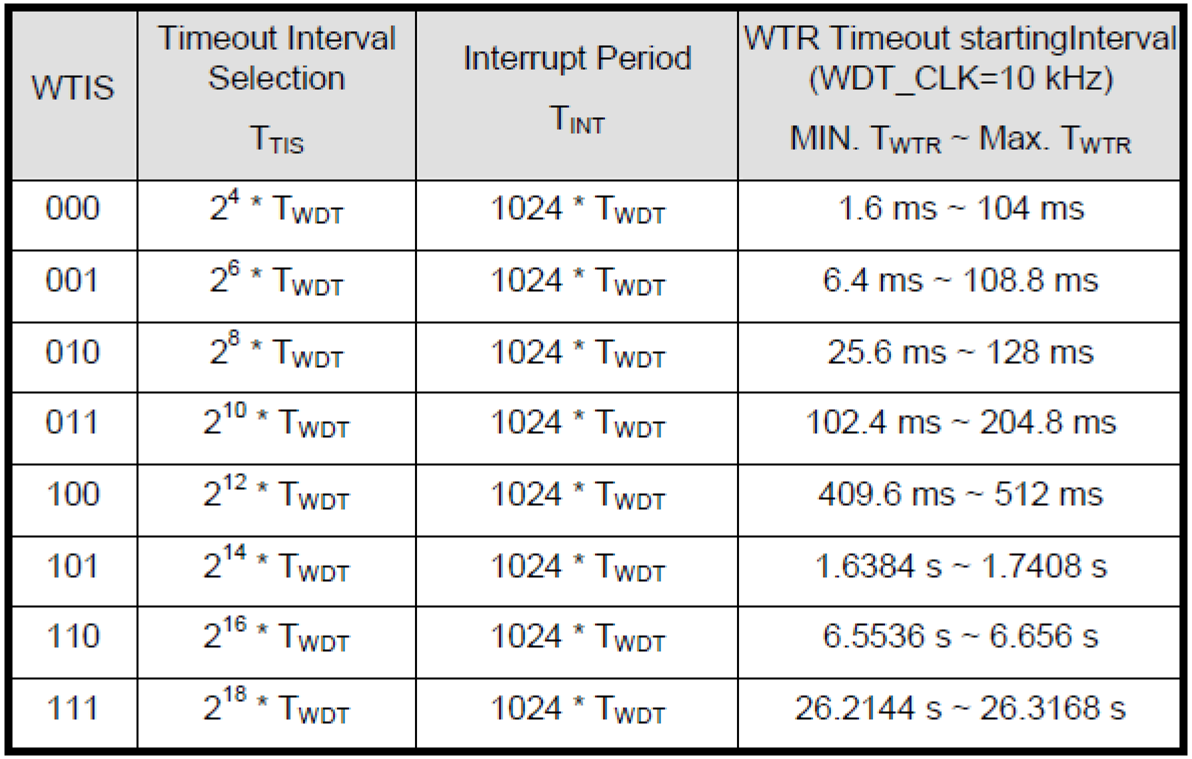
