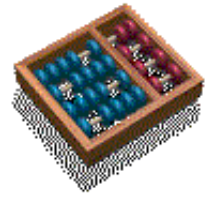




MC 404C - Arquitetura e Programação de microcontroladores

1º semestre de 2015



Profs. Célio Guimarães e Nelson Machado
Atualizado em 26/Fev/2015

Horário de atendimento: IC1 - sala 47: 2ª a 4ª 18:00 às 19:00hs

Avisos importantes:

- [Critérios de avaliação](#) e [Programa da disciplina](#).
- Login e senha para acesso aos laboratórios do IC3: na Secretaria de Graduação (IC1)
- Login e senha para acesso ao sistema Susy: RA (apenas os 6 dígitos) e senha da DAC; use também o e-mail da DAC.
- Aula às 2as Feiras: 21-23hs Sala 351, 4as feiras: 19-20hs sala 353, 20-21hs sala 304

[Overview do ARM \(Wikipedia\)](#).

[Diagrama de blocos simplificado do ARM](#). Veja aqui o [ciclo de execução de uma instrução no ARM](#) usando o diagrama acima.

Um resumo dos [modos de endereçamento do ARM](#) (traga nos laboratórios).

[Resumo das diretivas](#) do assembler do ARM (inclui exemplo de macros, traga nos laboratórios).

Como escrever [programas estruturados em assembler](#)

[Instruções adicionais thumb2, instruções de deslocamento e execução condicional de instruções](#)

[Introdução ao gdb](#) (traga nos laboratórios).

["Dicas" para depurar programas em assembler do ARM](#)

Manual detalhado de cada instrução do ARM [Cortex M3 instruction set](#)

Exercícios de [programação com o ARM](#). A **maioria será feita em aula pelos alunos**.

[Overview da arquitetura do ARM Cortex M3](#)

[E/S e interrupções no ARM Cortex M3](#).

Tutorial da Universidade de Stuttgart sobre [interrupções e exceções](#) no ARM.

[Tutorial sobre a MMU do ARM \(v7A\)](#).

Formato IEEE/ISO para números em

Sugestões para [uso do simulador jarm](#) do Prof. Anido. [ponto flutuante \(32 bits\)](#)

Atividades desafio:

A realização de uma dessas atividades dispensará o aluno das provas e da maioria das atividades subsequentes.

Condições para fazer o desafio serão negociadas caso a caso, antes da realização da 1a prova..

[Atividade desafio 1 \(opcional, individual\): extensão do algoritmo do número mágico de Kaprekar](#)

[Atividade desafio2 \(opcional, individual\): implementação de uma rotina printf simplificada](#)

Esta atividade tem uma aplicação direta no protótipo do Arm Cortex 3 mostrado em aula,

onde um terminal virtual é implementado usando a interface Usart do microcontrolador.

Atividade desafio3 (opcional, individual): núcleo de um sistema operacional de tempo real

Atividades obrigatórias

As atividades obrigatórias serão **individuais e deverão ser submetidas via sistema Susy** (a saída de cada atividade é um arquivo texto que será gerado via função printf).

Atividade 1: [Geração da sequencia de Fibonacci com valores < 1000](#)

Data de submissão: 02/03/15 Prazo susy: 23/03/15

Exercícios

Considere a instrução do ARM: *ldr r1, [r0]*. Descreva os passos para execução dessa instrução tomando como modelo o [ciclo de execução de uma instrução](#) e usando o [diagrama de blocos da CPU do ARM](#).

Os exercícios opcionais a seguir servirão de subsídios às atividades obrigatórias, e, portanto, são altamente recomendáveis:

- Escreva um trecho de programa contendo apenas uma instrução lógico/aritmética e saltos condicionais apropriados, para verificar se um inteiro com sinal em um registrador é positivo, negativo, par ou ímpar. Desenhe os arcos onde há saltos e verifique se o número de cruzamentos de arcos é mínimo.
- Escreva uma subrotina *convnb* que toma como parâmetro de entrada em r1 um inteiro de 4 bits (nibble) e retorna em r1 a representação ascii-hexadecimal do nibble. Teste-a num laço em que r1 varia de 0 a 15, mostrando no vídeo cada valor retornado.
- Escreva uma subrotina *getnbs* que toma como parâmetro de entrada em r1 um valor binário qualquer e em r0 um apontador para um **vetor de bytes** na memória RAM. A subrotina deve extrair cada nibble de r1 e armazená-lo no vetor apontado por r0. Teste-a num laço com o valor de r1=0x89abcdef e leia e exiba no vídeo cada byte do vetor.

Os dois exercícios acima são parte integrante da atividade 2a.

- Escreva um programa que **em tempo de montagem** preenche um vetor de palavras na memória RAM com os números ímpares 3, 5, ..., 63 usando as diretivas *.equ*, *.rpt*, *.word* e *.endr*. Escreva uma rotina *showvet* para exibir o conteúdo do vetor no vídeo. **dica:** dentro do bloco delimitado por *.rpt .endr* altere o valor da constante definida via *.equ*
- Escreva uma subrotina *mod* que toma como parâmetros de entrada dois inteiros sem sinal em r1 e r2 e devolve em 2 registradores da sua escolha o quociente e o resto da divisão de r2 por r1. Sugestão: use as instruções *udiv* e *umull* (p. 85-86 do manual de instruções)

Esses 2 últimos exercicios fazem parte integrante das atividades obrigatórias 2b e 3.