QOS结构介绍及SharkL5QOS设置

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[QOS结构介绍及SharkL5QOS设置 1](#_Toc524509116)

[Revision History 2](#_Toc524509117)

[一 . QOS基本概念介绍、（相关的基础概念介绍） 4](#_Toc524509118)

[1.1 QOS机制简介 4](#_Toc524509119)

[1.2 QOS配置部分 5](#_Toc524509120)

[二 .QOS框架介绍和QOS传输机制 6](#_Toc524509121)

[2.1QOS正常传输机制 6](#_Toc524509122)

[2.2 QOS阈值urgent产生机制 6](#_Toc524509123)

[2.2 QOS Controller结构简介 7](#_Toc524509124)

[三. DMC对QOS的处理 10](#_Toc524509125)

[四.SharkL5的QOS结构及配置 11](#_Toc524509126)

[4.1DPU、DCAM 11](#_Toc524509127)

[4.2MM…………. 12](#_Toc524509128)

[4.3WTLCP…….. 13](#_Toc524509129)

[4.4PUBCP/AUDCP 15](#_Toc524509130)

[五.相关问题预览 15](#_Toc524509131)

### 一 . QOS基本概念介绍、（相关的基础概念介绍）

#### 1.1 QOS机制简介

QOS可以简单的理解为DDR通道仲裁选择的优先级的控制，访问DDR同一时刻目前只有一条通路，但是DMC就包含大约8个port，而每个port都与相对应的subsys相连接，每个subsys都包含多个master。

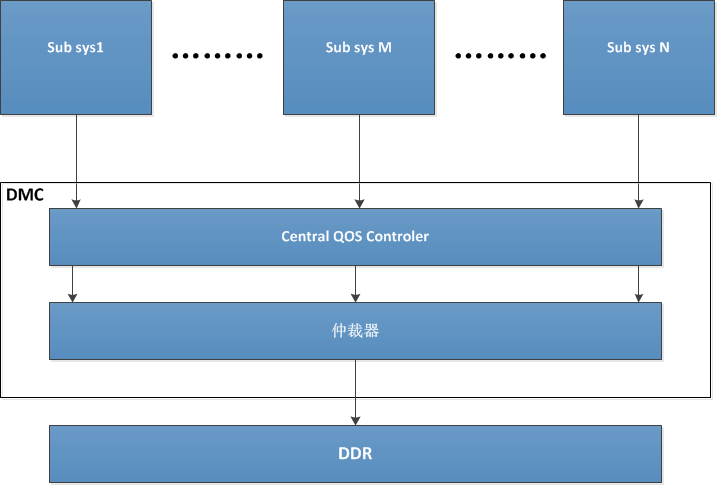
这些多个Master访问DDR就需要一定的优先级顺序来指定，而优先级的值主要根据Master的实时性以及能够wait的时间等条件来决定的。

目前的QOS机制是硬件动态调整，支持HW urgent，Starvation(time-out)urgent,latency monitor urgent 、SW force urgent，如：

1. HW urgent(1):Subsys内有read-time需求的IP，由本身的FIFO or line-buffer status直接产生urgent 信号
2. HW urgent(2):没有real-time需求的IP，在IP内部提高QOS value，由Subsys 的Compare with QOS-threshold产生的urgent
3. Starvation urgent(time-out)：每个通道都可以设置阈值，当outstanding command latency大于threshold时产生urgent。
4. Latentcy monitor urgent：当累加的actural latency大于target latency时可发出urgent。

由于目前有urgent机制，因此Starvation urgent(time-out)目前都是关闭的，主要原因是其无法区分real和no real。

#### 1.2 QOS配置部分



QOS的配置主要包含两个部分，分别是Master的配置和PUB的配置，Master配置包含normal QOS、High QOS(部分)、Threadshold QOS等。

