CC通过主状态机（如下）处理所有事件

(\*iwu\_glb\_callStateHandlerArray[pCallContext->currentCallState])(pCallContext->dCallId, g\_pCallData, ccCallEvent,pErrorCode)

本地主动发出L3\_DISC\_IND消息给CC，CC经Iwu\_fn\_CcConvertMsgIdToEvent方法将L3\_DISC\_IND转换为Iwu\_en\_L3DisconnectIndication消息进行处理：

|  |
| --- |
| Idle状态  Incoming call proceeding状态  disconnect indication状态  RemoteTerminating状态  InternalTerminatingResp状态  ReleaseRequest状态  IdleTerminatingReq状态  TerminatingQ931Req状态：  无处理 |
| OverlapSending状态：  case Iwu\_en\_L3DisconnectIndication:  {  if (pCallContext->dParsedDigitsData.dmlTransId != 0xffff)  {  iwu\_stop\_MonitorDM(pCallContext->dParsedDigitsData.dmlTransId, &dDmlErr);  pCallContext->dParsedDigitsData.dmlTransId = 0xffff;  }  retVal = Iwu\_fn\_sendReleaseReqMsgToL3(pCallContext);  pCallContext->currentCallState = CcCallState\_RELEASE\_REQUEST;  }  break; |
| Outgoing call proceeding状态  Call deliver状态  active状态  callPresent状态  call received状态  connect request状态：  case Iwu\_en\_L3DisconnectIndication:  {  pCallContext->subStateFlag |= IwuCcsubStateFlag\_ReleaseToBeSent;  }  case Iwu\_en\_L3ReleaseIndication:  {  retVal = iwu\_fn\_handleRemoteTerminateReq(pCallContext, pCallData, pErr);  |--> retVal = iwu\_fn\_sendTermCallReq(pCallContext, causeValue, pErrorCode);  | |--> return iwu\_internal\_msg\_mgr::getInstance()->post\_internal\_message(  Iwu\_en\_callControlFsm,  Iwu\_en\_callmgntFsm,  Iwu\_en\_callmgntTerminateCallReq,  pCallContext->logicalBChannelId,  pCallContext->dCallId,  (iwu\_icf\_msg\_st\*)pTerminateCallRqst,  Iwu\_en\_false);  |-->pCallContext->currentCallState = CcCallState\_REMOTE\_TERMINATING;  |-->retVal = iwu\_fn\_startTimer(dGlbTimerListId,ClearCallRespTimerDuration,  pCallContext->logicalBChannelId,  pCallContext->dCallId,  ICF\_MODULE\_PA,  Iwu\_en\_ClearCallRespTimer,  &(pCallContext->dClearCallRespTimer),  &err);  break;  } |
| TerminatingCmReq状态：  case Iwu\_en\_L3DisconnectIndication:  {  pCallContext->subStateFlag |= IwuCcsubStateFlag\_ReleaseToBeSent;  }  break; |