Lab 2: A simple Trigger Exercise

Introduction

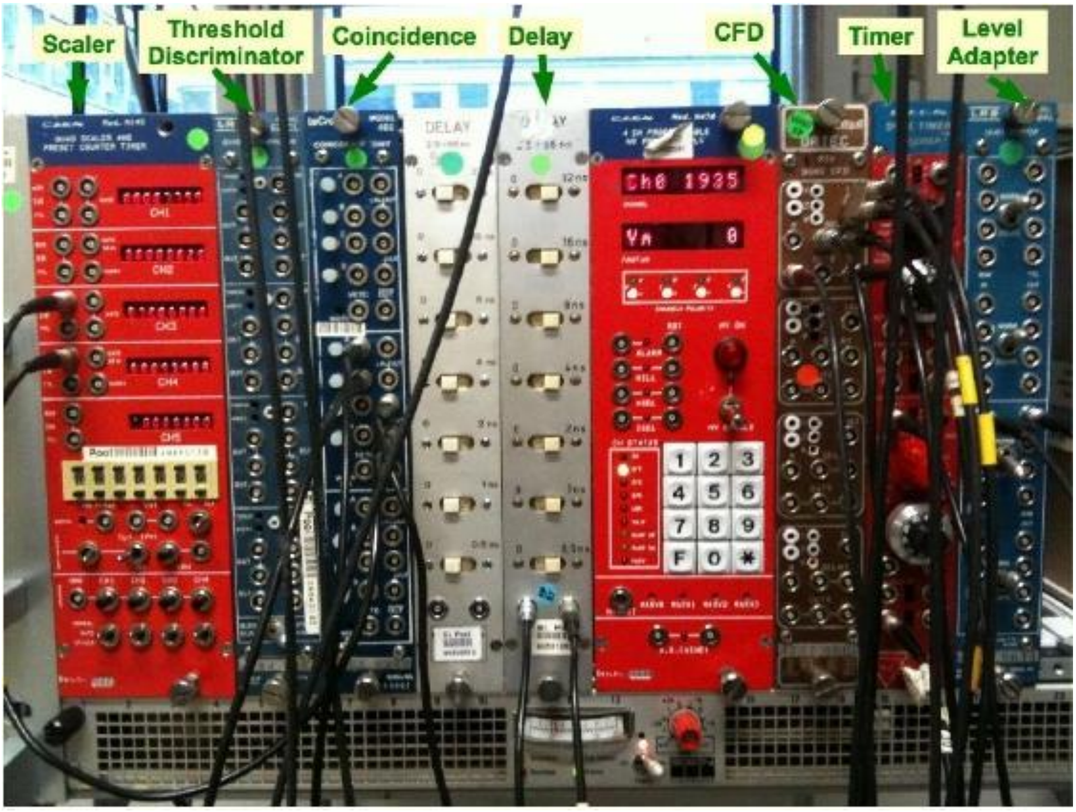

Figure 1: NIM modules.

This is a basic exercise based on the trigger lecture. It introduces all the elements and concepts needed in exercise 3 and 4. The available NIM modules are shown in Fig.1. The exercise is composed of 4 parts. At each step, look at the corresponding schema and follow the instructions.

A trigger is given by the transition of a signal from the logical 0 to 1. Before setting up any trigger system, you must have decided the levels corresponding to these logical levels and all the components of the system need to be correctly configured.

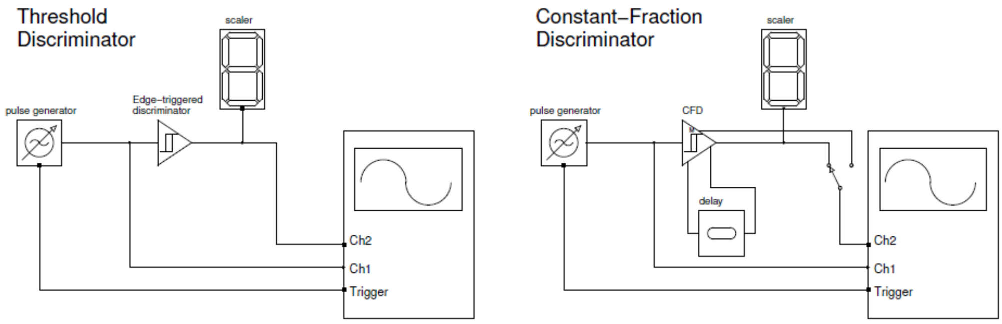

Figure 2: Scheme of threshold and constant fraction discriminators.

