**基于FPGA的多光子符合计数模块**

David Branning\*a, Mark Beckb

aDept. of Physics, Trinity College, 300 Summit St., Hartford, CT 06106; bDept. of Physics, Whitman College, Walla Walla WA, 99362

# 摘要

我们提出了一个用于量子光学实验的多通道符合计数模块。 该电路最多可以接收四个TTL脉冲输入并进行所有通道的符合计数，用户选择的符合时间窗口可精确到12ns。 该模块可以精确计数八组多通道符合，输入速率高达84 MHz。 由于其成本低，体积小，多个模块可以很容易地组合在一起，计算N个输入之间的任意M阶重合。

**关键词:** 符合计数，相关测量，多光子

# 1. 介绍

符合计数是在不同检测器上同时检测两个或更多个粒子。 这种技术在实验物理中被广泛使用，在量子光学中起着特别重要的作用。 光子的符合计数是探索和/或利用相关光源非经典特征的重要工具。 许多这样的实验只需要一组两次重合测量，而对于其他实验则需要计算多个检测器之间的多光子符合。1-3

从历史上看，最常见的符合计数方法是使用时间 - 幅度转换器（TAC），每个TAC都有给一对光子计数的能力。 多光子或多通道重合计数很快就会变得麻烦而昂贵，而最大的重合计数率受到每次启动/停止事件所需的转换时间的限制，一般为〜1μs。 近年来，针对这些问题的几种解决方案已经发展为特定的应用，包括量子信息处理，4-6 荧光测量，7-8 X-射线显微镜，9 和物理教育。10-11

在这篇文章里，我们会详细介绍一个新的多通道符合计数模块（CCM），可以使用现成的集成电路组件来构建成本低于600美元的符合计数模块（CCM）。 从多达四个TTL信号作为输入开始，CCM可以记录任意2,3或4组符合（或单个通道计数）的组合，并具有短至12 ns的重合窗口。 编程到现场可编程门阵列（FPGA）的八个板载寄存器可以计数用户定义的符合时间间隔为20μs到1s的符合。 计数数据通过USB接口传输到个人计算机，通过可自由使用的软件收集，集成，显示和存储到磁盘。12-13 可以从我们的网站免费下载本CCM的构建和操作资源，包括组装指南，操作手册和数据采集软件（用于LabVIEW或作为独立的可执行文件）。13

\*david.branning@trincoll.edu; phone 1 860 297-4048

# 2. 设计

## 2.1 概述

框图如图1所示（整个电路的完整原理图可在线获得）。13

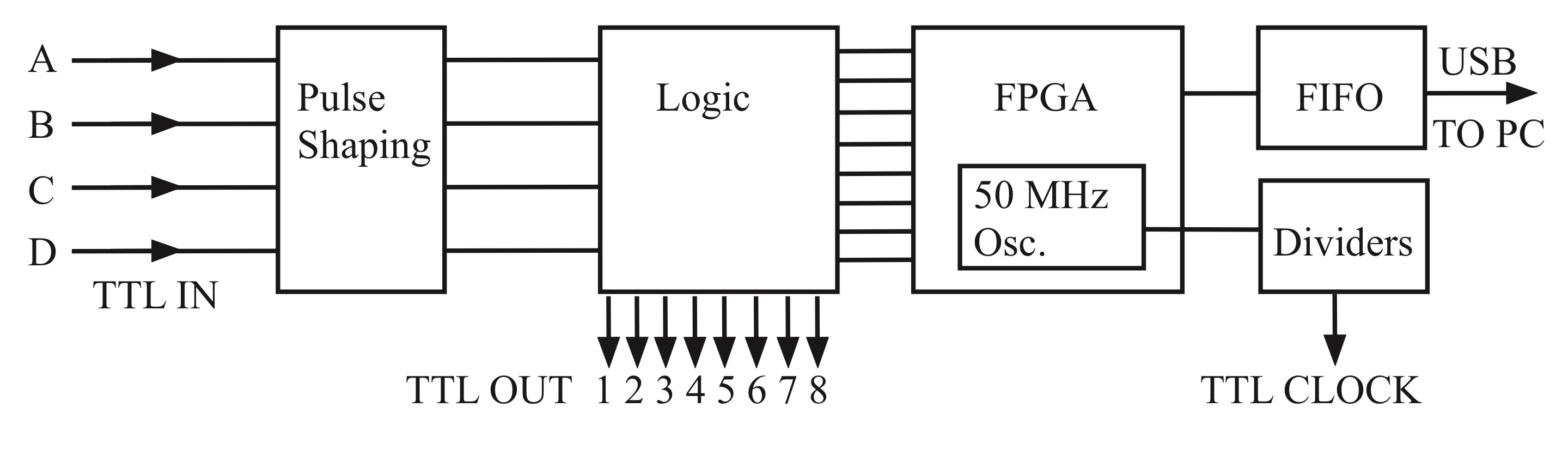


Fig. 1. CCM架构的框图。

输入A，B，C和D是由单光子计数模块（SPCM）构成的5伏TTL脉冲。 CCM的每个输入都具有50Ω或1kΩ的可选阻抗。 输入脉冲的持续时间缩短，然后扇出，形成图2中的八个重合逻辑电路的输入。八个输出通道被发送到BNC输出，并且还被发送到FPGA上的计数寄存器。 计数寄存器中的值被发送到先进先出（FIFO）缓冲区，在那里它们被捆绑发送到到由计算机通过USB接口读取的阵列中。 此外，FPGA的50 MHz主振荡器被分频以产生1 Hz至10 MHz的TTL时钟输出，可用于同步其他设备或自我测试CCM。

