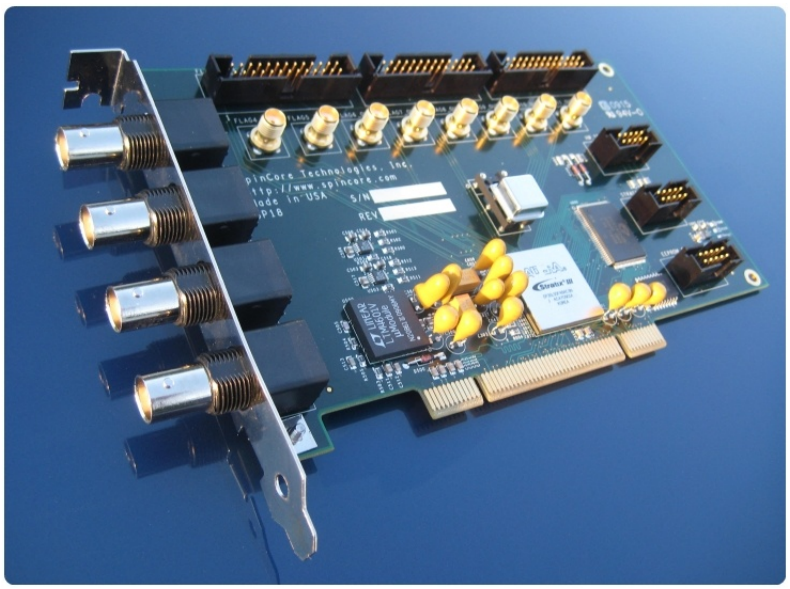
PulseBlasterESR QuadCore™ 500 (Turbo)

Owner's Manual



SpinCore Technologies, Inc.
http://www.spincore.com

Congratulations and *thank you* for choosing a design from SpinCore Technologies, Inc.

We appreciate your business!

At SpinCore we aim to fully support the needs of our customers. If you are in need of assistance, please contact us and we will strive to provide the necessary support.

Table of Contents

I. Introduction

Product Overview

The SpinCore PulseBlasterESR QuadCore 500 (Turbo) is a 4-Core PulseBlaster design implemented on a new series of PulseBlasterESR PCI boards. The 4-Core design uses four of SpinCore's proprietary PulseBlaster processor cores on a single chip. This new design allows the user to program and run completely independent programs on each core, in parallel, while maintaining precise timing synchronization between the cores.

Each individual PulseBlaster core has one output bit (flag/channel) available as a TTL signal on the corresponding BNC connector of the PC bracket. For example, the output bit for Core0 is on the BNC0 connector closest to the PCI slot as seen in Figure 1.

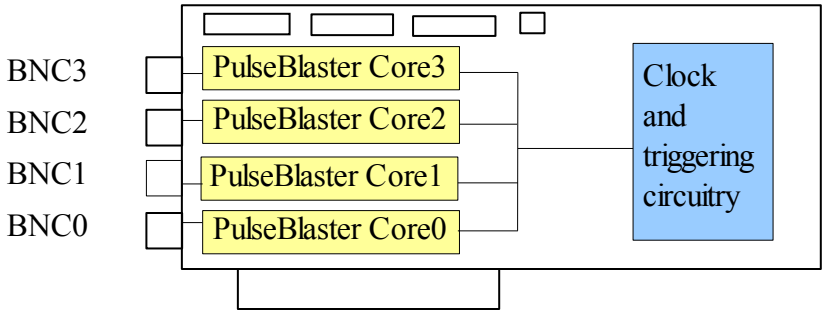


Figure 1: SpinCore PulseBlasterESR QuadCore design topology and connector locations. All four PulseBlaster cores and triggering circuitry have been implemented on a single chip.

All four cores are driven by the same single clock source at 500 MHz. They are synchronized to start at the same time and run four unique pulse programs/sequences concurrently. At 500 MHz, the available resolution of each pulse/delay/interval is 2 ns (one clock cycle), the minimum pulse/delay/interval length is 6 clock cycles, or 12 ns, and the maximum pulse/delay/interval length is 2^{27} clock cycles (~268 miliseconds). Each core has 2k (2048) memory words available for writing pulse programs, i.e., there can be up to 2048 lines in your pulse program per core.

