流量清洗演示系统概要设计方案

2013-7-20

# 1. 引言

## 1.1编写目的

本文档根据流量清洗系统的演示需求，对演示功能进行了划分，设计了与驱动层与阻断设备之间的文件接口，并对接口文件定义、配置文件、管理工具等进行了规定。

## 1.2背景

说明：

1. 项目名称：流量清洗演示系统；
2. 任务提出者：戴琼
3. 开发者：姜磊，刘新星，白旭，杨嘉佳，唐球
4. 用户：

## 1.3定义

## 1.4参考资料

* 1. Nutz说明文档：nutz-1.b.48-manual.pdf
  2. Amchart介绍：<http://www.amcharts.com/>

# 2总体设计

## 2.1基本功能

设计一个J2ee演示界面，可以对硬件组开发的流量清洗设备各个功能进行演示。具体功能包括发包功能、收包功能、协议分流功能以及阻断功能。

* 输入输出需求：
  1. 硬件方面：提供流量清洗设备，采用PCIE接口与上位机进行数据交互。
  2. 软件方面：内核层的驱动，发包程序，J2EE界面演示系统。
* 功能需求：

1. 发包演示功能
2. 收包演示功能
3. 协议分流功能
4. 阻断功能

## 2.2运行环境

开发环境：WIN7，JDK7u25，apache tomcat v7

运行环境：Fedora 14 x86\_64（64位）操作系统，具有空闲的PCIEX8接口的服务器，JDK7u25，apache tomcat v7

## 2.4系统结构

如图2所示，系统分为3个模块：用户模块，内核模块以及硬件模块。硬件模块实现在FPGA板卡上，完成数据包检测，正则匹配和负载均衡功能。数据通过DMA接口传递给上位机，通过网卡驱动把数据包传递给用户层应用程序。



图2 智能网卡系统结构图

# 3用户层接口设计

## 3.1 用户层API接口

API接口定义在libinicpag.h中，本文档规定基本的四个捕包接口和一个调试接口。接口定义如下：

int inicpag\_open(void)

函数说明：打开网卡设备；

返回值：成功返回1, 失败返回－1

void inicpag\_close(void)

函数说明：关闭网卡设备；

void \*inicpag\_get(int sid)

函数说明：包获取函数；从网卡的一个数据队列中读取一个IP包;

sid：指定特定数据队列的序号（0开始的连续数字）。

返回值：报文的IP头指针；如果没有报文，则返回NULL。

int inicpag\_set\_qq\_mode(void)

int inicpag\_unset\_qq\_mode(void)

函数说明：设置成QQ捕包模式；让网卡中只返回QQ数据包（0x02起始，0x03结尾），其余类型的数据包全部丢弃。

返回值：报文的IP头指针；如果没有报文，则返回NULL。

int inicpag\_enable\_regex (void)

int inicpag\_disable\_regex(void)

函数说明：使能正则表达式匹配功能

返回值：报文的IP头指针；如果没有报文，则返回NULL。

int inicpag\_config\_regex(void \* dfa\_transition\_table)

