NIM+ Development Project

Varun Ursekar and Mohit Pandey
Boston University, Boston, MA 02215

(Dated: December 15, 2018)

In this report we test the latency time, decision time, minimum input pulse width, minimum input pulse height, and high-frequency performance of a ZedBoard programmed to detect two-channel signal coincidence. We test the FPGA using both square-pulse generators and a pair of muon-detecting scintillators and compare the results with a standard NIM coincidence module. We find a latency time of 25 ± 5 ns, decision time of 6 ± 3 s, minimum width of ≤ 5 ns, minimum height between +1.5 V and +2.4 V, and essentially optimal high-rate performance.

The Nuclear Instrumentation Module standard has defined the specifications for electronics used in experimental particle and nuclear physics since the 1960s [1]. It has enabled the set-ups in these experiments to be completely modular: a broken or outdated module can easily be substituted by another. Support and spare parts for NIM-based hardware, however, is becoming increasingly rare. Moreover, settings on NIM devices, such as those shown in Fig. 1, must be programmed physically and individually, a drawback in large projects which might need hundreds of these devices. In view of these shortcomings, many research groups are looking to replicate the functionality of NIM modules using programmable FPGA technology, an effort dubbed NIM+. The functions of these devices are encoded in software, which is infinitely replicable and much more convenient to modify and fix than a physical device. In this report we test some specifications of one particular candidate for use in particle experiments: the ZedBoard. In particular we test its capacity to perform a two-channel coincidence operation and benchmark it against a a standard NIM coincidence module.

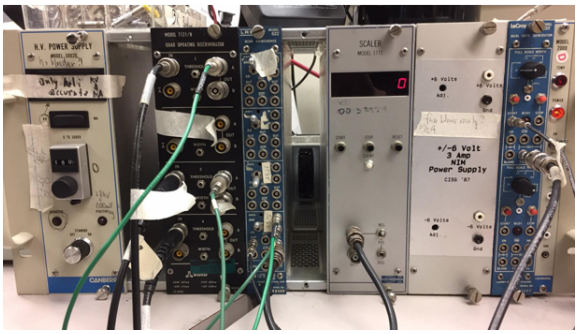

FIG. 1: NIM Crate. A standard NIM crate consisting of a metal chassis encasing several replaceable, device modules.

The ZedBoard, shown in Fig. 2, is a field-programmable gate array (FPGA): an integrated circuit that can be programmed to perform arbitrary logical operations. Two examples would be the OR and AND operations. Given two input channels, an OR operation will yield a signal if either an input from channel 1 or channel 2 is registered by the device. An AND operation will yield a signal only if the device registers inputs from channel 1 and channel 2 simultaneously. These operations can easily be encoded into the FPGA using hardware description language or a high-level synthesis language such as Vivado [2]. Once programmed, the FPGA will act as a logic board operating at a clock-speed of about 40 MHz and can, therefore, replicate many of the functions of standard NIM modules. The LeCroy coincidence module, for example, essentially implements an AND operation: it will output a signal if two inputs are registered simultaneously.

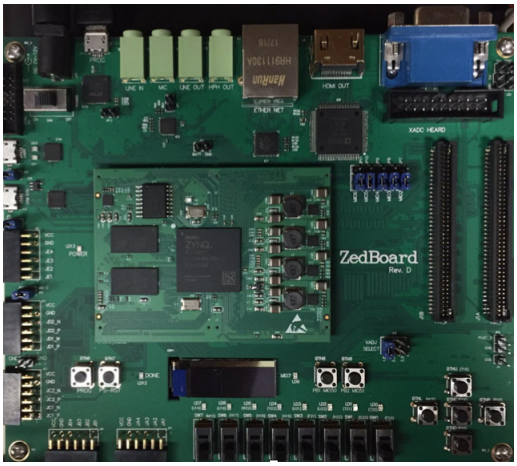

FIG. 2: ZedBoard. The particular device used in this experiment. The FPGA is the black square shaped chip at the centre of the ZedBoard.

We use two sources of pulses to test the AND/OR functionality of the ZedBoard: a square-pulse generator and a pair of parallel BisMSB scintillators that detect cosmic muons. The pulse generator can generate -0.7 V NIM signals of minimum width 5 ns at a maximum frequency on the order of megahertz. The scintillators generate noisy signals that are converted to NIM signals using a LeCroy discriminator module. All NIM signals must be converted to a +3.0 V TTL signal before being sent to the Zedboard. This is easily done by using a NIM to TTL level shifter. Figure 3 shows a schematic of our set-up.

In addition to the pulse sources, discriminator, and level shifter, we have delay generators that split each square pulse into two pulses separated by an adjustable time delay. This will be used to test the ability of the FPGA to detect coincidence. Figure 3 also shows the LeCroy coincidence module against which we will benchmark the NIM+.

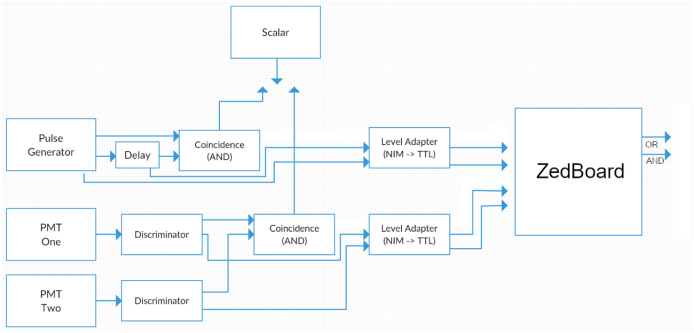

FIG. 3: Schematic of set-up [3]. The outputs from the pulse sources are modified before being sent to the ZedBoard, which will then process them and generate its own output.

