# TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 문한나

mhn97@naver.com

# Cortex-M4

```
; Reset handler
Reset_Handler   PROC
                EXPORT  Reset_Handler           [WEAK]
        IMPORT  SystemInit
        IMPORT  __main

                ;FPU settings
                LDR     R0, =0xE000ED88         ; Enable CP10,CP11
                LDR     R1,[R0]
                ORR     R1,R1,#(0xF << 20)
                STR     R1,[R0]

                LDR     R0, =SystemInit
                BLX     R0
                LDR     R0, =__main
                BX      R0
                ENDP

; Dummy Exception Handlers (infinite loops which can be modified)

NMI_Handler     PROC
                EXPORT  NMI_Handler             [WEAK]
                B       .
                ENDP
HardFault_Handler₩
                PROC
                EXPORT  HardFault_Handler       [WEAK]
                B       .
```

부팅하면 POR → 리셋 핸들러 동작

```
                ;FPU settings
                LDR     R0, =0xE000ED88         ; Enable CP10,CP11
```

R0 레지스터에 값을 넣는다. 0xE000ED88 을
=을 쓰면 바로 값을 넣는다.

Cortex-M4 Technical Reference Manual
검색 : 0xE000ED88

**Table 4-1 System control registers (continued)**

| Address | Name | Type | Reset | Description |
|---|---|---|---|---|
| 0xE000ED64 | ID_ISAR1 | RO | 0x02112000 | Instruction Set Attributes Register 1 |
| 0xE000ED68 | ID_ISAR2 | RO | 0x21232231 | Instruction Set Attributes Register 2 |
| 0xE000ED6C | ID_ISAR3 | RO | 0x01111131 | Instruction Set Attributes Register 3 |
| 0xE000ED70 | ID_ISAR4 | RO | 0x01310102 | Instruction Set Attributes Register 4 |
| 0xE000ED88 | CPACR | RW | - | Coprocessor Access Control Register |
| 0xE000EF00 | STIR | WO | 0x00000000 | Software Triggered Interrupt Register |

a. Bits [10:8] are reset to zero. The ENDIANNESS bit, bit [15], can reset to either state, depending on the implementation.
b. BFAR and MFAR are the same physical register. Because of this, the BFARVALID and MFAEVALID bits are mutually exclusive.

Cortex-M4 Device Generic User Guide
검색 : CPACR

### 4.6.1 Coprocessor Access Control Register

The CPACR register specifies the access privileges for coprocessors. See the register summary in *Cortex-M4F floating-point system registers* for its attributes. The bit assignments are:

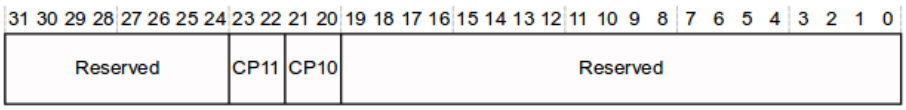



**Table 4-50 CPACR register bit assignments**

| Bits | Name | Function |
|---|---|---|
| [31:24] | - | Reserved. Read as Zero, Write Ignore. |
| [2*n*+1:2*n*] for *n* values10 and 11 | CP*n* | Access privileges for coprocessor *n*. The possible values of each field are:<br>0b00 = Access denied. Any attempted access generates a NOCP UsageFault.<br>0b01 = Privileged access only. An unprivileged access generates a NOCP fault.<br>0b10 = Reserved. The result of any access is Unpredictable.<br>0b11 = Full access. |
| [19:0] | - | Reserved. Read as Zero, Write Ignore. |

CPACR 레지스터는 보조 프로세서에 대한 액세스 권한을 지정.
CP11 CP10 레지스터 → 부동소수점을 사용할 수 있게 해준다.

