Configure SystemInit Fuction

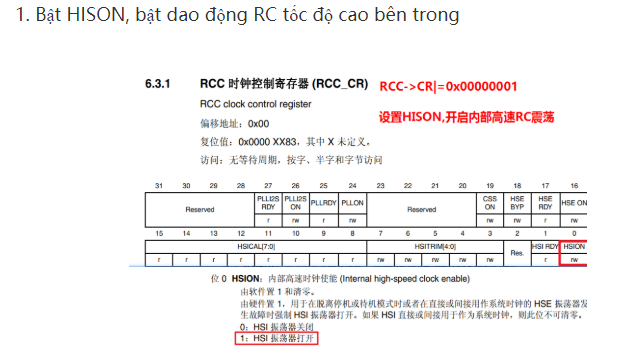
#if (\_\_FPU\_PRESENT == 1) && (\_\_FPU\_USED == 1)

SCB->CPACR |= ((3UL << 10\*2)|(3UL << 11\*2)); /\* set CP10 and CP11 Full Access \*/

#endif

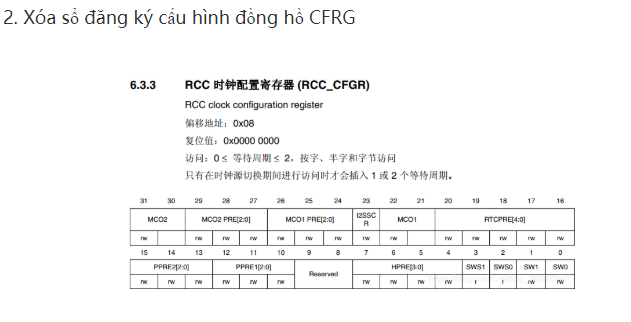
/\* Reset the RCC clock configuration to the default reset state ------------\*/

/\* Set HSION bit \*/



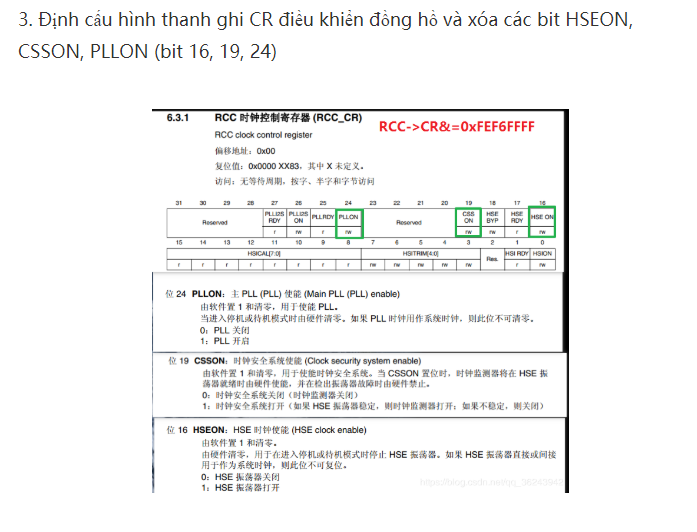
RCC->CR |= (uint32\_t)0x00000001;

/\* Reset CFGR register \*/



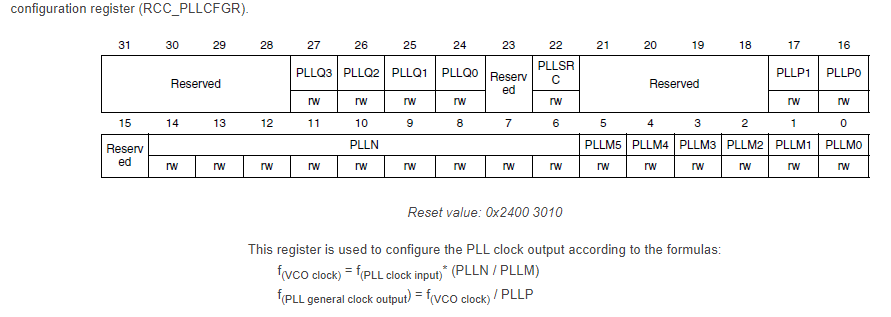
RCC->CFGR = 0x00000000;

