Sistemas Digitais I
1ª Repetição Escrita de 2001/01

Leia atentamente o enunciado. Seja breve nas respostas, mas justifique-as convenientemente. **Por favor**, use letra legível ! Com os melhores desejos de bom trabalho nesta repetição ...

I

Os recentes acontecimentos mundiais, entre muitas outras coisas, fizeram-nos lembrar a importância da segurança nos sistemas de comunicações. Na Marinha, as comunicações podem ser feitas em "clara voz" (sem encriptação), ou "usando cifra". Neste último caso, as mensagens, antes de ser transmitidas, são "passadas pela máquina de cifra" (as célebres BID), e depois de recebidas no destinatário são passadas por outra máquina de cifra que as "desencripta".

As máquinas de cifra recebem uma sequência de bytes representando o texto da mensagem ou a codificação da voz. Fazem depois uma série de transformações a esses dados (normalmente função de uma dada "chave"), produzindo outra sequência de bytes que não é fácilmente entendível por quem não tenha uma máquina idêntica e a chave correcta. Essa nova sequência é então enviada ao transmissor.

Por hipótese, queremos começar a usar cifra (nem que seja "fraca") em todas as nossas comunicações, mas dados os cortes orçamentais não temos dinheiro para comprar BIDs. Por isso foi decidido usar os velhos SDK85 da Escola Naval como máquinas de cifra, a que chamaremos M.Á.S. (Máquina Automática de Sifra). Assim sendo, vamos ligar a saída de dados digitais do centro de mensagens à entrada SID do 8085, e a saída SOD do 8085 à entrada do transmissor.

Como os Kits SDK85 têm pouca memória, vai ser necessário acrescentar RAM extra. Os actuais SDK têm, como decerto se lembrará, uma ROM nos endereços 0000H-0FFFH, uma RAM nos endereços 2000H-20FFH, e ainda "dispositivos especiais" nos endereços 1800H-1FFFH.

I.1) Que tipo de memória sugere que se use para aumentar a capacidade dos Kits ? Identifique explicitamente o tamanho, tipo de tecnologia, e modo de operação a usar. Ao justificar a sua escolha compare-a com outra alternativa.

I.2) Uma breve consulta ao mercado mostra que há memórias RAM estáticas baratas com 32Kx8 bits, que irão ser usada nas M.Á.S.. para Faça um diagrama das ligações entre um integrado de memória destes e o SDK85, indicando **todas** as ligações necessárias, e indicando quais os endereços para onde fica mapeada esta nova memória.

A construção e manutenção de um sistema destes só é exequível se o subdividirmos em blocos e sub-rotinas independentes facilmente entendíveis. Alguém já escreveu uma rotina, a que daremos o "label" *LerByte*, que lê um byte do porto série (SID), e deixa esse byte no Acumulador. Escreveu também uma rotina *EscreveByte*, que envia o dado contido no Acumulador para o porto série (SOD).

I.3) Escreva uma rotina (usando mnemónicas) que recebe no par DE um endereço de memória onde deverá começar a guardar os dados, e depois vai lendo dados do porto série e guardando-os nesse endereço (e endereços consecutivos), até que seja encontrado entre os dados o byte 00H (que ainda deverá ser guardado em memória). Esta rotina deverá ter como label *LerBloco*.

I.4) Um dos tipos de encriptação mais simples (e menos seguros), é simplesmente trocar a ordem dos bits dentro de cada byte. Escreva uma rotina que encripte os dados trocando em cada byte os 4 bits mais significativos com os outros 4. A rotina deverá receber no Acumulador o número de bytes a encriptar, e em HL o endereço do primeiro. Os dados encriptados deverão ficar no local onde estavam os dados originais. Esta rotina deverá ter como label *TrocaNibble*.

I.5) Para que um sistema de encriptação seja minimamente seguro, é fundamental que esta não seja sempre feita da mesma maneira. Assim é normal haver uma "chave" que altera a maneira de encriptar os dados. Vamos então fazer uma rotina que roda os bits *n* posições para a direita. A rotina deverá receber no Acumulador o número de bits a rodar, e no par HL o endereço do primeiro. A rotina deverá terminar quando o byte a encriptar fôr 00H. Esta rotina deverá ter como label *RodaBits*.

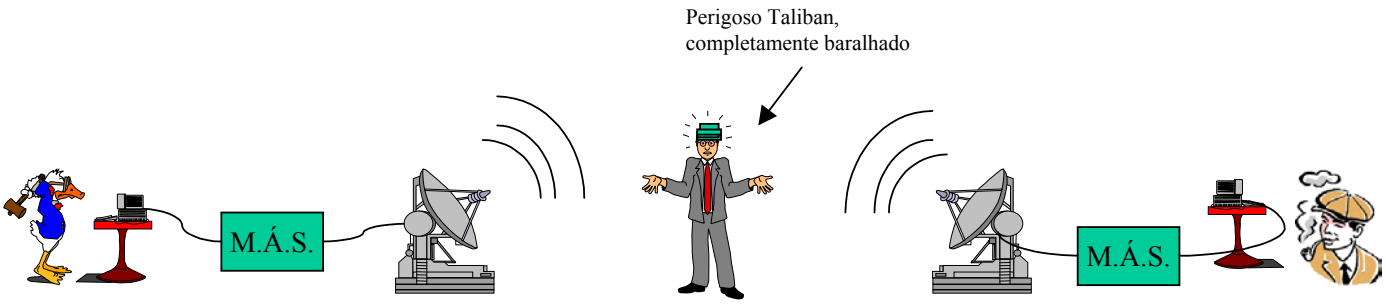
I.6) Escreva um programa que lê a entrada até encontrar o caracter com código 00H, encripta os dados recebidos rodando os bits 4 posições, e depois os envia pelo porto de saída.

II

Como alternativa a este sistema, podemos ter um sistema mais simples que recebe os bits à entrada, e escreve à saída 1 se o bit é igual ao que foi recebido anteriormente, e 0 em caso contrário. Pressupõe-se que no início, o sistema "faz de conta" que o bit anterior foi 1.

II.1) Desenhe o diagrama de estados deste sistema.
II.2) Implemente este sistema usando apenas uma ROM e um Registo.

Boa sorte !



P.S: Como aprenderá mais tarde, um sistema de cifra destes, embora tenha sido muito eficaz no tempo dos romanos e mesmo na idade média e renascença, é muito facilmente "crackado" por qualquer pessoa que saiba ler e escrever. Por isso, não perca as próximas repetições escritas, onde usaremos um microprocessador mais potente para fazer cifra muito mais eficaz….