# Coincidence Counting Unit Using the Altera DE2

*J.W. Lord and M. Beck**
*Dept. of Physics, Whitman College, Walla Walla, WA 99362*
**beckmk@whitman.edu ; http://www.whitman.edu/~beckmk/QM/*

We have implemented a coincidence counting unit (CCU) on the Altera DE2 development and education board. The CCU takes inputs from up to four detectors, and has eight 32-bit counters that are used to count four singles counts and four arbitrary 2-, 3-, or 4-fold coincidences. The CCU has a coincidence-time resolution of less than 8 ns. Data is streamed from the CCU to a host computer over an RS232 serial interface. Here we provide software and instructions for implementing this CCU.

## I. Preliminaries

These are the preparations you need to make before you can install our software.

1. The DE2 board can be purchased from Altera (Academic price of about $270). Information about purchasing the board is available at

    - http://www.altera.com/education/univ/materials/boards/unv-de2-board.html

    Make sure you get the DE2 and not one of the other education boards (don't get the DE2-70, our software might work on that board but we haven't tested it).

2. To download the design files to the DE2 board you will need to install Quartus II software on your PC. Quartus has a free version which is available at

    - https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we.jsp

    When you get your DE2 from Altera it will come with a Quartus installation CD, but the website will almost certainly have a more up to date version. We used V. 7.2 for developing our software, so you'll need to have this version or newer; as of this writing Altera has released V. 8.0 and our design works fine with this version.

    Quartus is an environment for developing programs for Altera field programmable gate arrays (FPGAs). We used Quartus to develop our design using a language called VHDL. **Note** that for our software you only need to install Quartus II; you don't need the MegaCore IP Library, NiosII, or Model Sim.

    Installation instructions come with the download. **Note** that it is probably best to install Quartus II in the default folder that it suggests at installation. Do **NOT** install it in your

"Program Files" folder as that folder has a space in its name, and spaces in the file path can cause problems for Quartus II.

3. Quartus uses the "USB-Blaster" to download the design files onto the DE2 board. The driver for the USB-Blaster needs to be installed. Instructions for doing this are in the document titled "Getting Started with Altera's DE2 Board"; this document can be found on the DE2 CD: DE2_tutorials\tut_initialDE2.pdf . Other useful information is found in the user manual: DE2_user_manual\ DE2_UserManual.pdf

4. While you don't need a complete and detailed understanding of how to use Quartus to program the FPGA on the DE2, it is probably a good idea to have a basic understanding of how to do this. The place to start is on the DE2 CD: DE2_tutorials\tut_quartus_intro_vhdl.pdf .

## II. Downloading the design files to the DE2

Before getting started you should be warned to NOT recompile the project before downloading it onto the DE2! If you recompile most of the program will work, but the pulse shortening circuit will not function. This is because the pulse shortening is implemented during a post compilation phase that Quartus calls "ECO fitting". If the project is recompiled then the fitting process needs to be done over (this is a little tedious). If you are interested in this, or in learning more about the code, you can contact us and we'll send you a copy of Jesse Lord's thesis which has all the details. The steps below describe how to download the files without recompiling.

The project files are set to download the program onto the DE2 board using "Active Serial Programming" mode; the procedure for using this mode is described below. In this mode the files are permanently stored on the DE2, so powering off the board will not erase them.

1. The folder DE2_RS232_CCU_Q72, which contains all the project files, should be placed somewhere where the path down to it from C:\ contains no folders with spaces in their names (e.g. anywhere in "My Documents" is a bad place.)

2. Open this folder, then double click on the file "RS232coincidencecounter.qpf". This opens Quartus II and the project.

3. If you're using a newer version of Quartus, you might get a message that says "Do you want to overwrite the database...". Just click "Yes" and keep going.

4. Make sure the DE2 board is turned on and the USB-Blaster is plugged in. In Quartus II click on the programmer icon (upper panel in Fig. 1). **Note:** on newer versions of Quartus the programmer might open in a separate window. Click on the "Hardware Setup…" icon (Fig. 2) to open the Hardware Setup window.