PUB上同时也要配置各个master的QOS，同时控制着QOS的机制的使能，当QOS关闭的时候，那么PUB会屏蔽下面传递过来的QOS，此时所有port的优先级相同。该部分在DDR INIT当中进行设置，当QOS机制被打开时，那么PUB会接收来自下面的Subsys传递的QOS进行DDR通道的仲裁，高QOS的会被优先处理，依次向下。当优先级相同的时候会以轮训的方式进行选择。

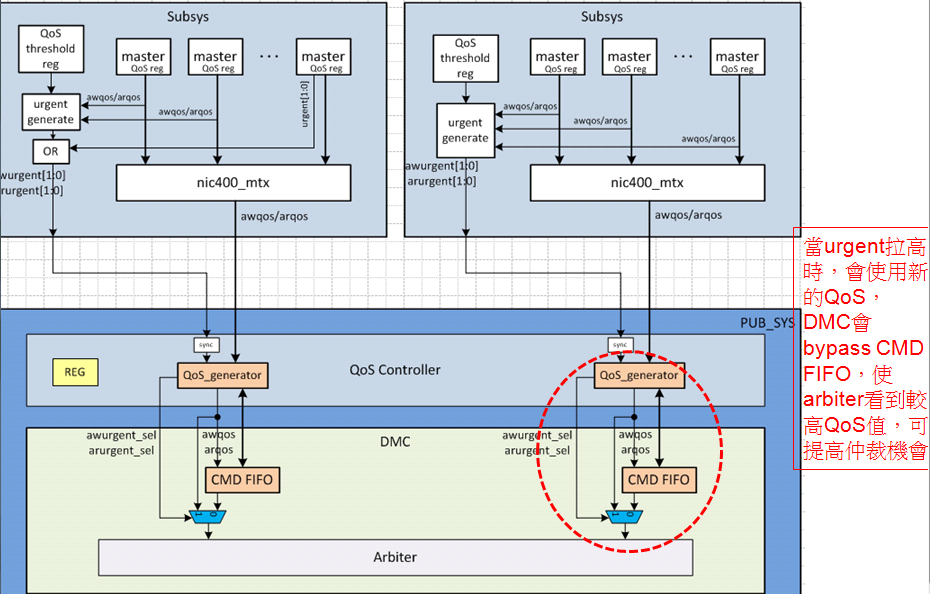
当在DDR初始化使能QOS传输机制时，那么此时就一定需要各个master已经按照要求配置了QOS，否则就有可能出现overflow的现象。因为每个master的QOS默认值是不相同的，最常见的就是WTLCP出现overflow现象。

### 二 . QOS框架介绍和QOS传输机制

#### 2.1QOS正常传输机制

Master配置的QOS会竞争Subsys的端口的输出，高优先级的优先输出到Subsys的端，多个Subsys端口会根据当前输出的QOS值来竞争DDR传输端口，如果遇到QOS值相同的情况，那么就采用轮训的方式依次执行。同时如果一个低优先级的Master正在占用当前的DDR通道，此时出现一个高优先级的进入等待状态，那么也需要等待低优先级的执行完毕，不会出现打断的情况。

#### 2.2 QOS阈值urgent产生机制



上图引用于SharkL3是一个典型的阈值产生机制框图，其中subsys包含多个Master，每个Master的QOS值会传递给Nic400\_MTX并同时传递给当前subsys的urgen产生模块。每个subsys的Nic400\_MTX会根据传递过来的QOS值进行传递，高优先级的会优先有效，如果Master优先级相同，那么采用轮训的方式进行选择。

