Cronômetro Digital

SEL0614 - Aplicação de Microprocessadores

Professor:

- Pedro Oliveira

Alunos:

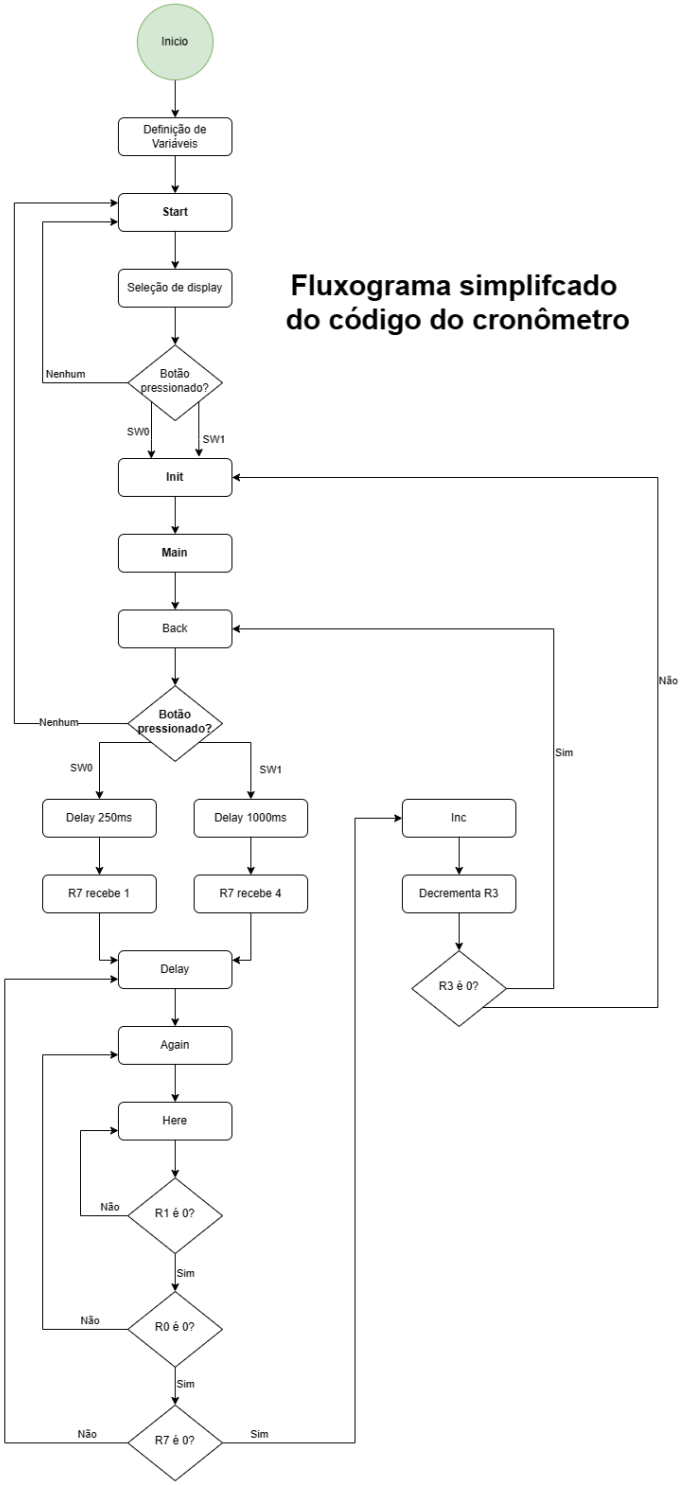
- Gustavo Moura Scarenci de Carvalho Ferreira, nºUSP 12547792
- Matheus Henrique Dias Cirillo, nºUSP 12547750