## 2.2 符合计数方法

下面显示了确定符合的基本方法。 输入信号A，B，C和D被发送到或门，然后被发送到四路与门的输入。 当且仅当所有四个输入同时为真时 - 也就是说，如果四个检测器脉冲同时到达门，与门的输出才为真。

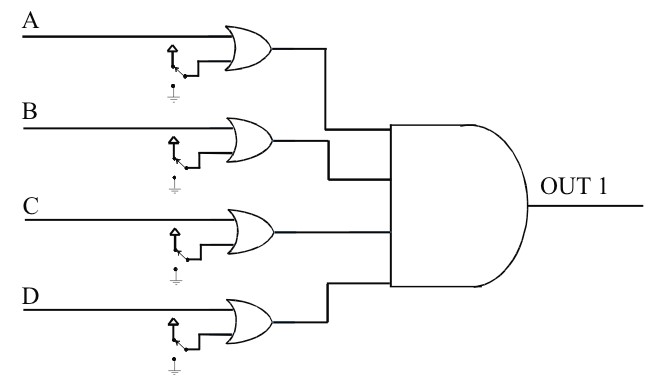


Fig. 2. 四路与门与每个输入上的或门。 对于每个输入（A，B，C，D），一个开关将其中一个或输入连接到0或5 V，以便在与门电路中将输入包括在内（0 V）或排除在外（5 V）。

或门允许用户定义四个检测器信号的任意子集，以便重合计数。 每个或门的第二个输入端保持高电平或低电平，由用户通过开关选择。 当任何特定输入的开关为高电平时，该输入被有效地从符合逻辑中移除。 任何与其对应开关的输入都保持低电平，但仍然必须同时到达以使与门的输出为真。 以这种方式，与门的输出可以确定四个输入之间2,3或4倍符合合的任意组合，或者简单地提供任何一个输入的单通道输入速率（通过排除其他三个）。 有八个4输入与门，每一个的输出都送到一个在FPGA上实现的计数器的输入端。 每个计数器通过USB接口定期将记录的计数数量传送到PC，然后重置以继续计数。

2.3 Pulse shaping

In order to improve the coincidence time resolution, each of the detector signals first enters a pulse-shaping circuit that reduces its width from the 20-50 ns pulse width typically obtained from commercial SPCMs.11 A diagram of the pulseshaping circuit is shown in Figure 3. The pulse shaping is accomplished by using two copies of the same input signal. One copy is time-delayed and inverted with respect to the other copy. Both copies are used as inputs of an AND gate. The output of the AND gate will only be high for the duration of the time delay. The time delays are accomplished by sending the signal through additional gates, e.g., AND gates with one input held high, that delay but do not otherwise alter the signal.

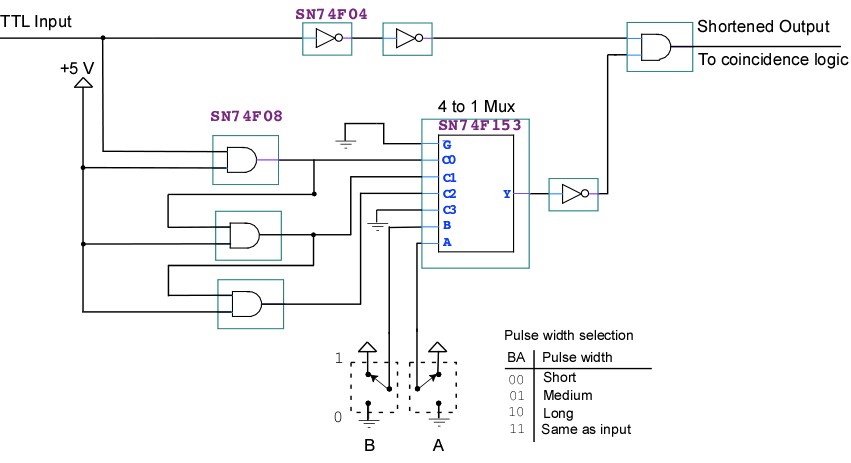


Fig. 3. Pulse-shaping circuit. The input signal (top line) and a time-delayed inverted copy of it are sent to the inputs of an AND gate (at top right). Toggle switches A and B are used with a multiplexer (Mux) to select the width of the shaped pulses of all four inputs, or to bypass the pulse-shaping circuit, leaving the pulse widths largely unchanged.

These manipulations allow for various discrete shortened pulsewidths, to be selected by the user by adjusting the position of two switches (A and B in Figure 3). The pulse-shaping section can also be bypassed, so that the full width of each pulse is passed directly to the logic section.

2.4 TTL channel outputs

In addition to being sent to the FPGA, the output of each 4-way AND gate is also connected to a line driver and a BNC output, providing TTL output pulses which can be monitored or counted externally. By using these output pulses as the inputs to additional CCM’s, coincidences among an arbitrarily high number of inputs can be monitored.

## 2.5 10 MHz Clock output

A TTL clock signal is provided at a BNC output by dividing the FPGA’s 50 MHz oscillator down to a user-selectable rate from 107 Hz to 1 Hz in decades. Because the TTL clock is derived directly from the master 50 MHz oscillator, it can be easily used to self-test the counting operations of the FPGA: the 10 MHz output should yield precisely 107 counts per second with no errors due to lack of synchronization. The 10 MHz clock output can also be used to synchronize other electronic pulse generators with the CCM for testing purposes, or to synchronize other data acquisition equipment in an experiment.

