

Revised PLL programming sequence for all Keystone I devices

Devices: KeyStone I family of devices

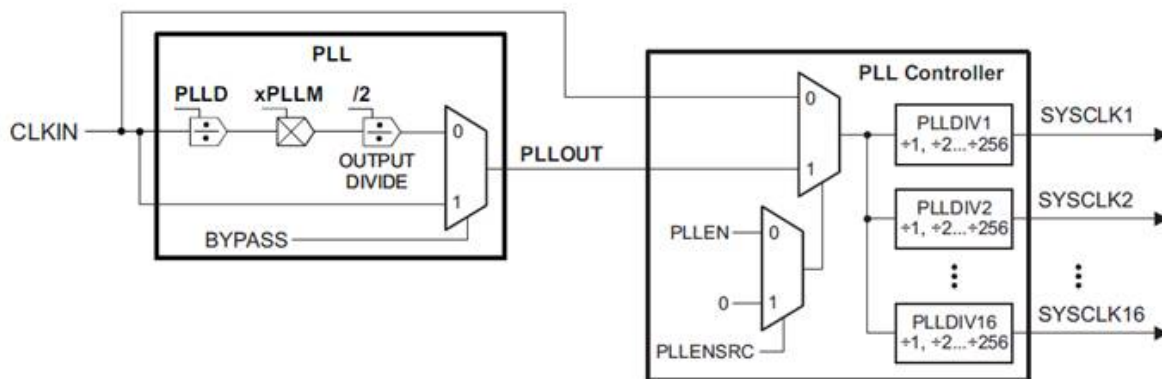
Revisions: All revisions

Detailed Description: It has been observed that on a few Keystone I devices, the CorePacs locks-up after reprogramming of the Core PLL. It has been identified that the incorrect PLL programming sequence was causing the Corepacs to lock-up.

Workaround: TI has revised the PLL programming sequence for Main PLL, DDR PLL and PASS PLL to eliminate the possibility of this lock-up issue. The revised sequence enables the by-pass mode in the PLL Controller via a MUX (by clearing PLEN and PLENSRC bits) when the Main PLL is being re-programmed. Also, the PLLM, PLLD and BWADJ fields are programmed prior to assertion of PLLRST signal for all three PLLs in the revised sequence.

Please use the revised Main PLL, DDR PLL and PASS PLL programming sequence as described below. We plan to include this revised sequence in the next release of Phase Locked Loop (PLL) User Guide (sprugv2) document on ti.com.

Keystone Main PLL and PLL controller Block Diagram



$$\text{PLL output clock (PLLOUT)} = \text{CLKIN} \times (\text{PLLM}+1) \div (2 \times (\text{PLLD}+1))$$

Main PLL initialization sequence

NOTE: Chip-level registers located in the Bootcfg space are write-protected at power-up. Software must first un-lock the KICK0 and KICK1 registers prior to writing to any chip-level registers. Refer to the Keystone data manual for the KICK register un-locking procedure.

Note: Unless specifically mentioned, move to the next following step in the below mentioned initialization sequence

Step 1. If executing this sequence immediately after device power-up, you must allow for the PLL to become stable. PLL stabilization time is 100 μ s.

Step 2. Check the status of BYPASS bit in SECCTL register, execute following steps if BYPASS == 1 (if bypass enabled), if BYPASS==0 then **Jump to Step 3**

- a. In MAINPLLCTL1, write ENSAT = 1 (for optimal PLL operation)
- b. In PLLCTL, write PLEN = 0 (bypass enabled in PLL controller mux)
- c. In PLLCTL, write PLENSRC = 0 (enable PLEN to control PLL controller mux)
- d. Wait 4 cycles of the reference clock (CLKIN) (to make sure the PLL controller mux switches properly to the bypass)
- e. In SECCTL, write BYPASS = 1 (bypass enabled in PLL mux)
- f. In PLLCTL, write PLLPWRDN = 1 (power down the PLL)
- g. Wait for at least 5 μ s based on the reference clock (CLKIN) (PLL power down toggling time)
- h. In PLLCTL, write PLLPWRDN = 0 (power up the PLL) **Jump to Step 4**

Step 3. Enable BYPASS in the PLL Controller

- a. In PLLCTL, write PLEN = 0 (bypass enabled in PLL controller mux)
- b. In PLLCTL, write PLENSRC = 0 (enable PLEN to control PLL controller mux)
- c. Wait 4 cycles of the reference clock (CLKIN) (to make sure the PLL controller mux switches properly to the bypass)

Step 4. PLLM is split in two different registers. Program PLLM[5:0] in PLL multiplier control register (PLLM) and PLLM[12:6] in MAINPLLCTL0 register

Step 5. BWADJ is split in two different registers. Program BWADJ[7:0] in MAINPLLCTL0 and BWADJ[11:8] in MAINPLLCTL1 register. BWADJ value must be set to $((\text{PLLM} + 1) \gg 1) - 1$

Step 6. Program PLLD in MAINPLLCTL0 register

Step 7. In SECCTL, write OD (Output Divide) = 1 (that is divide-by-2)

Step 8. This step is needed only if default ratio values in PLLDIV n registers (it controls SYSCLK n clock) need to be changed. Otherwise skip this step. Note that only some of the SYSCLKs are software programmable, refer to the respective KeyStone data manual to see which SYSCLKs are programmable:

- a. Check that the GOSTAT bit in PLLSTAT is cleared to show that no GO operation is currently in progress.
- b. Program the RATIO field in PLLDIV n to the desired new divide-down rate. If the RATIO field changed, the PLL controller will flag the change in the corresponding bit of DCHANGE.
- c. Set the respective ALN n bits in ALNCTL to align any SYSCLKs after the GO operation.
- d. Set the GOSET bit in PLLCMD to initiate the GO operation to change the divide values and align the SYSCLKs as programmed.
- e. Read the GOSTAT bit in PLLSTAT to make sure the bit returns to 0 to indicate that the GO operation has completed.

