

STUE(国产化平台) - 错误 #1437

STUE FH RX 数据只能跑一次，怀疑内存地址有问题。
从fpga看，rx的描述符地址跟demo发送的地址不太一样，没有对齐。

2023-11-27 15:59 - 战 弋戈

状态:	已关闭	开始日期:	2023-11-27
优先级:	普通	计划完成日期:	
指派给:	战 弋戈	% 完成:	0%
类别:		预期时间:	0.00 小时
目标版本:		耗时:	0.00 小时

描述
现象： 对大页内存分配出来的内存池地址进行打印，发现从大页里分配出来的内存池地址没有对齐。 FPGA_FRONTHAUL cpa_sector_get_instances -----[wyw-debug]-fh_get_bar_addr: tempVirBarAddr=0xfffe77e50000 zyg:pAllocPool: 0x400000000000, nBufferPoolSize: 0x0002c000 =zyg:pRxFhRingBuffer: 0x40000000a008 ===zyg:pTxFhRingBuffer: 0x400000015008 ===zyg:pRxFhSycRingBuffer: 0x400000020008 ===zyg:pRxPrachRingBuffer: 0x40000002b008 im_accel_constructor: Descriptor Queue is created zyg:pAllocPool: 0x40000002e008, nBufferPoolSize: 0x00770000 zyg:pAllocPool: 0x4000007a0010, nBufferPoolSize: 0x00770000 zyg:pAllocPool: 0x400000f12018, nBufferPoolSize: 0x00770000 zyg:pAllocPool: 0x400001684020, nBufferPoolSize: 0x00690000 zyg:pAllocPool: 0x400001d16028, nBufferPoolSize: 0x00690000 zyg:pAllocPool: 0x4000023a8030, nBufferPoolSize: 0x00690000 zyg:pAllocPool: 0x400002a3a038, nBufferPoolSize: 0x000c0000 Success at cpa_bb_bm_init[UL SYNC]! zyg:pAllocPool: 0x400002afc040, nBufferPoolSize: 0x000c0000 Success at cpa_bb_bm_init[UL SYNC]! zyg:pAllocPool: 0x400002bbe048, nBufferPoolSize: 0x000c0000 Success at cpa_bb_bm_init[UL SYNC]! cpa_5g_sync_req: [finished] cpa_5g_sync_req: [finished] cpa_5g_sync_req: [finished] cpa_5g_fronthault_config Handle is OK cpa_5g_fronthault_config Handle is OK cpa_5g_fronthault_config Handle is OK zyg:pAllocPool: 0x400002c80050, nBufferPoolSize: 0x00009000 [fpga_cfg_set_time_reg] start phy Addr=58042c8a058 phy addr2 is 58042c8a058 nr_start_driver 94 -Driver Thread create is called ===zyg:pDesc addr: 0x400001832028, 0 ===zyg:pDesc addr: 0x400001841028, 1 ===zyg:pDesc addr: 0x400001850028, 2 ===zyg:pDesc addr: 0x40000185f028, 3 ===zyg:pDesc addr: 0x40000186e028, 4 ===zyg:pDesc addr: 0x40000187d028, 5 ===zyg:pDesc addr: 0x40000188c028, 6 ===zyg:pDesc addr: 0x40000189b028, 7 ===zyg:pDesc addr: 0x4000018aa028, 8 ===zyg:pDesc addr: 0x4000018b9028, 9 ===zyg:pDesc addr: 0x4000018c8028, 10 ===zyg:pDesc addr: 0x4000018d7028, 11 ===zyg:pDesc addr: 0x4000018e6028, 12 ===zyg:pDesc addr: 0x4000018f5028, 13

