JingZhao HCA 软硬件接口设计文档

V1.3-b

Date	Author	Comment	Version
07/05/2020	康宁	IB HCA 软硬件接口的第一个正式版本，主要介绍了软硬件接口的内存布局，以及几个典型的控制路径数据流模式	V0.9-a
11/05/2020	康宁	该版本主要添加了如下内容： 1. 添加 BAR 空间的存储结构； 2. 添加 HCR 命令相关存储； 3. ICM 所含有的字段及各字段的含义	V1.0-a
27/06/2020	康宁	该版本主要添加了如下内容： 1. 传输数据包中各个字段的内容来源； 2. 明确了文档中一些含糊的地方；	V1.1-a
18/01/2021	康宁	修改了命令 mailbox 中一些格式错误; 删除了目前硬件未实现的功能；	V1.2-b
25/06/2021	康宁	1. 修改 PIO 模块的字节序；	V1.2-b
09/07/2021	康宁	1. 修改了 Doorbell 的格式 2. 修改了 QPC 字段格式（MODIFY_QP）	V1.2-c
30/07/2021	康宁	1. 修改了 MTT start_index 的定义； 2. 细化了 QPC 中 mtu_msgmax 的定义	V1.2-d
31/08/2021	康宁	1. 添加了 CQE 相关的字段说明； 2. 添加了 WQE 相关的字段说明	V1.3-a
13/09/2021	康宁	1. 在 QPC 中添加了 LID，MAC，IP 字段； 2. 在 WQE，UD_UNIT 中添加了 IP 字段	V1.3-b

目录

1 IB HCA 软硬件的存储布局

该软硬件接口，目前考虑的是采用 memory free 的形式，即 ICM 最终会保存在 Host Memory 中。

1.1 Host Memory 中的存储布局

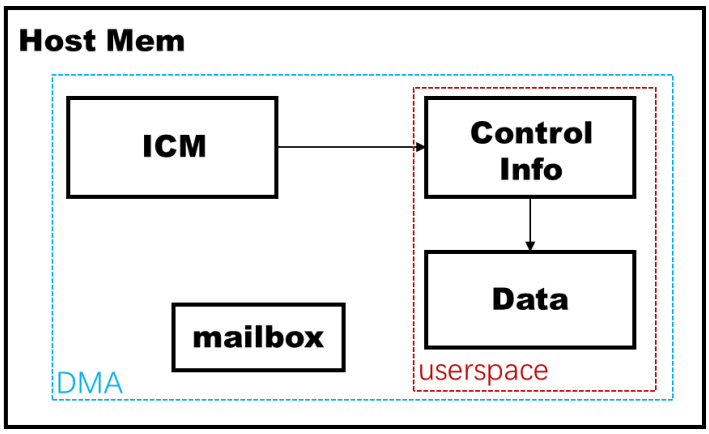

图 1-1 Host Memory 中的内存资源布局

Host Memory 中的内存资源布局如图 1-1 所示。注意，图中的各个内存区域并不一定代表一块完整的物理或虚拟内存空间，而是按照存储资源类型划分的内存区域集合。

ICM 全称为 Interconnection Context Memory，它用于保存 HCA 资源（包括 Queue Pair，Completion Queue，Event Queue，Memory Protection Table，Memory Translation Table）的上下文内容。ICM 并不是一块完整连续的内存空间，它只是软件为 HCA 分配，并由 HCA 管理的，用于保存 HCA 全部资源上下文的内存空间。ICM 中的资源在第一次使用时分配好其需要的内存空间，并将该内存空间的 DMA 地址发送给 HCA，此刻开始，这块内存空间就完全交由 HCA 来控制。而这部分内存的读（读取上下文）写（写入新的上下文条目）操作都将由 HCA 来完成。

Mailbox 空间是由内核态驱动分配的 DMA 池，这部分内存主要用于软件向 HCA 提交命令时，保存命令的输入输出参数。这一段空间的大小为 4KB。

Control Info 空间主要用于保存 QP/CQ/EQ 的描述符队列，这些信息在用户空间分配，并且被配置成可以由 HCA 通过 DMA 的方式进行访问。这块内存空间的主要目的是用于 HCA 数据传输过程中相关控制信息的保存。

Data 空间用于存储 HCA 传输的数据，该空间也是用户分配的空间，并在初始化配置阶段由用户通过内核态驱动向 HCA 对这部分内存进行了注册。

1.2 HCA 中的存储布局

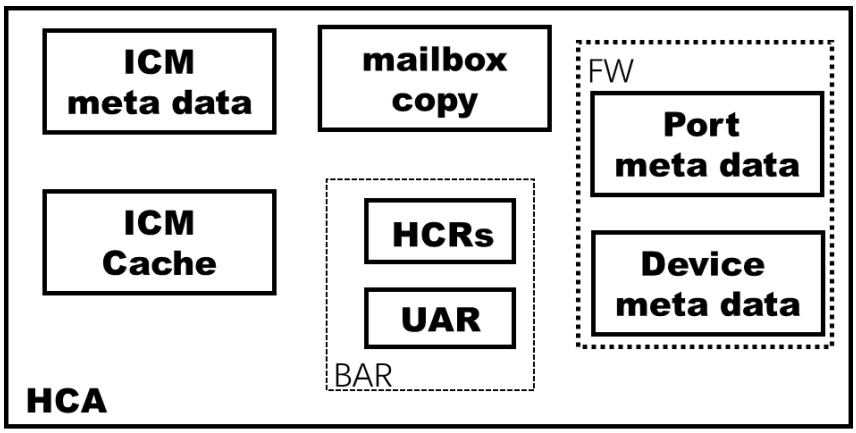

图 1-2 HCA 中的资源存储布局

ICM meta data 存储了 ICM 的一些元数据信息，包括各种资源在（HCA 虚拟）ICM 空间的偏移，占用大小，以及 ICM（HCA 虚拟）地址与 Host Memory 物理地址映射关系等信息。该部分存在的目的是当 HCA 本地存储 ICM 不够时，可以利用该部分信息，将 ICM 换出到 Host Memory 中。

ICM Cache 用于缓存 ICM 数据。当 HCA 资源在软件端被创建后，首先软件端将资源上下文信息发送给 HCA，随后，HCA 硬件将这部分信息保存在 ICM Cache 中。当 ICM Cache 保存不下时，再利用 ICM meta data 中的信息，将资源换出，保存在 Host Memory 中。

Mailbox copy 用于拷贝 Host Memory 中的输入 mailbox 或将信息发给 Host Memory 中的输出 mailbox。对于一些带有输入 mailbox 的命令来说，HCA 通过 HCR 寄存器接收到命令后，首先采用 DMA 的方式，从 Host Memory 中将 mailbox 中的数据复制到 HCA 上来，随后，再根据命令类型及 mailbox 中的信息进行后续处理操作。对于带有输出 mailbox 的命令来说，CEU 首先根据命令，将信息写入 mailbox copy 区域，再通过 DMA 的方式，将该区域的信息拷贝到 Host Memory 中的 mailbox 区域，从而完成 HCA 命令参数的输出。

BAR 空间是软件可以访问的 HCA 内存空间，注意 BAR 空间在 HCA 内部并不是一块连续的存储空间，而是一块能够被寻址的物理逻辑。BAR 空间包括 HCRs（HCA Configuration Registers）和 UAR（User Access Region）。对软件端来说，HCRs 空间只能由内核态空间来访问，用于软件端向 HCA 提交 HCR 命令，对 HCA 进行复位。而 UAR 空间由用户态空间访问，用于软件端向 HCA 提交 Doorbell，以及进行进程间隔离。Doorbell 功能比较简单，用户态进程通过向 Doorbell 写内容，告知 HCA QP 请求的提交，随后 HCA 根据 Doorbell 中各字段的内容进行请求的处理。此处详细对 UAR 的进程隔离功能进行说明。在用户态空间执行 ibv_open_device 函数时，会为该进程新分配一个用户上下文，该上下文中保存着一个 UAR 页面的虚拟地址，通过该页面，用户可以直接向 HCA 提交 Doorbell 内容，而不必进入内核态，从而实现了内核旁路；同时，由于不同进程被分配了不同的 UAR 页面，在 HCA 端就可以有效地对不同进程间的 QP 进行隔离。

FW 空间用于保存固件信息，设备信息，端口信息以及 CEU 的代码等内容，这些内容都是从固件 ROM 中读取而来的。固件信息包括设备版本，固件版本等信息；设备信息包括设备资源限制，对设备中各种资源的管理配置信息（如 PCIe 功耗管理等），以及 RDMA 引擎中对各个字段的配置信息；端口信息包括端口 LID，子网管理器所在 LID，物理状态，子网管理服务等级，最大 MTU，在用 MTU 等有关端口的信息。

3 软硬件接口相关数据结构

3.1 BAR 空间数据结构

本节将讨论 BAR 空间的数据结构。在该版本的 HCA 实现中，仅考虑 memory free（Arbel）的实现方法，因此，BAR 空间中仅包含 BAR0 和 BAR2 空间。HCA 的 BAR 空间分配如表 3-1-1 所示。

