```
// FEPR register settings
          output_mux           = <1>;
          LV_power_on_polarity     = <1>;
          LV_power_on_output_enable = <1>;
          over_temp_polarity      = <0>;
          over_temp_input_enable    = <0>;
          fuser_polarity         = <0>;
          fuser_output_enable      = <3>;
```

output_mux	bit 21:20 OUTPUT_MUX
LV_power_on_polarity	bit 18 LVPOWERON_POLARITY
LV_power_on_output_enable	bit 17 LVPOWERON_OE
over_temp_polarity	bit 12:14 FUSEROVERTEMP_POLARITY
over_temp_input_enable	bit 8 FUSEROVERTEMP_IE
fuser_polarity	bit 4:6 FUSER_POLARITY
fuser_output_enable	bit 0:2 FUSER_OE

---

dec fuser module **ouput** 3 fuser interfaces,用于控制外部的fusers。

FuserOn_OE[0]   Output Enable

FuserOn[0]

FuserOn_OE[1] Output Enable

FuserOn[1]

FuserOn_OE[2] Output Enable

FuserOn[2]

FuserOn[2:0]输出的应该是某种方式modulation的PWM wave。G2 Fuser block支持delta-sigma mode, PWM mode, and phase-angle mode.

dts中的

```
mode = <ASYNC_DSM ASYNC_DSM>;
```

用于指定modulation mode。

下面**output**是控制线电压的

LV --- Line Voltage,线电压，也就是高压

LVPowerOn_OE Line Voltage Output Enable

LVPowerOn

针对3个fuser control module，有3个对应的over temperature **input**

FuserOverTemp[0]

FuserOverTemp[1]

FuserOverTemp[2]

也就是下图中的FuserOverTemp[2:0]

When FuserOverTemp[2:0] input is asserted, the over-temperature interrupt is generated.

也就是提供了一个外部能感知over temperature并通知CPU的能力。是否over temperature是由外部电路实现的，对CPU而言只是从FuserOverTemp[2:0]接受一个asserted信号而已（高/低电平?）

对于同步modulation有ZeroCrossDetect **input**

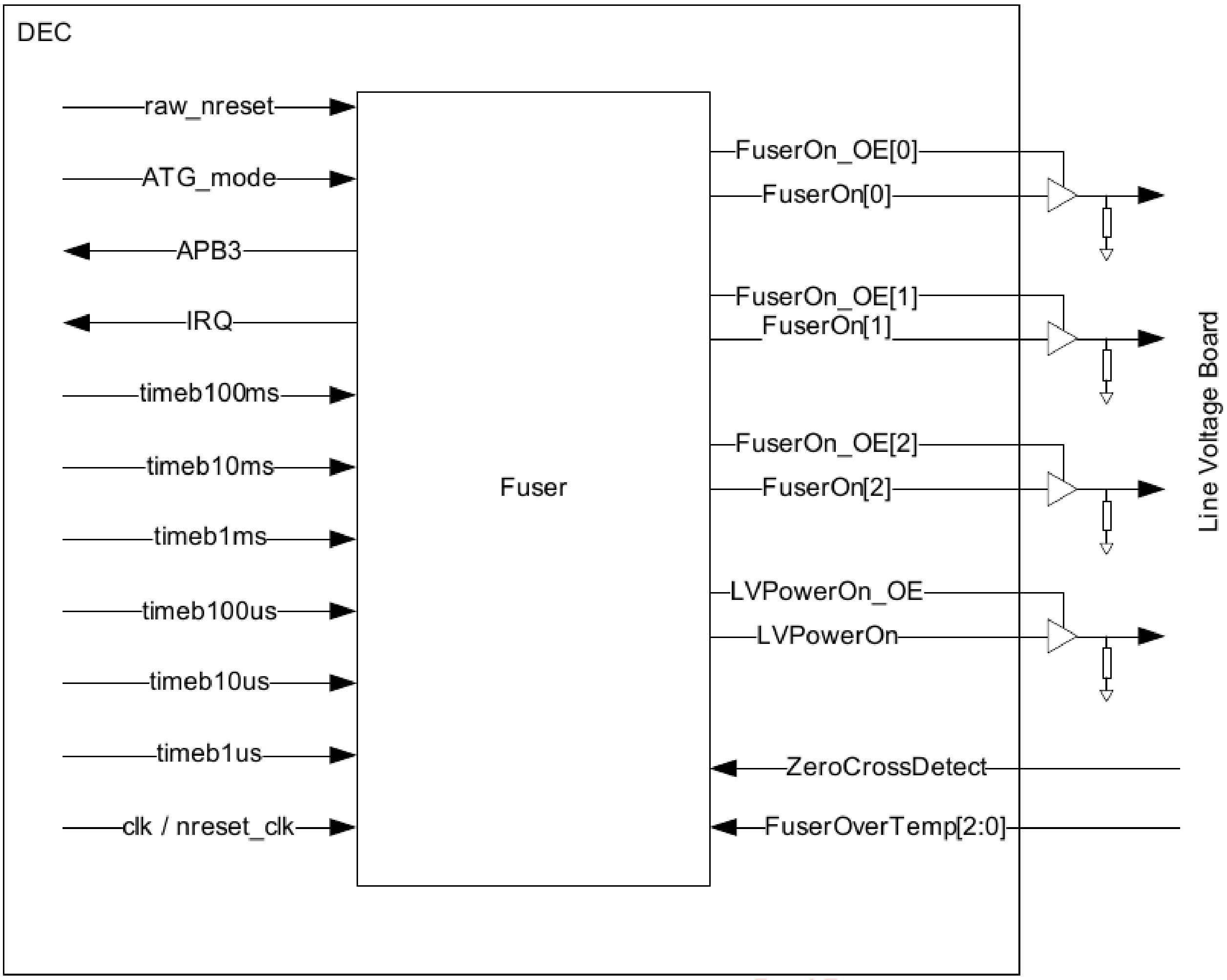
外部电路应该提供过零点检测，对SoC而言只是接受该外部电路给出的过零点signal,以便SoC可以产生ZeroCross interrupt.