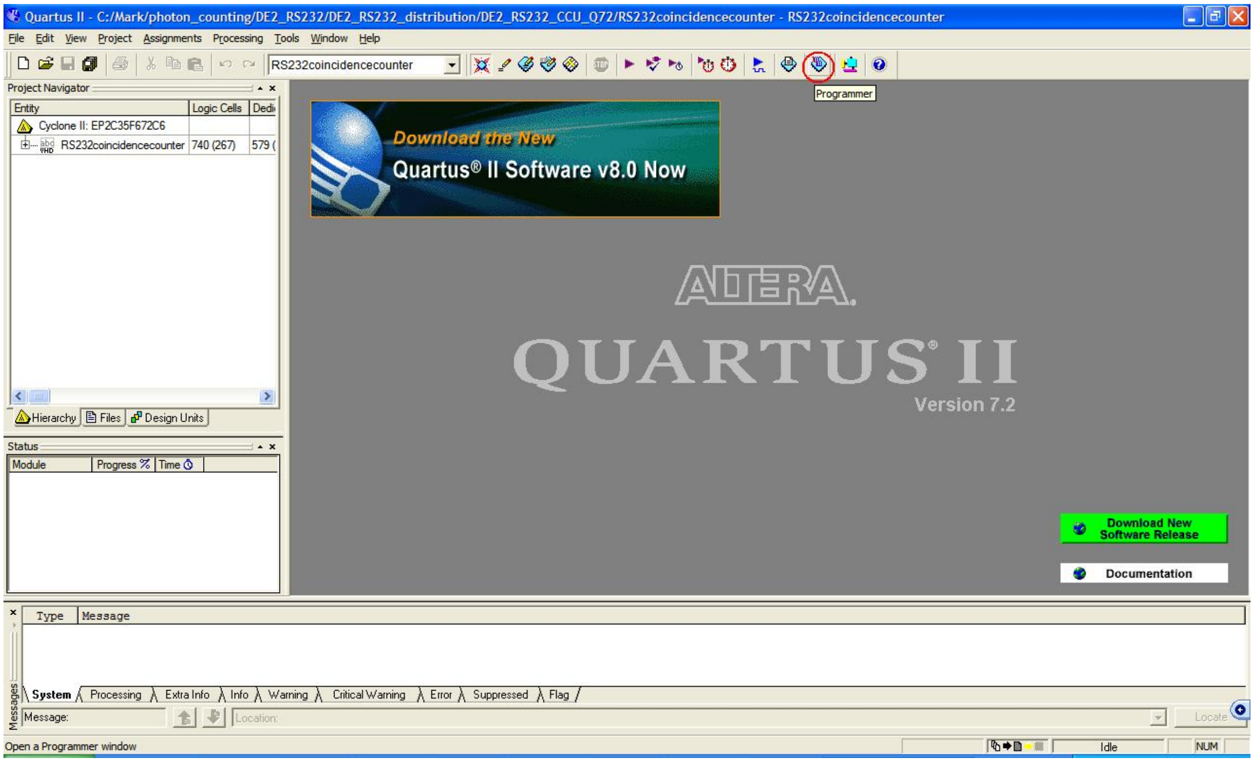


*Figure 1: Programmer Button is circled.*

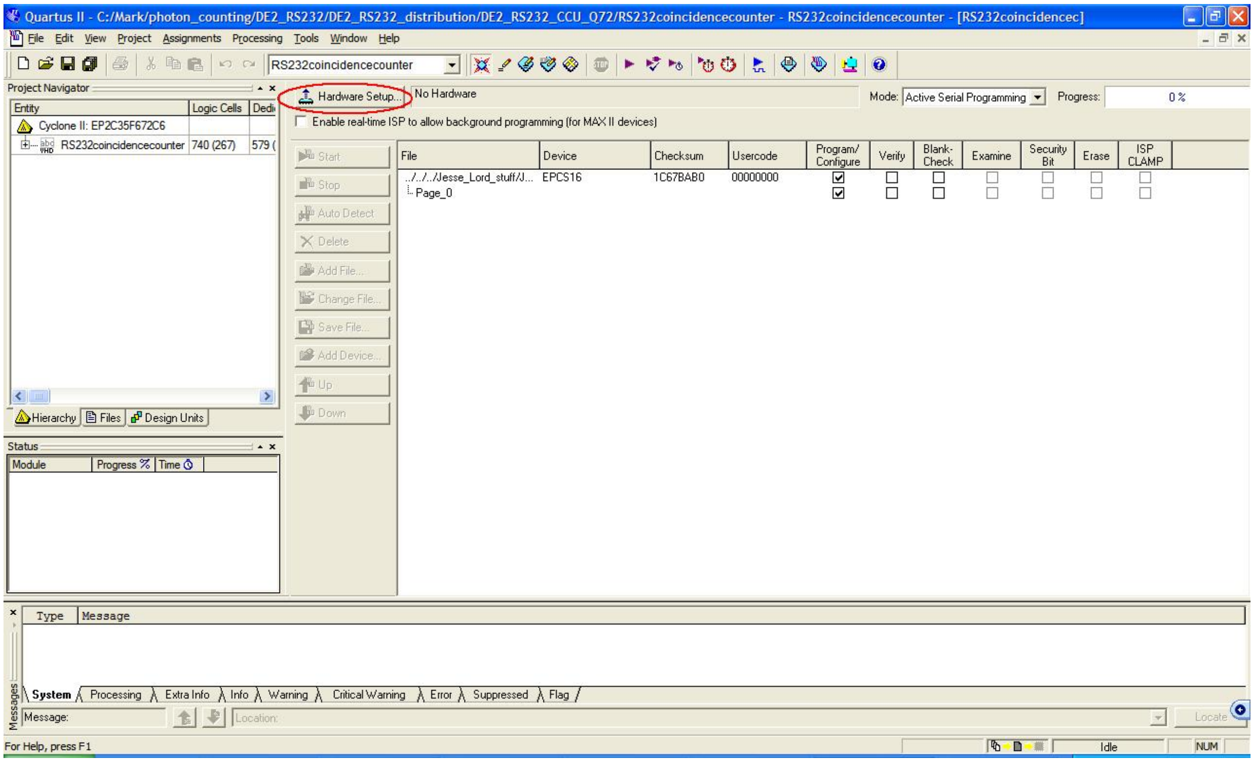


*Figure 2: Hardware Setup Button is circled.*

5. If the driver is properly installed, the USB-Blaster [USB-0] option should be available under "Currently selected hardware:" (Fig. 3). Make sure it is selected.

