# **Software Flow**

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

FW uses FX3 I2C interface for Register R/W and 32b parallel fast data transfer (Synchronous Slave FIFO Interface in FX3) for memory read write.

## Configure\_PLL

This function is common to both RX and TX FW functionality. It gets called by LabVIEW first before other functions whenever Capture or Send button is pressed in HSDC Pro SW

It takes in PLL\_Type (TXPMA/RXPMA/CMUPLL), MIF Command Array, MIF Command Length, fabric PLL (fPLL value) value 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\14J56revD Details\MIF Files folder of HSDC Pro.

They are used to reconfigure JESD TX & RX PHY IP & the transmitter PLL (CMU PLL) parameters.

Following are the sequence of operations called by this function

1. Write the fPLL value passed from LabVIEW to the fPLL counters. fPLL is used to generate the parallel clocks named as ‘link clock’ (lanerate/40) in FW used for both TX and RX operation
2. Write 1 to address 0x 801000 + 0x00 to set the fPLL reconfig controller in polling mode. Setting it in polling mode makes the controller use Status register to indicate its busy (reads 0 when busy). If it is in wait request mode, controller will use wait request signal to indicate its busy
3. Write the fPLL value passed from LabVIEW layer to Address 0x14 (0x 801000 + 0x14), to configure C counter settings. There are two counters used, C1 for TX and C2 for RX. fPLL value extracted from the INI file looks like 0x081010 and is passed by the LabVIEW layer.

**INI File: Fabric PLL Counter = 0.611G to 0.7G: 0x081010, 0.7G to 2.4G: 0x080808, 2.4G to 8G:0x080202**

0x**08**\_xx\_xx- indicates the address for C2 counter, and it is 0x04\_xx\_xx for C1 counter

0x08**1010**- indicates the value to be written for C counter. Value is split by half and is written to HIGH Count and LOW Count registers. Sum of both HIGH and LOW counts give the actual C counter setting, in this case of 1010 actual C counter value is 20. Refer the Implementing Fractional PLL Reconfiguration with Altera PLL and Altera PLL Reconfig Megafunctions for more details

Above INI parameter indicates that fPLL counter needs to be configured with different settings for 3 different lane rate ranges.

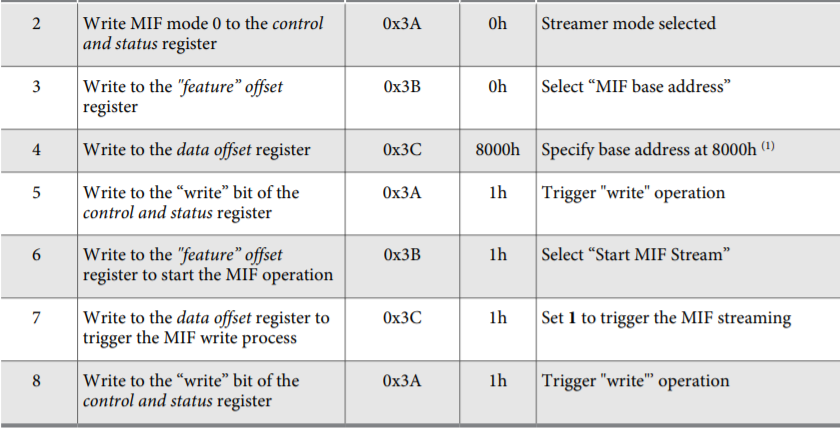
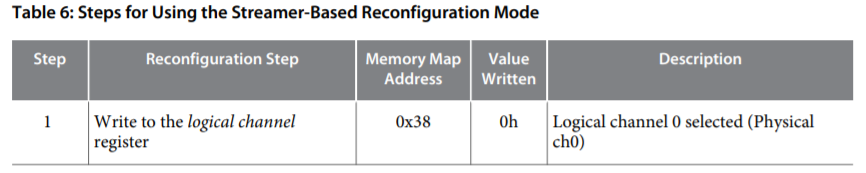
0x081010- Configure C2 with value of 20 and this is only for lane rates between 0.611G to 0.7G.

