TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – GJ (박현우)
uc820@naver.com

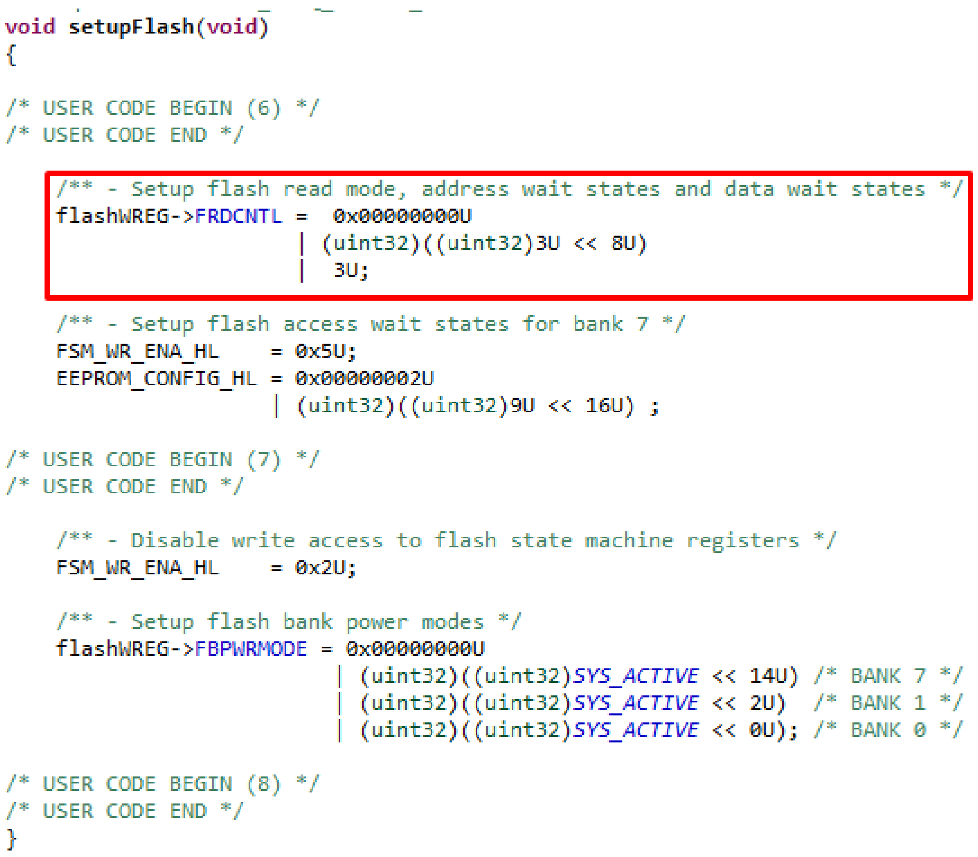
목차

1. Cortex-R5F Hercules Safety MCU – boot code 분석하기

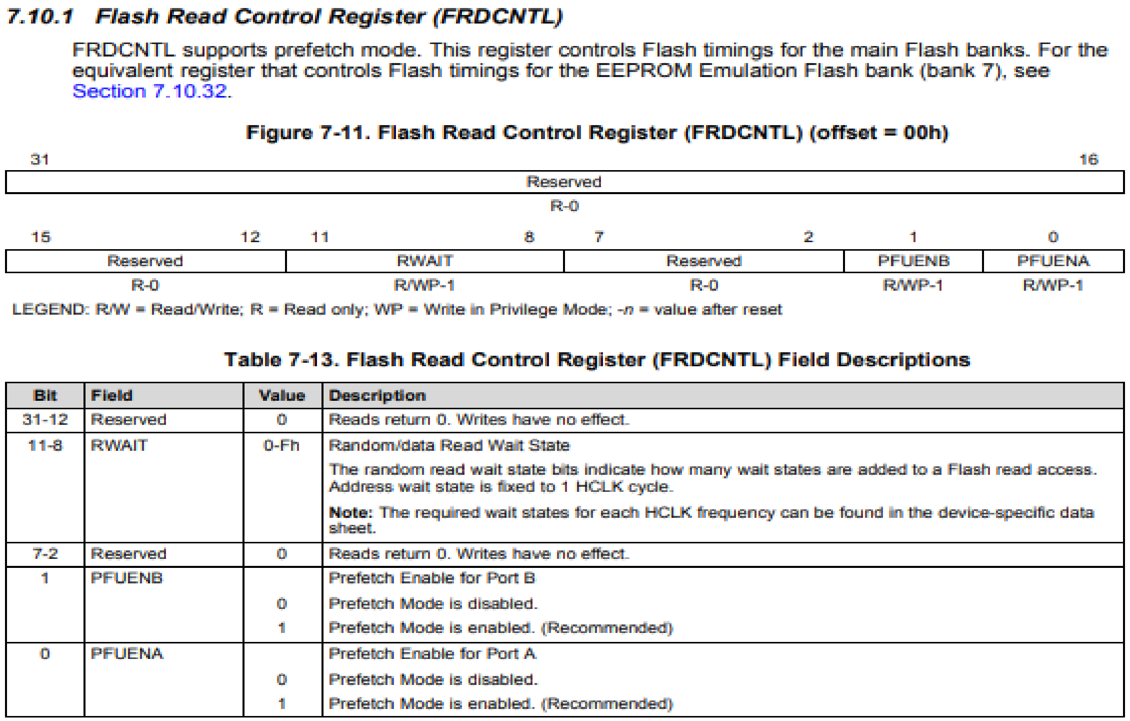
7) systemInit()

- setupFlash()
- trimLPO()
- mapClocks()

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – setupFlash())



```
void setupFlash(void)
{

/* USER CODE BEGIN (6) */
/* USER CODE END */

    /** - Setup flash read mode, address wait states and data wait states */
    flashWREG->FRDCNTL =  0x00000000U
                       | (uint32)((uint32)3U << 8U)
                       |  3U;

    /** - Setup flash access wait states for bank 7 */
    FSM_WR_ENA_HL    = 0x5U;
    EEPROM_CONFIG_HL = 0x00000002U
                     | (uint32)((uint32)9U << 16U) ;

/* USER CODE BEGIN (7) */
/* USER CODE END */

    /** - Disable write access to flash state machine registers */
    FSM_WR_ENA_HL    = 0x2U;

    /** - Setup flash bank power modes */
    flashWREG->FBPWRMODE = 0x00000000U
                         | (uint32)((uint32)SYS_ACTIVE << 14U) /* BANK 7 */
                         | (uint32)((uint32)SYS_ACTIVE << 2U)  /* BANK 1 */
                         | (uint32)((uint32)SYS_ACTIVE << 0U); /* BANK 0 */

/* USER CODE BEGIN (8) */
/* USER CODE END */
}
```



7.10.1 *Flash Read Control Register (FRDCNTL)*

FRDCNTL supports prefetch mode. This register controls Flash timings for the main Flash banks. For the equivalent register that controls Flash timings for the EEPROM Emulation Flash bank (bank 7), see Section 7.10.32.

Figure 7-11. Flash Read Control Register (FRDCNTL) (offset = 00h)

31 – 16
Reserved
R-0

15 – 12	11 – 8	7 – 2	1	0
Reserved	RWAIT	Reserved	PFUENB	PFUENA
R-0	R/WP-1	R-0	R/WP-1	R/WP-1

LEGEND: R/W = Read/Write; R = Read only; WP = Write in Privilege Mode; -*n* = value after reset

Table 7-13. Flash Read Control Register (FRDCNTL) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reads return 0. Writes have no effect.
11-8	RWAIT	0-Fh	Random/data Read Wait State The random read wait state bits indicate how many wait states are added to a Flash read access. Address wait state is fixed to 1 HCLK cycle. **Note:** The required wait states for each HCLK frequency can be found in the device-specific data sheet.
7-2	Reserved	0	Reads return 0. Writes have no effect.
1	PFUENB		Prefetch Enable for Port B
		0	Prefetch Mode is disabled.
		1	Prefetch Mode is enabled. (Recommended)
0	PFUENA		Prefetch Enable for Port A
		0	Prefetch Mode is disabled.
		1	Prefetch Mode is enabled. (Recommended)

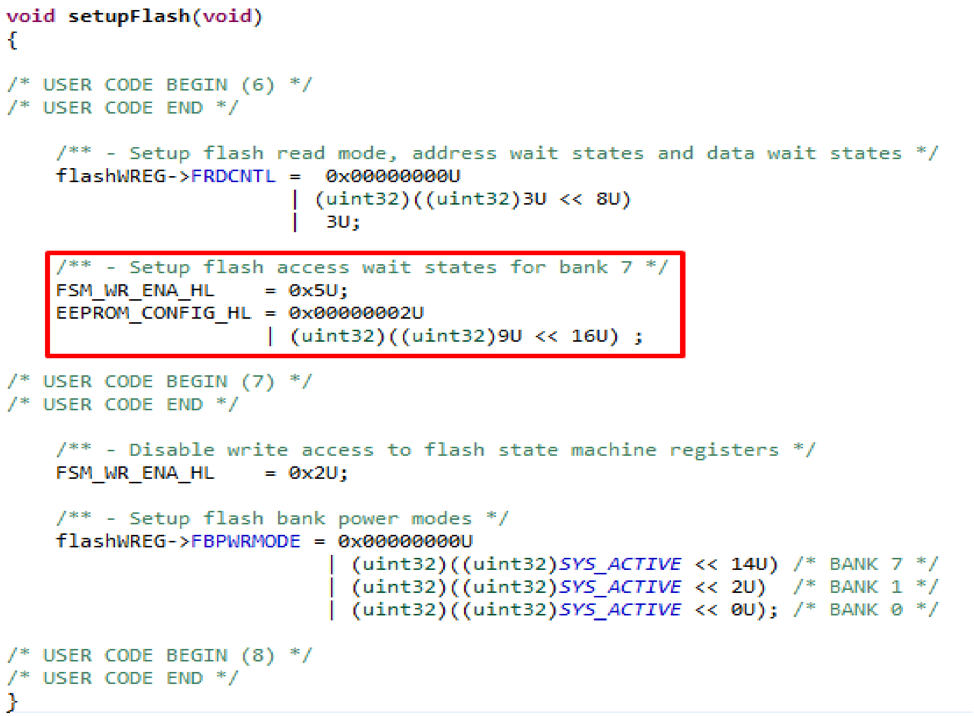
flashWREG->FRDCNTL

0000 0000 0000 0000 0000 0011 0000 0011

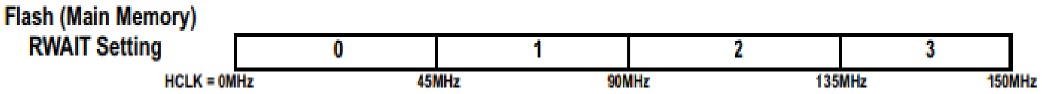
RWAIT – Address wait state = 1 HCLK cycle

PFUENA : Prefetch Mode is enabled.

