****
  337453 Tue Jan 15 10:23:16 2019
new/usr/src/uts/common/io/sata/adapters/ahci/ahci.c
10091 smatch fixes for ahci.c
****
______unchanged_portion_omitted_

```
/*
 * Allocate the ahci_port_t including Received FIS and Command List.
 * The argument - port is the physical port number, and not logical
 * port number seen by the SATA framework.
 */
static int
ahci_alloc_port_state(ahci_ctl_t *ahci_ctlp, uint8_t port)
{
        dev_info_t *dip = ahci_ctlp->ahcictl_dip;
        ahci_port_t *ahci_portp;
        char taskq_name[64] = "event_handle_taskq";

        ASSERT(MUTEX_HELD(&ahci_ctlp->ahcictl_mutex));

        ahci_portp =
            (ahci_port_t *)kmem_zalloc(sizeof (ahci_port_t), KM_SLEEP);

        ahci_ctlp->ahcictl_ports[port] = ahci_portp;
        ahci_portp->ahciport_port_num = port;

        /* Initialize the port condition variable */
        cv_init(&ahci_portp->ahciport_cv, NULL, CV_DRIVER, NULL);

        /* Initialize the port mutex */
        mutex_init(&ahci_portp->ahciport_mutex, NULL, MUTEX_DRIVER,
            (void *)(uintptr_t)ahci_ctlp->ahcictl_intr_pri);

        mutex_enter(&ahci_portp->ahciport_mutex);

        /*
         * Allocate memory for received FIS structure and
         * command list for this port
         */
        if (ahci_alloc_rcvd_fis(ahci_ctlp, ahci_portp) != AHCI_SUCCESS) {
                goto err_case1;
        }

        if (ahci_alloc_cmd_list(ahci_ctlp, ahci_portp) != AHCI_SUCCESS) {
                goto err_case2;
        }

        /* Setup PxCMD.CLB, PxCMD.CLBU, PxCMD.FB, and PxCMD.FBU */
        if (ahci_setup_port_base_addresses(ahci_ctlp, ahci_portp) !=
            AHCI_SUCCESS) {
                goto err_case3;
        }

        (void) snprintf(taskq_name + strlen(taskq_name),
            sizeof (taskq_name) - strlen(taskq_name),
            "_port%d", port);

        /* Create the taskq for the port */
        if ((ahci_portp->ahciport_event_taskq = ddi_taskq_create(dip,
            taskq_name, 2, TASKQ_DEFAULTPRI, 0)) == NULL) {
                cmn_err(CE_WARN, "!ahci%d: ddi_taskq_create failed for event "
                    "handle", ddi_get_instance(ahci_ctlp->ahcictl_dip));
                goto err_case3;
        }
```

```
        /* Allocate the argument for the taskq */
        ahci_portp->ahciport_event_args =
            kmem_zalloc(sizeof (ahci_event_arg_t), KM_SLEEP);

        ahci_portp->ahciport_event_args->ahciea_addrp =
            kmem_zalloc(sizeof (ahci_addr_t), KM_SLEEP);

        if (ahci_portp->ahciport_event_args == NULL)
                goto err_case4;

        /* Initialize the done queue */
        ahci_portp->ahciport_doneq = NULL;
        ahci_portp->ahciport_doneqtail = &ahci_portp->ahciport_doneq;
        ahci_portp->ahciport_doneq_len = 0;

        mutex_exit(&ahci_portp->ahciport_mutex);

        return (AHCI_SUCCESS);

err_case4:
        ddi_taskq_destroy(ahci_portp->ahciport_event_taskq);

err_case3:
        ahci_dealloc_cmd_list(ahci_ctlp, ahci_portp);

err_case2:
        ahci_dealloc_rcvd_fis(ahci_portp);

err_case1:
        mutex_exit(&ahci_portp->ahciport_mutex);
        mutex_destroy(&ahci_portp->ahciport_mutex);
        cv_destroy(&ahci_portp->ahciport_cv);

        kmem_free(ahci_portp, sizeof (ahci_port_t));

        return (AHCI_FAILURE);
}
```
______unchanged_portion_omitted_