Module Document: PFC_ICMD

Texas Instruments – C2000 DSP System Applications Group

Contents

Figures

Tables

1 PFC_ICMD module documentation

This software module performs a computation of the current command for the power factor correction, as per the flow shown in **Figure 1**. The inputs are generally connected to the inverse-squared/averaged line voltage, the rectified line voltage and the output of the voltage controller. The PFC_ICMD block then generates an output command profile that is half-sinusoidal, with an amplitude dependent on the output of the voltage controller. The output is then connected to the current controller to produce the required inductor current.

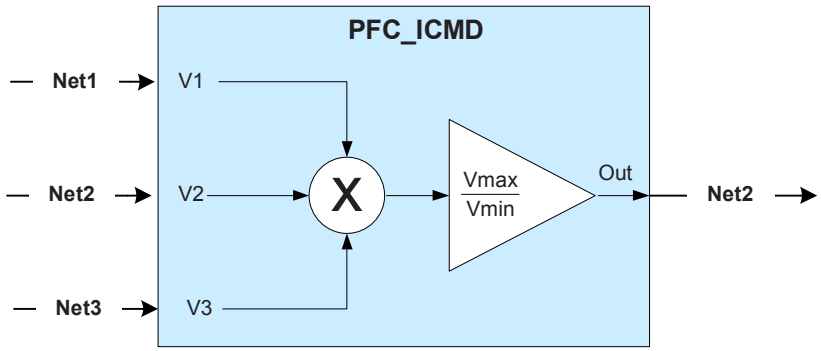

Figure 1. Input and output of the PFC_ICMD block, and internal data flow

1.1 Module Properties

This section describes module properties, such as compatible devices, components, invocation etc. The PFC_ICMD module has the following dependencies:

Module	Dependency
CPU dependency	C28x
Device dependency	Any 28x device

Table 1. PFC_ICMD module dependencies

The PFC_ICMD module has the following components:

Component	Present
C-based initialization	No
ASM interrupt initialization	Yes
ASM runtime macro	Yes

Table 2 PFC_ICMD module components

The INV_SQR module has the following miscellaneous properties:

Property name	Property value
Multiple instance support	Yes
Reentrant	Yes
Accessible from 'C' environment	Yes
Full configuration from 'C' environment	Yes
Input / Output connection	Pointer to signal net.

Table 3. PFC_ICMD module miscellaneous properties

Component	Files
Macro source:	C:\tidcs\DPS_C280x\vXYZ[1]\dplib280x\dplib280x.inc
C Interface:	C:\tidcs\DPS_C280x\vXYZ\dplib280x\dplib280x.h
C source	C:\tidcs\DPS_C280x\vXYZ\dplib280x\dplib280x.src
Object file archive	C:\tidcs\DPS_C280x\vXYZ\dplib280x\dplib280x.lib

Table 4. PFC_ICMD module component files

1.2 Module Input Definition

Input name	Description	Format	Range
PFC_ICMD_V1X	Input	Pointer to 16-bit fixed point input data	Q15: [0, 1) or [-32768, 32767]
PFC_ICMD_V2X	Input	Pointer to 16-bit fixed point input data	Q15: [0, 1) or [-32768, 32767]
PFC_ICMD_VacX	Input	Pointer to 16-bit fixed point input data	Q15: [0, 1) or [-32768, 32767]

(X is the instance number)

Table 5. PFC_ICMD terminal inputs

[1] The xyz represents the version number directory level. For instance, a 1.00 release would have v100 in its directory path, and v210 would indicate a release 2.10.

1.3 Module Output Definition

Output name	Description	Format	Range
PFC_ICMD_OutX	Output	Pointer to 16-bit fixed point input data	Q15: [0, 1) or [0, 32767]

(X is the instance number)

Table 6. PFC_ICMD terminal outputs

1.4 Module API Description: PFC_ICMD_INIT

Function Name: PFC_ICMD_INIT

Prototype: PFC_ICMD_INIT nInstance

Return value: None.

Preconditions: None

This function is the assembler initialization macro, and must be called prior to running the PFC_ICMD runtime macro, for proper operation of the runtime macro routine. This initialization routine must be executed as part of an assembler initialization routine. This macro routine declares variables, initializes variables to known values, and sets up constants for the runtime macro routines.

1. nInstance**:** Specifies which instance is initialized.

Example: Call the PFC_ICMD_INIT to initialize PFC_ICMD module.

```
;-------------------------------------------------------
; ISR Initialization
;-------------------------------------------------------
_ISR_Init:  ...
                    ...
                    PFC_ICMD_INIT           1

                    LRETR
```

1.5 Module API Description: PFC_ICMD

Function Name: PFC_ICMD

Prototype: PFC_ICMD nInstance

Return value: None.

Preconditions: The following precondition must be satisfied:

- The ISR initialization macro PFC_ICMD_INIT must be instanced in an assembler initialization routine, and the assembler init routine must run prior to this routine.

This function is the assembler run time macro, this routine performs the scaled inverse square computation.

nInstance: Specifies which instance is run.

Example: Call the PFC_ICMD in an assembler ISR

```
;-------------------------------------------------------
; Runtime interrupt service routine
;-------------------------------------------------------
_ISR_Run:           CONTEXT_SAVE                ;call macro

;-------------------------------------------------------
; Run the PFC_ICMD instance 1
;-------------------------------------------------------
                    PFC_ICMD    1
;-------------------------------------------------------
EXIT_ISR: ;Interrupt management before exit
;-------------------------------------------------------
                    MOVW        DP,#ETCLR1>>6
                    MOV         @ETCLR1,#0x01; Clear EPWM1 Int flag

;-------------------------------------------------------
; Restore context & return
;-------------------------------------------------------
                    CONTEXT_REST
                    IRET
```

1.6 Usage Example:

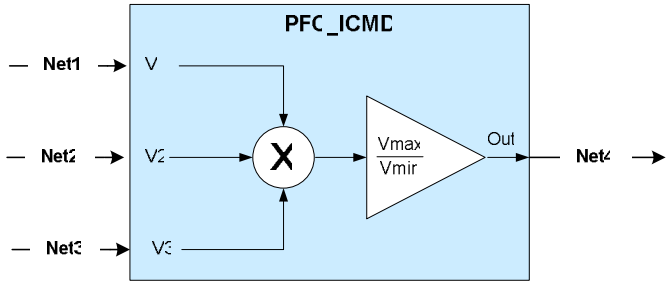

Figure 2. Connecting the PFC_ICMD module

Step1. Instantiate the INIT macro in assembly (this is one-time pass through code)

```
; Instantiate the init macro

       PFC_ICMD_INIT1
```

Step2. Instantiate the run time macro in assembly (this is usually looped or ISR code)

```
; “call” the runtime macro

       PFC_ICMD                1
```

Step3. Declare signal nets to which the module will be connected

```
int16   Net1, Net2, Net3, Net4;
```

Step4. Declare the terminal pointers in C

```
// PFC_ICMD terminal pointers, external references
```

```
extern int16   *PFC_ICMD_V12, *PFC_ICMD_V21, *PFC_ICMD_Vac1, *PFC_ICMD_Out1;
```

Step5. “Connect” the module terminals to the Signal Nets in “C”.

```
// PFC_ICMD connections
       PFC_ICMD_V11   = &Net1;
       PFC_ICMD_V21   = &Net2;
       PFC_ICMD_Vac1  = &Net3;
       PFC_ICMD_Out1  = &Net4;
```