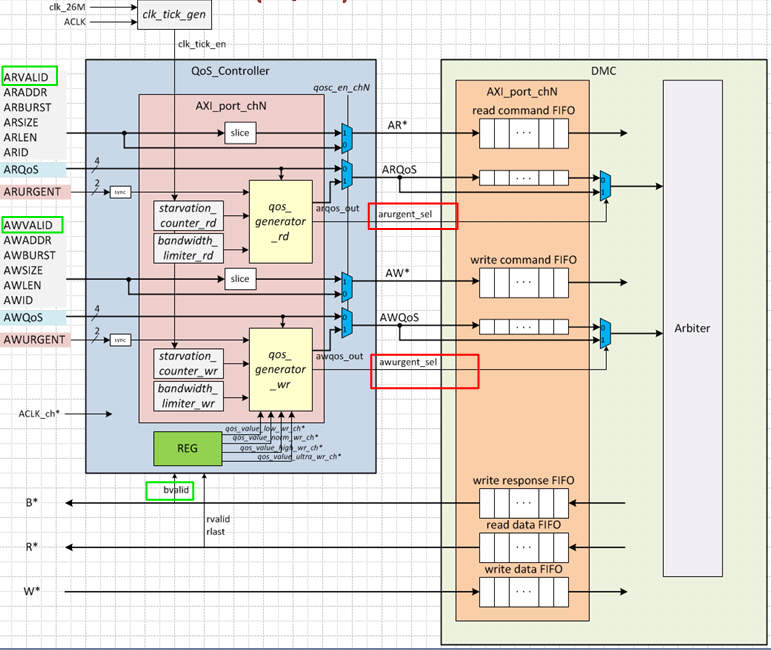
一些Master都会配置两种不同的QOS值，分为HIGH和LOW，在一般的情况下采用Low优先级，什么时候切换为HIGH则为MASTER的内部IP自己根据情况切换。

当有master的QOS超过了QOS阈值寄存器设置的value时，那么整个subsys就会发出urgent信号，该信号会直接传递给PUB当中相应的端口上，PUB在收到这个信号针对QOS会有两种选择1、QOS直接使用PUB上配置的，整个port端口的优先级都将会被抬高（一般PUB的QOS都跟master设置的QOS相同，保证不会出现优先级反转）2、继续采用QOS传递过来的QOS。两种方式的选择是根据PUB寄存器配置决定（sharkl5，其他项目不确定，默认采用PUB的QOS生效，以此来抬高Port的优先级）。

根据上图圈出的部分，每个PORT都会有类似的CMD FIFO进行指令的排队，当URGENT被拉起的时候，那么当前被拉起的访问就会绕过该FIFO直接传递给仲裁器。

#### 2.2 QOS Controller结构简介

将上面的结构图进行细化如下：



从上图可以得出以下的结论：

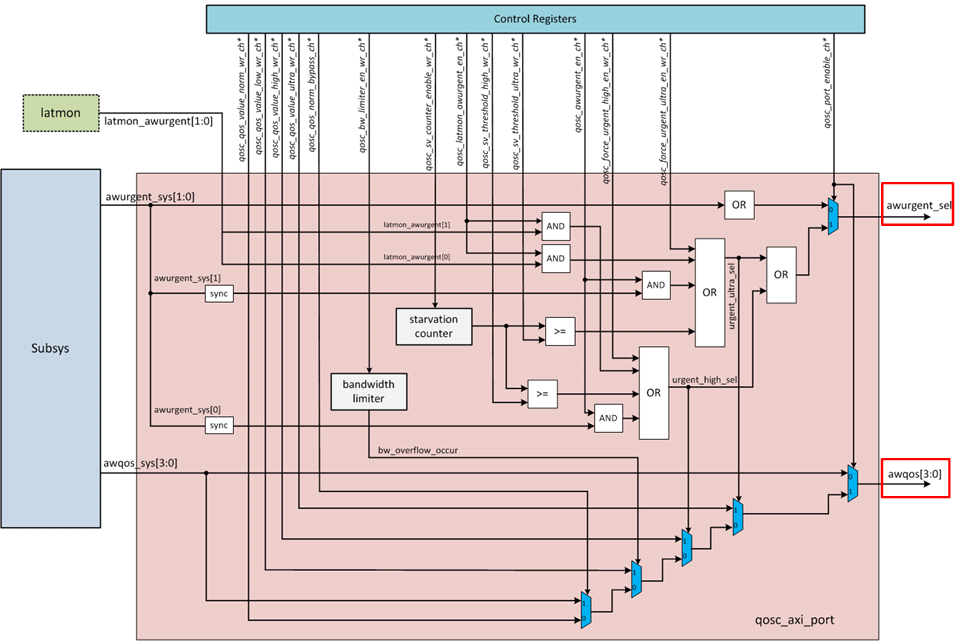
(1)urgent信号最终由qos\_generator直接发出，该结构含有多组输入，两项输出。将所有的urgent信号与QOS进行同步和汇总。

