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

;

; Author : ADI - Apps

;

; Date : 11/9/99

;

; File : Slave.asm

;

; Hardware : ADuC824

;

; Description : This slave program transmits the numbers 11-20 in

; binary form continuously down the spi port after

; receiving a clock signal.

;

; After the transmission of each byte the incoming

; byte is saved in order at a internal RAM address

; between #40h and #50h

;

; This program can be used with the master program

; spimast.asm (which generates a clock signal for

; the slave)

; The Slave program (spislav.asm) should be started

; after the master program (spimast.asm) but within

; the time delay of 5s in order that the slave

; program is synchronised by the first outputted

; clock of the master.

;

; The clock is inputted at sclock (pin 26)

; The data is outputted at MISO (pin 14)

; The data is inputted at sdata/MOSI (pin 27)

;

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

;

$MOD824 ;Use 8052 predefined Symbols

LED EQU P3.4

FLAG BIT 00H

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

; BEGINNING OF CODE

CSEG

ORG 0000H

JMP MAIN

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

; SPI INTERRUPT ROUTINE

ORG 003BH

CLR FLAG ; Clear flag to leave loop

MOV @R1, SPIDAT ; move input into memory

INC R1 ; increment memory location so new

; data is stored in new address

CJNE R1, #50H, CONT ; reset memory location to 40h when

; memory location reaches 50h saving

; 16 bytes of data

MOV R1, #40H

CONT: RETI

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

ORG 0060H ; Start code at address above interrupts

MAIN: ; Main program

MOV SPICON,#24h ; Initialise SPICON to have

; -Enable SPI serial port

; -slave mode select

; -CPOL=0, clk idling low

; -CPHA=1

; note: it is important to have CPHA in the master and the slave

; program equal, otherwise uncertainty will exist, as the input

; will be measued during its change of state, and not is at

; its final value.

MOV IEIP2, #01h ; Enable SPI interrupt

MOV R0, #0AH ; initialise R0 to 10 to start

; transmission from 11

MOV R1, #40h ; initialise R1 to 40 to store the

; input data from memory location 40

SETB EA ; Enable interrupts

TRNSMT:

INC R0

MOV SPIDAT, R0 ; transmit the current value on R0

SETB FLAG ; set flag so that we wait here until

; the spi interrupt routine clears

; the FLAG

JB FLAG, $ ; stay here until flag is cleared

; by interrupt

; check if R0 is equal to 20. If so the number 20 has been

; transmitted and we should reset R0 to 10 to start transmission

; from 11 again.

MOV A, R0

CJNE A, #14H, TRNSMT ; if R0 is not 20, jump to TRNSMT

MOV R0, #0AH ; if R0=20 make R0=10 & jump to TRNSMT

JMP TRNSMT

END