

EA869 – Introdução a Sistemas de Computação Digital

Lista de Exercícios Computacionais

Prof. Levy Boccato – 1º semestre de 2019 – Turma A

Data de entrega: 21/05/2019

1º exercício

Prepare um programa em linguagem *Assembly* para o ATmega328P que compute o n -ésimo valor da série de Fibonacci de maneira recursiva. Adote as seguintes convenções:

- O valor de n é inicialmente colocado no registrador r16.
- O programa chama uma subrotina FIB, a qual, de maneira recursiva, determina o valor desejado da série, devolvendo o resultado no registrador r17.
- Utilize a pilha para preservar o conteúdo dos registradores que serão utilizados pela subrotina.

Índice	0	1	2	3	4	5	6	7	8	9	10	...
Valor da série	0	1	1	2	3	5	8	13	21	34	55	...

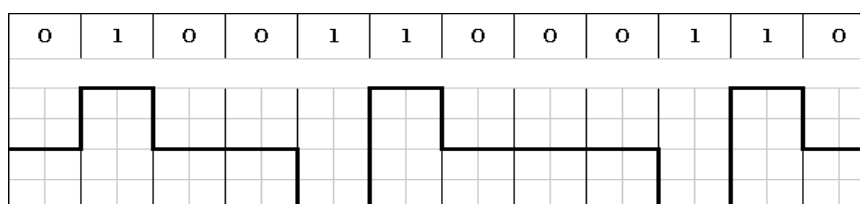
2º exercício

Uma etapa bastante importante na transmissão digital de dados corresponde à codificação da informação. O objetivo da codificação é aumentar a robustez da comunicação em relação a erros de detecção dos bits transmitidos, respeitando as restrições de potência e banda impostas pelo contexto do sistema projetado.

Uma técnica simples de codificação de linha (banda base), denominada AMI (*Alternated Mark Inversion*), representa cada bit de informação (b_i) por meio de um símbolo do alfabeto ternário $c_i \in \{-1, 0, 1\}$ de acordo com a seguinte regra:

$$c_i = \begin{cases} 0, & \text{caso } b_i = 0 \\ +1, & \text{caso } b_i = 1 \text{ e o último bit 1 foi codificado como } -1, \\ -1, & \text{caso } b_i = 1 \text{ e o último bit 1 foi codificado como } +1 \end{cases}$$

Exemplo:



Neste exercício, você deverá preparar um programa, usando o conjunto de instruções do ATmega328P, que gere a sequência de símbolos de acordo com a codificação AMI.

A sequência de bits de informação (entrada) está armazenada em um conjunto de N bytes da memória de programa a partir da posição rotulada como *vetor*, sendo que o primeiro *byte* nesta área de memória indica o valor de N . Para criar esta área de dados (e testar o seu programa), use o seguinte trecho de código:

```
.cseg
vetor: .db 3, 0x86, 0x73, 0xa4
```

Na memória de dados (SRAM), deverá ser gravada a sequência de símbolos codificados. Desta forma, a codificação de cada bit do vetor de entrada será um *byte* armazenado na memória de dados. Ao todo, portanto, serão gravados $8N$ bytes. Para isso, reserve uma porção da área de dados para a gravação da sequência codificada utilizando o seguinte trecho de código:

```
.dseg  
codigo: .byte P,
```

em que P deve ser substituído pelo valor correspondente a $8N$.

Exemplo de saída esperada:

- **Sequência original:**

0001 1000 1010 0000 1111 0100 0111 0010 0000 0100 0001 0011 1001 1101 1110 0101

- **Sequência codificada:**

0001 -1000 10-10 0000 1-11-1 0100 0-11-1 0010 0000 0-100 0001 00-11 -1001 -110-1 1-110 0-101

Avaliação:

Cada grupo deve entregar um pequeno relatório (no máximo, 2 páginas), indicando o raciocínio utilizado para resolver cada exercício. Além disso, os códigos fonte (pasta compactada de cada projeto) devem também ser encaminhados via e-mail.