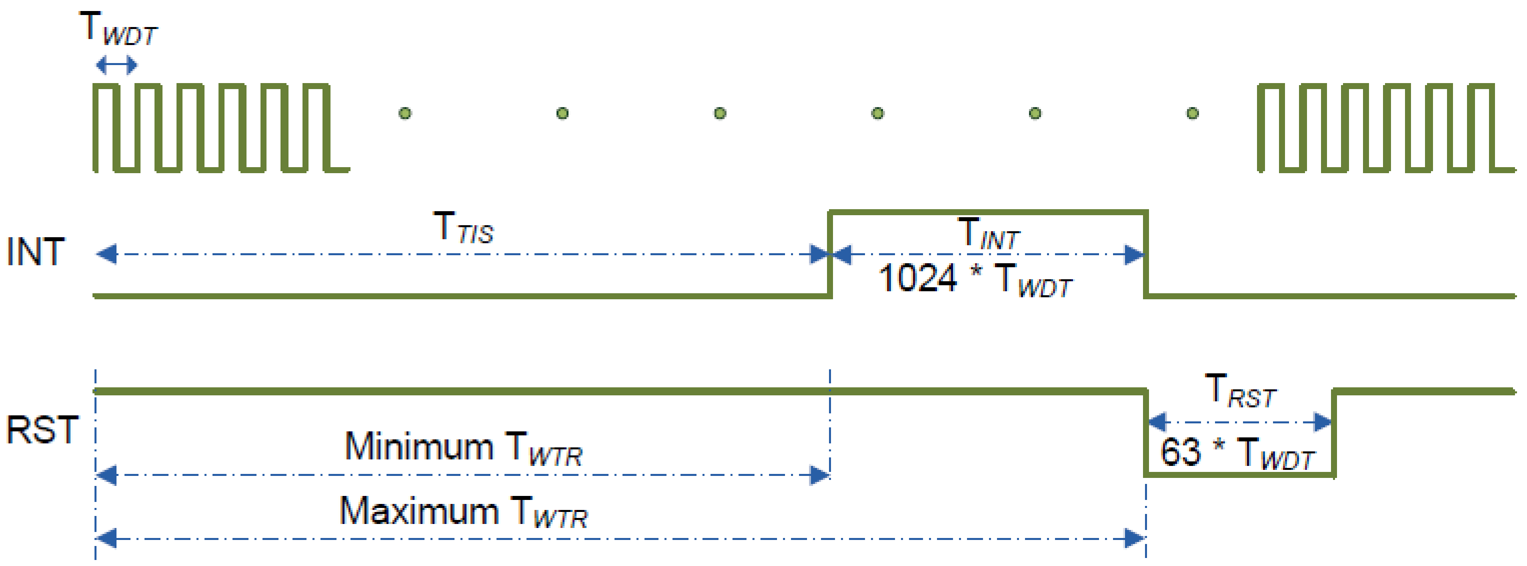
# Watchdog Timer (WDT) :