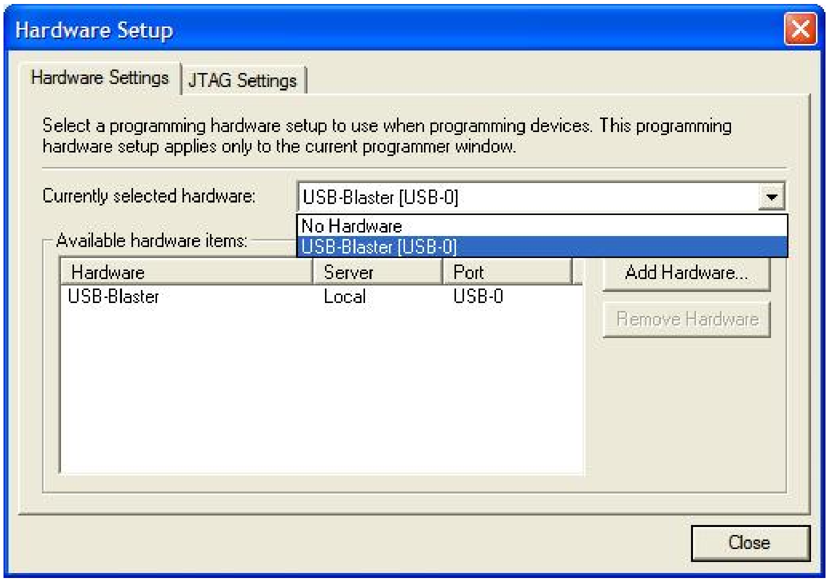


*Figure 3: Setting up the USB-Blaster connection.*

6. Make sure that next to Mode: "Active Serial Programming" is selected.

7. It doesn't hurt to make sure you've selected the correct file. Click on the file name (the line of information about the file should become highlighted), then click on the "Change File" button. For Active Serial Programming you want to make sure to use the file "RS232coincidencecounter.pof".

8. Make sure that both boxes are checked in the column labeled "Program/Configure". **Note:** On newer versions of Quartus there may be only one line of program information and one check box.

9. On the DE2 board, make sure that the "PROG/RUN" switch on the left side of the board is set to "PROG".

10. Press the "Start" button to download the program onto the DE2 board (Fig. 4). Once the progress indicator reads "100%" the programming is finished (this may take a minute or two to complete--check the messages at the bottom to see if an error occurred—Fig. 5).

