Xilinx Zynq FPGA,TI DSP, MCU 기반의 프로그래밍 전문가 과정

날 짜 : 2018 . 5. 2

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

MCU□ □ □.□

CORETEX-R5 <BOOT CODE>

```
trimLPO()
      LPO_TRIM_VALUE → (F008000U) >> 16    //□ □ □ □ □ □ □.. □ □ □

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

      //clock monitor control Register □ □.


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


mapClocks()
      systemREG2->HCLKCNTL = 1U;    //GCLK1 □ □ □ □2□□ □ □.□

    /** - 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
*/
      // AVCLK 1,3,4,  VCLK 3,4  AVCLK 2

      SYS_CSVSTAT = systemREG1→ CSVSTAT;  // □ □ □ □PLL□ □ □ □ 1□ □□ □ □ .□ □
      SYS_CSDIS = systemREG1→ CSDIS;      // Flex_Rey → can □ □ □ □ .□ □

      while ((SYS_CSVSTAT & ((SYS_CSDIS ^ 0xFFU) & 0xFFU)) != ((SYS_CSDIS ^
0xFFU) & 0xFFU))
      {
        SYS_CSVSTAT = systemREG1→ CSVSTAT;      //
        SYS_CSDIS = systemREG1→ CSDIS;          // □ □ □ □ □ □ □ □ □
      } /* Wait */

      systemREG1->GHVSRC = (uint32)((uint32)SYS_PLL1 << 24U)
                         | (uint32)((uint32)SYS_PLL1 << 16U)
                         | (uint32)((uint32)SYS_PLL1 << 0U);
      // clock source 1 □ PLL1 □ □.

    /** - Setup RTICLK1 and RTICLK2 clocks */
      systemREG1->RCLKSRC = (uint32)((uint32)1U << 24U)        /* RTI2 divider
      (Not applicable for lock-step device)  */
                          | (uint32)((uint32)SYS_VCLK << 16U) /* RTI2 clock source
      (Not applicable for lock-step device) */
                          | (uint32)((uint32)1U << 8U)         /* RTI1 divider */
                          | (uint32)((uint32)SYS_VCLK << 0U); /* RTI1 clock source
*/
```

```
      systemREG1->VCLKASRC = (uint32)((uint32)SYS_VCLK << 8U)
                           | (uint32)((uint32)SYS_VCLK << 0U);
      // □ □ □ □ □ □ □ □ □ □ □ □ □

      systemREG1->SYSPC1 = 0U;

    /** - set ECLK pins default output value */
      systemREG1->SYSPC4 = 0U;

    /** - set ECLK pins output direction */
      systemREG1->SYSPC2 = 1U;

    /** - set ECLK pins open drain enable */
      systemREG1->SYSPC7 = 0U;

    /** - set ECLK pins pullup/pulldown enable */
      systemREG1->SYSPC8 = 0U;

    /** - set ECLK pins pullup/pulldown select */
      systemREG1->SYSPC9 = 1U;

    /** - Setup ECLK */
      systemREG1->ECPCNTL = (uint32)((uint32)0U << 24U)
                          | (uint32)((uint32)0U << 23U)
                          | (uint32)((uint32)(8U - 1U) & 0xFFFFU);
```

MPU	memory protection unit
dsb	data synchronized berrier
isb	instruction synchronized berrier