FAQ0051
Frequently Asked Questions

How to improve the efficiency of reading Flash in NZW area?

Questions:

As some AT32 MCU are equipped with non-zero-wait areas (NZW), when code runs beyond the zero-wait area (ZW) and the clock is lower than 72 MHz, the efficiency of reading Flash memory will be reduced. In this case, how to quickly modify the program to improve the speed rate of reading Flash without changing peripheral clocks?

Answer:

With the user's peripheral clocks unchanged, follow the instructions below to make slight modification on the program:

Increase the max frequency to an integer multiple of the original one, adjust the APB1PSC[2: 0] and APB2PSC[2: 0] to ensure that APB1 and APB2 bus clocks maintain their respective original value; if necessary, the prescaler DIV register of the TMR is also adjusted.

Example:

1. If the user's original MCU max frequency is working at 72 MHz, APB1=36MHz,APB2=72MHz:
2. Increase PLL max frequency to 72X2=144MHz;

```
#define   RCC_CFGR_PLLRANGE_GT72MHZ   ((uint32_t)0x80000000)
/*   PLL configuration: PLLCLK = HSE * 18 = 144 MHz */
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL18 |
RCC_CFGR_PLLRANGE_GT72MHZ);
```

If the user is using Artery's official BSP, the reference code is as follows:

```
crm_pll_config(CRM_PLL_SOURCE_HEXT_DIV, CRM_PLL_MULT_60,
 CRM_PLL_OUTPUT_RANGE_GT72MHZ);
```

3. Modify APB1 and APB2 clock division to maintain their respective clock frequency before change;
   If the user is using BSP which is not provided by Artery, refer to the following code:

```
RCC_APB1CLKConfig(RCC_AHBCLK_Div4);   //Keep APB1 at 36MHz
RCC_APB2CLKConfig(RCC_AHBCLK_Div2);   //Keep APB2 at 72MHz
```

   If the user is using Artery's official BSP, refer to the following code:

```
crm_apb1_div_set(CRM_APB1_DIV_4 );   // Keep APB1 at 36MHz
crm_apb2_div_set(CRM_APB2_DIV_2 );   // Keep APB2 at 72MHz
```

4. When APB division factor is equal to 1, the timer clock frequency is equal to APB clock frequency;
   When APB division factory is not equal to 1, the timer clock frequency is equal to APB clock frequency X 2;

	Before change (PLL=72M)	After change (PLL=144M)
TMR2 on APB1	36*2=72MHz	36*2=72MHz
TMR1 on APB2	72*1=72MHz	72*2=144MHz

   Before change, APB1=36 MHz, the prescaler is 2, then the timer clock frequency on APB1 is 72 MHz; After change, the prescaler is 4, the timer clock frequency is still 72 MHz, and so the timer clock division on APB1 does not need change.

   Before change, APB2=72 MHz, the prescaler is 1, the timer clock frequency on APB2 is 72 MHz; after change, the prescaler is 2, the timer clock frequency is changed to 144 MHz, so the timer clock division on APB2 is equal to the original value x 2.

This calls for the need to modify the DIV(SXX 为 PSC) register of the TMR1. If the user is using other BSP (not Artery's), refer to the following code:

```
TIM_Cmd(TIM1, DISABLE);
TIM_TimeBaseStructure.TIM_Prescaler =psc; // modify frequency division factor
TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);
TIM_Cmd(TIM1, ENABLE);
```

If the user is using Artery's official BSP, refer to the following code:

```
tmr_base_init(TMR1, tmr_pr-1, tmr_div); // modify frequency division factor to tmr_pr-1
```

5. Based on above, the speed rate of code is increased from 72*0.4=28.8Mbyte/s to 144*0.4=57.6Mbyte/s.

Type: MCU

Applicable products: AT32F403, AT32F403A, AT32F407, AT32F413, AT32F435, AT32F437

Main function: Flash

Minor function: Timer

Document revision history

Date	Revision	Changes
2022.3.5	2.0.0	Initial release