**Created By: Waseem Orphali**

**BIT Stream Processor (BSP):**

1. The module shall be initialized in configuration mode.
2. The module shall enter configuration mode when ‘i\_reset’ is set to 1.
3. The following outputs shall be set to 0 when ‘i\_reset’ is set to 1 or when ‘i\_cen’ is set to 0:
   1. ‘o\_acker’
   2. ‘o\_berr’
   3. ‘o\_ster’
   4. ‘o\_fmer’
   5. ‘o\_crcer’
   6. ‘o\_bbsy’
   7. ‘o\_bidle’
   8. ‘o\_normal’
   9. ‘o\_txok’
   10. ‘o\_arblst’
   11. ‘o\_error’
   12. ‘o\_bsoff’
   13. ‘o\_sleep’
   14. ‘o\_lback’
   15. ‘o\_wakeup’
   16. ‘o\_can\_ready’
4. ‘o\_config’ and ‘o\_tx\_bit’ shall be set to 1 when ‘i\_reset’ is set to 1 or ‘i\_cen’ is set to 0.
5. ‘o\_busy\_can’ shall be set to 1 when ‘i\_reset’ is set to 1.
6. ‘o\_rec’ [7:0] and ‘o\_tec’ [7:0] shall be set to 0 when ‘i\_reset’ is set to 1 or ‘i\_cen’ is set to 0.
7. ‘o\_estat’ [1:0] shall be set to 01 when ‘i\_reset’ is set to 1 or ‘i\_cen’ is set to 0.
8. If ‘i\_cen’ is set to 0 during a transmit operation, then the transmit shall be aborted and retried when ‘i\_cen’ is set to 1 and ‘i\_reset’ is set to 0.
9. If ‘i\_reset’ is set to 0 and ‘i\_cen’ is set to 1 and either ‘o\_busy\_can’ is set to 1 or the module receives data from ‘i\_rx\_bit’, then ‘o\_bbsy’ shall be set to 1.
10. If ‘i\_reset’ is set to 0 and ‘i\_cen’ is set to 1 and ‘o\_bbsy’ is set to 0, then ‘o\_bidle’ shall be set to 1.
11. If ‘i\_reset’ is set to 0, ‘i\_cen’ is set to 1 and ‘i\_lback’ is set to 1, then ‘o\_lback’ shall be set to 1.
12. If ‘i\_reset’ is set to 0, ‘i\_cen’ is set to 1, ‘i\_lback’ is set to 0 and ‘i\_sleep’ is set to 1, then ‘o\_sleep’ shall be set to 1.
13. If ‘i\_reset’ is set to 0, ‘i\_cen’ is set to 1, ‘i\_sleep’ is set to 0 and ‘i\_lback’ is set to 0, then ‘o\_normal’ shall be set to 1.
14. If ‘i\_reset’ is set to 0 and ‘i\_cen’ is set to 1, then the module shall sample serial data coming from ‘i\_rx\_bit’ when ‘i\_samp\_tick’ transitions from 0 to 1.
15. The module shall remove stuffed bits when receiving data.
16. The module shall respond with an acknowledge bit according to CAN protocol when receiving data.
17. If ‘i\_send\_en’ is set to 1 and ‘o\_busy\_can’ is set to 0, then data from ‘i\_send\_data’ [127:0] shall be latched and CRC is calculated for it.
18. If ‘i\_send\_en’ is set to 1 and ‘o\_busy\_can’ is set to 0, then ‘o\_busy\_can’ shall be set to 1 until the data frame is sent successfully.
19. If data is queued for sending and arbitration is lost, ‘o\_tx\_bit’ shall be set to 1.
20. If data is queued for sending and arbitration is lost, ‘o\_arblst’ shall be pulsed for one clock cycle.
21. If data is queued for sending and arbitration is lost, sending shall be retried on the next arbitration period.
22. If data is queued for sending and arbitration is won, the module shall send the data frame according to CAN format.
23. The module shall send the serial data to ‘o\_tx\_bit’ when ‘i\_samp\_tick’ transitions from 0 to 1.
24. If the data frame is sent successfully, then ‘o\_txok’ shall be pulsed for one clock cycle.
25. If a data frame is received successfully, then ‘o\_rx\_message’ [127:0] shall be set to the data frame after removing the CRC, stuffed bits and control bits.
26. If a data frame is received successfully, then ‘o\_can\_ready’ shall be pulsed for one clock cycle.
27. The module shall calculate the CRC for every data frame received from ‘i\_rx\_bit’.
28. If the calculated CRC is not equal to the one received from ‘i\_rx\_bit’ and the module is the transmitter of the data frame, then a transmit error has been detected and ‘o\_crcer’ shall be pulsed for one clock cycle.
29. If the calculated CRC is not equal to the one received from ‘i\_rx\_bit’ and the module is not the transmitter of the data frame, then a receive error has been detected and ‘o\_crcer’ shall be pulsed for one clock cycle.
30. If the module is transmitting data and ‘i\_rx\_bit’ differs from ‘o\_tx\_bit’, then a transmit error has been detected and the ‘o\_berr’ shall be pulsed for one clock cycle.
31. If the module is transmitting data and it doesn’t detect an acknowledge bit, then a transmit error has been detected and ‘o\_acker’ shall be pulsed for one clock cycle.
32. If the module is transmitting data and it doesn’t detect a stuffed bit after five consecutive bits of the same level, then a transmit error has been detected and ‘o\_ster’ shall be pulsed for one clock cycle.
33. If the module is not the transmitter of data and it doesn’t detect a stuffed bit after five consecutive bits of the same level, then a receive error has been detected and ‘o\_ster’ shall be pulsed for one clock cycle.
34. If the module is transmitting data and it detects an error in the form of the data frame according to the CAN protocol, then a transmit error has been detected and ‘o\_fmer’ shall be pulsed for one clock cycle.
35. If the module is not the transmitter of data and it detects an error in the form of the data frame according to the CAN protocol, then a receive error has been detected and ‘o\_fmer’ shall be pulsed for one clock cycle.
36. If the module detects an error and it is in active error state, then an active error frame shall be sent if an error frame is not detected.
37. If the module detects an error and it is in passive error state, then a passive error frame shall be sent.
38. If a transmit error or receive error is detected, ‘o\_error’ shall be pulsed for one clock cycle.
39. If a transmit error has been detected, ‘o\_tec’ shall be incremented by 8.
40. If a receive error has been detected, ‘o\_rec’ shall be incremented by 1.
41. If ‘o\_tec’ or ‘o\_rec’ are larger than 127, then the module shall become in error passive state.
42. If ‘o\_tec’ or ‘o\_rec’ are larger than 127, then ‘o\_estat’[1:0] shall be set to 11.
43. If ‘o\_tec’ or ‘o\_rec’ are larger than 255, then the module shall become in buss off state.
44. If ‘o\_tec’ or ‘o\_rec’ are larger than 255, then ‘o\_estat’[1:0] shall be set to 10.
45. If ‘o\_tec’ or ‘o\_rec’ are larger than 255, then ‘o\_bsoff’ shall be pulsed for one clock cycle.