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – setupFlash())

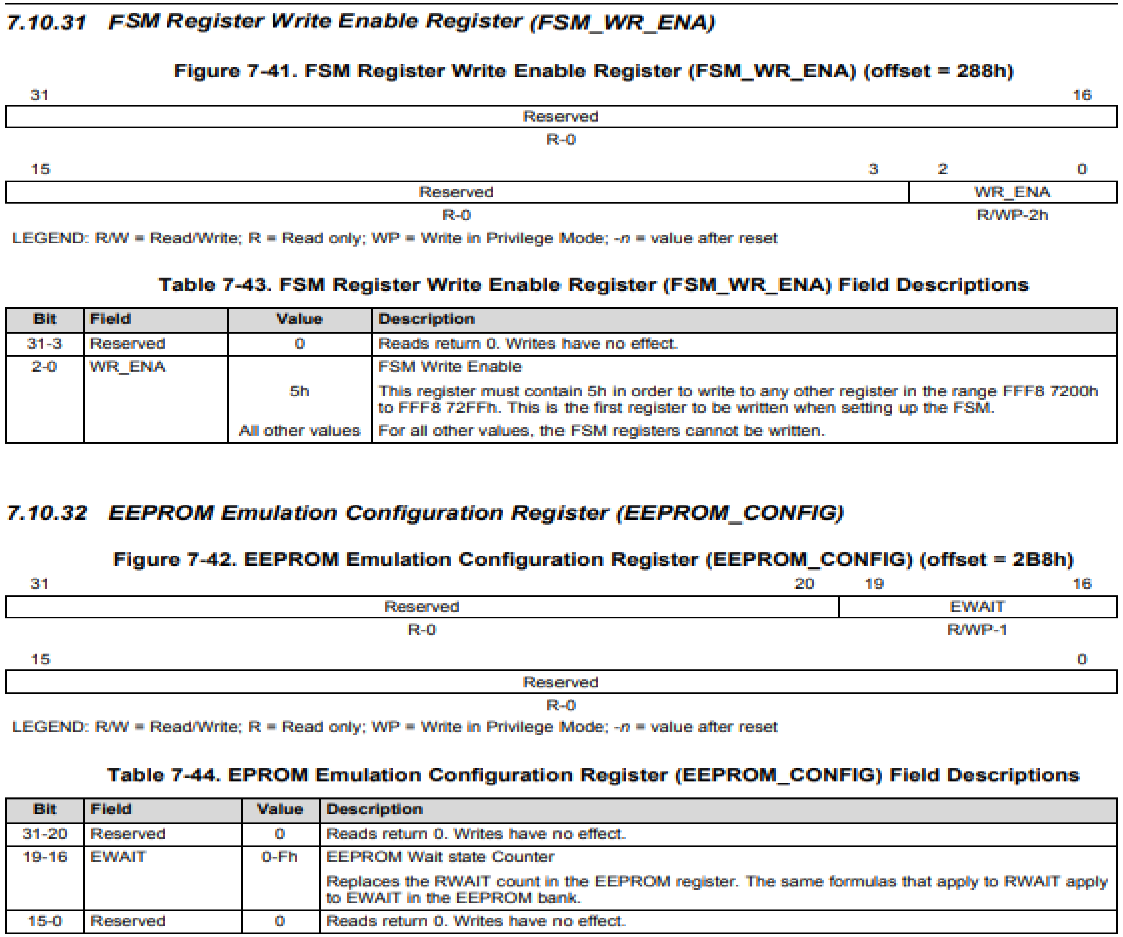


```
void setupFlash(void)
{

/* USER CODE BEGIN (6) */
/* USER CODE END */

    /** - Setup flash read mode, address wait states and data wait states */
    flashWREG->FRDCNTL =  0x00000000U
                       | (uint32)((uint32)3U << 8U)
                       |  3U;

    /** - Setup flash access wait states for bank 7 */
    FSM_WR_ENA_HL     = 0x5U;
    EEPROM_CONFIG_HL = 0x00000002U
                     | (uint32)((uint32)9U << 16U) ;

/* USER CODE BEGIN (7) */
/* USER CODE END */

    /** - Disable write access to flash state machine registers */
    FSM_WR_ENA_HL     = 0x2U;

    /** - Setup flash bank power modes */
    flashWREG->FBPWRMODE = 0x00000000U
                          | (uint32)((uint32)SYS_ACTIVE << 14U) /* BANK 7 */
                          | (uint32)((uint32)SYS_ACTIVE << 2U)  /* BANK 1 */
                          | (uint32)((uint32)SYS_ACTIVE << 0U); /* BANK 0 */

/* USER CODE BEGIN (8) */
/* USER CODE END */
}
```



FSM_WR_ENA_HL
0000 0000 0000 0000 0000 0000 0000 0101

EEPROM_CONFIG_HL
0000 0000 0000 1001 0000 0000 0000 0010



7.10.31 FSM Register Write Enable Register (FSM_WR_ENA)

Figure 7-41. FSM Register Write Enable Register (FSM_WR_ENA) (offset = 288h)

31–16
Reserved
R-0

15–3	2–0
Reserved	WR_ENA
R-0	R/WP-2h

LEGEND: R/W = Read/Write; R = Read only; WP = Write in Privilege Mode; -*n* = value after reset

Table 7-43. FSM Register Write Enable Register (FSM_WR_ENA) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reads return 0. Writes have no effect.
2-0	WR_ENA		FSM Write Enable
		5h	This register must contain 5h in order to write to any other register in the range FFF8 7200h to FFF8 72FFh. This is the first register to be written when setting up the FSM.
		All other values	For all other values, the FSM registers cannot be written.

7.10.32 EEPROM Emulation Configuration Register (EEPROM_CONFIG)

Figure 7-42. EEPROM Emulation Configuration Register (EEPROM_CONFIG) (offset = 2B8h)

31–20	19–16
Reserved	EWAIT
R-0	R/WP-1

15–0
Reserved
R-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in Privilege Mode; -*n* = value after reset

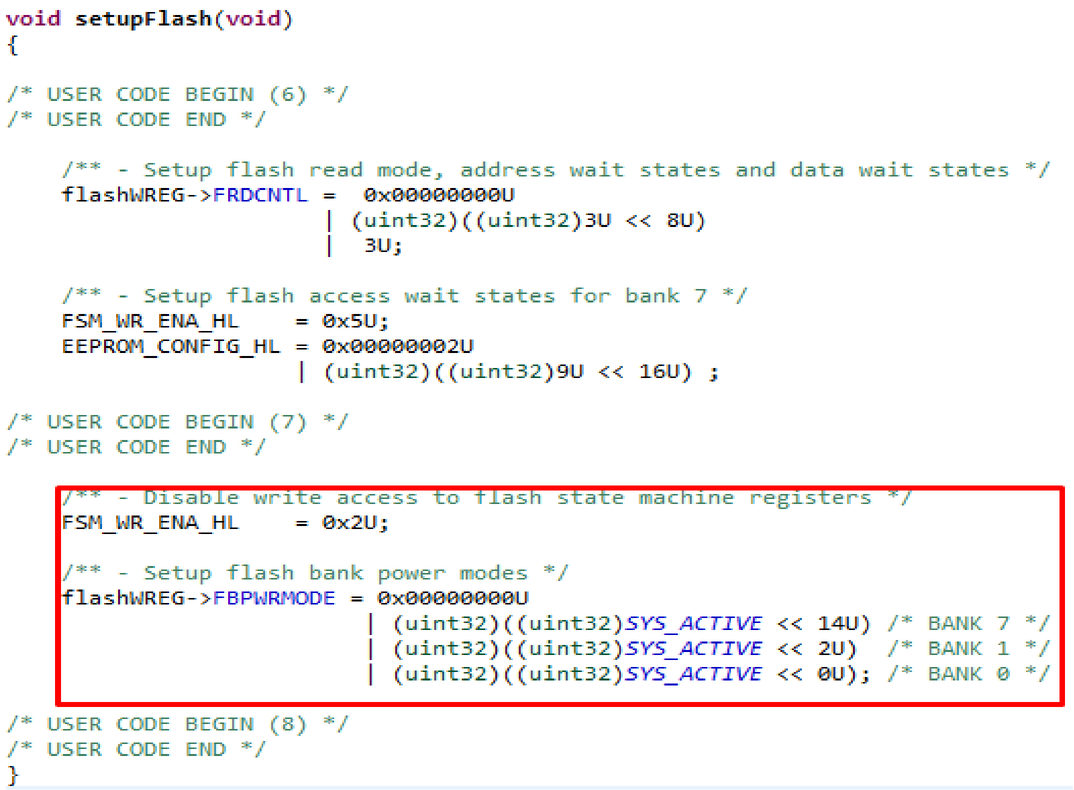
Table 7-44. EPROM Emulation Configuration Register (EEPROM_CONFIG) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reads return 0. Writes have no effect.
19-16	EWAIT	0-Fh	EEPROM Wait state Counter Replaces the RWAIT count in the EEPROM register. The same formulas that apply to RWAIT apply to EWAIT in the EEPROM bank.
15-0	Reserved	0	Reads return 0. Writes have no effect.

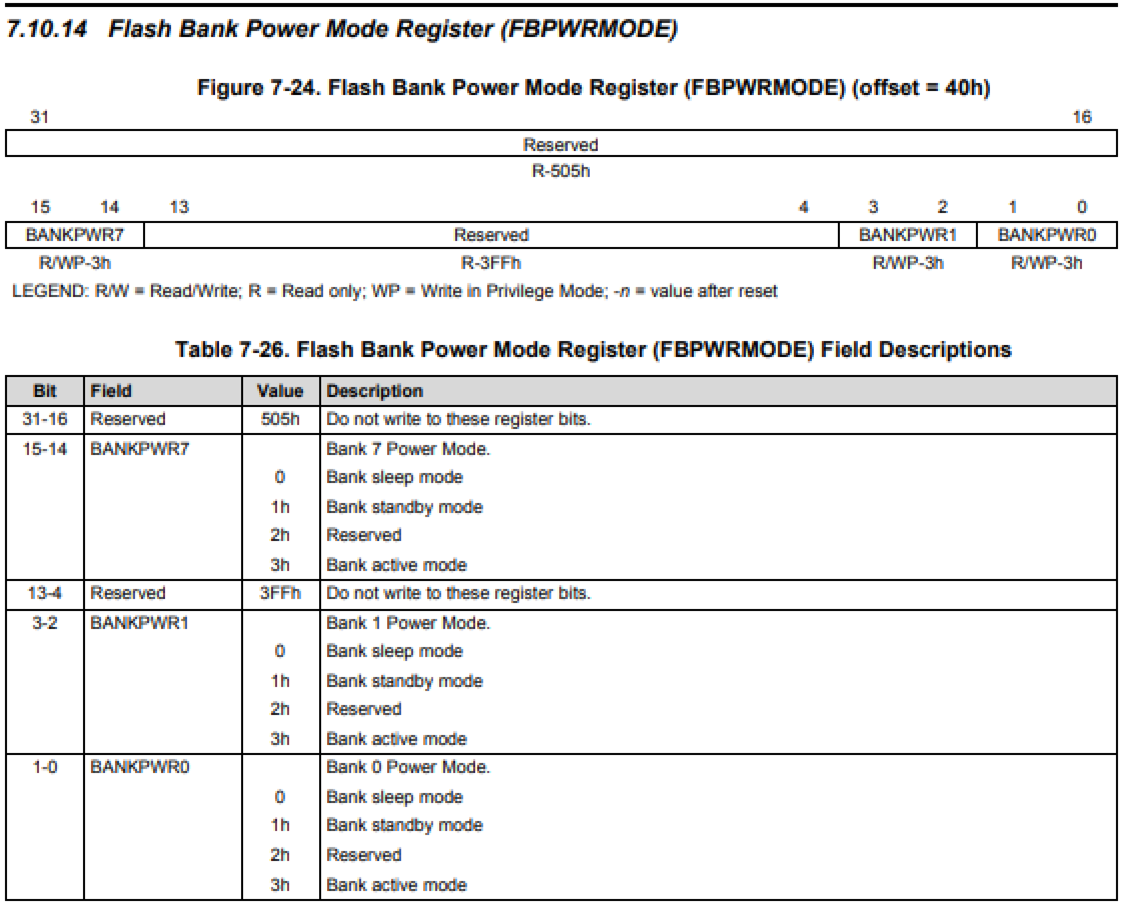
FSM_WR_ENA
2-0 : 5h FFF8 7200~72FF 쓰기권한

EEPROM_CONFIG_HL
19-16 : 9h EEPROM 레지스터에 있는 RWAIT count로 교체

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – setupFlash())



```
void setupFlash(void)
{

/* USER CODE BEGIN (6) */
/* USER CODE END */

    /** - Setup flash read mode, address wait states and data wait states */
    flashWREG->FRDCNTL =  0x00000000U
                       | (uint32)((uint32)3U << 8U)
                       |  3U;

    /** - Setup flash access wait states for bank 7 */
    FSM_WR_ENA_HL     = 0x5U;
    EEPROM_CONFIG_HL = 0x00000002U
                     | (uint32)((uint32)9U << 16U) ;

/* USER CODE BEGIN (7) */
/* USER CODE END */

    /** - Disable write access to flash state machine registers */
    FSM_WR_ENA_HL     = 0x2U;

    /** - Setup flash bank power modes */
    flashWREG->FBPWRMODE = 0x00000000U
                          | (uint32)((uint32)SYS_ACTIVE << 14U) /* BANK 7 */
                          | (uint32)((uint32)SYS_ACTIVE << 2U)  /* BANK 1 */
                          | (uint32)((uint32)SYS_ACTIVE << 0U); /* BANK 0 */

/* USER CODE BEGIN (8) */
/* USER CODE END */
}
```



