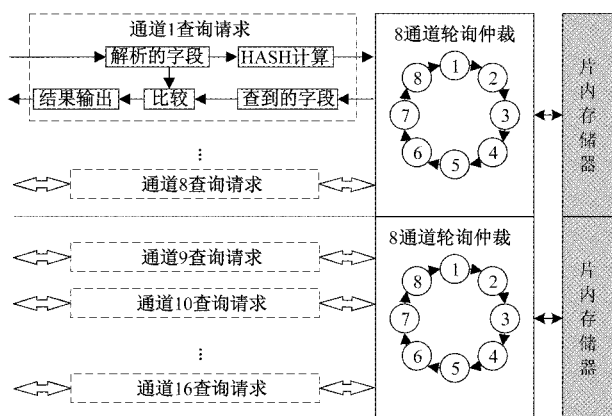


图7 查询匹配仲裁模块硬件实现结构

块中解析字段缓存FIFO非空时,该通道的查询请求模块将读取解析的关键字段(本设计包含5个关键字段),并对每个字段进行哈希计算,所使用的哈希函数与插入删除模块中的一致,计算得到的2个值作为读哈希表的地址。在对5个关键字段进行并行计算后,得到10个地址后向轮询仲裁模块进行仲裁,当该通道获得仲裁后,将计算得到的地址分别作为片内存储器中不同字段哈希表的地址,分别读出8个值返回进行匹配,若解析所得字段与读出的8个值中任意一个值一致,则认为该数据帧符合过滤规则,并向帧过滤模块返回“1”,否则返回“0”,由后续模块完成数据帧的过滤。

为提升查询匹配性能和节省片内存储资源,本文利用双端口RAM的两个端口可同时读写的特点,通过8通道轮询仲裁器对8个通道的查询请求进行仲裁,其中仲裁器采用轮询仲裁机制,即1个时钟内只查询1个通道,各通道按照通道序轮询获得,8个时钟作为1个轮询周期,各通道仲裁的时序示意图如图8所示。仲裁得到某个查询请求通道后,由于查询每个字段需要读取2个地址的值,而双端口RAM可满足1个时钟周期完成两次读操作,该过程可流水实现,将进一步提升查询效率。由上述设计可知,实现16通道的查询匹配,只需两块相同的存储器即可满足需求。

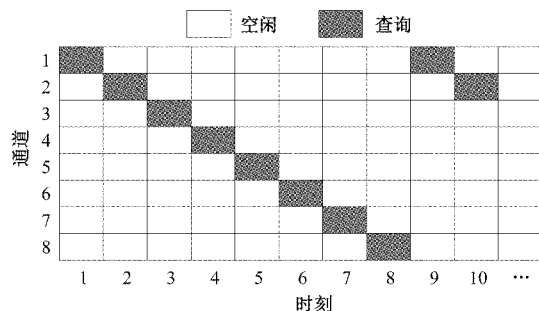


图8 轮询仲裁时序示意图

以太网规定最短帧为64 Byte,最长帧为1 518 Byte,除去4 Byte的帧校验序列,则以太网帧长范围在

60~1 514 Byte,可分析极端情况:对于万兆以太网,当8个通道均线速接收帧长为60 Byte的数据帧时,根据每个时钟传输8 Byte,通道每8个时钟周期完成一帧传输且只读取一次RAM,故当通道1获得仲裁并查询时,通道2需要等待1个时钟,以此类推,通道8需要等待7个时钟后才能查询,故当缓存深度超过可容纳60 Byte以上的数据时,可以保证各通道均可查询且无数据帧丢失。

在上述极端情况下,本文设计的结构依然可满足多通道查询。而在实际传输中,帧与帧之间必然存在一定的帧间隙,且并非长时间为60 Byte长度的数据帧传输。本文针对实际情况,设计多组仿真实验进一步验证查询匹配过滤的性能。

3组单通道查询匹配过滤仿真实验:1)单通道发送定长数据帧,帧长为60 Byte,帧间隙为2个时钟,发送的数据帧全部符合过滤规则,过滤后从单个通道输出;2)单通道发送定长数据帧,帧长为1 514 Byte,帧间隙为2个时钟,发送的数据帧全部符合过滤规则,过滤后从单个通道输出;3)单通道发送随机帧长的数据帧,帧长为60~1 514 Byte不等,帧间隙为2个时钟,发送的数据帧全部符合过滤规则,过滤后从单个通道输出。仿真实验结果如图9所示,可知从数据输入到过滤输出延迟为371.2~2 662.4 ns,平均延迟为1 516.8 ns(1.5  $\mu$ s),与输入帧长相关。

