Room monitoring example using z64 processor

by Antonio Bernardini

1 Project

1.1 Requirements

A room is monitored by four temperature sensors (therefore a `SENSOR` peripheral must be set up), which are driven by a z64 processor. The latter constantly checks that the average temperature value detected in the room is within the range $[T_{\min}, T_{\max}]$.

If the temperature does not fall within this range, the microprocessor sends an alarm signal to a specific peripheral (`ALARM`). The alarm signal used is the value 1 coded with 8 bits. If the temperature returns to the range $[T_{\min}, T_{\max}]$, the CPU transmits the value 0 to the peripheral.

The sensors return the measured temperature as a 16-bit integer, using tenths of a degree Celsius as the unit of measurement.

Write Assembly code to control temperature sensors and of the `ALARM` peripheral, using the interruption mechanism vectorized.

1.2 Implementation

1.2.1 Hardware

Below are the project choices:

- The temperature measurements of the four sensors are stored within a vector of four elements
- When the system starts, the four measures are forced to the center of the interval $[T_{\min}, T_{\max}]$
- The `SENSOR` is an input device consisting of a read-only register that contains the measured value
- If the temperature is negative, the sensor still returns the value 0
- `ALARM` is an output device, which activates/deactivates a siren flashing. A flip/flop connected to the least significant bit of the data bus turns the alarm on/off when set/reset.
- `ALARM` is a *passive peripheral*: it does not have a CU

The `ALARM` peripheral is represented as following:

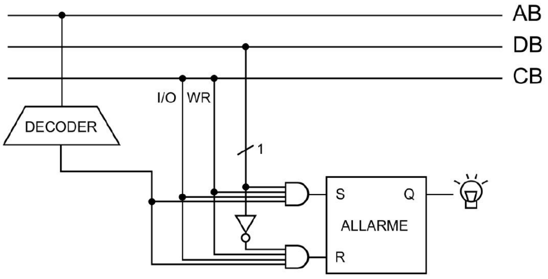

Figure 1. The `ALARM` peripheral

The `SENSOR` peripheral is represented as following:

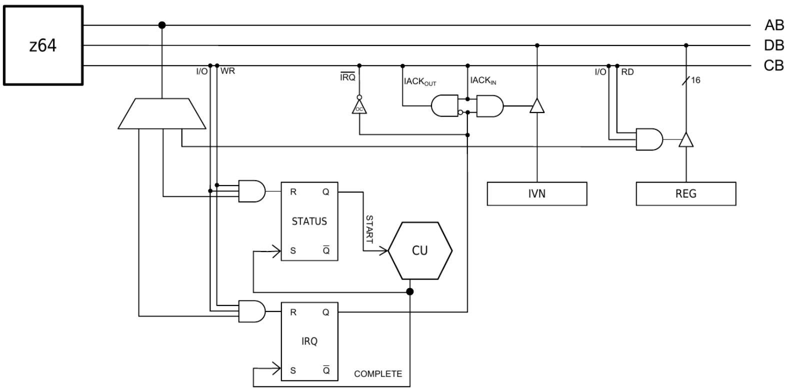

Figure 2. The `SENSOR` peripheral

1.2.2 Firmware

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