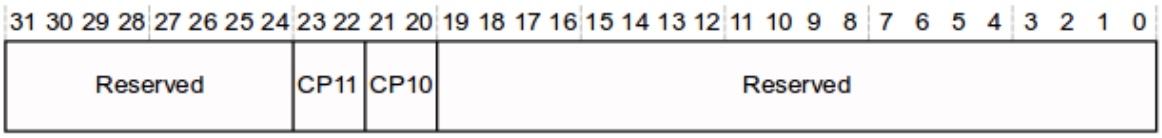
```
LDR    R1,[R0]
```

R1 레지스터에 R0 을 넣음
[]는 거기 주소를 가서 데이터를 가져오라는 뜻이다.

```
ORR    R1,R1,#(0xF << 20)
```

ORR → 논리 or

1111
0000 0000 1111 0000 0000 0000 0000 0000




20 비트 밀어서 or 연산으로 23 22 21 20 비트에 1 을 세팅해준다.

```
STR    R1,[R0]
```

STR → 레지스터에서 메모리로 저장 r0 에 주소로 가서 거기에 있는 값을 변경한다.

따라서 20 21 22 23 비트에 1 세팅됨

0b11 → Full access 커널도 유저도 쓸 수 있다.

```
LDR     R0, =SystemInit
BLX     R0
```

함수의 이름은 주소이다. systeminit 이라는 함수포인터를 저장한다.
Blx 로 함수 이동

```
/**
  * @brief  Micro Controller System을 설정한다.
  *         Embedded Flash Interface, PLL을 초기화하고 SystemFrequency 변수를 갱신한다.
  * @param  None
  * @retval None
  */
void SystemInit(void)
{
  /* RCC Clock 구성을 Default Reset State로 reset(재설정)한다. */
  /* 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;

  /* 모든 Interrupt를 비활성화한다. */
  RCC->CIR = 0x00000000;

#ifdef DATA_IN_ExtSRAM
  SystemInit_ExtMemCtl();
#endif /* DATA_IN_ExtSRAM */

  /* System Clock Source, PLL 곱셈기, 나눗셈기, AHB/APBx Prescalers와 Flash 설정을 구성한다. */
  SetSysClock();

  /* Offset Address를 더한 Vector Table 위치를 구성한다. */
#ifdef VECT_TAB_SRAM
  SCB->VTOR = SRAM_BASE | VECT_TAB_OFFSET; /* 내부 SRAM에 Vector Table 재배치 */
#else
  SCB->VTOR = FLASH_BASE | VECT_TAB_OFFSET; /* 내부 FLASH(NAND)에 Vector Table 재배치 */
#endif
}
```

```
RCC→CR |= (unit32_t)0x00000001
```

## 6.3 RCC registers

Refer to *Section 1.1: List of abbreviations for registers* for a list of abbreviations used in register descriptions.

### 6.3.1 RCC clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 XX83 where X is undefined.

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | PLLSAIRDY | PLLSAION | PLLI2SRDY | PLLI2SON | PLLRDY | PLLON | Reserved | | | | CSSON | HSEBYP | HSERDY | HSEON |
| | | r | rw | r | rw | r | rw | | | | | rw | rw | r | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HSICAL[7:0] | | | | | | | | HSITRIM[4:0] | | | | | Res. | HSIRDY | HSION |
| r | r | r | r | r | r | r | r | rw | rw | rw | rw | rw | | r | rw |

0 번비트를 1 로 세팅함(16MHZ 내부주파수를 세팅한다.)

