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

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

;; DUALADC\_1.asm

;; Version: 2.2, Updated on 2010/12/27 at 15:27:25

;; Generated by PSoC Designer 5.4.3191

;;

;; DESCRIPTION: DualADC 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 2010. All Rights Reserved.

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

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

include "DUALADC\_1.inc"

include "m8c.inc"

include "memory.inc"

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

; Global Symbols

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

export DUALADC\_1\_Start

export \_DUALADC\_1\_Start

export DUALADC\_1\_SetPower

export \_DUALADC\_1\_SetPower

export DUALADC\_1\_Stop

export \_DUALADC\_1\_Stop

export DUALADC\_1\_GetSamples

export \_DUALADC\_1\_GetSamples

export DUALADC\_1\_StopAD

export \_DUALADC\_1\_StopAD

export DUALADC\_1\_fIsData

export \_DUALADC\_1\_fIsData

export DUALADC\_1\_fIsDataAvailable

export \_DUALADC\_1\_fIsDataAvailable

export DUALADC\_1\_iGetData1

export \_DUALADC\_1\_iGetData1

export DUALADC\_1\_iGetData2

export \_DUALADC\_1\_iGetData2

export DUALADC\_1\_ClearFlag

export \_DUALADC\_1\_ClearFlag

export DUALADC\_1\_iGetData1ClearFlag

export \_DUALADC\_1\_iGetData1ClearFlag

export DUALADC\_1\_iGetData2ClearFlag

export \_DUALADC\_1\_iGetData2ClearFlag

export DUALADC\_1\_SetResolution

export \_DUALADC\_1\_SetResolution

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

; EQUATES

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

; Calctime parameters

wCalcTime: equ DUALADC\_1\_bCALCTIME

; Constants

LowByte: equ 1

HighByte: equ 0

AREA UserModules (ROM, REL)

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_Start

; FUNCTION NAME: DUALADC\_1\_SetPower

;

; DESCRIPTION:

; Applies power setting to the module's analog PSoc block.

;

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

;

; ARGUMENTS:

; A Contains power level setting 0 to 3

;

; RETURNS: none

;

; 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.

;

; Currently only the page pointer registers listed below are modified:

; CUR\_PP

;

; THEORY of OPERATION or PROCEDURE:

;

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

DUALADC\_1\_Start:

\_DUALADC\_1\_Start:

DUALADC\_1\_SetPower:

\_DUALADC\_1\_SetPower:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_PROLOGUE RAM\_USE\_CLASS\_2

RAM\_SETPAGE\_CUR >DUALADC\_1\_bfStatus

and A,DUALADC\_1\_bfPOWERMASK ; Mask only the valid power setting bits

mov X, SP ; Get location of next location on stack

push A ; Save power value on temp location

; Set power for ADC1

mov A, reg[DUALADC\_1\_bfADC1cr3] ; Get current value of ADC1cr3

and A, ~DUALADC\_1\_bfPOWERMASK ; Mask off old power value

or A, [X] ; OR in new power value

or A, f0h ; Make sure other register is set correctly

mov reg[DUALADC\_1\_bfADC1cr3], A ; Reload CR with new power value

; Set power for ADC2

mov A, reg[DUALADC\_1\_bfADC2cr3] ; Get current value of ADC2cr3

and A, ~DUALADC\_1\_bfPOWERMASK ; Mask off old power value

or A, [X] ; OR in new power value

or A, f0h ; Make sure other register is set correctly

mov reg[DUALADC\_1\_bfADC2cr3], A ; Reload CR with new power value

tst reg[DUALADC\_1\_bfADC1cr2], DUALADC\_1\_fRES\_SET

jz .DoNotLoadRes

mov A,DUALADC\_1\_bNUMBITS - DUALADC\_1\_bMINRES ; get and set the resolution

mov [DUALADC\_1\_bfStatus], A ; place it in the status variable

.DoNotLoadRes:

pop A ; Restore the stack and power value

RAM\_EPILOGUE RAM\_USE\_CLASS\_2

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_Stop

;

; DESCRIPTION:

; Removes power from the module's analog PSoc blocks, but the digital

; blocks keep on running.

;

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

;

; 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.