Código Comentado

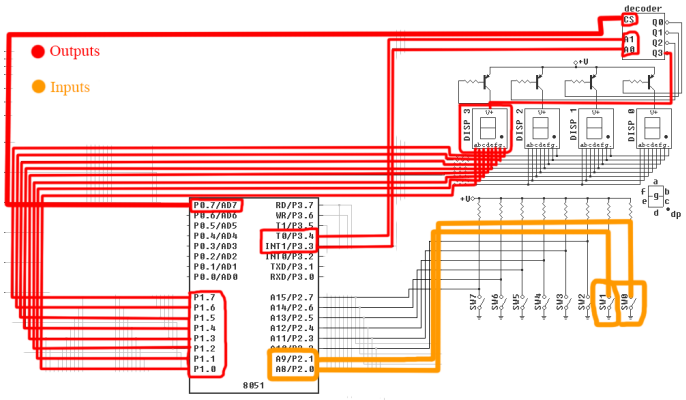
```
SW0 Equ P2.0                ;Definição de variavel para legibilidade do código
SW1 Equ P2.1                ;Definição de variavel para legibilidade do código
CS EQU P0.7                 ;Definição de variavel para legibilidade do código
A0 EQU P3.3                 ;Definição de variavel para legibilidade do código
A1 EQU P3.4                 ;Definição de variavel para legibilidade do código

org 00h
Start:
                CLR CS
                JNB SW0, Select     ;Caso SW0 pressionado (low) inicializa o programa
                JNB SW1, Select     ;Caso SW1 pressionado (low) inicializa o programa
                SJMP Start          ;Volta para o start esperando SW0 ou SW1 ser pressionado

Select:
                SETB CS             ;Chip select
                SETB A0             ;Select Disp
                SETB A1             ;Select Disp

Init:           Mov R3,#00Ah        ;Inicializador do loop para rodar o programa 10(0Ah) vezes
Main:
                MOV DPTR,#LUT       ;Traz a lookup table para o DPTR
Back:           CLR A               ;Limpa o valor de A e estabelece o ponto de retorno do loop 0 a
9
                MOVC A,@A+DPTR      ;Insere DPTR com um offset de A em A
                MOV P1,A            ;Mostra A no display selecionado
                ACALL Press_check       ;Verifica se algum botão foi pressionado
                JNB SW0, Delay_250ms    ;Caso SW0 pressionado (low) faz o delay de 250ms
                JNB SW1, Delay_1000ms   ;Caso SW1 pressionado (low) faz o delay de 1000ms
Number_inc:     INC DPTR            ;Próximo valor da tabela de procura e ponto de retorno dos
delays,
                                    ;com esse ponto de retorno caso SW0 e SW1 estejam pressionados
o delay não sera 1250ms
                DJNZ R3,Back        ;Decrementa nosso loop de rodar o programa em 10 vezes e
retorna para back
                SJMP Init           ;Caso ele passe do decremento e retorno do loop retorna para
```

```
init e reinicia a contagem

Delay_250ms:
                Mov R7, #01          ;Coloca 1 em R7 para que o loop de delay seja realizado apenas
1x
                Jmp Delay            ;Vai para a função de delay

Delay_1000ms:
                Mov R7, #04          ;Coloca 4 em R7 para que o loop de delay seja realizado 4x
                Jmp Delay            ;Vai para a função de relay

Delay:
                MOV R0, #200         ;Loop geral (1x para 250ms e 4x para 1000ms)
Again:          MOV R1, #230         ;Loop externo (reinicia o loop interno)
Here:           NOP                  ;Loop interno
                NOP
                DJNZ R1, Here        ;Continua o loop interno enquanto R1 não for 0
                DJNZ R0, Again       ;Continua o loop externo enquanto R0 não for 0
                DJNZ R7, Delay       ;Continua o loop geral enquanto R7 não for 0
                JMP Number_inc       ;Retorna do delay para o ponto de incremento

    ;Calculo de loop:
        ;No loop interno temos 1 + 1 + 2 ciclos de máquina rodando 200 vezes, cada ciclo tem
1.085us com 11.0592MHz = 1247.75us
        ;O loop externo faz com que o loop interno rode 200 vezes = 249550us = 249.55ms
        ;O overhead do loop externo 1 + 2 ciclos de maquina rodando 200 vezes = 651us
        ;O loop geral também tem um overhead para cada delay:
            ;250ms -> 1 + 2 ciclos de máquina rodando 1 vez = 3.255us
            ;1000ms -> 1 + 2 ciclos de máquina rodando 4 vezes = 13.02us
        ;Logo, para 250ms teremos um tempo total de 250204.255us = 250.20ms
        ;e para 1000ms teremos um tempo total de 1000856.08us = 1000.86ms

Press_check:
                Mov A, P2            ;Coloca o valor de P2 em A
                ANL A, #0FFh         ;Faz um AND com 11111111b para verificar se SW0 ou SW1 estão
pressionados
                CPL A                ;Inverte o valor de A efetivamente realizando um NAND pra SW0 e
SW1
                                     ;nesse ponto os 6 bits mais significativos de A são 0 e os 2
menos significativos contem informação
                JZ Start             ;Caso ninguém pressionado retorna para o start
                RET                  ;caso algum pressionado retorna para o fluxo do código

Org 0200h

LUT:            DB 0C0h, 0F9h, 0A4h, 0B0h, 99h, 92h, 82h, 0F8h, 80h, 90h, 01
    ;Lookup table com os codigos para mostrar os digitos em 7 segmentos

END
```

