```
**********************************************************
  337435 Fri May  3 08:30:29 2019
new/usr/src/uts/common/io/sata/adapters/ahci/ahci.c
10055 recursive mutex enter in ahci
**********************************************************
______unchanged_portion_omitted_

1123 /*
1124  * The detach entry point for dev_ops.
1125  */
1126 static int
1127 ahci_detach(dev_info_t *dip, ddi_detach_cmd_t cmd)
1128 {
1129         ahci_ctl_t *ahci_ctlp;
1130         int instance;
1131         int ret;

1133         instance = ddi_get_instance(dip);
1134         ahci_ctlp = ddi_get_soft_state(ahci_statep, instance);

1136         AHCIDBG(AHCIDBG_ENTRY, ahci_ctlp, "ahci_detach enter", NULL);

1138         switch (cmd) {
1139         case DDI_DETACH:

1141                 /* disable the interrupts for an uninterrupted detach */
1142                 mutex_enter(&ahci_ctlp->ahcictl_mutex);
1143                 ahci_disable_all_intrs(ahci_ctlp);
1144                 mutex_exit(&ahci_ctlp->ahcictl_mutex);

1146                 /* unregister from the sata framework. */
1147                 ret = ahci_unregister_sata_hba_tran(ahci_ctlp);
1148                 if (ret != AHCI_SUCCESS) {
1149                         mutex_enter(&ahci_ctlp->ahcictl_mutex);
1150                         ahci_enable_all_intrs(ahci_ctlp);
1151                         mutex_exit(&ahci_ctlp->ahcictl_mutex);
1152                         return (DDI_FAILURE);
1153                 }

1155                 ahci_em_fini(ahci_ctlp);

1157                 mutex_enter(&ahci_ctlp->ahcictl_mutex);

1159                 /* stop the watchdog handler */
1160                 (void) untimeout(ahci_ctlp->ahcictl_timeout_id);
1161                 ahci_ctlp->ahcictl_timeout_id = 0;

1163                 mutex_exit(&ahci_ctlp->ahcictl_mutex);

1165                 /* uninitialize the controller */
1166                 ahci_uninitialize_controller(ahci_ctlp);

1168                 /* remove the interrupts */
1169                 ahci_rem_intrs(ahci_ctlp);

1171                 /* deallocate the ports structures */
1172                 ahci_dealloc_ports_state(ahci_ctlp);

1174                 /* destroy mutex */
1175                 mutex_destroy(&ahci_ctlp->ahcictl_mutex);

1177                 /* teardown the pci config */
1178                 pci_config_teardown(&ahci_ctlp->ahcictl_pci_conf_handle);

1180                 /* remove the reg maps. */
1181                 ddi_regs_map_free(&ahci_ctlp->ahcictl_ahci_acc_handle);
```



```
1183                 /* release fma resource */
1184                 ahci_fm_fini(ahci_ctlp);

1186                 /* free the soft state. */
1187                 ddi_soft_state_free(ahci_statep, instance);

1189                 return (DDI_SUCCESS);

1191         case DDI_SUSPEND:

1193                 /*
1194                  * The steps associated with suspension must include putting
1195                  * the underlying device into a quiescent state so that it
1196                  * will not generate interrupts or modify or access memory.
1197                  */
1198                 mutex_enter(&ahci_ctlp->ahcictl_mutex);
1199                 if (ahci_ctlp->ahcictl_flags & AHCI_SUSPEND) {
1200                         mutex_exit(&ahci_ctlp->ahcictl_mutex);
1201                         return (DDI_SUCCESS);
1202                 }

1204                 ahci_ctlp->ahcictl_flags |= AHCI_SUSPEND;

1206                 ahci_em_suspend(ahci_ctlp);

1206                 /* stop the watchdog handler */
1207                 if (ahci_ctlp->ahcictl_timeout_id) {
1208                         (void) untimeout(ahci_ctlp->ahcictl_timeout_id);
1209                         ahci_ctlp->ahcictl_timeout_id = 0;
1210                 }

1212                 mutex_exit(&ahci_ctlp->ahcictl_mutex);

1214                 ahci_em_suspend(ahci_ctlp);

1216                 /*
1217                  * drain the taskq
1218                  */
1219                 ahci_drain_ports_taskq(ahci_ctlp);

1221                 /*
1222                  * Disable the interrupts and stop all the ports.
1223                  */
1224                 ahci_uninitialize_controller(ahci_ctlp);

1226                 return (DDI_SUCCESS);

1228         default:
1229                 return (DDI_FAILURE);
1230         }
1231 }
______unchanged_portion_omitted_
```