# **Software Flow**

Following are the major HSDC DLL functions involved in Capture and Generation sequence

## ADC Related

### Initialize\_ADC\_Session

This function extracts the ADC INI parameter values and assigns it to DLL flags and global variables.

### Configure\_PLL

This function takes in PLL\_Type (TXPHY/RXPHY/ATXPLL), MIF Command Array, MIF Command Length, Clock Ratio and Reference clock frequencies as arguments, these values gets passed from LabVIEW layer. MIF command array holds the MIF values parsed by the LabVIEW layer from MIF files available in the \\*\Texas Instruments\High Speed Data Converter Pro\14J57revE Details\MIF Files folder of HSDC Pro. They are used to reconfigure JESD PHY IP parameters.

Following are the sequence of operations called by this function

1. Firstly NIOS commands to corresponding NIOS registers are written to assert the transceiver, link and frame clock resets
2. MIF file content passed by LabVIEW layer are stored as 32bit words in a buffer and is written to on-chip MIF BRAM in firmware
3. Configure\_IOPLL\_Counter\_Settings function is called based on PLLType and skip\_reconfig or skip\_pll\_reconfig (added for debug) INI parameters

* This function writes the required M,N and C counter settings to the IOPLL/ATX PLL based on reference clock frequency and clock ratio

1. Once the PLL settings are written, NIOS commands to initiate JESD PHY IP reconfiguration are issued and is skipped if skip\_phy\_reconfig INI parameter is enabled

### Configure\_TSW14J57\_ADC\_Session

This function takes care of configuring RX JESD Base IP related parameters and few other SERDES related parameters: SERDES Polarity Inversion, SYNC Polarity Inversion and Lane Mapping configuration. Following is the sequence of operations involved

1. Function Configure\_RX\_JESD\_Megacore\_IP takes care of configuring JESD Base IP related parameters: LMFSK, Scrambling and Lane Power down settings
2. Following this function call, other SERDES options: SERDES & SYNC Polarity and Lane Mapping are configured for the targeted mode based on INI parameters
3. Once done, the transceiver link and frame resets that are asserted so far are de-asserted with NIOS commands

### Start\_ADC\_to\_DDR

This function initiates capture process, following is the sequence of operations involved

1. Calculate the samples to be captured in terms of 256 bit words from the samples/channel value entered by user in HSDC Pro. This is because FW writes data to memory in terms of 256bit words. DLL takes into account the bit packing pattern or channel pattern mentioned and calculates the required number of samples
2. Set the trigger settings (if trigger mode enabled) to RX\_TRIGGER\_REGISTER (0x400000 + 0x20004). DLL sets bit3 of the trigger register and keeps polling it. FW clears this bit once a trigger is received
3. If it is normal capture, set the capture start bit (Bit0 of the capture register) with required number of samples to RX\_CAPTURE\_REG (0x400000 + 0x20000). If it is triggered capture, write only the samples to be captured as the capture start bit will be set by FW Logic whenever trigger is received
4. With capture start bit set, FW starts capturing the ADC data on JESD link to memory and HSDC DLL keeps polling for the capture done bit (Bit1 of the capture register) which is set by FW once the required number of samples are captured from JESD link

### Read\_DDR\_to\_File

1. In this function, if it is normal capture DLL keeps polling for the DONE bit, bit1 of the RX\_CAPTURE\_REG (0x400000 + 0x20000) until it is set by firmware
2. If it is triggered capture, LabVIEW layer keeps polling for bit3 of the RX\_TRIGGER\_REGISTER to be cleared by firmware, once cleared and capture DONE bit is set by FW, DLL proceeds to read the captured data
3. DLL writes the number of samples to be read from memory (in terms of 32bit words) to FX3 register 0x22010004. DLL sends a pulse to FX3 modules to start reading data from memory- Write 1 to 0x22010000 followed by 0
4. Read\_From\_FPGA\_Queued USB3 API gets called which reads data from FPGA memory through FX3 Synchronous Slave FIFO Interface

## DAC Related

### Initialize\_DAC\_Session

This function extracts the DAC INI parameter values and assigns it to DLL flags and global variables.

### Configure\_PLL

This function is the same as mentioned in ADC section

### Configure\_TSW14J57\_DAC\_Session

This function takes care of configuring TX JESD Base IP related parameters and few other SERDES related parameters: SERDES Polarity Inversion, SYNC Polarity Inversion and Lane Mapping configuration and any other TX related INI parameters.

Following is the sequence of operations involved

1. Function Configure\_TX\_JESD\_Megacore\_IP takes care of configuring JESD Base IP related parameters: LMFSK, Scrambling and Lane Power down settings
2. Following this function call, other SERDES options: SERDES & SYNC Polarity and Lane Mapping are configured for the targeted mode based on INI parameters.
3. Following this, few other TX related INI parameters: TX SYNC Selection to choose either normal SYNC or Alternate SYNC or both and Auto Duplicate Channels are configured in this function. DLL sets the stop bit, bit2 of the TX\_TRANSMIT\_REG (0x0+ 0x20000)to stop the previous Send operation
4. Once done, the transceiver link and frame resets that are asserted so far are de-asserted with NIOS commands

### Start\_DDR\_to\_DAC

This function initiates generation process, following is the sequence of operations involved

1. Calculate the samples to be captured in terms of 256 bit words from the samples/channel value entered by user in HSDC Pro. This is because FW reads data from memory in terms of 256bit words. DLL takes into account the bit packing pattern or channel pattern mentioned and calculates the required number of samples and writes to TX\_TRANSMIT\_REG (0x0+ 0x20000)
2. Set the trigger settings (if trigger mode enabled) to TX\_TRIGGER\_REGISTER (0x0+ 0x20008). DLL sets bit3 of the trigger register and keeps polling it. FW clears this bit once a trigger is received
3. Set the stop bit (bit2 of the transmit register) to stop the previous Send operation
4. DLL calculates the number of samples to be written to memory in terms of 32bit words and writes them to FX3 register 0x22010004. And set bit3 high if transceiver mode is enabled. Based on this bit, DDR4 external memory address is calculated, this is required as the memory is shared for both TX and RX operation
5. DLL sends a pulse to FX3 modules to start writing data to memory- Write 2 to 0x22010000 followed by 0. Data transfer happens through FX3 Synchronous Slave interface

### Write\_DDR\_from\_File

1. In this function, BCM related parameters are calculated and written to FW. BCM Pulse enable or disable bit and BCM Start & Stop sample locations (coerced in terms of 512bit words) are calculated and written to the BCM registers in FW
2. If it is normal Send, set the start bit (Bit0 of the transmit register) & loop bit (bit1 of the transmit register) with required number of samples to TX\_TRANSMIT\_REG (0x0+ 0x20000).

It will initiate the Send process and the DAC data gets played continuously on TX JESD link

1. If it is triggered Send, write only the sample count and the loop bit set as the start bit will be set by FW Logic whenever trigger is received. And HSDC LabVIEW layer waits for the bit3 of the TX trigger register to be cleared (cleared by FW once a trigger pulse is received). If not, ‘No trigger Occurred’ pop up shows up in HSDC Pro
2. With start bit set, FW plays the DAC data continuously on JESD link until the stop bit (bit2 of the transmit register) is set by the DLL during next Send