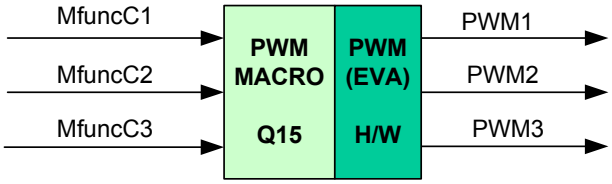
PWM_DRV	*Full Compare PWM Driver*

Description

This module uses the duty ratio information and calculates the compare values for generating PWM outputs. The compare values are used in the full compare EPWM unit in 2833x. This also allows PWM period modulation.



Availability

This 16-bit module is available in one interface format:

1) The C interface version

Module Properties

Type: Target Dependent, Application Independent

Target Devices: 2833x Floating Point device

C Version File Names: f2833xpwm.h (for x2833x)

IQmath library files for C: N/A

C Interface

Object Definition

The structure of PWMGEN object is defined by following structure definition

```
typedef struct {  Uint16 PeriodMax;     // Parameter: PWM Half-Period in CPU clock cycles (Q0)
                  int16 MfuncPeriod;    // Input: Period scaler (Q15)
                  int16 MfuncC1;        // Input: PWM 1 Duty cycle ratio (Q15)
                  int16 MfuncC2;        // Input: PWM 2 Duty cycle ratio (Q15)
                  int16 MfuncC3;        // Input: PWM 3 Duty cycle ratio (Q15)
               } PWMGEN;

typedef PWMGEN *PWMGEN_handle;
```

Item	Name	Description	Format	Range(Hex)
Inputs	MfuncCx (x=1,2,3)	PWM duty cycle ratio	Q15	8000-7FFF
	MfuncPeriod	Period scaler	Q15	8000-7FFF
Outputs	PWMx (x=1,2,3,4,5,6)	Output signals from the 6 PWM pins	N/A	0-3.3 V
PWMGEN parameter	PeriodMax	PWM Half-Period in CPU clock cycles	Q0	8000-7FFF

Special Constants and Data types

PWMGEN
The module definition is created as a data type. This makes it convenient to instance an interface to the PWMGEN driver. To create multiple instances of the module simply declare variables of type PWMGEN.

PWMGEN_handle
User defined Data type of pointer to PWMGEN module

PWMGEN_DEFAULTS
Structure symbolic constant to initialize PWMGEN module. This provides the initial values to the terminal variables as well as method pointers.

Methods

PWM_INIT_MACRO (v);
PWM_MACRO (v);

This default definition of the object implements two methods – the initialization and the runtime compute macro for PWMGEN generation. This is implemented by means of a pointer, and the initializer sets this to PWM_INIT_MACRO and PWM_MACRO macros for x2833x. The argument to this macro is the address of the PWMGEN object.

Module Usage

Instantiation
The following example instances one PWMGEN object
PWMGEN pwm1;

Initialization
To Instance pre-initialized object
PWMGEN pwm1 = PWMGEN_DEFAULTS;

Invoking the computation macro
PWM_INIT_MACRO (pwm1);
PWM_MACRO (pwm1);

Example

The following pseudo code provides the information about the module usage.

```
main()
{

        pwm1.PeriodMax = 7500;          // PWM frequency = 10 kHz, clock = 150 MHz
        PWM_INIT_MACRO(pwm1);     // Call init macro for pwm1

}

void interrupt periodic_interrupt_isr()
{

        pwm1.MfuncC1 = (int)_IQtoIQ15(svgen_dq1.Ta);    // svgen_dq1.Ta is in GLOBAL_Q
        pwm1.MfuncC2 = (int)_IQtoIQ15(svgen_dq1.Tb);    // svgen_dq1.Tb is in GLOBAL_Q
        pwm1.MfuncC3 = (int)_IQtoIQ15(svgen_dq1.Tc);    // svgen_dq1.Tc is in GLOBAL_Q
        PWM_MACRO (pwm1);                               // Call update macro for pwm1


}
```