7.10.14 Flash Bank Power Mode Register (FBPWRMODE)

Figure 7-24. Flash Bank Power Mode Register (FBPWRMODE) (offset = 40h)

31–16
Reserved
R-505h

15–14	13–4	3–2	1–0
BANKPWR7	Reserved	BANKPWR1	BANKPWR0
R/WP-3h	R-3FFh	R/WP-3h	R/WP-3h

LEGEND: R/W = Read/Write; R = Read only; WP = Write in Privilege Mode; -*n* = value after reset

Table 7-26. Flash Bank Power Mode Register (FBPWRMODE) Field Descriptions

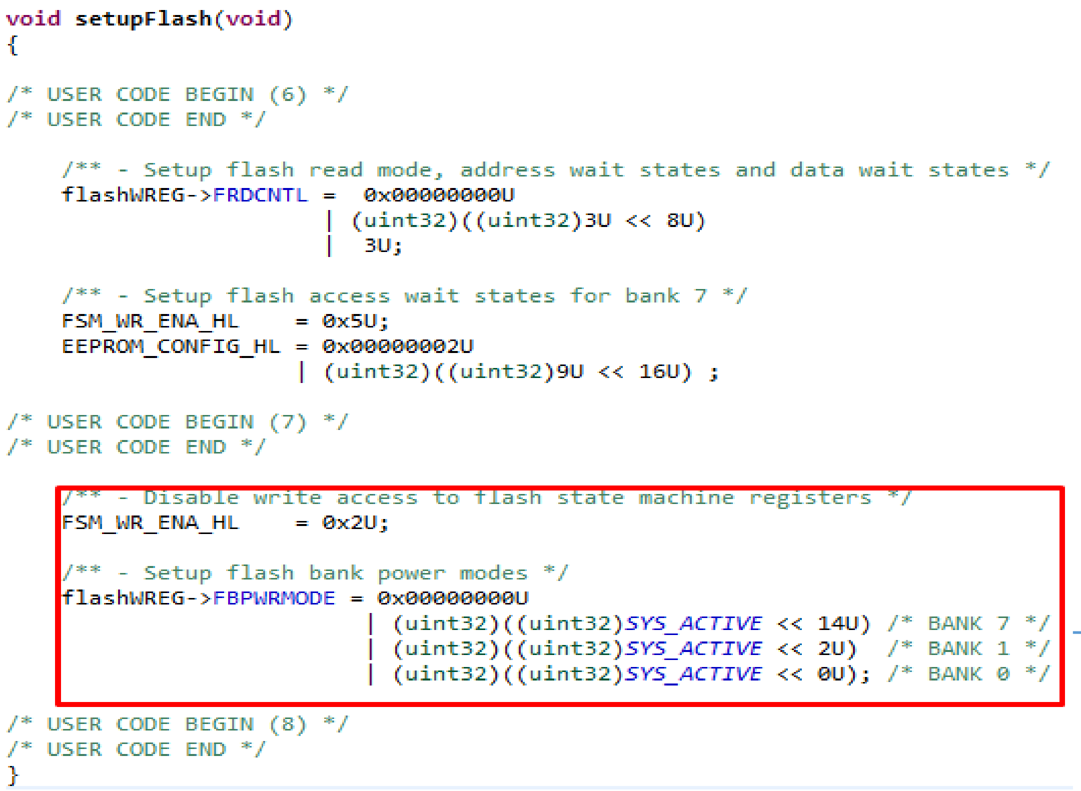
Bit	Field	Value	Description
31-16	Reserved	505h	Do not write to these register bits.
15-14	BANKPWR7		Bank 7 Power Mode.
		0	Bank sleep mode
		1h	Bank standby mode
		2h	Reserved
		3h	Bank active mode
13-4	Reserved	3FFh	Do not write to these register bits.
3-2	BANKPWR1		Bank 1 Power Mode.
		0	Bank sleep mode
		1h	Bank standby mode
		2h	Reserved
		3h	Bank active mode
1-0	BANKPWR0		Bank 0 Power Mode.
		0	Bank sleep mode
		1h	Bank standby mode
		2h	Reserved
		3h	Bank active mode

FSM_WR_ENA_HL : other value = disable write
0000 0000 0000 0000 0000 0000 0000 0010

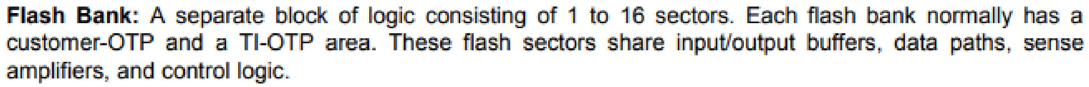
flashWREG->FBPWRMODE
0000 0000 0000 1100 0000 0000 0000 1111

BANKPWR0 1-0 : 3h = bank active mode
BANKPWR1 3-2 : 3h = bank active mode
BANKPWR7 15-14 : 3h = bank active mode

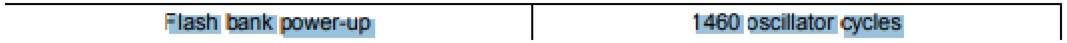
1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – setupFlash())



```
void setupFlash(void)
{

/* USER CODE BEGIN (6) */
/* USER CODE END */

    /** - Setup flash read mode, address wait states and data wait states */
    flashWREG->FRDCNTL =  0x00000000U
                       | (uint32)((uint32)3U << 8U)
                       |  3U;

    /** - Setup flash access wait states for bank 7 */
    FSM_WR_ENA_HL    = 0x5U;
    EEPROM_CONFIG_HL = 0x00000002U
                     | (uint32)((uint32)9U << 16U) ;

/* USER CODE BEGIN (7) */
/* USER CODE END */

    /** - Disable write access to flash state machine registers */
    FSM_WR_ENA_HL    = 0x2U;

    /** - Setup flash bank power modes */
    flashWREG->FBPWRMODE = 0x00000000U
                         | (uint32)((uint32)SYS_ACTIVE << 14U) /* BANK 7 */
                         | (uint32)((uint32)SYS_ACTIVE << 2U)  /* BANK 1 */
                         | (uint32)((uint32)SYS_ACTIVE << 0U); /* BANK 0 */

/* USER CODE BEGIN (8) */
/* USER CODE END */
}
```



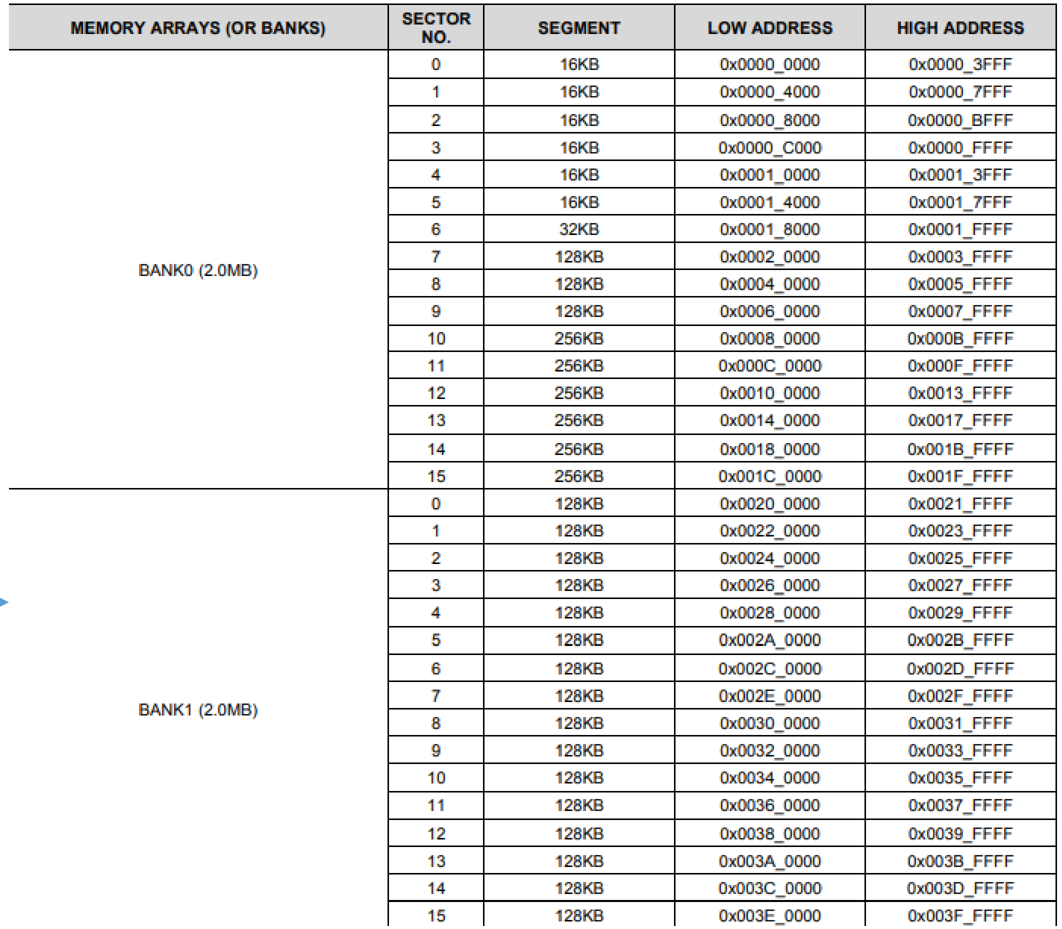
Flash Bank: A separate block of logic consisting of 1 to 16 sectors. Each flash bank normally has a customer-OTP and a TI-OTP area. These flash sectors share input/output buffers, data paths, sense amplifiers, and control logic.



Flash bank power-up	1460 oscillator cycles

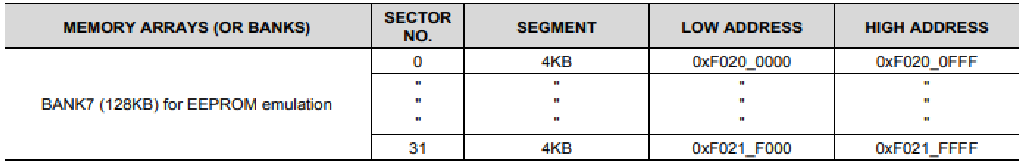
Bank 0, 1, 7을 파워 모드로 설정.