Bit 0 **HSION**: Internal high-speed clock enable

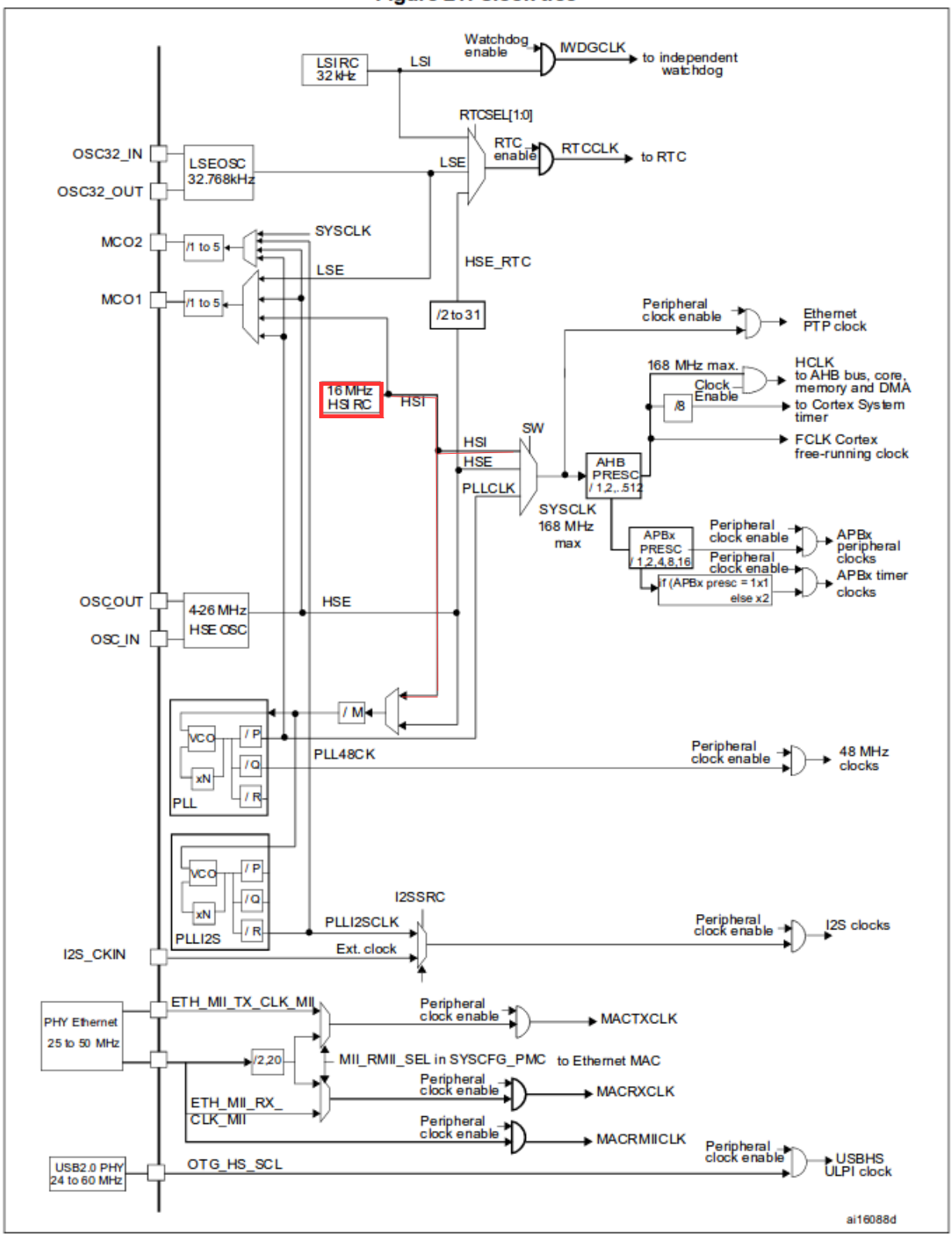
Set and cleared by software.

Set by hardware to force the HSI oscillator ON when leaving the Stop or Standby mode or in case of a failure of the HSE oscillator used directly or indirectly as the system clock. This bit cannot be cleared if the HSI is used directly or indirectly as the system clock.

0: HSI oscillator OFF

1: HSI oscillator ON

**Figure 21. Clock tree**



```
RCC→CFGR = 0x00000000;
```

### 6.3.3 RCC clock configuration register (RCC_CFGR)

Address offset: 0x08

Reset value: 0x0000 0000

Access: 0 ≤ wait state ≤ 2, word, half-word and byte access

1 or 2 wait states inserted only if the access occurs during a clock source switch.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MCO2 | | MCO2 PRE[2:0] | | | MCO1 PRE[2:0] | | | I2SSCR | MCO1 | | RTCPRE[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PPRE2[2:0] | | | PPRE1[2:0] | | | Reserved | | HPRE[3:0] | | | | SWS1 | SWS0 | SW1 | SW0 |
| rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | r | r | rw | rw |

클락제어관련 레지스터
0~31 비트까지 전부 초기화를 해준다.

```
RCC→CR &= (unit32_t) 0xFEF6FFFF
```

## 6.3 RCC registers

Refer to *Section 1.1: List of abbreviations for registers* for a list of abbreviations used in register descriptions.

