Discussion:

The “box” component is a generic binary signal repeater, which can be customized to support any signal pattern consisting of binary 1/0 values by way of generic map. It has clk, reset, and enable inputs. Reset initializes the box to the beginning of the signal pattern, and on each rising edge of clk, it steps to the next unit of the signal pattern. If enable is set to 0, no output is produced, and the box holds its output to 0. The signal is generated by connecting a sequencer (which decrements its way through states, and underflows to the beginning of the sequence once it hits zero) to a decoder (which can interpret the states generated by the sequencer and produce either a 1 or a zero, depending on the current state and the given signal pattern).

The distress box incorporates two instances of morse code signal generators; one for “SOS” and one for “DEVO”. Each box is defined in terms of generic values, so the same component can be used for both generators. When instantiating the two boxes, all we have to do is pass the data width, length of our signal, and a std_logic_vector containing our desired signal via generic map. These generators have built-in support for enable/ disable and asynchronous reset functionality, and require a clock signal to function. These signals are directly mapped to the enable, reset, and clk inputs of the distress box.

The distress box also implements additional functionality: a selection switch by way of which the user may choose whether they want to generate a SOS signal or a DEVO signal, which is accomplished by a simple multiplexer. The final waveform is sent to the code_out output of the distress box.