(2)QOS Controler产生urgent\_sel(awurgent\_sel/arurgent\_sel)和最终的QOS value 给DMC，以此控制FIFO的传输和当前port的最终QOS值。

(3)当urgent\_sel为0时DMC用AXI port command FIFO内的awqos/arqos 来决定通道的仲裁

(4)当urgent\_sel为0时 DMC就会绕过AXI port command FIFO ，直接改用外部的QOS数值来进行仲裁可以让该端口用在提升QOS的场景，直到urgent\_sel被解除。

对上面的qos\_generator进行细化如下:



由上图可知urgent\_sel被拉起的情况有以下的几种

(1)在打开reg\_qosc\_awurgent\_en /reg\_qosc\_arurgent\_en为1时（QOS使能的情况下）由subsys拉的awurgent或arurgent为1时。

(2) 在打开reg\_qosc\_sv\_counter\_en\_wr/reg\_qosc\_sv\_counter\_en\_wrd为1时 统计的starvation-counter超过High/ultra threshold时信号线会被拉高。

(3)在打开reg\_qosc\_latmon\_awurgent\_en/reg\_qosc\_latmon\_arurgent\_en时（latency的检测）latmon\_awurgent或latmon\_arurgent被置位的时候。

(4)软件强制提高优先级，将reg\_qosc\_force\_urgent\_high\_en=1 (软件强制拉 urgent)上面应该都是硬件的相关的操作和判断来根据情况拉高urgent\_sel，而这里是软件强制发出high urgent。

### 三. DMC对QOS的处理

经过上面的层层传递最终8个DDR PORT端口的QOS值传递给了仲裁器，仲裁器会将传递的QOS按照F~0的顺序映射成3个优先级group，分别为normal、LOW urgent和High urgent，如下图：



仲裁器有一个长度64个的Command\_q，仲裁器会按照0~11为normal、12~13为low urgent、14~15为high urgent往该Q里填充，QOS高的将优先填充。从该阶段开始QOS的值将不再使用，所有的访问都变成了3个优先级group。

Command\_q并不是所有的都对外开放，其会预留4个位置专供给Urgent的QOS填充，因为如果整个queue对外开放，那么一旦queue被填满，那么urgent的请求将无法填充，无法被相应，势必会造成overflow的现象。

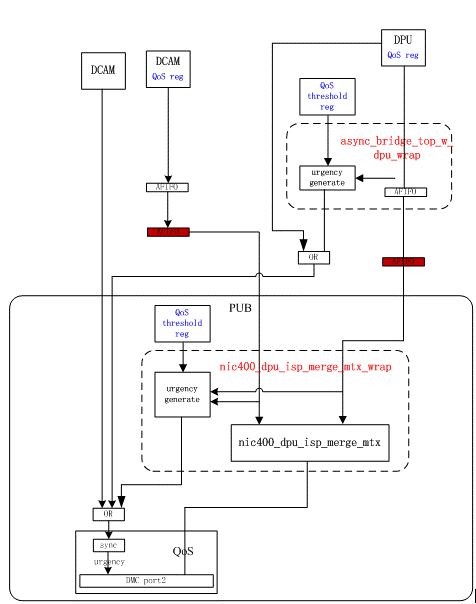
在映射完成后，此时访问DDR 的顺序会按照两点进行：1、优先级的Group 2、性能需求（如同一个CS的访问会连续进行，避免频繁切换CS，提升性能）

