

Projeto de um cronômetro digital usando Assembly e 8051

Erik Melges - 12547399
Gabriel Sotto Rodrigues - 11800952
Pedro A. B. Grando - 12541766

1 Programa desenvolvido

1.1 Apresentação do Código

```
ORG 00h
SELECT:
    SetB P0.7 ; chip selection
    SetB P3.3 ; select display
    SetB P3.4 ; select display
    MOV R3, #00Ah ; inicia R3 com o valor 10
    MOV DPTR,#LUT ; Inicia DPTR com o endereço do primeiro elemento na LUT
MAIN:
    JNB P2.0, SLOW ; Se SW0 pressionado, ir para rotina de 0.25s
    JNB P2.1, FAST ; se SW1 pressionado, ir para rotina de 1s
    SJMP MAIN; se nada foi pressionado, volta para a MAIN

INIT:
    MOV R3, #00Ah ; inicia R3 com o valor 10
    MOV DPTR,#LUT ; Inicia DPTR com o endereço do primeiro elemento na LUT
    RET; retorna para o local em que foi chamada a subrotina
SLOW:
    ACALL DELAY_250MS ; Delay de 0.25s
    JNB P2.1, FAST; Se SW1 for pressionado, vai para rotina de 1s
    ACALL UPDATE_DISPLAY ; Vai para UPDATE_DISPLAY
    DJNZ R3, SLOW; enquanto R3 !=0, volta para rotina
    ACALL INIT; chama a subrotina INIT
    SJMP SLOW; volta para o inicio da subrotina

FAST:
    ACALL DELAY_1S ; Delay de 1s
    JNB P2.0, SLOW; se SW0 for pressionado, vai para rotina de 0.25s
```

```

ACALL UPDATE_DISPLAY ;vai para UPDATE_DISPLAY
DJNZ R3, FAST; enquanto R3 !=0, volta para rotina
ACALL INIT; chama a subrotina INIT
SJMP FAST; volta para o inicio da subrotina

```

UPDATE_DISPLAY:

```

CLR A ; Limpa o acumulador
MOVC A,@A+DPTR ; Coloca o valor que est no endere o de mem ria mostrad
MOV P1,A ; Atualiza o display com o valor de A
INC DPTR ; Incrementa DPTR para o pr ximo d gito na LUT
RET; retorna para o local em que foi chamada a subrotina

```

DELAY_1S:

```

MOV R2, #4 ;Inicia R2 com o valor 4

```

DELAY_LOOP_1S:

```

ACALL DELAY_250MS ; Chama a subrotina
DJNZ R2, DELAY_LOOP_1S ; Enquanto R2 !=0, chama a subrotina
RET ; retorna ao local da chamada

```

DELAY_250MS:

```

MOV R1, #500 ; Inicia R1 com o valor 500
ACALL DELAY ; chama a subrotina

```

DELAY:

```

MOV R0,#250 ; Inicia R0 com 250
DJNZ R0, $ ; enquanto R0 != 0, fica em loop
DJNZ R1, DELAY ; Enquanto R1 !=0, chama a subrotina

