Characterizing the Time Sensitivity of a Coincidence Detector

Tayari Coleman & Eric Ayars
California State University, Chico
eayars@csuchico.edu

The Problem

Before using a coincidence detector for research work, it's important to characterize the response of the detector. We need to know...

- What's the limit on frequency? How fast can pulses arrive at this device before it starts losing track of how many pulses there were?
- How much time can there be between two incoming pulses before the coincidence detector decides that they are not coincident?

The second one is really important. The coincidence detector we're using was designed and built locally, and we weren't sure it would work. Our quantum optics experiment requires that we be able to reject pulses that happen more than 40 nanoseconds apart.

Just for comparison: a nanosecond is the time it takes light to go one foot; so we needed to create two precise electrical pulses separated by the time it takes light to cross an average-sized classroom.

The Solution, part 1

Finding the maximum pulse rate appeared to be pretty easy: we just had to send pulses to the coincidence counter at a known frequency and see how many it counted in a known time. For example, if we sent the device a frequency of 42 kHz and asked it to count for 1 second, it should have seen 42,000 pulses. This worked.

It worked so well that the problem was finding the *maximum* count rate! Our best function generator maxed out at 10 MHz, and the coincidence counter was still working fine at that rate. Since we needed to be able to count at up to 10 kHz, we decided that 1000x better than necessary was probably good enough and we gave up on finding the actual maximum.