### 四. SharkL5的QOS结构及配置

#### 4.1DPU、DCAM

DCAM:Low:r/w:10/10

High:r:13



R/W:13/13

PUB配置

DCAM和DPU的IP能够直接拉urgent

DPU:

Low:r/w:10/6

High:r:13

R/W:12/12

R/W:12/15

DPU和DCAM属于实时模块，对DDR优先级要求较高，因此相对于其他的模块，QOS较大。

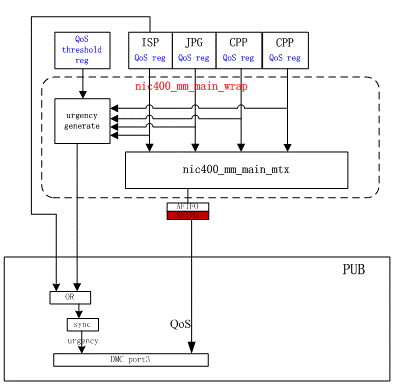
DPU：写优先级相比于读优先级可以适当降低些，不会出现overflow的现象。Low：r/w:10/6，High：r:13 阈值设置为12/15，因此只有当DPU自行将read从low切换到high状态（13>12），那么此时会同时超过Subsys和PUB上的阈值，两部分会同时发出urgent信号进行相或（效果相同）。

（关于subsys内有一个阈值比较产生机制而在PUB当中还具有一个urgent比较产生机制，请教过evan并没有一个好的结果，可能是为了设置的便利性）

DCAM：DCAM写的优先级要比读的优先级要高，Low:r/w 10/10，High:13，PUB的QOS阈值为12，当DCAM内部切换write的优先级为HIGH时，那么此时会超过pub\_sys当中的threshold，会产生urgent信号，如果选择PUB寄存器控制，那么此时的QOS由PUB的寄存器值13进行替换。

上面发现当产生urgent时，PUB替换的QOS值和master切换的HIGH QOS值相同，其设置成一致的意义在于urgent值产生后整个port的优先级都会有提升，直到urgent被关闭，期间当前port的端口将一直维持着urgent QOS值，在normal状态是每个请求的QOS与请求的指令之间是同步的。

#### 4.2MM



R/W:7/7

R/W:6/6

MM内部全是normal的模块，并没有实时性的需求，因此相应的QOS相比较real time较低，**ISP**:low :r/w:6/6, High:r/w:7/7，**JPG**:low：r/w:1/1,High:r/w7/7 ，

