

SEL0433/SEL0336 - APLICAÇÃO DE MICROPROCESSADORES



Prática 1 - Introdução aos recursos e funcionalidades de Microcontroladores em baixo-nível

Objetivos

- Revisar conceitos relativos às primeiras aulas do curso e o propósito da disciplina no âmbito de sistemas embarcados e microcontroladores.
- Realizar manipulação básica de dados em registradores e endereços de memória e exercitar o uso do set de instruções por meio de ferramenta de simulação computacional visando potencializar a compreensão sobre o funcionamento dos microcontroladores.
- Exercitar o uso do EdSim51, dos registradores de propósito geral e de função especial, e do set de instruções de transferência de dados, lógicas, aritméticas, booleanas, incondicionais e condicionais usando a família MCS-51.
- Desenvolver o primeiro projeto prático implementando funcionalidades de microcontroladores em “baixo-nível”, com linguagem Assembly e MCS-51.

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. 0 - apresentação da disciplina; Cap. 1 - Fundamentos de Microcontroladores; Cap. 2 - Programação em Assembly para MCS-51.

Parte 2: Projeto prático de um cronômetro com timer e interrupção em Assembly

Tendo sido exercitada a manipulação básica de dados em registradores e endereços de memória utilizando o set de instruções do MCS-51 na Atividade Prática 1. Esta segunda atividade prática versa sobre o desenvolvimento do primeiro projeto prático de sistema embarcado utilizando funcionalidades de microcontroladores em “baixo-nível” (temporização, contagem de eventos, interrupção, ativação de entradas e saídas como LEDs e botões, uso de display etc.).

2.1 - Motivação

O 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 da seguinte forma:

- **Von Neumann:** mais lento pois não permite acesso simultâneo às memórias (geralmente CISC).
- **Harvard:** Mais rápido, pois permite acesso simultâneo às memórias (geralmente RISC - permite o *pipelining*).
- **CISC:** Instruções podem ocupar espaços diferentes na memória de programa (Opcode + operando). Acabam tendo durações diferentes (mais instruções disponíveis = programas mais simples). Exemplos: **8051 – 111 instruções; processador 8085 – 150 instruções; processador Z80 – mais de 500 instruções**
- **RISC:** Cada instrução ocupa o mesmo espaço na memória de programa (Opcode + operando); Todas instruções possuem a mesma duração (exceto as de “salto”); (menos instruções disponíveis = programas mais complexos). Exemplos: **MCU PIC 16F - 35 instruções; MCU PIC 18F – 74 instruções.**

Conforme discutido, a família MCS-51 vem sendo estudada para consolidar conceitos de microcontroladores. As principais características são: **CPU de 8 bits, ROM (programa) de 4 KB, RAM (dados) de 256B, I/O de 32 bits (4 bancos de 8 registradores), barramento externo e endereçamento de até 16 bits (64 kB externa), interrupção programável via software, 2 timers/contadores de 16 bits, 1 interrupção serial, 2 interrupções a eventos externos, e comunicação serial (tais recursos são habilitados na Porta P3).** Os controles são: **Reset** via circuito externo, 2 pinos para ligação de **crystal oscilador externo**, pinos de controle de memória externa. Possui pino de alimentação **5V, GND**, registradores **SFR e GPR**, e **modos de endereçamento direto, indireto, por registrador e imediato**. Possui **111 instruções** distintas que combinadas **totalizam 255**, entre **transferência de dados, aritméticas, lógicas, booleanas, incondicionais e condicionais**, as quais foram abordadas na atividade pré-lab.

Alguns dos recursos mais importantes 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 sub-rotinas. 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.

- **Interrupções:** permitem que o microcontrolador responda a eventos externos, como uma entrada de botão, sem ter que monitorar constantemente a entrada. O processador interrompe sua rotina atual e executa uma função de tratamento de interrupção, que é responsável por lidar com o evento ou entrada externa. Tal vantagem permite que o microcontrolador execute várias tarefas simultaneamente, tornando-o mais flexível em aplicações em tempo real.
- **Temporizadores e contadores de evento (C/T):** permitem que o microcontrolador controle o tempo de execução do programa, sendo útil na criação de atrasos e medições de tempo em cronômetros digitais comumente utilizados em sistemas embarcados para controle de processo (medição do tempo de ciclos de produção, tempo de eventos, ajuste dinâmico do tempo de contagem de ciclos etc.). Os contadores permitem que o microcontrolador conte eventos externos, como pulsos de um sensor, sendo útil para aplicações de controle de velocidade e medições de frequência.

2.2 Roteiro

I - Aplicação prática

Desenvolvimento de um projeto em linguagem Assembly para 8051 que explore os seguintes recursos no simulador EdSim51: interrupções (e, consequentemente, o uso da pilha), temporização, portas de entradas e saídas, e periféricos (botões, LEDs e displays de 7 segmentos).

II - Requisitos do projeto

- Ao pressionar um botão, um Display de 7 segmentos deve mostrar a contagem de números na sequência de 0 a 9 em loop, com determinado intervalo de tempo.
- Quando um segundo botão for pressionado, tal ação deve alterar o intervalo de tempo da contagem deste mesmo display, isto é, a contagem de 0 a 9 em loop continuará, porém, o display passará a contar em um período de tempo diferente do anterior (mais rápido ou mais lento).
- Você poderá escolher os dois períodos de tempo. Usar um temporizador (modo 1 ou modo 2) e interrupções para gerar as bases de tempo do cronômetro e promover a mudança de período de tempo solicitada. Usar um dos displays de 7 segmentos e dois switches disponíveis no EdSim51.

III - Formato de entrega

- Apresentar em arquivo PDF 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 (interrupções, temporizadores, botões, 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.

- Além do arquivo em pdf acima referido, enviar também o código fonte 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 (lembrando que os/as dois/duas integrantes devem fazer a envio no e-Disciplinas).
- 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 um dos monitores: Vitor (Whatsapp: 11 97223-2728) e Guilherme (Whatsapp: 16 98221-9830).

IV - Critérios de avaliação

Item	Pontuação
<u>Formato: entrega no formato solicitado (arquivo em PDF com programa, discussão/diagramas + código fonte “.asm”)</u>	1
<u>Boas práticas de programação: programa com as linhas de código devidamente comentadas</u>	1
<u>Explicação e discussão textual sobre o programa, suportada com diagramas</u>	2
<u>Correção lógica do programa: atendimento ao enunciado e uso dos recursos solicitados, como interfaces I/O (botões, display), correta utilização de interrupções e de temporizadores etc.</u>	6

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