/\* Reset HSEON, CSSON and PLLON bits \*/



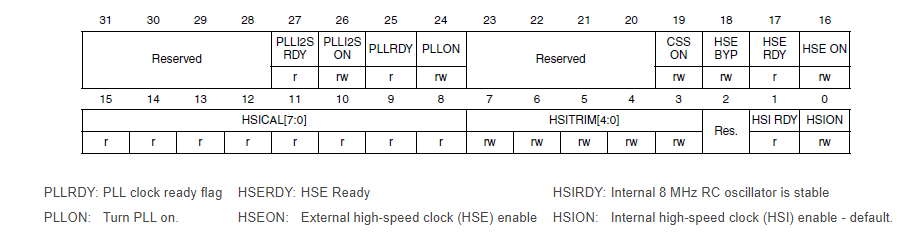
RCC->CR &= (uint32\_t)0xFEF6FFFF;

/\* Reset PLLCFGR register \*/



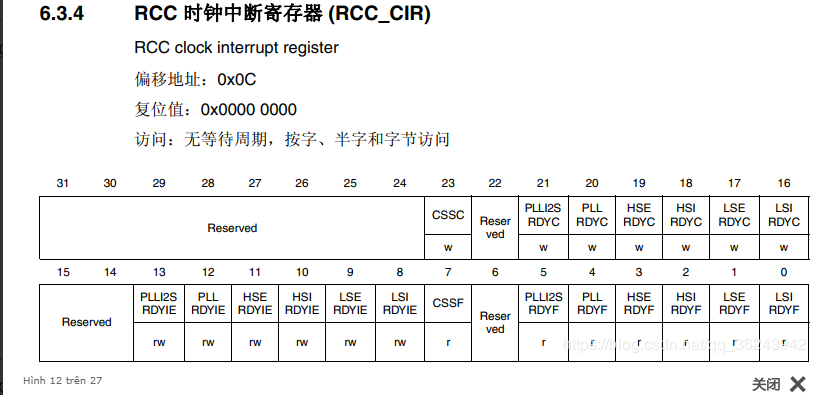
RCC->PLLCFGR = 0x24003010;

/\* Reset HSEBYP bit \*/



RCC->CR &= (uint32\_t)0xFFFBFFFF;

/\* Disable all interrupts \*/



RCC->CIR = 0x00000000;

#ifdef DATA\_IN\_ExtSRAM

SystemInit\_ExtMemCtl();

#endif /\* DATA\_IN\_ExtSRAM \*/

SetSysClock();

Configure SetSysClock Fuction

/\* Enable HSE \*/

RCC->CR |= ((uint32\_t)RCC\_CR\_HSEON);

/\* Wait till HSE is ready and if Time out is reached exit \*/

do

{

HSEStatus = RCC->CR & RCC\_CR\_HSERDY;

StartUpCounter++;

} while((HSEStatus == 0) && (StartUpCounter != HSE\_STARTUP\_TIMEOUT));

if ((RCC->CR & RCC\_CR\_HSERDY) != RESET)

{

HSEStatus = (uint32\_t)0x01;

}

else

{

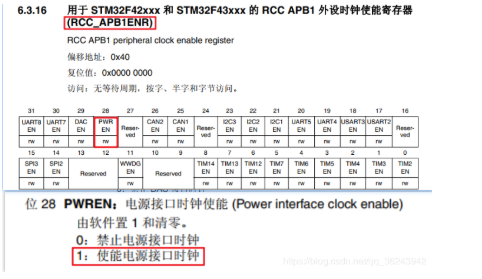
HSEStatus = (uint32\_t)0x00;

}

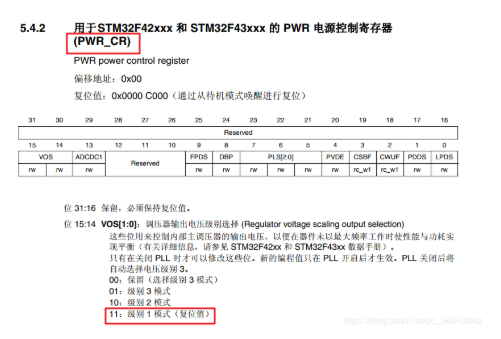
if (HSEStatus == (uint32\_t)0x01)