The basic architecture of the individual PulseBlaster processor cores is described in multiple documents, including the manuals for PulseBlaster and PulseBlasterESR boards, available on-line at the SpinCore's website www.spincore.com. (Note that the PulseBlasterESR QuadCore designs use simplified PulseBlaster Cores that allow for 'continue' and 'stop' operations only.)

Programming Paradigm

Each core can be individually programmed with an arbitrary sequence of intervals. Each interval can be of unique length, and up to 2048 intervals can be accommodated per sequence. Since each interval can be a pulse or a delay, the programming of each core involves the loading of two basic parameters per interval: the output state (logical 0 or 1), and the duration of the state (in nanoseconds, microseconds, milliseconds).

Each core can be independently selected for programming and program execution. The low-level interaction is through the set of specific functions in the dedicated Application Programming Interface (API) package called SpinAPI, available for download on SpinCore's website www.spincore.com. Virtually any higher-level application package (Java, C, Matlab, LabVIEW, Visual Basic, etc.) can interact with the board through the provided SpinAPI functions.

II. Installing and Using Your PulseBlasterESR QuadCore

Installation

To install the board you must complete the following three steps:

- Install the latest SpinAPI version, available at the address http://spincore.com/support/spinapi/
- Shut down computer, insert PulseBlasterESR QuadCore Turbo 500 card, and fasten the PC bracket securely with a screw.
- Power up and follow the installation prompts.

Now you are ready to run the test programs provided in the SpinAPI package.

Note: When installing the hardware, the device may show up as PulseBlasterESR-Pro, this is OK.
Note: The PulseBlaster Interpreter that is included in the SpinAPI package CANNOT be used to program the PBESR QuadCore board.
Note: To compile and run your own C programs, you may want to download the *SpinAPI Tools* package that contains a pre-configured compiler; the *SpinAPI Tools* package is also available for download at the URL above.

General API Programming Information

Four test programs (executables and their C source files) are available for testing the boards. Assuming the default installation, the test programs will be available on the computer at the following location: Windows "Start" → All Programs → SpinAPI → PBESR_QuadCore (the default installation location is: C:\Program Files\SpinCore\SpinAPI\PBESR-QuadCore). The .c files can be modified and recompiled to create custom test programs.

Each core can be programmed with a unique pulse program by using the *pb_select_core(unsigned int core_sel)* function, where the lower four bits of *core_sel* are used to select the cores (bit0 corresponds to Core0, bit1 corresponds to Core1, etc) and multiple combinations are acceptable (i.e. a value of 0xF, or 15 will select all four cores).

Two separate SpinAPI functions are used to write pulse programs for the PulseBlasterESR QuadCore: *pb_4C_inst(int flag, double length)* and *pb_4C_stop(void)*.

pb_4C_inst(...) is used to define the pulse program. The input parameter 'flag' must either be '1' to turn the flag on, or '0' to turn the flag off. The input parameter 'length' defines the time interval for the current instruction and must be multiplied by 'ns' (nanoseconds), 'us' (microseconds), or 'ms' (milliseconds).

pb_4C_stop() defines the end of the pulse program. Note that the appropriate core stops as soon as the *pb_4C_stop()* function is seen. This means that output flag will maintain its last value on a stop. If you wish for all the output flags to be zero at the end of your program, your last instruction prior to *pb_4C_stop()* should set the output flag to zero.

III. Test Programs

Four test programs have been packaged with the SpinAPI driver suite to illustrate the basic features and functionality of the PulseBlasterESR QuadCore 500 (Turbo) design. All programs can be found following the path Windows "Start" -> All Programs -> SpinCore -> SpinAPI -> PBESR_QuadCore.

Example 1

The first test program, PB_QuadCore_Example1.exe, demonstrates that all four cores (channels) can generate identical pulses that are precisely synchronized.

