MSS R4 Initial Configuration (By Host Application)

* DMM PinMux
* DMM1 reset
* DMM1\_WRITE (0xFCFFF700, 0x00010000)
* DMM1\_WRITE (0xFCFFF700, 0x00000000)

/\* DMM1 Configuration in Functional Mode \*/

* DMM1\_WRITE (0xFCFFF76C, 0x0007FFFF)
* DMM1\_WRITE (0xFCFFF78C, 0x0007FFFF)
* /\* DMM1 Configuration in TRACE-Mode \*/

DMM1\_WRITE (0xFCFFF700, 0x0004000A)

* DEST0REG1 <0xFCFF\_F72C> : 0xFCFF\_xxxx
* DEST0BL1 <0xFCFF\_F730> : 0x9

DMM1 Trace Mode Initial Configuration:

* ADDR:DEST1REG1 -> DATA: 0x5000\_xxxx
* ADDR:DEST1BL1 -> DATA: 0x9
* ADDR:DEST2REG1 -> DATA: 0xFFFF\_xxxx
* ADDR:DEST2BL1 -> DATA: 0x9

Psudo-Code :

1. DMM1\_TRACE\_WRITE (DEST0, 0xFF73C, 0x5000\_xxxx)
2. DMM1\_TRACE\_WRITE (DEST0, 0xFF740, 0x9)
3. DMM1\_TRACE\_WRITE (DEST0, 0xFF74C, 0xFFFF\_xxxx)
4. DMM1\_TRACE\_WRITE (DEST0, 0xFF750, 0x9)

/\* Configuring PING buffer \*/

/\* PING value , 0x420000

PONG value 0x634000 \*/

Current buffer = PING

1. DMM1\_TRACE\_WRITE (DEST1, 0x660, Current buffer)

/\* 2 idle clock cycles would be required \*/

/\* (Set Ping pong interrupt Sel and frame start interrupt sel) \*/

1. DMM1\_TRACE\_WRITE (DEST2, 0xFF954, 0x880)

/\* (Set Frame start interrupt) \*/

1. DMM1\_TRACE\_WRITE (DEST2, 0xFF94C, 0x080)

/\* 2 idle clock cycles would be required \*/

/\* DMM2 reset \*/

1. DMM2\_WRITE (0xFCFFF600, 0x00010000)
2. DMM2\_WRITE (0xFCFFF600, 0x00000000)

/\* DMM2 Configuration in Functional Mode \*/

1. DMM2\_WRITE (0xFCFFF66C, 0x0007FFFF)
2. DMM2\_WRITE (0xFCFFF68C, 0x0007FFFF)

/\* DMM2 Configuration in DDM Mode 32 DDM\_WIDTH assumed \*/

1. DMM2\_WRITE (0xFCFFF600, 0x0004050A)

/\* configure ADC buffer with the fixed block size of 32k for each chirip \*/

* ADDR:DDMDEST -> DATA: 0x5209\_xxxx
* ADDR:DDMBL -> DATA: 0xB

1. DMM1\_TRACE\_WRITE (DEST0, 0xFF61C, 0x52090000)
2. DMM1\_TRACE\_WRITE (DEST0, 0xFF620, 0xB)

/\* Now as there is a transition from Trace to DDM mode, 2 idle clock cycles would be required.

DMM\_MUX (PAD\_BC) should be driven to value “1” to use DMM2 (Direct data Mode).

After this activity transfer data

/\* Now as there is a transition from DDM to TRACE mode, 2 idle clock cycles would be required.

DMM\_MUX (PAD\_BC) should be driven to value “0” to use DMM1 (Trace mode).

/\* If the current buffer = PING next buffer = Pong

Else next buffer =PING

/\* Configuring PONG buffer \*/

1. DMM1\_TRACE\_WRITE (DEST1, 0x660, NEXT Buffer)

Current buffer =Next Buffer

/\* The PING PONG interrupt is set to give the Chirp Available Interrupt by writing to the following register using the DMM1 trace mode with DEST2 to indicate that one chirp data is available for processing \*/

/\* Set Ping pong interrupt Sel \*/

1. DMM1\_TRACE\_WRITE (DEST2, 0xFF954, 0x880)

/\* Set Ping pong interrupt \*/

1. DMM1\_TRACE\_WRITE (DEST2, 0xFF94C, 0x800)

/\* Add inter chip data/delay if needed \*/

/\* At each chirp repeat the above steps from 8 using the DMM interface until the entire frame data is transferred. \*/

/\* After all the chirps have been transferred add inter frame data/delay if needed \*/

/\* And at each frame repeat the above steps from 6 using the DMM interface until the entire data is transferred. \*/