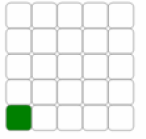
**Signal Strength Indicator: An Application Developed Using 8-bit RISC Processor**

An 8-bit non-pipelined RISC processor was designed in the course: CSE206 Computer Organization and Architecture. The instruction set consisted of all basic RISC instructions, including the categories: data transfer, logical, arithmetic, and branching. The processor was designed in the Logisim tool, using combinational and sequential digital-logic building blocks.

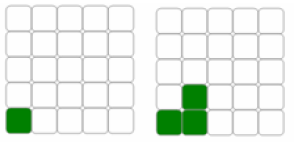
**Application development:**

Signal strength indicator bars are very common in mobile phones. We developed an assembly language program (based on the 8-RISC processor and its instruction set), to detect the signal strength (assumed to be available on an 8-bit input port of the processor, and then depending on the following specifications, it will display a certain sequence of signal strength bars, on 5x5 LED matrix.

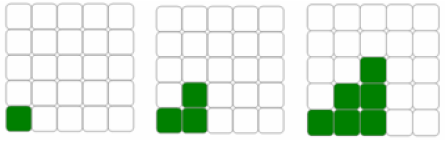
If 1 ≤ Signal Strength ≤ 20, the LED-pixels will display the following images (on the 5x5 LEDs) in a loop.



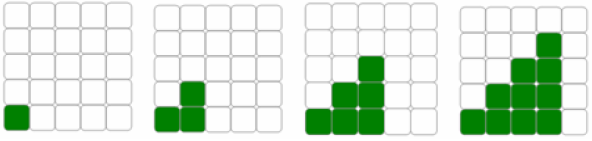
If 21 ≤ Signal Strength ≤ 40, the LED-pixels will display the following images (on the 5x5 LEDs) in a loop.



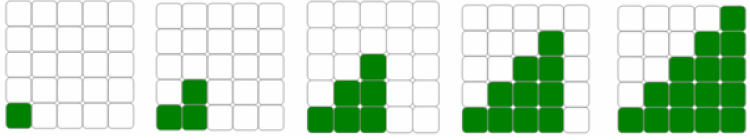
If 41 ≤ Signal Strength ≤ 60, the LED-pixels will display the following images (on the 5x5 LEDs) in a loop.



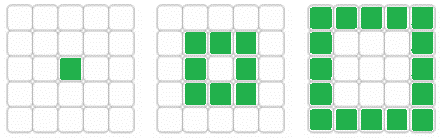
If 61 ≤ Signal Strength ≤ 80, the LED-pixels will display the following images (on the 5x5 LEDs) in a loop.



If 81 ≤ Signal Strength ≤ 100, the LED-pixels will display the following images (on the 5x5 LEDs) in a loop.

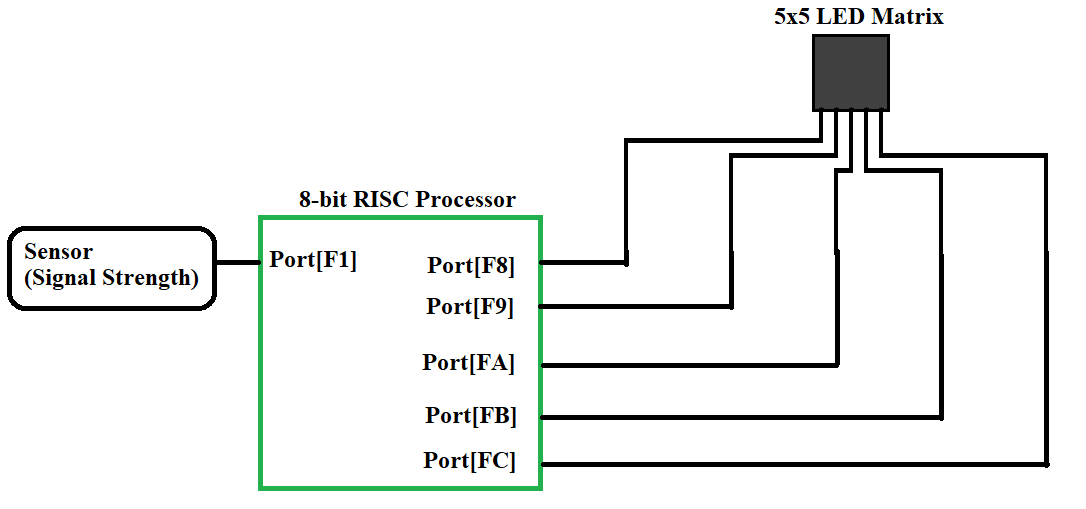


When Signal Strength = 0, then the LED-pixels will display the following images (on the 5x5 LEDs) in a loop, to indicate the signal searching process.



**Connections/Interface with the RISC Processor:**

The sensor that detects signal-strength is assumed to be connected to the 8-bit input Port[F1] of the processor. The 5-bit busses that control the 5x5 LED Matrix are connected to 8-bit output Port[F8] through Port[FC]. There is some additional logic connected to Port[FD], not shown in the figure below.



**Assembly Language Program / Hex Code:**

The following Hex code (program), implements this application, and is loaded in the instruction memory of the processor.

