System Clock

Friday, December 19, 2014
9:11 AM

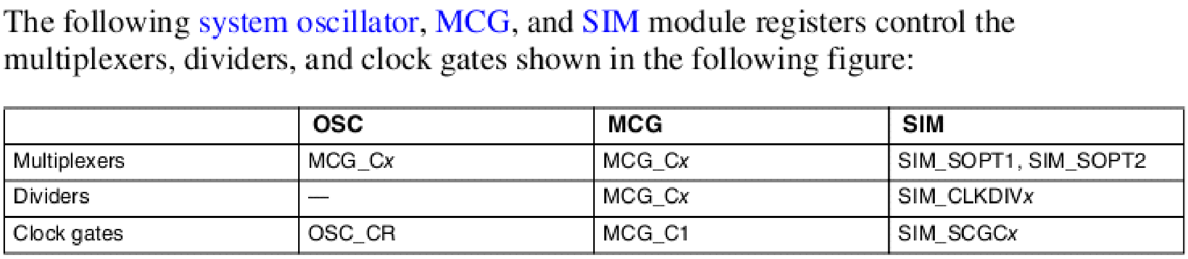
1) INTRODUCTION

The Cortex M0+ resides within a synchronous core platform, where the processor and bus masters, flash memory, and peripheral clocks can be configured independently. The clock distribution figure shows how clocks from the MCG and XOSC modules are distributed to the microcontroller's other function units. Some modules in the microcontroller have selectable clock input.

2) PROGRAMMING MODEL

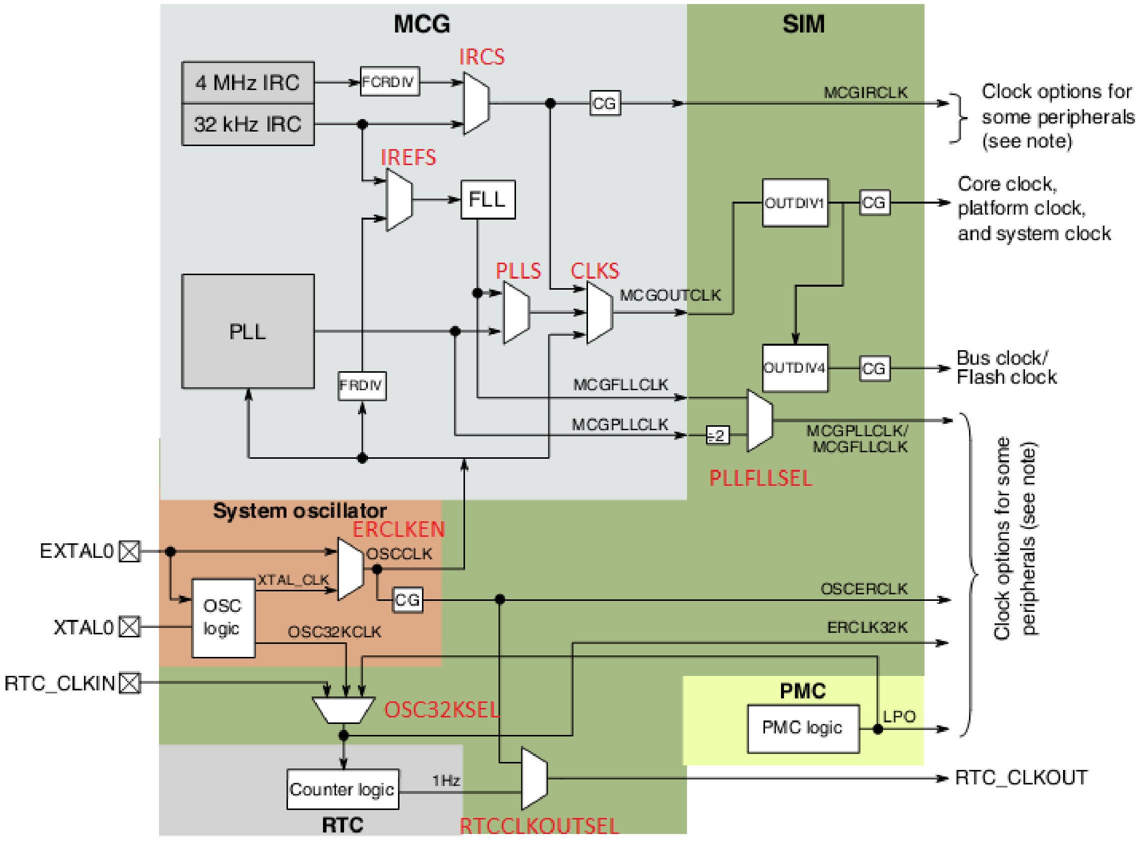
The selection and multiplexing of system clock sources is controlled and programmed via the MCG module. The setting of clock dividers and module clock gating for the system are programmed via the SIM module. Refer to the MCG and SIM" sections for detailed register and bit descriptions

3) High-level device clocking diagram



The following system oscillator, MCG, and SIM module registers control the multiplexers, dividers, and clock gates shown in the following figure:

	OSC	MCG	SIM
Multiplexers	MCG_C*x*	MCG_C*x*	SIM_SOPT1, SIM_SOPT2
Dividers	—	MCG_C*x*	SIM_CLKDIV*x*
Clock gates	OSC_CR	MCG_C1	SIM_SCGC*x*



CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 5-1. Clocking diagram

4) KEIL programming model

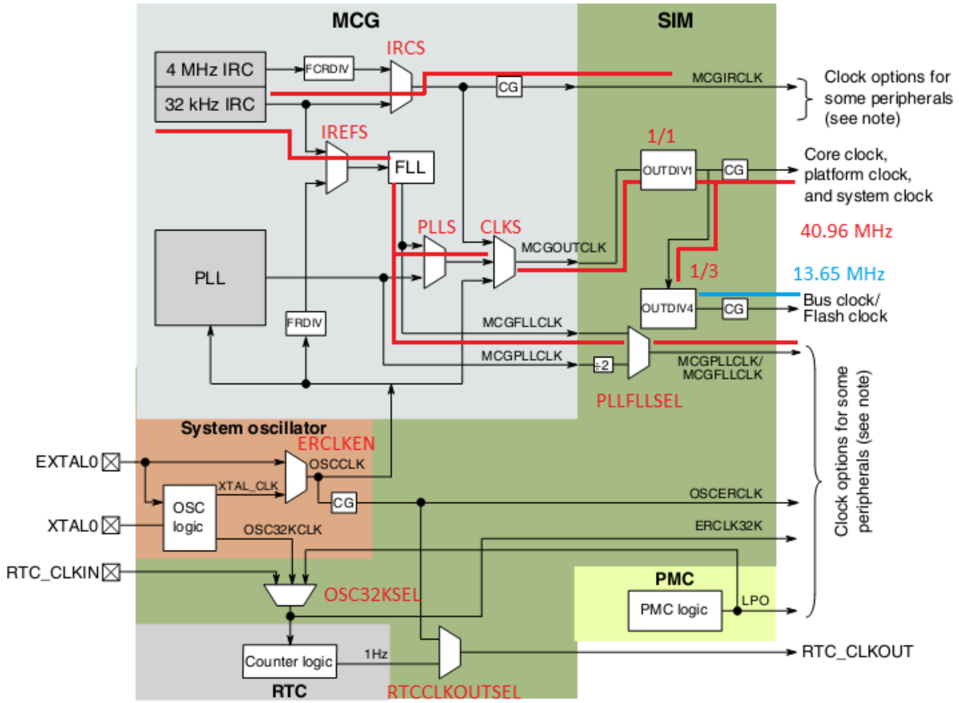
4.1) CLOCK_SETUP = 0

```
void SystemInit (void) {

    #if (DISABLE_WDOG)
      /* Disable the WDOG module */
      /* SIM_COPC: COPT=0,COPCLKS=0,COPW=0 */
      SIM->COPC = (uint32_t)0x00u;
    #endif /* (DISABLE_WDOG) */

    #if (CLOCK_SETUP == 0)
      /* OUTDIV1 = 0 (1/1), OUTDIV4 = 2 (1/3) */
      SIM->CLKDIV1 = (uint32_t)0x00020000UL;
      /* Switch to FEI Mode */
      /* MCG->C1: CLKS=0,FRDIV=0,IREFS=1,IRCLKEN=1,IREFSTEN=0 */
      MCG->C1 = (uint8_t)0x06U;
      /* MCG_C2: LOCRE0=0,RANGE0=0,HGO0=0,EREFS0=0,LP=0,IRCS=0  */
      MCG->C2 &= (uint8_t)~(uint8_t)0xBFU;
      /* MCG->C4: DMX32=0,DRST_DRS=1, FEI mode, FFL factor = 1280, Fout = 40.96 MHz */
      MCG->C4 = (uint8_t)((MCG->C4 & (uint8_t)~(uint8_t)0xC0U) | (uint8_t)0x20U);
      /* OSC0->CR: ERCLKEN=1,??=0,EREFSTEN=0,??=0,SC2P=0,SC4P=0,SC8P=0,SC16P=0 */
      OSC0->CR = (uint8_t)0x80U;
      /* MCG->C5: ??=0,PLLCLKEN0=0,PLLSTEN0=0,PRDIV0=0 */
      MCG->C5 = (uint8_t)0x00U;
      /* MCG->C6: LOLIE0=0,PLLS=0,CME0=0,VDIV0=0 */
```

```
        MCG->C6 = (uint8_t)0x00U;
        while((MCG->S & MCG_S_IREFST_MASK) == 0x00U) {
          /* Check that the source of the FLL reference clock is the internal reference clock. */
        }
        while((MCG->S & 0x0CU) != 0x00U) {
          /* Wait until output of the FLL is selected */
        }
      #endif /* CLOCK_SETUP == 0 */
}
```



CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 5-1. Clocking diagram

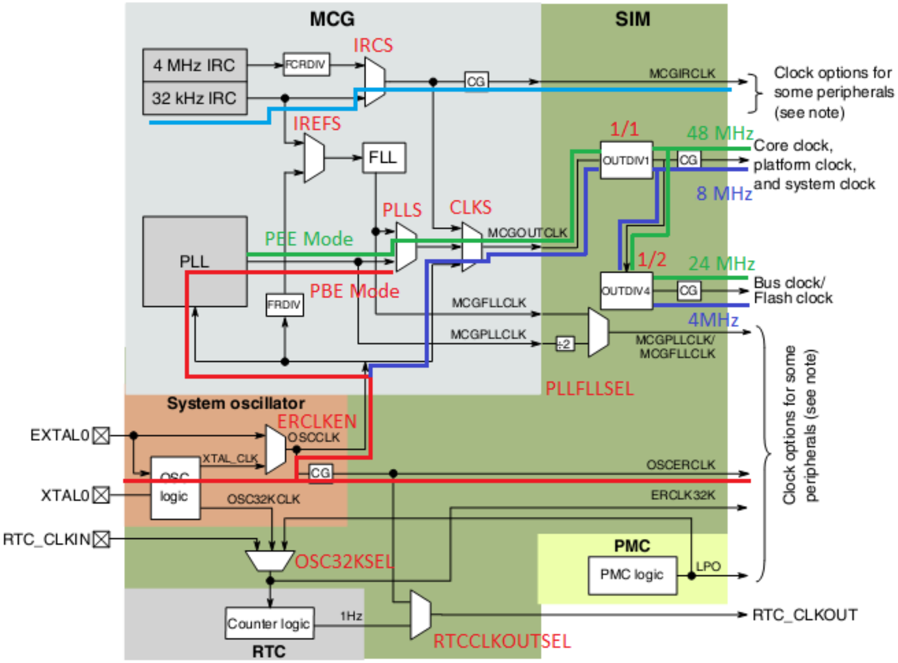
4.1) CLOCK_SETUP = 1

```
void SystemInit (void) {

    #if (DISABLE_WDOG)
      /* Disable the WDOG module */
      /* SIM_COPC: COPT=0,COPCLKS=0,COPW=0 */
      SIM->COPC = (uint32_t)0x00u;
    #endif /* (DISABLE_WDOG) */

    #if (CLOCK_SETUP == 1)
      /* OUTDIV1 = 8 (1/1), OUTDIV4 = 1 (1/2) */
      SIM->CLKDIV1 = (uint32_t)0x00010000UL;  /* Update system prescalers */
      /* Switch to FBE Mode */
      /* MCG_C2: LOCRE0=0,RANGE0=2,HGO0=0,EREFS0=1,LP=0,IRCS=0  */
      MCG->C2 = (uint8_t)((MCG->C2 & (uint8_t)~(uint8_t)0x9BU) | (uint8_t)0x24U);
      /* OSC0->CR: ERCLKEN=1,??=0,EREFSTEN=0,??=0,SC2P=1,SC4P=0,SC8P=0,SC16P=0 */
      OSC0->CR = (uint8_t)0x80U;
      /* MCG_C1: CLKS=2,FRDIV=3,IREFS=0,IRCLKEN=1,IREFSTEN=0 */
      MCG->C1 = (uint8_t)0x9AU;
      /* MCG->C4: DMX32=0,DRST_DRS=0  */
      MCG->C4 &= (uint8_t)~(uint8_t)0xE0U;
      /* MCG->C5: ??=0,PLLCLKEN0=0,PLLSTEN0=0,PRDIV0=3 */
      MCG->C5 = (uint8_t)0x03U;
      /* MCG->C6: LOLIE0=0,PLLS=0,CME0=0,VDIV0=0  */
      MCG->C6 = (uint8_t)0x00U;
      while((MCG->S & MCG_S_IREFST_MASK) != 0x00U) {
        /* Check that the source of the FLL reference clock is the external reference clock. */
      }
      while((MCG->S & 0x0CU) != 0x08U) {
        /* Wait until external reference clock is selected as MCG output */
      }
      /* Switch to PBE Mode */
      /* MCG->C6: LOLIE0=0,PLLS=1,CME0=0,VDIV0=0 */
      MCG->C6 = (uint8_t)0x40U;
      while((MCG->S & 0x0CU) != 0x08U) {
        /* Wait until external reference clock is selected as MCG output */
      }
      while((MCG->S & MCG_S_LOCK0_MASK) == 0x00U) {
        /* Wait until locked */
      }

      /* Switch to PEE Mode */
      /* PLL output = (OSCCLK / PLL_R) x M = 8MHz / 4 x 24 = 48MHz   */
      /* PLL_R = 4 dependences on PRDIV0  */
      /* M = 24 dependences on VDIV0  */
      /* MCG->C1: CLKS=0,FRDIV=3,IREFS=0,IRCLKEN=1,IREFSTEN=0 */
      MCG->C1 = (uint8_t)0x1AU;
      while((MCG->S & 0x0CU) != 0x0CU) {
      /* Wait until output of the PLL is selected */
      }
    #endif /* CLOCK_SETUP == 0 */
}
```



CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 5-1. Clocking diagram