Fluxo simplificado do código



Interfaces



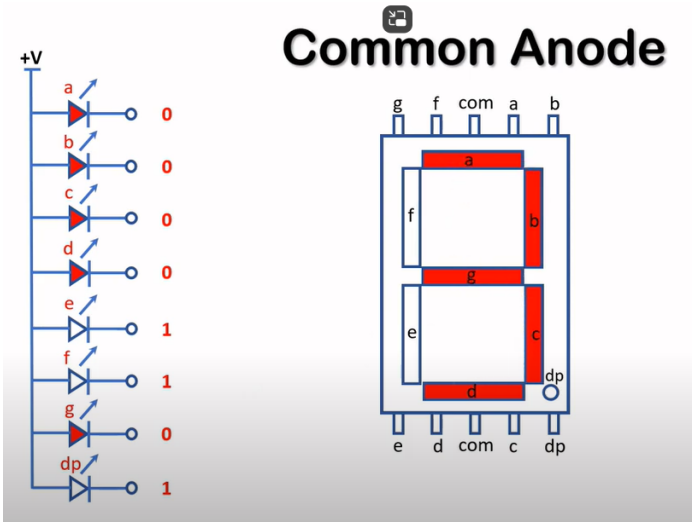
Utiliza-se CS, A0 e A1 para escolher o display que se quer exibir o número.

- CS -> ativa o chip o seletor e permite escolher um dos displays para funcionar
- A0 e A1 -> linhas de seleção 2 para 4, usando 2 sinais podemos controlar 4 saídas diferentes, o que permite escolher um dos 4 displays.

Utiliza-se P1 para exibir o valor desejado no display selecionado.

Lê-se os valores de P2.0 e P2.1 (SW0 e SW1) para definir o tempo de delay entre a troca de números.

7 segments display



Numbers table

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

Assim, caso o objetivo seja exibir o valor 7 no display, devemos colocar P1 com o valor de `#0F8h`.

Delay

SW0 e SW1 controlam o delay do contador e são entradas detectadas por P0 no bit 0 e 1. Quando SW1 está ligado a rotina de delay de 0,25s roda 4 vezes, gerando um delay de 1s. Já quando SW0 está ligado ele da um override em SW1 e faz com a rotina de delay de 0,25s rode apenas uma vez, gerando um delay de 0,25s.

Verificação de botão

Idealmente o projeto contaria com um interrupt que altera o valor de R7 quando um dos botões é pressionado, entretanto não tivemos esse conceito em aula e a solução foi manter o switch apertado e fazendo verificações antes de iniciar as funções de delay, dando preferência para SW0.

Existe também um teste para caso nenhum dos dois botões estar apertado o código desligar o display e voltar a esperar uma nova entrada.

References

Time delay calculation for various 8051 chips
8051 Instruction Set (tue.nl)
#24 EdSim51 Seven Segment Display - YouTube