```

RET

;define uma nova origem

ORG 0200H

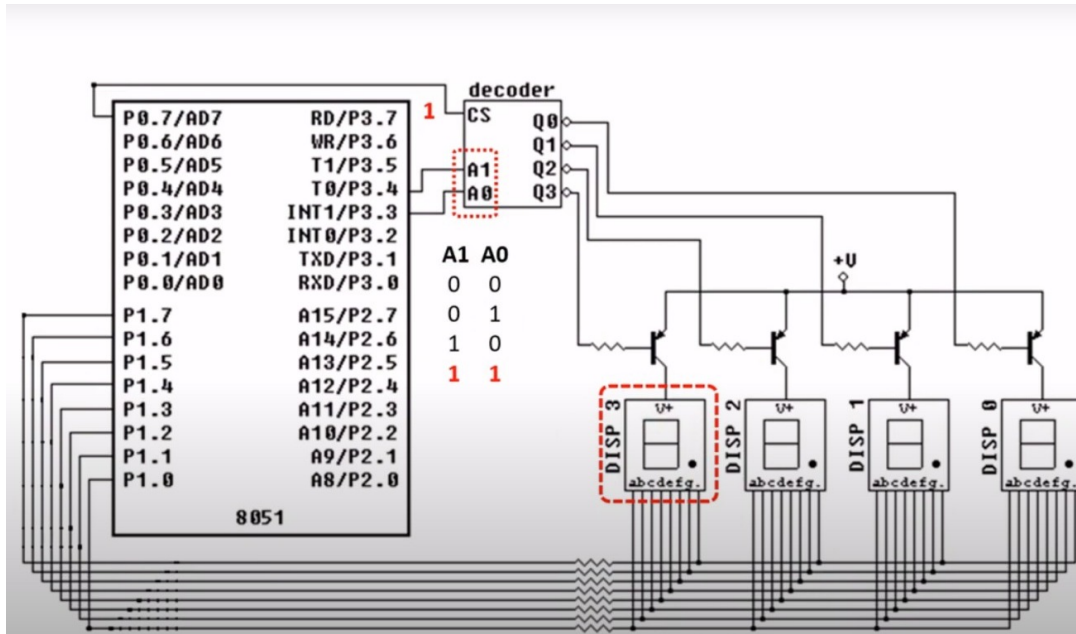
;Coloca os valores de 0 a 9 em hexa

LUT: DB 0C0h,0F9h,0A4h,0B0h,099h,092h,082h,0F8h,080h,090h,000h

END

1.2 Discussão sobre o código

SELECT: Neste bloco, alguns pinos são configurados no microcontrolador 8051. O código SetB é usado para definir os bits correspondentes nos registradores de controle de porta (P0.7, P3.3 e P3.4) para ativar a seleção do decoder e um display. Além disso, o registrador R3 é carregado com o valor 10 (0Ah), e o registrador de ponteiro de dados (DPTR) é configurado para apontar para o início de uma tabela de pesquisa (LUT). A seguir podemos ver um esquemático que mostra os bits necessários a serem ativados para selecionar o display apontado



Pela imagem, ao setar os bits P3.3 e P3.4, o último display a esquerda é selecionado.

MAIN: Este é o ponto de entrada primal do programa. Ele verifica as condições dos botões SW0 (P2.0) e SW1 (P2.1). Se SW0 for pressionado, ele salta para a sub-rotina SLOW, e se SW1 for pressionado, ele salta para a sub-rotina FAST. Se nenhum botão for pressionado, ele entra em um loop infinito SJMP MAIN para continuar verificando.

INIT: Esta sub-rotina é chamada para reinicializar algumas configurações, como definir novamente o registrador R3 para 10 (0Ah), já que é necessário redefinir R3 a cada loop, e configurar o ponteiro de dados para o início da tabela de pesquisa (LUT). Em seguida, a instrução RET é usada para retornar à chamada anterior, que geralmente seria SLOW ou FAST.

SLOW: Esta sub-rotina é usada quando o botão SW0 é pressionado. Ela chama DELAY_250MS para criar um atraso de 0,25 segundos, verifica novamente o estado do botão SW1 e, em seguida, chama UPDATE_DISPLAY. Se R3 não for zero, ela decrementa R3 e retorna a SLOW. Se R3 for zero, ela chama INIT e reinicia o processo.

FAST: Esta sub-rotina é usada quando o botão SW1 é pressionado. Ela chama DELAY_1S para criar um atraso de 1 segundo, verifica novamente o estado do botão SW0 e, em seguida, chama UPDATE_DISPLAY. Da mesma forma que em SLOW, se R3 não for zero, ele decrementa R3 e retorna a FAST. Se R3 for zero, ele chama INIT e reinicia o processo.

UPDATE_DISPLAY: Nesta sub-rotina, o acumulador é limpo (CLR A), e o valor da tabela de pesquisa é carregado no acumulador usando MOVC. O valor no acumulador é então movido para o pino de saída P1 para atualizar o display. O ponteiro de dados é incrementado para apontar para o próximo dígito na tabela de pesquisa, e a sub-rotina retorna.

DELAY_1S: Esta sub-rotina cria um atraso de 1 segundo chamando repetidamente DELAY_250MS quatro vezes.

DELAY_250MS: Esta sub-rotina cria um atraso de 0,25 segundos chamando DELAY repetidamente.

DELAY: Sub-rotina auxiliar que criar atrasos aguardando até que os contadores R0 e R1 cheguem a zero.

O delay funciona da seguinte maneira, considerado que cada ciclo de clock é de $1\mu s$, fazendo $R0 = 250$ e $R1 = 500$, e sabendo que as instruções DJNZ tomam dois ciclos de clock para serem executadas, para a sub-rotina de 250MS teremos:

$$\Delta T = 250 \times 500 \times 2\mu s = 250ms$$

Como a sub-rotina de 1s chama essa sub-rotina 4 vezes, teremos o total de 1s gasto.

LUT: Neste bloco de dados, os valores hexadecimais que representam dígitos para exibição são armazenados em uma tabela de pesquisa. Cada valor na tabela corresponde a um dígito de 0 a 9 em um display. Os valores em hexa foram retirados da seguinte tabela:

Digit	dp	g	f	e	d	c	b	a	Hex
0	1	1	0	0	0	0	0	0	C0
1	1	1	1	1	1	0	0	1	F9
2	1	0	1	0	0	1	0	0	A4
3	1	0	1	1	0	0	0	0	B0
4	1	0	0	1	1	0	0	1	99
5	1	0	0	1	0	0	1	0	92
6	1	0	0	0	0	0	1	0	82
7	1	1	1	1	1	0	0	0	F8
8	1	0	0	0	0	0	0	0	80
9	1	0	0	1	0	0	0	0	90

Em suma, o programa opera com base nas entradas dos botões SW0 e SW1, criando atrasos e atualizando um display de acordo com as ações do usuário. Ele permite que o usuário escolha entre dois modos de exibição com diferentes atrasos e comportamentos.