CPP:r/w:1/1。

其中当ISP和JPG的qos值由LOW->HIGH 时会触发urgent，提升qos的值。

#### 4.3WTLCP

R/W:14/14

R/W:12/12

R/W:13/13

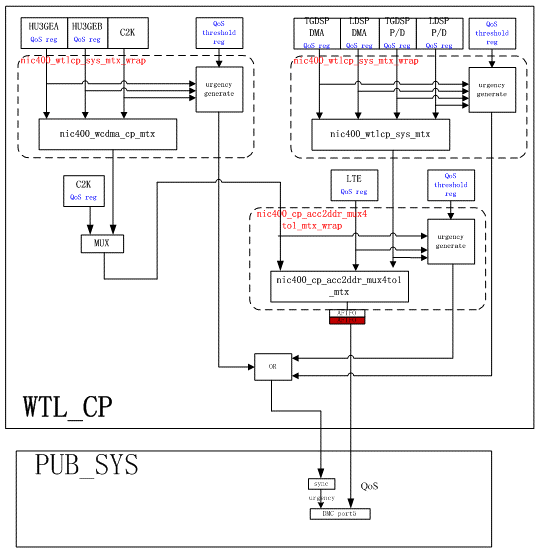
R/W:10/10

R/W:15/15

R/W:12/12

R/W:12/12

R/W:12/12



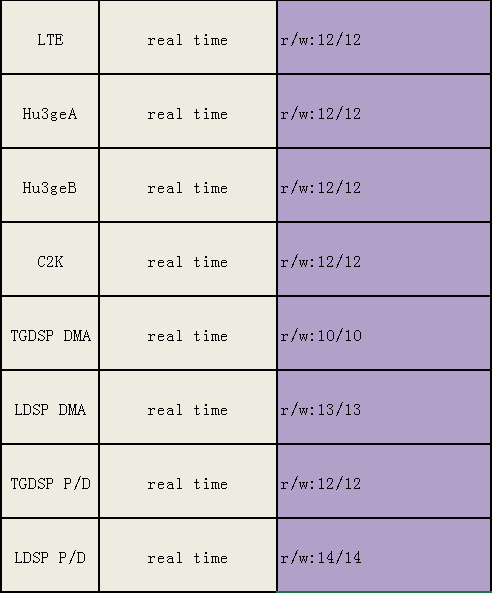
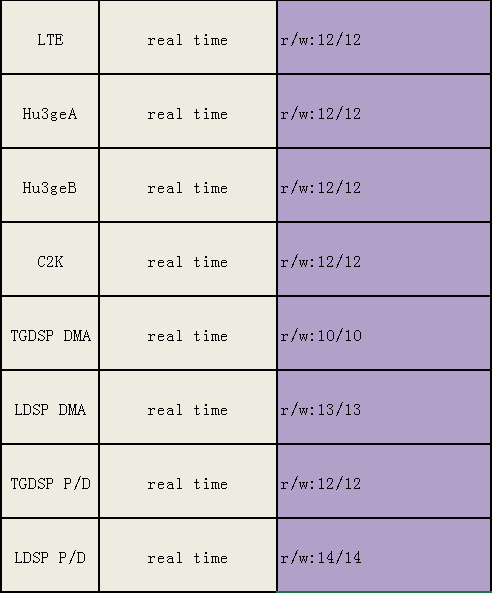
R/W:13/13

R/W:14/14

R/W:12/12

R/W:13/13

WTLCP是实时性的模块，对实时要求较高，各个Master的设置如下：



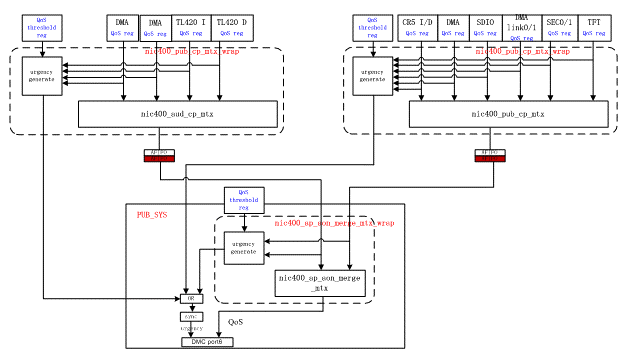
在上面的结构图的左上侧部分HU3GEA、HU3GEB、C2K部分同一设置QOS值为12，其中的subsys的compare阈值设置为15、13，那么表示当前部分是关闭了urgent触发机制。由于WTLCP基本是整个SYS当中优先级最高的模块，因此只需在该模块中选取对DDR访问要求较高的特别的master开启urgent即可

在结构图的右上侧部分的TGDSP DMA、LDSP DMA、TGDSP P/D、LDSP P/D的QOS值分别为10、13、12、14，相应的compare阈值设置为13、13，因此只有LDSP P/D开启了urgent机制，并且由于LDSP是所有模块对DDR响应要求最高的一个模块，因此该模块的QOS值设置为最高，并且当该模块一旦发出DDR的请求，就会立刻触发urgent，会由PUB上的QOS值进行替换（PUB的QOS值与subsys的值相同，不同之处在于由subsys提供的值是与请求指令相同步的，而PUB替换的qos会立刻替换到当前port端口上）提升整个port的优先级，优先访问DDR。