目前使用的是异步modulation，所以未用。

"同步"/"异步"是指对外部的Line Volatge而言的。

比如中国的市电是50HZ,也就是一秒内有100个过零点，即100个half cycle。而PWM的周期是与过零点同步的，而不是象"异步"一样是与市电频率无关的。

而"异步"PWM的period则由timebase和timestep指定，与line voltage无关。



dts中

```
fuser_output_enable         = <3>;
```

即FEPR.FUSER_OE[2:0] = 0x3 = 011

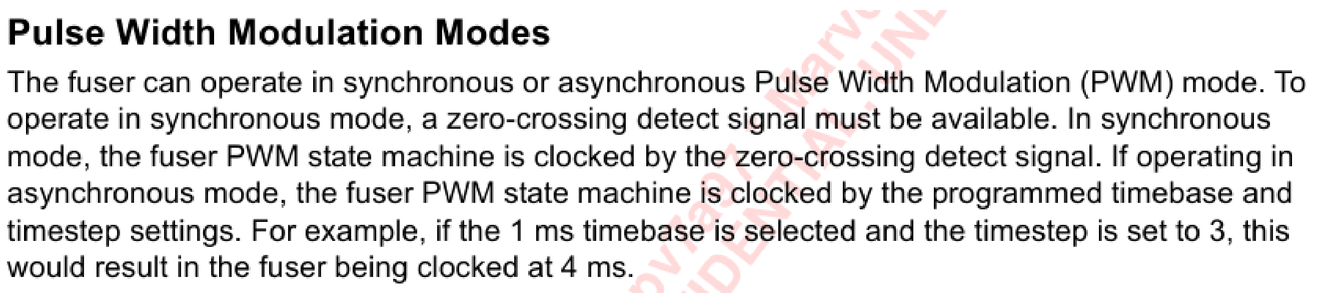
即enable了FuserOn[0]和FuserOn[1],而FuserOn[2]没有用到。

offset	IO_PAD	function (0x1)		
0xCC	IO_PAD_51	Fuser_Ctrl[0]	nFUSER_CTRL_ON	
0xD0	IO_PAD_52	Fuser_Ctrl[1]	RELAY_AC_IN	
0xD4	IO_PAD_53	Fuser_Ctrl[2]	24V_ON	
0xD8	IO_PAD_54	Fuser_Ctrl[3]	N/A (没用到)	
0xF4	IO_PAD_61	FuserOverTemp	nOVER_TEMP	

FuserOn[0] (IO_PAD_51)连接nFUSER_CTRL_ON, "element" fuser

FuserOn[1] (IO_PAD_52)连接RELAY_AC_IN, "relay" fuser

PWM的clock，在Programmer Guide中有如下描述



Pulse Width Modulation Modes

The fuser can operate in synchronous or asynchronous Pulse Width Modulation (PWM) mode. To operate in synchronous mode, a zero-crossing detect signal must be available. In synchronous mode, the fuser PWM state machine is clocked by the zero-crossing detect signal. If operating in asynchronous mode, the fuser PWM state machine is clocked by the programmed timebase and timestep settings. For example, if the 1 ms timebase is selected and the timestep is set to 3, this would result in the fuser being clocked at 4 ms.

在同步mode下，PWM的周期是由Line Voltage来驱动的，通过过零点的signal(zero-crossing detect)

而异步mode下，则完全有设置的timebase和timestep参数来决定。

Programmer Guide中timestep的description如下

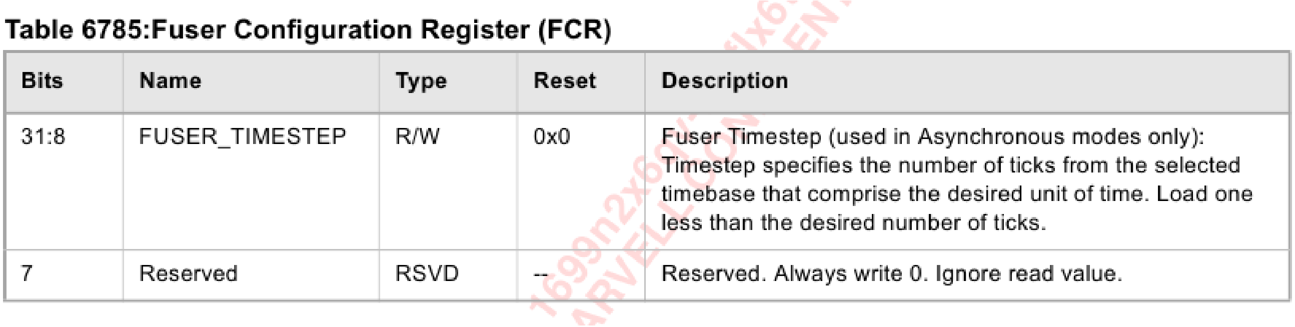


Table 6785:Fuser Configuration Register (FCR)

Bits	Name	Type	Reset	Description
31:8	FUSER_TIMESTEP	R/W	0x0	Fuser Timestep (used in Asynchronous modes only): Timestep specifies the number of ticks from the selected timebase that comprise the desired unit of time. Load one less than the desired number of ticks.
7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

即如果timebase选了1ms，timestep = 5,那么整个PWM的周期应该为5 * 1ms = 5ms