Table 6-29. Flash Memory Banks and Sectors



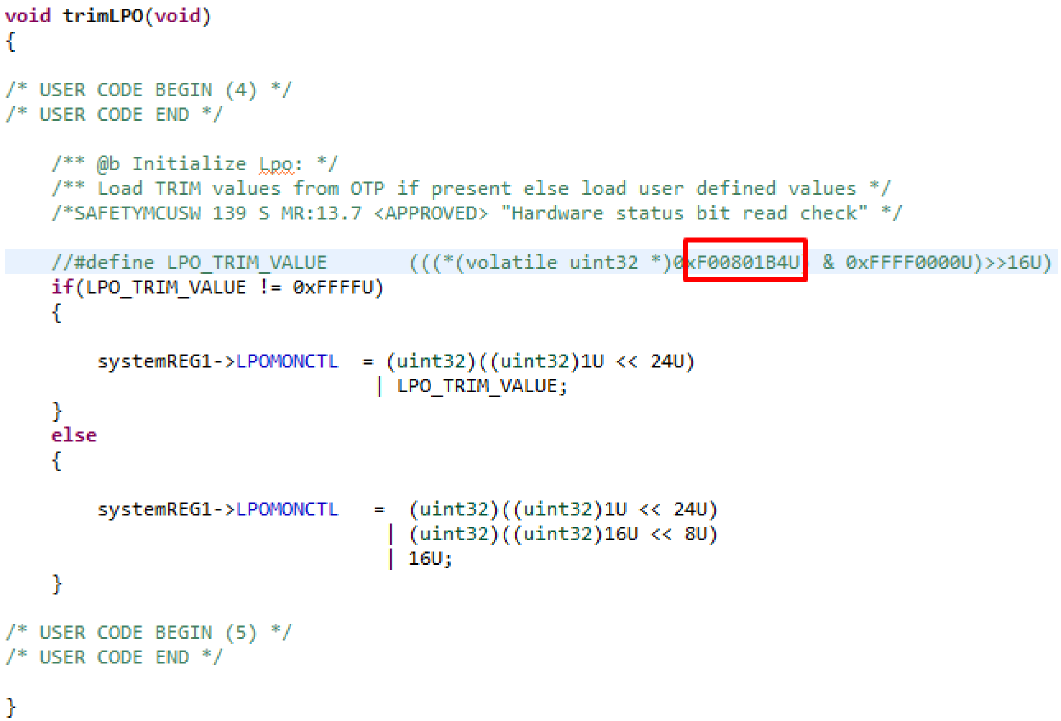
MEMORY ARRAYS (OR BANKS)	SECTOR NO.	SEGMENT	LOW ADDRESS	HIGH ADDRESS
BANK0 (2.0MB)	0	16KB	0x0000_0000	0x0000_3FFF
	1	16KB	0x0000_4000	0x0000_7FFF
	2	16KB	0x0000_8000	0x0000_BFFF
	3	16KB	0x0000_C000	0x0000_FFFF
	4	16KB	0x0001_0000	0x0001_3FFF
	5	16KB	0x0001_4000	0x0001_7FFF
	6	32KB	0x0001_8000	0x0001_FFFF
	7	128KB	0x0002_0000	0x0003_FFFF
	8	128KB	0x0004_0000	0x0005_FFFF
	9	128KB	0x0006_0000	0x0007_FFFF
	10	256KB	0x0008_0000	0x000B_FFFF
	11	256KB	0x000C_0000	0x000F_FFFF
	12	256KB	0x0010_0000	0x0013_FFFF
	13	256KB	0x0014_0000	0x0017_FFFF
	14	256KB	0x0018_0000	0x001B_FFFF
	15	256KB	0x001C_0000	0x001F_FFFF
BANK1 (2.0MB)	0	128KB	0x0020_0000	0x0021_FFFF
	1	128KB	0x0022_0000	0x0023_FFFF
	2	128KB	0x0024_0000	0x0025_FFFF
	3	128KB	0x0026_0000	0x0027_FFFF
	4	128KB	0x0028_0000	0x0029_FFFF
	5	128KB	0x002A_0000	0x002B_FFFF
	6	128KB	0x002C_0000	0x002D_FFFF
	7	128KB	0x002E_0000	0x002F_FFFF
	8	128KB	0x0030_0000	0x0031_FFFF
	9	128KB	0x0032_0000	0x0033_FFFF
	10	128KB	0x0034_0000	0x0035_FFFF
	11	128KB	0x0036_0000	0x0037_FFFF
	12	128KB	0x0038_0000	0x0039_FFFF
	13	128KB	0x003A_0000	0x003B_FFFF
	14	128KB	0x003C_0000	0x003D_FFFF
	15	128KB	0x003E_0000	0x003F_FFFF

Table 6-30. EEPROM Flash Bank

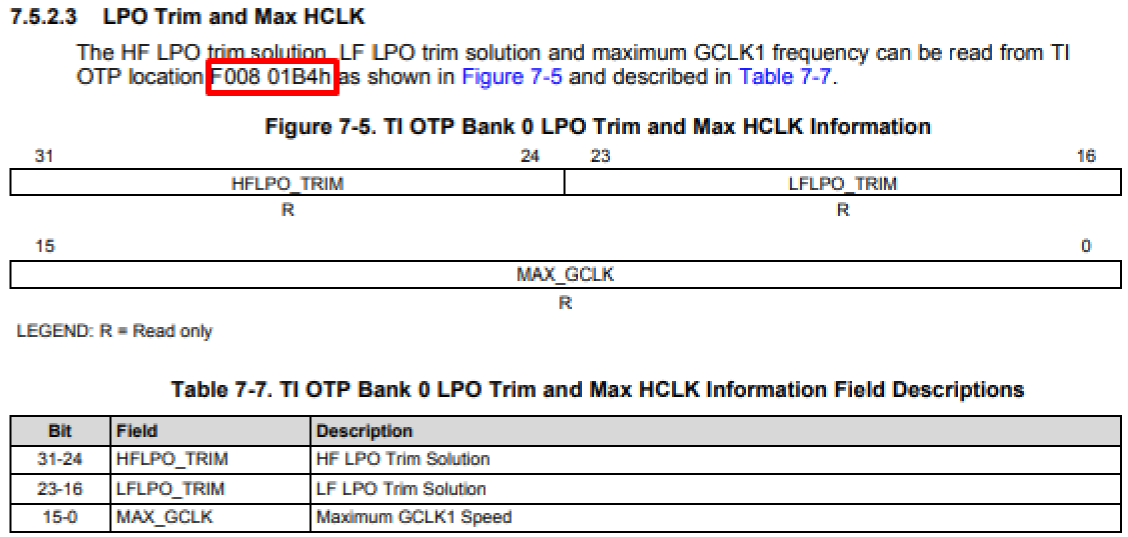


MEMORY ARRAYS (OR BANKS)	SECTOR NO.	SEGMENT	LOW ADDRESS	HIGH ADDRESS
BANK7 (128KB) for EEPROM emulation	0	4KB	0xF020_0000	0xF020_0FFF
	: :	: :	: :	: :
	31	4KB	0xF021_F000	0xF021_FFFF

1. Cortex-R5F Hercules Safety MCU - boot code 분석 (systemInit() - trimLPO())



```
void trimLPO(void)
{

/* USER CODE BEGIN (4) */
/* USER CODE END */

    /** @b Initialize Lpo: */
    /** Load TRIM values from OTP if present else load user defined values */
    /*SAFETYMCUSW 139 S MR:13.7 <APPROVED> "Hardware status bit read check" */

    //#define LPO_TRIM_VALUE       (((*(volatile uint32 *)0xF00801B4U) & 0xFFFF0000U)>>16U)
    if(LPO_TRIM_VALUE != 0xFFFFU)
    {

        systemREG1->LPOMONCTL  = (uint32)((uint32)1U << 24U)
                                | LPO_TRIM_VALUE;
    }
    else
    {

        systemREG1->LPOMONCTL   =  (uint32)((uint32)1U << 24U)
                                 | (uint32)((uint32)16U << 8U)
                                 | 16U;
    }

/* USER CODE BEGIN (5) */
/* USER CODE END */

}
```



7.5.2.3 LPO Trim and Max HCLK

The HF LPO trim solution, LF LPO trim solution and maximum GCLK1 frequency can be read from TI OTP location F008 01B4h as shown in Figure 7-5 and described in Table 7-7.

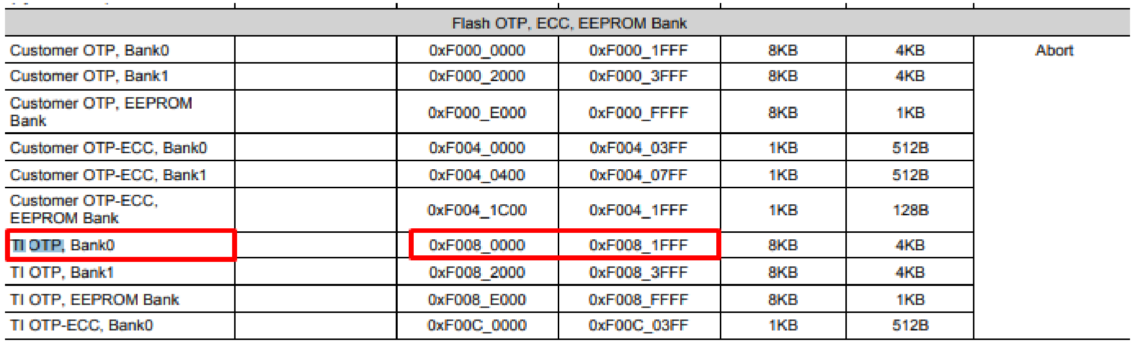
Figure 7-5. TI OTP Bank 0 LPO Trim and Max HCLK Information

31
24
23
16
HFLPO_TRIM
R
LFLPO_TRIM
R
15
0
MAX_GCLK
R

LEGEND: R = Read only

Table 7-7. TI OTP Bank 0 LPO Trim and Max HCLK Information Field Descriptions

Bit	Field	Description
31-24	HFLPO_TRIM	HF LPO Trim Solution
23-16	LFLPO_TRIM	LF LPO Trim Solution
15-0	MAX_GCLK	Maximum GCLK1 Speed

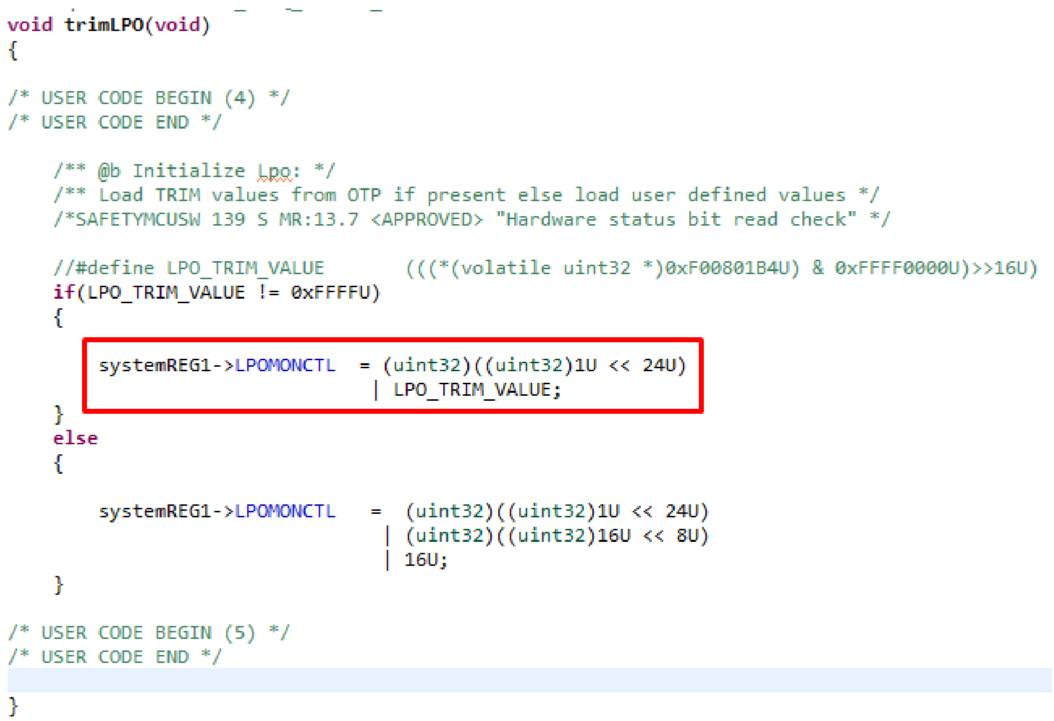


Flash OTP, ECC, EEPROM Bank						
Customer OTP, Bank0		0xF000_0000	0xF000_1FFF	8KB	4KB	Abort
Customer OTP, Bank1		0xF000_2000	0xF000_3FFF	8KB	4KB	
Customer OTP, EEPROM Bank		0xF000_E000	0xF000_FFFF	8KB	1KB	
Customer OTP-ECC, Bank0		0xF004_0000	0xF004_03FF	1KB	512B	
Customer OTP-ECC, Bank1		0xF004_0400	0xF004_07FF	1KB	512B	
Customer OTP-ECC, EEPROM Bank		0xF004_1C00	0xF004_1FFF	1KB	128B	
TI OTP, Bank0		0xF008_0000	0xF008_1FFF	8KB	4KB	
TI OTP, Bank1		0xF008_2000	0xF008_3FFF	8KB	4KB	
TI OTP, EEPROM Bank		0xF008_E000	0xF008_FFFF	8KB	1KB	
TI OTP-ECC, Bank0		0xF00C_0000	0xF00C_03FF	1KB	512B	

