;======================================================================

;

; Author : ADI - Apps www.analog.com/MicroConverter

;

; Date : 30 April 1999

;

; File : master.asm

;

; Hardware : ADuC812

;

; Description : Code for a master in an I2C system.

;

; Reference : Tech Note, uC001: "Using the ADuC812 I2C Interface"

; find it at www.analog.com/microconverter

;

;======================================================================

$MOD812 ; use ADuC812 & 8052 predefined symbols

BITCNT DATA 8h ; bit counter for I2C routines

BYTECNT DATA 030h ; byte counter for I2C routines

SLAVEADD DATA 032h ; slave address for I2C routines

FLAGS DATA 28h

NOACK BIT FLAGS.0 ; I2C no acknowledge flag

BUSY BIT FLAGS.1 ; I2C busy flag

ERROR BIT FLAGS.2 ; I2C error flag

MISTAKE BIT P3.4

;======================================================================

CSEG

ORG 0000H

JMP START

;======================================================================

ORG 007BH ; Subroutines

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

; DELAY: Create a delay for the main signals ( SCLOCK , SDATA )

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

DELAY:

NOP

RET

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

; SENDSTOP: Send the bit stop of the transmission

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

SENDSTOP:

SETB MDE ; to enable SDATA pin as an output

CLR MDO ; get SDATA ready for stop

SETB MCO ; set clock for stop

ACALL DELAY

SETB MDO ; this is the stop bit

CLR BUSY ; bus should be released

RET

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

; SENDBYTE: Send a 8-bits word to the slave

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

SENDBYTE:

MOV BITCNT,#8 ; 8 bits in a byte

SETB MDE ; to enable SDATA pin as an output

CLR MDO

CLR MCO

LOOP: RLC A ; send one bit

MOV MDO,C ; put data bit on pin

SETB MCO ; send clock

CLR MCO ; clock is off

DJNZ BITCNT,LOOP

CLR MDE ; release data line for acknowledge

SETB MCO ; send clock for acknowledge

JNB MDI,NEXT ; this is a check

SETB NOACK ; no acknowledge

NEXT: CLR MCO ; clock for acknowledge

RET

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

; BITSTART: Send the bit start of the transmission and the slave

; address to the slave

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

BITSTART:

SETB BUSY ; I2C is in progress

SETB MDE ; to enable SDATA pin as an output

CLR NOACK

CLR ERROR

JNB MCO,FAULT

JNB MDO,FAULT

CLR MDO ; this is

ACALL DELAY ; the

CLR MCO ; start bit

FAULT: CLR MISTAKE ; set error flag

MOV A,SLAVEADD ; Get slave address

ACALL SENDBYTE ; call routine to send slave addr. byte

RET

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

; SENDATA: Send all the sequence to the slave ( slave address + data )

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

SENDATA:

ACALL BITSTART

JB MDI,NEXT1

MOV A,#00

SLOOP: MOVX A,@DPTR

ACALL SENDBYTE

INC DPTR

JB NOACK,NEXT1

DJNZ BYTECNT,SLOOP

NEXT1: ACALL SENDSTOP

MOV A,FLAGS

ANL A,#07h

JZ RETOUR

CLR P3.4

CLR I2CRS

RETOUR: RET

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

; RCVBYTE: receives one byte of data from an I2C slave device.

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

RCVBYTE:

MOV BITCNT,#8 ;Set bit count.

CLR MDE ;Data pin of the master is now an input

CLR MCO

LOOP2: SETB MCO

CLR MCO

MOV C,MDI ;Get data bit from pin.

RLC A ;Rotate bit into result byte.

DJNZ BITCNT,LOOP2 ;Repeat until all bits received.

;result byte is in the accumulator

PUSH ACC ;Save result byte in the stack

SETB MDE ;Data pin of the master must be an..

;..output for the acknowledge

MOV A,BYTECNT

CJNE A,#1,SACK ;Check for last byte of frame.

SETB MDO ;Send no acknowledge on last byte.

SJMP NACK

SACK: CLR MDO ;Send acknowledge bit.

NACK: SETB MCO ;Send acknowledge clock.

POP ACC ;Restore accumulator

ACALL DELAY

CLR MCO

SETB MDO ;Clear acknowledge bit.

ACALL DELAY

CLR MDE

RET

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

; RCVDATA: receives one or more bytes of data from an I2C slave device.

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

RCVDATA: INC SLAVEADD ;Set for READ of slave.

ACALL BITSTART ;Acquire bus and send slave address.

JB NoAck,RDEX ;Check for slave not responding.

RDLoop: ACALL RCVBYTE ;Receive next data byte.

MOV @R1,A ;Save data byte in buffer.

INC R1 ;Advance buffer pointer.

DJNZ BYTECNT,RDLoop ;Repeat untill all bytes received.

RDEX: ACALL SENDSTOP ;Done, send an I2C stop.

RET

;======================================================================

; Main program

;======================================================================

START:

MOV SP,#040h

CLR NOACK

MOV SLAVEADD,#088H

MOV BYTECNT,#3

MOV I2CCON,#0A8h

; code for a write mode ( master-transmitter to slave-receiver )

; MOV DPTR,#080H ; master transmits to slave

; MOV A,#055H ; datas which are located in

; MOVX @DPTR,A ; the external memory

; MOV DPTR,#081H

; MOV A,#044H

; MOVX @DPTR,A

; MOV DPTR,#082H

; MOV A,#033H

; MOVX @DPTR,A

; MOV DPTR,#080h

; ACALL SENDATA

; code for a read mode ( master reads immediately after first byte )

MOV R1,#035h

ACALL RCVDATA

END