表 3-1-1 HCA 的 BAR 空间分配

BAR 0-1	BAR 2-3
1MB	8MB
HCR	UAR

3.1.1 BAR0 空间数据结构

BAR0 空间主要保存一些内核态使用的寄存器字段，包括 HCR 寄存器，复位寄存器，EQ Doorbell 等。相关字段信息如表 3-1-2 所示。

表 3-1-2 HCA BAR0 空间相关寄存器

Offset	Size(Byte)	Name	Description
0xf0010	4	Reset	值为 1：启动复位；值为 0：不启动复位。默认为 0
0x80000	28	HCR	用于向 HCA 提交命令，各字段含义如表 3-1-3 所示

表 3-1-3 HCR 寄存器结构

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	in_param[63:32]																															
04h	in_param[31:0]																															
08h	in_modifier																															
0Ch	out_param[63:32]																															
10h	out_param[31:0]																															
14h	token																															
18h	status								go	E			op_modifier												op							

表 3-1-4 HCR 寄存器字段描述

Offset	Bits	Name	Description	Access
00h	64	in_param	输入参数或系统用于存放输入参数的地址（详见 3.2 节）	WR
0Ch	64	out_param	输出参数或用于存放输出参数的地址（详见 3.2 节）	RD/WR
08h	31:0	in_modifier	in_param 的修饰符（详见 3.2 节）	WR
14h	31:16	token	系统分配给命令的唯一令牌符	WR
18h	31:24	status	命令执行的状态报告，有效的状态见表 3-1-5	RD
	23:23	go	由软件置 1，表明 HCR 处于 busy，不可写该寄存器；为 0 时表示可以向 HCR 寄存器写入命令	RD/WR
	22:22	E	event，为 1 时，命令执行完成后向 EQ 报告；为 0 时不报告	WR

	19:12	op_modifier	操作码修饰符（详见 3.2 节）	WR
	11:0	op	操作码，每条命令都有自己的操作码（详见 3.2 节）	WR

表 3-1-5 HCR 寄存器 status 字段有效状态

State	Bits	Description
HGHCA_CMD_STAT_OK	8'h00	HCR 命令执行成功
HGHCA_CMD_STAT_BAD_OP	8'h02	HCR 命令操作码错误
HGHCA_CMD_STAT_BAD_PARAM	8'h03	HCR 命令参数错误
HGHCA_CMD_STAT_BAD_SYS_STATE	8'h04	CEU 处于复位状态
HGHCA_CMD_STAT_BAD_NVMEM	8'h05	HCA 固件内容损坏

3.1.2 BAR2 空间数据结构

BAR2 空间主要用于用户态进程对 HCA 的直接访问，HCA 为用户态的每一个进程分配一个 UAR 页面，页面中包含有 QP Doorbell，CQ Doorbell 等内容。在该版本的 HCA 实现中， BAR2 空间的大小为 8MB，允许最多 2K 个用户进程使用该 HCA。BAR2 空间中一个页面的存储布局如表 3-1-6 所示。

表 3-1-6 UAR 页面内容布局

Register	offset	+3								+2								+1								+0							
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
SQ Doorbell	00h	sq_head																										F	opcode				
	04h	qpn																								size0							

表 3-1-7 UAR 页面各字段含义

Offset	Bits	Name	Description	Access
00h	23:8	sq_head	指向 SQ 中要处理的第一个 WQE	WR
	5:5	F	是否使用 FENCE 机制，即发送队列中断 WQE 是否要保序	WR
	4:0	opcode	操作码修饰符 0x0a: HGHCA_OPCODE_SEND 0x0b: HGHCA_OPCODE_SEND_IMM 0x08: HGHCA_OPCODE_RDMA_WRITE 0x09: HGHCA_OPCODE_RDMA_WRITE_IMM 0x10: HGHCA_OPCODE_RDMA_READ	WR
04h	31:8	qpn	QP 号	WR
	7:0	size0	发送的第一个 WQE 的大小，以 16 字节为单位	WR

3.2 HCR 命令相关数据结构

HCR 命令主要是通过 HCR 寄存器，向 HCA 发送一些命令，命令格式如表 3-1-3 所示，其中，输入参数（in_param）和输出参数（out_param）可能是一个可以通过 DMA 访问的 Host Memory 地址，从而使 HCR 命令可以承载更多的输入输出参数。HCR 命令列表如表 3-2-1 所示。

表 3-2-1 HCR 命令列表

Group	Opcode	Command Name	Link to Command Description
Device Config	0x3	CMD_QUERY_DEV_LIM	参见 3.2.1.1 hghca_QUERY_DEV_LIM
	0x6	CMD_QUERY_ADAPTER	参见 3.2.1.2 hghca_QUERY_ADAPTER
	0x7	CMD_INIT_HCA	参见 3.2.1.3 hghca_INIT_HCA
	0x8	CMD_CLOSE_HCA	参见 3.2.1.4 hghca_CLOSE_HCA
ICM	0xffa	CMD_MAP_ICM	参见 3.2.2.1 hghca_MAP_ICM
	0xff9	CMD_UNMAP_ICM	参见 3.2.2.2 hghca_UNMAP_ICM
TPT	0xd	CMD_SW2HW_MPT	参见 3.2.3.1 hghca_SW2HW_MPT
	0xf	CMD_HW2SW_MPT	参见 3.2.3.2 hghca_HW2SW_MPT
	0x11	CMD_WRITE_MTT	参见 3.2.3.3 hghca_WRITE_MTT
EQ	0x12	CMD_MAP_EQ	参见 3.2.4.1 hghca_MAP_EQ
	0x13	CMD_SW2HW_EQ	参见 3.2.4.2 hghca_SW2HW_EQ
	0x14	CMD_HW2SW_EQ	参见 3.2.4.3 hghca_HW2SW_EQ
CQ	0x16	CMD_SW2HW_CQ	参见 3.2.5.1 hghca_SW2HW_CQ
	0x17	CMD_HW2SW_CQ	参见 3.2.5.2 hghca_HW2SW_CQ
	0x2c	CMD_RESIZE_CQ	参见 3.2.5.3 hghca_RESIZE_CQ
QP	0x19	CMD_RST2INIT_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x1a	CMD_INIT2RTR_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x1b	CMD_RTR2RTS_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x1c	CMD_RTS2RTS_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x1d	CMD_SQERR2RTS_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x1e	CMD_2ERR_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x1f	CMD_RTS2SQD_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x38	CMD_SQD2SQD_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x20	CMD_SQD2RTS_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x21	CMD_ERR2RST_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x2d	CMD_INIT2INIT_QPEE	参见 3.2.6.1 hghca_MODIFY_QP
	0x22	CMD_QUERY_QP	参见 3.2.6.2 hghca_QUERY_QP
	0x23	CMD_CONF_SPECIAL_QP	参见 3.2.6.3 hghca_CONF_SPECIAL_QP
	0x24	CMD_MAD_IFC	参见 3.2.6.4 hghca_MAD_IFC
Others	0x31	CMD_NOP	参见 3.2.7.1 hghca_NOP

3.2.1 设备配置相关命令

3.2.1.1 hghca_QUERY_DEV_LIM

该命令用于查询 HCA 设备各种参数的限制，该命令向 HCR 寄存器写入的值如表 3-2-2 所示。

表 3-2-2 QUERY_DEV_LIM 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0
out_param	mailbox->dma，mailbox 的总线地址
in_modifier	0
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY

go	1
E	0
op_modifier	0
op	CMD_QUERY_DEV_LIM =0x3

QUERY_DEV_LIM 命令只有 out_param，没有 in_param。out_param 存储了 mailbox 的 DMA 物理地址，其参数内容如表 3-2-3 所示。

表 3-2-3 QUERY_DEV_LIM 命令 out_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	reserved_qp								reserved_cq								reserved_eq								reserved_mtt							
04h									reserved_pd																reserved_lkey							
08h	max_qp_sz																max_cq_sz															
0Ch	max_qps								max_cqs								max_eqs								max_mpts							
10h	max_pds																max_gids								max_pkeys							
14h									max_mtt_seg																							
18h	qpc_entry_sz																cqc_entry_sz															
1Ch	eqc_entry_sz																mpt_entry_sz															
20h					ack_delay				max_mtu				max_port_width												max_vl				num_ports			
24h									min_page_sz																							
28h									max_sg								max_desc_sz															
2Ch									max_sg_rq								max_desc_sz_rq															
30h	max_icm_sz[63:32]																															
34h	max_icm_sz[31:0]																															
38h																																
3Ch																																