;

DUALADC\_1\_Stop:

\_DUALADC\_1\_Stop:

RAM\_PROLOGUE RAM\_USE\_CLASS\_1

and reg[DUALADC\_1\_bfADC1cr3], ~DUALADC\_1\_bfPOWERMASK

and reg[DUALADC\_1\_bfADC2cr3], ~DUALADC\_1\_bfPOWERMASK

RAM\_EPILOGUE RAM\_USE\_CLASS\_1

ret

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_Get\_Samples

;

; DESCRIPTION:

; Starts the A/D convertor and will place data is memory. A flag

; is set whenever a new data value is available.

;

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

;

; ARGUMENTS:

; A Number of samples to be taken. A zero will cause the ADC to run

; continuously. A flag is set whenever a new data values are available.

;

; 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.

;

; Currently only the page pointer registers listed below are modified:

; CUR\_PP

;

DUALADC\_1\_GetSamples:

\_DUALADC\_1\_GetSamples:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_bSampC

mov [DUALADC\_1\_bSampC], A ; Store sample count

; Interrupts, Put A/D in reset

mov A,[DUALADC\_1\_bfStatus] ; get and set the resolution

and A,DUALADC\_1\_bRES\_MASK

add A,DUALADC\_1\_bMINRES

call DUALADC\_1\_SetResolution

DUALADC\_1\_LoadMSBCounter: ; The PWM has been setup by SetResolution, now set the upper

; counter which will be the same as the period.

; Reset MSB of counter to most negative value

mov A,reg[DUALADC\_1\_bPWM\_IntTime\_MSB] ; Get MSB of PWM and move it into RAM

mov [DUALADC\_1\_cCounter1U], A ; Use counter as temp location

mov A, 00h ; Load A with zero for the calculation

sub A, [DUALADC\_1\_cCounter1U] ; 0 - MSB\_PWM = MSB\_of\_most\_neg\_value

asr A ; Half the range (+ and -)

IF (DUALADC\_1\_DATA\_FORMAT)

mov [DUALADC\_1\_cCounter1U], A ; Place result back into MSB of counter1

mov [DUALADC\_1\_cCounter2U], A ; Place result back into MSB of counter2

ELSE

mov [DUALADC\_1\_cCounter1U], 00h ; Always start at zero for unsigned values for ADC1

mov [DUALADC\_1\_cCounter2U], 00h ; Always start at zero for unsigned values for ADC2

ENDIF

mov A, reg[DUALADC\_1\_bPWM\_IntTime\_LSB] ; Dummy Read - required do not remove

mov reg[DUALADC\_1\_bPeriod1], FFh ; Make sure counter1 starts at FF

mov reg[DUALADC\_1\_bPeriod2], FFh ; Make sure counter2 starts at FF

and reg[DUALADC\_1\_bfADC1cr3],~DUALADC\_1\_fFSW0 ; Take Integrator out of reset

and reg[DUALADC\_1\_bfADC2cr3],~DUALADC\_1\_fFSW0 ; Take Integrator out of reset

IF DUALADC\_1\_NoAZ

and reg[DUALADC\_1\_bfADC1cr2],~DUALADC\_1\_fAutoZero ; Take Integrator out of AutoZero

and reg[DUALADC\_1\_bfADC2cr2],~DUALADC\_1\_fAutoZero ; Take Integrator out of AutoZero

ENDIF

or reg[DUALADC\_1\_bCounter1\_CR0], (DUALADC\_1\_fDBLK\_ENABLE|DUALADC\_1\_fPULSE\_WIDE) ; Enable the Counter

or reg[DUALADC\_1\_bCounter2\_CR0], (DUALADC\_1\_fDBLK\_ENABLE|DUALADC\_1\_fPULSE\_WIDE) ; Enable the Counter

or reg[DUALADC\_1\_fPWM\_LSB\_CR0], DUALADC\_1\_fDBLK\_ENABLE ; Enable PWM

or reg[DUALADC\_1\_bfPWM16\_INT\_REG], DUALADC\_1\_bfPWM16\_Mask ; Enable PWM and Counter interrupts

or reg[DUALADC\_1\_bfCounter1\_INT\_REG], DUALADC\_1\_bfCounter1\_Mask

or reg[DUALADC\_1\_bfCounter2\_INT\_REG], DUALADC\_1\_bfCounter2\_Mask

ret

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_StopAD

;

; DESCRIPTION:

; Completely shuts down the A/D is an orderly manner. Both the

; Timer and Counter are disabled and their interrupts are deactivated.