Part 1a: Threshold Discriminator

The Signal Generator is pre-configured to provide a triangular pulse with a period of 300 µs. Look at the signal (Channel Output) with the oscilloscope (CH1), using the Trigger Output of the generator as oscilloscope trigger (EXT). The Trigger Output is TTL signal.

What do you expect?

Why do we use it?

Now try to characterize the signal:

Leading edge time:	
Trailing edge time:	
Width:	

Using the LEMO cables, try to implement the schema shown in the left part of Fig. 2, i.e.:

- Split the generator output signal: connect the two parts to the input of the **Threshold Discriminator** and to the oscilloscope.
- Connect one output signal of the discriminator to the scaler module and a second output to the oscilloscope (CH2).

We have set-up a simple trigger system: you have a digital answer based on the amplitude of a signal. Reproduce the oscilloscope display shown in Fig. 3 and observe the amplitude of both the signal and its corresponding trigger.

Can you modify their amplitude?

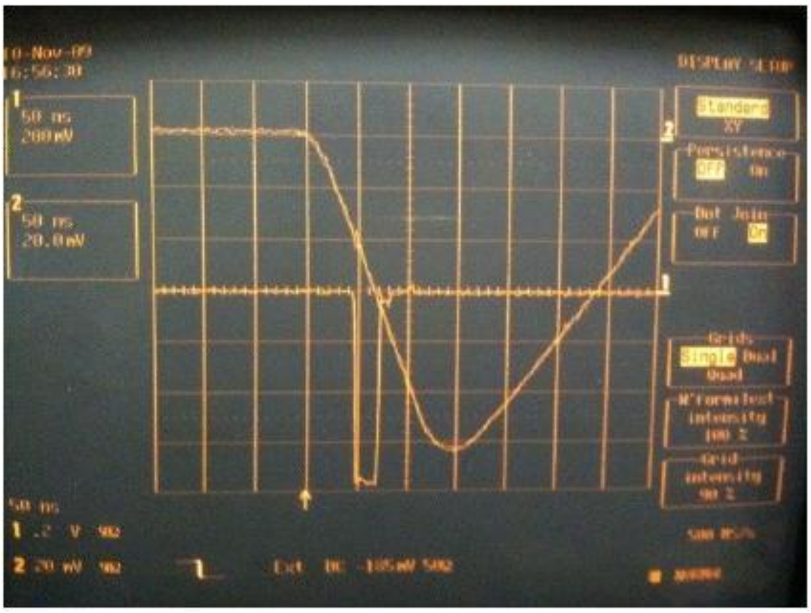

Figure 3: Input signal and threshold discriminator output.

The threshold set on the discriminator can be measured with a Voltmeter (x10 output) and changed with a screwdriver. Change the threshold value: observe the behavior of the discriminated signal on the scope and its rate on the scaler.

Can you relate them to the threshold values?

In real experiments, how is the best threshold value found?

Part 1b: Threshold Discriminator, the jitter

Using the above set-up, set the discriminator threshold to 60 mV and change the amplitude of the input signal.

What is the effect on the discriminated signal?

How does it affect a timing measurement?

Measure the discriminated signal delay with respect to the reference as a function of the amplitude of the input signal (-100, -150, -200, -250 mV) and fill up Table 1 with your numbers.

Input signal amplitude (mV)	Threshold D (ns)	CFD (ns)
100		
150		
200		
250		

Table 1: Measured delays on the discriminated signal with respect to reference.

Part 2: Constant Fraction Discriminator (CFD)

Using the above set-up, set the discriminator threshold to 60 mV and change the amplitude of the input signal.

Now use the Constant Fraction Discriminator to make a trigger from the generator signal and implement the layout shown in the right diagram of Fig. 2.

Using the Voltmeter and the screwdriver, set these CFD parameters:

- threshold (T): 60 mV Measure with Voltmeter (x10 output)
- walk (Z): 2 mV Measure with Voltmeter
- delay (D): 80 ns Set with delay module + 2x10ns cables

Connect the CFD monitor output (M) to the scope CH2 and reproduce Fig. 4.