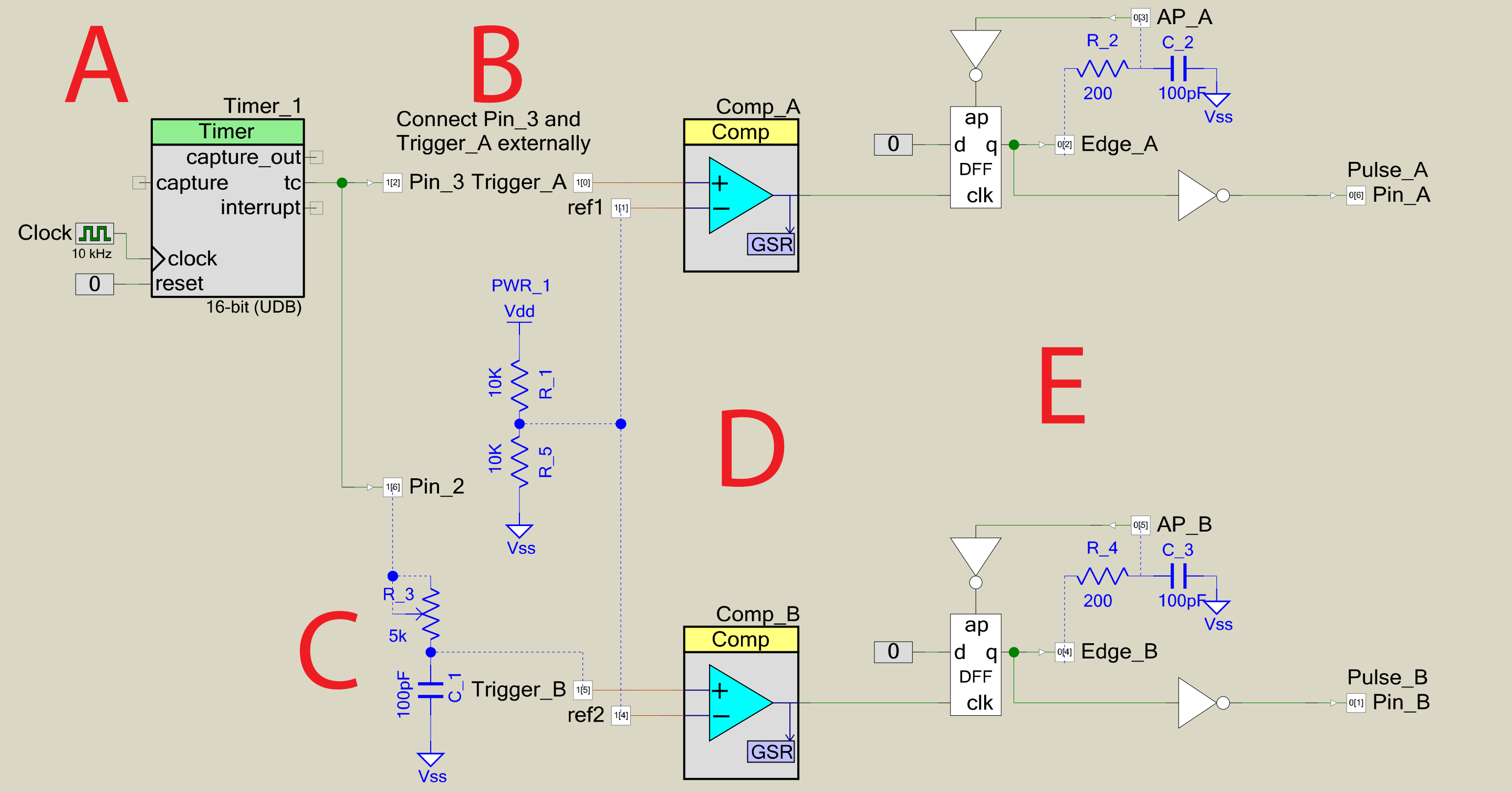
The Solution, part 2

We decided the best way to generate our test pulses would be to build a device that generated one pulse, then used an RC delay circuit to create a second pulse at a later time. By varying R in the RC delay, we could create the second pulse at a controlled time.

We could measure the delay using an oscilloscope, and carefully adjust the delay until the coincidence detector started recording coincidences.

How we did it

We used a Cypress Programmable System on Chip (PSoC) to generate the pair of pulses. The circuit is shown below, with annotations.



A) A Counter/Timer generates 1ms pulses at a rate of 1 kHz.

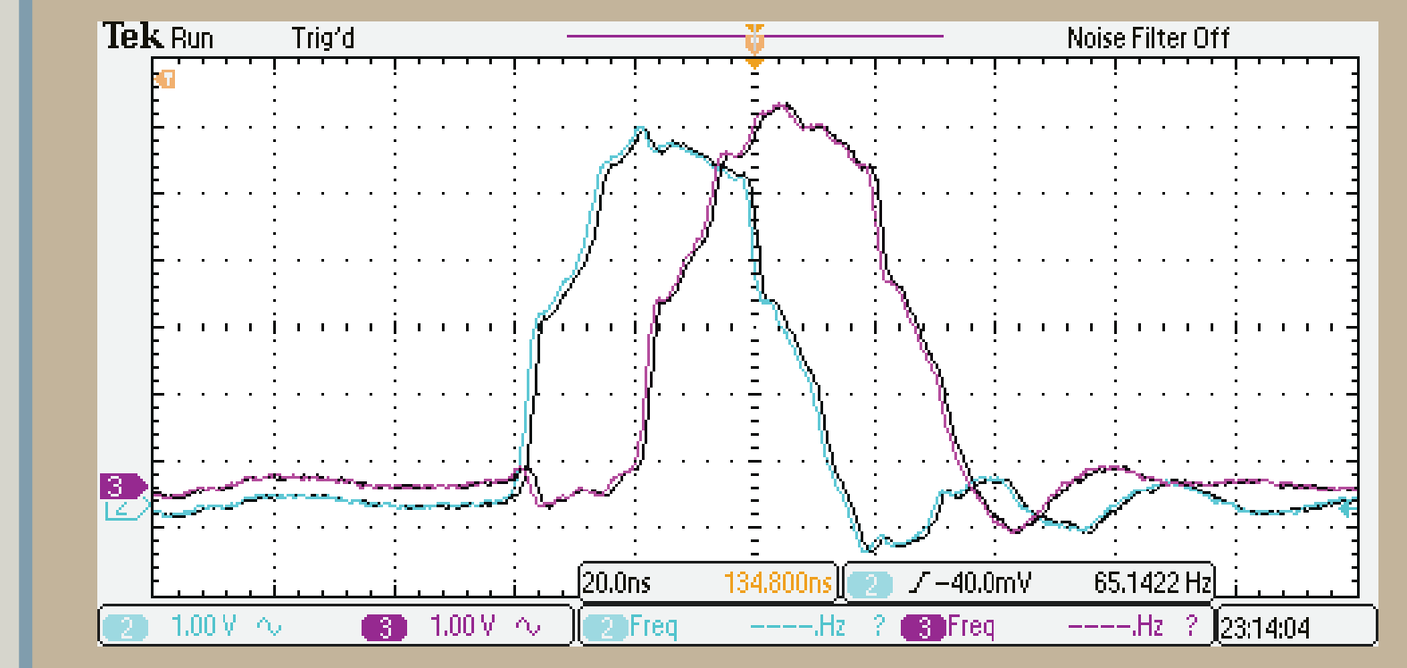
B) The pulse is split: part goes directly to a comparator (Comp_A) which immediately sends the pulse along.