Step 9. In PLLCTL, write PLLRST = 1 (PLL reset is asserted)

Step 10. Wait for at least 7 μ s based on the reference clock (CLKIN) (PLL reset time)

Step 11. In PLLCTL, write PLLRST = 0 (PLL reset is de-asserted)

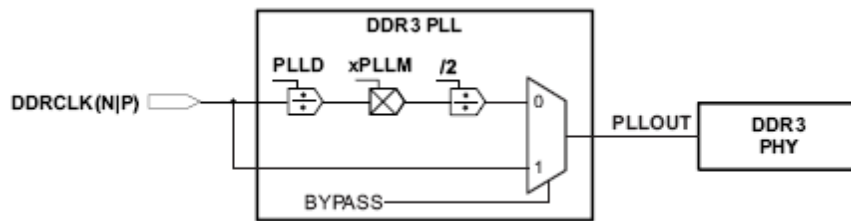
Step 12. Wait for at least 500 * CLKIN cycles * (PLLD + 1) (PLL lock time)

Step 13. In SECCTL, write BYPASS = 0 (enable PLL mux to switch to PLL mode)

Step 14. In PLLCTL, write PLEN = 1 (enable PLL controller mux to switch to PLL mode)

Step 15. Main PLL is now initialized

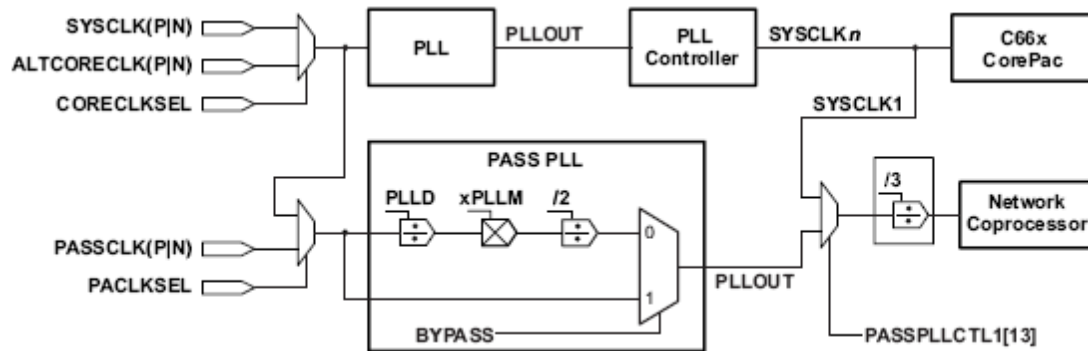
DDR3 PLL initialization sequence



Main PLL sequence must always be executed prior to the DDR3 PLL sequence shown below.

- Step 1.** In DDR3PLLCTL1, write ENSAT = 1 (for optimal PLL operation)
- Step 2.** In DDR3PLLCTL0, write BYPASS = 1 (set the PLL in Bypass)
- Step 3.** Program PLLM and PLLD in DDR3PLLCTL0 register
- Step 4.** Program BWADJ[7:0] in DDR3PLLCTL0 and BWADJ[11:8] in DDR3PLLCTL1 register. BWADJ value must be set to $((\text{PLLM} + 1) \gg 1) - 1$
- Step 5.** In DDR3PLLCTL1, write PLLRST = 1 (PLL is reset)
- Step 6.** Wait for at least 7 us based on the reference clock (PLL reset time)
- Step 7.** In DDR3PLLCTL1, write PLLRST = 0 (PLL reset is released)
- Step 8.** Wait for at least $500 * \text{REFCLK cycles} * (\text{PLLD} + 1)$ (PLL lock time)
- Step 9.** In DDR3PLLCTL0, write BYPASS = 0 (switch to PLL mode)
- Step 10.** DDR PLL is now initialized

PA_SS PLL initialization sequence



Main PLL sequence must always be executed prior to the PA_SS PLL sequence shown below.

- Step 1.** In PASSPLLCTL1, write ENSAT = 1 (for optimal PLL operation)
- Step 2.** In PASSPLLCTL0, write BYPASS = 1 (set the PLL in Bypass)
- Step 3.** Program PLLM and PLLD in PASSPLLCTL0 register
- Step 4.** Program BWADJ[7:0] in PASSPLLCTL0 and BWADJ[11:8] in PASSPLLCTL1 register. BWADJ value must be set to $((\text{PLLM} + 1) \gg 1) - 1$
- Step 5.** In PASSPLLCTL1, write PLLRST = 1 (PLL is reset)
- Step 6.** Wait for at least 7 us based on the reference clock (PLL reset time)
- Step 7.** In PASSPLLCTL1, write PLLSELECT = 1 (for selecting the output of PA PLL as the input to PASS)
- Step 8.** In PASSPLLCTL1, write PLLRST = 0 (PLL reset is released)
- Step 9.** Wait for at least $500 * \text{REFCLK cycles} * (\text{PLLD} + 1)$ (PLL lock time)
- Step 10.** In PASSPLLCTL0, write BYPASS = 0 (switch to PLL mode)
- Step 11.** PASS PLL is now initialized