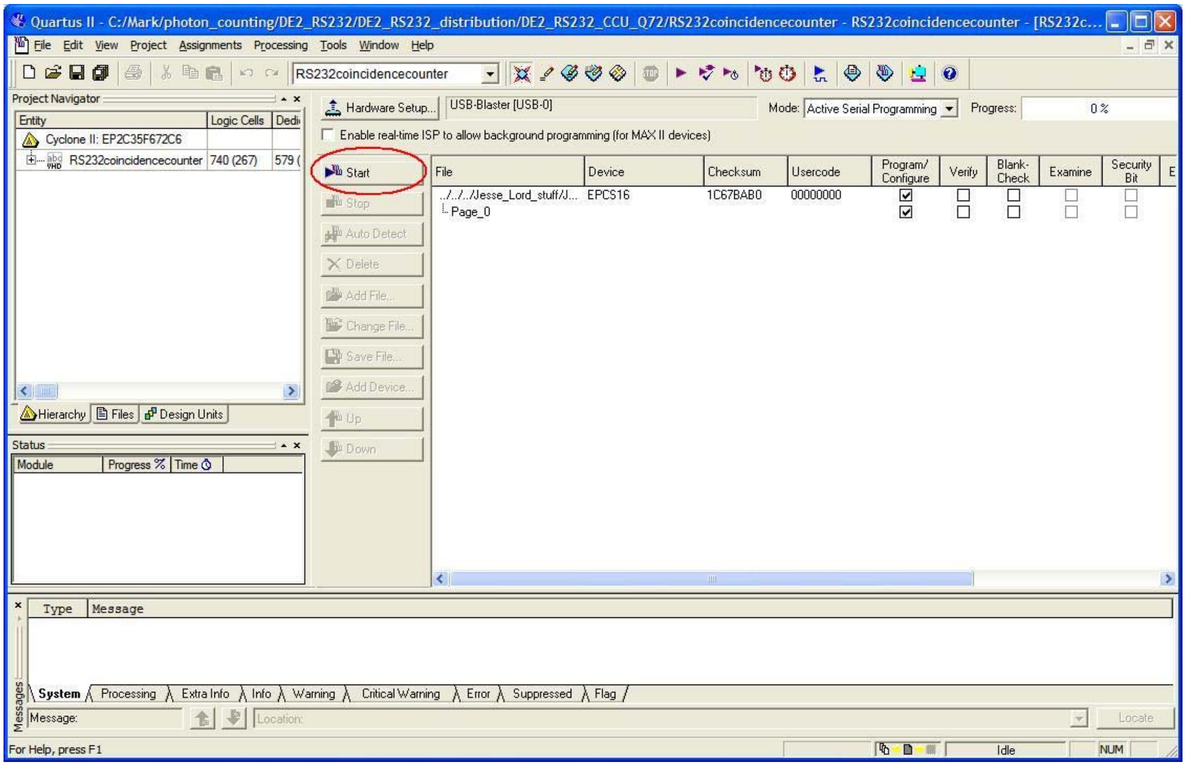

Figure 4: Starting the programming of the FPGA. Note that in more recent versions of Quartus there may be only one line and one check box under "Program/Configure"

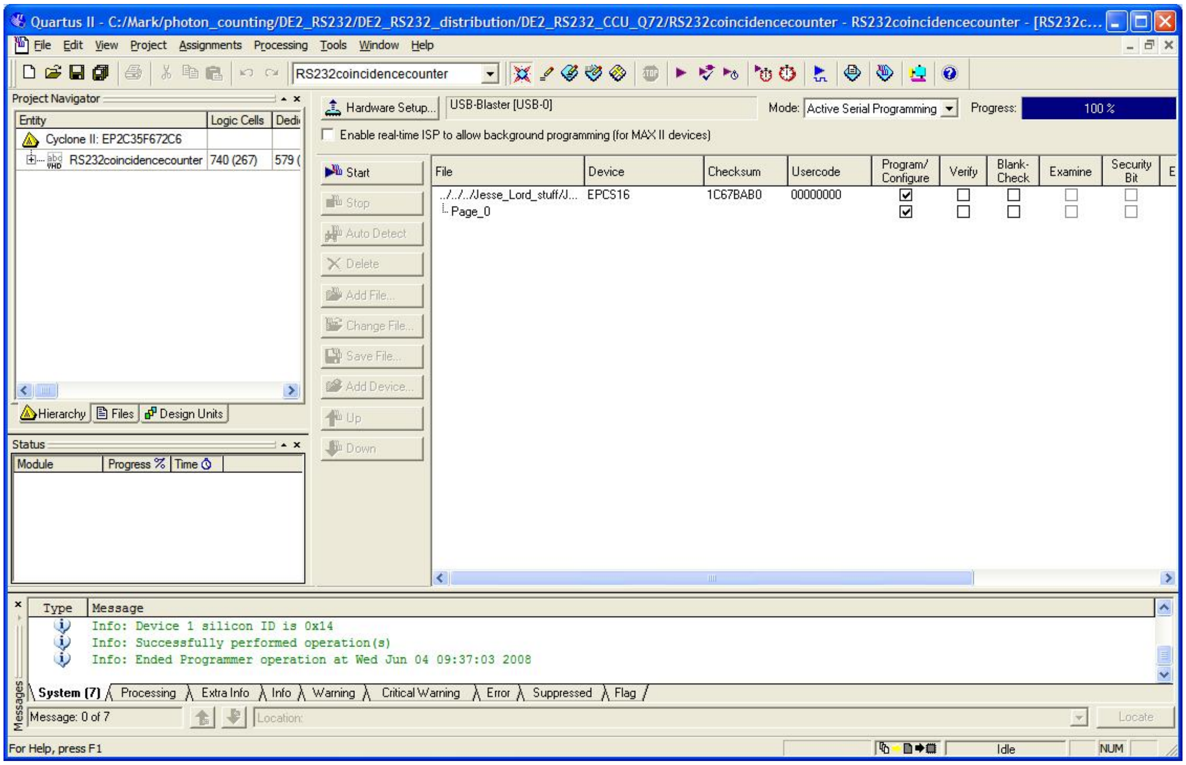

Figure 5: Successful completion.

11. When Quartus has finished downloading the software, switch the “PROG/RUN” switch on the left side of the DE2 board to “RUN”.

12. If the green “TXD” light on the back right of the board is flashing at 10 Hz then chances are you’ve successfully installed the software. If not, it is sometimes necessary to turn off the power to the board and then turn it back on to get the board properly configured.

## III. Building the adapter box

The only thing you need to build is an adapter box to get the signals from the single photon counting modules (SPCM’s) onto the DE2 board. Included with this distribution is a parts list for this box; all parts for the box are from Mouser Electronics (www.mouser.com).

A few notes about the parts list and assembling the box:

1. The ribbon cable specified already has female connectors on it, so you don’t necessarily need to buy the two other 40-pin female connectors. However, the connectors that come with the cable do not have a tab specifying the orientation of the connector (it’s possible to connect the cable backwards) whereas the specified connectors do have a tab. I just cut the old connectors off the cable and replace them with the new connectors.

   Alternatively, if you already have 40-conductor ribbon cable (minimum length from Mouser is 100 ft, which is why I suggest the 18 inch cable) you don’t need to buy the cable with the connectors on it. This cable can frequently be scrounged from an old dead computer.