### 6.3.1 RCC clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 XX83 where X is undefined.

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | PLLSAI RDY | PLLSAI ON | PLLI2S RDY | PLLI2S ON | PLLRD Y | PLLON | Reserved | | | | CSS ON | HSE BYP | HSE RDY | HSE ON |
| | | r | rw | r | rw | r | rw | | | | | rw | rw | r | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HSICAL[7:0] | | | | | | | | HSITRIM[4:0] | | | | | Res. | HSI RDY | HSION |
| r | r | r | r | r | r | r | r | rw | rw | rw | rw | rw | | r | rw |

```
0x    F     E     F     6     F     F     F     F
0x 1111 1110 1111 0110 1111 1111 1111 1111
```

24번, 19번, 16번 비트를 0으로 세팅하겠다.

Bit 24 **PLLON**: Main PLL (PLL) enable

Set and cleared by software to enable PLL.

Cleared by hardware when entering Stop or Standby mode. This bit cannot be reset if PLL clock is used as the system clock.

0: PLL OFF

1: PLL ON

```
<System Clock(syscall)>
1. HSI Oscillator Clock
- 32 kHz Internal RC, 32.768 kHz External Crystal
2. HSE Oscillator Clock
3. PLL Clock
```

클럭은 내부, 외부, pll로 총 3종류이다.
MCU칩은 회로로 설계되어있는데 16MHZ로 고정되어있다.
더 큰 걸 사용하고 싶을 때 pll을 쓴다.
sw로 원하는 것을 뽑아서 쓸 수 있다.

Bit 19 **CSSON**: Clock security system enable

Set and cleared by software to enable the clock security system. When CSSON is set, the clock detector is enabled by hardware when the HSE oscillator is ready, and disabled by hardware if an oscillator failure is detected.

0: Clock security system OFF (Clock detector OFF)

1: Clock security system ON (Clock detector ON if HSE oscillator is stable, OFF if not)

보안 시스템 레지스터를 끈다

Bit 16 **HSEON**: HSE clock enable

Set and cleared by software.

Cleared by hardware to stop the HSE oscillator when entering Stop or Standby mode. This bit cannot be reset if the HSE oscillator is used directly or indirectly as the system clock.

0: HSE oscillator OFF

1: HSE oscillator ON

HSE → 외부클락이 준비될 때 1, 준비가 안됐을 때 0

```
RCC→PLLCFGR = 0x24003010;
```

### 7.3.2 RCC PLL configuration register (RCC_PLLCFGR)

Address offset: 0x04

Reset value: 0x2400 3010

Access: no wait state, word, half-word and byte access.

This register is used to configure the PLL clock outputs according to the formulas:

- $f_{(VCO\ clock)} = f_{(PLL\ clock\ input)} \times (PLLN / PLLM)$
- $f_{(PLL\ general\ clock\ output)} = f_{(VCO\ clock)} / PLLP$
- $f_{(USB\ OTG\ FS,\ SDIO,\ RNG\ clock\ output)} = f_{(VCO\ clock)} / PLLQ$

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | PLLQ3 | PLLQ2 | PLLQ1 | PLLQ0 | Reserved | PLLSRC | Reserved | | | | PLLP1 | PLLP0 |
| | | | | rw | rw | rw | rw | | rw | | | | | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | PLLN | | | | | | | | | PLLM5 | PLLM4 | PLLM3 | PLLM2 | PLLM1 | PLLM0 |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

초기값을 넣어준다. (초기화를 한다)

```
RCC→CR &= (unit32_t)0xFFFBFFFF
```

## 6.3 RCC registers

Refer to *Section 1.1: List of abbreviations for registers* for a list of abbreviations used in register descriptions.

