Xilinx
Zynq FPGA
TI DSP MCU 기반의
프로그래밍 및 회로 설계 전문가

강사 이상훈
gcccompil3r@gmail.com

학생 김민호
minking12@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_Handler₩
                PROC
                EXPORT  HardFault_Handler         [WEAK]
                B       .
```

LDR R0,=0xE000ED88

R0 에 0xE000ED88 Memory to Register 한다.

Table 4-49 Cortex-M4F floating-point system registers

Address	Name	Type	Reset	Description
0xE000ED88	CPACR	RW	0x00000000	*Coprocessor Access Control Register*
0xE000EF34	FPCCR	RW	0xC0000000	*Floating-point Context Control Register* on page 4-49
0xE000EF38	FPCAR	RW	-	*Floating-point Context Address Register* on page 4-50
-	FPSCR	RW	-	*Floating-point Status Control Register* on page 4-50
0xE000EF3C	FPDSCR	RW	0x00000000	*Floating-point Default Status Control Register* on page 4-52

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:

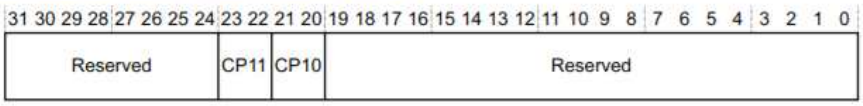


Table 4-50 CPACR register bit assignments

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: 0b00 = Access denied. Any attempted access generates a NOCP UsageFault. 0b01 = Privileged access only. An unprivileged access generates a NOCP fault. 0b10 = Reserved. The result of any access is Unpredictable. 0b11 = Full access.
[19:0]	-	Reserved. Read as Zero, Write Ignore.

부동소수점 연산을 가능케 하는 CP11 , CP10 비트는 23 ~ 20.

LDR R1,[R0]

(R0 특정 값을 R1에 LOAD => 0000,0000,1111,0000,0000,0000,0000,0000)

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

(0xF 20만큼 shift 한 후 R1에 넣고 기존 R1 값에 | 연산.

기존 R1이나 shift 한 값이 같아서 그냥 같다 ...^^)

STR R1,[R0]

(R1 값을 R0 주소에 저장.)

LDR R0,=SystemInit

```
/**
  * @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 |= (uint32_t)0x00000001;

(저기 맨 마지막 1비트만 켠다.)

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

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.

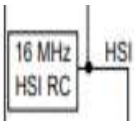
0: HSI oscillator OFF

1: HSI oscillator ON

HSI 1 -> 발진기 ON , 내부 클록 사용 준비.

RCC->CFGR = 0x00000000;

(CFGR 레지스터의 값을 0으로 초기화. CFGR : 클록 환경설정 레지스터.)



RCC->CR &= (uint32_t)0xFEF6FFFF;

(E 부분과 6만 끈다. => 24번 비트, 19,16 비트)

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

(PLLON,CSSON,HSEON 을 0으로 초기화.)

(PLLON -> 0 씹고주파 클록 0으로)

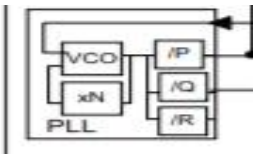
Bit 24 **PLLON**: Main PLL (PLL) enable

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



(CSSON -> 0 클록의 보안 0 으로)

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.

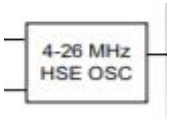
0: Clock security system OFF (Clock detector OFF)

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

(HSEON -> 외부 클록 0으로, 외부클록은 4~ 26 MHZ)

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. 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



RCC->PLLCFGR=0x24003010;

(위상 고정 루프 ? 값을 초기 값으로 세팅.)

6.3.2 RCC PLL configuration register (RCC_PLLCFGR)

Address offset: 0x04

Reset value: 0x2400 3010

RCC->CR&=(uint32_t)0xFFFBFFFF;

(B부분만 0으로 세팅. -> 18번 비트 HSEBYP)

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

(외부 클록을 못 가게 한 다 ?)

RCC->CIR=0x0000 0000;

(인터럽트를 초기 값으로 세팅. 생략할시 이상한 값이 나올 수 있다.)

