

**TI DSP,Xilinx zynq FPGA,MCU 및
Xilinx
zynq FPGA 프로그래밍 전문가 과정**

강사-INNOVA LEE(이상훈)

Gccompil3r@gmail.com

학생-윤지원

Yoonjw7894@naver.com

```

; 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_HandlerW
PROC
EXPORT HardFault_Handler        [WEAK]
B .

```

LDR R0, =0XE000ED88

//0XE000ED88 의 주소값을 R0 에 삽입

LDR R1,[R0]

//R0 에 있는 데이터값을 R1 에 넣는다 ,즉 R1=0XE000ED88 이 들어있다.

ORR R1,R1,#(0XF<<20)

//0XF=0X1111 에서 SHIFT 를 20 번한다 그럼 0X0000 0000 0000 0000 0000 1111 과 R1 에 있는 0XE000ED88 과 OR 연산을 한다.

Cp10cp11:다른 프로세스로 접근할 권한을 주는 역할을 하고 부동 소수점 표현을 활성화 해준다.

PFU(부동 소수점 표현)

```

/**
 * @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)0xFE6FFFFF;

    /* Reset PLLCFGR register */
    RCC->PLLCFGR = 0x24003010;

    /* Reset HSEBYP bit */
    RCC->CR &= (uint32_t)0xFFBFFFFF;

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

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

    /* System Clock Source, PLL 공셈기, 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
}

```

RM0090 Reset and clock control for STM32F405xx/07xx and STM32F415xx/17xx(RCC)

7.3 RCC registers

Refer to [Section 1.1: List of abbreviations for registers](#) for a list of abbreviations used in register descriptions.

7.3.1 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				PLLI2S RDY	PLLI2S ON	PLL RDY	PLL ON	Reserved				CSS ON	HSE BYP	HSE RDY	HSE ON
				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

Bits 31:28 Reserved, must be kept at reset value.

Bit 27 **PLLI2SRDY**: PLLI2S clock ready flag

Set by hardware to indicate that the PLLI2S is locked.

0: PLLI2S unlocked

1: PLLI2S locked

Bit 26 **PLLI2SON**: PLLI2S enable

Set and cleared by software to enable PLLI2S.

Cleared by hardware when entering Stop or Standby mode.

0: PLLI2S OFF

1: PLLI2S ON

Bit 25 **PLLRDY**: Main PLL (PLL) clock ready flag

Set by hardware to indicate that PLL is locked.

0: PLL unlocked

1: PLL locked

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

위 사진은 코드에서 RCC_CR 의 각 비트마다의 동작을 알려준다.

처음 $RCC \rightarrow CR = 0X00000001$ 이 부분은

$0X\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001$ 이 비트로 0 인 부분은

RESET 으로 설정하고 마지막 0 비트에는 1 로 설정을 했다

이 부분은 단순한 부팅 작업용으로 생각한다.

그 다음 $RCC \rightarrow CR \&=0XFEF6FFFF$

$0XFEF6FFFF=1111\ 1110\ 1111\ 0110\ 1111\ 1111\ 1111\ 1111$ 으로 표현되는데 위 사진을 보면 PLLON,CSS ON HSE ON 을 RESET 시킨다.

HSERDY:HSE clock ready flag 의 역할

회로를 동작을 시킬때 전원 전압을 넣는데 그 순간 엄청나게 많은 전류와 전압이 들어오는데 그 전압이 설정된 전압으로 될때까지 0 으로 대기상태로 있다가 안정권이 되면 flag 가 1 로 바뀌어 실행되는 것 같습니다.

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

7.3.3 RCC clock configuration register (RCC_CFGR)

Address offset: 0x08

Reset value: 0x0000 0000

Access: $0 \leq \text{wait state} \leq 2$, 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]			I2SSC R	MCO1		RTCPRE[4:0]				
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

$RCC \rightarrow CFGR = 0X00000000$ 로 모든 비트를 RESET 시킨다.

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:

- $f_{(VCO \text{ clock})} = f_{(PLL \text{ clock input})} \times (PLL_N / PLL_M)$
- $f_{(PLL \text{ general clock output})} = f_{(VCO \text{ clock})} / PLL_P$
- $f_{(USB OTG FS, SDIO, RNG \text{ clock output})} = f_{(VCO \text{ clock})} / PLL_Q$

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	PLL_N									PLLM5	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:28, Reserved, must be kept at reset value

RCC → PLLCFGR=2400 3010 으로 초기값을 설정

PLL 은 회로 안정화의 역할과 파형의 평균을 정해서 그 파형이 계속나올수 있게 해주는 역할을 하는것 같다.

PLLCLK,HSI,HSE 이 세 가지 비트가 SW(스위치)역활을 한다. 그 이유는 이 녀석들은 회로 안정이 되었을때 그때서야 동작을 하기때문에 애네의 동작이 sw 를 연것이나 다름이 없고 HSI 는 회로 안정보다는 부팅역활이며 HSE 가 바로 사용이 되지않을때 하드웨어의 동작가에 의해 설정이 된다고 해석이 된다.

```

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. */
}
}

```

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 84 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 42 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

위에 코드를 보면 system frequency 를 168MH 로 설정하고
그 다음 코드를 보면

RCC → CRFG= RCC_CFGR_HPRE_DIV1 이 부분은 168MH 를 그대로 사용하겠다는 의미이고

RCC → CRFG= RCC_CFGR_PPRE_DIV2 는 위에 사진을 보면 PPRE2 의 정격 주파수가 84MH 를 넘지 않는다는 조건을 만족해야 할때 2로 나눠주면 조건이 맞기에 DIV2 를 설정해 준것이고 아래 코드도 이와 동일하게 정격 주파수를 맞추기 위해 DIV4 로 설정한 것이다. 이것을 맞추는 이유는 기준 보다 많은 주파수가 들어온다면 기준보다 많은 전류가 들어온다는 것이고 이것은 과전류로 인해 회로가 망가지게 되는 일이 발생한다. 그래서 무조건 이 조건은 만족을 시켜줘야 한다.