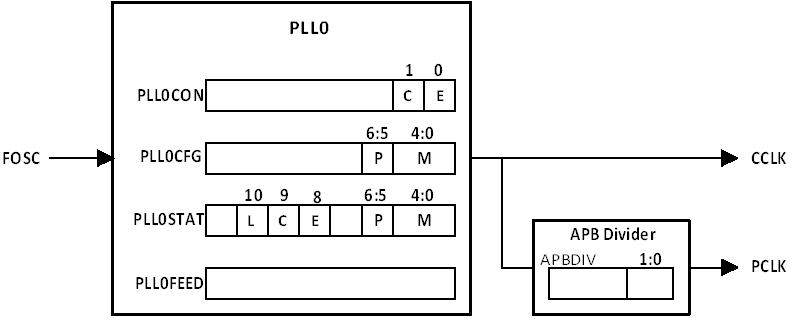
-Phase Locked Loop (PLL)

* LPC214x has two PLLs: PLL0 and PLL1
* PLL0 is used to generate clock for CPU (CCLK) & peripherals (PCLK)
* PLL1 is used for USB
* It uses crystal oscillator input frequency (FOSC) 10–25 MHz and generates output up to 60 MHz
* Parameters m & p determine output frequency, CCLK = m x FOSC
* In the feedback path of PLL there is current controlled osc (CCO) whose frequency should be kept within 156 – 320 MHz.

Computing m and p values for generating **CCLK of 60 MHz from FOSC 12 MHz**

* **m**= CCLK/FOSC = 60/12 = 5
* If **p** is chosen to be 2, CCO (= 2 x p x 60) is within 156 – 320 MHz.



#### PLL Registers

* PLLxCON – Controls PLL activation; used to enable and connect PLL
* PLLxCFG – Multiplier value (bits [4:0]) and divider value (bits [6:5] are configured
* PLLxSTAT – Shows current enable & connect status, m & p values, lock status
* PLLxFEED – Writing feed sequence, i.e. ‘0xAA’ and ‘0x55’ in sequence, to this register– It enables changes made to PLLCON, PLLCFG registers to take effect APBDIV – Sets PCLK freq to 1/4, same or 1/2 of CCLK (by writing 0x00, 0x01, 0x02)

#### PLL configuration sequence

1. Determine m and p values for desired CCLK and PCLK from FOSC
2. Enter (m-1) in bits [4:0] and enter 00/01/10/11 for p values 1/2/4/8 in bits [6:5] of PLLCFG register.
3. Enable PLL using PLLCON register (by setting bit[0]).
4. Issue feed sequence, i.e. write constants 0xAA, 0x55 into PLLFEED reg
5. Wait for PLL to lock (by checking bit[10],
6. Connect & enable PLL using PLLCON reg. (by setting bits[1:0])

Issue again feed sequence **Examples**

* Configuring PLL0 for generating 60 MHz CCLK from 12 MHz FOSC multiplier = CCLK / FOSC = 60 / 12 = 5; (m-1) = 4 is placed in bits [4:0]
* ‘p’  is chosen as 2, since CCO (= 2 x p x 60) is to be within 156 – 300 MHz

**Enabling but not connecting PLL0, enabling and connecting:**

* Bit [0] of PLL0CON register is set for only enabling and bits[1:0] are set for enabling and connecting.
* PLL0CON    =    0x01;    // enable PLL but not connect
* PLL0CON    =    0x03;    // enable PLL and connect

**Issue of PLL feed sequence:**

* PLL0FEED    =    0xAA;    // first fixed value
* PLL0FEED    =    0x55;    // second fixed value

Checking PLL0 lock status & waiting till PLL0 locks onto the target frequency:

PLOCK, the bit [10] of PLL0STAT register, is polled and checked for lock status.

PLOCK = 0x40;

**while(!(PLLSTAT & PLOCK));  // poll lock status & wait till PLL0 locks**

### VPB Divider

**Programming APB divider for desired PCLK frequency:**

Value 0x00, 0x01 or 0x02 in APBDIV register sets the PCLK frequency to one-fourth, same as or half of CCLK frequency.

APBDIV = 0x02    // set PCLK frequency to (CCLK/2)