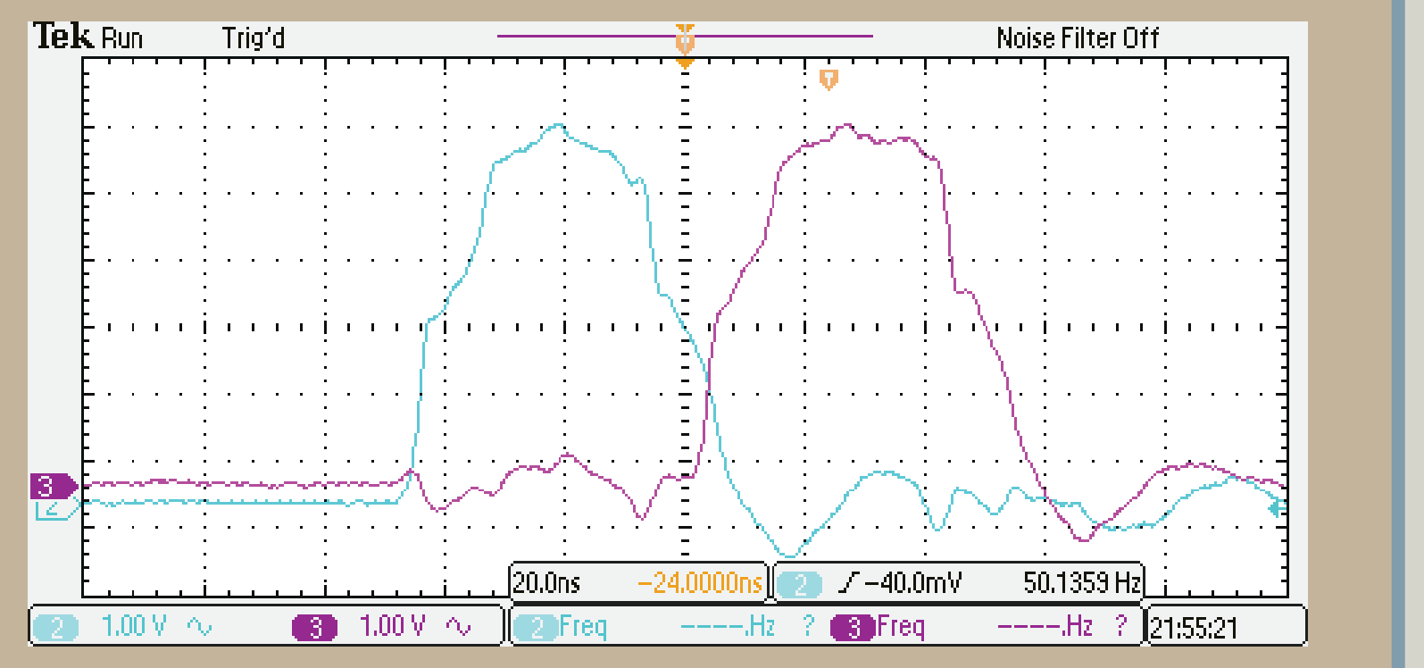
C) The pulse also begins charging a small capacitor through a variable resistor.

D) When the capacitor charge reaches 0.5Vdd, the voltage triggers comparator Comp_B, which sends the delayed pulse along. (Both original and delayed pulses are sent through comparators to equalize any comparator delay.)

E) Both original and delayed pulses are sent through an edge-detecting circuit, and from there sent out to the Coincidence Detector being tested.