历史记录

#1 - 2023-11-27 16:00 - 战 弋戈

- 指派给 被设置为 战 弋戈

#2 - 2023-11-27 16:06 - 战 弋戈

经分析发现，内存池分配时，带有信息的头部结构体在arm体系下的大小跟x86下不一致。具体表现在CpaBufferPoolStruct，

```
typedef struct CpaBufferPool {
void pBufferPoolStart;
/*< Pointer to the starting address of the buffer Pool*/
uint32_t nBufferPoolSize;
/*< Size of the buffer Pool created by a buffer manager*/
uint32_t nBufferPoolIndex;
/*< Buffer pool id to identify the buffer pool*/
uint64_t nAllocStats:1;
/*< Allocation status like 1 for allocated and 0 for free*/
/*< deliberately increased the size to 64 bit to keep align*/
struct CpaMemorySegment pMemorySegment;
/*< Buffer pool allocated from this Memory Segment /
struct CpaBufferPool *pNext;
/*< pointer to the next buffer pool to form as a linked list*/
struct CpaBufferPool pPrevious;
/*< pointer to the previous buffer pool to form as a linked list*/
struct CpaBufferList pFreeBufferList;
/*< pointer to the free buffer list in the buffer pool*/
pthread_mutex_t sBufferPoolLock; // x86下40字节，arm下48字节。
/*< To lock the buffer pool from access by multiple threads /
uint64_t nReserved1011; //arm系统需要改成 1011 才能1024对齐
/*< reserved, inorder to maintain the header size of 64 byte boundary*/
}CpaBufferPoolStruct;
```

其中 pthread_mutex_t sBufferPoolLock; // x86下40字节，arm下48字节。若 uint64_t nReserved¹⁰¹² 还是保留1012的化，会导致整个结构体的长度在1025*8个字节。所以在arm下，需要将nReserved¹⁰¹² 修改成 nReserved¹⁰¹¹，保持结构体总长度为1024 * 8 字节。

#3 - 2023-11-27 16:08 - 战 弋戈

fpga 的地址读取跟写入是每64字节一次。这可能是导致数据不变的原因。

#4 - 2023-11-27 16:09 - 战 弋戈

经分析发现，内存池分配时，带有信息的头部结构体在arm体系下的大小跟x86下不一致。具体表现在CpaBufferPoolStruct，

```
typedef struct CpaBufferPool {
void pBufferPoolStart;
/*< Pointer to the starting address of the buffer Pool*/
uint32_t nBufferPoolSize;
/*< Size of the buffer Pool created by a buffer manager*/
uint32_t nBufferPoolIndex;
/*< Buffer pool id to identify the buffer pool*/
uint64_t nAllocStats:1;
/*< Allocation status like 1 for allocated and 0 for free*/
/*< deliberately increased the size to 64 bit to keep align*/
struct CpaMemorySegment pMemorySegment;
/*< Buffer pool allocated from this Memory Segment /
struct CpaBufferPool *pNext;
/*< pointer to the next buffer pool to form as a linked list*/
struct CpaBufferPool pPrevious;
/*< pointer to the previous buffer pool to form as a linked list*/
struct CpaBufferList pFreeBufferList;
/*< pointer to the free buffer list in the buffer pool*/
pthread_mutex_t sBufferPoolLock; // x86下40字节，arm下48字节。
/*< To lock the buffer pool from access by multiple threads /
uint64_t nReserved1011; //arm系统需要改成 1011 才能1024对齐
/*< reserved, inorder to maintain the header size of 64 byte boundary*/
}CpaBufferPoolStruct;
```

其中 pthread_mutex_t sBufferPoolLock; // x86下40字节，arm下48字节。若 uint64_t nReserved¹⁰¹² 还是保留1012的化，会导致整个结构体的长度在1025*8个字节。所以在arm下，需要将nReserved¹⁰¹² 修改成 nReserved¹⁰¹¹，保持结构体总长度为1024 * 8 字节。

战 弋戈 写到：

经分析发现，内存池分配时，带有信息的头部结构体在arm体系下的大小跟x86下不一致。具体表现在CpaBufferPoolStruct，

```
typedef struct CpaBufferPool {
void pBufferPoolStart;
```

```

/*< Pointer to the starting address of the buffer Pool*/
uint32_t nBufferPoolSize;
/**< Size of the buffer Pool created by a buffer manager*/
uint32_t nBufferPoolIndex;
/*< Buffer pool id to identify the buffer pool*/
uint64_t nAllocStats:1;
/**< Allocation status like 1 for allocated and 0 for free*/
/**< deliberately increased the size to 64 bit to keep align*/
struct CpaMemorySegment pMemorySegment;
/*< Buffer pool allocated from this Memory Segment /
struct CpaBufferPool *pNext;
/*< pointer to the next buffer pool to form as a linked list*/
struct CpaBufferPool pPrevious;
/*< pointer to the previous buffer pool to form as a linked list*/
struct CpaBufferList pFreeBufferList;
/*< pointer to the free buffer list in the buffer pool*/
pthread_mutex_t sBufferPoolLock; // x86下40字节，arm下48字节。
/**< To lock the buffer pool from access by multiple threads /
uint64_t nReserved1011; //arm系统需要改成 1011 才能1024对齐
/*< reserved, inorder to maintain the header size of 64 byte boundary*/
}CpaBufferPoolStruct;