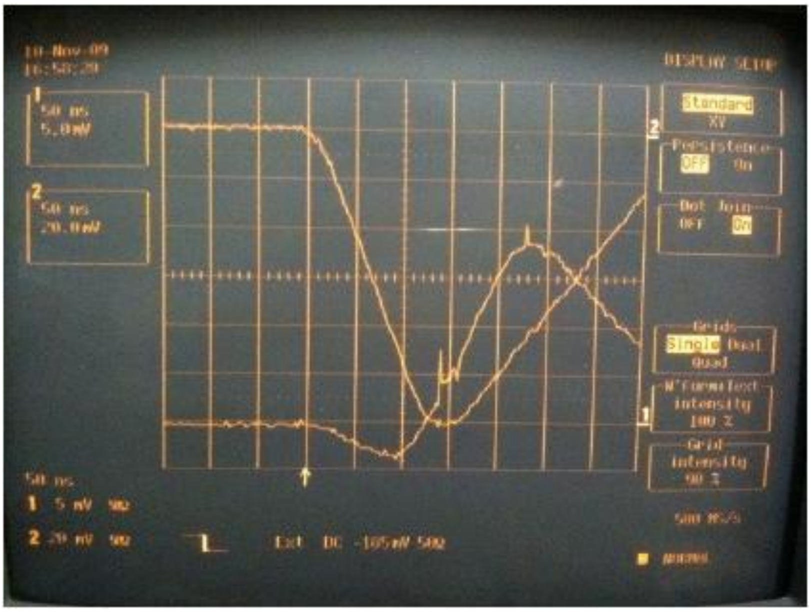

Figure 4: Input signal and CFD monitor output.

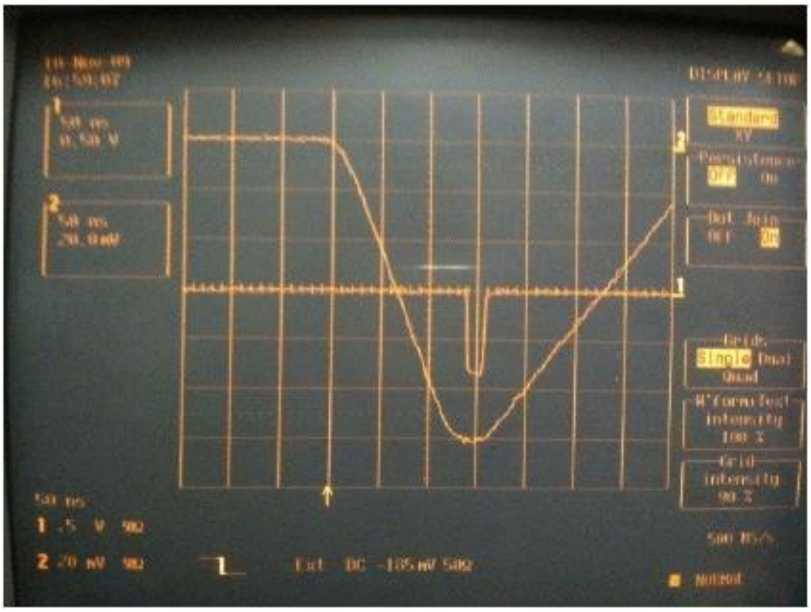

Figure 5: Input signal and CFD output.

Can you recognize the CFD technique?

What is the effect of varying the value of the delay D?

Now connect the CFD output to the scope (CH2) and change the amplitude of the input signal.

What happens to the output of the discriminator?

Measure the discriminated signal delay with respect to the reference as a function of the amplitude of the input signal (-100, -150, -200, -250 mV). Fill up Table 1 with your numbers. Compare the results with the previous measurements.

Can you see the advantage?

Can you make the CFD behave like a normal threshold discriminator?

Which configuration parameter has to be modified?

Part 3: Making a timing coincidence

We now try to simulate the coincidence of two different trigger signals, in a simplified way. For that, use an additional output of the signal generator, which is configured to generate a triangular pulse similar to the first one. Use a standard threshold discriminator unit (attached to ch. 2 of the generator) and a CFD (ch. 1) to discriminate both signals, as described in Fig.6.