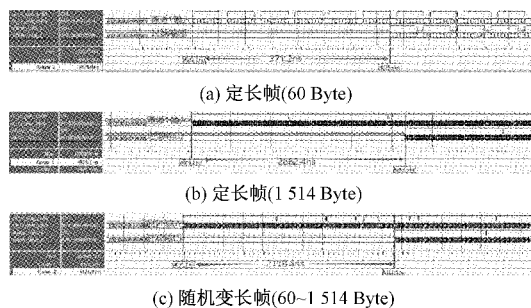


图9 单通道查询匹配过滤仿真

两组16通道查询匹配过滤仿真实验:1)16个通道发送定长数据帧,帧间隙为2个时钟,帧长为60 Byte,单通道的数据帧包含符合过滤规则的和不符合规则的,两者之比为1:19,两种数据帧混合发送,经过滤后从同一通道输出;2)16个通道发送变长数据帧,帧间隙为2个时钟,帧长为60~1 514 Byte随机发送,符合过滤规则和不符合过滤规则的数据帧之比为1:19,同样两种数据帧混合发送,经过滤后从同一通道输出,并对输出结果进行统计。仿真实验结果如图10所示,可知符合过滤规则的数据帧均已被过滤出,本文设计的查询匹配过滤结构满足16通道线速处理要求,总体延迟较低。

## 2.4 系统测试与分析

为进一步验证本文设计的高速网络数据流过滤系统在真实情况下的性能,故设计板级测试实验。所设计的160 Gbit/s网络数据包过滤系统如图11(a)所示,其测试

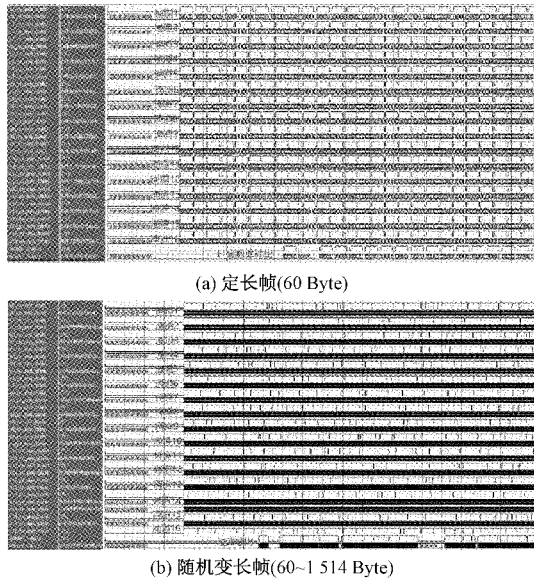


图 10 16 通道查询匹配过滤仿真

平台如图 11(b) 所示,其中网络测试仪采用 Spirent 公司的 SPT N4U,具体板卡为 FX3-25GD-S8,最高可支持 8×25 Gbit/s 以太网测试,测试仪软件采用 Spirent TestCenter Application 4.91,设备间连接使用单模光纤、分光器和光模块若干。

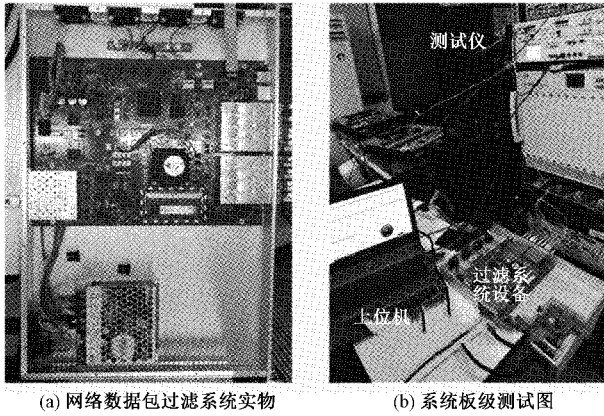


图 11 高速网络数据过滤系统板级测试

为测试单通道过滤性能,将网络测试仪的一个收发端口与过滤系统单个通道相连,构造吞吐量为 10 Gbit/s、帧长为 60~1514 Byte 的随机网络数据流,设置符合过滤规则的关键帧分别占总帧数的 25%、50%、75%和 100%的 4 组测试,测试结果如图 12 所示,其中有效过滤比采用式(2)计算得到。

$$\text{有效过滤比} = \frac{\text{过滤帧总帧数}}{\text{关键帧总帧数}} \times 100\% \quad (2)$$

由单通道测试结果可知,本文设计的过滤系统的单通道可对 99.8% 以上的关键帧进行有效过滤,相较于文献[8-9]存在匹配时间随规则集增大而变长、无法满足线速处理的问题,本文所设计的结构在查询匹配的时间复杂

