

SEL0614 - APLICAÇÃO DE MICROPROCESSADORES



Projeto de um cronômetro digital usando Assembly e 8051

Objetivos

- Desenvolvimento de um projeto em linguagem Assembly para 8051 que explore os seguintes recursos no simulador EdSim51: registradores GPR e SFR, contagem de tempo, detecção de eventos, pilha, sub-rotinas, portas de entradas e saídas, e interfaces externas (botões, LEDs e displays de 7 segmentos).

Conceitos: sistemas embarcados, microprocessadores, microcontroladores, arquitetura e organização de computadores, set de instruções, CISC, RISC, memória de dados, memória de programa, registradores, clock, ciclos de máquina, linguagem Assembly, *timer*, *counter*, *interrupt*, display, I/O.

Material relacionado: Cap. 2 - Rev. de Org. Comp. e Microcontroladores; Cap. 3 - MCS-51; e Cap. 4 - Programação em Assembly para MCS-51.

I - Motivação

Um microcontrolador necessita de um programa (set de instruções) que normalmente é escrito em linguagem de alto, médio ou baixo nível (C, C++, MicroPython, Assembly), compilado para um código hex/bin (linguagem de máquina) que é gravado na ROM (flash, EEPROM etc.), geralmente interna. Sendo um sistema computacional, os microcontroladores podem apresentar arquiteturas e sets de instruções (CISC/RISC, Von-Neumann, Harvard)/ Alguns dos recursos importantes que foram abordados até aqui e que estão disponíveis nos microcontroladores para serem usados em programas são:

- **Pilha e subrotinas:** A pilha permite que o microcontrolador salve e recupere informações importantes durante a execução de um programa, sendo esse recurso bastante útil nas chamadas e retorno de subrotinas. As sub-rotinas permitem modularidade, a reutilização de trechos de código (podendo a mesma sub-rotina ser chamada em diferentes partes do programa), e simplificam o processo de programação.
- **Portas paralelas de entrada e saída:** permitem a comunicação de dados entre o microcontrolador e outros dispositivos, como sensores, atuadores, teclado, chaves e botões, displays, motores etc, podendo ser configuradas como entrada ou saída. Uma interface com display de 7 segmentos, por exemplo, é bastante popular pela sua simplicidade e baixo custo, comumente usada em aplicações em sistemas embarcados

e produtos que exigem a exibição de informações numéricas, como em relógios digitais/cronômetros, termômetros, medição de grandezas etc.

II- Requisitos do projeto

Escrever um programa em Assembly para 8051 no simulador EdSim51 que atenda os seguintes requisitos:

- Ao pressionar um botão/chave (por ex.: SW0), um Display de 7 segmentos deve mostrar a contagem de números na sequência de 0 a 9 em loop (ao chegar em 9, a contagem é reiniciada automaticamente), com intervalo de tempo de 0,25 s.
- Quando um segundo botão/chave (por ex. SW1) for pressionado, tal ação deve alterar o intervalo de tempo da contagem deste mesmo display para 1s, isto é, a contagem de 0 a 9 em loop continuará, porém, o display passará a contar em um período de tempo mais rápido.
- Caso SW0 seja pressionado novamente, a contagem de números retorna ao intervalo de tempo de 0,25s e vice-versa, mantendo a execução do programa em loop.
- Ao executar o programa pela primeira vez (após clicar em “Run”), a contagem não se inicia automaticamente e o display estará desligado. Somente quando uma das chaves SW0 ou SW1 for pressionada é que a contagem se inicia e o display passa a mostrar os valores de 0 a 9 nos intervalos de tempo mencionados anteriormente atribuídos para cada chave.
- Usar sub-rotinas de delay para gerar as bases de tempo do cronômetro e promover a mudança de período de tempo solicitada, bem como instruções condicionais de verificação de acionamento das chaves.
- Usar um dos displays de 7 segmentos e dois switches disponíveis no EdSim51.

III - Formato de entrega

- Apresentar em arquivo PDF (ou arquivo “README.md” no GitHub) o programa desenvolvido e devidamente comentado. Cada linha de código deve ser brevemente comentada. Adicionalmente, uma explicação/discussão sobre o programa deve ser fornecida (no máximo 1 página de texto), explicando os blocos, lógica e quais recursos foram usados e manipulados no programa (registradores, interfaces externas, portas, bases de tempo adotadas, como foi feita a varredura no display etc.). Para complementar a explicação textual, apresentar um diagrama esquemático do microcontrolador 8051 com a ligação das interfaces de entrada e saída usadas no projeto (segundo a estrutura disponível no EdSim51), e um diagrama ou tabela de como é feita a varredura no display de 7 segmentos disponível no EdSim51 para acender números de 0 a 9 usando os 8 bits do registrador da Porta P1.
- Caso preferir, ao invés de apresentar em documento de texto, poderá ser gravado um vídeo curto com a explicação breve e objetiva sobre os requisitos solicitados acima, no formato “screencast” (gravação da tela do computador com narração) compartilhando a tela do computador que mostre o programa desenvolvido e sua a execução no simulador EdSim51 enquanto explica (neste caso, as partes referentes ao diagrama do projeto e a varredura do display de 7 segmentos podem ser mostradas

usando os recursos do próprio EdSim51). A gravação da tela pode ser feita via algum software diretamente no computador ou pode ser feita pelo celular.

- Além do arquivo em pdf acima referido (ou README.md ou vídeo), enviar também o código fonte funcional desenvolvido e simulado no EdSim51 (programa em Assembly: arquivo “.asm”).
- Fazer o upload dos arquivos na respectiva tarefa atribuída no e-Disciplinas até a data especificada.
- A atividade poderá ser feita em duplas.
- Entregas atrasadas não serão consideradas ou, consideradas com o devido desconto de pontos proporcional ao tempo de atraso. O canal oficial para entrega de tarefas é por meio do e-Disciplinas . Não enviar arquivos por e-mail.
- Qualquer dúvida sobre o formato de envio ou sobre a implementação da atividade prática, entrar em contato com o professor ou com o monitor: Giovanni Oliveira de Sousa (giovanni_oliveiradesousa@usp.br - Whatsapp: 92 8114-0320).

IV - Critérios de avaliação

Item	Pontuação
<u>Entrega no formato: (arquivo em PDF com programa, discussão/diagramas; ou Readme no Github; ou vídeo) + código fonte “.asm”</u>	1
<u>Programa com as linhas de código devidamente comentadas</u> <u>Explicação e discussão textual sobre o programa, suportada com diagramas (ou vídeo com essa explicação)</u>	2
<u>Correção lógica do programa: atendimento ao enunciado e uso dos recursos solicitados, como interfaces I/O (botões, display), rotinas de delay, instruções de verificação dos eventos, programação das bases de tempo, contagem em loop no display etc.</u>	7

OBS.: não será considerado como entrega somente o envio do programa (arquivo “.asm”), sem algum arquivo com a explicação do projeto no formato solicitado.