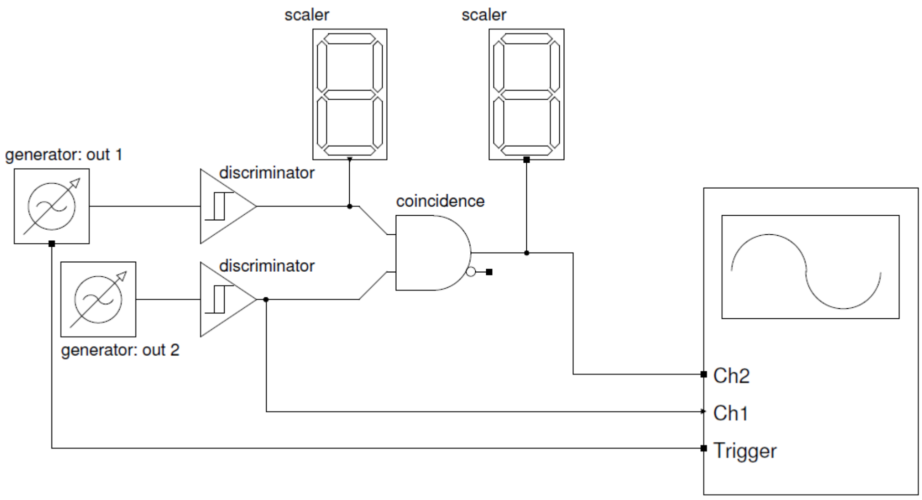

Figure 6: Coincidence layout.

We now have two independent trigger signals with similar characteristics. Look at them in the scope.

Which parameters are important when making a coincidence?

Use one unit of the **Coincidence Module**, which is able to generate the logical AND of its input signals. The module has two outputs: OUT and LIN-OUT.

Can you guess the timing behaviour of the AND output?

When are you expecting the AND output to rise?

The Scaler Module is a simple and useful tool in a trigger system: it allows you to simply count the triggers and verify if your system is behaving correctly. Use the scaler to measure the counting rate of your coincidence and try to answer these questions:

Can you count any trigger? How can you recover the coincidence rate?

After your adjustments, what is the width of the coincidence signal?

Can you explain the different behavior of the OUT and the LIN-OUT signals?

Which is better to use in a real trigger system?

How can you preserve good trigger efficiency if one of the signals has a large jitter?

Which is the drawback?