```

其中 pthread_mutex_t sBufferPoolLock; // x86下40字节，arm下48字节。若 uint64_t nReserved¹⁰¹² 还是保留1012的化，会导致整个结构体的长度在1025*8 个字节。所以在arm下，需要将nReserved¹⁰¹² 修改成 nReserved¹⁰¹¹，保持结构体总长度为1024 * 8 字节。

这样修改对齐后问题解决。

#5 - 2023-11-27 16:11 - 战 弋戈

对齐结构体后：相应的打印如下：

```

GA_FRONTHAUL cpa_sector_get_instances
-----[wyw-debug]-fh_get_bar_addr: tempVirBarAddr=0xffffda8000000
zyg:pAllocPool: 0x400000000000, nBufferPoolSize: 0x0002c000
==zyg:pRxFhRingBuffer: 0x40000000a000
===zyg:pTxFhRingBuffer: 0x400000015000
===zyg:pRxFhSycRingBuffer: 0x400000020000
===zyg:pRxPrachRingBuffer: 0x40000002b000
im_accel_constructor: Descriptor Queue is created
zyg:pAllocPool: 0x40000002e000, nBufferPoolSize: 0x00770000
zyg:pAllocPool: 0x4000007a0000, nBufferPoolSize: 0x00770000
zyg:pAllocPool: 0x400000f12000, nBufferPoolSize: 0x00770000
zyg:pAllocPool: 0x400001684000, nBufferPoolSize: 0x00690000
zyg:pAllocPool: 0x400001d16000, nBufferPoolSize: 0x00690000
zyg:pAllocPool: 0x4000023a8000, nBufferPoolSize: 0x00690000
zyg:pAllocPool: 0x400002a3a000, nBufferPoolSize: 0x000c0000
Success at cpa_bb_bm_init[UL SYNC]!
zyg:pAllocPool: 0x400002afc000, nBufferPoolSize: 0x000c0000
Success at cpa_bb_bm_init[UL SYNC]!
zyg:pAllocPool: 0x400002bbe000, nBufferPoolSize: 0x000c0000
Success at cpa_bb_bm_init[UL SYNC]!
cpa_5g_sync_req: [finished]
cpa_5g_sync_req: [finished]
cpa_5g_sync_req: [finished]
cpa_5g_fronthault_config Handle is OK
cpa_5g_fronthault_config Handle is OK
cpa_5g_fronthault_config Handle is OK
zyg:pAllocPool: 0x400002c80000, nBufferPoolSize: 0x00009000
[fpga_cfg_set_time_reg] start
phy Addr=58042c8a000
phy addr2 is 58042c8a000
nr_start_driver 94 -Driver Thread create is called
[UE_Start_FPGA]: Init HARDWARE successfully, FPGA_Start1

```

pAllocPool 分配出来的地址是对齐的。

#6 - 2023-11-27 16:14 - 战 弋戈

```

===zyg:pDesc addr: 0x400001aa8000, 0
===zyg:pDesc addr: 0x400001ab7000, 1
===zyg:pDesc addr: 0x400001ac6000, 2
===zyg:pDesc addr: 0x400001ad5000, 3
===zyg:pDesc addr: 0x400001ae4000, 4
===zyg:pDesc addr: 0x400001af3000, 5
===zyg:pDesc addr: 0x400001b02000, 6

```

```
===zyg:pDesc addr: 0x400001b11000, 7
===zyg:pDesc addr: 0x400001b20000, 8
===zyg:pDesc addr: 0x400001b2f000, 9
===zyg:pDesc addr: 0x400001b3e000, 10
===zyg:pDesc addr: 0x400001b4d000, 11
===zyg:pDesc addr: 0x400001b5c000, 12
===zyg:pDesc addr: 0x400001b6b000, 13
```

这个地址也是4K对齐。

#7 - 2023-11-27 16:22 - 战 弋戈

- 状态 从 新建 变更为 进行中

#8 - 2023-11-27 16:34 - 战 弋戈

- 状态 从 进行中 变更为 已关闭

#9 - 2023-11-28 15:45 - 高峰

PCIe读取是按照256byte/次进行的，为了提升性能避免bytes拼接，所以要求PHY的buf起始地址要按照256byte字节进行。
buf header中的有个变量pthread_mutex_t，X86操作系统是6个u64；ARM
操作系统下是6个u64。所以导致ARM下，buffer地址不对齐，进而导致FPGA处理错误。