



MC 404C - Arquitetura e Programação de microcontroladores 2º semestre de 2014



Profs. Célio Guimarães e Nelson Machado
Atualizado em 05/09/2014

Horário de atendimento: IC1 - sala 47: 2ª a 4ª 18:00 às 19:30hs

[Diagrama de blocos de uma CPU simples](#). Veja aqui o [ciclo de execução de uma instrução](#) usando o diagrama da CPU simples.

O [programa minimal](#) visto em aula e sua codificação em linguagem de máquina. [Diagrama de blocos simplificado do ARM](#)

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

Um resumo das [instruções do ARM Cortex M3](#)

Detalhamento das instruções do ARM [Cortex M3 instruction set](#)

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

Como escrever [programas estruturados em assembler](#)

Formato IEEE/ISO para números em [ponto flutuante \(32 bits\)](#)

[Arquitetura/Programação do microcontrolador ARM](#)

Atividades desafio:

A realização de uma dessas atividades dispensará o aluno das provas e das atividades subsequentes que não envolvam interrupções.

[Atividade desafio 1 \(opcional, individual\): o quebra-cabeça dos 12 pentaminós](#)

Condições para fazer o desafio serão negociadas caso a caso.

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

Condições para fazer o desafio serão negociadas caso a caso.

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](#)

Condições para fazer o desafio serão negociadas caso a caso.

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: 10/09/14 **Prazo susy:**

Atividade 2: [Conversão de valores binários de 32 bits para Ascii-hexadecimal](#)

Data de submissão: 10/09/14 **Prazo susy:**

Atividade 3: [Geração de números primos usando o "crivo de Eratóstenes"](#)

Data de submissão: 10/09/14 Prazo susy:

Exercícios

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

- 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 2.
- Escreva um programa que **em tempo de montagn** 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 *showcrivo* 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 liga o flag Z caso $r2 \bmod r1 = 0$; ou seja, ao executar as 2 instruções a seguir:

```

    b1 mod
    beq mult    @ desvia para mult se r1 é múltiplo de r2
    ....      @ executa esta instrução caso contrário
Sugestão: use as instruções udiv e umull

```

Esses 2 últimos exercícios farão parte integrante da Atividade obrigatória 3.