**Discussão sobre a Prática: Desenvolvimento de um Cronômetro Digital em Linguagem C para PIC18F**

Nesta prática, foram conceitos fundamentais relacionados a microcontroladores de 8 bits, especificamente utilizando o PIC18F4550, e desenvolvemos um cronômetro digital em linguagem C. Abordamos aspectos importantes tais como timers, interrupções, manipulação de I/O e acionamento de um display de 7 segmentos.

**Configuração dos Principais Blocos:**

* Timers (TMR0): O temporizador TMR0 foi configurado para gerar as bases de tempo para a contagem do cronômetro. A escolha do TMR0 está relacionada à sua capacidade de operar com interrupções externas, permitindo uma resposta imediata a eventos como pressionar um botão. A frequência do clock do PIC18F foi ajustada para 8 MHz, proporcionando uma base de tempo confiável para o cronômetro.
* Interrupções: As interrupções foram implementadas para lidar com eventos assíncronos, como pressionar os botões. As interrupções externas foram configuradas nos botões RB0 e RB1, utilizando a mudança de borda como condição de acionamento. Isso garante que a contagem seja iniciada apenas quando um botão for pressionado, evitando-se assim, contagens indesejadas.
* Acionamento do Display de 7 Segmentos: O controle do display de 7 segmentos foi realizado através das chaves de SW4. Cada chave é responsável por ligar um dos displays, enquanto a Porta D do PIC18F é usada para controlar os segmentos. A configuração do display como cátodo comum foi considerada, acendendo os segmentos com nível lógico 1.

**Resultados Obtidos:**

O projeto resultou em um cronômetro digital funcional que atende aos requisitos propostos. Quando o botão RB0 é pressionado, o display conta de 0 a 9 em loop, com um período de 1 segundo. Pressionando o botão RB1, a contagem ocorre com um período reduzido de 0,25 segundos. A contagem inicia apenas quando um dos botões é pressionado, proporcionando uma operação intuitiva.

**Comparação com o Projeto Anterior em Assembly (8051):**

Comparativamente ao projeto anterior em Assembly para o 8051, a implementação em linguagem C para o PIC18F oferece vantagens significativas em termos de facilidade de desenvolvimento e portabilidade. A estrutura modular da linguagem C, juntamente com bibliotecas reutilizáveis, simplifica a manutenção do código. A portabilidade do código para diferentes microcontroladores é uma vantagem importante, haja vista que ela permite adaptações mais rápidas e eficientes para diferentes plataformas.Porém, apresenta desvantagens como menor controle de baixo nível em comparação com o Assembly e possível overhead em termos de tamanho do código e tempo de execução

Em termos de desempenho, embora Assembly permita um controle mais preciso do hardware, a eficiência do código gerado pelo compilador C moderno é notável.Logo, a escolha entre C e Assembly dependerá do contexto específico do projeto, priorizando o equilíbrio entre controle de baixo nível, eficiência de desenvolvimento e desempenho.