表 3-2-4 QUERY_DEV_LIM 命令 out_param 各字段含义

Offset	Bits	Name	Description	Access
00h	31:24	reserved_qp	预留 QP 的个数的对数（低 4 位有效）	RD
	23:16	reserved_cq	预留 CQ 的个数的对数（低 4 位有效）	RD
	15:8	reserved_eq	预留 EQ 的个数的对数（低 4 位有效）	RD
	7:0	reserved_mtt	预留 MTT 的个数的对数（低 4 位有效）	RD
04h	23:16	reserved_pd	预留 PD 的个数的对数（低 4 位有效）	RD
	7:0	reserved_lkey	预留的 lkey 的个数的对数（低 4 位有效）	RD
08h	31:16	max_qp_sz	WQE 个数的最大值的对数	RD
	15:0	max_cq_sz	一个 CQ 中 CQE 个数的最大值的对数	RD
0Ch	31:24	max_qps	QP 个数的最大值的对数（低 5 位有效）	RD
	23:16	max_cqs	CQ 个数的最大值的对数（低 5 位有效）	RD
	15:8	max_eqs	EQ 个数的最大值的对数（低 3 位有效）	RD
	7:0	max_mpts	MTT 个数的最大值的对数（低 6 位有效）	RD
10h	31:24	max_pds	PD 个数的最大值的对数（低 6 位有效）	RD
	15:8	max_gids	gid 个数的最大值的对数（低 4 位有效）	RD
	7:0	max_pkeys	pkey 个数的最大值的对数（低 4 位有效）	RD

14h	23:16	max_mtt_seg	MTT 一个 seg 的条目大小，以字节为单位	RD
18h	31:16	qpc_entry_sz	QPC 条目大小，以字节为单位	RD
	15:0	cqc_entry_sz	CQC 条目大小，以字节为单位	RD
1Ch	31:16	eqc_entry_sz	EQC 条目大小，以字节为单位	RD
	15:0	mpt_entry_sz	MPT 条目大小，以字节为单位	RD
20h	27:24	ack_delay	本地 ack 响应延迟（目前没用）	RD
	23:20	max_mtu	HCA 支持的最大 MTU `enum ib_mtu { IB_MTU_256 = 1, IB_MTU_512 = 2, IB_MTU_1024 = 3, IB_MTU_2048 = 4, IB_MTU_4096 = 5 };`	RD
	19:16	max_port_width	HCA 支持的最大端口带宽	RD
	7:4	max_vl	HCA 支持的最大 VL 个数	RD
	3:0	num_ports	HCA 支持的端口个数	RD
24h	23:16	min_page_sz	HCA 支持的最小页大小的对数	RD
28h	23:16	max_sg	每个 WQE 中 SG List 支持的最大条目个数	RD
	15:0	max_desc_sz	支持的描述符最大的字节大小（一个描述符）	RD
2Ch	23:16	max_sg_rq	RQ 中每个 WQE 中 SG List 支持的最大条目个数	RD
	15:0	max_desc_sz_rq	RQ 支持的描述符最大的字节大小	RD
30h	31:0	max_icm_sz[63:32]	支持的 ICM 大小的最大值（高 32 位）	RD
34h	31:0	max_icm_sz[31:0]	支持的 ICM 大小的最大值（低 32 位）	RD

3.2.1.2 hghca_QUERY_ADAPTER

该命令用于查询板子相关的信息，如板卡 ID 等。该命令向 HCR 寄存器写入的值如表 3-2-5 所示。

表 3-2-5 QUERY_ADAPTER 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0
out_param	mailbox->dma，mailbox 的总线地址
in_modifier	0
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_QUERY_ADAPTER =0x6

该命令只有输 out_param 没有 in_param。out_param 存储了 mailbox 的 DMA 物理地址，其参数内容如表 3-2-6 所示。

表 3-2-6 QUERY_ADAPTER 命令 out_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

00h	
04h	
08h	
0Ch	
10h	
14h	
18h	board_id[63:32]
1Ch	board_id[31:0]

表 3-2-7 QUERY_ADAPTER 命令 out_param 各字段含义

Offset	Bits	Name	Description	Access
18h	31:0	board_id[63:32]	板卡 ID 信息（高 32 位）	RD
	31:0	board_id[31:0]	板卡 ID 信息（低 32 位）	RD

3.2.1.3 hghca_INIT_HCA

该命令用于将 QPC/CQC/EQC/TPT 等资源的属性（基地址，条目个数）填写到 HCA 中。该命令向 HCR 寄存器写入的值如表 3-2-8 所示。

表 3-2-8 INIT_HCA 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	0
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_INIT_HCA =0x7

该命令只有输 in_param 没有 out_param。in_param 存储了 mailbox 的 DMA 物理地址，其参数内容如表 3-2-9 所示。

表 3-2-9 INIT_HCA 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h																																
04h																																
08h	qpc_base[63:32]																															
0Ch	qpc_base[31:8]																								log_num_qps							
10h	cqc_base[63:32]																															
14h	cqc_base[31:8]																								log_num_cqs							
18h	eqc_base[63:32]																															
1Ch	eqc_base[31:8]																								log_num_eqs							
20h																																
24h																																

28h		
2Ch		
30h	mpt_base[63:32]	
34h	mpt_base[31:8]	log_mpt_sz
38h	mtt_base[63:32]	
3Ch	mtt_base[31:0]	

表 3-2-10 INIT_HCA 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
08h	31:0	qpc_base[63:32]	QP 上下文基地址（高 32 位）	WR
0Ch	31:8	qpc_base[31:8]	QP 上下文基地址的[31:8]位，低 8 位为 0	WR
	7:0	log_num_qps	QP 个数，取以 2 为底的对数	WR
10h	31:0	cqc_base[63:32]	CP 上下文基地址（高 32 位）	WR
14h	31:8	cqc_base[31:8]	CP 上下文基地址的[31:8]位，低 8 位为 0	WR
	7:0	log_num_cqs	CP 个数，取以 2 为底的对数	WR
18h	31:0	eqc_base[63:32]	EQ 上下文基地址（高 32 位）	WR
1Ch	31:8	eqc_base[31:8]	EQ 上下文基地址的[31:8]位，低 8 位为 0	WR
	7:0	log_num_eqs	EQ 个数，取以 2 为底的对数	WR
30h	31:0	mpt_base[63:32]	MPT 表基地址（高 32 位）	WR
34h	31:8	mpt_base[31:0]	MPT 表基地址的[31:8]位，低 8 位为 0	WR
	7:0	log_mpt_sz	MPT 表条目数，取以 2 为底的对数	WR
38h	31:0	mtt_base[63:32]	MTT 表基地址的高 32 位	WR
3Ch	31:0	mtt_base[31:0]	MTT 表基地址的低 32 位	WR

3.2.1.4 hghca_CLOSE_HCA

该命令用于销毁之前存储在 HCA 中的 ICM 资源元数据。该命令向 HCR 寄存器写入的值如表 3-2-11 所示。

表 3-2-11 CLOSE_HCA 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0
out_param	0
in_modifier	0
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_CLOSE_HCA =0x8

3.2.2 ICM 相关命令

3.2.2.1 hghca_MAP_ICM

该命令用于将分配好的用于存储固件的 ICM 地址空间的每个 chunk 的物理地址及虚拟

地址发送给 HCA，该命令向 HCR 寄存器写入的值如表 3-2-12 所示。

表 3-2-12 MAP_ICM 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	nent，entry 的个数
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	1: QPC，CQC，EQC 2: MPT，MTT
op	CMD_MAP_ICM =0xffa

该命令只有输 in_param 没有 out_param。in_param 存储了 mailbox 的 DMA 物理地址，其参数内容如表 3-2-13 所示，注意，该输入参数中以一个虚实地址组合（128-bit）为一个 entry，而以两个 entry 为一组（256-bit），每组中高字节存储小编号，低字节存储大编号，若 entry 个数为奇数，则先在高字节存储，低字节空位；in_modifier 指示了 DMA 地址中所拥有的 ICM 映射地址 chunk 的个数。op_modifier 指示了该 ICM 空间所存储的资源类型，当为 1，表示存储类型为上下文类型，即 QPC、CQC、EQC；当为 2 时，表示存储类型为 TPT 类型，即 MPT、MTT。

表 3-2-13 MAP_ICM 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	virt[i+1][63:32]																															
04h	virt[i+1][31:0]																															
08h	page[i+1][63:32]																															
0Ch	page[i+1][31:12]																page_num[i+1]															
10h	virt[i][63:32]																															
14h	virt[i][31:0]																															
18h	page[i][63:32]																															
1Ch	page[i][31:12]																page_num[i]															

表 3-2-14 MAP_ICM 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
00h+i*16	31:0	virt[i][63:32]	chunk[i]的虚拟首地址（高 32 位）	WR
04h+i*16	31:0	virt[i][31:0]	chunk[i]的虚拟首地址（低 32 位）	WR
08h+i*16	31:0	page[i][63:32]	chunk[i]的页的首地址（高 32 位）	WR
0Ch+i*16	31:12	page[i][31:12]	chunk[i]的页的首地址的[31:12]位，低 12 位为 0	WR
	11:0	page_num[i]	该 chunk[i]的大小，以页为单位	WR

3.2.2.2 hghca_UNMAP_ICM

该命令用于将映射好的 ICM 虚拟地址发送给 HCA，用于解映射该内存空间。该命令向 HCR 寄存器写入的值如表 3-2-15 所示。op_modifier 指示了该 ICM 空间所存储的资源类型，

