**Initialization**

1. ‘o\_rx\_bit’ shall initialize to 0 on system startup
2. ‘o\_samp\_tick’ shall initialize to 1 on system startup
3. ‘CAN\_PHY\_TX’ shall initialize to 1 on system startup
4. ‘int\_state’ shall initialize to “sync” on system startup
5. ‘int\_next\_state’ shall initialize to “sync” on system startup
6. Internal signal ‘int\_ts1’ shall initialize to ‘i\_ts1’ + 1 on system startup
7. Internal signal ‘int\_ts2’ shall initialize to i\_ts2 + 1 on system startup

**Reset**

1. ‘o\_rx\_bit’ shall be set to 1 when ‘i\_reset’ is 1
2. ‘o\_samp\_tick’ shall be set to 0 when ‘i\_reset’ is 1
3. ‘CAN\_PHY\_TX’ shall be set to 1 when ‘i\_reset’ is 1
4. ‘int\_state’ shall be set to SYNC when ‘i\_reset’ is 1
5. ‘int\_next\_state’ shall be set to SYNC when ‘i\_reset’ is 1
6. Internal signal ‘int\_ts1’ shall be set to ‘i\_ts1’ + 1 when ‘i\_reset’ is 1
7. Internal signal ‘int\_ts2’ shall be set to ‘i\_ts2’ + 1 when ‘i\_reset’ is 1

**FSM**

1. ‘int\_state’ shall be set to ‘int\_next\_state’ on the rising edge of ‘i\_can\_clk’
2. Internal counter ‘int\_count’ shall be incremented on the rising edge of ‘i\_can\_clk’
3. Internal counter ‘int\_count’ shall be set to 0 when ‘int\_state’ is SYNC
4. ‘int\_next\_state’ shall be set to SYNC when ‘int\_count’ is greater than or equal to ‘int\_ts1’ + ‘int\_ts2’
5. ‘int\_next\_state’ shall be set to TS1 when ‘int\_state’ is SYNC or PROP and ‘int\_count’ is less than ‘int\_ts1’
6. ‘int\_next\_state’ shall be set to TS2 when ‘int\_count’ is less than ‘int\_ts1’ + ‘int\_ts2’

**Synchronization**

1. ‘int\_samp\_tick’ shall be set to 1 when ‘int\_state’ is TS1 and ‘int\_next\_state’ is TS2
2. ‘o\_samp\_tick’ shall be set to ‘int\_samp\_tick’ on the rising edge of ‘i\_can\_clk’
3. ‘CAN\_PHY\_TX’ shall be set to ‘i\_tx\_bit’ when ‘int\_state’ is SYNC
4. ‘o\_rx\_bit’ shall be set to ‘CAN\_PHY\_RX’ when ‘int\_samp\_tick’ is 1
5. Comparison bit ‘int\_rx\_comp’ shall be set to ‘CAN\_PHY\_RX’ on the rising edge of ‘i\_can\_clk’
6. Comparison bit ‘int\_rx\_prev’ shall be set to ‘int\_rx\_comp’ on the rising edge of ‘i\_can\_clk’
7. ‘int\_ts2\_next’ shall be set to ‘int\_ts2’ – ‘i\_sjw’ when ‘int\_rx\_prev’ is not equal to ‘int\_rx\_comp’, and ‘int\_count’ is less than ‘int\_ts1’ – ‘i\_sjw’
8. ‘int\_ts2\_next’ shall be set to ‘int\_ts2’ – ‘int\_count’ when ‘int\_rx\_prev’ is not equal to ‘int\_rx\_comp’, and ‘int\_count’ is less than ‘int\_ts1’, and ‘int\_count’ is greater than or equal to ‘int\_ts1’ – ‘i\_sjw’
9. ‘int\_ts1\_next’ shall be set to ‘int\_count’ when ‘int\_rx\_prev’ is not equal to ‘int\_rx\_comp’, and ‘int\_count’ is greater than or equal to ‘int\_ts1’, and ‘int\_count’ is less than ‘int\_ts1’ + ‘i\_sjw’
10. ‘int\_ts1\_next’ shall be set to ‘int\_ts1’ + ‘i\_sjw’ when ‘int\_rx\_prev’ is not equal to ‘int\_rx\_comp’ and ‘int\_count’ is less than ‘int\_ts1’ + ‘i\_sjw’
11. ‘int\_ts1’ shall be set to ‘i\_ts1’ + 1 when ‘int\_state’ is SYNC and ‘int\_rxbit\_history [5:0]’ is 6’b000000 or 6’b111111
12. ‘int\_ts2’ shall be set to ‘i\_ts2’ + 1 when ‘int\_state’ is SYNC and ‘int\_rxbit\_history [5:0]’ is 6’b000000 or 6’b111111
13. ‘int\_ts1’ shall be set to ‘int\_ts1\_next’ when ‘int\_state’ is SYNC and ‘int\_rxbit\_history [5:0]’ is not equal to 6’b000000 or 6’b111111
14. ‘int\_ts2’ shall be set to ‘int\_ts2\_next’ when ‘int\_state’ is SYNC and ‘int\_rxbit\_history [5:0]’ is not equal to 6’b000000 or 6’b111111
15. ‘int\_rxbit\_history [5:0]’ shall be set to ‘int\_rxbit\_history [4:0]’ & ‘CAN\_PHY\_RX’ when ‘int\_state’ is SYNC