Nano-edge placement of eFlexP)WM on KV5x family

1. The feature of KV5x family.

The assumed application field of Kv5x family is motor control and switch mode power supply, most of the peripherals of KV5x family are the same as that of DSC, but the core changes to Cortex-M7, it is sub-family of Kinetis.

In switch mode power supply application, high speed ADC converter, advanced PWM module which can generate complicated PWM signal with high resolution, timers, comparator are prerequisite. The KV5x family has eFlexPWM, FTM, high speed cyclic ADC converter, SAR ADC converter, crossbar module, on-chip comparator, all the peripherals module make it suitable in switch mode power supply application and advanced level motor control.

it is desired to improve controlling voltage resolution, but the higher the PWM signal frequency, the low the voltage resolution is, for example, the usual PWM frequency is 100KHz, if we set the PWM driving clock frequency is 100MHz, the Voltage resolution is about less than 10 bits.

The eflexPWM module can generate complicated PWM waveform, one eFlexPWM module has 4 sub-modules, from theory, it can generate 3*4=12 independent PWM signals, both the rising and falling edge of each PWM signal can be controlled by firmware.

In order to improve the PWM resolution, the eFlexPWM supports the nano-edge placement and duty cycle dithering features, but not each eFlexPWM module supports nano-edge placement, the eFlexPWM module with internal PLL can support the nano-edge placement feature, but all eFlexPWM module supports dithering duty cycle mode.

2. nano-edge placement feature

The Nano-edge placement:

The nano-edge placement means that the PWM duty cycle can extend a fractional tick cycle time in order to increase PWM resolution, in other words, the PWM module can generate integer plus fractional duty cycle. For example, the PWM driving clock for KV58 is 100MHz, the 100MHz clock is the main tick, the main tick cycle time is 10ns. There is an internal PLL inside the eFlexPWM module, the PLL can multiply the 100MHz main clock with 32 and can output 3.2GHz clock, the 3.2KHz clock is the fractional clock. User can set the duty cycle with a number of main clock plus a number of fractional clock.

Assume that the main clock is 100MHz, the fractional clock is 3.2GHz. If you set the PWM frequency in 1MHz, the duty cycle is 50%, the High logic covers 50 main tick cycles, the low logic convers 50 main tick cycles. For nano-edge placement, the High and low logic can cover a number of both main tick cycles and a number of fractional tick cycles, in other words, the duty cycle can be x+y/32, x demotes the main tick cycle, the y denote the number of fractional cycle, it means that the duty cycle can be 50.03125, 50.0625, 50.09325,50.125,…….50.96875 when y is equals to 1, 2 3,4…31.

3)Enable PLL inside the eFlexPWM module

A: set up the MCG in PEE mode so that the main clock of eFlexPWM is 100MHz.

```
Core clock      = 200MHz
Fast bus clock  = 100MHz
Bus/Flash clock = 25MHz
```

B:initialize the eFlexPWM so that PLL is enabled with the following code.

```
 //set PLL begin
PMC_REGSC|=0x01;//setting the BGBE bit
PWM0_SM0FRCTRL=0x114;
PWM0_SM1FRCTRL=0x114;
SIM_PWRC=0x100;
while(!(SIM_PWRC&0x10000)) {}
SIM_SOPT2&=0x30000;
SIM_SOPT2|=0x10000;
PWM0_MCTRL1=0x03;
while(PWM0_MCTRL1!=0x03) {}
//PLL is okay now
PWM0_SM0FRACVAL5=15<<11;
//set PLL end
//PWM0 global register setting
PWM0_OUTEN|=0x0FF0;  //enable PWM output
PWM0_MASK=0x0000;  //disable PWM mask
PWM0_SWCOUT=0x0000;  //determine dead time logic
//     PWM0_FCTRL)=0xF000; //fault logic setting
PWM0_MCTRL0|=0x0007; //must use the instruction, otherwise, the counter
will disorder, IPOL is cleared, PWM23 manipulate the duty cycle
```