EE 311 **December 19, 2017**
Phase Lock Loop Setup Notes

Phase lock loop set up for the ARM Nucleo Board with the STM32F446 processor.

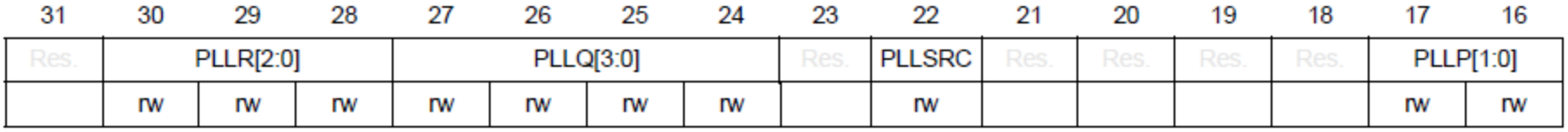
```
//This function resets the system clock to 180 MHz.
void InitializeClock()
   {RCC_CFGR = 0x00000000;         //Reset Clock Configuration Register
    RCC_CR &= 0xFEF6FFFF;          //Reset HSEON, CSSON and PLLON Bits
    RCC_CR |= (1 << 16);           //Turn on HSE clock
    while((RCC_CR & (1 << 17)) == 0); //Wait until HSE is ready
    RCC_CR |= (1 << 19);
    RCC_PLLCFGR = 0x27405A08;      //Set PLLP = 0, PLLN = 360, PLLM = 8,
    FLASH_ACR &= 0xFFFFFFF8;       //Set flash wait states to 5
    FLASH_ACR |= 0x5;
    RCC_CR |= (1 << 24);           //Enable PLL on
    while((RCC_CR & (1 << 25)) == 0); //Wait for PLL to lock on
    RCC_CFGR = 0x9402;             // APB2/2, APB1/4, AHB/1
   }
```

Notes:

1. The two lines which set the Flash wait states to five can go last and this works for most boards. However, on some boards putting these lines last causes the system to fail. I think this is on the boards that are labeled as C-01 and possibly C-02. All boards appear to work OK if the Flash wait states are set before the PLL is renabled.

2. The RCC_PLLCFGR register has a number of reserved bits that should not be changed. However these bits all reset to zero so using `RCC_PLLCFGR = 0x27405A08;` keeps these bits at zero and it is not necessary to use AND and OR operators to preserve the state of the reserve bits.

3. The clock circuitry takes in an 8 MHz signal from the crystal oscillator and divides it by 8 to get 1 MHz. The PLL multiplies this by 180 to get 180 MHz internal clock that is crystal stabilized.



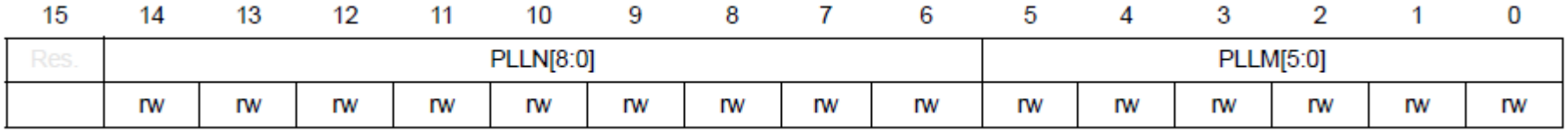
NC	0	1	0	0	1	1	1	NC	1	NC	NC	NC	NC	0	0

AND MASK

1	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0
8				0				B				C			

OR MASK

0	0	1	0	0	1	1	1	0	1	0	0	0	0	0	0
2				7				4				0			



NC	1	0	1	1	0	1	0	0	0	0	0	1	0	0	0

AND MASK

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8				0				0				0			

OR MASK

0	1	0	1	1	0	1	0	0	0	0	0	1	0	0	0
5				A				0				8			

Use the following so as NOT to alter any of the reserved bits:

```
RCC_PLLCFGR &= 0x80BC8000;
RCC_PLLCFGR |= 0x27405A08;

Reset value of RCC_PLLCFGR = 0x2400 3010 = 0010 0100 0000 0000 0011 0000 0001 0000 so all of the reserve bits
reset to 0.  You can therefore use
RCC_PLLCFGR = 0x27405A08;
```

Bit 31 Reserved, must be kept at reset value.