- The **purpose** of Watchdog Timer is to perform a **system reset** when system runs into an unknown state
- another function -> **wake-up** chip from **power down mode**.
- The watchdog timer includes an 18-bit free running counter with programmable time-out intervals.

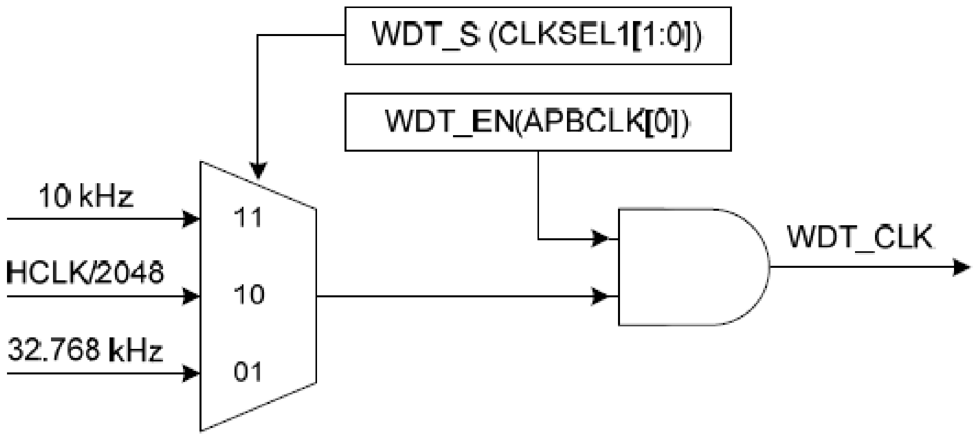


| WTIS | Timeout Interval Selection $T_{TIS}$ | Interrupt Period $T_{INT}$ | WTR Timeout startingInterval (WDT_CLK=10 kHz) MIN. $T_{WTR}$ ~ Max. $T_{WTR}$ |
|---|---|---|---|
| 000 | $2^4 * T_{WDT}$ | $1024 * T_{WDT}$ | 1.6 ms ~ 104 ms |
| 001 | $2^6 * T_{WDT}$ | $1024 * T_{WDT}$ | 6.4 ms ~ 108.8 ms |
| 010 | $2^8 * T_{WDT}$ | $1024 * T_{WDT}$ | 25.6 ms ~ 128 ms |
| 011 | $2^{10} * T_{WDT}$ | $1024 * T_{WDT}$ | 102.4 ms ~ 204.8 ms |
| 100 | $2^{12} * T_{WDT}$ | $1024 * T_{WDT}$ | 409.6 ms ~ 512 ms |
| 101 | $2^{14} * T_{WDT}$ | $1024 * T_{WDT}$ | 1.6384 s ~ 1.7408 s |
| 110 | $2^{16} * T_{WDT}$ | $1024 * T_{WDT}$ | 6.5536 s ~ 6.656 s |
| 111 | $2^{18} * T_{WDT}$ | $1024 * T_{WDT}$ | 26.2144 s ~ 26.3168 s |