当为 1，表示存储类型为上下文类型，即 QPC、CQC、EQC；当为 2 时，表示存储类型为 TPT 类型，即 MPT、MTT。

表 3-2-15 UNMAP_ICM 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	virt，解映射的虚拟地址
out_param	0
in_modifier	page_count，解映射的页数
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	1: QPC，CQC，EQC 2: MPT，MTT
op	CMD_UNMAP_ICM =0xff9

3.2.3 TPT 相关命令

3.2.3.1 hghca_SW2HW_MPT

该命令用于将 MPT 表的一个表项内容发送给 HCA。该命令向 HCR 寄存器写入的值如表 3-2-16 所示。

表 3-2-16 SW2HW_MPT 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	mpt_index，MPT 表内偏移（以一个 MPT entry 为单位）
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_SW2HW_MPT =0xd

该命令只有 in_param，没有 out_param。其中，in_param 用于存储 mailbox 的 DMA 物理地址，该地址下具体参数内容见表 3-2-17；in_modifier 指示了该 MPT 条目的表内偏移（以一个 MPT entry 为单位）。

表 3-2-17 SW2HW_MPT 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	flags																															
04h	page_size																															
08h	key																															
0Ch	pd																															
10h	start[63:32]																															
14h	start[31:0]																															
18h	length[63:32]																															

1Ch	length[31:0]
20h	lkey
24h	window_count
28h	window_count_limit
2Ch	mtt_seg[63:32]
30h	mtt_seg[31:0]
34h	mtt_sz
38h	
3Ch	

表 3-2-18 SW2HW_MPT 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
00h	31:0	flags	MPT 表项属性标志位和访问权限标志位，详见表 3-2-19	WR
04h	31:0	page_size	该 MPT 条目指向的页大小	WR
08h	31:0	key	Memory Key，与 MPT 最大条目数相与就是该 MPT 条目表内偏移	WR
0Ch	31:0	pd	该 MPT 表项所在的保护域号	WR
10h	31:0	start[63:32]	该 MPT 表项起始的虚拟地址（高 32 位）	WR
14h	31:0	start[31:0]	该 MPT 表项起始的虚拟地址（低 32 位）	WR
18h	31:0	length[63:32]	该 MPT 表项保护的地址空间大小（高 32 位）	WR
1Ch	31:0	length[31:0]	该 MPT 表项保护的地址空间大小（低 32 位）	WR
20h	31:0	lkey	reserved（目前没用）	WR
24h	31:0	window_count	reserved（目前没用）	WR
28h	31:0	window_count_limit	reserved（目前没用）	WR
2Ch	31:0	mtt_seg[63:32]	该 MPT 表指向的 MTT 表偏移（高 32 位）	WR
30h	31:0	mtt_seg[31:0]	该 MPT 表指向的 MTT 表偏移（低 32 位）	
34h	31:0	mtt_sz	reserved（目前没用）	WR

表 3-2-19 flags 字段各 bit 位的含义

Flag Type	Bit	Description
MPT 表项属性标志位	0xfUL << 28	HGHCA_MPT_FLAG_SW_OWNS
	1 << 17	HGHCA_MPT_FLAG_MIO
	1 << 15	HGHCA_MPT_FLAG_BIND_ENABLE
	1 << 9	HGHCA_MPT_FLAG_PHYSICAL
	1 << 8	HGHCA_MPT_FLAG_REGION
MPT 表项访问权限标志位	1 << 0	IBV_ACCESS_LOCAL_WRITE
	1 << 1	IBV_ACCESS_REMOTE_WRITE
	1 << 2	IBV_ACCESS_REMOTE_READ
	1 << 3	IBV_ACCESS_REMOTE_ATOMIC
	1 << 4	IBV_ACCESS_MW_BIND
	1 << 5	IBV_ACCESS_ZERO_BASED
	1 << 6	IBV_ACCESS_ON_DEMAND

3.2.3.2 hghca_HW2SW_MPT

该命令用于将 MPT 表的一个表项内容注销掉。该命令向 HCR 寄存器写入的值如表 3-

2-20 所示。

表 3-2-20 HW2SW_MPT 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0
out_param	0
in_modifier	mpt_index，mpt 表项偏移，以一个 MPT entry 为单位
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_HW2SW_MPT =0xf

3.2.3.3 hghca_WRITE_MTT

该命令用于将 MTT 表项条目发送给 HCA。该命令向 HCR 寄存器写入的值如表 3-2-21 所示。

表 3-2-21 WRITE_MTT 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	num_mtt，输入的 MTT 条目个数
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_WRITE_MTT =0x11

该命令只有 in_param，没有 out_param。其中，in_param 用于存储 mailbox 的 DMA 物理地址，该地址下具体参数内容见表 3-2-22，注意，输入参数中以一个物理地址（64-bit）为一个 entry，以四个 entry 为一组（256-bit），每组中高字节存储小编号，低字节存储大编号，若 entry 个数不是 4 的倍数，则先在高字节存储，低字节空位；in_modifier 表示输入的 MTT 条目的个数。

表 3-2-22 WRITE_MTT 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h																																
04h																																
08h																																
0Ch																																
10h																																
14h																																
18h	mtt_start_index[63:32]																															
1Ch	mtt_start_index[31:0]																															

20h+(i+0)*8	mtt_phy_addr[i+3][63:32]
24h+(i+0)*8	mtt_phy_addr[i+3][31:0]
20h+(i+1)*8	mtt_phy_addr[i+2][63:32]
24h+(i+1)*8	mtt_phy_addr[i+2][31:0]
20h+(i+2)*8	mtt_phy_addr[i+1][63:32]
24h+(i+2)*8	mtt_phy_addr[i+1][31:0]
20h+(i+3)*8	mtt_phy_addr[i][63:32]
24h+(i+3)*8	mtt_phy_addr[i][31:0]

表 3-2-23 WRITE_MTT 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
18h	31:0	mtt_start_index[63:32]	要写入 MTT 表的起始 MTT index（高 32 位）	WR
1Ch	31:0	mtt_start_index[31:0]	要写入 MTT 表的起始 MTT index（低 32 位）	WR
20h+i*8	31:0	mtt_phy_addr[63:32]	物理地址数组，每个地址代表一页（高 32 位）	WR
24h+i*8	31:0	mtt_phy_addr[31:0]	物理地址数组，每个地址代表一页（低 32 位）	WR

3.2.4 EQ 相关命令

3.2.4.1 hghca_MAP_EQ

该命令用于设置或清除 EQ 的 event_mask。该命令向 HCR 寄存器写入的值如表 3-2-24 所示。

表 3-2-24 MAP_EQ 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	event_mask，标志要启用的事件类型，各类型内容见表 3-2-25 HGHCA_ASYNC_EVENT_MASK
out_param	0
in_modifier	((unmap<<31) | eqn);unmap 为设置(0)或清除(1)标志，eqn 为所属的 EQ 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_MAP_EQ =0x12

表 3-2-25 event_mask 中各个 bit 位的含义

Bit	Description
1ULL << 0x02	HGHCA_EVENT_TYPE_COMM_EST
1ULL << 0x03	HGHCA_EVENT_TYPE_SQ_DRAINED
1ULL << 0x04	HGHCA_EVENT_TYPE_CQ_ERROR
1ULL << 0x05	HGHCA_EVENT_TYPE_WQ_CATAS_ERROR
1ULL << 0x10	HGHCA_EVENT_TYPE_WQ_INVAL_REQ_ERROR
1ULL << 0x11	HGHCA_EVENT_TYPE_WQ_ACCESS_ERROR
1ULL << 0x08	HGHCA_EVENT_TYPE_LOCAL_CATAS_ERROR
1ULL << 0x09	HGHCA_EVENT_TYPE_PORT_CHANGE

3.2.4.2 hghca_SW2HW_EQ

该命令用于将 EQ 上下文条目写入 EQ 表中。该命令向 HCR 寄存器写入的值如表 3-2-26 所示。

表 3-2-26 SW2HW_EQ 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	eq_num，EQ 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_SW2HW_EQ =0x13

该命令只有 in_param，没有 out_param。其中，in_param 用于存储 mailbox 的 DMA 物理地址，该地址下具体参数内容见表 3-2-27；in_modifier 表示输入的 EQC 的 EQ 号。

表 3-2-27 SW2HW_EQ 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	flags																															
04h	start[63:32]																															
08h	start[31:0]																															
0Ch	logsize																															
10h																																
14h																									intr							
18h	pd																															
1Ch	lkey																															
20h																																
24h																																
28h	consumer_index																															
2Ch	producer_index																															
30h																																
34h																																
38h																																
3Ch																																

表 3-2-28 SW2HW_EQ 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
00h	31:0	flags	EQ 的相关配置属性，详见表 3-2-29	WR
04h	31:0	start[63:32]	EQ 队列的起始虚拟地址（高 32 位）	WR
08h	31:0	start[31:0]	EQ 队列的起始地址（低 32 位）	WR
0Ch	31:24	logsize	EQ 队列可存放 EQE 个数，以 2 为底数	WR
14h	7:0	intr	中断向量	WR
18h	31:0	pd	EQ 所在的保护域的 PD 号	WR