0x080808- Configure C2 with value of 16 and this is only for lane rates between 0.7G to 2.4G

0x080202- Configure C2 with value of 4 and this is only for lane rates between 2.4G to 8G

1. Write 1 to Start Address 8 (0x 801000 + 0x8) to start the reconfiguration process
2. Keep polling the status register4 (0x 801000 + 0x04) to check if the reconfig controller is busy or finished executing the instruction successfully (reads 1)
3. MIF file content parsed by LabVIEW layer are stored as 32bit words in a buffer and is written to on-chip MIF BRAM in firmware
4. Based on the PLL Type parameter, either TX PMA, RX PMA or CMU PLL the MIF file streaming commands are sent to the transceiver reconfig controller Base address = 0x800400

MIF related register R/Ws happen in Altera\_Streamer\_Based\_Config function as per the Transceiver Reconfig controller registers, please refer "Transceiver Reconfiguratio Controller for Dynamic reconfiguration in Arria V and Cyclone V devices"



Note that QSYS Interconnect used in FW is of byte addressing scheme. So the register addresses from DLL are of byte-addressing and needs to be divided by 4 to indicate word addressing. Online documents from Intel might use word addressing scheme

When configuring RX PMA, repeat these steps 8 times with a different logical channel number each time. Similarly TXPMA and CMUPLL gets configured if they are available in FW. Logical Channel Numbers are mentioned below

|  |  |  |
| --- | --- | --- |
| TX | RX | CMUPLL |
| 8 | 0 | 12 |
| 9 | 1 | 13 |
| 10 | 2 | 14 |
| 11 | 3 | 15 |
| 16 | 4 | 20 |
| 17 | 5 | 21 |
| 18 | 6 | 22 |
| 19 | 7 | 23 |

## ADC Related

### Initialize\_ADC\_Session

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

### Configure\_TSW14J56\_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. In this function SERDES options: SERDES & SYNC Polarity and Lane Mapping are configured for the targeted mode based on INI parameters
2. Following it Function Configure\_RX\_JESD\_Megacore\_IP gets called, it takes care of configuring JESD Base IP related parameters: LMFSK, Scrambling and Lane Power down settings. Base IP link layer is put in reset at the start by issuing Frame soft reset (write 1 to 0x440000+0x2c000) and is de-asserted at the end after configuring IP parameters

### 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. DLL takes into account the bit packing pattern or channel pattern mentioned and calculates the required number of samples
2. If trigger mode enabled, set the trigger settings 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). Number of samples is written to bits [31:4] of RX\_CAPTURE\_REG. 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. FW has a register to indicate if the memory uses 32b fast parallel (FX3 Synchronous Slave FIFO Interface) or I2C interface for data transfer

* If fast parallel interface, 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
* Data transfer happens through bulk transfer and Read\_From\_FPGA\_Queued USB3 API gets called which reads data from FPGA through FX3 Synchronous Slave FIFO Interface

1. If it is I2C interface, data transfer happens through FX3 control end point and Write\_U32\_register is called for multiple iterations

## DAC Related

### Initialize\_DAC\_Session

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

### Configure\_TSW14J56\_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 Base IP link layer is put in reset at the start by issuing Frame soft reset (write 1 to 0x440000+0x2c000) and is de-asserted at the end after configuring IP parameters.
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: Auto Duplicate Channels are configured in this function.
4. DLL sets the stop bit, bit2 of the TX\_TRANSMIT\_REG (0x0+ 0x20000)to stop the previous Send operation

### 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. FW has a register to indicate if the memory uses 32b fast parallel (FX3 Synchronous Slave FIFO Interface) or I2C interface for data transfer

* If fast parallel interface, 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

### Write\_DDR\_from\_File

1. If it is 32b fast parallel interface, data transfer happens through bulk transfer and Write\_To\_FPGA USB3 API gets called which writes data to FPGA through FX3 Synchronous Slave FIFO Interface
2. If it is I2C interface, data transfer happens through FX3 control end point and Write\_U32\_register is called for multiple iterations
3. 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