Module Document: INV_SQR

Texas Instruments – C2000 DSP System Applications Group

Contents

Figures

Tables

1 INV_SQR module documentation

This software module performs a reciprocal function on a unipolar input signal. An input signal which is represented in Q15, is scaled and inverted. The scaling allows for the fact that what is input is an average of a half-sine, whereas what is desired for the PFC_ICMD block is a representation of the peak of the sine. In addition the input signal is clamped to a minimum to allow the PFC system to work with very low line voltages without overflows, which can cause undesired effects.

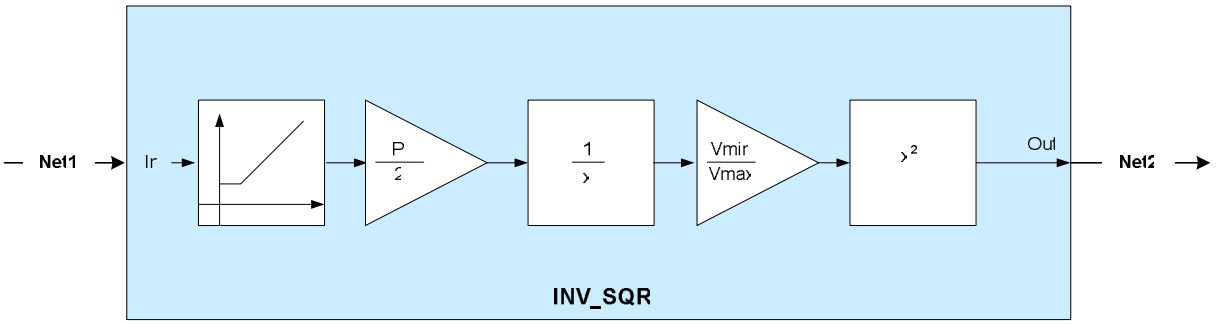

Figure 1. Input and output for the INV_SQR module

1.1 Module Properties

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

Module	Dependency
CPU dependency	C28x
Device dependency	Any 28x device

Table 1. INV_SQR module dependencies

The INV_SQR module has the following components:

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

Table 2 INV_SQR 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. INV_SQR 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. INV_SQR module component files

1.2 Module Input Definition

Input name	Description	Format	Range
INV_SQR_InX (X is the instance number)	Input	Pointer to 16-bit fixed point input data	Q15: [0, 1) or [-32768, 32767]

Table 5. INV_SQR terminal inputs

1.3 Module Output Definition

Output name	Description	Format	Range
INV_SQR_OutX (X is the instance number)	Output	Pointer to 16-bit fixed point input data	Q15: [0, 1) or [0, 32767]

Table 6. INV_SQR terminal outputs

[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.

1.4 Module API Description: INV_SQR_INIT

Function Name: INV_SQR_INIT

Prototype: INV_SQR_INIT nInstance

Return value: None.

Preconditions: None

This function is the assembler initialization macro, and must be called prior to running the INV_SQR 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 INV_SQR_INIT to initialize INV_SQR module.

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

                    LRETR
```

1.5 Module API Description: INV_SQR

Function Name: INV_SQR

Prototype: INV_SQR nInstance

Return value: None.

Preconditions: The following preconditions must be satisfied:

- The ISR initialization macro INV_SQR_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.

2. nInstance**:** Specifies which instance is run.

Example: Call the INV_SQR in an assembler ISR

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

;-------------------------------------------------------
; Run the INV_SQR instance 1
;-------------------------------------------------------
                    INV_SQR      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:

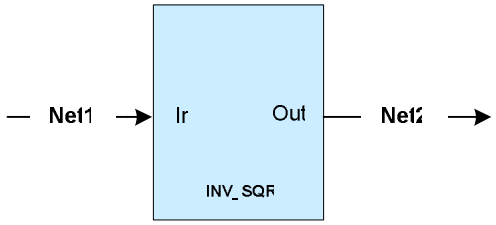

Figure 2. Connecting the INV_SQR module

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

```
; Instantiate the init macro

       INV_SQR_INIT 1
```

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

```
; “call” the runtime macro

       INV_SQR                  1
```

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

```
int16  Net1, Net2;
```

Step4. Declare the terminal pointers in C

```
// INV_SQR terminal pointers, external references
```

```
extern int16          * INV_SQR_In1, * INV_SQR_Out1;
```

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

```
// INV_SQR connections
// Note this can be done once during init, or dynamically during
// run time operation, i.e. module connections can be
// re-configured to other Nets as required by the application.
       INV_SQR_In1  = &Net1;
       INV_SQR_Out1 = &Net2;
```