度为  $O(1)$ ,可流水的对数据包进行过滤,可基本满足线速处理。

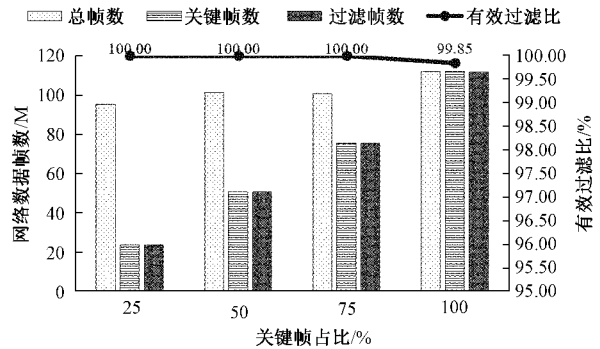
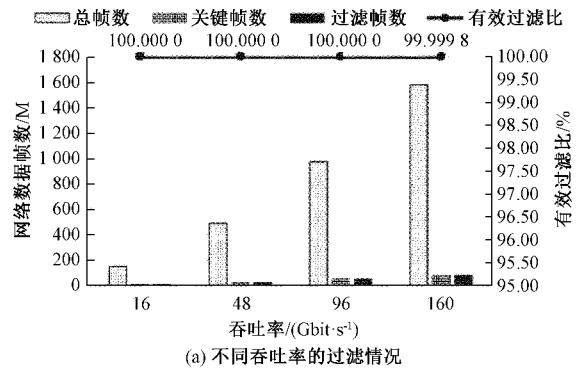
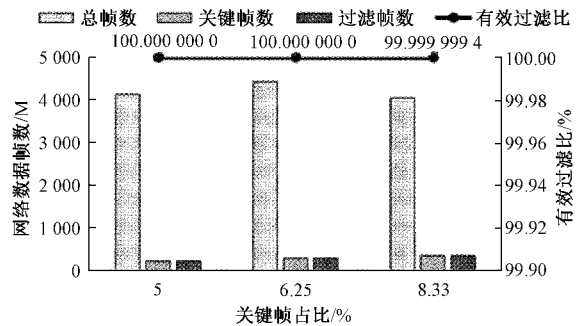


图 12 单通道网络数据流过滤测试

针对 16 通道过滤性能,本文设计不同吞吐率和不同关键帧占比两组过滤测试实验,其中不同吞吐率实验中构造关键帧占总帧数 5%、帧长为 60~1514 Byte 的随机网络数据流,分别设置 16、48、96 和 160 Gbit/s 4 次不同吞吐速率的测试,而不同关键帧占比实验中构造吞吐率为 16~160 Gbit/s、帧长为 60~1514 Byte 的随机网络数据流,分别设置关键帧占比为 5%、6.25%和 8.33% 3 次测试,在两组测试中,16 通道过滤后从一个通道进行输出,并对过滤前后过滤后进行统计,结果如图 13(a)和(b)所示。由 16 通道过滤测试实验可知,在不同吞吐率和不同关键帧占比的情况下,本文设计的过滤系统有效过滤比在 99.9% 以上,均能达到良好的过滤效果且所有通道均满足线速处



(a) 不同吞吐率的过滤情况



(b) 不同关键帧占比的过滤情况

图 13 16 通道网络数据流过滤测试

理,相较于文献[14-16],本设计可处理的总带宽更高,且查询匹配的规则集可扩展。

本文工程基于Xilinx FPGA的XC7VX485TFFG1761-2设计实现,在支持5种共40 960个的关键字段查询匹配时,其资源消耗如表1所示,逻辑资源消耗占FPGA芯片总资源的50%以下,而存储资源占总资源的80%以下,具有进一步提升查询匹配容量的空间。

