

TI DSP, MCU 및 Xilinx Zynq FPGA

프로그래밍 전문가 과정

2018-05-09 (50회차)

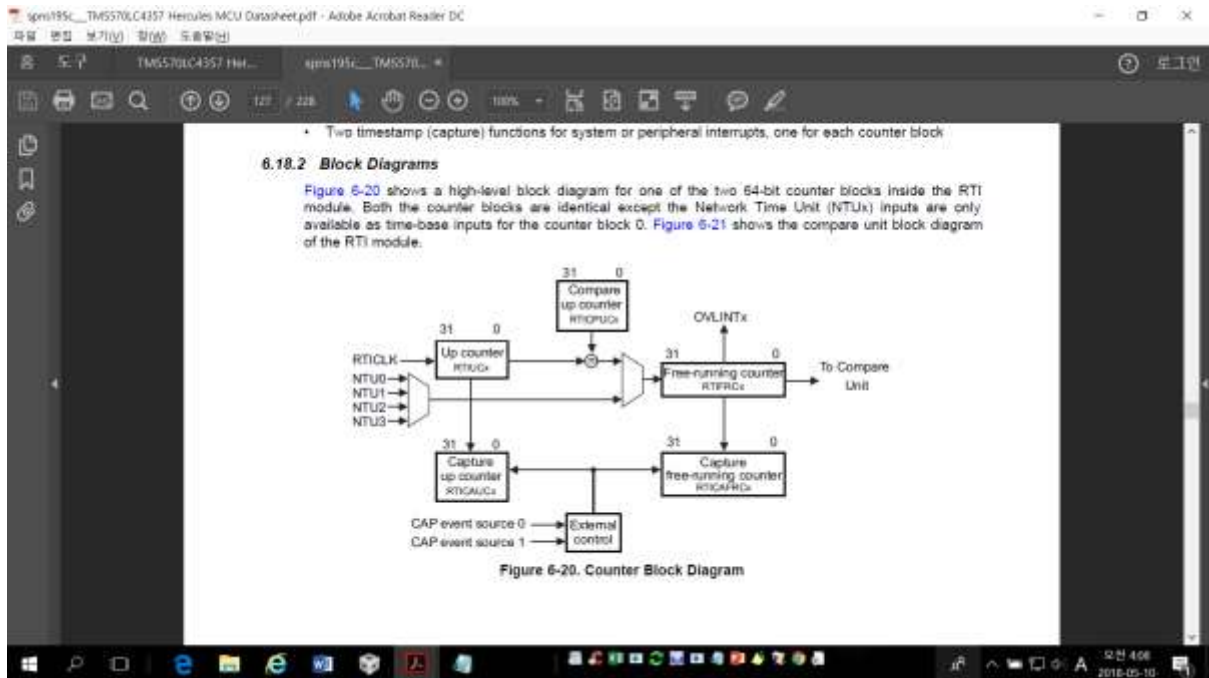
강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - 정유경

ucong@naver.com

RTI_BLINK 코드분석



```
void rtiInit(void)
```

```
{
```

```
    /** - Setup NTU source, debug options and disable both counter blocks */
```

```
    rtiREG1->GCTRL = (uint32)((uint32)0x5U << 16U) | 0x00000000U;
```

NTU1번 input signal 이 external timebase로 사용된다

```
    /** - Setup timebase for free running counter 0 */
```

```
    rtiREG1->TBCTRL = 0x00000000U;
```

free running counter 0가 failing clock on the NTU signal is detected. 되었을때 자동으로 증가하지않도록 한다.

free running counter 0 (RTIFRC0)가 internal up counter 0 (RTIUC0)에 의해 clocked된다

```
    /** - Enable/Disable capture event sources for both counter blocks */
```

```
    rtiREG1->CAPCTRL = 0U | 0U;
```

RTIUC나 RTIFRC를 외부인터럽트가 트리거 할 때 캡처하는 기능은 사용하지 않는다

```
    /** - Setup input source compare 0-3 */
```

```
rtiREG1->COMPCTRL = 0x00001000U | 0x00000100U | 0x00000000U | 0x00000000U;
```

source for the compare registers를 결정한다.

Compare select 3, 0 모두 RTIFRC0와 비교된다.

```
/** - Reset up counter 0 */
```

```
rtiREG1->CNT[0U].UCx = 0x00000000U;
```

```
/** - Reset free running counter 0 */
```

```
rtiREG1->CNT[0U].FRCx = 0x00000000U;
```

리셋 후 카운트 한다는 의미

```
/** - Setup up counter 0 compare value
```

```
* - 0x00000000: Divide by 2^32
```

```
* - 0x00000001-0xFFFFFFFF: Divide by (CPUC0 + 1)
```

```
*/
```

```
rtiREG1->CNT[0U].CPUCx = 7U;
```

Divide by (CPUC0 + 1)라고 위에 쓰여 있으므로 8분주한다는 의미이다.

```
/** - Reset up counter 1 */
```

```
rtiREG1->CNT[1U].UCx = 0x00000000U;
```

```
/** - Reset free running counter 1 */
```

```
rtiREG1->CNT[1U].FRCx = 0x00000000U;
```

```
/** - Setup up counter 1 compare value
```

```
* - 0x00000000: Divide by 2^32
```

```
* - 0x00000001-0xFFFFFFFF: Divide by (CPUC1 + 1)
```

```
*/
```

```
rtiREG1->CNT[1U].CPUCx = 7U;
```

마찬가지고 리셋 후 카운트 시키고 8분주 한다.

따라서 $f_{FRC0} = \frac{RTICLK}{(RTICPUC0+1)}$ f_{FRC0} , f_{FRC1} 은 $75 / 8 = 9.375\text{MHz}$ 로 동작

```
/** - Setup compare 0 value. This value is compared with selected free running counter. */
```

```
rtiREG1->CMP[0U].COMPx = 9375U;
```

free running counter와 비교할 compare 0의 값을 설정한다.

```
/** - Setup update compare 0 value. This value is added to the compare 0 value on each compare match. */
```

```
rtiREG1->CMP[0U].UDCPx = 9375U;
```

매치 후 compare 0 value의 기준을 다시 설정해준다.

```
/** - Clear all pending interrupts */
```

```
rtiREG1->INTFLAG = 0x0007000FU;
```

16,17,18번 비트 활성화 : Timebase interrupt flag, Free running counter 0 overflow interrupt flag, Free running counter 1 overflow interrupt flag를 초기화한다.

```
/** - Disable all interrupts */
```

```
rtiREG1->CLEARINTENA = 0x00070F0FU;
```

compare interrupt 끝

compare DMA request , DMA는 사용하지 않으므로 끝

```
/** @note This function has to be called before the driver can be used.\\n
```

```
* This function has to be executed in privileged mode.\\n
```

```
* This function does not start the counters.
```

```
*/
```

```
/* USER CODE BEGIN (3) */
```

```
/* USER CODE END */
```

```
}
```