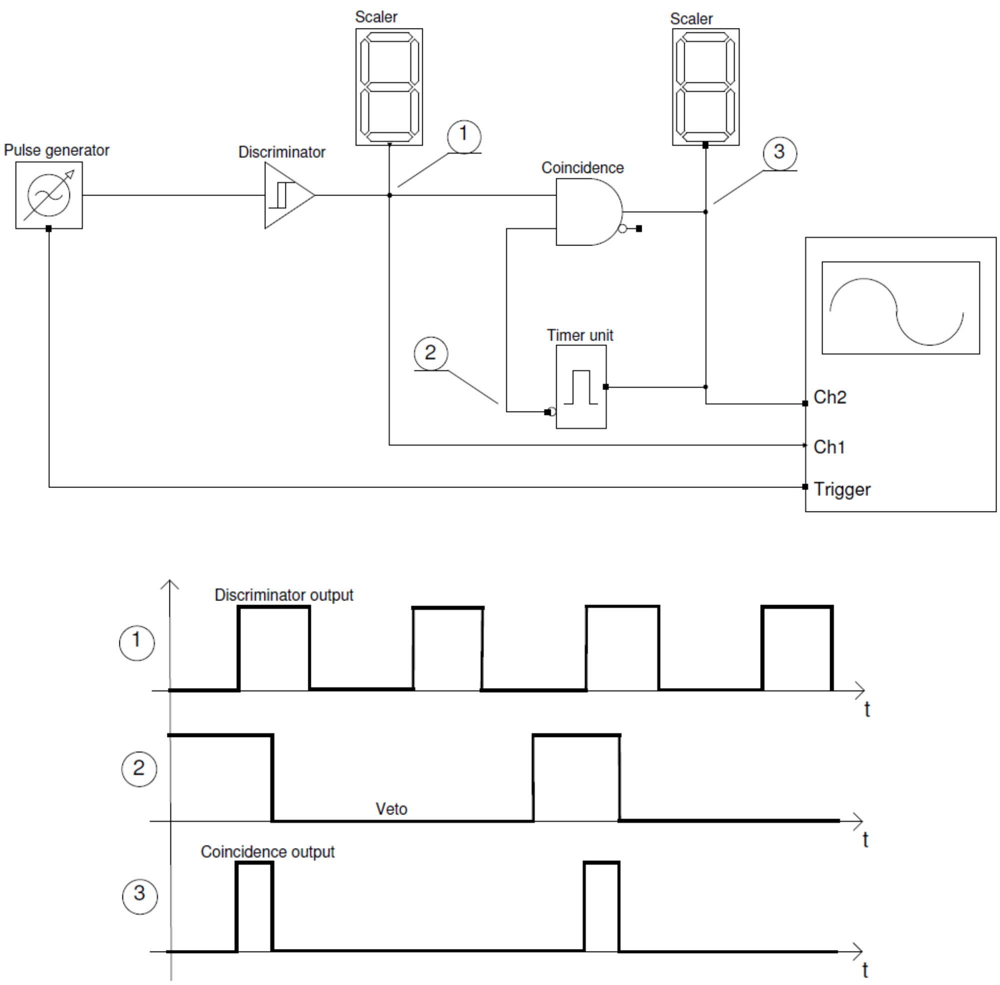

Figure 7: Top: busy logic schema with readout processing time simulated via a dual timer module. Bottom: time diagram of signals at the discriminator output (1), after the veto (2) and at the coincidence output (3).

Part 4: Trigger veto and dead-time

A busy logic can be implemented using the coincidence module and a Dual-Timer Module which simulates a readout system with a fixed processing time (readout dead-time). Configure one stage of a dual timer module to generate signals with 10 ms width. Then implement the busy logic in a second stage of the coincidence unit as shown in Fig. 7

- one input of the coincidence unit is the trigger signal;
- to simulate the start of the readout, and so the trigger ACCEPT signal sent to the readout system, use the output of the busy coincidence to drive the timer module (START);
- use the output of the timer as the VETO of the busy coincidence: this is the BUSY signal sent back to the trigger system;
- connect the trigger signals before and after the busy logic to the scaler and check the correct logic

You can easily make a rate measurement configuring the Scaler to work with a time gate of 1s with a GT+CLR configuration. Compare the trigger ACCEPT rate and the readout rate (after the BUSY) on the scalers.

How do they relate to the timer module setting?

Can you reproduce the numbers using the LIN-OUT of the coincidence unit?

Alternatively you can make an AND between the trigger and the output of the timer (with inverted logic) and not as a veto.

Where is the difference in the logic? What risk are we taking?

Can you explain the behaviors observed disabling either one or the other input of the coincidence unit?

Appendix: the Constant Fraction Discriminator

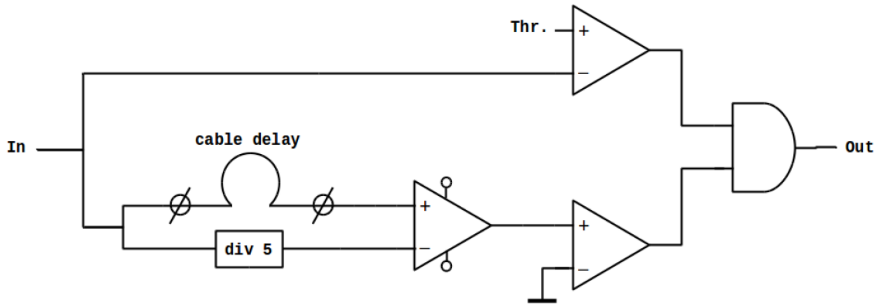

Figure 8: CFD function diagram.

The CFD functional diagram is shown in fig. 8. The input signal is split in two different discrimination branches, whose results are then merged by the final AND gate. The top branch is a standard threshold discriminator, where the input signal is compared against a (configurable) threshold Thr.

