8.3 Initialization Setup

1. Power up the FPGA/ASIC and start the JESD204B transmitter system.
2. Set TXENABLE pin to logic LOW. This set the DAC3XJ8X output to midscale and prevents any potential output glitches during the power up process from propagating through the rest of the signal chain.
3. Set SLEEP pin to logic LOW.

Note: by default, config35 sleep\_cntl bits are set to 0xFFFF. This allows the SLEEP pin state to be routed directly to various DAC3XJ8X circuits to control the power down mode of the circuits directly. Setting SLEEP pin to logic LOW ensures that all DAC3XJ8X circuits start up correctly upon power up.

1. Supply all 0.9V supplies (VDDDIG, VDDT, VDDDAC, VDDCLK), all 1.8V supplies (VDDR, VDDS, VQPS, VDDIO, VDDAPLL, VDDAREF), and all 3.3V supplies (VDDADAC). The supplies can be powered up simultaneously or in any order. There are no specific requirements on the ramp rate for the supplies.
2. RESET the JTAG port by either toggling TRSTB low if using the JTAG port or holding TRSTB low if not using JTAG. Failure to do so may cause the device to operate incorrectly.
3. Provide the DACCLK to the device. This will be the sampling clock after interpolation if on-chip PLL is used, or this can be the reference clock to the on-chip PLL.
4. Toggle RESETB to logic LOW to reset the SIF registers. This step ensures all the fuse logics are reset correctly. SIF RESET (register software reset) cannot take place of the RESETB pin at this step. Failure to do so may cause the device to operate incorrectly.
5. SYSREF, Clock Divider Initializer, and JESD204B Initializer State (before completing the device configuration)

The goal of this step is to set the DAC3XJ8X clock divider and JESD204B logic to stand-by mode and gate any initialization activities until all internal clocks, JESD204B, and SERDES blocks are programmed.

* 1. For system implementing continuous SYSREF:
     1. Disarm the clock divider initialization circuit: cdrvser\_sysref\_mode = 2b’000 in config36 (0x24).
     2. Disarm the JESD204B link0 initialization circuit: sysref\_mode\_link0 = 2b’000 in config92 (0x5C).
     3. Disarm the JESD204B link1 (if used in dual link setup) initialization circuit: sysref\_mode\_link1 = 2b’000 in config92 (0x5C).

Note: both the clock divider and JESD204B link0/link1 initialization circuits are not armed so the SYSREF signal will not start the circuit initialization. They will be armed again later after all the internal clocks, JESD204B, and SERDES blocks are programmed to ensure proper startup sequencing.

* + 1. If continuous SYSREF is implemented in the system, the SYSREF can be provided to the device at this point.
  1. For system implementing pulsed SYSREF:
     1. Arm the clock divider initialization circuit to the desired triggering mode. For instance, set cdrvser\_sysref\_mode = 2b’011 in config36 (0x24) to set the initialization logic to skip one SYSREF pulse and then use only the next pulse.
     2. Arm the JESD204B link0 initialization circuit to the desired triggering mode. For instance, set sysref\_mode\_link0 = 2b’101 in config92 (0x5C) to set the initialization logic to skip two SYSREF pulses and then use only the next pulse.
     3. Program sysref\_mode\_link1 = 2b’101 in config92 (0x5C) if used.
     4. Hold off the triggering of pulsed SYSREF until the DAC3XJ8X device is near the completion of programming.
  2. Ensure the JESD204B block is at initialization and reset state: init\_state = 2b’1111 and JESD\_reset\_n = 2b’0 in config74 (0x4A). (Note: this is the default register setting).

