# Motorcycle Gear Shift using z64 processor

by Antonio Bernardini

## 1 Project

### 1.1 Requirements

A z64 processor manages the display that indicates the `GEAR` currently in gear on a motorcycle.

The processor receives an interruption request from the `GEARBOX` peripheral every time the driver changes gear. Since the gearbox is sequential, the processor must recover the information if from the `GEARBOX` device the driver is shifting up or down, reading the value of an appropriate register of interface.

The `GEAR` device is equipped with a seven-segment display. The values that can be shown are the following:

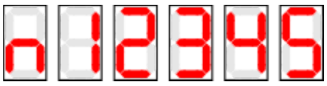
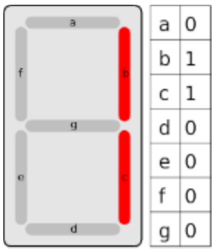


**Figure 1.** Gears that can be represented on the seven-segment display

where the first symbol ($n$) indicates that the motorcycle is in neutral. The `GEAR` device uses 7-bit words for determine which segments of the LED should be lit. Each bit represents the status of one of the segments ($0 = \text{off}, 1 = \text{on}$), according to the following scheme, where " $a$ " is the least significant bit:



| | |
|---|---|
| a | 0 |
| b | 1 |
| c | 1 |
| d | 0 |
| e | 0 |
| f | 0 |
| g | 0 |

**Figure 2.** Coding to be used to turn on the LED diodes of the seven-segment display

To design:

- The interfaces of the `GEARBOX` and `GEAR` devices

- All code necessary for the system to function

### 1.2 Implementation

#### 1.2.1 Hardware

The `GEARBOX` peripheral is represented as a classic *asynchronous daisy chain* input device:

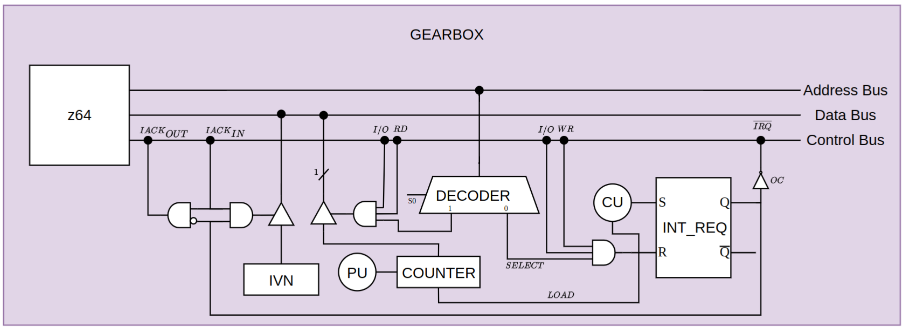


**Figure 3.** The `GEARBOX` peripheral

The `GEAR` peripheral is represented as a classic *synchronous* output device:

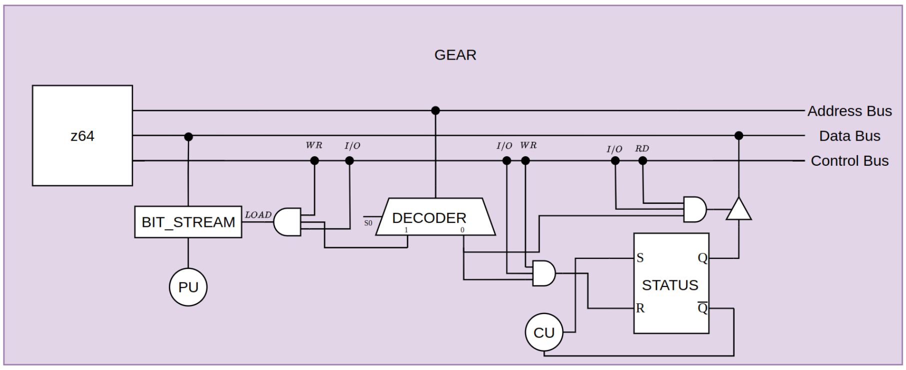


**Figure 4.** The `GEAR` peripheral

### 1.2.2 Firmware

So, a possible *firmware implementation* can be found here.