2. You only need a few inches of ribbon cable on the inside of the box connected to the 40-pin male connector. When I cut the connectors off of the 18-inch cable, I also cut off a few inches of cable for this purpose.

   Cutting apart the individual conductors on the ribbon cable inside the box must be done carefully, and is best accomplished with a pair of scissors (not a knife). We suggest cutting apart the conductors before you attach the cable to the male 40-pin connector. Just cut apart about 3 inches of each conductor, then leave another inch or so connected together for crimping in the male connector.

3. The serial cable connects the DE2 board to the computer.

4. If your computer doesn’t have an RS232 serial port (some newer computers don’t) you can buy a PCI plug-in card that implements RS232. We’ve used the SIIG 2-port serial universal PCI card (part # 15-150-134 from newegg.com)—it has 2 serial ports (we use one for the DE2 board and the other for the Newport NewStep controller).

We've tried USB-RS232 converters with mixed success (that's why we recommend the PCI card). These converters are known for having USB driver issues, and we found a conflict between our converter driver and the USB-Blaster driver on one computer.

Depending on which photon counter you have, the adapter box needs to be assembled slightly differently.

For the SPCM-AQ4C:

These SPCMs output 5V pulses into 50Ω, but the DE2 operates on 3.3V logic. It is therefore necessary to put a voltage divider in each signal path to get the correct signal voltage for the DE2; this voltage divider also needs to provide the correct 50Ω termination for the signal. The voltage divider circuit we use for each signal (4 dividers total) is shown in Fig. 6a.

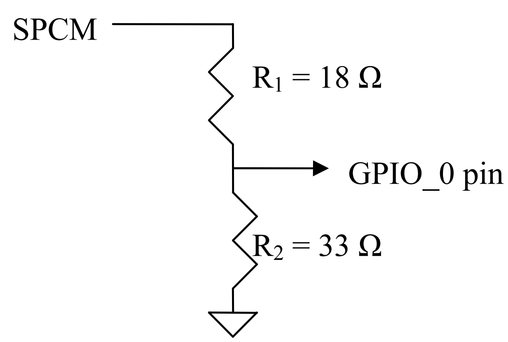


*Figure 6a: This is the voltage divider used in the adapter box that connects the outputs from the SPCM-AQ4C to the GPIO_0 pins.*

For the SPCM-EDU (available from ALPhA):

These SPCMs output 2.2V pulses into 50Ω, which is high enough to be above threshold for the 3.3V logic on the DE2. No voltage divider is necessary, but you still need to provide the correct 50Ω termination for the signal (Fig. 6b). In the parts list you'll need to replace the 33Ω and 18 Ω resistors by 50Ω resistors.

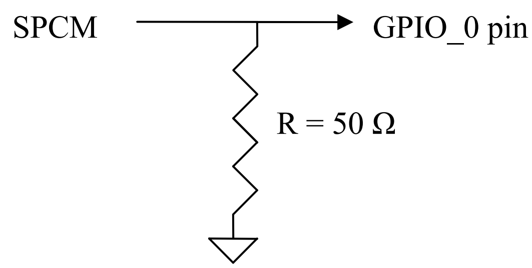


*Figure 6b: This is the termination resistor in the adapter box that connects the outputs from the SPCM-EDU to the GPIO_0 pins.*

When building the box, the only trick is making sure that you wire the correct pin on the header to the correct connector in the box. The pins on the header are labeled starting from the top left—odd pins on the left and even pins on the right. Put the female connectors on the ribbon cable to ensure that pin 1 of the DE2 header is connected to Pin 1 of the male connector on the adapter box. There are two headers on the DE2 board: the input signals are connected to the GPIO_0 header (the header on the left). The SPCM signals must be connected to the pins shown in Table 1.

| SPCM | GPIO_0 pin |
|---|---|
| A | 13 |
| B | 15 |
| A’ | 17 |
| B’ | 19 |

Table 1: *This table shows the connections between the SPCMs and the DE2 board through the GPIO_0 pins.*

Finally, to ensure signal integrity the following GPIO_0 pins should all be connected to the case ground of the box: 12, 14, 16, 18, 20.