{

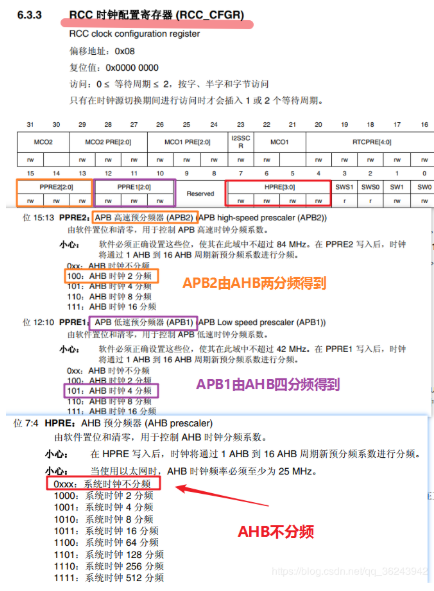
/\* Select regulator voltage output Scale 1 mode, System frequency up to 168 MHz \*/



RCC->APB1ENR |=1<<28; ;



PWR->CR|=3<<14;



RCC-> CFGR | = (0 << 4) | (5 << 10) | (4 << 13);

// HCLK không bị chia; APB1 chia cho 4; APB2 chia cho 2.

/\* HCLK = SYSCLK / 1\*/

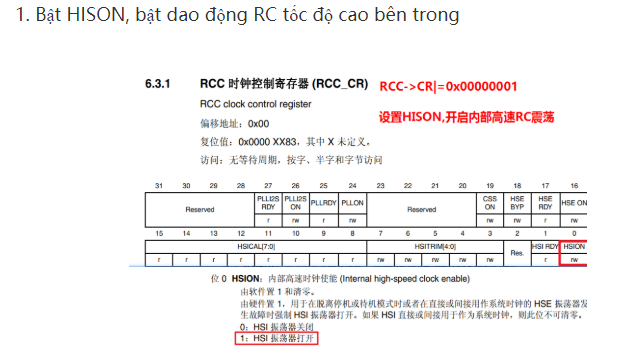
RCC->CFGR |= RCC\_CFGR\_HPRE\_DIV1;

/\* PCLK2 = HCLK / 2\*/

RCC->CFGR |= RCC\_CFGR\_PPRE2\_DIV2;

/\* PCLK1 = HCLK / 4\*/

RCC->CFGR |= RCC\_CFGR\_PPRE1\_DIV4;



RCC-> CR & = ~ (1 << 24); // Tắt PLL chính

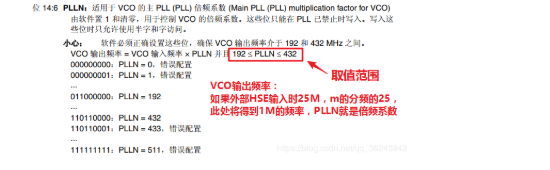
/\* Configure the main PLL \*/



RCC->PLLCFGR = PLL\_M | (PLL\_N << 6) | (((PLL\_P >> 1) -1) << 16) | (1 << 22)| (PLL\_Q << 24);

(1 << 22), chọn nguồn đồng hồ của PLL chính là đồng hồ dao động bên ngoài HSE ,

/\* Enable the main PLL \*/



RCC->CR |= RCC\_CR\_PLLON;

/\* Wait till the main PLL is ready \*/

while((RCC->CR & RCC\_CR\_PLLRDY) == 0)

{

}

/\* Configure Flash prefetch, Instruction cache, Data cache and wait state \*/

FLASH->ACR = FLASH\_ACR\_ICEN |FLASH\_ACR\_DCEN |FLASH\_ACR\_LATENCY\_5WS;

FLASH-> ACR | = 1 << 8; // Cho phép tìm nạp trước lệnh.

FLASH-> ACR | = 1 << 9; // Kích hoạt bộ đệm lệnh.

FLASH-> ACR | = 1 << 10; // Bật bộ nhớ cache dữ liệu.

FLASH-> ACR | = 5 << 0; // 5 chu kỳ chờ CPU.

/\* Select the main PLL as system clock source \*/

RCC->CFGR &= (uint32\_t)((uint32\_t)~(RCC\_CFGR\_SW));

RCC->CFGR |= RCC\_CFGR\_SW\_PLL;

/\* Wait till the main PLL is used as system clock source \*/

while ((RCC->CFGR & (uint32\_t)RCC\_CFGR\_SWS ) != RCC\_CFGR\_SWS\_PLL);

{

}

else

{

//\*\*\*\*\*//

}