1Ch	31:0	lkey	EQ 队列的 lkey	WR
28h	31:0	consumer_index	消费者指针	WR
2Ch	31:0	producer_index	生产者指针	WR

表 3-2-29 SW2HW_EQ 命令中 flag 字段各 bit 含义

Bit	Description
0 << 28	HGHCA_EQ_STATUS_OK
1 << 24	HGHCA_EQ_OWNER_HW
1 << 18	HGHCA_EQ_FLAG_TR
1 << 8	HGHCA_EQ_STATE_ARMED

3.2.4.3 hghca_HW2SW_EQ

该命令用于将位于 HCA 中的 EQ 上下文的内容无效。该命令向 HCR 寄存器写入的值如表 3-2-30 所示。

表 3-2-30 HW2SW_EQ 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0
out_param	0
in_modifier	eq_num，要释放 EQ 的 EQ 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_HW2SW_EQ =0x14

3.2.5 CQ 相关命令

3.2.5.1 hghca_SW2HW_CQ

该命令用于将 CQ 上下文条目写入 CQ 表中。该命令向 HCR 寄存器写入的值如表 3-2-31 所示。

表 3-2-31 SW2HW_CQ 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	cq_num，CQ 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_SW2HW_CQ =0x16

该命令只有 in_param，没有 out_param。其中，in_param 用于存储 mailbox 的 DMA 物理地址，该地址下具体参数内容见表 3-2-32；in_modifier 表示输入的 CQC 的 CQ 号。

表 3-2-32 SW2HW_CQ 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	flags																															
04h	start[63:32]																															
08h	start[31:0]																															
0Ch	logsize								usrpage																							
10h	comp_eqn																															
14h	pd																															
18h	lkey																															
1Ch																																
20h																																
24h																																
28h																																
2Ch	cqn																															
30h																																
34h																																
38h																																
3Ch																																

表 3-2-33 SW2HW_CQ 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
00h	31:0	flags	CQ 的相关配置属性，详见表 3-2-34	WR
04h	31:0	start[63:32]	CQ 队列的起始虚拟地址（高 32 位）	WR
08h	31:0	start[31:0]	CQ 队列的起始虚拟地址（低 32 位）（目前没用）	WR
0Ch	31:24	logsize	CQ 队列可存放 CQE 的个数，以 2 为底数	WR
	23:0	usrpage	UAR 页面指针（目前没用）	WR
10h	31:0	comp_eqn	与该 CQ 关联的完成事件队列的 EQ 号（目前没用）	WR
14h	31:0	pd	与该 CQ 关联的 PD 的 PD 号（目前没用）	WR
18h	31:0	lkey	CQ 队列的 lkey	WR
2Ch	31:0	cqn	该 CQ 的 CQ 号	WR

表 3-2-34 SW2HW_CQ 命令中 flag 字段各 bit 含义

Bit	Description
0 << 28	HGHCA_CQ_STATUS_OK
1 << 18	HGHCA_CQ_FLAG_TR
0 << 8	HGHCA_CQ_STATE_DISARMED
1 << 8	HGHCA_CQ_STATE_ARMED

3.2.5.2 hghca_HW2SW_CQ

该命令用于将位于 HCA 中的 CQ 上下文的内容无效。该命令向 HCR 寄存器写入的值如表 3-2-35 所示。

表 3-2-35 HW2SW_CQ 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0
out_param	0
in_modifier	cq_num，要释放 CQ 的 CQ 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_HW2SW_CQ =0x17

3.2.5.3 hghca_RESIZE_CQ

该命令用于修改 CQ 上下文的属性，包括 CQ 队列的 MR 及 CQ 队列的大小。该命令向 HCR 寄存器写入的值如表 3-2-36 所示。

表 3-2-36 RESIZE_CQ 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	cq_num，CQ 的 CQ 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_RESIZE_CQ =0x2c

该命令只有 in_param，没有 out_param。其中，in_param 用于存储 mailbox 的 DMA 物理地址，该地址下具体参数内容见表 3-2-37；in_modifier 表示输入的 CQC 的 CQ 号。

表 3-2-37 RESIZE_CQ 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	logsize																															
04h	lkey																															
08h																																
0Ch																																
10h																																
14h																																
18h																																
1Ch																																

表 3-2-38 RESIZE_CQ 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
00h	31:24	logsize	CQ 队列可存放 CQE 的个数，以 2 为底数	WR
04h	31:0	lkey	CQ 队列的 lkey	WR

3.2.6 QP 相关命令

3.2.6.1 hghca_MODIFY_QP

该命令用于修改 QP 上下文的属性，从而使能 QP 的发送和接收功能。该命令向 HCR 寄存器写入的值如表 3-2-39 所示。

表 3-2-39 MODIFY_QP 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	op_mod 为 3 时， 该字段为 NULL； op_mod 为 0 时， 该字段为 mailbox->dma，mailbox 的总线地址
out_param	0
in_modifier	qp_num， 要修改 QP 的 QP 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	op_mod 0x0: 有 inbox 0x3: 无 inbox（用于 any->RST 中）

op	各状态含义及状态转移需要的属性分别如表 A1-1、表 A1-2 所示
	CMD_RST2INIT_QPEE = 0x19, // RST->INIT
	CMD_INIT2RTR_QPEE = 0x1a, // INIT->RTR
	CMD_ RTR2RTS_QPEE = 0x1b, // RTR->RTS
	CMD_RTS2RTS_QPEE = 0x1c, // RTS->RTS
	CMD_SQERR2RTS_QPEE = 0x1d, // SQE->RTS
	CMD_2ERR_QPEE = 0x1e, // any->ERR
	CMD_RTS2SQD_QPEE = 0x1f, // RTS->SQD
	CMD_SQD2SQD_QPEE = 0x38, // SQD->SQD
	CMD_SQD2RTS_QPEE = 0x20, // SQD->RTS
	CMD_ERR2RST_QPEE = 0x21, // any->RST
	CMD_INIT2INIT_QPEE = 0x2d, // INIT->INIT

该命令只有 in_param，没有 out_param。其中，in_param 用于存储 mailbox 的 DMA 物理地址，该地址下具体参数内容见表 3-2-40； in_modifier 表示输入的 QPC 的 QP 号；op_modifier 用于指示该命令是否有 in_param。

表 3-2-40 MODIFY_QP 命令 in_param 存储布局

offset	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	opt_param_mask																															
04h																																
08h	flags																															
0Ch	mtu_msgmax								rq_entry_sz_log								sq_entry_sz_log								rlkey_arbel_sched_queue							
10h	usr_page																															
14h	local_qpn																															

18h	remote_qpn			
1Ch	port_pkey			
20h	rnr_retry	g_mylmc		
24h	ackto	mgid_index	static_rate	hop_limit
28h	sl_tclass_flowlabel			
2Ch	rgid[127:96]			
30h	rgid[95:64]			
34h	rgid[63:32]			
38h	rgid[31:0]			
3Ch	dlid(dmac[15:0])		slid(smac[15:0])	
40h	smac[47:16]			
44h	dmac[47:16]			
48h	sip			
4Ch	dip			
50h				
54h				
58h				
5Ch	pd			
60h	wqe_base			
64h	wqe_lkey			
68h				
6Ch	next_send_psn			
70h	cqn_snd			
74h	snd_wqe_base_l			
78h	snd_wqe_len			
7Ch	last_acked_psn			
80h	ssn			
84h	rnr_nextrecvpsn			
88h	ra_buff_indx			
8Ch	cqn_rcv			
90h	rcv_wqe_base_l			
94h	rcv_wqe_len			
98h	qkey			
9Ch	rmsn			
A0h				
A4h				
A8h				
ACh				
B0h				
B4h				
B8h				
BCh	rq_wqe_counter		sq_wqe_counter	