# 3. IMPLEMENTATION

## 3.1 F-series logic and FPGA hardware

The circuits of Figures 2 and 3 are implemented using F-series 5V TTL logic gates: these consist of AND gates, OR gates, inverters, multiplexers, and line buffers in the familiar 14-, 16-, or 20- pin DIP packages. The counting registers and USB capabilities are provided by an 80-pin MORPH-IC module from Future Technology Devices International (FTDI), which contains an Altera Acex 1K FPGA and a USB interface with FTDI’s FT2232D FIFO buffer.

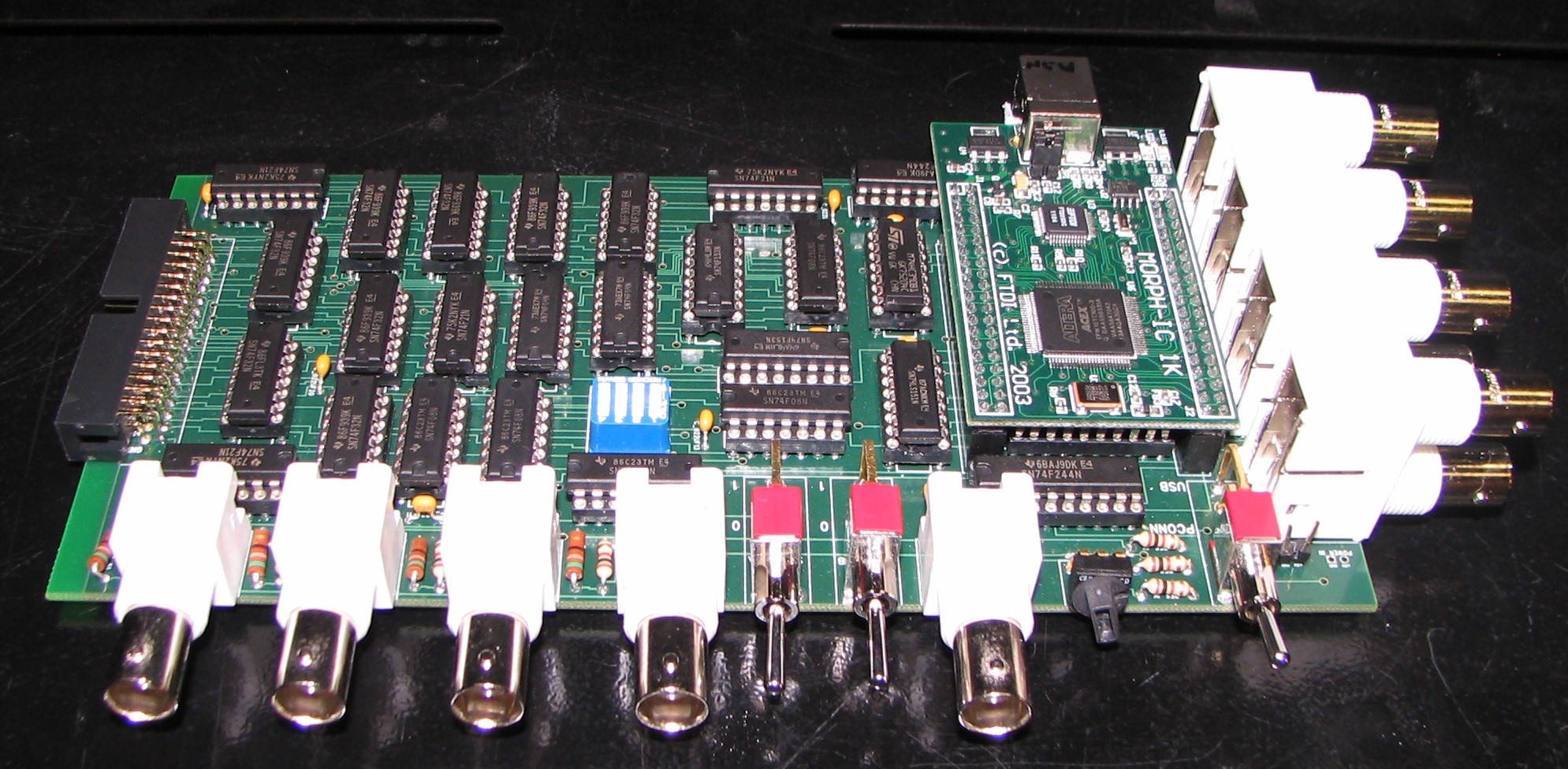


Fig. 4. An assembled logic board. BNC inputs A, B, C, and D are on the lower left. The TTL outputs 1-8 are on the far right. The MORPH-IC module is just to the left of these.

The logic chips and FPGA are mounted on a custom-manufactured 4-layer circuit board as shown in Figure 4. The boards can be manufactured by various online suppliers using the gerber files that are freely available from our web site.13

## Pushbutton controls

The switches connected to the OR gates that determine which coincidences are counted are latching pushbuttons, with an embedded orange (590 nm) LED. The switches are double-pole double-throw (DPDT), with one pole used to control the logic and the other used to control the LED. When a switch is depressed, the center pole for the logic is connected to ground and the LED is lit, indicating that the corresponding input is included in the 4-way AND logic.