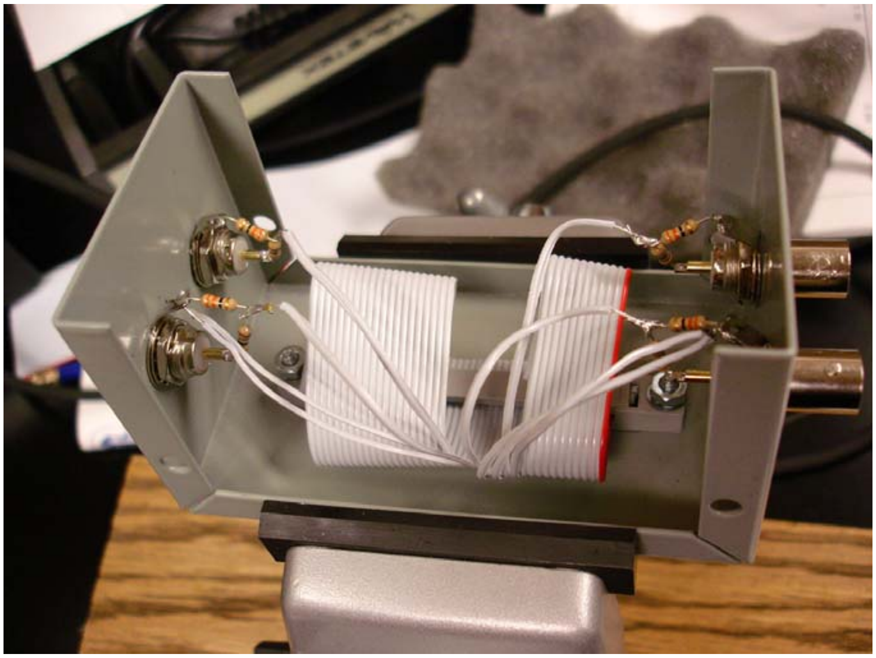

*Figure 7: The inside of the adapter box,*

A picture of our adapter box is shown in Fig. 7.

## IV. Using the coincidence counter

A picture of the assembled setup is shown in Fig. 8. The counters all count for 0.1 s, write their data to the serial port, then reset. Thus, data streams from the DE2 to the computer at a rate of 10 Hz. The LabView vi “Simple_altera_rs232.vi” demonstrates how to read data from the DE2. You can use the sub-vi’s in this program to write your own code. Furthermore, more sophisticated programs for using the counter can be found by going to the project webpage <http://www.whitman.edu/~beckmk/QM/> , and then following the link to the LabView vi’s.

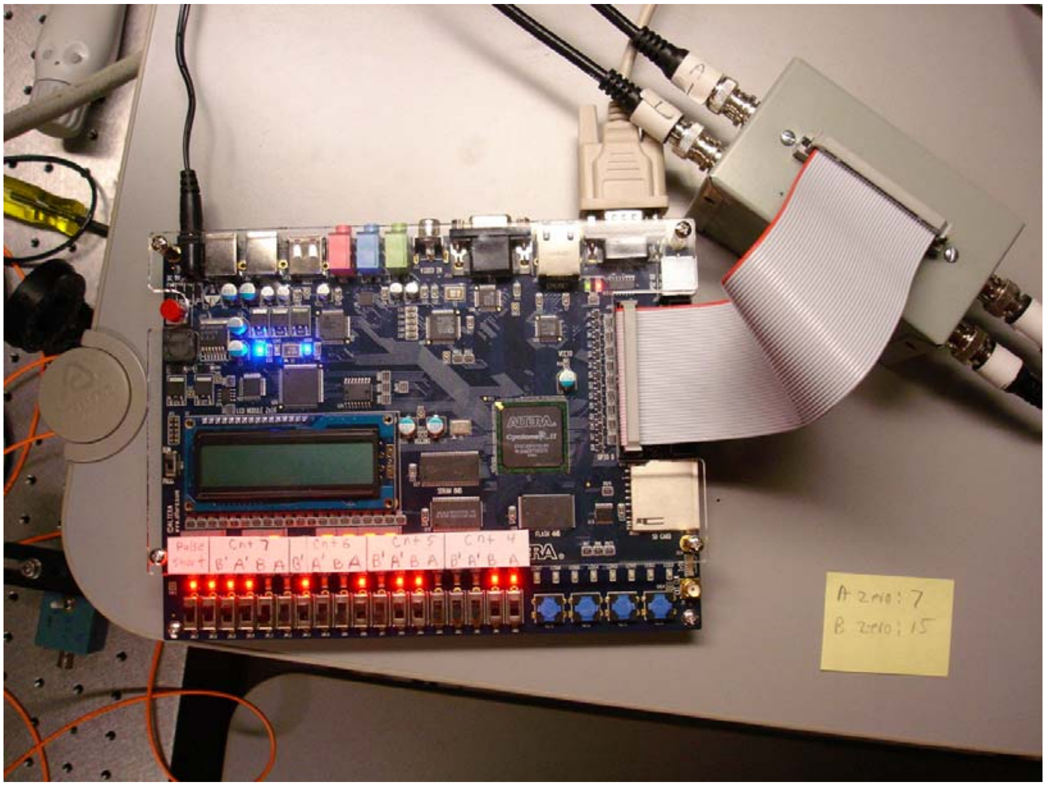

*Figure 8: The assembled coincidence unit.*

Essentially, the DE2 board sends to the computer an array of 8 32-bit numbers, representing the data from 8 different counters. The first 4 elements of the array are the singles counts (Counter 0 – Counter 3), in order: A, B, A', B'. The last 4 elements are coincidence counts (Counter 4 – Counter 7). The coincidences are determined by the settings of the switches SW0 – SW15 on the DE2 board. Each counter uses four switches to determine which coincidence it counts—Counter 4 uses switches SW0 – SW3, Counter 5 uses SW4 – SW7, etc; when a switch is turned on the corresponding LED lights up. With this arrangement any of these 4 counters can determine any arbitrary 2-, 3- or 4-fold coincidence. How this works is best illustrated by Fig. 9.