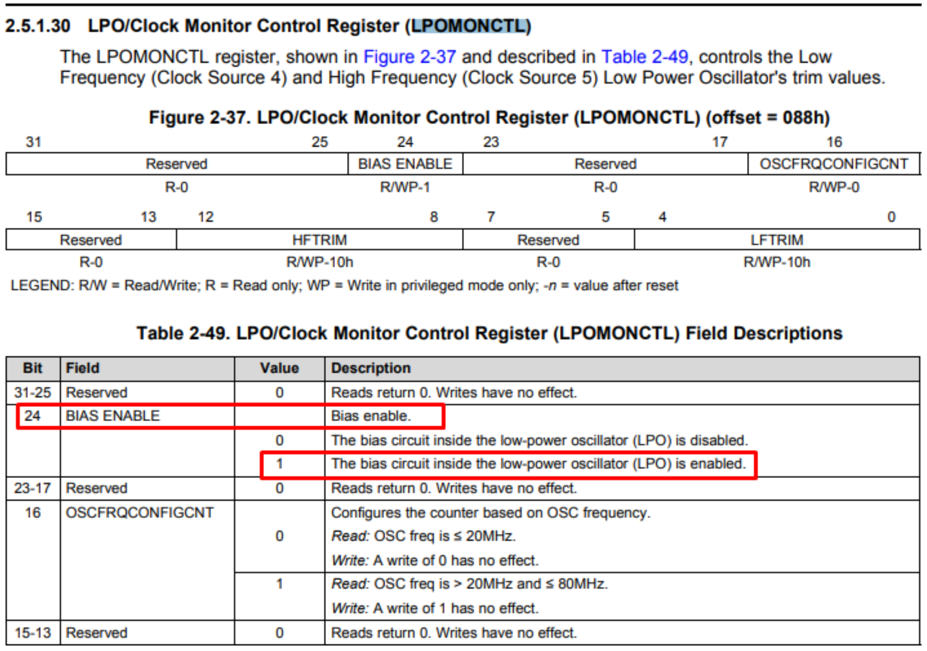
HFLPO_TRIM과 LFLPO_TRIM에 들어 있는 값을

오른쪽으로 16bit shift한다.

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – trimLPO())



```
void trimLPO(void)
{

/* USER CODE BEGIN (4) */
/* USER CODE END */

    /** @b Initialize Lpo: */
    /** Load TRIM values from OTP if present else load user defined values */
    /*SAFETYMCUSW 139 S MR:13.7 <APPROVED> "Hardware status bit read check" */

    //#define LPO_TRIM_VALUE      (((*(volatile uint32 *)0xF00801B4U) & 0xFFFF0000U)>>16U)
    if(LPO_TRIM_VALUE != 0xFFFFU)
    {

        systemREG1->LPOMONCTL  = (uint32)((uint32)1U << 24U)
                               | LPO_TRIM_VALUE;
    }
    else
    {

        systemREG1->LPOMONCTL   =  (uint32)((uint32)1U << 24U)
                                 | (uint32)((uint32)16U << 8U)
                                 | 16U;

    }

/* USER CODE BEGIN (5) */
/* USER CODE END */

}
```



2.5.1.30 LPO/Clock Monitor Control Register (LPOMONCTL)

The LPOMONCTL register, shown in Figure 2-37 and described in Table 2-49, controls the Low Frequency (Clock Source 4) and High Frequency (Clock Source 5) Low Power Oscillator's trim values.

Figure 2-37. LPO/Clock Monitor Control Register (LPOMONCTL) (offset = 088h)

31–25	24	23–17	16
Reserved	BIAS ENABLE	Reserved	OSCFRQCONFIGCNT
R-0	R/WP-1	R-0	R/WP-0

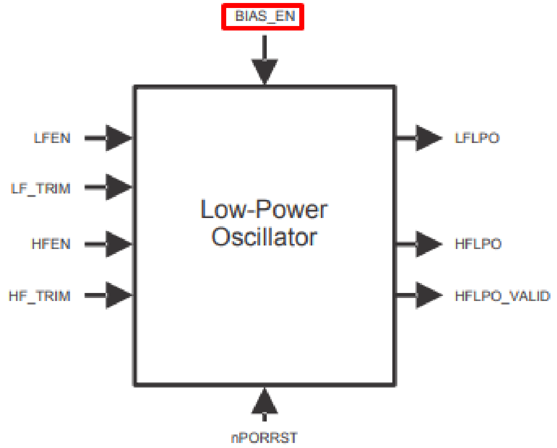
15–13	12–8	7–5	4–0
Reserved	HFTRIM	Reserved	LFTRIM
R-0	R/WP-10h	R-0	R/WP-10h

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 2-49. LPO/Clock Monitor Control Register (LPOMONCTL) Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reads return 0. Writes have no effect.
24	BIAS ENABLE		Bias enable.
		0	The bias circuit inside the low-power oscillator (LPO) is disabled.
		1	The bias circuit inside the low-power oscillator (LPO) is enabled.
23-17	Reserved	0	Reads return 0. Writes have no effect.
16	OSCFRQCONFIGCNT		Configures the counter based on OSC frequency.
		0	*Read:* OSC freq is ≤ 20MHz. *Write:* A write of 0 has no effect.
		1	*Read:* OSC freq is > 20MHz and ≤ 80MHz. *Write:* A write of 1 has no effect.
15-13	Reserved	0	Reads return 0. Writes have no effect.

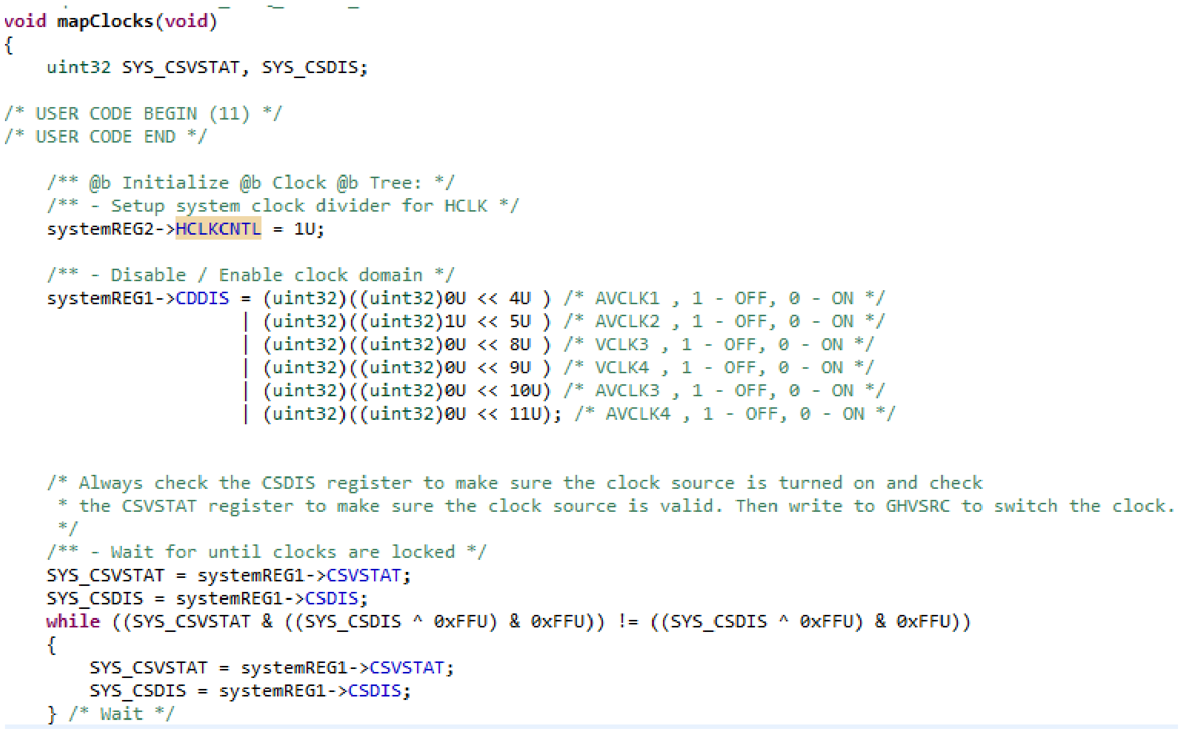
mparison clock for the crystal oscillator failure detection circuit.



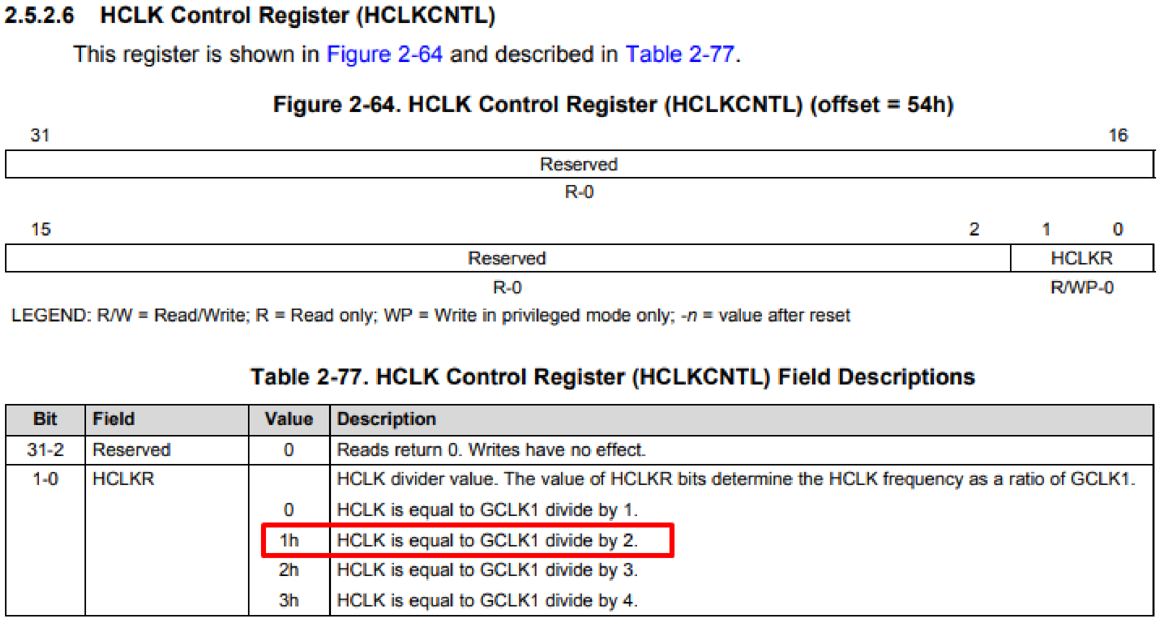
systemREG1->LPOMONCTL

0000 0001 0000 0000 0000 0000 0000 0000

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – mapClocks())



```c
void mapClocks(void)
{
    uint32 SYS_CSVSTAT, SYS_CSDIS;

/* USER CODE BEGIN (11) */
/* USER CODE END */

    /** @b Initialize @b Clock @b Tree: */
    /** - Setup system clock divider for HCLK */
    systemREG2->HCLKCNTL = 1U;

    /** - Disable / Enable clock domain */
    systemREG1->CDDIS = (uint32)((uint32)0U << 4U ) /* AVCLK1 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)1U << 5U ) /* AVCLK2 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 8U ) /* VCLK3 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 9U ) /* VCLK4 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 10U) /* AVCLK3 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 11U); /* AVCLK4 , 1 - OFF, 0 - ON */


    /* Always check the CSDIS register to make sure the clock source is turned on and check
     * the CSVSTAT register to make sure the clock source is valid. Then write to GHVSRC to switch the clock.
     */
    /** - Wait for until clocks are locked */
    SYS_CSVSTAT = systemREG1->CSVSTAT;
    SYS_CSDIS = systemREG1->CSDIS;
    while ((SYS_CSVSTAT & ((SYS_CSDIS ^ 0xFFU) & 0xFFU)) != ((SYS_CSDIS ^ 0xFFU) & 0xFFU))
    {
        SYS_CSVSTAT = systemREG1->CSVSTAT;
        SYS_CSDIS = systemREG1->CSDIS;
    } /* Wait */
```



2.5.2.6 HCLK Control Register (HCLKCNTL)

This register is shown in Figure 2-64 and described in Table 2-77.