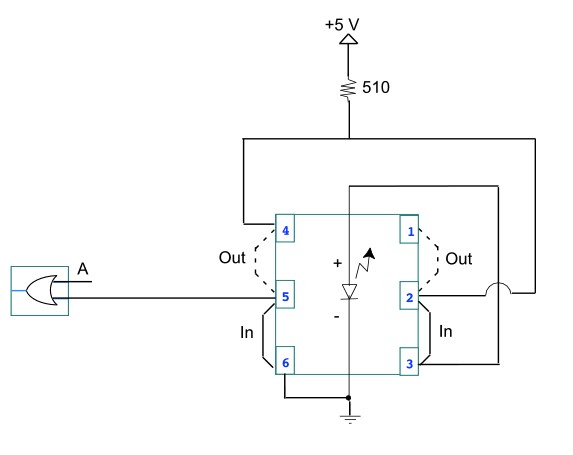


Fig. 5. Pushbutton wiring. When the button is IN, the left center pole (terminal 5) connects one input of the OR gate to ground. This means that input A, at the other OR input, is included in the coincidence circuit (see Fig. 2). To indicate this, the LED is lit by connecting it to +5 V using right terminals 2 and 3. A 510-ohm resistor limits the current through the LED. When the button is OUT, the LED is not lit, and the OR gate terminal is raised to +5 V, removing input A from the coincidence logic.

The switches are arranged in a 4x8 grid as shown in Figure 6. The four rows correspond to the four inputs, and the eight columns correspond to the eight counters. In this way the user can very easily set (and observe) which coincidences are being registered by which counter.

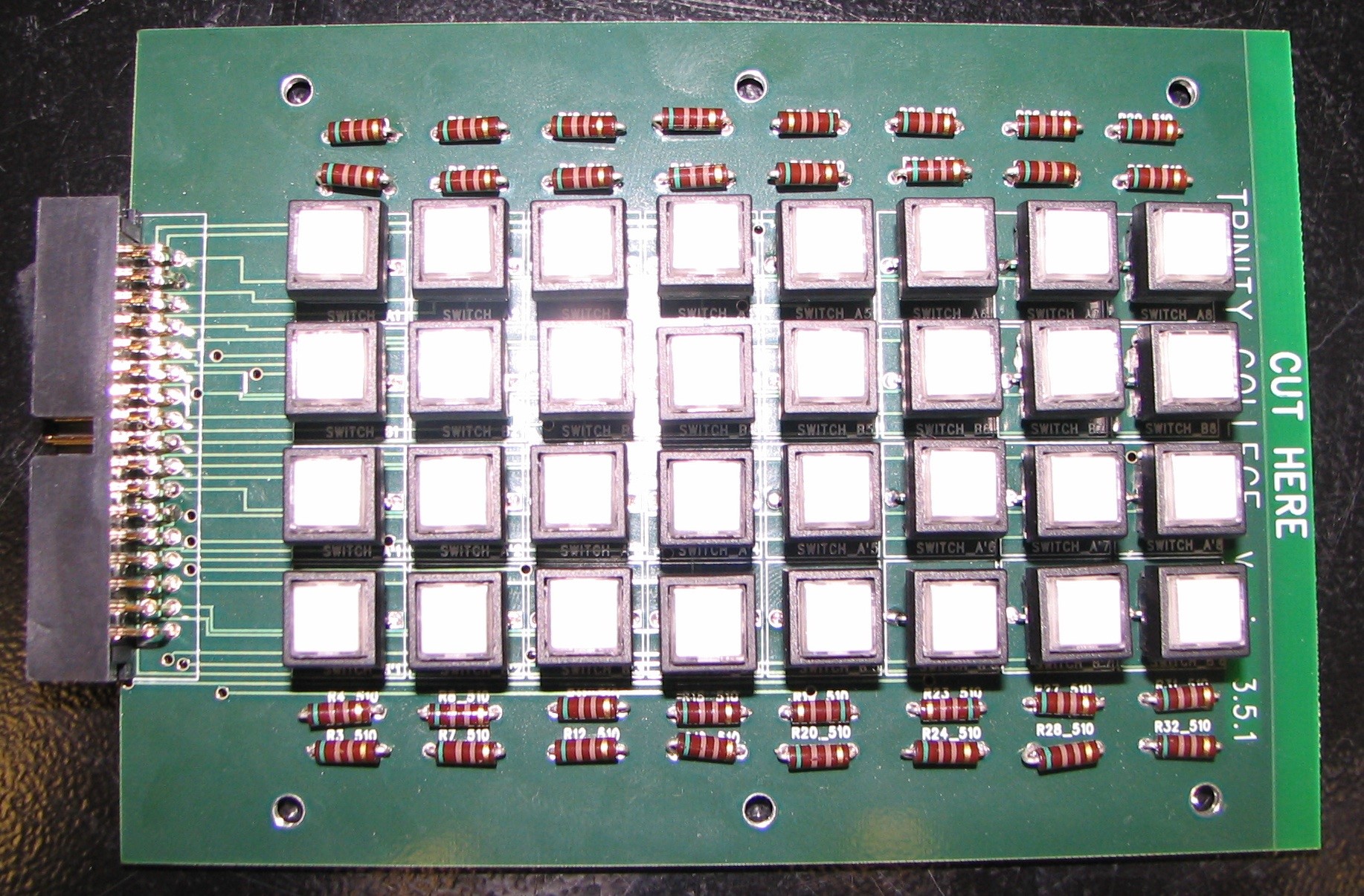


Fig. 6. An assembled control board. The pushbuttons indicate, for each output channel 1-8 (left to right), which of the four inputs A – D (top to bottom) are in the coincidence circuit. The +5 V or 0 V control signals for each OR gate (Fig. 2) are sent to the logic board (Fig. 4) using a ribbon connector which attaches at the left side.