表1 FPGA工程资源消耗情况

类别	总资源	消耗资源	消耗占比/%
LUT	303 600	145 502	47.93
FF	607 200	239 492	39.44
BRAM	1 030	822.5	79.85

### 3 结 论

本文设计了一个基于FPGA的160 Gbit/s网络数据流过滤系统,接收来自16个通道的网络数据包并进行解析,获得关键字段后进行查询匹配,若接收的符合过滤的规则数据包则输出,否则就作丢弃处理,从而实现对高速网络数据流的过滤。针对关键字段查询匹配设计基于哈希表实现,可支持5种共40 960个以上的关键字段的插入、删除和查询,针对多通道查询匹配设计了查询匹配轮询仲裁机制,可支持16通道的网络数据帧进行关键字段的查询匹配。本文还针对过滤系统的功能与性能,设计并进行仿真和板级测试,仿真结果表明过滤系统的平均整体延迟为1.5  $\mu$ s,实际板级测试表明在不同环境下,均能达到99.5%以上的有效过滤性能,可满足16通道、单通道10 Gbit/s的线速过滤要求。由于本设计结构中关键字段的存储采用片内存储器,存储空间有限,不能满足更大规模集合的查询匹配需求,故未来的工作是利用片外大容量存储器存储规则集,以满足更大规模规则集合查询匹配的应用需求。

### 参考文献

- [1] 张焕国,韩文报,来学嘉,等.网络空间安全综述[J].中国科学:信息科学,2016,46(2):125-164.
- [2] 袁方.基于网络流量的安全态势分析系统设计与实现[D].哈尔滨:哈尔滨工业大学,2020.
- [3] 王玮.基于数据驱动的无线网络安全态势评估[J].国外电子测量技术,2020,39(7):22-26.
- [4] 苏醒.基于网络行为的计算机网络安全预警与响应系统研究[J].电子测量技术,2019,42(21):123-126.
- [5] 赵双.基于多分类器融合的移动网络流量识别方法研究[D].长沙:国防科技大学,2018.

- [6] 张杰鑫,张铮.包分类算法研究综述[J].计算机工程,2015,41(12):111-118.
- [7] 亓亚烜,李军.高性能网包分类理论与算法综述[J].计算机学报,2013,36(2):408-421.
- [8] 刘洋.数据包过滤规则匹配与并行化技术研究[D].北京:北京邮电大学,2019.
- [9] 王君君.基于GPU加速的包分类算法研究与实现[D].广州:华南理工大学,2020.
- [10] VEGESNA S, SHIV V, NOOR M S. A TCAM-based caching architecture framework for packet classification[C]. ACM Transactions on Embedded Computing Systems, 2020, 20(1):1-19.
- [11] 刘宗宝,赵鑫,张力,等.基于分组TCAM的T比特高性能路由器快速查找更新技术[J].计算机工程与设计,2021,42(2):343-348.
- [12] 文丰,韩雨龙.千兆以太网MAC控制器软核设计[J].电子测量技术,2021,44(1):150-155.
- [13] 徐斌,施周荣,胡庆昆,等.基于Kintex UltraScale+器件的200G网络加速卡硬件电路设计[J].电子测量技术,2020,43(13):13-18.
- [14] 鲁佳琪,黄芝平,刘纯武,等.基于FPGA+TCAM架构的网络分流系统的设计与实现[J].微型机与应用,2016,35(15):65-68,71.
- [15] 路琪.高速网络流量分析处理技术研究[D].长沙:国防科技大学,2017.
- [16] 滕飞.基于Trie的流水式IP查找结构的设计与实现[D].大连:大连理工大学,2017.
- [17] JIANWEI Z, RUI Y, XUEFENG C, et al. A resource-saving TCAM structure based on SRAM[C]. 2019 IEEE 5th International Conference on Computer and Communications, 2019: 365-369.
- [18] GUO D, WU J, CHEN H, et al. The dynamic bloom filters[C]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(1): 120-133.
- [19] FAN B, ANDERSEN D G, KAMINSKY M, et al. Cuckoo filter: Practically better than bloom [C]. Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies, 2014, 3(1): 75-88.
- [20] 王飞越.基于负载均衡的高效布谷鸟过滤器研究[D].武汉:华中科技大学,2019.

### 作者简介

周榕,硕士研究生,主要研究方向为FPGA网络通信。  
E-mail: rongod@shu.edu.cn