A CODE CONVERTER

By Luiz Amaral
PY1LL/AC2BR

Many times we get a challenge of code conversion to solve some problems. For example, we have a binary code and we need to present it in decimal digits. In this specific case there are IC's designed for the conversion. But many times the commercial IC's don't satisfy our special needs. As a possible case we have a binary code representing certain number N and we need an output with the binary code N+2 (or any other conversion we want). In this case, how to act?

A conversion can be done in a dynamic mode using a microprocessor (a PIC, for example) that, receiving a code at its input, generates the new code at its output or in a static mode, using a ROM.

Let's focus in the latter that is quicker and combinational.

The address bit set of the ROM (may be an EPROM, EEROM or any other type we want) is fed with the code to be converted and the data bit set make the new code. It is a very versatile process for being rather general, that is, it may be used as converter between any pair of codes.

A practical limitation is that the commercial ROM data are normally an 8 bit set and this limits the number of bits of the output code when we want to use only one IC. We can, however, use more than one ROM eventually with additional logic to get an output with more than 8 bit.

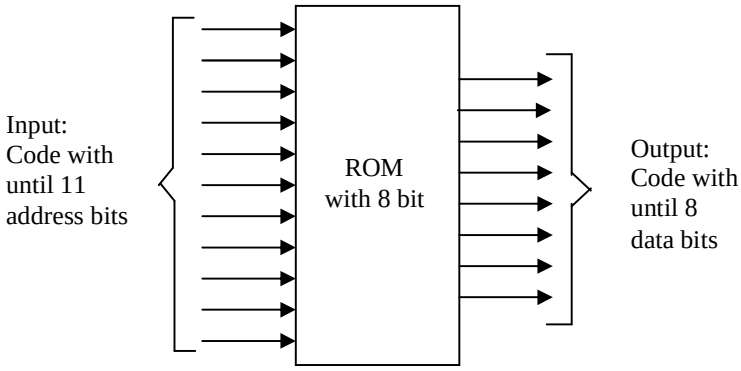

Figure 1

Figure 1 shows a conversion case using only one ROM IC.

For the above example, we could use a 2K X 8 2716 EPROM, that is, 2048 addresses with 8 data bits.

Let's talk about a practical case.

I had a Midland 79-82 40 channel CB radio, with AM, LSB and USB and I transformed it into a 6m equipment with AM, LSB, USB and CW.

Agora vamos falar sobre um caso prático real.

The details of the transformation are out of the aim of this article, but two points must be explained.

The first problem is that I wanted that the dial (originally it was a 40 position disc, from 1 to 40) showed the frequency (10 in 10 kHz) in the 6m band, that is, the channel 15, for example, meant 50,015kHz, the channel 32 meant 50,032kHz, etc. As I wanted to access the frequency of 50,000kHz, I needed a display with the number 0 (zero). It was easy: I deleted the figure 4 of the channel 40 and it was transformed into 0! (This was possible because the frequency switch doesn't latch on the highest channel, that is, we can pass from the last channel to the first one with no need to switch back 39 positions).

Here a second problem arises: the adjacent channels of the 27MHz band are not always separated by 10kHz. There are points with the separation of 20kHz and other with a smaller number channel corresponding to a higher frequency and this would destroy my dial indication. What to do?

For the correct CB channeling, the channel switch is mechanically very complex. Its output just corresponds to the input bits of the radio PLL IC.

The solution was simple: I inserted a 2716 EPROM (it might be a ROM smaller physically but I had none) between the channel switch and the PLL in such a way that the switch bits were the address of the EPROM and the output bits of the latter were the bits that applied to the PLL input, generated the frequency showed on the dial. I must remember here that I had to change one radio crystal for the correct operation of the radio PLL and this had to be taken into account for the EPROM programming.

The channel switch had 6 output bits because they were not ordered channels and, so, we didn't need to use all the 11 address bits of the EPROM. As the channels were ordered, 6 bit were enough and not all the 8 data bits of the EPROM were used (the memory was really oversized). How the EPROM would have to be programmed?

Let's suppose that, in certain channel, the switch delivered the binary code 100101 and the PLL needed the code 110110 for correct frequency display on the dial. So, in the position 00000100101 (11 EPROM's

address bits with the 6 most significant bits connected to the channel switch and the 5 less significant bits connected to ground or equal to 0), we have to program the binary code XX110110 (8 EPROM's data bits with the 6 less significant bits connected to the PLL input and the 2 most significant bits abandoned. X means irrelevant or 1as they are abandoned bits).

The results were excellent! After some years the radio was offered as a gift to Santos, PY1YM, now SK.

The reader can easily conclude that this code conversion method is practically unlimited, that is, only limited by the designer imagination.