## Final Assembly

The logic board (Figure 4) and control board (Figure 6) are connected together with a 34-conductor ribbon cable and housed in a project box as shown in Figure 7.

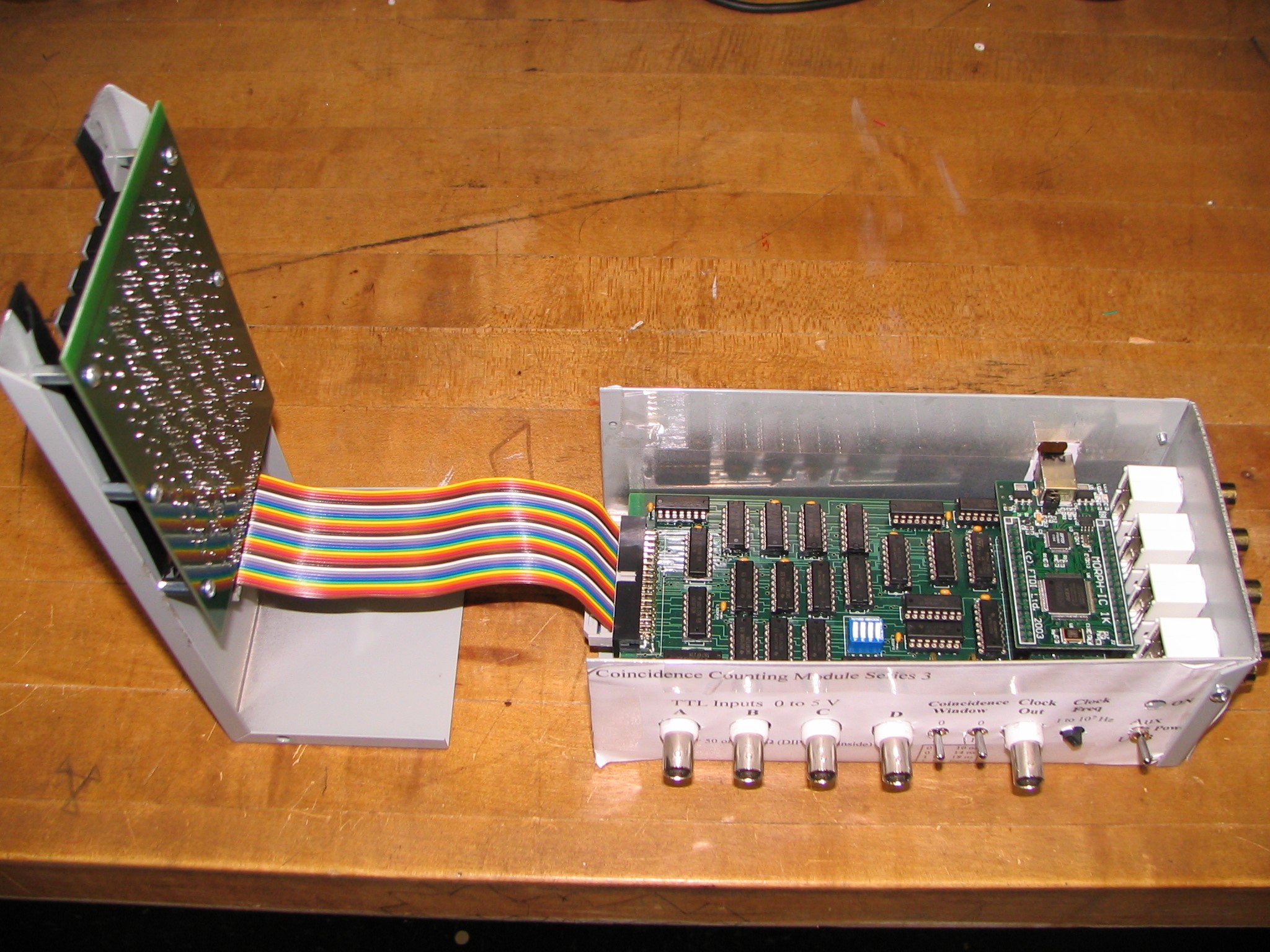


Fig. 7. Final CCM assembly.