Bits 30:28 **PLLR[2:0]:** Main PLL division factor for I2Ss, SAIs, SYSTEM and SPDIF-Rx clocks

Set and cleared by software to control the frequency of the clock. These bits should be written only if PLL is disabled.

Clock frequency = VCO frequency / PLLR with 2 ≤ PLLR ≤ 7

000: PLLR = 0, wrong configuration

001: PLLR = 1, wrong configuration

010: PLLR = 2

011: PLLR = 3

...

111: PLLR = 7

DO NOT CHANGE

Bits 27:24 **PLLQ[3:0]:** Main PLL (PLL) division factor for USB OTG FS, SDIOclocks

Set and cleared by software to control the frequency of USB OTG FS clock and the SDIOclock. These bits should be written only if PLL is disabled.

Caution: The USB OTG FS requires a 48 MHz clock to work correctly. The SDIOneeds a frequency lower than or equal to 48 MHz to work correctly.

USB OTG FS clock frequency = VCO frequency / PLLQ with 2 ≤PLLQ ≤15

0000: PLLQ = 0, wrong configuration

0001: PLLQ = 1, wrong configuration

0010: PLLQ = 2

0011: PLLQ = 3

0100: PLLQ = 4

...

1111: PLLQ = 15

Bit 23 Reserved, must be kept at reset value.

DO NOT CHANGE

Bit 22 **PLLSRC:** Main PLL(PLL) and audio PLL (PLLI2S) entry clock source

Set and cleared by software to select PLL and PLLI2S clock source. This bit can be written only when PLL and PLLI2S are disabled.

0: HSI clock selected as PLL and PLLI2S clock entry

1: HSE oscillator clock selected as PLL and PLLI2S clock entry

Should be PLLSRC = 1

Bits 21:18 Reserved, must be kept at reset value.

Bits 17:16 **PLLP[1:0]:** Main PLL (PLL) division factor for main system clock

Set and cleared by software to control the frequency of the general PLL output clock. These bits can be written only if PLL is disabled.

Caution: The software has to set these bits correctly not to exceed 180 MHz on this domain.

PLL output clock frequency = VCO frequency / PLLP with PLLP = 2, 4, 6, or 8

00: PLLP = 2
01: PLLP = 4
10: PLLP = 6
11: PLLP = 8

Should be PLLP = 2 -> 00

Bits 14:6 **PLLN[8:0]:** Main PLL (PLL) multiplication factor for VCO

Set and cleared by software to control the multiplication factor of the VCO. These bits can be written only when PLL is disabled. Only half-word and word accesses are allowed to write these bits.

Caution: The software has to set these bits correctly to ensure that the VCO output frequency is between 100 and 432 MHz.

VCO output frequency = VCO input frequency × PLLN with 50 ≤ PLLN ≤ 432

000000000: PLLN = 0, wrong configuration
000000001: PLLN = 1, wrong configuration ...
000110010: PLLN = 50
...
001100011: PLLN = 99
001100100: PLLN = 100
...
110110000: PLLN = 432
110110001: PLLN = 433, wrong configuration ...
111111111: PLLN = 511, wrong configuration

Note: Between 50 and 99 multiplication factors are possible for VCO input frequency higher than 1 MHz. However care must be taken to fulfill the minimum VCO output frequency as specified above.

Should be PLLN = 360_{10} = 168_{16} = 1 0110 1000

Bits 5:0 **PLLM[5:0]:** Division factor for the main PLL (PLL) input clock

Set and cleared by software to divide the PLL input clock before the VCO. These bits can be written only when the PLL is disabled.

Caution: The software has to set these bits correctly to ensure that the VCO input frequency ranges from 1 to 2 MHz. It is recommended to select a frequency of 2 MHz to limit PLL jitter.

VCO input frequency = PLL input clock frequency / PLLM with 2 ≤ PLLM ≤ 63

000000: PLLM = 0, wrong configuration
000001: PLLM = 1, wrong configuration
000010: PLLM = 2
000011: PLLM = 3
000100: PLLM = 4
...
111110: PLLM = 62
111111: PLLM = 63

Should be PLLM = 8 = 001000