



=====

Description:
 association state machine init, including state transition and timer init
 All functions run in process context

Parameters:
 S - pointer to the association state machine

Note:
 The state machine looks like the following

	ASSOC_IDLE	ASSOC_WAIT_RSP	REASSOC_WAIT_RSP	DISASSOC_WAIT_RSP
MT2_MLME_ASSOC_REQ	mlme_assoc_req_action	invalid_state_when_assoc	invalid_state_when_assoc	invalid_state_when_assoc
MT2_MLME_REASSOC_REQ	mlme_reassoc_req_action	invalid_state_when_reassoc	invalid_state_when_reassoc	invalid_state_when_reassoc
MT2_MLME_DISASSOC_REQ	mlme_disassoc_req_action	mlme_disassoc_req_action	mlme_disassoc_req_action	mlme_disassoc_req_action
MT2_PEER_DISASSOC_REQ	peer_disassoc_action	peer_disassoc_action	peer_disassoc_action	peer_disassoc_action
MT2_PEER_ASSOC_REQ	drop	drop	drop	drop
MT2_PEER_ASSOC_RSP	drop	peer_assoc_rsp_action	drop	drop
MT2_PEER_REASSOC_REQ	drop	drop	drop	drop
MT2_PEER_REASSOC_RSP	drop	drop	peer_reassoc_rsp_action	drop
MT2_CLS3ERR	cls3err_action	cls3err_action	cls3err_action	cls3err_action
MT2_ASSOC_TIMEOUT	timer_nop	assoc_timeout_action	timer_nop	timer_nop
MT2_REASSOC_TIMEOUT	timer_nop	timer_nop	reassoc_timeout_action	timer_nop
MT2_DISASSOC_TIMEOUT	timer_nop	timer_nop	timer_nop	disassoc_timeout_action

=====

```
VOID AssocStateMachineInit(
    IN  PRTMP_ADAPTER      pAd,
    IN  STATE_MACHINE *S,
    OUT STATE_MACHINE_FUNC Trans[])
{
    StateMachineInit(S, (STATE_MACHINE_FUNC*)Trans, MAX_ASSOC_STATE, MAX_ASSOC_MSG, (STATE_MACHINE_FUNC)Drop, ASSOC_IDLE, ASSOC_MACHINE_BASE);

    // first column
    StateMachineSetAction(S, ASSOC_IDLE, MT2_MLME_ASSOC_REQ, (STATE_MACHINE_FUNC)MlmeAssocReqAction);
    StateMachineSetAction(S, ASSOC_IDLE, MT2_MLME_REASSOC_REQ, (STATE_MACHINE_FUNC)MlmeReassocReqAction);
    StateMachineSetAction(S, ASSOC_IDLE, MT2_MLME_DISASSOC_REQ, (STATE_MACHINE_FUNC)MlmeDisassocReqAction);
    StateMachineSetAction(S, ASSOC_IDLE, MT2_PEER_DISASSOC_REQ, (STATE_MACHINE_FUNC)PeerDisassocAction);

    // second column
    StateMachineSetAction(S, ASSOC_WAIT_RSP, MT2_MLME_ASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenAssoc);
    StateMachineSetAction(S, ASSOC_WAIT_RSP, MT2_MLME_REASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenReassoc);
    StateMachineSetAction(S, ASSOC_WAIT_RSP, MT2_MLME_DISASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenDisassociate);
    StateMachineSetAction(S, ASSOC_WAIT_RSP, MT2_PEER_DISASSOC_REQ, (STATE_MACHINE_FUNC)PeerDisassocAction);
    StateMachineSetAction(S, ASSOC_WAIT_RSP, MT2_PEER_ASSOC_RSP, (STATE_MACHINE_FUNC)PeerAssocRspAction);
    StateMachineSetAction(S, ASSOC_WAIT_RSP, MT2_ASSOC_TIMEOUT, (STATE_MACHINE_FUNC)AssocTimeoutAction);

    // third column
    StateMachineSetAction(S, REASSOC_WAIT_RSP, MT2_MLME_ASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenAssoc);
    StateMachineSetAction(S, REASSOC_WAIT_RSP, MT2_MLME_REASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenReassoc);
    StateMachineSetAction(S, REASSOC_WAIT_RSP, MT2_MLME_DISASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenDisassociate);
    StateMachineSetAction(S, REASSOC_WAIT_RSP, MT2_PEER_DISASSOC_REQ, (STATE_MACHINE_FUNC)PeerDisassocAction);
    StateMachineSetAction(S, REASSOC_WAIT_RSP, MT2_PEER_REASSOC_RSP, (STATE_MACHINE_FUNC)PeerReassocRspAction);
    StateMachineSetAction(S, REASSOC_WAIT_RSP, MT2_REASSOC_TIMEOUT, (STATE_MACHINE_FUNC)ReassocTimeoutAction);

    // fourth column
    StateMachineSetAction(S, DISASSOC_WAIT_RSP, MT2_MLME_ASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenAssoc);
    StateMachineSetAction(S, DISASSOC_WAIT_RSP, MT2_MLME_REASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenReassoc);
    StateMachineSetAction(S, DISASSOC_WAIT_RSP, MT2_MLME_DISASSOC_REQ, (STATE_MACHINE_FUNC)InvalidStateWhenDisassociate);
    StateMachineSetAction(S, DISASSOC_WAIT_RSP, MT2_PEER_DISASSOC_REQ, (STATE_MACHINE_FUNC)PeerDisassocAction);
    StateMachineSetAction(S, DISASSOC_WAIT_RSP, MT2_DISASSOC_TIMEOUT, (STATE_MACHINE_FUNC)DisassocTimeoutAction);

    // timer init
    RTMPInitTimer(pAd, &pAd->MlmeAux.AssocTimer, &AssocTimeout);
    RTMPInitTimer(pAd, &pAd->MlmeAux.ReassocTimer, &ReassocTimeout);
}
```