#### 4.4PUBCP/AUDCP

R/W:9/9

R/W:9/9



R/W:15/15

R/W:15/15

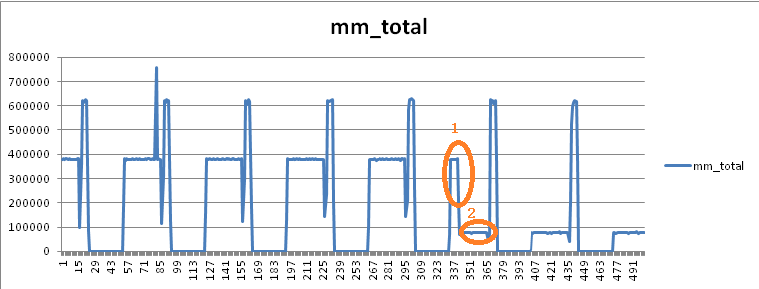
R/W:15/15

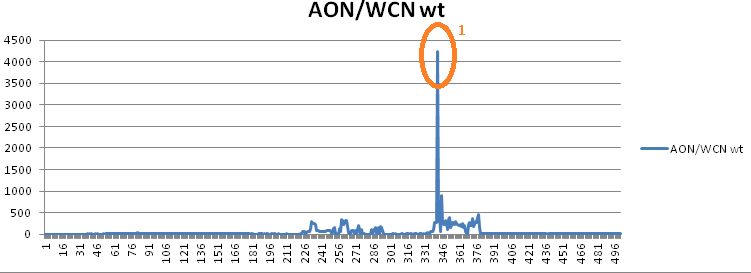
PUBCP和AUDCP的Master全部属于normal级别，并且其全部属于非实时模块对DDR访问的等待具有一定的耐性，并不会造成overflow的现象，根据以往的项目配置经验，所有master的QOS全部配置成9，并且配置copare threshold为15，关闭了urgent触发机制。

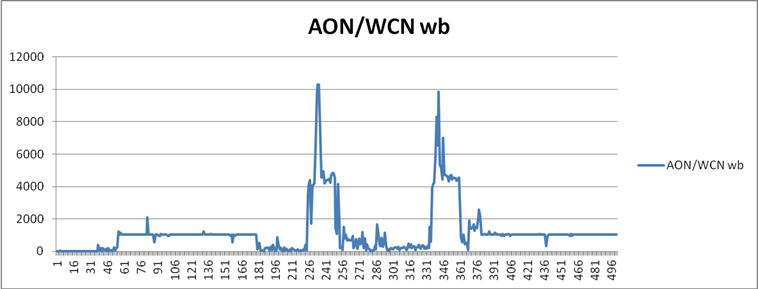
### 五. 相关问题预览

* [SharkLe]DCAM在拍照的时候出现overflow现象

在出现类似的问题的时候，能够影响的模块有限，这里利用busmonitor抓取了MM和AON/WCN访问DDR的次数，如下：







上图中的mm\_total是表示访问的数据量，从标识的1->2出发现访问的数量突然降低同一时间点（342），AON/WCN 写入数据的次数为4243次，总共写入的数据大小为9847byte,那么每次访问写入的数据为9847/4243=2.32byte，这样这对DDR的访问的效率会很低。

原因是DCAM会根据内部的IP调整QOS的值，当DCAM的QOS值从high urgent值溢出到了（降低了）low urgent，此时DCAM和AON处在一个Group当中，因此在low urgent区中大量AON/WCN sys的短burst block dcam的传输。

当将AON/WCN传输的write tr 移出low urgent区域到normal zone，DCAM溢出现象消失， 因此该现象是由AON的短burst的缘故导致，最终由AON同时找出导致短burst的原因。

短burst影响DDR效率的原因：

每次读写操作都会经过发送行地址，然后发送列地址和读写命令后经过一定的时间进行读写，这其中行指令到读写命令tRCD和读写潜伏期CL时间

  有效的读写方式单独的访问减少，Burst访问长度加大，因为burst访问当中只有第一个数据传输时需要tRCD +CL,剩余的数据是没一个周期就会传送一组数据，相同的道理大量的短burst会出现多次发送行地址+读写指令（包含列地址，两者是同时发出）因此，每次效率严重降低。