Figure 2-64. HCLK Control Register (HCLKCNTL) (offset = 54h)

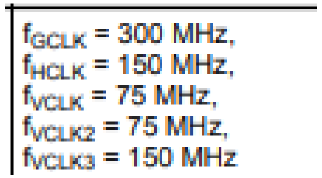
31		16
Reserved		
R-0		

15	2	1 0
Reserved		HCLKR
R-0		R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

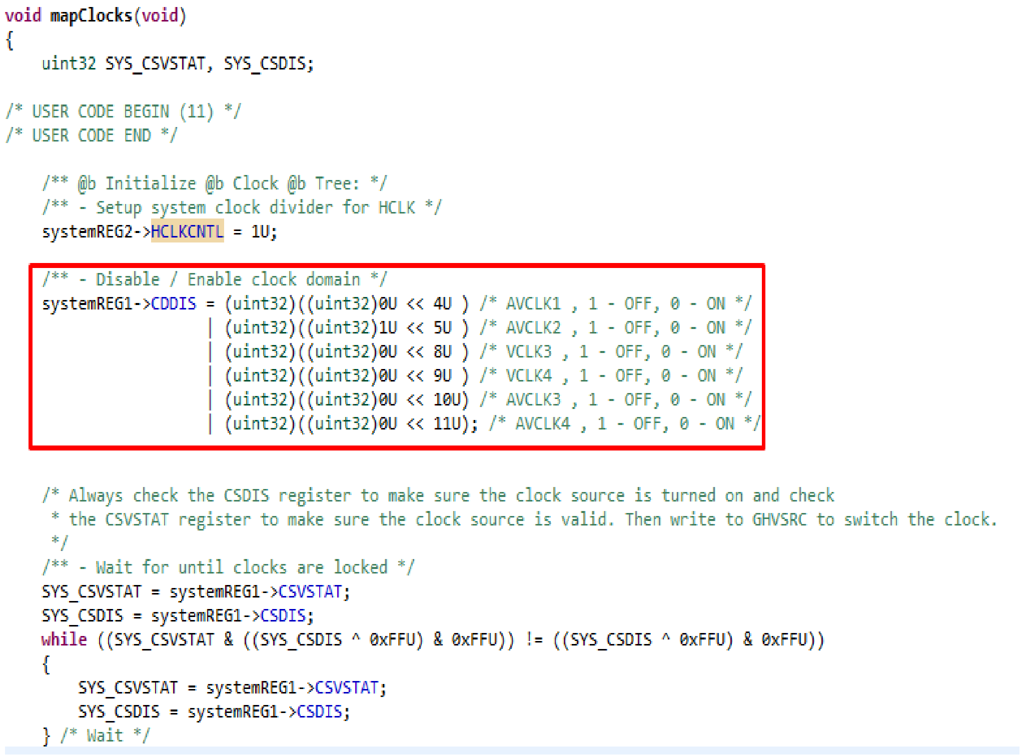
Table 2-77. HCLK Control Register (HCLKCNTL) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reads return 0. Writes have no effect.
1-0	HCLKR		HCLK divider value. The value of HCLKR bits determine the HCLK frequency as a ratio of GCLK1.
		0	HCLK is equal to GCLK1 divide by 1.
		1h	HCLK is equal to GCLK1 divide by 2.
		2h	HCLK is equal to GCLK1 divide by 3.
		3h	HCLK is equal to GCLK1 divide by 4.



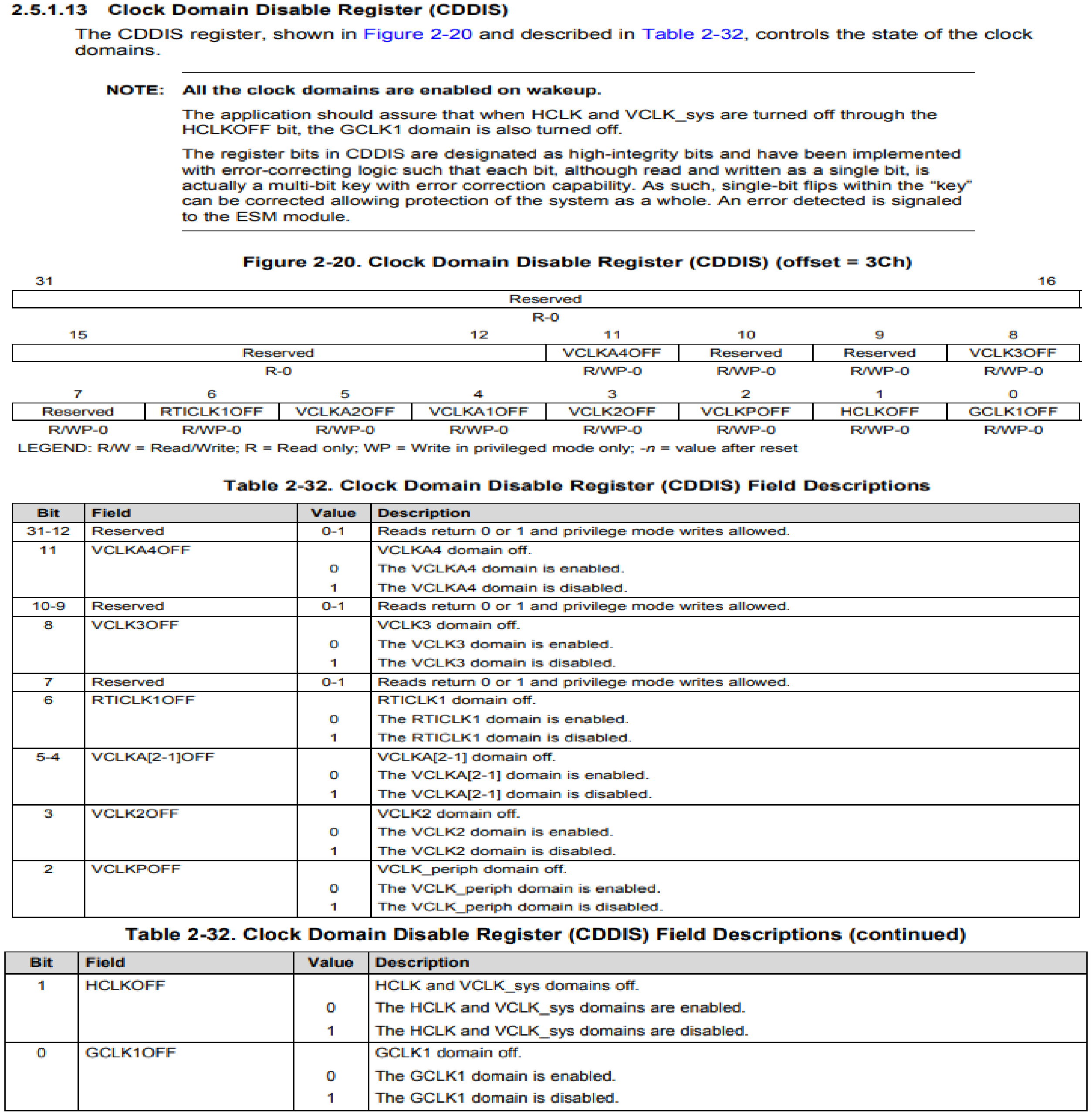
f_{GCLK} = 300 MHz,
f_{HCLK} = 150 MHz,
f_{VCLK} = 75 MHz,
f_{VCLK2} = 75 MHz,
f_{VCLK3} = 150 MHz

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – mapClocks())



```
void mapClocks(void)
{
    uint32 SYS_CSVSTAT, SYS_CSDIS;

/* USER CODE BEGIN (11) */
/* USER CODE END */

    /** @b Initialize @b Clock @b Tree: */
    /** - Setup system clock divider for HCLK */
    systemREG2->HCLKCNTL = 1U;

    /** - Disable / Enable clock domain */
    systemREG1->CDDIS = (uint32)((uint32)0U << 4U ) /* AVCLK1 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)1U << 5U ) /* AVCLK2 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 8U ) /* VCLK3 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 9U ) /* VCLK4 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 10U) /* AVCLK3 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 11U); /* AVCLK4 , 1 - OFF, 0 - ON */

    /* Always check the CSDIS register to make sure the clock source is turned on and check
     * the CSVSTAT register to make sure the clock source is valid. Then write to GHVSRC to switch the clock.
     */
    /** - Wait for until clocks are locked */
    SYS_CSVSTAT = systemREG1->CSVSTAT;
    SYS_CSDIS = systemREG1->CSDIS;
    while ((SYS_CSVSTAT & ((SYS_CSDIS ^ 0xFFU) & 0xFFU)) != ((SYS_CSDIS ^ 0xFFU) & 0xFFU))
    {
        SYS_CSVSTAT = systemREG1->CSVSTAT;
        SYS_CSDIS = systemREG1->CSDIS;
    } /* Wait */
```

VCLKA1 - ON
VCLKA2 - OFF
VCLK3 – ON
VCLKA4 - ON



2.5.1.13 Clock Domain Disable Register (CDDIS)

The CDDIS register, shown in Figure 2-20 and described in Table 2-32, controls the state of the clock domains.

NOTE: All the clock domains are enabled on wakeup.

The application should assure that when HCLK and VCLK_sys are turned off through the HCLKOFF bit, the GCLK1 domain is also turned off.

The register bits in CDDIS are designated as high-integrity bits and have been implemented with error-correcting logic such that each bit, although read and written as a single bit, is actually a multi-bit key with error correction capability. As such, single-bit flips within the "key" can be corrected allowing protection of the system as a whole. An error detected is signaled to the ESM module.

Figure 2-20. Clock Domain Disable Register (CDDIS) (offset = 3Ch)

31							16
Reserved							
R-0							
15			12	11	10	9	8
Reserved				VCLKA4OFF	Reserved	Reserved	VCLK3OFF
R-0				R/WP-0	R/WP-0	R/WP-0	R/WP-0
7	6	5	4	3	2	1	0
Reserved	RTICLK1OFF	VCLKA2OFF	VCLKA1OFF	VCLK2OFF	VCLKPOFF	HCLKOFF	GCLK1OFF
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

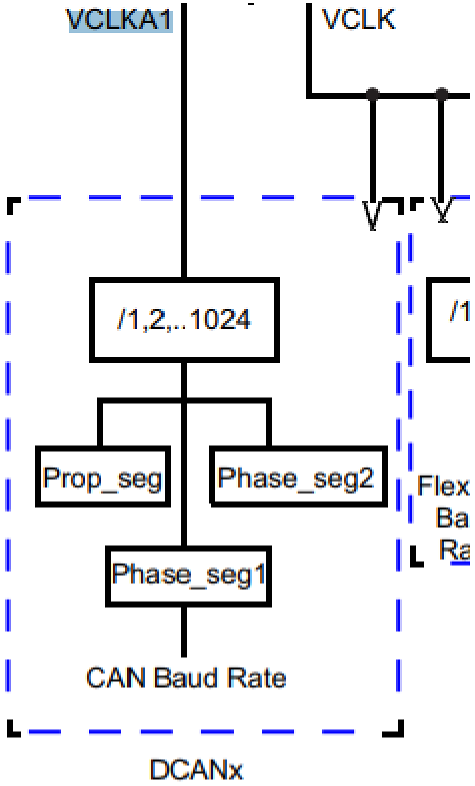
Table 2-32. Clock Domain Disable Register (CDDIS) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0-1	Reads return 0 or 1 and privilege mode writes allowed.
11	VCLKA4OFF		VCLKA4 domain off.
		0	The VCLKA4 domain is enabled.
		1	The VCLKA4 domain is disabled.
10-9	Reserved	0-1	Reads return 0 or 1 and privilege mode writes allowed.
8	VCLK3OFF		VCLK3 domain off.
		0	The VCLK3 domain is enabled.
		1	The VCLK3 domain is disabled.
7	Reserved	0-1	Reads return 0 or 1 and privilege mode writes allowed.
6	RTICLK1OFF		RTICLK1 domain off.
		0	The RTICLK1 domain is enabled.
		1	The RTICLK1 domain is disabled.
5-4	VCLKA[2-1]OFF		VCLKA[2-1] domain off.
		0	The VCLKA[2-1] domain is enabled.
		1	The VCLKA[2-1] domain is disabled.
3	VCLK2OFF		VCLK2 domain off.
		0	The VCLK2 domain is enabled.
		1	The VCLK2 domain is disabled.
2	VCLKPOFF		VCLK_periph domain off.
		0	The VCLK_periph domain is enabled.
		1	The VCLK_periph domain is disabled.