### 6.3.1 RCC clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 XX83 where X is undefined.

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | PLLSAIRDY | PLLSAION | PLLI2SRDY | PLLI2SON | PLLRDY | PLLON | Reserved | | | | CSSON | HSEBYP | HSERDY | HSEON |
| | | r | rw | r | rw | r | rw | | | | | rw | rw | r | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HSICAL[7:0] | | | | | | | | HSITRIM[4:0] | | | | | Res. | HSIRDY | HSION |
| r | r | r | r | r | r | r | r | rw | rw | rw | rw | rw | | r | rw |

```
0x  F    F    F    B    F    F    F    F
0x 1111 1111 1111 1011 1111 1111 1111 1111
```

Bit 18 **HSEBYP**: HSE clock bypass

Set and cleared by software to bypass the oscillator with an external clock. The external clock must be enabled with the HSEON bit, to be used by the device.
The HSEBYP bit can be written only if the HSE oscillator is disabled.
0: HSE oscillator not bypassed
1: HSE oscillator bypassed with an external clock

```
RCC→CIR = 0x00000000;
```

### 7.3.4 RCC clock interrupt register (RCC_CIR)

Address offset: 0x0C

Reset value: 0x0000 0000

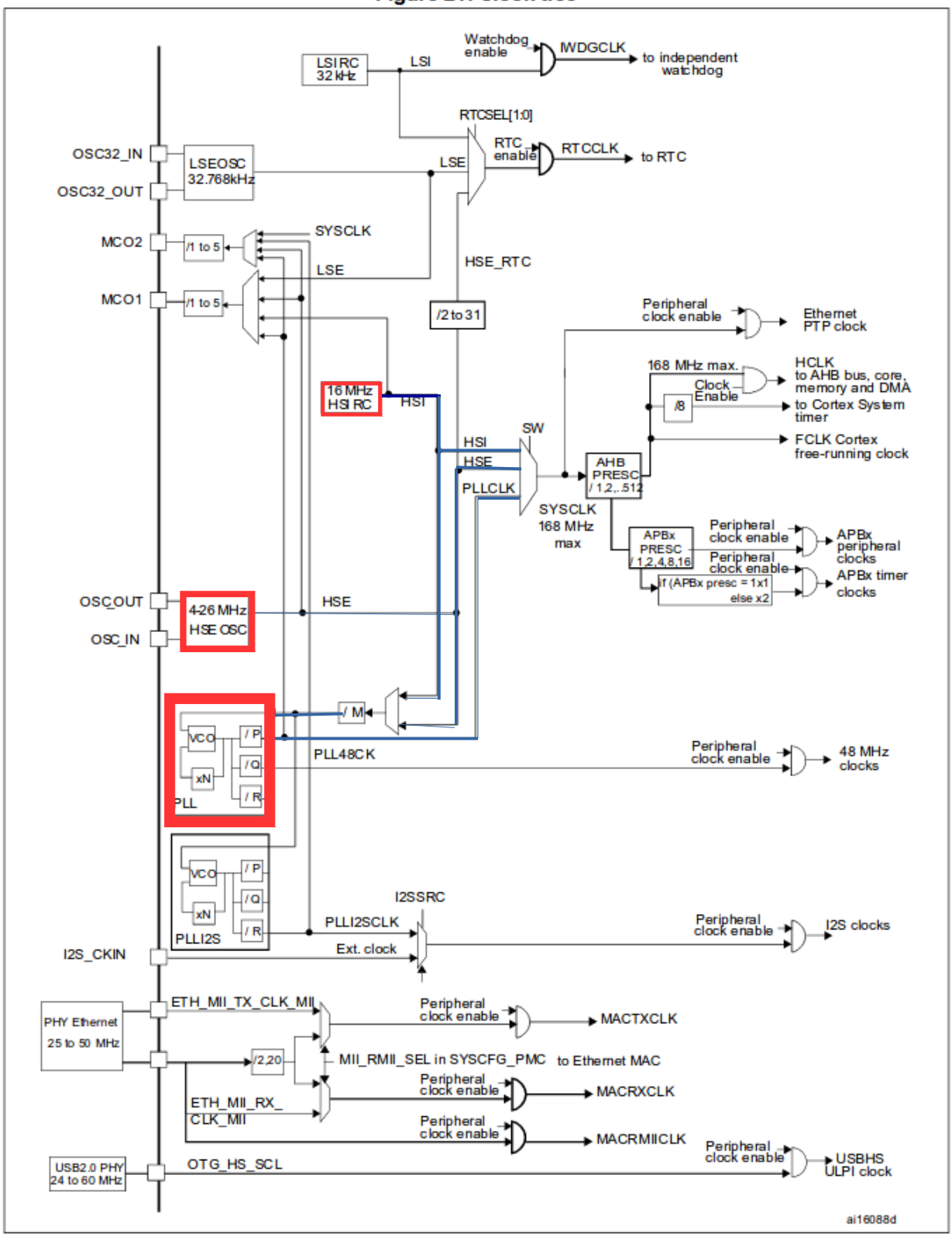
Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | CSSC | Reserved | PLLI2S RDYC | PLL RDYC | HSE RDYC | HSI RDYC | LSE RDYC | LSI RDYC |
| | | | | | | | | w | | w | w | w | w | w | w |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | PLLI2S RDYIE | PLL RDYIE | HSE RDYIE | HSI RDYIE | LSE RDYIE | LSI RDYIE | CSSF | Reserved | PLLI2S RDYF | PLL RDYF | HSE RDYF | HSI RDYF | LSE RDYF | LSI RDYF |
| | | rw | rw | rw | rw | rw | rw | r | | r | r | r | r | r | r |

