Registradores de funções especiais

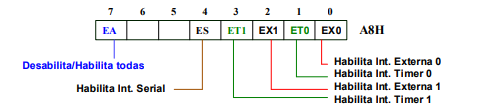
Acumulador (E0H), B(F0H), PSW(D0H), IP(B8H), P3(B0H), IE(A8H), P2(A0H), SCON(98H), SBUF(99H), P1(90H), TCON(88H), TMOD(89H), TL0(8AH), TL1(8BH), TH0(8CH), TH1(8DH), P0(80H), SP(81H), DPTR(DPL 82H e DPH 83H), PCON(87H)

Temporizadores/contadores e interrupções:

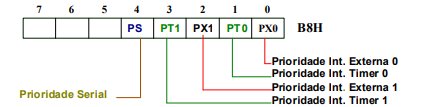
O 8051 possui dois módulos temporizadores/contadores (Timer 0 e Timer 1) de 16 bits. Quando na função temporizador, o módulo utiliza um sinal de referência 12 vezes menor que a frequência do oscilador, isto é, na frequência do ciclo de máquina. Já na função contador, esse módulo utiliza um pino de entrada para contar pulsos. Os pinos de entrada desses dois módulos são: P3.4 para o Timer0 e P3.5 para o Timer1. A configuração desses módulos é realizada pelos registradores TCON e TMOD. Além disso, esses módulos podem gerar interrupções quando a condição de overflow for detectada, isto é, quando o valor máximo de contagem for obtido. Deve-se lembrar que o registrador IE tem que ser configurado, caso o programador queira que esses módulos gerem interrupções. Caso haja a necessidade de mudar o nível de prioridade de uma fonte de interrupção, o registrador IP também deverá ser configurado.

Registradores:

IE (endereço A8H)



IP (endereço B8H)



TCON (endereço 88H)

TCON.PNG

Esses flip-flops são configurados pelo próprio hardware:

TF0 e TF1 em 0: Quando o processador inicia a rotina de interrupção localizada no endereço 000BH ou 001BH.

TF0 e TF1 em 1: Quando ocorre overflow do timer 0 ou do timer 1.

Esses flip-flops são configurados pelo próprio programador:

TR0 e TR1 em 0: Desativa o timer 0 ou o timer 1.

TR0 e TR1 em 1: Ativa o timer 0 ou o timer 1.

TMOD (endereço 89H)



Gate em 0: O módulo será habilitado se o flip-flop TR1 do registrador TCON for igual a 1.

Gate em 1: O módulo será habilitado se o flip-flop TR1 do registrador TCON for igual a 1 e o pino INT estiver em nível lógico alto (utilizado para medir a largura do pulso de entrada no pino INTx).

C/T em 0: Temporizador.

C/T em 1: Contador.

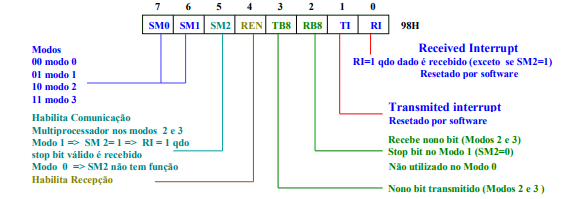
M1/M0 em 00: Modo 0 – Temporizador/Contador de 13 bits (8 bits de THx e 5 bits menos significativos de TLx). THx incrementa apenas quando ocorre um overflow em TLx.

M1/M0 em 01: Modo 1 – Temporizador/Contador de 16 bits (8 bits de TLx e 8 bits de THx).

M1/M0 em 10: Modo 2 – Temporizador/Contador de 8 bits (8 bits de TLx). THx é carregado com um certo valor Y. Quando ocorre um overflow em TLx, o valor Y de THx é carregado em TLx, THx permanece inalterado.

M1/M0 em 11: Modo 3 – Timer dividido. Acontece uma independência dos contadores TH0 e TL0 como dois contadores de oito bits que quando sofrem overflow chamam as interrupções 1 e 0, respectivamente. O timer 1 não pode gerar interrupção quando o timer 0 estiver no modo 3, pois o seu TF1 está ocupado pelo overflow do TH0 (não entendi a vantagem desse modo)

SCON (endereço 98H)



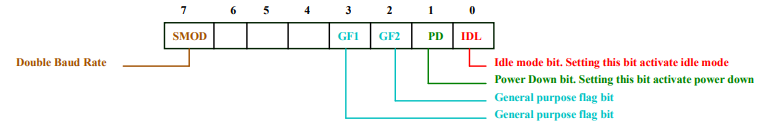
SM0/SM1 em 00: Modo 0 – ??

SM0/SM1 em 01: Modo 1 – ??