Table 2-32. Clock Domain Disable Register (CDDIS) Field Descriptions (continued)

Bit	Field	Value	Description
1	HCLKOFF		HCLK and VCLK_sys domains off.
		0	The HCLK and VCLK_sys domains are enabled.
		1	The HCLK and VCLK_sys domains are disabled.
0	GCLK1OFF		GCLK1 domain off.
		0	The GCLK1 domain is enabled.
		1	The GCLK1 domain is disabled.

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – mapClocks())



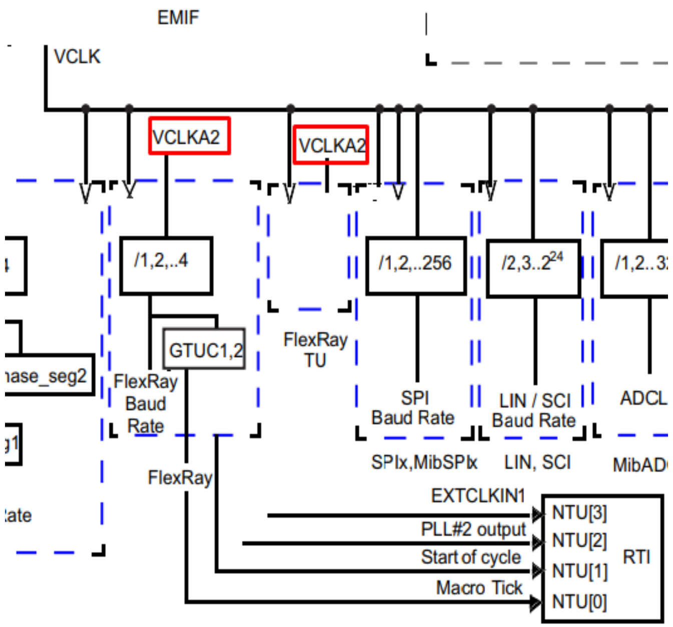
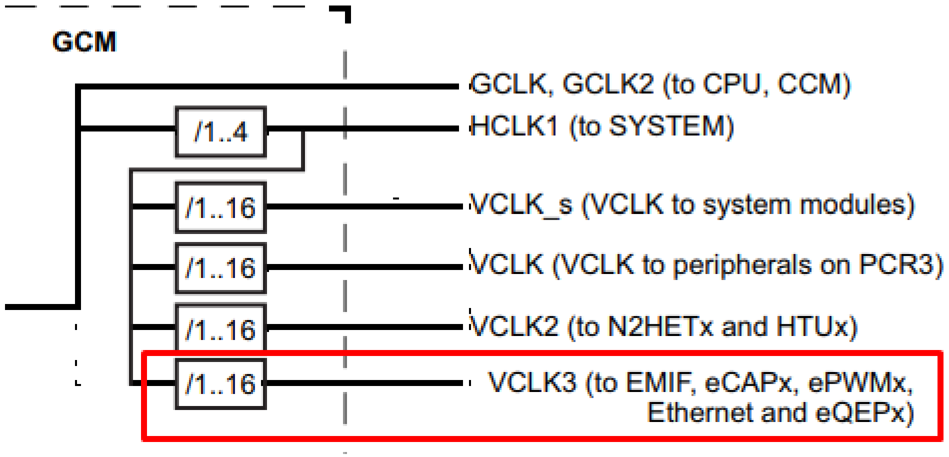
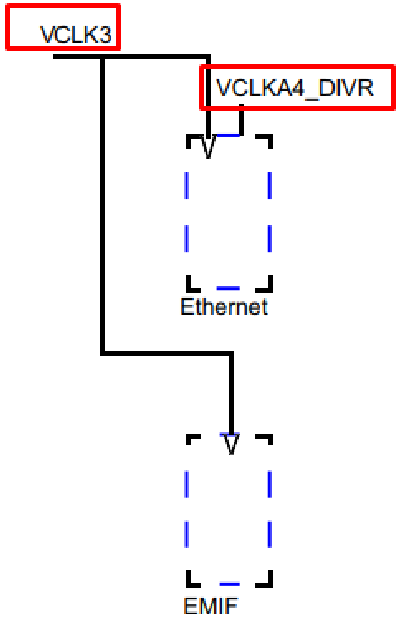


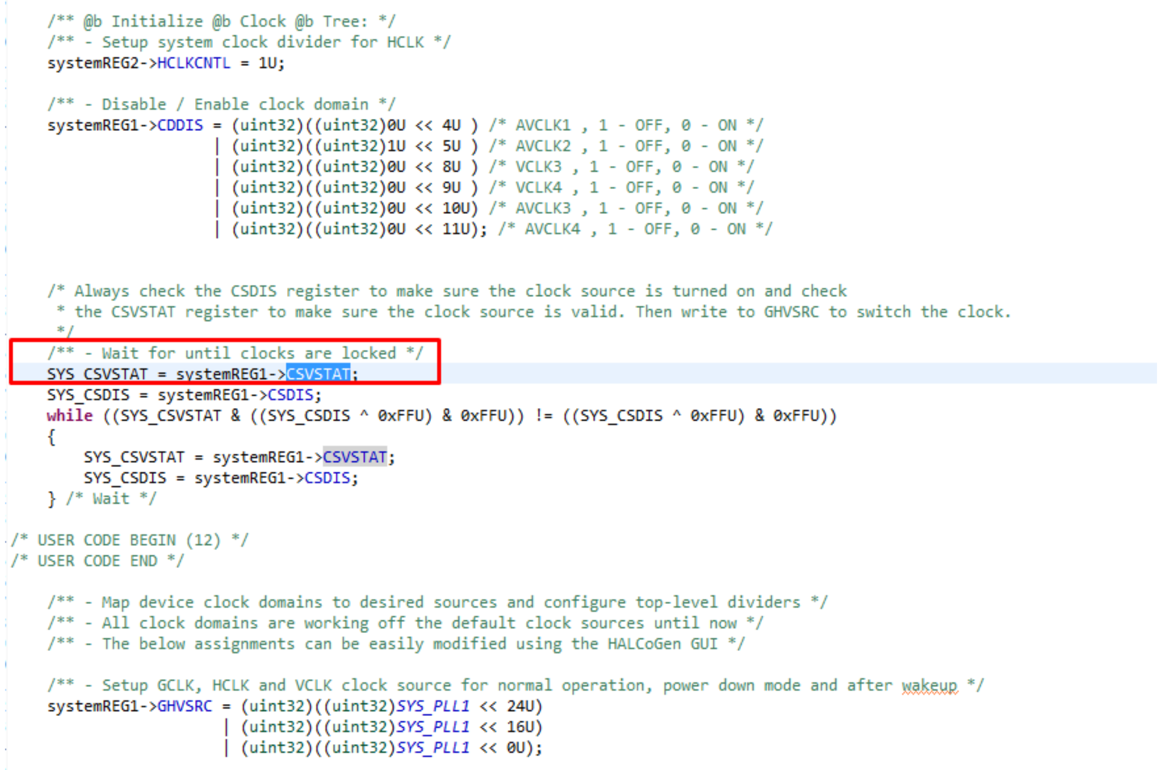
Figure 6-6. Device Clock Dom


GCM
GCLK, GCLK2 (to CPU, CCM)
/1..4
HCLK1 (to SYSTEM)
/1..16
VCLK_s (VCLK to system modules)
/1..16
VCLK (VCLK to peripherals on PCR3)
/1..16
VCLK2 (to N2HETx and HTUx)
/1..16
VCLK3 (to EMIF, eCAPx, ePWMx, Ethernet and eQEPx)

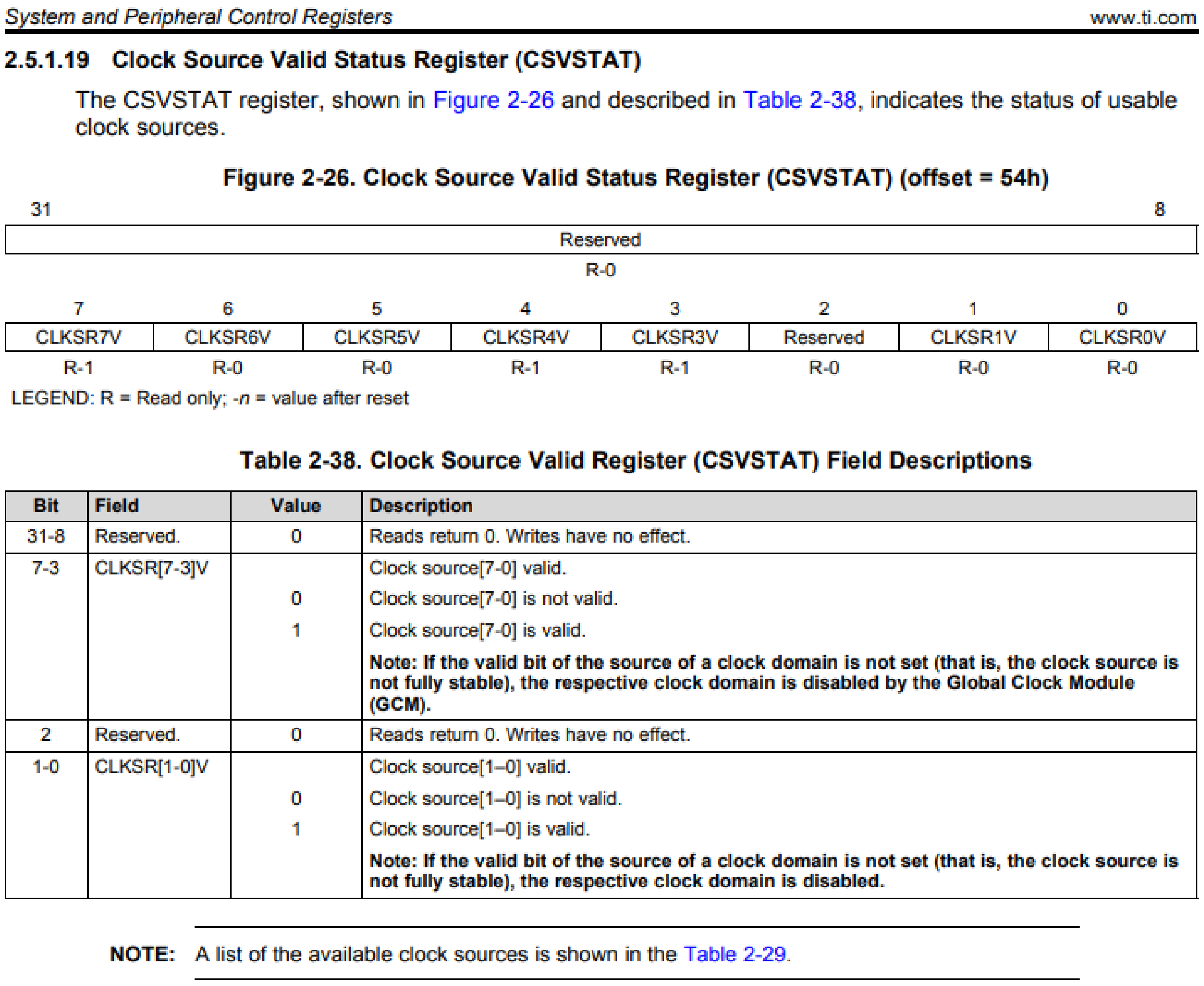


VCLKA1 - ON
VCLKA2 - OFF
VCLK3 – ON
VCLKA4 - ON

1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – mapClocks())



```
    /** @b Initialize @b Clock @b Tree: */
    /** - Setup system clock divider for HCLK */
    systemREG2->HCLKCNTL = 1U;

    /** - Disable / Enable clock domain */
    systemREG1->CDDIS = (uint32)((uint32)0U << 4U ) /* AVCLK1 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)1U << 5U ) /* AVCLK2 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 8U ) /* VCLK3 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 9U ) /* VCLK4 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 10U) /* AVCLK3 , 1 - OFF, 0 - ON */
                      | (uint32)((uint32)0U << 11U); /* AVCLK4 , 1 - OFF, 0 - ON */


    /* Always check the CSDIS register to make sure the clock source is turned on and check
     * the CSVSTAT register to make sure the clock source is valid. Then write to GHVSRC to switch the clock.
     */
    /** - Wait for until clocks are locked */
    SYS_CSVSTAT = systemREG1->CSVSTAT;
    SYS_CSDIS = systemREG1->CSDIS;
    while ((SYS_CSVSTAT & ((SYS_CSDIS ^ 0xFFU) & 0xFFU)) != ((SYS_CSDIS ^ 0xFFU) & 0xFFU))
    {
        SYS_CSVSTAT = systemREG1->CSVSTAT;
        SYS_CSDIS = systemREG1->CSDIS;
    } /* Wait */

/* USER CODE BEGIN (12) */
/* USER CODE END */

    /** - Map device clock domains to desired sources and configure top-level dividers */
    /** - All clock domains are working off the default clock sources until now */
    /** - The below assignments can be easily modified using the HALCoGen GUI */

    /** - Setup GCLK, HCLK and VCLK clock source for normal operation, power down mode and after wakeup */
    systemREG1->GHVSRC = (uint32)((uint32)SYS_PLL1 << 24U)
                       | (uint32)((uint32)SYS_PLL1 << 16U)
                       | (uint32)((uint32)SYS_PLL1 << 0U);
```



System and Peripheral Control Registers www.ti.com

2.5.1.19 Clock Source Valid Status Register (CSVSTAT)

The CSVSTAT register, shown in Figure 2-26 and described in Table 2-38, indicates the status of usable clock sources.