表 3-2-41 MODIFY_QP 命令 in_param 各字段含义

Offset	Bits	Name	Description	Access
00h	31:0	opt_param_mask	本次命令要修改属性的 mask，详见表 A1-3	WR
08h	31:0	flags	QP 状态标志	WR
0Ch	31:24	mtu_msgmax	MTU [7:5] & 消息最大值[4:0] `enum ib_mtu { IB_MTU_256 = 1, IB_MTU_512 = 2, IB_MTU_1024 = 3, IB_MTU_2048 = 4, IB_MTU_4096 = 5 };` maxmsg 为字节数的 log 值	WR
	23:16	rq_entry_sz_log	RQ 中一个 WQE 条目的大小(byte)，以 2 为底数	WR
	15:8	sq_entry_sz_log	SQ 中一个 WQE 条目的大小（byte），以 2 为底数	WR
	7:0	rlkey_arbel_sched_queue	（目前没用）	WR
10h	31:0	usr_page	pfn of UAR page（目前没用）	WR
14h	31:0	local_qpn	该 QP 的 QP 号	WR
18h	31:0	remote_qpn	远端的 QP 号，用于连接服务类型	WR
1Ch	31:0	port_pkey	端口号[26:24] pkey index[6:0]	WR
20h	31:24	rnr_retry	3bit 的请求方接收到远端 RNR NAK 后重发的次数（在报告错误之前）。7 代表无限重发。	WR
	23:16	g_mylmc	has grh[7:7]：是否使用 GRH local mask control[6:0]：用于向端口指定 LID（目前没用）	WR
24h	31:24	ackto	ack timeout（目前没用）	WR
	23:16	mgid_index	sgid_index,端口 GID 表的 index（目前没用）	WR
	15:8	static_rate	获得端口静态速率（目前没用）	WR
	7:0	hop_limit	数据包经历的跳数限制（目前没用）	WR
28h	31:0	sl_tclass_flowlabel	sl & traffic class & flow label（目前没用）	WR
2Ch	31:0	rgid[127:96]	目的 GID（目前没用）	WR
30h	31:0	rgid[95:64]	目的 GID（目前没用）	WR
34h	31:0	rgid[63:32]	目的 GID（目前没用）	WR
38h	31:0	rgid[31:0]	目的 GID（目前没用）	WR
3Ch	31:16	dlid	目的 LID，或目的 MAC 的低 16 位（仅在 RoCE 模式下有用）	WR
	15:0	slid	源 LID，或源 MAC 的低 16 位（仅在 RoCE 模式下有用）	WR
40h	31:0	smac[47:16]	源 MAC 的高 32 位（仅在 RoCE 模式下有用）	WR
44h	31:0	dmac[47:16]	目的 MAC 的高 32 位（仅在 RoCE 模式下有	WR

			用）	
48h	31:0	sip	源 IP（仅在 RoCE 模式下有用）	WR
4Ch	31:0	dip	目的 IP（仅在 RoCE 模式下有用）	WR
5Ch	31:0	pd	QP 所在保护域	WR
60h	31:0	wqe_base	（目前没用）	WR
64h	31:0	wqe_lkey	QP 队列所在 Memory Region 的 lkey（目前没用）	WR
6Ch	31:0	next_send_psn	下一个要发送的消息的 PSN（包序列号）	WR
70h	31:0	cqn_snd	SQ 的 CQ 号	WR
74h	31:0	snd_wqe_base_l	发送队列的基地址的 lkey	WR
78h	31:0	snd_wqe_len	发送队列的总长度（byte）	WR
7Ch	31:0	last_acked_psn	之前 ACK 的 PSN	WR
80h	31:0	ssn	（目前没用）	WR
84h	31:0	rnr_nextrecvpsn	Recv not Ready[31:24] & ePSN[23:0]	WR
88h	31:0	ra_buff_indx	（目前没用）	WR
8Ch	31:0	cqn_rcv	RQ 的 CQ 号	WR
90h	31:0	rcv_wqe_base_l	接收队列的基地址的 lkey	WR
94h	31:0	rcv_wqe_len	接收队列的总长度（byte）	WR
98h	31:0	qkey	在数据报服务类型中用于验证远端发送方对本地接收队列的访问权限，须在接收队列 WQE 提交前建立好（目前没用）	WR
9Ch	31:0	rmsn	（目前没用）	WR
A0h	31:16	rq_wqe_counter	（目前没用）	WR
	15:0	sq_wqe_counter	（目前没用）	WR

表 3-2-42 MODIFY_QP 命令中 flags 字段各 bit 位含义

Flags Section	Value	
31:28	HGHCA_QP_STATE_RST	= 0
	HGHCA_QP_STATE_INIT	= 1
	HGHCA_QP_STATE_RTR	= 2
	HGHCA_QP_STATE_RTS	= 3
	HGHCA_QP_STATE_SQE	= 4
	HGHCA_QP_STATE_SQD	= 5
	HGHCA_QP_STATE_ERR	= 6
23:16	HGHCA_QP_ST_RC	= 0
	HGHCA_QP_ST_UC	= 1
	HGHCA_QP_ST_RD	= 2
	HGHCA_QP_ST_UD	= 3

3.2.6.2 hghca_QUERY_QP

该命令用于查询 QP 上下文的属性。该命令向 HCR 寄存器写入的值如表 3-2-43 所示。

表 3-2-43 QUERY_QP 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0

out_param	mailbox->dma，mailbox 的总线地址
in_modifier	qpn，QP 的 QP 号
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_QUERY_QP =0x22

该命令只有 out_param，没有 in_param。其中，out_param 用于存储 mailbox 的 DMA 物理地址，该地址下具体参数内容同 MODIFY_QP 命令，此处不再列出；in_modifier 表示要查询 QP 的 QP 号。

3.2.7 其它命令

3.2.7.1 hghca_NOP

该命令用于让 HCA 触发一次无操作命令，用于测试 HCA 中 HCR 命令是否可用。该命令的 HCR 寄存器的参数入表 3-2-44 所示。

表 3-2-44 NOP 命令向 HCR 寄存器写入的值

HCR Fields	Value
in_param	0
out_param	0
in_modifier	0x1f
token	CMD_POLL_TOKEN 0xffff
status	READ ONLY
go	1
E	0
op_modifier	0
op	CMD_NOP =0x31

3.3 CQ 相关数据结构

3.3.1 CQE 相关数据结构

3.3.1.1 CQE 正常状态数据结构

表 3-3-1 为正常状态下 CQE 的数据结构，注意，收到的数据格式为小端模式。表 3-3-2 为相关数据字段的含义。

表 3-3-1 正常状态下 CQE 数据结构

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	my_qpn																															
04h	my_ee																															
08h	rqpn																															
0Ch	rlid																g_mlpath								sl_ipok							
10h	imm_etype_pkey_eec																															

14h	byte_cnt			
18h	wqe			
1Ch	owner		is_send	opcode

表 3-3-2 正常状态下 CQE 数据结构中各个字段含义

Offset	Bits	Name	Description	Access
00h	31:0	my_qpn	本地发送完成事件的 QP 的 QPN	WR
04h	31:0	my_ee	（暂未使用）	WR
08h	31:0	rqpn	远端 QPN	WR
0Ch	7:0	sl_ipok	service_level， ipaddr csum ok（暂未使用）	WR
	15:8	g_mlpath	has grh[7:7]：是否使用 GRH local mask control[6:0]：用于向端口指定 LID（暂未使用）	WR
	31:16	rlid	远端 LID	WR
10h	31:0	imm	接收完成事件保存的立即数	WR
14h	31:0	byte_cnt	该完成事件所完成工作发送/接收数据量大大小	WR
18h	31:0	wqe	WQE offset in WQ	WR
1Ch	7:0	opcode	发送：WQ 中的的操作码（见表 3-3-3）； 接收：收到最后一个数据包的操作码	WR
	15:8	is_send	1 表示为发送完成事件	WR
	31:24	owner	HGHCA_CQ_ENTRY_OWNER_SW (0<<7) HGHCA_CQ_ENTRY_OWNER_HW (1<<7)	WR

表 3-3-3 工作请求的操作码

Name	Value	Note
HGHCA_OPCODE_NOP	0x00	
HGHCA_OPCODE_RDMA_WRITE	0x08	
HGHCA_OPCODE_RDMA_WRITE_IMM	0x09	
HGHCA_OPCODE_SEND	0x0a	
HGHCA_OPCODE_SEND_IMM	0x0b	
HGHCA_OPCODE_RDMA_READ	0x10	
HGHCA_OPCODE_ATOMIC_CS	0x11	
HGHCA_OPCODE_ATOMIC_FA	0x12	
HGHCA_OPCODE_BIND_MW	0x18	
HGHCA_OPCODE_INVALID	0xff	无效操作码，CQ 中指示错误状态时使用

3.3.1.1 CQE 错误状态数据结构

表 3-3-1 为错误状态下 CQE 的数据结构，注意，收到的数据格式为小端模式。表 3-3-2 为相关数据字段的含义。

表 3-3-4 错误状态下 CQE 数据结构

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	my_qpn																															
04h																																
08h																																

0Ch			
10h	db_cnt	vender_err	syndrome
14h			
18h	wqe		
1Ch	owner		opcode

表 3-3-2 错误状态下 CQE 数据结构中各个字段含义

Offset	Bits	Name	Description	Access
00h	31:0	my_qpn	本地发送完成事件的 QP 的 QPN	WR
10h	7:0	syndrome	CQE 的错误码，详见表 3-3-3	WR
	15:8	vender_err	（暂未使用）	
	31:16	db_cnt	（暂未使用）	
18h	31:0	wqe	WQE offset in WQ	WR
1Ch	7:0	opcode	发送：WQ 中的的操作码（见表 3-3-3）； 接收：收到最后一个数据包的操作码	WR
	31:24	owner	HGHCA_CQ_ENTRY_OWNER_SW (0<<7) HGHCA_CQ_ENTRY_OWNER_HW (1<<7)	WR