The bottom branch instead implements the constant fraction technique. Technically, the input signal is split: one copy is delayed, while the other is attenuated by a factor of 5. The two copies are then subtracted and the final result is compared with a threshold of (close to) zero. In fact, the zero-crossing time of the resulting signal in nearly independent from the input signal leading edge gradient (i.e. the source of time jitter in a standard threshold discriminator).

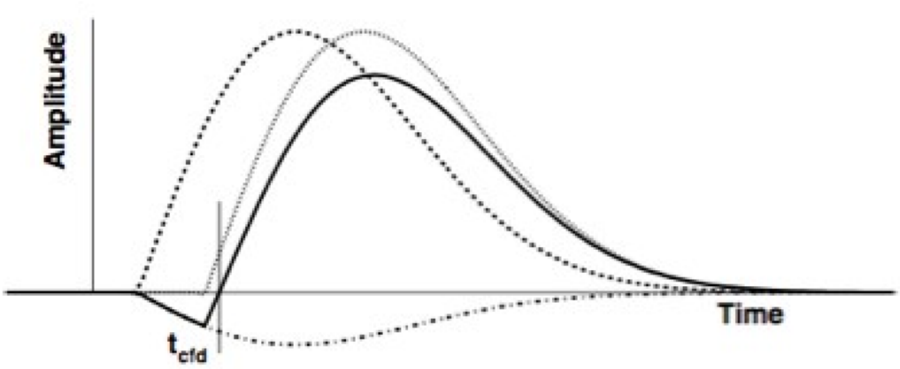

Figure 9: CFD function diagram.

Fig. 9 shows in detail the signals in the bottom branch of the CFD. The input pulse (dashed curve) is delayed (dotted) and added to an attenuated inverted pulse (dash-dot) yielding a bipolar pulse (solid curve). The output of the bottom branch fires when the bipolar pulse changes polarity which is indicated by time t_{cfd}. From a practical point of view, a small threshold, as close as possible, is actually used in the final comparator of the bottom branch. This is needed to avoid fake signals possibly caused by noise. Such a small threshold is normally called walk (Z).

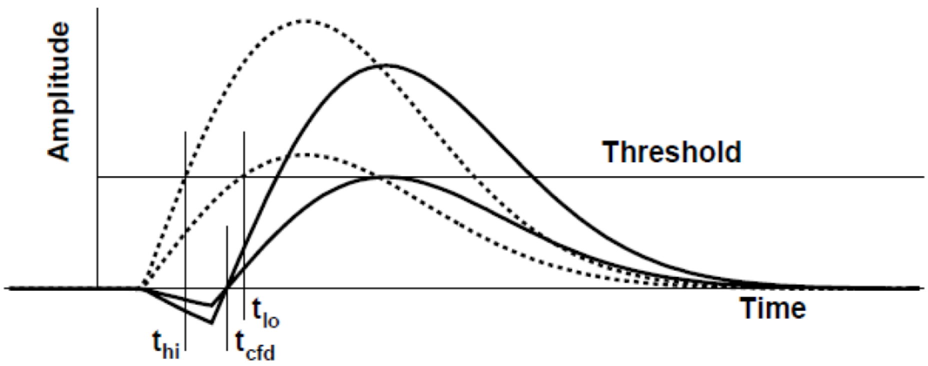

Figure 10: CFD function diagram.

In order to complete the CFD description, the merging of the top and bottom branch signals has to be considered, with the help of fig. 10. In the top branch, the threshold discriminator fires at time t_{hi}, that depends on pulse leading edge characteristics. The bottom branch instead fires at a time t_{cfd}, as discussed above, which is almost constant, due to the delay introduced in the bottom branch, normally $t_{cfd} > t_{hi}$. Therefore, the overall CFD, defined as the signal generated by the final AND gate, will fires at t_{cfd}, achieving both our requirements:

- only select signal above a given amplitude Thr;
- provide an output trigger whose timing is independent from input signal amplitude.

As can be seen in the above figure, the CFD operating principle is not retained for all the possible combinations of configured delay, threshold and input signal amplitude. As the top branch timing depends on the signal amplitude, a small enough signal can make it fire at a time $t_{lo} > t_{cfd}$. In this case the CFD will behave like a normal threshold discriminator, as the output AND gate will be driven by t_{lo}.