UM CONVERSOR DE CÓDIGO

Por Luiz Amaral
PY1LL/PY4LC

Muitas vezes nos deparamos com um problema de converter códigos para resolvermos determinados problemas. Por exemplo, temos um código binário e necessitamos apresenta-lo em dígitos decimais. Nesse caso específico existem CI's projetados para tal fim. Mas muitas vezes os CI's comerciais não atendem às nossas necessidades especiais. Como um caso possível, tem-se um código binário representando um determinado número N e necessita-se uma saída com o código binário de N+2 (ou qualquer outra conversão que se deseje). Nesses casos, como proceder?

A conversão pode ser feita de modo dinâmico usando-se um microprocessador (um PIC, por exemplo) que, recebendo um código na entrada, gera o novo código em sua saída ou de modo estático, usando-se uma ROM. Vamos nos fixar nesse último tipo de conversão que é mais rápida por ser combinacional.

O conjunto de bits de endereçamento da ROM (pode ser uma EPROM, EEROM ou qualquer tipo que se deseje) é alimentado com o código a ser convertido e os bits de dados formam o novo código. É um processo muito versátil por ser bastante geral, isto é, a princípio pode ser usado como conversor entre quaisquer dois códigos.

Uma limitação prática é que, normalmente, os dados das ROM's comerciais são de, no máximo, 8 bits e isso limita o número de bits do código de saída quando se deseja usar apenas um CI. Pode-se, no entanto, usar mais de uma ROM com lógica adicional para se conseguir códigos de saída com mais de 8 bits.

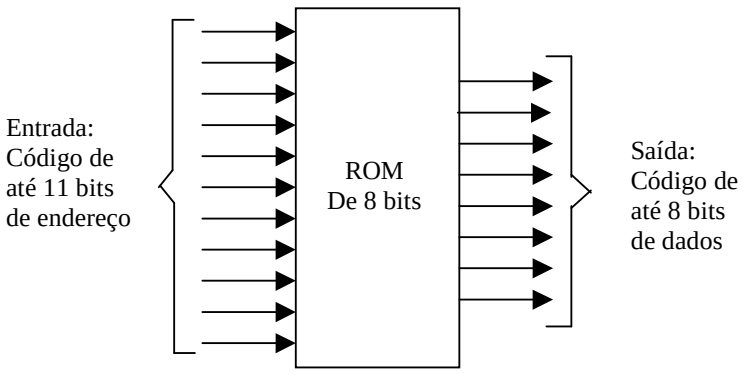

Figura 1

A Figura 1 mostra um caso de conversão usando-se apenas um CI de ROM.

Para o exemplo acima, poderia ser usada uma EPROM 2716, de 2K X 8, isto é, de 2048 endereços com 8 bits de dados.

Agora vamos falar sobre um caso prático real.

Eu possuía um rádio CB Midland 79-82 de 40 canais com AM, LSB e USB e o transformei em um equipamento de 6 m com AM, LSB, USB e CW.

Os detalhes da transformação estão fora do escopo deste artigo, mas dois pontos devem ser explicados.

O primeiro problema era que eu desejava que o dial (que originalmente era um disco de 40 posições de 1 a 40) mostrasse a freqüência (de 10 em 10 KHz) na banda de 6m, isto é, que o canal 15, por exemplo, significasse 50.015 KHz, o canal 32 significasse 50.032 KHz, etc. Como eu queria acessar a freqüência de 50.000 KHz, eu necessitava que existisse no dial o algarismo 0 (zero). Foi fácil: apaguei o algarismo 4 do canal 40 e o transformei em algarismo 0! (isto foi possível porque a chave não trava no canal mais alto, isto é, pode-se passar do último canal ao primeiro sem ter que voltar 39 posições atrás).

Aqui aparece o segundo problema: os canais da faixa de 27 MHz não são de 10 em 10 KHz um seguido do outro. Há pontos em que a distância entre dois canais sucessivos é de 20 KHz e há outros em que um canal de número menor corresponde a uma freqüência maior e isto destruiria minha indicação no dial. O que fazer?

Para que a canalização da faixa do cidadão seja correta, a chave de canais tem de ser mecanicamente bastante complexa. Sua saída corresponde justamente aos bits de entrada do CI do PLL do rádio. A solução foi simples: intercalei uma EPROM 2716 (poderia ter sido uma PROM que é menor fisicamente, mas eu não dispunha de nenhuma na ocasião) entre a chave de canais e o PLL de tal modo que os bits da chave correspondem aos bits de endereço da EPROM e os bits de dados da mesma correspondem justamente aos bits que, aplicados ao PLL, geram a freqüência marcada no dial. Devo lembrar aqui que, para a correta operação do PLL. Houve troca de um cristal do rádio e isso teve de ser levado em consideração na programação da EPROM.

Por Luiz Amaral
PY1LL/PY4LC

A chave de canais tinha 6 bits de saída por não serem os canais ordenados e, portanto, não se utilizariam os 11 bits de endereço da EPROM. Como os canais em 6m agora seriam ordenados, 6 bits seriam suficientes, ou seja, não seriam utilizados todos os bits de dados da EPROM (esta foi, portanto, superdimensionada, nesse caso). Como a EPROM deveria ser programada?

Suponhamos que, num determinado canal, a chave fornecesse o código binário 100101 e que o PLL necessitasse, para a correta apresentação da freqüência no dial, o código 110110. Assim, na posição de memória 00000100101 (11 bits de endereço da EPROM, com 6 bits menos significativos ligados à chave de canais e ligando-se à terra os 5 bits mais significativos, ou seja, fazendo-os iguais a 0), deve-se programar o código binário XX110110 (8 bits de dados da EPROM, com os 6 bits menos significativos ligados ao PLL e abandonando-se os 2 bits mais significativos. X significa irrelevante) ou 1, porque são bits abandonados).

O resultado foi excelente! Após alguns anos de uso, esse rádio foi presenteado ao Santos, PY1YM.

O leitor pode facilmente concluir que esse método de conversão de códigos é praticamente ilimitado, ou seja, somente limitado por nossa imaginação.

Por Luiz Amaral
PY1LL/PY4LC