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

;

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

;

; Date : 2nd July 2003

;

; File : pllcon.asm

;

; Hardware : ADuC845

;

; Description : Demonstrates that the CPU can run at different

; speeds determined by the CD bits in the PLLCON SFR.

; 2 to the power of CD (a 3 bit number), is the divider

; ratio that determines the clock frequency at which

; the CPU will run. (CD=0 =>fcore=12.58MHz,

; CD=7 => fcore=98.3kHz)

;

;

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

$MOD845 ; Use ADuC845 predefined symbols

LED EQU P3.4 ; P3.4 drives red LED on eval board

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; BEGINNING OF CODE

CSEG

ORG 0000h

JMP MAIN ; jump to main program

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; INTERRUPT VECTOR SPACE

ORG 0003h ; (INT0 ISR)

CPL LED ; complemant LED to indicate INT0

; press.

MOV R2,#136 ; reinitialise R7 and R6 so that

MOV R3,#256 ; after interrupt the full delay

; loop is completed

MOV A, PLLCON ; Only increment CD bits of PLLCON

INC A ; Rollover to PLLCON = xxxxx000b (fmax)

ANL A, #07h ; after PLLCON = xxxxx111b (fmin)

MOV PLLCON, A ; where the x's are 1's and 0's as rqd

RETI

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

ORG 0060H ; Start code at address above interrupts

MAIN:

MOV PLLCON, #00H

SETB IT0

SETB EX0 ; enable ext int INT0

; (button on eval board)

SETB EA ; enable interrupts

BLINK: CPL LED

CALL DELAY ; wait for 840000 machine cycles

;

JMP BLINK

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

DELAY: ; This loop delays the program for 840000

; (approx) machine cycles,

MOV R4, #8 ;

DLY0: MOV R2,#136 ;

DLY1: MOV R3,#256 ;

DJNZ R3,$ ;

;

DJNZ R2,DLY1 ; repeat 136 times

DJNZ R4,DLY0

RET

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

END