The paper labels, which also serve as guide templates for drilling and cutting the project box, are available as part of the assembly guide for the CCM.13

## FPGA counter operation

The FPGA is configured by flashing a pre-written and compiled VHDL program onto it over USB. This program creates eight (or six) independent counting registers from cells in the FPGA, with 16 bits (or 20 bits) available in each channel register. The number stored in each counting register is incremented on the leading edge of each TTL pulse from the 4input AND gate. After a user-defined counting time (20 µs to 1 sec) has elapsed, the value in each counting register is copied to a storage register, and the counting registers are reset to zero. While the counting registers begin incrementing again, the storage register values are written into the FIFO buffer. After a predefined number of storage values are written to the buffer, they are transferred in a block to an array in the computer RAM via USB. The sets of count values in this array are then integrated for a user defined time interval, displayed on the computer monitor, and/or stored to hard disk; these tasks, and the flashing of the VHDL program, are managed by a LabVIEW routine that is freely available.12 **3.5 Blind cycles**

The transfer of the counting register values to the storage registers within the FPGA occupies one cycle of the FPGA’s 50 MHz master oscillator; during this 0.2 µs time interval, the counting registers cannot be incremented, and are therefore “blind” to the arrival of any new TTL pulses. One such “blind cycle” will occur after each counting time interval has elapsed; thus, for an elapsed time *T*, the true duration of active data acquisition time is

*Tactive* =*T*⎡⎣1-*R*(50 *MHz*)⎤⎦ (1) where *R* is the (user-selected) rate of data acquisition. The available values for *R* range from 1 Hz to 50 kHz.

# 4. PERFORMANCE

## 4.1 Pulse-shaping

The pulse-shaping circuit of Figure 3 was tested with 3-V TTL signals from a function generator. For this input (Channel A), the shortened pulses have durations of 7.5, 9.0, or 11.5 ns (± 0.5 ns) measured full width at half maximum, while the “11” setting creates a signal that is ~10 ns longer than the input pulse.

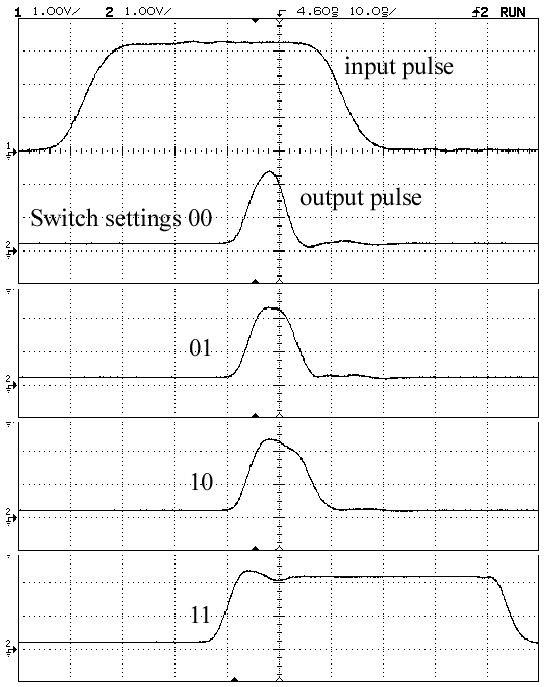


Fig. 8. Pulse profiles before and after the pulse-shaping circuit, for different settings of the A and B toggle switches.

## 4.2 Counting periodic pulses

The CCM was tested with a TTL pulse generator and was able to count coincidences at fixed frequencies of up to 37 MHz without losses. To achieve this, the pulse generator was phase-locked to the FPGA’s master oscillator using the CCM’s 10 MHz clock output, and a phase offset was added to prevent input pulses from coinciding with the blind cycles. Above 37 MHz, the blind cycles could not be avoided, and exactly *R* counts per second were missing from the totals in the counting registers. Above 74 MHz, exactly 2*R* counts per second were missing. The totals remained stable up to 84 MHz; above this input rate, the coincidences fluctuated, and ultimately fell to zero at 147 MHz, as successive pulses overlapped within the rise/fall times of the AND gates.

Because the coincidences and the single-channel counts are sent to the FPGA through different logic gates, they may arrive at the counting register inputs at slightly different times due to chip-to-chip variations in the rise and fall times of the gates. For periodic pulse trains that are synchronized to the CCM, it may turn out that some of the single-channel counts arrive at the FPGA during a blind cycle, while all of coincidence counts avoid them. This leads to the odd result that more coincidences are counted than single-channel events. This artifact of the FPGA counting routines does not affect the single-channel and coincidence count pulses that are produced at the TTL outputs. It can be overcome by phase-shifting the pulse train relative to the 10 MHz clock signal.

## 4.3 8-fold coincidences

To test the scalability with multiple modules, the phase-locked pulses from the generator were fanned out to eight copies, and delivered to the inputs of two separate CCM’s. The 4-way coincidence TTL output from each CCM was fed to a third CCM, which counted them in coincidence as shown in Figure 9.

