;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;; FILENAME: PGA\_2.asm ( PGA )

;; Version: 3.2, Updated on 2015/3/4 at 22:26:46

;; Generated by PSoC Designer 5.4.3191

;;

;; DESCRIPTION: PGA User Module software implementation file.

;;

;; NOTE: User Module APIs conform to the fastcall16 convention for marshalling

;; arguments and observe the associated "Registers are volatile" policy.

;; This means it is the caller's responsibility to preserve any values

;; in the X and A registers that are still needed after the API functions

;; returns. For Large Memory Model devices it is also the caller's

;; responsibility to perserve any value in the CUR\_PP, IDX\_PP, MVR\_PP and

;; MVW\_PP registers. Even though some of these registers may not be modified

;; now, there is no guarantee that will remain the case in future releases.

;;-----------------------------------------------------------------------------

;; Copyright (c) Cypress Semiconductor 2015. All Rights Reserved.

;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;; -----------------------------------------------------------------

;; Register Definitions

;;

;; Uses 1 Continuous Time Block configured as shown.

;;

;; \* For a Mask/Val pair, this indicates that the value is

;; determined by the user either through config-time parameteriza-

;; tion or run-time manipulation.

;;

;; BIT FIELD Mask/Val Function

;; ----------------- ----- --------------------

;; GAIN\_CR0.RES\_RATIO\_T2B F0/\* User Parameter (by table)

;; GAIN\_CR0.GAIN\_ATTEN 08/\* Gain (by table)

;; GAIN\_CR0.RES\_SOURCE 04/1 Res source to output

;; GAIN\_CR0.RES\_REF 03/\* Res ref

;;

;; GAIN\_CR1.A\_OUT 80/\* User Parameter (Output bus)

;; GAIN\_CR1.COMP\_EN 40/0 Comparator bus disabled

;; GAIN\_CR1.CT\_NEG\_INPUT\_MUX 38/4 Neg mux to analog f.b. tap

;; GAIN\_CR1.CT\_POS\_INPUT\_MUX 07/\* Pos mux, typically to col. input mux

;;

;; GAIN\_CR2.CP\_COMP 80/0 Latch transparent on PH1

;; GAIN\_CR2.CK\_COMP 40/0 Latch transparent

;; GAIN\_CR2.CC\_COMP 20/1 Mode OP-AMP (not comparator)

;; GAIN\_CR2.BYPASS\_OBUS 1C/0 Bypass OFF

;; GAIN\_CR2.PWR\_SELECT 03/\* Power OFF (0h) at start-up

;;

;; --------------------------------------------------------------------

include "PGA\_2.inc"

include "m8c.inc"

include "memory.inc"

;-----------------------------------------------

; Global Symbols

;-----------------------------------------------

export PGA\_2\_Start

export \_PGA\_2\_Start

export PGA\_2\_SetPower

export \_PGA\_2\_SetPower

export PGA\_2\_SetGain

export \_PGA\_2\_SetGain

export PGA\_2\_Stop

export \_PGA\_2\_Stop

IF (PGA\_2\_AGNDBUFAPI)

export PGA\_2\_EnableAGNDBuffer

export \_PGA\_2\_EnableAGNDBuffer

export PGA\_2\_DisableAGNDBuffer

export \_PGA\_2\_DisableAGNDBuffer

ENDIF

;-----------------------------------------------

; EQUATES

;-----------------------------------------------

POWERMASK: equ 03h

GAINREGMASK: equ f8h

GAINMASK: equ fCh

HIGHGAIN: equ 04h

HIGHGAINMASK: equ 18h

EXGAIN: equ 01h

AGNDBUFMASK: equ 20h

AREA UserModules (ROM, REL)

.SECTION

;-----------------------------------------------------------------------------

; FUNCTION NAME: PGA\_2\_Start

; FUNCTION NAME: PGA\_2\_SetPower

;

; DESCRIPTION:

; Applies power setting to the module's PSoC block.

;-----------------------------------------------------------------------------

;

; ARGUMENTS:

; A Contains the power settings 0=Off, 1=Low, 2=Med, and 3=High

;

; RETURNS: NA

;

; SIDE EFFECTS:

; The A and X registers may be modified by this or future implementations

; of this function. The same is true for all RAM page pointer registers in

; the Large Memory Model. When necessary, it is the calling function's

; responsibility to perserve their values across calls to fastcall16

; functions.

;

PGA\_2\_Start:

\_PGA\_2\_Start:

PGA\_2\_SetPower:

\_PGA\_2\_SetPower:

RAM\_PROLOGUE RAM\_USE\_CLASS\_2

and A, POWERMASK ; mask A to protect unchanged bits

mov X, SP ; define temp store location

;

push A ; put power value in temp store

mov A, reg[PGA\_2\_GAIN\_CR2] ; read power value

and A, ~POWERMASK ; clear power bits in A

or A, [X] ; combine power value with balance of reg.

mov reg[PGA\_2\_GAIN\_CR2], A ; move complete value back to register

pop A

RAM\_EPILOGUE RAM\_USE\_CLASS\_2

ret

.ENDSECTION

.SECTION

;-----------------------------------------------------------------------------

; FUNCTION NAME: PGA\_2\_SetGain

;

; DESCRIPTION:

; This function sets the Gain/Atten of the amplifier. Valid gain settings

; are defined in the .inc file.