*Figure 9: How the switches determine which coincidence is counted. Here, Counter 4 will count AB coincidences, and Counter 5 will count A'B coincidences.*

| SW0 | SW1 | SW2 | SW3 | Output Coincidence |
|---|---|---|---|---|
| A | B | A’ | B’ | |
| 0 | 0 | 0 | 0 | None |
| 0 | 0 | 0 | 1 | B’ |
| 0 | 0 | 1 | 0 | A’ |
| 0 | 0 | 1 | 1 | A’·B’ |
| 0 | 1 | 0 | 0 | B |
| 0 | 1 | 0 | 1 | B·B’ |
| 0 | 1 | 1 | 0 | B·A’ |
| 0 | 1 | 1 | 1 | B·A’·B’ |
| 1 | 0 | 0 | 0 | A |
| 1 | 0 | 0 | 1 | A·B’ |
| 1 | 0 | 1 | 0 | A·A’ |
| 1 | 0 | 1 | 1 | A·A’·B’ |
| 1 | 1 | 0 | 0 | A·B |
| 1 | 1 | 0 | 1 | A·B·B’ |
| 1 | 1 | 1 | 0 | A·B·A’ |
| 1 | 1 | 1 | 1 | A·B·A’·B’ |

*Table 2: The truth table showing how the switches choose which coincidences will be counted. Note that while the switches SW0-SW3 are used in this truth table, the same is true for the switches SW4-SW7, SW8-SW11, and SW12-SW15.*

A truth table for the switch operation of Counter 4 (other counters are the same with different switch numbers) is shown in Table 2.

In order to allow users access to several of the relevant internal signals, copies of these signals have been wired as outputs to the DE2 GPIO_1 pins (the 40-pin header on the right of the board), as shown in Table 3. These signals can be used to debug the circuit design, to trigger other events, etc.

| Signal | GPIO_1 Pin |
|---|---|
| A | 40 |
| B | 38 |
| A’ | 36 |
| B’ | 34 |
| $A_s$ | 32 |
| $B_s$ | 28 |
| $A'_s$ | 26 |
| $B'_s$ | 24 |
| Coincidence_4 | 22 |
| Coincidence_5 | 20 |
| Coincidence_6 | 18 |
| Coincidence_7 | 16 |
| 10 Hz clock | 14 |

*Table 3: This table shows the signals that are wired to GPIO_1 pins. The signals A, B, etc. are copies of the inputs coming from the SPCMs. The signals $A_s$, $B_s$, etc. are the pulse shortened versions of these input signals. Coincidence_4, Coincidence_5, etc. represent the outputs of the coincidence determination logic, which are input to Counter 4, Counter 5, etc. The 10 Hz clock is a pulse that is synchronous with the 10 Hz rate that the counters are gated at.*

## V. Performance

Control of the shortening of the input pulses (control of the coincidence time resolution) is done using switches SW16 and SW17. There are 4 possible settings for these switches: one setting passes the pulses from the SPCM’s unchanged (both switches off), while the other 3 settings provide different amounts of pulse shortening. Table 4 presents some measured data on coincidence time resolutions using the DE2 CCU. This data was obtained by illuminating all 4 SPCM’s with Poissonian light (scattered laser light). With no pulse shortening we obtain a time resolution of about 40 ns, while at the shortest setting we obtain a 7-8 ns coincidence time resolution.

| Coincidences | Setting | Δt (ns) |
|---|---|---|
| AB | 00 | 41.55 ± 0.31 |
| A`B | 00 | 41.76 ± 0.42 |
| AB` | 00 | 41.08 ± 0.42 |
| A`B` | 00 | 41.31 ± 0.41 |
| AB | 11 | 7.05 ± 0.09 |
| A`B | 11 | 7.75 ± 0.20 |
| AB` | 11 | 7.03 ± 0.18 |
| A`B` | 11 | 7.20 ± 0.33 |
| AB | 11 | 7.13 ± 0.08 |
| A`B | 11 | 7.89 ± 0.16 |
| AB` | 11 | 7.45 ± 0.21 |
| A`B` | 11 | 7.14 ± 0.18 |
| AB | 10 | 11.63 ± 0.26 |
| A`B | 10 | 11.63 ± 0.16 |
| AB` | 10 | 11.20 ± 0.17 |
| A`B` | 10 | 10.95 ± 0.22 |
| AB | 01 | 16.55 ± 0.18 |
| A`B | 01 | 15.02 ± 0.21 |
| AB` | 01 | 15.76 ± 0.25 |
| A`B` | 01 | 14.05 ± 0.23 |

*Table 4: This table shows the coincidence time resolution of each of four two-fold coincidences, as determined by the setting of switches SW16 and SW17. The second '11' setting was a separate measurement of the time resolution using a different DE2 board that was programmed by a different computer. The results between the two boards compare favorably. The error quoted is given by the standard deviation of six measurements.*

It has been well documented that coincidence counting systems based on time-to-amplitude converters suffer from dead-time problems that cause a saturation in the number of detected coincidence counts at high count rates. Our previous discreet-IC-based CCU showed no such saturation [1-3]. In this respect our DE2-based system behaves identically to the discrete-IC-based version, also showing no saturation as seen in Fig. 10.