Statistics that would be crucial in determining the ZedBoard's performance in real experiments might include: the latency time, AND decision time, minimum input pulse height and width, and its high-rate performance. Figure 4 shows a typical OR output signal (channel 3 in the figure) when two delayed square pulses (channels 1 and 2) are inputted to the FPGA. With the oscilloscope grid spacing set to 25 ns, the figure suggests

$$\text{NIM+ Latency Time} = 25 \pm 5 \text{ ns}. \tag{1}$$

Our error estimate 5 ns is the minimum possible pulse width that the pulse generator can create. Similarly, the NIM coincidence module has

$$\text{NIM Latency Time} = 15 \pm 5 \text{ ns}, \tag{2}$$

which suggests that the NIM device processes signals faster than the FPGA. Since the differences are on the nanosecond scale, we believe part of the time lag is due to the length of the cables used to transmit signals between the pulse source, FPGA, and oscilloscope. For 1 m of two-way wire, this would add 5-10 ns to the latency time. So the FPGA's latency is comparable to that of the NIM module. Next, we define the decision time to be the minimum overlap between two pulses in order for the FPGA to register an AND signal. We find this to be

$$\text{NIM+ Decision Time} = 6 \pm 3 \text{ s} \tag{3}$$

The LeCroy module has approximately the same decision time. We test the input pulse height requirement by attenuating the TTL input signal. An 80% attenuation still gives an output, while 50% attenuation does not. Therefore,

$$+1.5 \text{ V} \leq \text{Min. Input Height} \leq +2.5 \text{ V}. \tag{4}$$

Given more time, we could have found a more precise estimate. We find that the smallest input width

$$\text{Min. Input Width} \leq 5 \text{ ns}. \tag{5}$$

This is an upper bound and not an equality, because our pulse generator could not generate pulses of width ≤ 5 ns. Finally, the high-rate performance is essentially optimal: we find that provided the input pulses do not overlap, the FPGA will generate non-overlapping outputs. We expect that the device will only get confused when the inputs themselves are not temporally separated.

FIG. 4: OR signal. Channels 1 and 2 show time-delayed square pulses from the pulse generator. Channel 3 is the OR output from the FPGA. We expect a separate OR for each signal since they do not overlap.

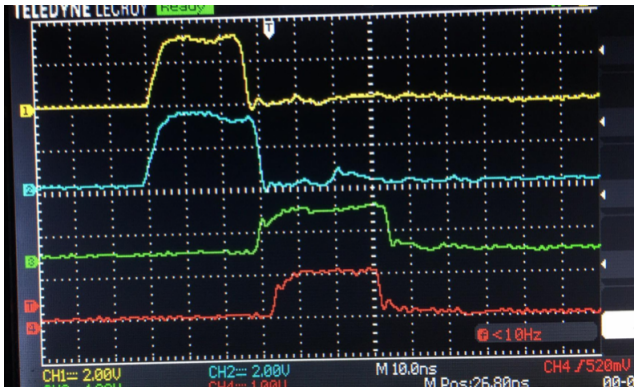

FIG. 5: AND signal. Channel 4 shows an AND output resulting from the overlap of the two square pulses in channels 1 and 2.

There are two logical continuations of the work done here. One is to obtain more accurate estimates of the presented statistics. In particular, the minimum input pulse width should be checked using a pulse generator with nanosecond precision and the minimum input pulse height should be checked by continuously attenuating the

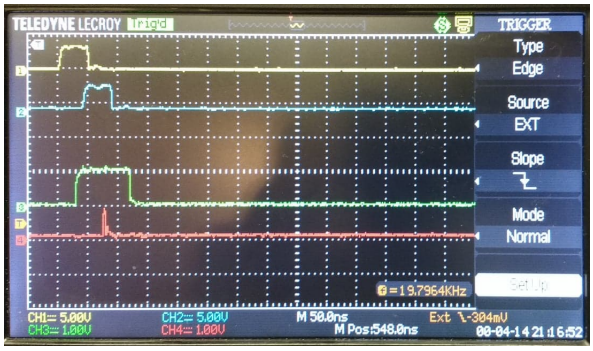

FIG. 6: OR/AND signal using scintillator input. Channel 3 is an OR signal from the FPGA. Channel 4 shows the AND signal resulting from the overlap of two scintillator signals. Notice that the AND signal is narrow, because it is sent only when the inputs overlap and here they barely do.

TTL input until no output signal is seen. The second task would be to implement a clocked version of the AND and OR operations. Recall that both the AND and OR signals are continuous – with an input signal of width x, the FPGA will in its current form generate an output of width x as well. This should be contrasted with the LeCroy coincidence module, which generates a pulse of fixed width if there is coincidence within a bracketed period of time. Implementing a clocked version of the AND operation will be essential to overcoming the time barrier imposed by the processing speed of the FPGA.

[1] W.R. Leo, Techniques for Nuclear and Particle Physics Experiments, 257-61 (1994).
[2] Xilinx Website, http://zedboard.org
[3] C. Cosby, Review of NIM+ Project at BU CMS Testbench.