모든 인터럽트를 비활성화 한다!

**Figure 21. Clock tree**



SetSysClock();

```
/**
  * @brief  System Clock Source, PLL 곱셈기 & 나눗셈기, AHB/APBx Prescalers와 Flash 설정을 구성한다.
  * @Note   이 함수는 RCC Clock 구성을 Default Reset State로 Reset하기 위해 단 한 번만 호출된다.
  * @param  None
  * @retval None
  */
static void SetSysClock(void)
{
/*****************************************************************************/
/*                PLL (clocked by HSE)을 System Clock Source로 사용한다.                */
/*****************************************************************************/
  __IO uint32_t StartUpCounter = 0, HSEStatus = 0;

  /* HSE를 활성화 */
  RCC->CR |= ((uint32_t)RCC_CR_HSEON);

  /* Time Out되서 종료되거나 HSE가 종료될때까지 대기한다. */
  do
  {
    HSEStatus = RCC->CR & RCC_CR_HSERDY;                    HSE_STARTUP_TIMEOUT의 값은 0x05000
    StartUpCounter++;
  } while((HSEStatus == 0) && (StartUpCounter != HSE_STARTUP_TIMEOUT));

  if ((RCC->CR & RCC_CR_HSERDY) != RESET)          RESET = 0
  {
    HSEStatus = (uint32_t)0x01;
  }
  else
  {
    HSEStatus = (uint32_t)0x00;
  }
```

RCC→CR != ((unit32_t)RCC_CR_HSEON

Bit 16 **HSEON**: HSE clock enable
Set and cleared by software.
Cleared by hardware to stop the HSE oscillator when entering Stop or Standby mode. This bit cannot be reset if the HSE oscillator is used directly or indirectly as the system clock.
0: HSE oscillator OFF
1: HSE oscillator ON

HSE 를 1 로 만들어 라인이 활성화 된다.(외부클럭)

HSEStatus = RCC→CR & RCC_CR_HSERDY

외부클럭이 준비가 되었는지 계속 상태를 확인함

while(HSEStatus == 0) && (StartUpCounter != HSE_STARTUP_TIMEOUT));

만약 HSEStatus 에 1 이 세팅되고, HSE_STARTUP_TIMEOUT 이 아니라면
while 을 빠져나간다.

RCC→CR & RCC_CR_HSERDY != RESET

Bit 17 **HSERDY**: HSE clock ready flag

Set by hardware to indicate that the HSE oscillator is stable. After the HSEON bit is cleared, HSERDY goes low after 6 HSE oscillator clock cycles.