;

;-----------------------------------------------------------------------------

;

; ARGUMENTS:

; A Contains gain settings.

;

; Gain values shown are for example. (See .inc file for gain equates)

;

; RETURNS: NA

;

; SIDE EFFECTS:

; The A and X registers may be modified by this or future implementations

; of this function. The same is true for all RAM page pointer registers in

; the Large Memory Model. When necessary, it is the calling function's

; responsibility to perserve their values across calls to fastcall16

; functions.

;

PGA\_2\_SetGain:

\_PGA\_2\_SetGain:

RAM\_PROLOGUE RAM\_USE\_CLASS\_2

and A, GAINMASK ; mask A to protect unchanged bits

mov X, SP ; define temp store location

;

push A ; put gain value in temp store

mov A, reg[PGA\_2\_GAIN\_CR0] ; read mux settings

and A, ~GAINREGMASK ; clear gain bits in A

tst [X],HIGHGAIN ; See if High Gain is set

jnz .SETHIGHGAIN

and reg[PGA\_2\_GAIN\_CR3],~EXGAIN ; Clear High Gain bit.

or A, [X] ; combine gain value with balance of reg.

mov reg[PGA\_2\_GAIN\_CR0], A ; move complete value back to register

pop A

RAM\_EPILOGUE RAM\_USE\_CLASS\_2

ret

.SETHIGHGAIN:

and [X],HIGHGAINMASK ; Make sure we have a valid high gain

or A, [X] ; combine gain value with balance of reg.

mov reg[PGA\_2\_GAIN\_CR0], A ; move complete value back to register

or reg[PGA\_2\_GAIN\_CR3], EXGAIN ; Set High Gain bit.

pop A

RAM\_EPILOGUE RAM\_USE\_CLASS\_2

ret

.ENDSECTION

.SECTION

;-----------------------------------------------------------------------------

; FUNCTION NAME: PGA\_2\_Stop

;

; DESCRIPTION:

; Turns off the power to the amplifier.

;

;-----------------------------------------------------------------------------

;

; ARGUMENTS: None

;

; RETURNS: NA

;

; SIDE EFFECTS:

; The A and X registers may be modified by this or future implementations

; of this function. The same is true for all RAM page pointer registers in

; the Large Memory Model. When necessary, it is the calling function's

; responsibility to perserve their values across calls to fastcall16

; functions.

;

PGA\_2\_Stop:

\_PGA\_2\_Stop:

RAM\_PROLOGUE RAM\_USE\_CLASS\_1

and REG[PGA\_2\_GAIN\_CR2], ~POWERMASK

RAM\_EPILOGUE RAM\_USE\_CLASS\_1

ret

.ENDSECTION

IF (PGA\_2\_AGNDBUFAPI)

.SECTION

;-----------------------------------------------------------------------------

; FUNCTION NAME: PGA\_2\_EnableAGNDBuffer

;

; DESCRIPTION:

; Turns on the AGND buffer power.

;

;-----------------------------------------------------------------------------

;

; ARGUMENTS: None

;

; RETURNS: NA

;

; SIDE EFFECTS:

; The A and X registers may be modified by this or future implementations

; of this function. The same is true for all RAM page pointer registers in

; the Large Memory Model. When necessary, it is the calling function's

; responsibility to perserve their values across calls to fastcall16

; functions.

;

PGA\_2\_EnableAGNDBuffer:

\_PGA\_2\_EnableAGNDBuffer:

RAM\_PROLOGUE RAM\_USE\_CLASS\_1

and reg[PGA\_2\_GAIN\_CR3], ~AGNDBUFMASK

RAM\_EPILOGUE RAM\_USE\_CLASS\_1

ret

.ENDSECTION

.SECTION

;-----------------------------------------------------------------------------

; FUNCTION NAME: PGA\_2\_DisableAGNDBuffer

;

; DESCRIPTION:

; Turns off the AGND buffer power.

;

;-----------------------------------------------------------------------------

;

; ARGUMENTS: None

;

; RETURNS: NA

;

; SIDE EFFECTS:

; The A and X registers may be modified by this or future implementations

; of this function. The same is true for all RAM page pointer registers in

; the Large Memory Model. When necessary, it is the calling function's

; responsibility to perserve their values across calls to fastcall16

; functions.

;

PGA\_2\_DisableAGNDBuffer:

\_PGA\_2\_DisableAGNDBuffer:

RAM\_PROLOGUE RAM\_USE\_CLASS\_1

or reg[PGA\_2\_GAIN\_CR3], AGNDBUFMASK

RAM\_EPILOGUE RAM\_USE\_CLASS\_1

ret

.ENDSECTION

ENDIF

; End of File PGA\_2.asm