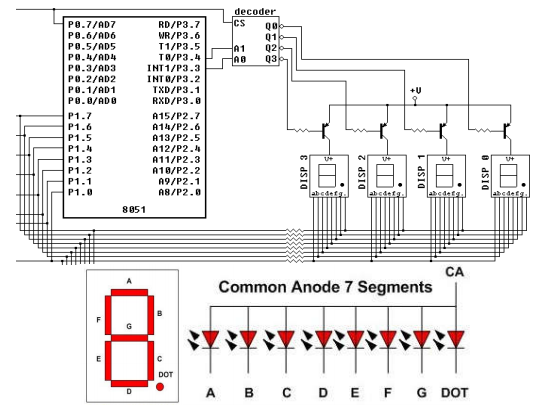
SM0/SM1 em 10: Modo 2 – ??

SM0/SM1 em 11: Modo 3 – ??

PCON (endereço ??H)



Modelo parcial do EDSIM51:



Programa em C para printar os valores de 0 a F no display 3:

//biblioteca obrigatória

#include <reg51.h>

//definida a função que irá converter os dados na porta P1

unsigned char converte\_7seg (unsigned char);

void main (void)

{

//variável i

short i;

//variável j

unsigned char j=0;

//deixa CS=1

P0=0x80;

//creio que não necessariamente precisasse ser FF, porém desse jeito A1=A0=1  
//selecionando o display 3

P3=0xFF;

while (1) {

if ( j == 16) j=0;

// atraso para Edsim 2.1.15

for (i = 0; i < 15000; i++);

//saída pro display 3

P1=converte\_7seg(j);

j++;

}

}

// função que retorna o valor a ser escrito no display

unsigned char converte\_7seg (unsigned char dado)  
{

unsigned char led;

//tabela de conversão

switch (dado)

{

case 0: led = 0x40; break;

case 1: led = 0x79; break;

case 2: led = 0x24; break;

case 3: led = 0x30; break;

case 4: led = 0x19; break;

case 5: led = 0x12; break;

case 6: led = 0x02; break;

case 7: led = 0x78; break;

case 8: led = 0x00; break;

case 9: led = 0x10; break;

case 10: led = 0x08; break;

case 11: led = 0x03; break;

case 12: led = 0x46; break;

case 13: led = 0x21; break;

case 14: led = 0x06; break;

case 15: led = 0x0E; break;

default: led = 0x80;

}

return led;

}

Interface Serial

O 8051 tem uma máquina de estados capaz de transmitir serialmente palavras de 8 ou 9 bits independentemente da participação da UC através do protocolo RS232. O modo de funcionamento da interface é programado através do uso dos dois flip flops mais significativos do SCON (SM0 e SM1). As palavras são transmitidas e recebidas através do registrador SBUF.

No modo síncrono o byte em SBUF é transmitido através do pino Rx(P3.0) e o clock de sincronismo é transmitido pelo Tx(P3.1). O sinal de sincronismo é sempre enviado pelo dispositivo que está enviando o dado.

No modo assíncrono, a transmissão de dados é feita através do pino Tx(P3.1), caso deseje-se transmitir um nono bit (modos 2 e 3), o seu valor será carregado no flip flop TB8 do SCON. Uma transição de um para zero identifica um start bit um stop bit é caracterizado pelo nível lógico 1 durante ao menos o período de transmissão de um bit.

Quando a transmissão for concluída, o flip-flop TI é setado pelo hardware e o programa é desviado para o tratador localizado na posição de memória de código 0023H. O programador deve limpar o flip-flop TI no tratador para permitir uma nova transmissão, analogamente ao RI no caso do recebimento de dados.

Para que se programe o recebimento de dados, deve-se setar o flip-flop REN em 1, e nesse caso, o tratador de interrupção deve verificar a origem da interrupção (RI=1 - buffer de recepção cheio; TI=1 - buffer de transmissão vazio).

Modo síncrono:

* Rx faz a transmissão de dados
* Tx supre o clock
* Não é possível receber e transmitir dados simultaneamente (half-duplex)

Modo assíncrono:

* Rx faz a transmissão serial
* RB8 do SCON recebe o nono bit nos modos 2 e 3
* É possível receber e transmitir dados simultaneamente (full-duplex)

Comunicação Multiprocessadores (Modos 2 e 3):

* SM2=1 habilita a comunicação multiprocessadores
* Quando acontece a comunicação, se o nono bit recebido for 1, são chamados tratadores de interrupção de recebimento dos 8051 escravos. Os outros bits dessa mesma palavra contêm o endereço do 8051 com que o mestre quer se comunicar. O 8051 que reconhece o seu endereço limpa o seu SM2 e começa a se comunicar através do barramento através de palavras de dados (bit 9 = 0). No fim da comunicação o código deve setar o SM2 do slave.