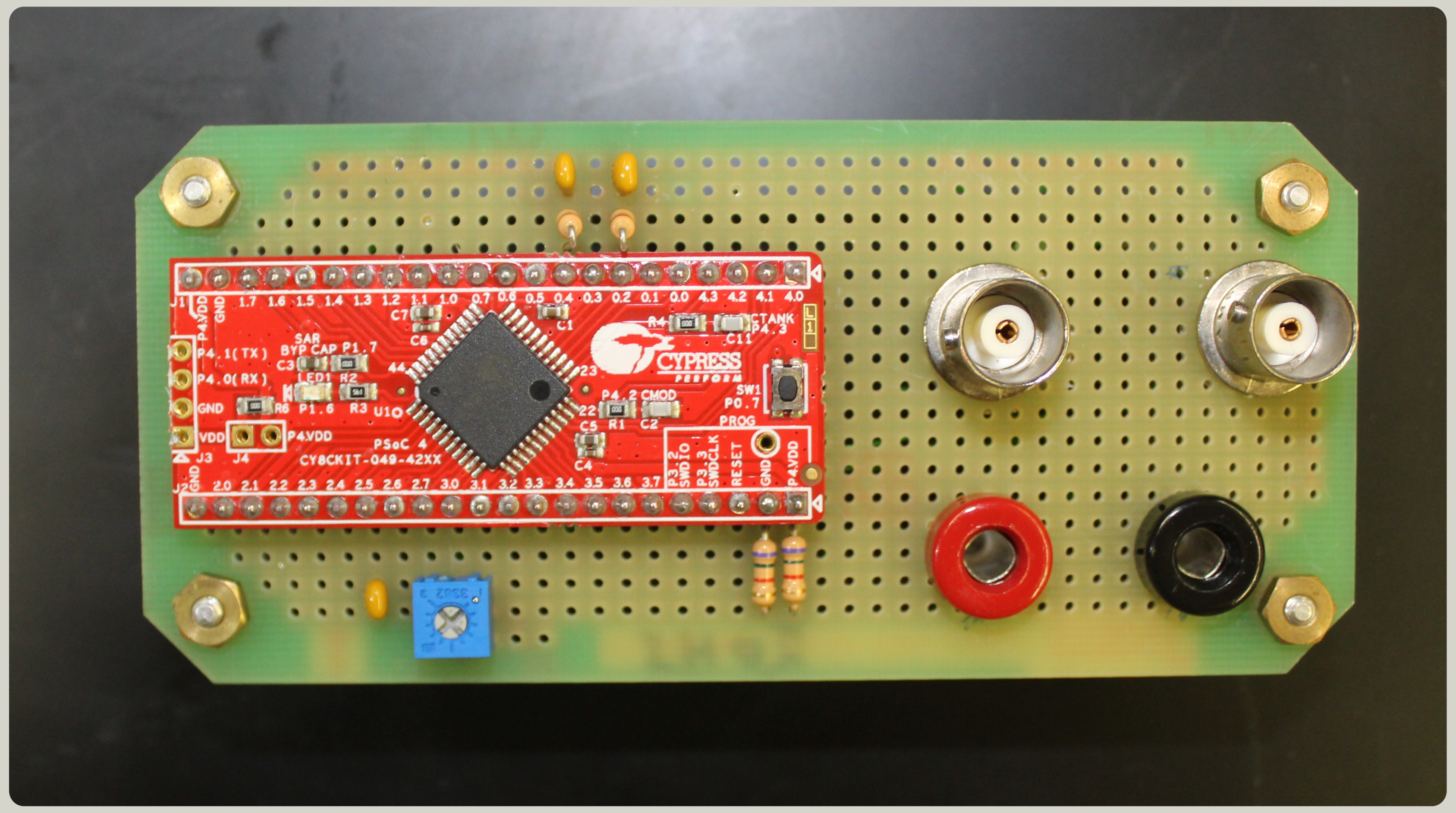
Success!





It worked quite well! We could generate the pulses, control their time delay, and use them to check the coincidence detector resolution.

We found that the coincidence detector rejected coincidences at times greater than about 25ns, which was with our necessary design spec.