函数说明：配置正则表达式DFA；把最新的正则表达式规则集生成的DFA转移表下发到SRAM中。

dfa\_transition\_table：DFA状态转移表。

返回值：成功返回1, 失败返回－1

## 3.2配置文件定义

inicpag.conf规定了最基本的应用程序参数，可在此基础上扩充，参数定义如下：

dev\_name = inicpag0

#dev\_name = inicpag1

说明：指定网卡设备名，比如eth1，并可以支持写多行等，每行代表一对网口

stream\_num = x

说明：指定网卡分流的数目，比如1，2，3，4，线程分流个数最大16，支持奇数的分流；

use\_regex

说明：指定是否使用正则表达式匹配，注释该行表示不使用

qq\_mode

说明：指定是否配置成QQ捕包模式，注释该行表示不使用

与linux的配置文件定义规范相同，该配置文件中的注释行以#开头。

## 3.3内部接口

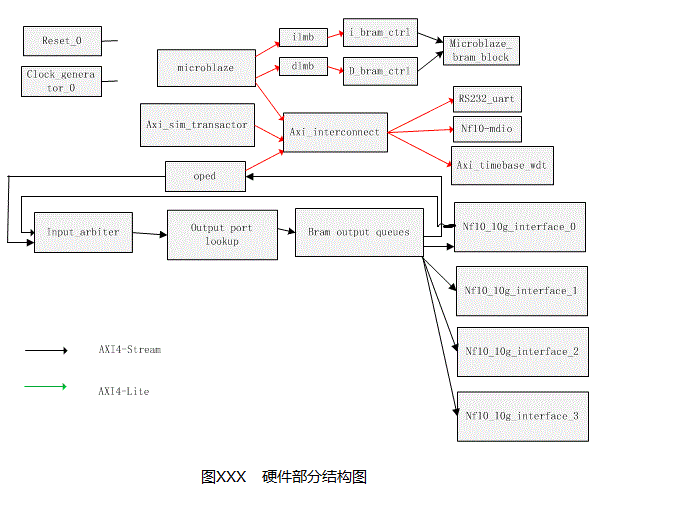
说明本系统之内的各个系统元素之间的接口的安排。

# 4驱动层接口

## 4.1 与用户层的函数接口

驱动程序直接使用Reference\_NIC设计，改动很少，此处接口定义忽略。

# 5硬件层接口



## 5.1 nf10\_input\_arbiter\_0 IP核

1、功能描述：合并一系列的（从nf10\_10g\_interface\_x,x表示0,1,2,3和oped传来的）输入流成一输出流（输出到下一IP核nf10\_nic\_output\_port\_lookup\_0），即将并行数据通路转换为串行数据通路。（补：由于工程采用的仲裁算法为轮询方法，实际上每一次只有一个队列或没有队列将数据包发送到该模块的数据接收端口。）

2、接口

axi\_resetn为输入端口

S\_AXIS\_0细分为输入端口axi\_aclk、输入端口s\_axis\_tdata\_0[255:0]、输入端口s\_axis\_tstrb\_0[31:0]、输入端口s\_axis\_tuser\_0[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready\_0、输入端口s\_axis\_tlast\_0;

S\_AXIS\_1细分为输入端口axi\_aclk、输入端口s\_axis\_tdata\_1[255:0]、输入端口s\_axis\_tstrb\_1[31:0]、输入端口s\_axis\_tuser\_1[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready\_1、输入端口s\_axis\_tlast\_1;

S\_AXIS\_2细分为输入端口axi\_aclk、输入端口s\_axis\_tdata\_2[255:0]、输入端口s\_axis\_tstrb\_2[31:0]、输入端口s\_axis\_tuser\_2[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready\_2、输入端口s\_axis\_tlast\_2;

S\_AXIS\_3细分为输入端口axi\_aclk、输入端口s\_axis\_tdata\_3[255:0]、输入端口s\_axis\_tstrb\_3[31:0]、输入端口s\_axis\_tuser\_3[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready\_3、输入端口s\_axis\_tlast\_3;

S\_AXIS\_4细分为输入端口axi\_aclk、输入端口s\_axis\_tdata\_4[255:0]、输入端口s\_axis\_tstrb\_4[31:0]、输入端口s\_axis\_tuser\_4[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready\_4、输入端口s\_axis\_tlast\_4;

M\_AXIS细分为输入端口axi\_aclk、输出端口m\_axis\_tdata[255:0]、输出端口m\_axis\_tstrb[31:0]、输出端口m\_axis\_tuser[127:0]，输出端口m\_axis\_tvalid，输入端口m\_axis\_tready，输出端口m\_axis\_last

3、与其它模块关系：从上一IP核nf10\_10g\_interface接收数据，经过处理后发送给下一IP核nf10\_nic\_output\_port\_lookup\_0

## 5.2 nf10\_nic\_output\_port\_lookup\_0 IP核

1、功能描述：读取nf10\_input\_arbiter\_0核发送来的报文，根据源端口src\_port判断报文来自网卡的哪个端口，如果是来自eth的报文，就将报文的目的端口dst\_port域映射为oped核对应的端口，如果是来自oped核，就将报文的目的端口dst\_port域映射为eth端口（下一个模块nf10\_bram\_output\_queues\_0就是根据dst\_port域的值将数据包发给对应的端口）

2、接口

axi\_resetn为输入端口

M\_AXIS细分为输入端口axi\_aclk、输出端口m\_axis\_tdata[255:0]、输出端口m\_axis\_tstrb[31:0]、输出端口m\_axis\_tuser[127:0]，输出端口m\_axis\_tvalid，输入端口m\_axis\_tready，输出端口m\_axis\_last

S\_AXIS\_0细分为输入端口axi\_aclk、输入端口s\_axis\_tdata[255:0]、输入端口s\_axis\_tstrb[31:0]、输入端口s\_axis\_tuser[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready、输入端口s\_axis\_tlast;

3、与其它模块关系：从上一IP核nf10\_input\_arbiter\_0接收数据，经过处理后发送给下一IP核nf10\_nic\_output\_port\_lookup\_0

