

# Traffic Lights System using z64 processor

BY ANTONIO BERNARDINI

## 1 Project

### 1.1 Requirements

A z64 processor controls the traffic light system of a pedestrian intersection. When the **BUTTON** is pressed, the processor programs the traffic lights of the cars to go from green to yellow and then from yellow to red. At the same time the pedestrian **TRAFFIC LIGHT** changes from red to green and then to yellow and then to red. To manage the transitions, the processor uses a programmable **TIMER** peripheral: it receives the number of seconds after which it will have to send an interrupt to the processor.

The transition times are defined as follows:

- After pressing the button, 10 seconds pass before switching to yellow for cars and a further 3 seconds to switch to red
- Green for pedestrians remains for 30 seconds, followed by 45 seconds of yellow

Even without pressing the button, the **TRAFFIC LIGHT** carries out a pedestrian crossing cycle every 5 minutes.

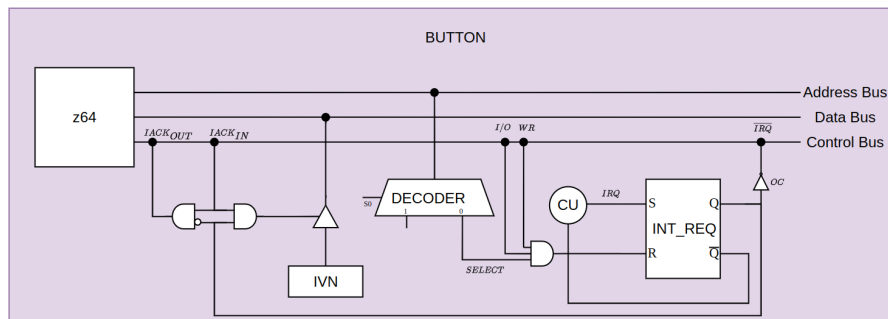
To design:

- The interfaces of the **TRAFFIC LIGHT**, **BUTTON**, **TIMER** peripherals.
- Device drivers and management software.

### 1.2 Implementation

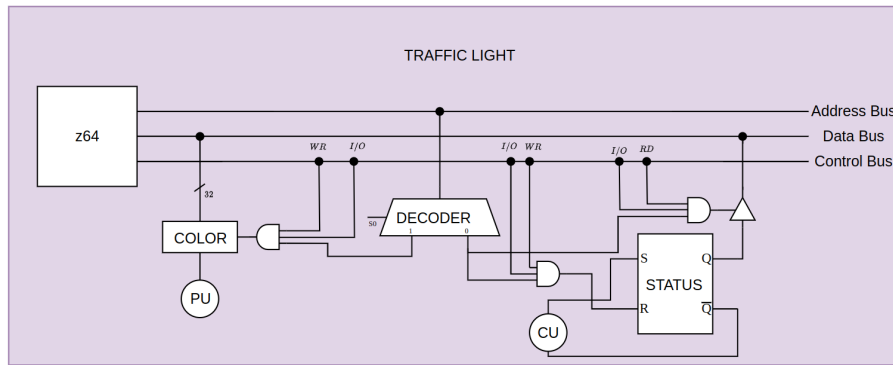
#### 1.2.1 Hardware

The **BUTTON** peripheral is represented as a classic *asynchronous daisy chain* device:



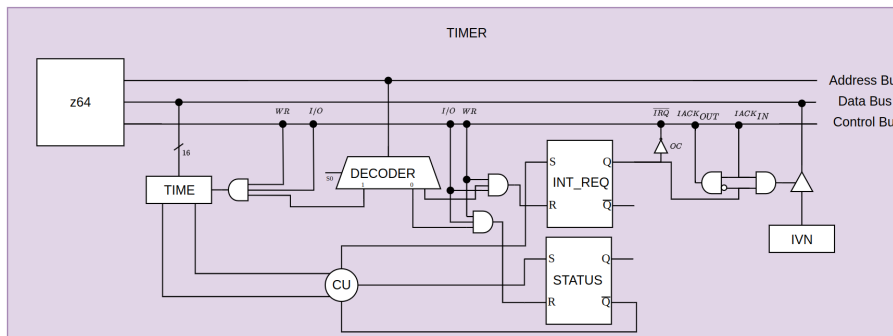
**Figure 1.** The **BUTTON** peripheral

The **TRAFFIC LIGHT** peripheral is represented as an output device that is used at the firmware level by implementing *busy waiting*:



**Figure 2.** The TRAFFIC LIGHT peripheral

Finally, the TIMER peripheral is represented as a device that operates in *mixed mode*, i.e. both *synchronously* and *asynchronously*:



**Figure 3.** The TIMER peripheral

### 1.2.2 Firmware

So, a possible *firmware implementation* can be found [here](#).