Figure 2-26. Clock Source Valid Status Register (CSVSTAT) (offset = 54h)

31–8
Reserved
R-0

7	6	5	4	3	2	1	0
CLKSR7V	CLKSR6V	CLKSR5V	CLKSR4V	CLKSR3V	Reserved	CLKSR1V	CLKSR0V
R-1	R-0	R-0	R-1	R-1	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 2-38. Clock Source Valid Register (CSVSTAT) Field Descriptions

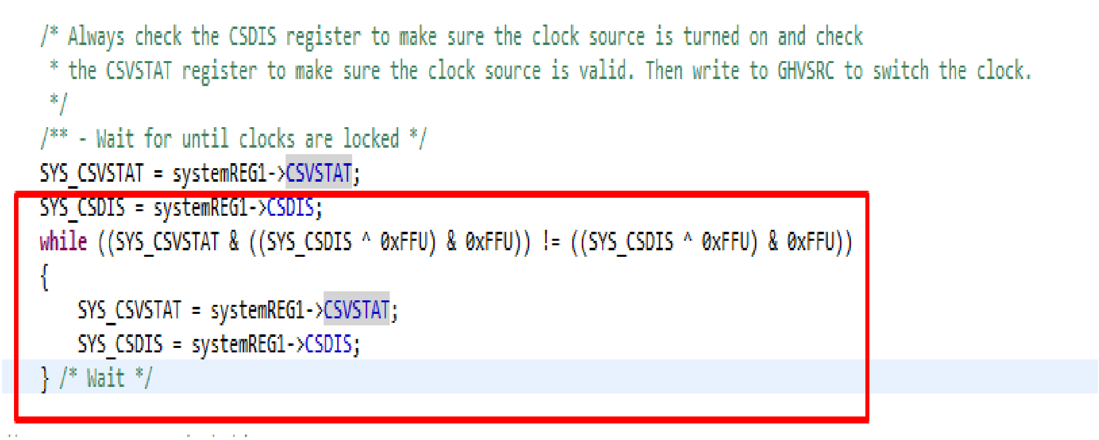
Bit	Field	Value	Description
31-8	Reserved.	0	Reads return 0. Writes have no effect.
7-3	CLKSR[7-3]V		Clock source[7-0] valid.
		0	Clock source[7-0] is not valid.
		1	Clock source[7-0] is valid.
			Note: If the valid bit of the source of a clock domain is not set (that is, the clock source is not fully stable), the respective clock domain is disabled by the Global Clock Module (GCM).
2	Reserved.	0	Reads return 0. Writes have no effect.
1-0	CLKSR[1-0]V		Clock source[1–0] valid.
		0	Clock source[1–0] is not valid.
		1	Clock source[1–0] is valid.
			Note: If the valid bit of the source of a clock domain is not set (that is, the clock source is not fully stable), the respective clock domain is disabled.

NOTE: A list of the available clock sources is shown in the Table 2-29.

SYS_CSVSTAT= systemREG1->CSVSTAT

CDDIS에서 clock domain을 설정했고 CSVSTAT으로 clock source에 valid한지 아닌지를 확인한다.

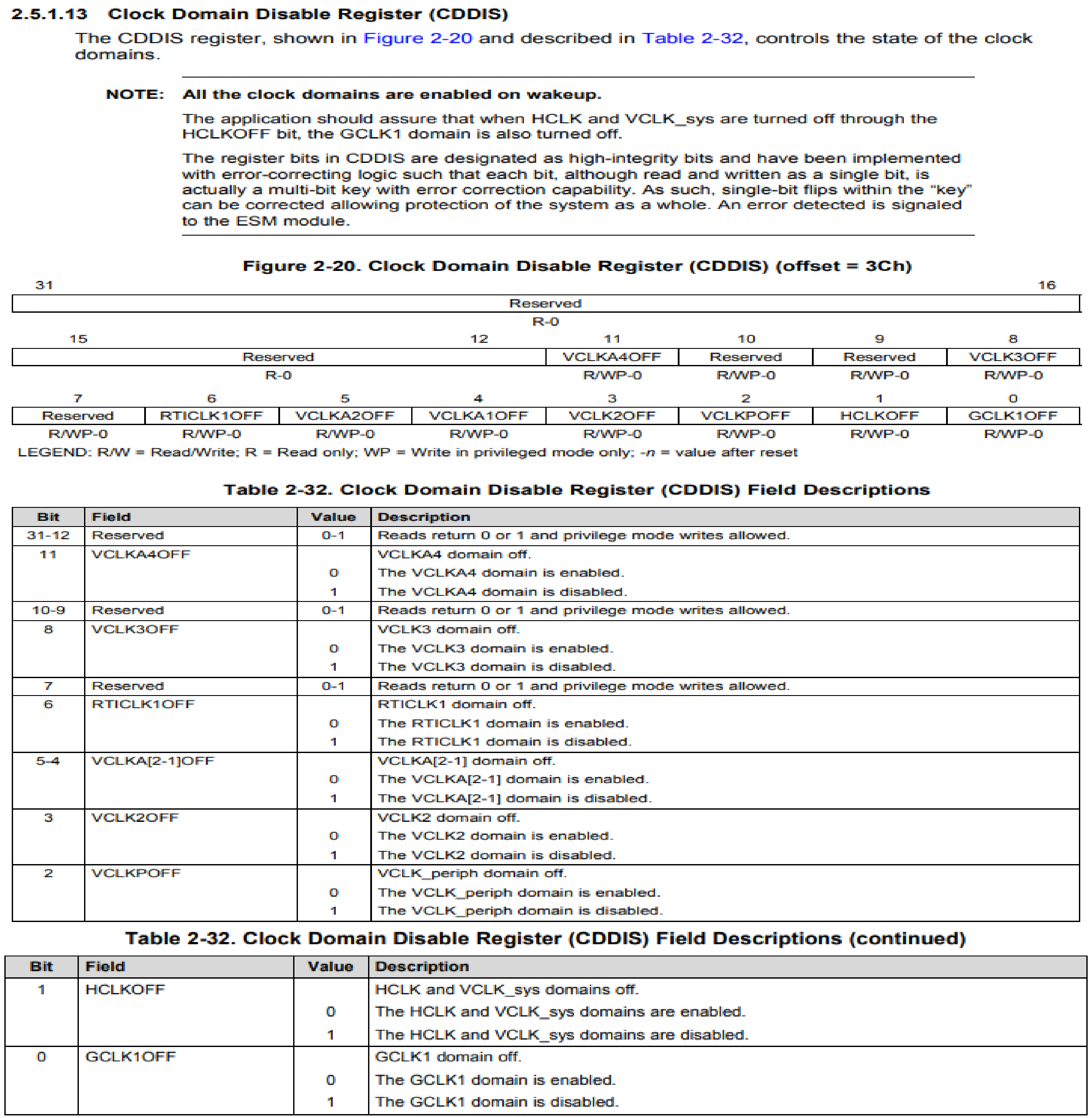
1. Cortex-R5F Hercules Safety MCU – boot code 분석 (systemInit() – mapClocks())



```
/* Always check the CSDIS register to make sure the clock source is turned on and check
 * the CSVSTAT register to make sure the clock source is valid. Then write to GHVSRC to switch the clock.
 */
/** - Wait for until clocks are locked */
SYS_CSVSTAT = systemREG1->CSVSTAT;
SYS_CSDIS = systemREG1->CSDIS;
while ((SYS_CSVSTAT & ((SYS_CSDIS ^ 0xFFU) & 0xFFU)) != ((SYS_CSDIS ^ 0xFFU) & 0xFFU))
{
    SYS_CSVSTAT = systemREG1->CSVSTAT;
    SYS_CSDIS = systemREG1->CSDIS;
} /* Wait */
```

위에서 셋팅한 CSDIS를 가져와서 SYS_CSDIS변수에 넣고 SYS_CSVSTAT과 SYS_CSDIS를 위와 같은 방식으로 논리 연산을 한다.

While문으로 계속 루프를 돌면서 클락 소스가 안정화 될 때까지 기다린다.



2.5.1.13 Clock Domain Disable Register (CDDIS)

The CDDIS register, shown in Figure 2-20 and described in Table 2-32, controls the state of the clock domains.

NOTE: All the clock domains are enabled on wakeup.

The application should assure that when HCLK and VCLK_sys are turned off through the HCLKOFF bit, the GCLK1 domain is also turned off.

The register bits in CDDIS are designated as high-integrity bits and have been implemented with error-correcting logic such that each bit, although read and written as a single bit, is actually a multi-bit key with error correction capability. As such, single-bit flips within the "key" can be corrected allowing protection of the system as a whole. An error detected is signaled to the ESM module.

Figure 2-20. Clock Domain Disable Register (CDDIS) (offset = 3Ch)

31–16
Reserved
R-0

15–12	11	10	9	8
Reserved	VCLKA4OFF	Reserved	Reserved	VCLK3OFF
R-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
Reserved	RTICLK1OFF	VCLKA2OFF	VCLKA1OFF	VCLK2OFF	VCLKPOFF	HCLKOFF	GCLK1OFF
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 2-32. Clock Domain Disable Register (CDDIS) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0-1	Reads return 0 or 1 and privilege mode writes allowed.
11	VCLKA4OFF		VCLKA4 domain off.
		0	The VCLKA4 domain is enabled.
		1	The VCLKA4 domain is disabled.
10-9	Reserved	0-1	Reads return 0 or 1 and privilege mode writes allowed.
8	VCLK3OFF		VCLK3 domain off.
		0	The VCLK3 domain is enabled.
		1	The VCLK3 domain is disabled.
7	Reserved	0-1	Reads return 0 or 1 and privilege mode writes allowed.
6	RTICLK1OFF		RTICLK1 domain off.
		0	The RTICLK1 domain is enabled.
		1	The RTICLK1 domain is disabled.
5-4	VCLKA[2-1]OFF		VCLKA[2-1] domain off.
		0	The VCLKA[2-1] domain is enabled.
		1	The VCLKA[2-1] domain is disabled.
3	VCLK2OFF		VCLK2 domain off.
		0	The VCLK2 domain is enabled.
		1	The VCLK2 domain is disabled.
2	VCLKPOFF		VCLK_periph domain off.
		0	The VCLK_periph domain is enabled.
		1	The VCLK_periph domain is disabled.

Table 2-32. Clock Domain Disable Register (CDDIS) Field Descriptions (continued)

Bit	Field	Value	Description
1	HCLKOFF		HCLK and VCLK_sys domains off.
		0	The HCLK and VCLK_sys domains are enabled.
		1	The HCLK and VCLK_sys domains are disabled.
0	GCLK1OFF		GCLK1 domain off.
		0	The GCLK1 domain is enabled.
		1	The GCLK1 domain is disabled.