**Instantiated Modules**

1. The module shall instantiate the microcontroller bus interface module as MCIF
2. The module shall instantiate the configuration registers module as CREG
3. The module shall instantiate a synchronous FIFO module as TX\_FIFO
4. The module shall instantiate a synchronous FIFO module as RX\_FIFO
5. The module shall instantiate the tx priority logic module as TXPL
6. The module shall instantiate the acceptance filter module as AFM
7. The module shall instantiate the bit stream processing module as BSP
8. The module shall instantiate the bit timing module as BTM
9. The module shall instantiate a DCM which generates ‘CAN\_CLK’ of frequency 25Mhz using ‘SYS\_CLK’ as the primary clock source.

**Port Mapping**

1. Top level input ‘SYS\_CLK’ shall be mapped to ‘i\_sys\_clk’ for every submodule
2. DCM generated ‘CAN\_CLK’ shall be mapped to ‘i\_can\_clk’ for TXPL, AFM, BSP, and BTM
3. Top level input ‘CAN\_PHY\_RX’ shall be mapped to ‘CAN\_PHY\_RX’ in BTM
4. Top level output ‘CAN\_PHY\_TX’ shall be mapped to ‘CAN\_PHY\_TX’ in BTM
5. ‘int\_reset’ shall be set to ‘BUS2IP\_RESET’ logic or ‘CREG.o\_soft\_reset’
6. ‘int\_reset’ shall be mapped to ‘i\_reset’ for every submodule
7. Top level input ‘Bus2IP\_Data’ shall be mapped to ‘MCIF.i\_bus\_data’
8. Top level input ‘Bus2IP\_Addr’ shall be mapped to ‘MCIF.i\_addr’
9. Top level input ‘Bus2IP\_CS’ shall be mapped to ‘MCIF.i\_cs’
10. Top level output ‘IP2Bus\_Data’ shall be mapped to ‘MCIF.o\_bus\_data’
11. Top level output ‘IP2Bus\_Ack’ shall be mapped to ‘MCIF.o\_ack’
12. Top level output ‘IP2Bus\_Error’ shall be mapped to ‘MCIF.o\_error’
13. Top level output ‘IP2Bus\_IntrEvent’ shall be mapped to ‘CREG.o\_interrupt’

**Single Read Operation**

1. ‘IP2Bus\_Data’ shall be set to the corresponding data at ‘Bus2IP\_Addr’ when ‘Bus2IP\_CS’ = 1 and ‘Bus2IP\_RNW’ = 1 and on the rising edge of ‘SYS\_CLK’
2. ‘IP2Bus\_Ack’ shall be set after ‘IP2Bus\_Data’ is set from a read operation
3. ‘Bus2IP\_CS’ must be toggled before a consecutive read operation can occur. ‘IP2Bus\_Data’ shall retain its value until ‘Bus2IP\_CS’ or ‘Bus2IP\_RNW’ is toggled.
4. Reserved and write only address reads shall return all 0s on ‘IP2Bus\_Data’

**Single Write Operation**

1. The register corresponding to ‘Bus2IP\_Addr’ shall be set to the value on ‘Bus2IP\_Data’ when ‘Bus2IP\_CS’ = 1 and ‘Bus2IP\_RNW’ = 0 and on the rising edge of ‘SYS\_CLK’
2. IP2Bus\_Ack shall be set on the ‘SYS\_CLK’ rising edge following any write operation for one clock cycle
3. ‘Bus2IP\_CS’ must be toggled before a consecutive write operation can occur. The register data associated with ‘Bus2IP\_Addr’ shall retain its value until ‘Bus2IP\_CS’ or ‘Bus2IP\_RNW’ is toggled.
4. Reserved and read only addresses shall retain their previous value when written to

**Reset**

1. The core shall remain in reset as long as ‘Bus2IP\_Reset is 1
2. The core shall enter configuration mode when ‘Bus2IP\_Reset’ transitions from 1 to 0

*Note: Operational mode transitions verified through configuration registers*

**Interrupts**

1. ‘IP2Bus\_IntrEvent’ is set to 1 when a bit in the ISR = 1 and the corresponding bit in the IER = 1
2. ‘IP2Bus\_IntrEvent’ is set to 0 when clearing a bit in the ISR that is 1. Achieved by writing a 1 to the corresponding bit in the ICR when the corresponding bit in the IER is 1
3. ‘IP2Bus\_IntrEvent’ is set to 0 when a bit in the ISR is 1 and the corresponding bit in the IER is set to 0