Desenvolver um cronômetro digital usando a linguagem Assembly para o microcontrolador 8051 no simulador EdSim51 é um projeto interessante que combina vários conceitos de sistemas embarcados. Abaixo está um esboço de como você pode abordar esse projeto. Lembre-se de que este é um guia geral, e você pode precisar ajustar o código conforme necessário.

1. Inicialização:

- Configure os pinos de entrada e saída para conectar o microcontrolador aos botões (SW0 e SW1) e ao display de 7 segmentos.

- Inicialize quaisquer variáveis e registradores necessários.

- Desligue o display de 7 segmentos inicialmente.

2. Sub-rotinas de Delay:

- Crie sub-rotinas que gerem atrasos com base em loops de contagem, para criar os intervalos de tempo desejados.

- Por exemplo, você pode criar sub-rotinas para gerar atrasos de 0,25 segundos e 1 segundo.

3. Loop Principal:

- No loop principal, monitore constantemente o estado dos botões SW0 e SW1.

- Quando SW0 é pressionado, inicie a contagem com um atraso de 0,25 segundos.

- Quando SW1 é pressionado, inicie a contagem com um atraso de 1 segundo.

- Use instruções condicionais para verificar o estado dos botões.

- Mantenha a contagem de 0 a 9 em um loop.

4. Exibição no Display de 7 Segmentos:

- Use instruções Assembly para controlar o display de 7 segmentos.

- Atualize os valores exibidos no display com base na contagem atual (0 a 9).

5. Reinício Automático:

- Quando a contagem atingir 9, reinicie automaticamente a contagem de volta para 0.

6. Lógica de Interrupção:

- Considere o uso de interrupções se desejar permitir que o usuário interaja com o cronômetro de outras maneiras.

7. Compilação e Simulação:

- Compile o código Assembly para o formato binário/hexadecimal apropriado.

- Carregue o código no simulador EdSim51 e teste o funcionamento do cronômetro digital.

8. Depuração e Ajustes:

- Realize depuração e ajustes conforme necessário para garantir que o cronômetro funcione conforme o especificado.

Este código em Assembly é um exemplo simples para um cronômetro digital. Ele inicia a contagem quando SW0 ou SW1 são pressionados, exibe a contagem no display de 7 segmentos e permite alternar entre intervalos de 0,25 segundos e 1 segundo. Certifique-se de configurar corretamente o simulador EdSim51 para testar o código. Lembre-se de que este é um exemplo simplificado e pode precisar de ajustes para funcionar perfeitamente com o simulador e o hardware específicos que você está usando.

; Definir constantes

SW0 equ P1.0 ; Constante para o botão SW0 conectado à porta P1.0

SW1 equ P1.1 ; Constante para o botão SW1 conectado à porta P1.1

DISPLAY equ P2 ; Constante para a porta P2 conectada ao display de 7 segmentos

; Definir variáveis

count equ 20 ; Valor inicial para 0,25 segundos (ajustável)

interval equ 50 ; Valor para 1 segundo (ajustável)

; Inicialização

mov P1, #0FFh ; Configura a porta P1 como entrada (botões)

mov P2, #0FFh ; Configura a porta P2 como saída (display)

mov R0, #0 ; Inicializa o registrador R0 para contagem

MainLoop:

clr DISPLAY ; Desliga o display

; Verifica se SW0 foi pressionado

jb SW0\_Pressed, StartCount025

; Verifica se SW1 foi pressionado

jb SW1\_Pressed, StartCount1

sjmp MainLoop ; Nenhum botão pressionado, continue no loop principal

StartCount025:

mov R0, #0 ; Reinicializa a contagem

acall Delay025 ; Aguarda 0,25 segundos

acall DisplayCount ; Exibe a contagem

sjmp MainLoop ; Volta ao loop principal

StartCount1:

mov R0, #0 ; Reinicializa a contagem

acall Delay1 ; Aguarda 1 segundo

acall DisplayCount ; Exibe a contagem

sjmp MainLoop ; Volta ao loop principal

Delay025:

mov R1, count

Delay025\_Loop:

djnz R1, Delay025\_Loop

ret

Delay1:

mov R1, interval

Delay1\_Loop:

djnz R1, Delay1\_Loop

ret

DisplayCount:

mov A, R0

add A, #48 ; Converter o valor em ASCII

mov DISPLAY, A ; Exibir no display

inc R0 ; Incrementar a contagem

cjne R0, #10, MainLoop ; Se R0 != 10, continue no loop principal

mov R0, #0 ; Reinicializar a contagem

sjmp MainLoop

SW0\_Pressed:

jnb SW0, $ ; Espera até que o botão SW0 seja liberado

ret

SW1\_Pressed:

jnb SW1, $ ; Espera até que o botão SW1 seja liberado

ret