1. Write the DAC3XJ8X registers. The registers can be written in any order depending on the circuit blocks being used. The following steps provide general guidelines for programming the DAC clocks, SERDES, and JESD204B logics. Clear and check the associated alarms for each circuit to ensure the correct programming.
   1. Program the DAC PLL settings in config26 (0x1A), config49 (0x31), config50 (0x32), and config51 (0x33).
      1. If the PLL is not used, set pll\_sleep and pll\_reset to “1” and pll\_ena to 2b’0.
      2. If the PLL is used, check config49 (0x31) for pll\_lpvolt(2:0) register and config108 for alarm\_from\_pll register for PLL lock status.
   2. Program the SERDES settings in config59 (0x3B the serdes\_clk\_sel and serdes\_refclk\_div) to config63 (0x3F)
   3. Program the JESD204B204B/SERDES lane settings in config63 (0x3F), config70 (0x46) to config74 (0x4A), and config96 (0x60).
   4. Verify the SERDES PLL lock status by checking the SERDES PLL alarms: alarm\_rw0\_pll (alarm for lanes 0 through 3) and alarm\_rw1\_pll (alarm for lanes 4 through 7).
   5. Program the JESD204B settings in config3 (0x03), config37 (0x25), config74 (0x4A) to config97 (0x61).
   6. Program the DSP block settings (NCO, PA protection, QMC, fractional delay, etc.) and enable the blocks through config0 (0x00) to config2 (0x02) and program respective filter coefficients from config 8 (0x08) to config25 (0x19). DSP block initializers are set from config30 (0x1E) to config32 (0x20). Preparing the DAC3XJ8X clock divider and JESD204B logic for initialization through continuous SYSREF
2. SYSREF, Clock Divider Initializer, and JESD204B Initializer State (preparing for the device initialization)

The goal of this step is to rearm the clock divider initializer and JESD204B initializer to prepare for SYSREF edges.

* 1. For continuous SYSREF, rearm the clock divider and JESD204B204B initializer settings:
     1. Arm the clock divider initialization circuit to the desired triggering mode. For instance, set cdrvser\_sysref\_mode = 2b’011 in config36 (0x24) to set the initialization logic to skip one SYSREF pulse and then use only the next pulse.
     2. Arm the JESD204B link0 initialization circuit to the desired triggering mode. For instance, set sysref\_mode\_link0 = 2b’101 in config92 (0x5C) to set the initialization logic to skip two SYSREF pulses and then use only the next pulse. Apply the same to link1 initialization circuit if used.

Note: at this point, the clock divider should be initialized, but the JESD204B block will not be initialized after JESD204B block has exit out of reset and initialization state. This is acceptable as long as the continuous SYSREF has deterministic edges. The JESD204B block will be initialized with deterministic SYSREF edges at later step and will have synchronous relationship with the clock divider.

* 1. For pulsed SYSREF, this step has been performed. Hold off on triggering SYSREF at this point.

1. Set the JESD204B block to exit out of the reset state: JESD\_reset\_n = 2b’1 in config74 (0x4A).
2. Set the JESD204B block to exit out of the initialization state: init\_state = 2b’0000 in config74 (0x4A).

Note: TI recommends programming step 12 and 13 separately.

1. SYSREF State (Initialize the device)

The goal of this step is to initialize the DAC3XJ8X clock divider and JESD204B logic through SYSREF edges. This will ensure all the internal logics of DAC3xJ8x are running correctly.

* 1. For continuous SYSREF, initialization is completed. Go to the next step.
  2. For pulsed SYSREF, apply the pulsed SYSREF to the DAC3XJ8X at this point. Initialization is now completed.

1. SYSREF State (after device initialization is completed)
   1. For continuous SYSREF, the SYSREF can be disabled at this point to prevent any coupling of SYSREF signal onto the DAC output.

Note: If the continuous SYSREF signal will be turned off to avoid potential coupling issue, system design should be aware of potential transient that may impact SYSREF edges.

* 1. For pulsed SYSREF, triggering of SYSREF should be completed at this point.

Note: pulsed SYSREF will require AC coupling with SYSREF driver having stable common mode voltage before and after switching or DC coupling network. This ensures stable common mode before and after switching.