0: HSE oscillator not ready

1: HSE oscillator ready

17 번비트의 값을 읽어오겠다.
RESET 은 0 으로 정의되어 있다.
만약 0 이아니면 외부 오실레이터를 사용할꺼다

```
  if (HSEStatus == (uint32_t)0x01)
  {
    /* High Performance Mode를 활성화하고, System Frequency를 168 MHz로 올린다. */
    RCC->APB1ENR |= RCC_APB1ENR_PWREN;
    PWR->CR |= PWR_CR_PMODE;
    RCC->APB1ENR |= RCC_APB1ENR_PWREN;

    /* 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;

    /* main PLL을 구성한다. */
    RCC->PLLCFGR = PLL_M | (PLL_N << 6) | (((PLL_P >> 1) -1) << 16) |
                   (RCC_PLLCFGR_PLLSRC_HSE) | (PLL_Q << 24);

    /* main PLL을 활성화 */
    RCC->CR |= RCC_CR_PLLON;

    /* main PLL이 준비될때까지 대기한다. */
    while((RCC->CR & RCC_CR_PLLRDY) == 0)
    {
    }

    /* Flash Prefetch, Instruction Cache, Data Cache를 구성하고 대기 상태 */
    FLASH->ACR = FLASH_ACR_ICEN |FLASH_ACR_DCEN |FLASH_ACR_LATENCY_5WS;

    /* System Clock Source로 main PLL을 선택한다. */
    RCC->CFGR &= (uint32_t)((uint32_t)~(RCC_CFGR_SW));
    RCC->CFGR |= RCC_CFGR_SW_PLL;

    /* System Clock Source로 main PLL이 사용될때까지 대기한다. */
    while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS ) != RCC_CFGR_SWS_PLL);
    {
    }
  }
  else
  { /* HSE가 Start-Up에 실패하면 Application은 잘못된 Clock을 구성할 것이다.
       사용자(학생분들)가 이러한 오류를 다루기 위한 Code를 이곳에 추가하면 된다. */
  }
}
```

```
RCC→APB1ENR |= RCC_APB1ENR_PWREN;
```

### 6.3.13 RCC APB1 peripheral clock enable register (RCC_APB1ENR)

Address offset: 0x40

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART8 EN | UART7 EN | DAC EN | PWR EN | Reser-ved | CAN2 EN | CAN1 EN | Reser-ved | I2C3 EN | I2C2 EN | I2C1 EN | UART5 EN | UART4 EN | USART 3 EN | USART 2 EN | Reser-ved |
| rw | rw | rw | rw | | rw | rw | | rw | rw | rw | rw | rw | rw | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPI3 EN | SPI2 EN | Reserved | | WWDG EN | Reserved | | TIM14 EN | TIM13 EN | TIM12 EN | TIM7 EN | TIM6 EN | TIM5 EN | TIM4 EN | TIM3 EN | TIM2 EN |
| rw | rw | | | rw | | | rw | rw | rw | rw | rw | rw | rw | rw | rw |



Bit 28 **PWREN:** Power interface clock enable

This bit is set and cleared by software.

0: Power interface clock disabled

1: Power interface clock enable

APB : 분주비이다. 주파수를 나눠준다

```
/* 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;
```