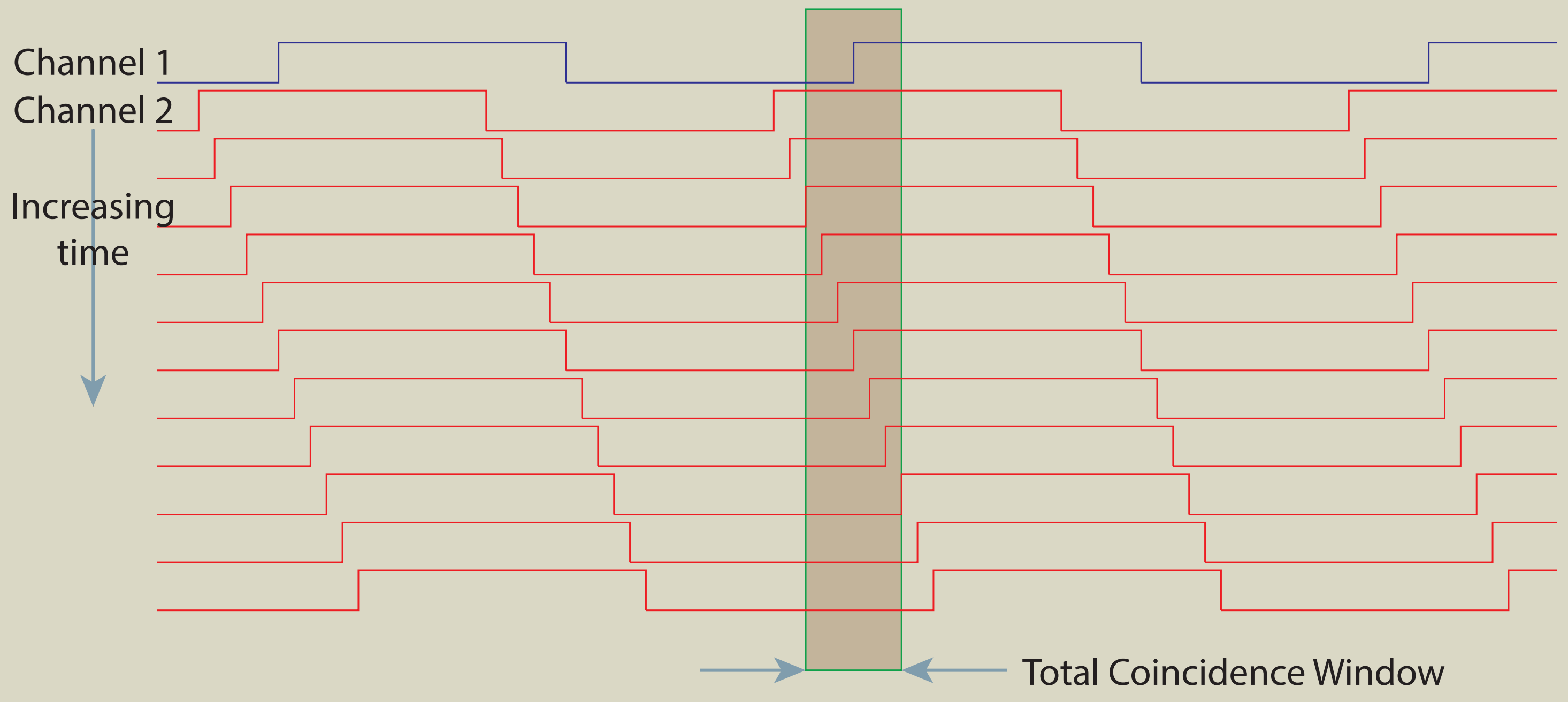
And here's another way to test it...

We can also test the time resolution by using beats between channels on a 2-channel function generator.

- Set channel 1 to a frequency of 100.00000 kHz and connect it to one channel of the coincidence detector.
- Set channel 2 to a frequency of 99.99990 kHz and connect it to another channel of the coincidence detector.
- The period of channel 2 is now 10 picoseconds longer than the period of channel 1, and...
- The two will overlap exactly once every 10 s.

As the two signals "pass" each other, they effectively generate pairs of pulses which step closer (and then farther away) in time by 10ps for each pair. The number of coincidences counted, multiplied by 10ps per coincidence, gives the total time range over which two pulses are close together enough to be considered coincident.

We measured between 4900 and 5090 coincidences per overlap, for a total coincidence window of 49-51 ns. This window includes all possible overlaps, both channel-1 first and channel-2 first, so we divide it in half to find the maximum time difference between the two pulses, which is between 24.5 and 25.5 ns. This result is consistent with results from our PSoC measurement.



For this conceptual example, the coincidence detector observes 5 coincidences. If the pulses on channel 2 are each 10ps longer than the pulses on channel 1, then this would indicate a total coincidence window width of 50 ps, and a time sensitivity of half that, or 25ps. We actually observed 5,000 coincidences, so our sensitivity is 25ns.