6.3.4 RCC clock interrupt register (RCC_CIR)

Address offset: 0x0C

Reset value: 0x0000 0000

SetSysClock();

(PLL 곱셈기, 나눗셈기, AHB/APBxPrescalers와 Flash 설정을 구성)

```
/**
  * @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;
    StartUpCounter++;
  } while((HSEStatus == 0) && (StartUpCounter != HSE_STARTUP_TIMEOUT));

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

HSE_STARTUP_TIMEOUT의 값은 0x05000

RESET = 0

StartupCounter=0, HSEStatus=0;

(변수선언과 함께 초기 값 설정.)

RCC->CR |=((uint32_t)RCC_CR_HSEON);

(HSE 활성화.)

do while 문이기 때문에 무조건 한번은 실행.

HSEStatus=RCC->CR & RCC_CR_HSERDY;

(클록 초기 발현 시는 불규칙하지만 점차 안정화 되고 안정화가 된다면 기계 내부스스로 HSERDY에 1을 넣어준다.)

HSEStatus 값이 0이 아니기 때문에 while 문을 탈출한다.

if((RCC->CR & RCC_CR_HSERDY)!=RESET)

{

HSEStatus=(uint32_t)0x01;

}

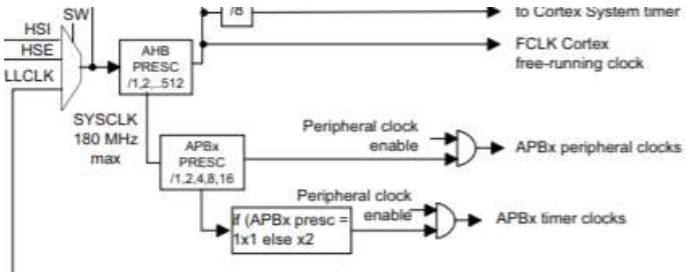
(RESET 값은 0, 0이 아니기 때문에 if 문 실행.)

```
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;

(APB-(PRESC 분주비) : 클록을 나눠서 주파수를 낮춘다.)

(peripheral :주파수가 완전 낮아야 하는 애들. 분주비 2번 거침.)



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

(Power interface -> 1로. 근데 무슨 설정인지 모르겠음...)

PWR->CR |=PWR_CR_PMODE;

(PWR_CR은 부팅 시 플로팅 방지를 위함이다.)

5.4.1 PWR power control register (PWR_CR) for STM32F405xx/07xx and STM32F415xx/17xx

Address offset: 0x00

Reset value: 0x0000 4000 (reset by wakeup from Standby mode)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	VOS	Reserved				FPDS	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
	rw					rw	rw	rw	rw	rw	rw	w	w	rw	rw

(파워 컨트롤 레지스터를 PMODE로 .... 모르겠음... 아마 초기화 ?)

RCC->CFGR | = RCC_CFGR_HPRE_DIV1;

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

Bits 7:4 **HPRE:** AHB prescaler

Set and cleared by software to control AHB clock division factor.

Caution: The clocks are divided with the new prescaler factor from 1 to 16 AHB cycles after HPRE write.

Caution: The AHB clock frequency must be at least 25 MHz when the Ethernet is used.

0xxx: system clock not divided
1000: system clock divided by 2
1001: system clock divided by 4
1010: system clock divided by 8
1011: system clock divided by 16
1100: system clock divided by 64
1101: system clock divided by 128
1110: system clock divided by 256
1111: system clock divided by 512

(분주비를 1로 설정 -> div by 1은 없다.근데 왜 써놓았을까 ?)

(AHB : Advanced High performance Bus)

RCC->CFGR |= RCC_CFGR_PPRE2_DIV2;

(분주비를 PPRE2 DIV2 설정 -> 100)

RCC->CFGR |= RCC_CFGR_PPRE2_DIV4;

(분주비를 PPRE2 DIV4 설정 -> 101)

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.

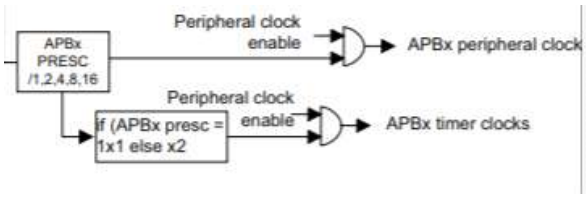
0xx: AHB clock not divided
100: AHB clock divided by 2
101: AHB clock divided by 4
110: AHB clock divided by 8
111: AHB clock divided by 16

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



(x는 밑에 더 있다는 뜻이다.)

RCC->CFGR |= RCC_CFGR_PPRE2_DIV2;

(0000/0000/0000/0000/1000/0000/0000/0000)

PPRE DIV2 는 15번째 비트부터 13번째 비트까지 -> 100

RCC->CFGR |= RCC_CFGR_PPRE2_DIV4;

(0000/0000/0000/0000/1001/0100/0000/0000)

PPRE DIV4 는 12번째 비트부터 10번째 비트까지 -> 101