

在cdma driver中，与damengine交互的是virtual dma controller，而不是physical dma controller.

同时，与damengine交互的也是virtual channel，而非physical channel。当然virtual channel最终需要

mapping to physical channel.

in mv61\_vdma\_probe()

下面code就是vdma与dmaengine framework的交互。

```

1. INIT_LIST_HEAD(&mv61v->dma.channels);
2. for (i = 0; i < pdata->nr_virt_chans[vid]; i++, mv61v->dma.chancnt++) {
    ②
3.     struct mv61_vdma_chan *mv61vc = &mv61v->chan[i];
4.     mv61vc->mv61v = mv61v;
5.     mv61vc->def.valid = 0;
6.     mv61vc->status = DMA_COMPLETE;
7.     mv61vc->chan.device = &mv61v->dma;
8.     mv61vc->chan.cookie = mv61vc->completed = 1;
9.     mv61vc->chan.chan_id = i;
10.    mv61vc->residue = 0;
11.
12.    list_add_tail(&mv61vc->chan.device_node, &mv61v->dma.channels);
    ④
13.    spin_lock_init(&mv61vc->lock);
14.    INIT_LIST_HEAD(&mv61vc->active_list);
15.    INIT_LIST_HEAD(&mv61vc->queue);
16.    INIT_LIST_HEAD(&mv61vc->complete_list);
17. }
18.
19.
20. mv61_vpmap_dispatch_init(mv61v, pdata);
21.
22. mv61v->dma.dev = &pdev->dev;
23. if(dma_has_cap(DMA_MEMCPY, mv61v->dma.cap_mask))
24.     mv61v->dma.device_prep_dma_memcpy = mv61vc_prep_dma_memcpy;
25. if(dma_has_cap(DMA_SLAVE, mv61v->dma.cap_mask))
26.     mv61v->dma.device_prep_slave_sg = mv61vc_prep_slave_sg;
27. if(dma_has_cap(DMA_CYCLIC, mv61v->dma.cap_mask))
28.     mv61v->dma.device_prep_dma_cyclic = mv61vc_prep_dma_cyclic;
29. mv61v->dma.device_control = mv61vc_device_control;
30. mv61v->dma.device_alloc_chan_resources = mv61vc_alloc_chan_resources;
31. mv61v->dma.device_free_chan_resources = mv61vc_free_chan_resources;
32. mv61v->dma.device_tx_status = mv61vc_tx_status;
33. mv61v->dma.device_issue_pending = mv61vc_issue_pending;
34. mv61v->dma.copy_align = MV61_MEMCPY_ALIGN;
35. mv61v->dma.fill_align = MV61_MEMFILL_ALIGN;
36.
37. {
38.     char *description = NULL;
39.
40.     switch(vid) {
41.     case MV61_VDMA_OWNED:
42.         description = "owned";
43.         break;
44.     case MV61_VDMA_SHARED:
45.         description = "shared";
46.         break;
47.     case MV61_VDMA_CYCLIC:
48.         description = "cyclic";
49.         break;
50.     case MV61_VDMA_MEMOPS:
51.         description = "memops";

```

```

52.         break;
53.     default:
54.         description = "unknown";
55.         break;
56.     }
57.
58.     dev_printk(KERN_INFO, &pdev->dev, "Virtual DMA Controller "
59.         "type %d:%s, "
60.         "%d virt channels, %d phys channels\n",
61.         vid, description,
62.         mv61v->dma.chancnt,
63.         pdata->nr_pool_chans[vid]);
64. }
65.
66.     ret = dma_async_device_register(&mv61v->dma); ⑥

```

①

mv61v->dma.channels是dmaengine的channel list。

②

pdata->nr\_virt\_chans[vid]

对应的virtual dma controller管理的virtual channel number(根据dts中setting)

③

mv61vc->chan.chan\_id = i;

chan\_id是virtual dma controller管理的virtual channel数组的index

④

virtual channel都会链接起来

⑤

下面都是根据dmaengine的要求的设置

⑥

从这里可看出是virtual dma controller与dmaengine交互。