表 3-3-3 CQE 中 syndrome 字段的错误码

Name	Value
SYNDROME_LOCAL_LENGTH_ERR	0x1
SYNDROME_LOCAL_QP_OP_ERR	0x2
SYNDROME_LOCAL_EEC_OP_ERR	0x3
SYNDROME_LOCAL_PROT_ERR	0x4
SYNDROME_WR_FLUSH_ERR	0x5
SYNDROME_MW_BIND_ERR	0x6
SYNDROME_BAD_RESP_ERR	0x10
SYNDROME_LOCAL_ACCESS_ERR	0x11
SYNDROME_REMOTE_INVAL_REQ_ERR	0x12
SYNDROME_REMOTE_ACCESS_ERR	0x13
SYNDROME_REMOTE_OP_ERR	0x14
SYNDROME_RETRY_EXC_ERR	0x15
SYNDROME_RNR_RETRY_EXC_ERR	0x16
SYNDROME_LOCAL_RDD_VIOL_ERR	0x20
SYNDROME_REMOTE_INVAL_RD_REQ_ERR	0x21
SYNDROME_REMOTE_ABORTED_ERR	0x22
SYNDROME_INVAL_EECN_ERR	0x23
SYNDROME_INVAL_EEC_STATE_ERR	0x24

3.4 QP 相关数据结构

QP 中包括一个发送队列（SQ）和一个接收队列（RQ）。他们作为软硬件之间的接口分别用于传递发送描述符和接收描述符一个描述符又被称为一个 Work Queue Element（WQE），包含了数据发送或接收的必要信息。每个 WQE 都是变长的，它们包含了多种单元（unit）

的组合，其主要包括的单元如表 3-4-1 所示。下面每一小节将对各个单元进行详细说明。

表 3-4-1 QP 中 WQE 包含的单元

Name	Description
hghca_next_unit	每个 WQE 的第一个单元；每个 WQE 都会用到该单元
hghca_raddr_unit	描述 RDMA 请求中远端内存地址的单元；仅在 RC 和 UC 服务中存在
hghca_atomic_unit	包含用于原子操作的数据的单元。swap_add 字段用于 FetchAndAdd；compare 字段用于 CmpAndSwap。
hghca_ud_unit	包含 UD 服务的所有基本信息
hghca_inline_unit	inline 方式下使用该单元，该字段为一个 32bit 的数值
hghca_data_unit	不使用 inline 操作时用到该单元。该单元指定了一段本地虚拟地址空间

各个单元之间有序的关系，从而保证了硬件可以准确解析出每个单元的含义。对于 RC 或 UC 的连接类型的 QP，其可能的 WQE 结构如图 3-4-1 所示。对于 UD 类型的 QP，其可能的 WQE 结构如图 3-4-2 所示。

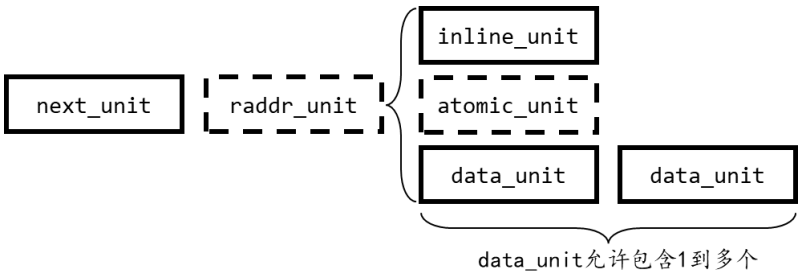

图 3-4-1 RC UC 类型的 WQE 的结构，其中虚线部分为可选内容

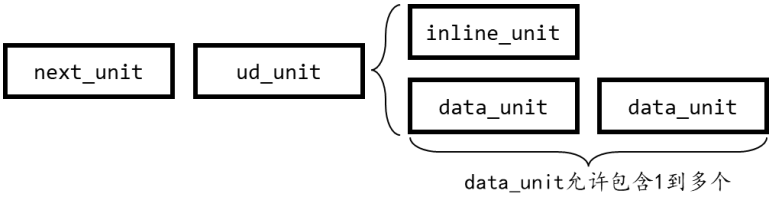

图 3-4-2 UD 类型的 WQE 结构，其中 inline 和 data unit 使用第一个双字的最高位来区分

3.4.1 hghca_next_unit 单元

表 3-4-2 hghca_next_unit 单元中各个字段的布局

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	nda																											nop				
04h	nee																								D	F	nds					
08h																												C	E	S		
0Ch	imm_data																															

表 3-4-3 hghca_next_unit 单元中各个字段的含义

Offset	Bits	Name	Description	Access
00h	4:0	nop	next opcode	WR
	31:6	nda	next descriptor address	WR
04h	5:0	nds	next descriptor size	WR
	6:6	F	next fence	WR
	7:7	D	next DBD（目前没用）	WR

	31:8	nee	next EE address	WR
08h	1:1	S	当前 Solicit Event，通知接收端产生 CQ 事件（目前没用）	WR
	2:2	E	当前 Event Generate（目前没用）	WR
	3:3	C	当前 CQ Update（目前没用）	WR
0Ch	31:0	imm_data	当前 WQE 的立即数	WR

3.4.2 hghca_raddr_unit 单元

表 3-4-4 hghca_raddr_unit 单元中各个字段的布局

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	raddr[31:0]																															
04h	raddr[63:32]																															
08h	rkey																															
0Ch																																

表 3-4-5 hghca_raddr_unit 单元中各个字段的含义

Offset	Bits	Name	Description	Access
00h	31:0	raddr[31:0]	远端存放数据的起始地址（虚拟地址）	WR
04h	31:0	raddr[63:32]		
08h	31:0	rkey	远端 MR 的 rkey	WR

3.4.3 hghca_atomic_unit 单元

表 3-4-6 hghca_atomic_unit 单元中各个字段的布局

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	swap_add[31:0]																															
04h	swap_add[63:32]																															
08h	cmp[31:0]																															
0Ch	cmp[63:32]																															

表 3-4-7 hghca_atomic_unit 单元中各个字段的含义

Offset	Bits	Name	Description	Access
00h	31:0	swap_add[31:0]	用于 swap 的数据	WR
04h	31:0	swap_add[63:32]		
08h	31:0	cmp[31:0]	用于 cmp 的数据	WR
0Ch	31:0	cmp[63:32]		

3.4.4 hghca_ud_unit 单元

表 3-4-8 hghca_ud_unit 单元中各个字段的布局

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

00h		port
04h	dlid(dmac[15:0])	slid(smac[15:0])
08h	smac[47:16]	
0Ch	dmac[47:16]	
10h	sip	
14h	dip	
18h		
1Ch		
20h	dqpn	
24h	qkey	
28h		
2Ch		

表 3-4-9 hghca_ud_unit 单元中各个字段的含义

Offset	Bits	Name	Description	Access
00h	31:0	port	要通过哪个端口号发送数据	WR
04h	15:0	slid(smac[15:0])	源 LID（或 MAC 的低 16 位）	WR
	31:16	dlid(dmac[15:0])	目的 LID（或 MAC 的低 16 位）	WR
08h	31:0	smac[47:16]	源 MAC 的高 32 位	WR
0Ch	31:0	dmac[47:16]	目的 MAC 的高 32 位	WR
10h	31:0	sip	源 IP	WR
14h	31:0	dip	目的 IP	WR
20h	31:0	dqpn	目的 QP 号	WR
24h	31:0	qkey	目的 pkey	WR

3.4.5 hghca_inline_unit 单元

表 3-4-10 为 hghca_inline_unit 单元中各个字段的布局，其中第一个双字的最高位为 1，用于标志该单元为一个 inline 单元。

表 3-4-10 hghca_inline_unit 单元中各个字段的布局

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	1	length																														
04h	Data																															
08h	...																															

表 3-4-11 hghca_inline_unit 单元中各个字段的含义

Offset	Bits	Name	Description	Access
00h	30:0	length	inline 单元中数据字段的长度	WR
04h	31:0	data	inline 单元要传输的数据	WR

3.4.6 hghca_data_unit 字段

表 3-4-12 hghca_data_unit 单元中各个字段的布局

offset	+3								+2								+1								+0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00h	0	byte_cnt																														
04h	lkey																															
08h	addr																															

表 3-4-13 hghca_data_unit 单元中各个字段的含义

Offset	Bits	Name	Description	Access
00h	30:0	byte_cnt	要发送数据的字节数	WR
04h	31:0	lkey	待发送数据所在的 MR 的 lkey	WR
08h	31:0	addr[31:0]	待发送数据的起始虚拟地址(低 32 位)	WR
0Ch	31:0	addr[63:32]	待发送数据的起始虚拟地址(高 32 位)	WR

A1 各种数据包及命令格式

表 A1-1 QP 各个状态的能力

State	Capability
RESET	该状态下写入 WQE 会返回立即错误； 到来的消息（目标为该 QP）会被静默丢弃
INIT	仅能从 RESET 状态进入该状态，且仅能通过 Modify_QP 进入该状态
RTR	在该状态下，向 RQ 提交的工作请求可以被正常处理； 收到的消息可以被正常处理
RTS	仅能从 RTR 或 SQD 状态进入该状态； SQ 及 RQ 中的工作请求都将被正常处理
SQD	该状态下可以提交工作请求； 收到的消息可以被正常处理； 仅能从 RTS 进入该状态
SQE	仅能由于完成错误，由 RTS 状态进入该状态； 收到的消息可以被正常处理
ERR	该状态下，所有的正常处理都会停止； 到来的消息（目标为该 QP）会被静默丢弃； 仅能通过 Modify_QP 命令，将 QP 从该状态转换到 RESET 状态