;

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

;

; 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.

;

DUALADC\_1\_StopAD:

\_DUALADC\_1\_StopAD:

RAM\_PROLOGUE RAM\_USE\_CLASS\_1

and reg[DUALADC\_1\_fPWM\_LSB\_CR0], ~DUALADC\_1\_fDBLK\_ENABLE ; Disable the PWM

and reg[DUALADC\_1\_bCounter1\_CR0], ~DUALADC\_1\_fDBLK\_ENABLE ; Disable the Counter

and reg[DUALADC\_1\_bCounter2\_CR0], ~DUALADC\_1\_fDBLK\_ENABLE ; Disable the Counter

IF DUALADC\_1\_NoAZ

or reg[DUALADC\_1\_bfADC1cr2], DUALADC\_1\_fAutoZero ; Put the Integrator into Autozero mode

or reg[DUALADC\_1\_bfADC2cr2], DUALADC\_1\_fAutoZero ; Put the Integrator into Autozero mode

ENDIF

or reg[DUALADC\_1\_bfADC1cr3], DUALADC\_1\_fFSW0 ; Put Integrator into reset

or reg[DUALADC\_1\_bfADC2cr3], DUALADC\_1\_fFSW0 ; Put Integrator into reset

push A

M8C\_DisableIntMask DUALADC\_1\_bfPWM16\_INT\_REG, DUALADC\_1\_bfPWM16\_Mask ; Disable interrupts

M8C\_DisableIntMask DUALADC\_1\_bfCounter1\_INT\_REG, DUALADC\_1\_bfCounter1\_Mask

M8C\_DisableIntMask DUALADC\_1\_bfCounter2\_INT\_REG, DUALADC\_1\_bfCounter1\_Mask

pop A

RAM\_EPILOGUE RAM\_USE\_CLASS\_1

ret

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_fIsData

; FUNCTION NAME: DUALADC\_1\_fIsDataAvailable

;

; DESCRIPTION:

; Returns the status of the A/D Data is set whenever a new data

; value is available.

;

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

;

; ARGUMENTS: None

;

; RETURNS:

; A Returns data status A == 0 no data available

; A != 0 data available

;

; 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.

;

; Currently only the page pointer registers listed below are modified:

; CUR\_PP

;

DUALADC\_1\_fIsData:

\_DUALADC\_1\_fIsData:

DUALADC\_1\_fIsDataAvailable:

\_DUALADC\_1\_fIsDataAvailable:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_bfStatus

mov A, [DUALADC\_1\_bfStatus] ; Get status byte

and A, DUALADC\_1\_fDATA\_READY ; Mask off other bits

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_iGetData1ClearFlag

; FUNCTION NAME: DUALADC\_1\_iGetData1

;

; DESCRIPTION:

; Returns the data from channel-1 of A/D. Does not check if data is available.

; iGetData1ClearFlag clears the result ready flag as well.

;

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

;

; ARGUMENTS: None

;

; RETURNS:

; A Contains the LSB of the 16 bit result

; X Contains the MSB of the 16 bit result

;

; 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.

;

; Currently only the page pointer registers listed below are modified:

; CUR\_PP

;

DUALADC\_1\_iGetData1ClearFlag:

\_DUALADC\_1\_iGetData1ClearFlag:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_iResult1

and [DUALADC\_1\_bfStatus], ~DUALADC\_1\_fDATA\_READY ; Clear Data ready bit

mov X, [(DUALADC\_1\_iResult1 + HighByte)]

mov A, [(DUALADC\_1\_iResult1 + LowByte)]

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

DUALADC\_1\_iGetData1:

\_DUALADC\_1\_iGetData1:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_iResult1

mov X, [(DUALADC\_1\_iResult1 + HighByte)]

mov A, [(DUALADC\_1\_iResult1 + LowByte)]

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_iGetData2ClearFlag

; FUNCTION NAME: DUALADC\_1\_iGetData2

;

; DESCRIPTION:

; Returns the data from channel-1 of A/D. Does not check if data is available.

; iGetData1ClearFlag clears the result ready flag as well.

;

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

;

; ARGUMENTS: None

;

; RETURNS:

; A Contains the LSB of the 16 bit result

; X Contains the MSB of the 16 bit result

;

; 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.