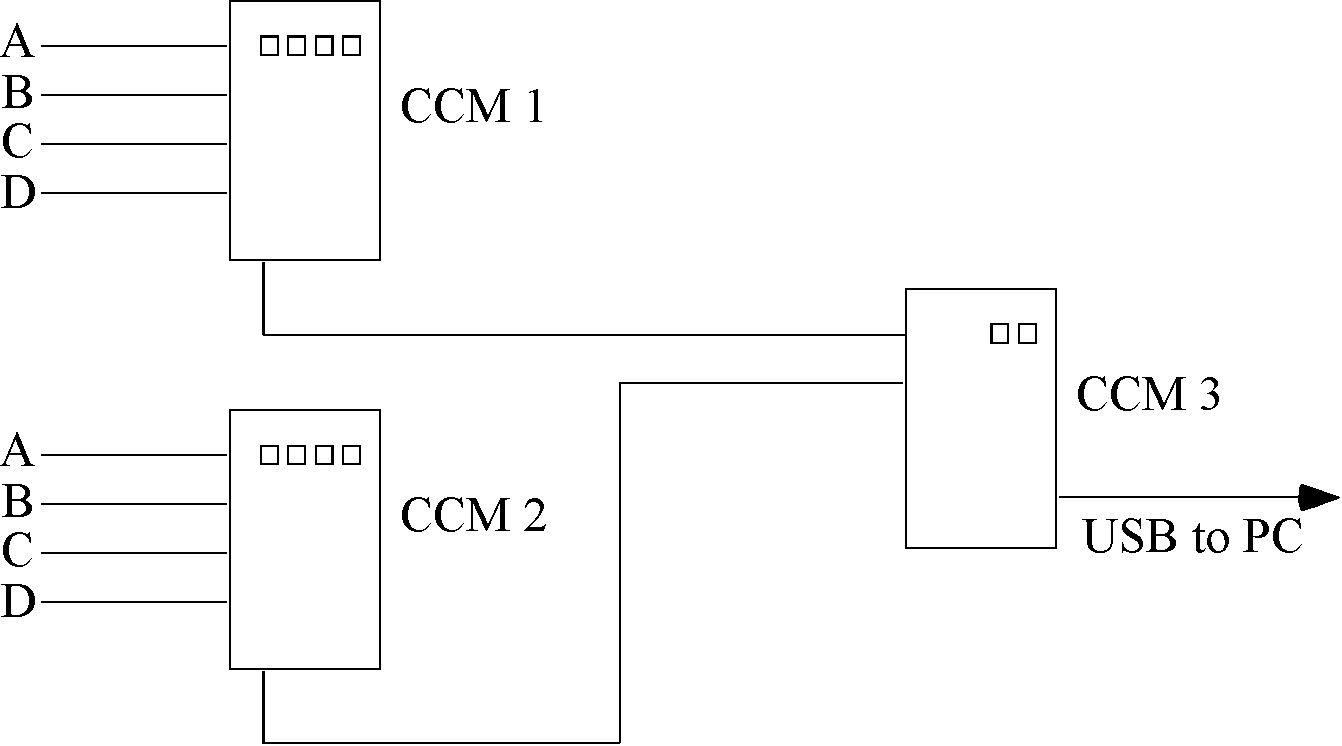


Fig.9. Configuration of 3 CCM’s used to count 8-fold coincidences.

In this manner the third CCM was able to register up to eight-fold coincidence counts at rates of up to 30 MHz (the limit of the fan-out), without losses.

## 4.4 Counting pseudo-random pulses

The CCM was also tested with pulses from a linear feedback shift register (LFSR), which generated a pseudo-random binary TTL output with controllable mean rates of up to 10 MHz. Figure 10a shows the single-channel response of the CCM for all four input channels. The pseudo-random input pulses were counted independently with external 50-MHz counters The CCM is observed to precisely count the input pulses, all the way up to the maximum output rate of the LFSR.

Fig. 10. (a) Mean single

-

channel counting rate in the CCM versus mean input pulse rate from an LFSR, acquired during 10

-

second intervals. A leas

t

-

squares fit (solid line) of the form

*y*

=

*mx*

yielded

*m*

= 1.002 ± 0.002. (b) Coincidence rates

*R*

*AB*

in

the CCM for pseudo

-

random input rates

*R*

*A*

and

*R*

*B*

on channels A and B, as a function of

*x*

=

*R*

*A*

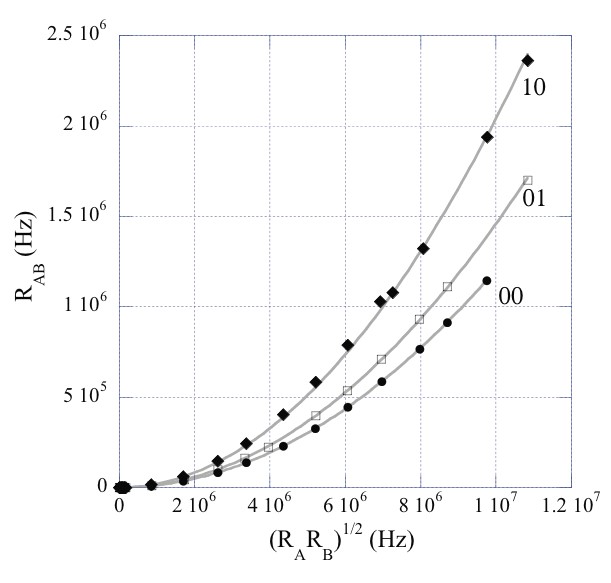
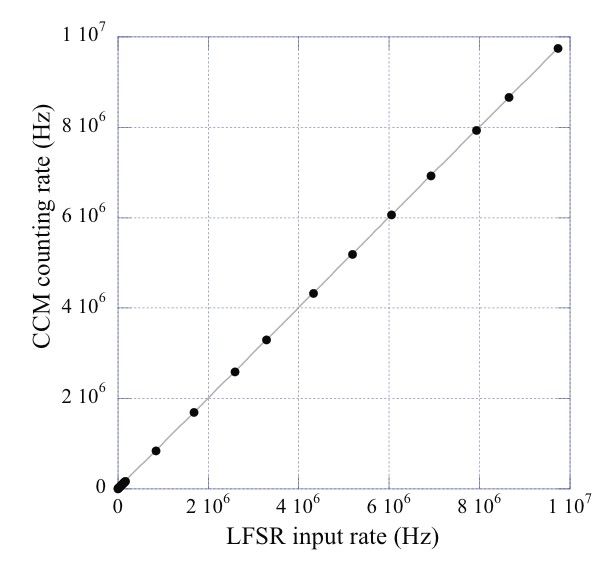
*R*