表 A1-2 QP 状态含义及状态转移所需属性

State	Transform	QP Type	req_param	op_param
RESET	RST->RST	UD/UC/RC	无	无
	RST->INIT	UD	IB_QP_PKEY_INDEX IB_QP_PORT IB_QP_QKEY	无
		UC/RC	IB_QP_PKEY_INDEX IB_QP_PORT IB_QP_ACCESS_FLAGS	无

INIT	INIT->RST	UD/UC/RC	无	无
	INIT->ERR	UD/UC/RC	无	无
	INIT->INIT	UD	IB_QP_PKEY_INDEX IB_QP_PORT IB_QP_QKEY	无
		UC/RC	IB_QP_PKEY_INDEX IB_QP_PORT IB_QP_ACCESS_FLAGS	无
	INIT->RTR	UD	无	IB_QP_PKEY_INDEX IB_QP_QKEY
		UC	IB_QP_AV IB_QP_PATH_MTU IB_QP_DEST_QPN IB_QP_RQ_PSN	IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_PKEY_INDEX
		RC	IB_QP_AV IB_QP_PATH_MTU IB_QP_DEST_QPN IB_QP_RQ_PSN IB_QP_MIN_RNR_TIMER	IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_PKEY_INDEX
RTR	RTR->RST	UD/UC/RC	无	无
	RTR->ERR	UD/UC/RC	无	无
	RTR->RTS	UD	IB_QP_SQ_PSN	IB_QP_CUR_STATE IB_QP_QKEY
		UC	IB_QP_SQ_PSN	IB_QP_CUR_STATE IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_PATH_MIG_STATE
		RC	IB_QP_TIMEOUT IB_QP_RETRY_CNT IB_QP_RNR_RETRY IB_QP_SQ_PSN	IB_QP_CUR_STATE IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_MIN_RNR_TIMER IB_QP_PATH_MIG_STATE
RTS	RTS->RST	UD/UC/RC	无	无
	RTS->ERR	UD/UC/RC	无	无
	RTS->RTS	UD	无	IB_QP_CUR_STATE IB_QP_QKEY
		UC	无	IB_QP_CUR_STATE IB_QP_ACCESS_FLAGS IB_QP_ALT_PATH IB_QP_PATH_MIG_STATE
		RC	无	IB_QP_CUR_STATE IB_QP_ACCESS_FLAGS IB_QP_ALT_PATH IB_QP_PATH_MIG_STATE

				IB_QP_MIN_RNR_TIMER
	RTS->SQD	UD/UC/RC	无	IB_QP_EN_SQD_ASYNC_NOTIFY
SQD	SQD->RST	UD/UC/RC	无	无
	SQD->ERR	UD/UC/RC	无	无
	SQD->RTS	UD	无	IB_QP_CUR_STATE IB_QP_QKEY
		UC	无	IB_QP_CUR_STATE IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_PATH_MIG_STATE
		RC	无	IB_QP_CUR_STATE IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_MIN_RNR_TIMER IB_QP_PATH_MIG_STATE
	SQD->SQD	UD	无	IB_QP_PKEY_INDEX IB_QP_QKEY
		UC	无	IB_QP_AV IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_PKEY_INDEX IB_QP_PATH_MIG_STATE
		RC	无	IB_QP_PORT IB_QP_AV IB_QP_TIMEOUT IB_QP_RETRY_CNT IB_QP_RNR_RETRY IB_QP_ALT_PATH IB_QP_ACCESS_FLAGS IB_QP_PKEY_INDEX IB_QP_MIN_RNR_TIMER IB_QP_PATH_MIG_STATE
SQE	SQE->RST	UD/UC/RC	无	无
	SQE->ERR	UD/UC/RC	无	无
	SQE->RTS	UD	无	IB_QP_CUR_STATE IB_QP_QKEY
		UC	无	IB_QP_CUR_STATE IB_QP_ACCESS_FLAGS
ERR	ERR->RST	UD/UC/RC	无	无
	ERR->ERR	UD/UC/RC	无	无

表 A1-3 Modify QP 可能修改的属性及含义

属性	含义
IB_QP_STATE = 1,	下一个 QP 状态

IB_QP_CUR_STATE = (1<<1),	当前 QP 状态
IB_QP_EN_SQD_ASYNC_NOTIFY = (1<<2),	使能 SQD 异步事件提醒，使能后，当 QP 状态变为 SQD 时，会产生相应的异步事件（目前没用）
IB_QP_ACCESS_FLAGS = (1<<3),	允许的远程操作，用于 RC 和 UC QP IBV_ACCESS_REMOTE_WRITE - Allow incoming RDMA Writes on this QP IBV_ACCESS_REMOTE_READ - Allow incoming RDMA Reads on this QP
IB_QP_PKEY_INDEX = (1<<4),	设置 p_key index（primary path），其描述了到远端 QP 的路径信息
IB_QP_PORT = (1<<5),	端口号（primary path）
IB_QP_QKEY = (1<<6),	设置 Q_Key,仅与 UD QP 相关
IB_QP_AV = (1<<7),	设置 Address Vector（primary path），详细参数见表 A1-4（目前没用）
IB_QP_PATH_MTU = (1<<8),	3bit 的路径 MTU 的值
IB_QP_TIMEOUT = (1<<9),	5bit 的 QP 等待远端 QP 发送的 ACK/NAK 的最小超时时间。仅与 RC QP 相关
IB_QP_RETRY_CNT = (1<<10),	3bit 的请求方未收到远端应答的重发次数（在报告错误之前）
IB_QP_RNR_RETRY = (1<<11),	3bit 的请求方接收到远端 RNR NAK 后重发的次数（在报告错误之前）。7 代表无限重发。
IB_QP_RQ_PSN = (1<<12),	24bit 的接收队列的 PSN，用于 RC 和 UC QP
IB_QP_MAX_QP_RD_ATOMIC = (1<<13),	该 QP 作为**请求发起方**可以同时执行的 RDMA Read & Atomic 操作的个数，仅在 RC QP 中有效（目前没用）
IB_QP_ALT_PATH = (1<<14),	暂不实现（目前没用）
IB_QP_MIN_RNR_TIMER = (1<<15),	5bit 最小的 RNR（Receive not Ready）NAK 计时器字段值，当一个消息到来时，RQ 中没有相关 WQE，需要发送 RNR NAK。仅与 RC QP 相关
IB_QP_SQ_PSN = (1<<16),	24bit 的发送队列的 PSN，用于任何类型的 QP
IB_QP_MAX_DEST_RD_ATOMIC = (1<<17),	该 QP 作为**接收方**可以同时执行的 RDMA Read & Atomic 操作的个数，仅在 RC QP 中有效（目前没用）
IB_QP_PATH_MIG_STATE = (1<<18),	暂不实现（目前没用）
IB_QP_CAP = (1<<19),	未实现（目前没用）
IB_QP_DEST_QPN = (1<<20),	24bit 的远端 RC，UC QP 的 QP 号

表 A1-4 Address Vector 参数及含义

属性	大小（bit）	含义
port_pkey	32	端口号 & Pkey index
rnr_retry	8	3bit 的请求方接收到远端 RNR NAK 后重发的次数（在报告错误之前）。7 代表无限重发。
g_mylmc	8	？？？？
rlid	16	dlid，目的 LID
ackto	8	ack timeout
mgid_index	8	sgid_index,端口 GID 表的 index

static_rate	8	获得端口静态速率
hop_limit	8	数据包经历的跳数限制
sl_tclass_flowlabel	32	sl & traffic class & flow label
rgid	128	dgid，目的 GID

A2 传输数据包各字段的来源

表 A2-1 传输数据包各字段的来源

header	Field	RC	UC	UD
LRH	Virtual lane	根据 SL 以及都那口中的 SL->VL 表获得		
	LRH version	Fixed		
	Service Level	QP		AV
	LRH next header – IBA transport bit	Fixed == 1		
	LRH next header – GRH bit	QP		AV
	DLID	QP		AV
	Packet length	从数据&包头长度计算获得		
	SLID(part not covered by LMC)	从端口属性中获得		
	SLID(part covered by LMC)	QP		AV
BTH	OpCode	WR		
	BTH version	Fixed == 0		
	Partition key	QP		
	Destination queue pair	QP		WR
	Pad count	从数据&包头长度计算获得		
	Solicited event	WR		
	Packet sequence number	Computed from QP state		
DETH	Queue key	N/A	N/A	WR or QP(WR 没写，就从 QP 里面拿)
	Source queue pair	QP		
RETH	Virtual address	WR		N/A
	R_Key	WR		N/A
	DMA length	WR		N/A
IETH	R_Key	WR	N/A	
AETH	Message Sequence Number	Computed	N/A	N/A
AETH	Syndrome	Computed	N/A	N/A
ImmDT	Immediate Data	WR		