;

; Currently only the page pointer registers listed below are modified:

; CUR\_PP

;

DUALADC\_1\_iGetData2ClearFlag:

\_DUALADC\_1\_iGetData2ClearFlag:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_iResult1

and [DUALADC\_1\_bfStatus], ~DUALADC\_1\_fDATA\_READY ; Clear Data ready bit

mov X, [(DUALADC\_1\_iResult2 + HighByte)]

mov A, [(DUALADC\_1\_iResult2 + LowByte)]

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

DUALADC\_1\_iGetData2:

\_DUALADC\_1\_iGetData2:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_iResult1

mov X, [(DUALADC\_1\_iResult2 + HighByte)]

mov A, [(DUALADC\_1\_iResult2 + LowByte)]

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_ClearFlag

;

; DESCRIPTION:

; Clears the data ready flag.

;

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

;

; 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.

;

; Currently only the page pointer registers listed below are modified:

; CUR\_PP

;

DUALADC\_1\_ClearFlag:

\_DUALADC\_1\_ClearFlag:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_bfStatus

and [DUALADC\_1\_bfStatus], ~DUALADC\_1\_fDATA\_READY ; Clear Data ready bit

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

.ENDSECTION

.SECTION

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

; FUNCTION NAME: DUALADC\_1\_SetResolution

;

; DESCRIPTION:

; Sets A/D resolution between 7 and 13 bits.

;

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

;

; ARGUMENTS:

; A Passes the number of bits of resolution, between 7 and 13.

;

; 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.

;

; Currently only the page pointer registers listed below are modified:

; CUR\_PP

;

; This function halts the PWM and the counter to sync the A/D , but

; does not re-enable the counter or PWM. To restart the A/D, "Get\_Samples"

; should be called. It sets the resolution for both channels at the

; same time.

;

DUALADC\_1\_SetResolution:

\_DUALADC\_1\_SetResolution:

RAM\_PROLOGUE RAM\_USE\_CLASS\_4

RAM\_SETPAGE\_CUR >DUALADC\_1\_bfStatus

and reg[DUALADC\_1\_bfADC1cr2], ~DUALADC\_1\_fRES\_SET

call DUALADC\_1\_StopAD ; Stop the A/D if it is running

mov [DUALADC\_1\_bfStatus], 00h ; and clear status and old resolution

; Check for resolution to be within min and max values

cmp A,DUALADC\_1\_bMINRES ; Check low end of resolution

jnc DUALADC\_1\_CHECKHI

mov A,DUALADC\_1\_bMINRES ; Too low - load legal low value

jmp DUALADC\_1\_RES\_OK

DUALADC\_1\_CHECKHI: ; Check high end of resolution

cmp A,DUALADC\_1\_bMAXRES

jc DUALADC\_1\_RES\_OK

mov A,DUALADC\_1\_bMAXRES ; Too high - load legal Max value

DUALADC\_1\_RES\_OK:

; Calculate compare value for the PWM which

; computes the integrate time

sub A, DUALADC\_1\_bMINRES ; Normalize with min resolution

or [DUALADC\_1\_bfStatus], A

; Since min resolution is 7, 2^^7 = 128, the clock

; is running 4x so 128\*4=512 or 0x0200

add A,01h ; The MSB is 02h.

mov X,A

mov A,01h

DUALADC\_1\_CALC\_INTTIME: ; Now shift the MSB left for every bit of resolution of min (7).

asl A

dec X

jnz DUALADC\_1\_CALC\_INTTIME

DUALADC\_1\_LOAD\_INTTIME: ; Load compare value and Calc time into registers

; Since minimum resolution is 7 bits, this value will always start at 0

mov reg[DUALADC\_1\_bPWM\_IntTime\_LSB], 00h

mov reg[DUALADC\_1\_bPWM\_IntTime\_MSB], A

; Load the CalcTime into the PWM Period

mov reg[DUALADC\_1\_bPWM\_Period\_LSB], <wCalcTime

add A, >wCalcTime

mov reg[DUALADC\_1\_bPWM\_Period\_MSB],A

RAM\_EPILOGUE RAM\_USE\_CLASS\_4

ret

.ENDSECTION

; End of File DUALADC\_1.asm