We have used this new CCU to perform all of the undergraduate quantum mechanics experiments described in [4-8], and find that it behaves as well as the discreet-IC-based CCU for all of them.

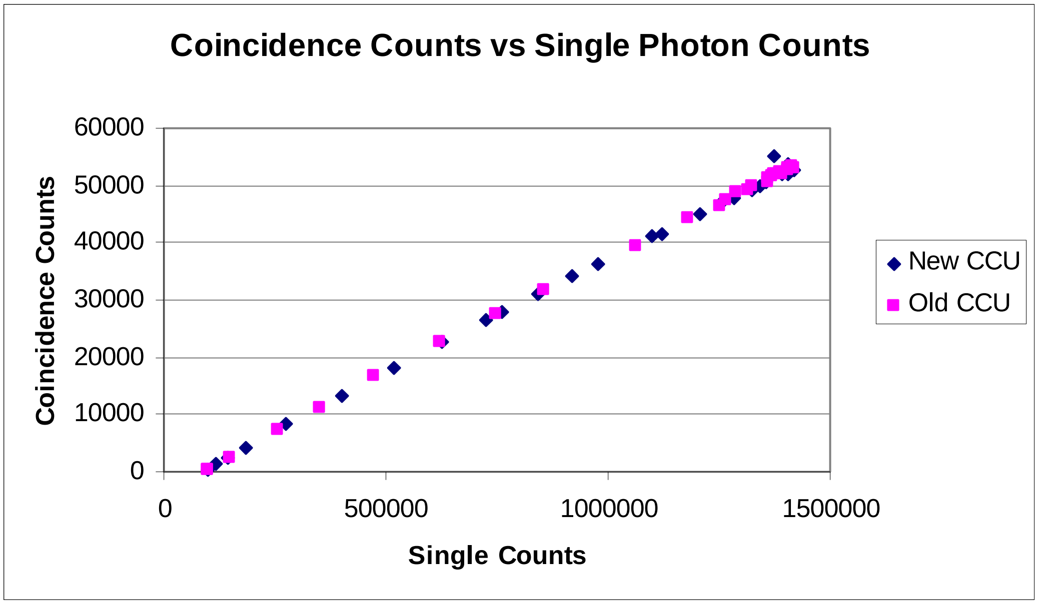


*Figure 10: The number of coincidence counts (after subtraction of accidental coincidences) vs the number of single photon counts. The old (discrete-IC-based) CCU is shown in squares and the new (DE2-based) CCU is shown in diamonds. The two agree nearly perfectly, and neither saturates for large numbers of coincidences.*

## VI. Last words

The implementation of the CCU described here uses RS232 to communicate with the computer. The DE2 board has both USB and ethernet capabilities built in, so it is also possible to update our CCU to use these faster communication protocols. We hope to do this in the future, so keep checking for info about upgrades. The nice thing about this upgrade would be that it would only involve updating the programming of the FPGA—it would not involve any hardware changes.

## Acknowledgements

We thank L. North, D. Branning, and S. Bhandari for their contributions to this project.

## References


[1] S. Bhandari, D. Branning, and M. Beck, "Low-cost coincidence-counting electronics for quantum optics," in *Coherence and Quantum Optics IX*, N. P. Bigelow, J. H. Eberly, and C. R. Stroud, eds. (Optical Society of America, Washington DC, 2008), pp. 330-331.

[2] D. Branning, S. Bhandari, and M. Beck, "Low-cost coincidence-counting electronics for undergraduate quantum optics," Am. J. Phys. (submitted, 2008).

[3] M. Beck, "Comparing measurements of $g^{(2)}(0)$ performed with different coincidence detection techniques," J. Opt. Soc. Am. B **24**, 2972-2978 (2007).

[4] D. Dehlinger and M. W. Mitchell, "Entangled photon apparatus for the undergraduate laboratory," Am. J. Phys. **70**, 898-902 (2002).

[5] D. Dehlinger and M. W. Mitchell, "Entangled photons, nonlocality, and Bell inequalities in the undergraduate laboratory," Am. J. Phys. **70**, 903-910 (2002).

[6] J. J. Thorn, M. S. Neal, V. W. Donato, G. S. Bergreen, R. E. Davies, and M. Beck, "Observing the quantum behavior of light in an undergraduate laboratory," Am. J. Phys. **72**, 1210-1219 (2004).

[7] E. J. Galvez, C. H. Holbrow, M. J. Pysher, J. W. Martin, N. Courtemanche, L. Heilig, and J. Spencer, "Interference with correlated photons: five quantum mechanics experiments for undergraduates," Am. J. Phys. **73**, 127-140 (2005).

[8] J. A. Carlson, M. D. Olmstead, and M. Beck, "Quantum mysteries tested: An experiment implementing Hardy's test of local realism," Am. J. Phys. **74**, 180-186 (2006).