*B*

, for pulse

-

width



(

)

a

b

(

)

settings 00, 01, and 10. The solid lines are fits of the form *y* =τ*cx*2 in accord with Eq. (2). Similar results were observed in the other input channels

The coincidence times were measured using two independent LFSR’s on pairs of inputs. For randomly-arriving pulses with mean rates *RA* and *RB* in inputs A and B, the coincidence rate *RAB* is given by

*R* =τ*R R* . (2)

*AB c A B*

where τc is the coincidence time, equal to twice the pulse duration τ minus a small amount necessary for sufficient overlap.14 Single-parameter fits of the data to Eq. (2), shown in Figure 10b, yielded the values τc = 12.033 ± 0.006 ns, 14.56 ± 0.02 ns, and 20.38 ± 0.09 ns for the toggle positions 00, 01, and 10.

## 4.5 Counting pulses from random photon events

The coincidence times were also measured using two SPCMs and scattered light from a laser (which should produce independent, random streams of photons at the two detectors), yielding values of τc = 12.140 ± 0.007 ns, 14.133 ± 0.008 ns, and 21.47 ± 0.014 ns via Eq. (2). These coincidence times differ slightly from those measured with the LFSR’s, due to differences in the input pulse heights and shapes from the SPCMs.

The measured values of τc from both methods are consistent with the times that we would expect, given the duration of the output pulses from the pulse shaping circuit. Note that because of chip-to-chip variations in the CCM components, the coincidence time may vary slightly for coincidences between different pairs of detectors, but does not vary over time for a fixed pair of detectors.

# 5. CONCLUSION

For applications where time-tagging of individual photon detections is not needed, our CCM offers some attractive features. It takes four inputs, and determines user selectable 2-, 3-, or 4-fold coincidences (or single-channel counts) on eight counting channels. The CCM has a high maximum count rate of 84 MHz, and its coincidence resolution is as low as 12 ns. Furthermore, several CCMs can be cascaded together to count arbitrary *M*-order coincidences among *N* inputs. Because of its small size, low cost, and intuitive user interface, the CCM is also well-suited to undergraduate physics laboratories.

# 6. ACKNOWLEDGEMENTS

We thank David Ahlgren, Sagar Bhandari, John Bower, Brandon Clary, Adam Katcher, Sarthak Khanal, Larry North,

Steve Petkovsek, Young Ho Shin, Wayne Strange, and Jared Zimmerman for help with design, assembly and testing.

This work was supported by NASA through the Connecticut Space Grant College Consortium

## REFERENCES

1. J. Gea-Banacloche, “Optical Realizations of Quantum Teleportation,” Prog. Optics 46, 311-353 (2004).
2. Kok, P., Munro, W. J., Ralph, T. C., Dowling, J. P., Dowling, and Milburn, G. J., “Linear optical quantum computing with photonic qubits,” Rev. Mod. Phys 79, 135-174 (2007).
3. Scarani, V., Bechmann-Pasquinucci, H., Cerf, N., Dusek, M., Lütkenhaus, N., and Peev, M., “The security of practical quantum key distribution,” Rev. Mod. Phys. 81, 1301-1350 (2009).
4. Gaertner, S., Weinfurter, H., and Kurtsiefer, C., “Fast and compact multichannel photon coincidence unit for quantum information processing,” Rev. Sci. Instrum. 76, 123108 (2005).
5. Stumner, A., “Coincidence Counter,” http://www.physics.utoronto.ca/~astummer/pub/mirror/Projects/index.html
6. Polyakov, S., Migdall, A., Nam, S. W., “FPGA-based multicoincidence recipe and software,” (November 24 2009), http://www.nist.gov/pml/div684/grp03/multicoincidence.cfm
7. Felekyan, S., Kuhnemuth, R., Kudryavtsev, V., Sandhagen, C., Becker, W., and Siedel, C. A. M., “Full correlation from picoseconds to seconds by time-resolved and time-correlated single photon detection,” Rev. Sci. Instrum. 76, 083104 (2005).
8. Wahl, M., Rahn, H-J., Gregor, I., Erdmann, R., and Enderlein, J., “Dead-time optimized time-correlated photon counting instrument with synchronized, independent timing channels,” Rev. Sci. Instrum. 78, 033106 (2007).
9. Acremann, Y., Chembrolu, V., Strachan, J.P., Tyliszczak, T., and Stohr, J., “Software defined photon counting system for time resolved x-ray experiments,” Rev. Sci. Instrum. 78, 014702 (2007).
10. Dehlinger, D., and Mitchell, M. W., “Entangled photon apparatus for the undergraduate laboratory,” Am. J. Phys. 70, 898-901 (2002).
11. Branning, D., Bhandari, S., and Beck, M., “Low-cost coincidence-counting electronics for undergraduate quantum optics,” Am. J. Phys 77, 667-670 (2009).
12. Branning, D., Khanal, S., Shin, Y.H., Clary, B., and Beck, M., “Note: Scalable multiphoton coincidence-counting electronics,” Rev. Sci. Instrum. 82, 016102 (2011)
13. Branning, D., http://www.trincoll.edu/~dbrannin/Coincidence%20Counting/CoincidenceHome.htm [14] Eckart, C., and Shonka, “Accidental Coincidences in Counter Circuits,” Phys. Rev. 53, 752-756 (1938).