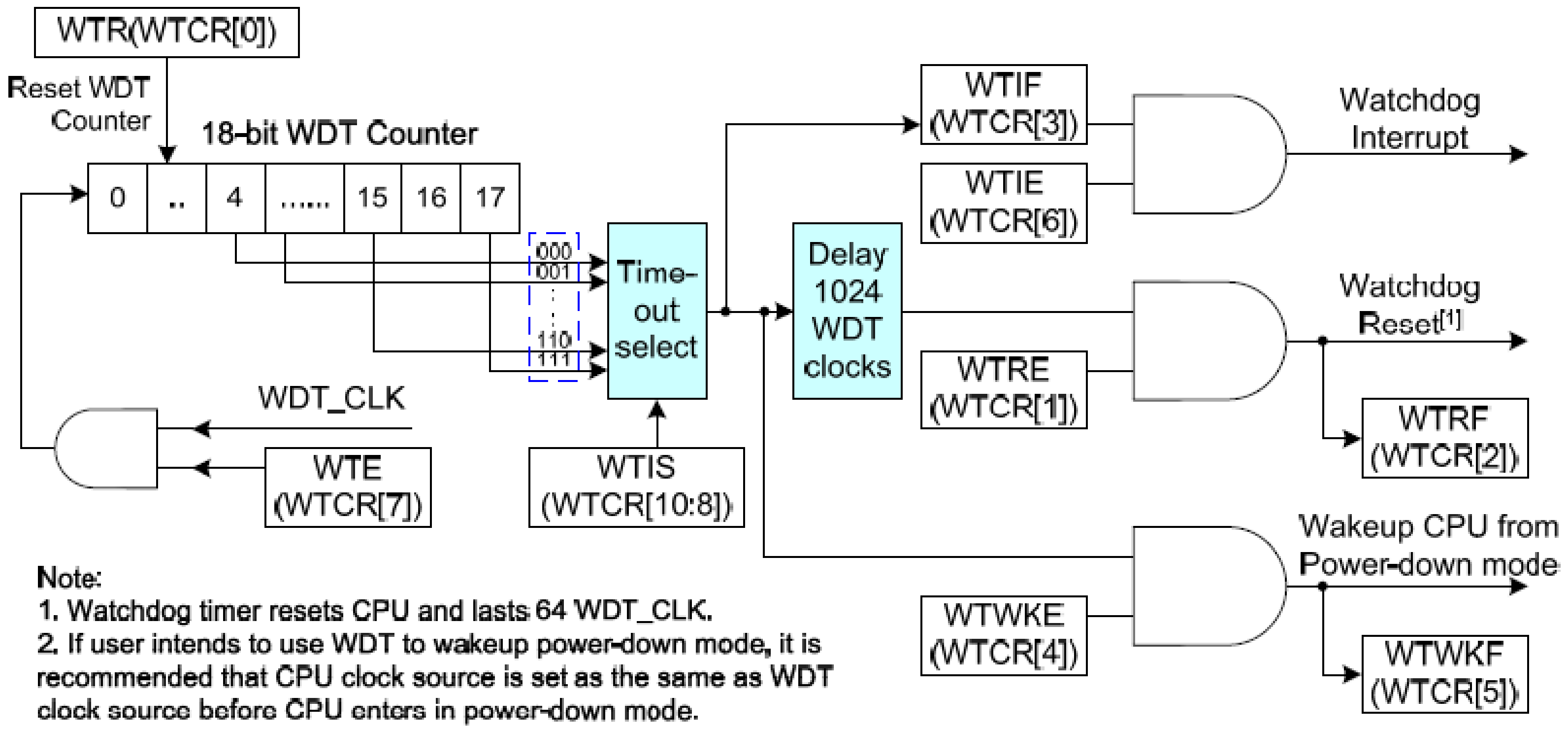
# WDT : Timing



- $T_{WDT}$ *: Watchdog Engine Clock Time Period*
- $T_{TIS}$ *: Watchdog Timeout Interval Selection Period*
- $T_{INT}$ *: Watchdog Interrupt Period*
- $T_{RST}$ *: Watchdog Reset Period*
- $T_{WTR}$ *: Watchdog Timeout Interval Period*



# WDT : Block Diagram



Note:
1. Watchdog timer resets CPU and lasts 64 WDT_CLK.
2. If user intends to use WDT to wakeup power-down mode, it is recommended that CPU clock source is set as the same as WDT clock source before CPU enters in power-down mode.

- User must set WTR (WDTCR [0]) (Watchdog timer reset) high to reset the 18-bit WDT counter to avoid chip from Watchdog timer reset before the delay time expires. WTR bit is cleared automatically by hardware after WDT counter is reset.
- WTRF will not be cleared by Watchdog reset. User may poll WTRF by software to recognize the reset source.