## 5.3 nf10\_bram\_output\_queues\_0 IP核

1 功能描述：接收上一模块传来的数据报文，根据TUSER通道的dst\_port域，将数据报文发送到对应的端口。

2、接口

axi\_resetn为输入端口

M\_AXIS\_x(x表示0,1,2,3,4)口可细分为输入端口axi\_aclk、输出端口m\_axis\_tdata\_x[255:0]、输出端口m\_axis\_tstrb\_x[31:0]、输出端口m\_axis\_tuser\_x[127:0]，输出端口m\_axis\_tvalid\_x，输入端口m\_axis\_tready\_x，输出端口m\_axis\_last\_x

S\_AXIS\_0细分为输入端口axi\_aclk、输入端口s\_axis\_tdata[255:0]、输入端口s\_axis\_tstrb[31:0]、输入端口s\_axis\_tuser[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready、输入端口s\_axis\_tlast;

3、与其它模块关系：从上一IP核nf10\_nic\_output\_port\_lookup\_0接收数据，然后将数据发至对应的nf10\_10g\_interface\_0核或oped核

## 5.4 nf10\_10g\_interface\_x(x表示0,1,2,3) IP核

1 功能描述：a、接收从物理收发器发送过来的数据报文，将输入端口号填充到AXI4-Stream的Tuser通道的SRC\_PORT域中，之后发送到nf10\_input\_arbiter\_0 核;b、将上一IP核nf10\_bram\_output\_queues\_0发送来的报文发送到物理收发器上。此IP核包含了Xilinx XAUI和10G MAC IP（10G MAC需要license），AXI4-Stream adapter。 这几个核的关系是：信号从AEL2005输入至XAUI，再进入10G MAC，最后传进AXI4-Stream。所以nf10\_10g\_interface\_x，具备收(rx)与发(tx)报文的功能

2 接口

输入端口:axi\_resetn、refclk、dclk、xaui\_rx\_IO\_p、xaui\_rx\_IO\_n、xaui\_rx\_I1\_p、xaui\_rx\_I1\_n、

Xaui\_I2\_p、xaui\_I2\_n、xaui\_I3\_p、xaui\_I3\_n

输出端口:xaui\_tx\_IO\_p、xaui\_tx\_IO\_n、xaui\_tx\_I1\_p、xaui\_tx\_I1\_n、xaui\_tx\_I2\_p、xaui\_tx\_I2\_n、Xaui\_tx\_I3\_p、xaui\_tx\_i3\_n

M\_AXIS细分为输入端口axi\_aclk、输出端口m\_axis\_tdata[255:0]、输出端口m\_axis\_tstrb[31:0]、输出端口m\_axis\_tuser[127:0]，输出端口m\_axis\_tvalid，输入端口m\_axis\_tready，输出端口m\_axis\_last

S\_AXIS\_0细分为输入端口axi\_aclk、输入端口s\_axis\_tdata[255:0]、输入端口s\_axis\_tstrb[31:0]、输入端口s\_axis\_tuser[127:0]、输入端口s\_axis\_tvalid、输出端口s\_axis\_tready、输入端口s\_axis\_tlast;

3、与其它模块关系：a、从上一IP核nf10\_bram\_output\_queues\_0接收数据，然后将数据发至物理收发器上；b、接收物理收发器发送过来的数据，处理后，发至nf10\_input\_arbiter\_0核

## 5.5 mdio IP核

1功能描述：用于配置和监听物理层器件的MDIO接口。

2接口

输入端口：PHY\_MDIO\_I

输出端口：IP2INTC\_irpt、PHY\_rst\_n、PHY\_MDC、PHY\_MDIO\_O、PHY\_MDIO\_T

输入输出双向端口：PHY\_MDIO

S\_AXI细分为:输入端口 S\_AXI\_ACLK、S\_AXI\_ARESETN、S\_AXI\_AWADDR、S\_AXI\_AWVALID、

S\_AXI\_WDATA、S\_AXI\_WSTRB、S\_AXI\_WVALID、S\_AXI\_ARADDR、S\_AXI\_ARVALID、S\_AXI\_RREADY

输出端口S\_AXI\_AWREADY、S\_AXI\_WREADY、S\_AXI\_BRESP、S\_AXI\_BVALID、S\_AXI\_ARREADY、S\_AXI\_RDATA、S\_AXI\_RRESP、S\_AXI\_RVALID

3、与其它模块关系：MDIO与AXI interconnect核相连，只能接收AXI interconnect核发来的数据（发送的数据在寄存器通路中传输），不能向AXI interconnect核发送数据