### 6.3.3 RCC clock configuration register (RCC_CFGR)

Address offset: 0x08

Reset value: 0x0000 0000

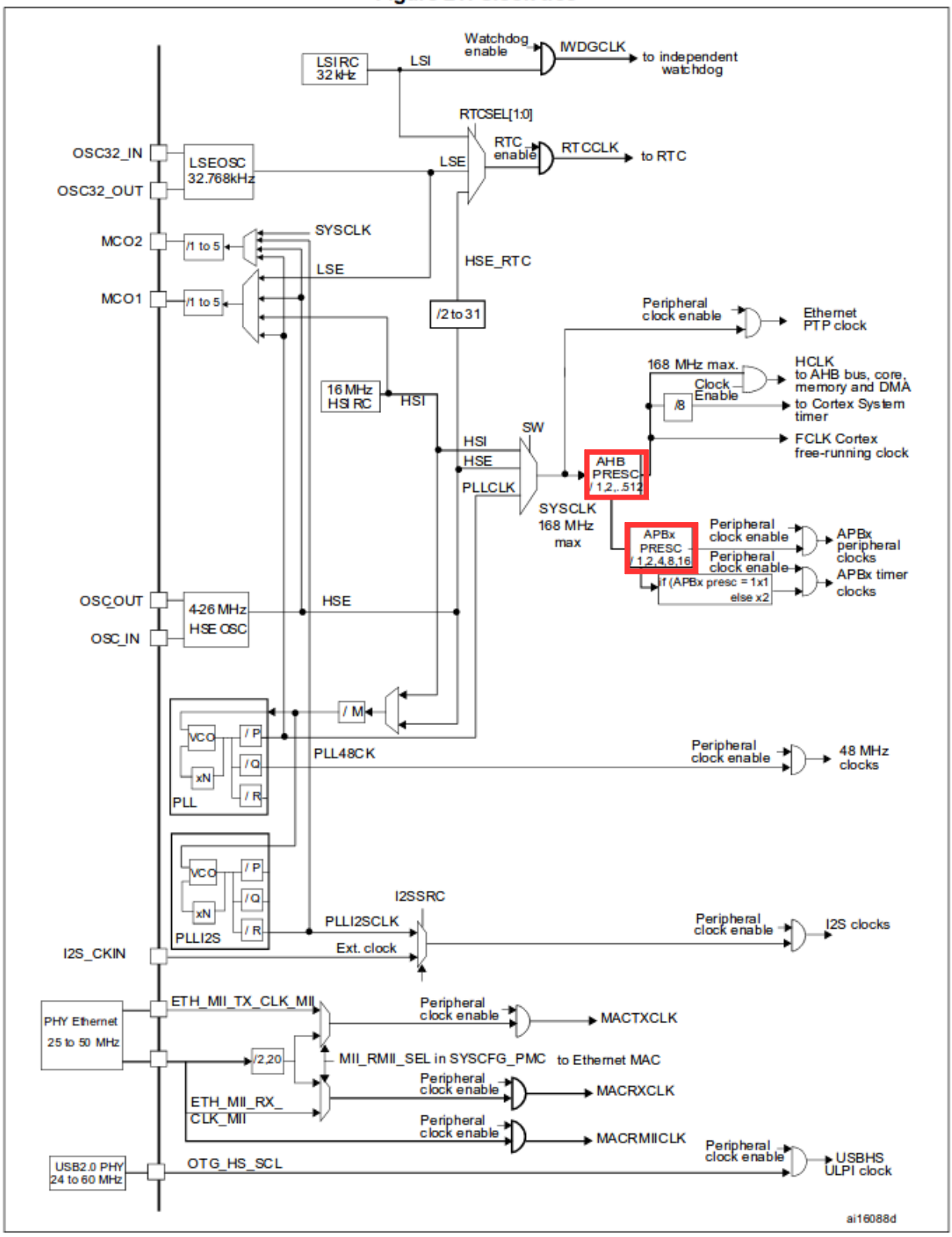
Access: 0 ≤ wait state ≤ 2, word, half-word and byte access

1 or 2 wait states inserted only if the access occurs during a clock source switch.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MCO2 | | MCO2 PRE[2:0] | | | MCO1 PRE[2:0] | | | I2SSCR | MCO1 | | RTCPRE[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PPRE2[2:0] | | | PPRE1[2:0] | | | Reserved | | HPRE[3:0] | | | | SWS1 | SWS0 | SW1 | SW0 |
| rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | r | r | rw | rw |

**Figure 21. Clock tree**




네트워크는 스케일러 안하고 사용
peripheral 주파수 적어야 해서 나눠서 쓴다

Bits 15:13 **PPRE2:** APB high-speed prescaler (APB2)

Set and cleared by software to control APB high-speed clock division factor.

**Caution:** The software has to set these bits correctly not to exceed 90 MHz on this domain. The clocks are divided with the new prescaler factor from 1 to 16 AHB cycles after PPRE2 write.

0xx: AHB clock not divided
100: AHB clock divided by 2
101: AHB clock divided by 4
110: AHB clock divided by 8
111: AHB clock divided by 16

Bits 12:10 **PPRE1:** APB Low speed prescaler (APB1)

Set and cleared by software to control APB low-speed clock division factor.

**Caution:** The software has to set these bits correctly not to exceed 45 MHz on this domain. The clocks are divided with the new prescaler factor from 1 to 16 AHB cycles after PPRE1 write.

0xx: AHB clock not divided
100: AHB clock divided by 2
101: AHB clock divided by 4
110: AHB clock divided by 8
111: AHB clock divided by 16

Bits 9:8 Reserved, must be kept at reset value.