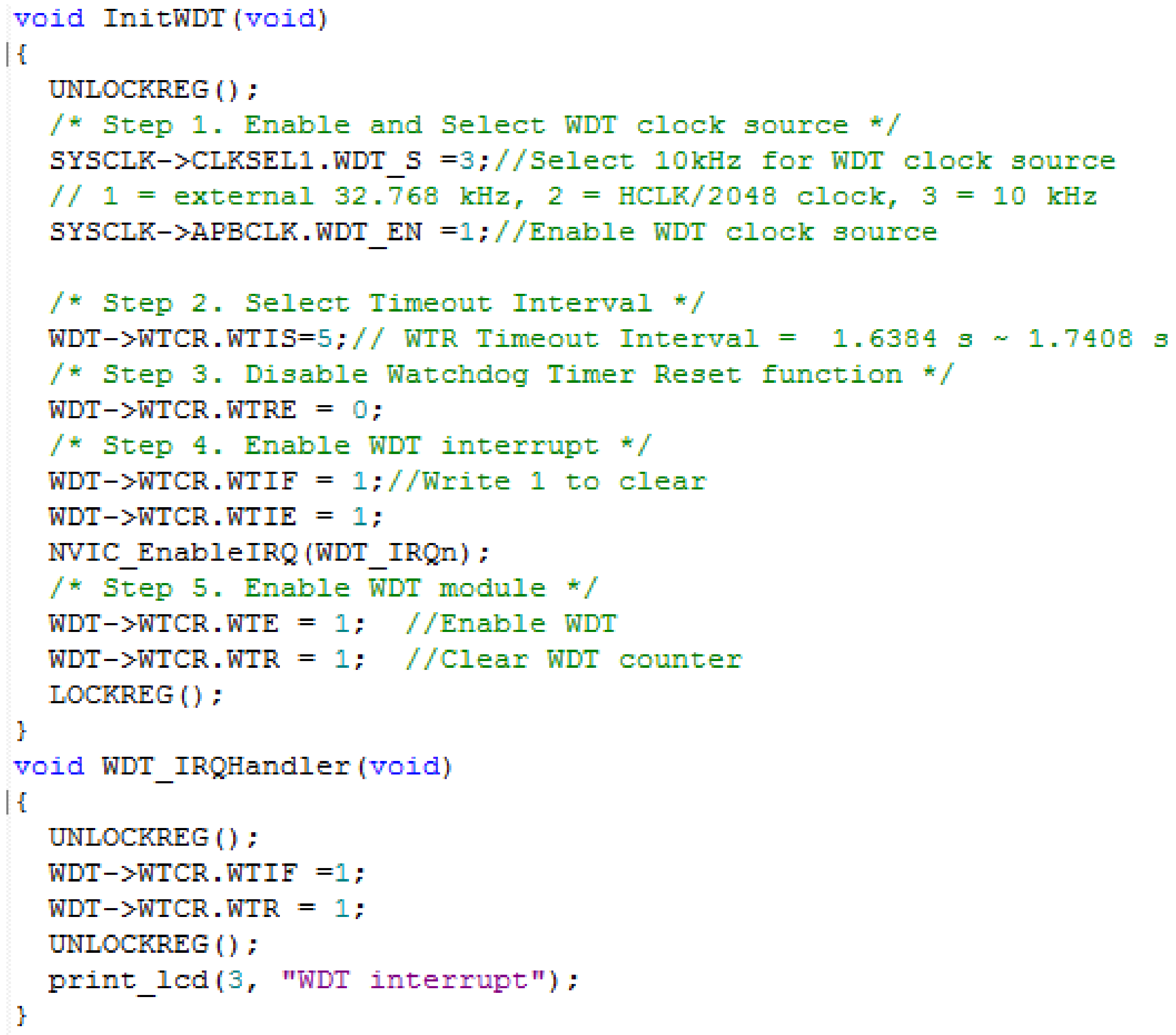
# WDT : WTCR

| Register | Offset | R/W | Description | Reset Value |
|---|---|---|---|---|
| **WTCR** | WDT_BA+0x00 | R/W | Watchdog Timer Control Register | 0x0000_0700 |

Note: All bits can be write in this register are write-protected. To program it needs to write “59h”, “16h”, “88h” to address 0x5000_0100 to disable register protection. Reference the register REGWRPROT at address GCR_BA+0x100.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **DBGACK_WDT** | **Reserved** | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | **WTIS** | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **WTE** | **WTIE** | **WTWKF** | **WTWKE** | **WTIF** | **WTRF** | **WTRE** | **WTR** |

# WDT : Program Example



```
void InitWDT(void)
{
  UNLOCKREG();
  /* Step 1. Enable and Select WDT clock source */
  SYSCLK->CLKSEL1.WDT_S =3;//Select 10kHz for WDT clock source
  // 1 = external 32.768 kHz, 2 = HCLK/2048 clock, 3 = 10 kHz
  SYSCLK->APBCLK.WDT_EN =1;//Enable WDT clock source

  /* Step 2. Select Timeout Interval */
  WDT->WTCR.WTIS=5;// WTR Timeout Interval =  1.6384 s ~ 1.7408 s
  /* Step 3. Disable Watchdog Timer Reset function */
  WDT->WTCR.WTRE = 0;
  /* Step 4. Enable WDT interrupt */
  WDT->WTCR.WTIF = 1;//Write 1 to clear
  WDT->WTCR.WTIE = 1;
  NVIC_EnableIRQ(WDT_IRQn);
  /* Step 5. Enable WDT module */
  WDT->WTCR.WTE = 1;  //Enable WDT
  WDT->WTCR.WTR = 1;  //Clear WDT counter
  LOCKREG();
}
void WDT_IRQHandler(void)
{
  UNLOCKREG();
  WDT->WTCR.WTIF =1;
  WDT->WTCR.WTR = 1;
  UNLOCKREG();
  print_lcd(3, "WDT interrupt");
}
```