An excerpt from the code to program the cores is as follows:

```
pb_select_core(0xF); // This line selects all four cores for programming, notice hexadecimal coding
   pb_start_programming (PULSE_PROGRAM);//This command is used  to begin writing the pulse program

        pb_4C_inst (1, 12.0 * ns);   // logical "high" for 12 ns (can be modified as required)
        pb_4C_inst (0, 14.0 * ns);   // logical "low" for 14 ns (can be modified as required)
        pb_4C_inst (1, 16.0 * ns);   // logical "high" for 16 ns (can be modified as required)
        pb_4C_inst (0, 18.0 * ns);   // logical "low" for 18 ns (can be modified as required)
        pb_4C_inst (1, 20.0 * ns);   // logical "high" for 20 ns (can be modified as required)
        pb_4C_inst (0, 22.0 * ns);   // logical "low" for 22 ns (can be modified as required)
        pb_4C_stop ();               // This signifies the end of a pulse program

   pb_stop_programming ();//This command is needed at the end of each pulse program
```

In Example 1, all cores are programmed with identical content. Later in the program, all four cores are triggered at the same time. The resulting output should be three pulses on each BNC output connector, with all appearing simultaneously on the four channels.

NOTE: When attaching an oscilloscope to the board to observe the pulses, care should be taken to use cables of the same type and length for each channel, as skew can be induced due to propagation delays. Conversely, any inherent variations in on-chip propagation delays can be compensated by appropriate variations in cable length.

Example 2

The second test program, PB_QuadCore_Example2.exe, demonstrates a group of simple output patterns that differ for each core but start simultaneously. This example is intended to show the simplicity of programming each individual core.

Example 3

The third test program, PB_QuadCore_Example3.exe, demonstrates the versatility and simplicity of programming of the 4-Core design. In this program, each core is loaded with a unique pulse program that allows one of the cores to start generating a pulse while other cores are in the middle of outputting their pulses. Any combination of pulses is possible as long as the minimum pulse time (12 ns) and pulse setting resolution (2 ns) are observed.

Figure 2 shows a selection of the output of example 3 captured by a digital logic analyzer. The horizontal scale is numbered in clock-cycles, where one clock-cycle equals 2.0 ns at 500 MHz. Notice the precise 1 clock-cycle resolution between pulses.

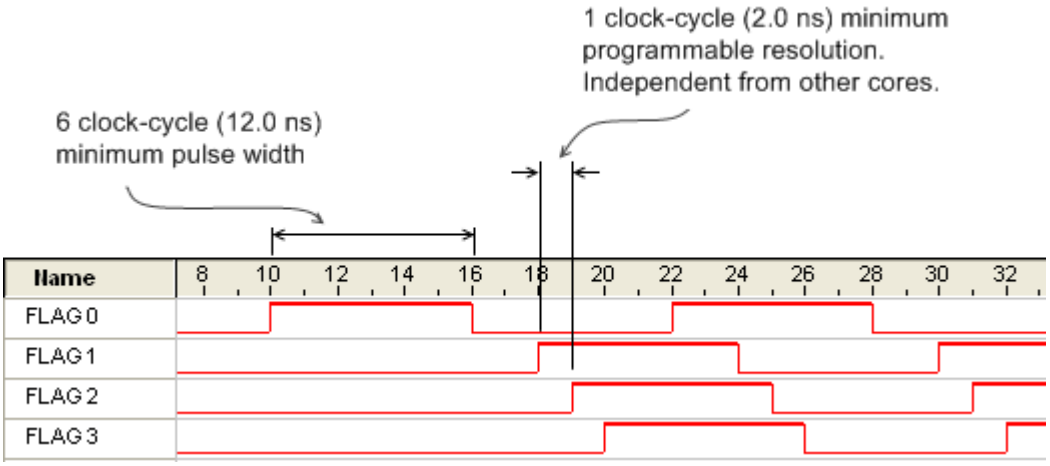


Figure 2: Output of Example 3 captured by Logic Analyzer.

IV. Available Options

A number of off-the-shelf and custom options are available for this product, including:

1. Oven Controlled Clock Oscillator, when sub-ppm stability is required.
2. Different clock frequency (custom design).
3. Additional TTL output bits per core (custom design).

Please contact SpinCore Technologies, Inc. for more information, questions, or suggestions. We look forward to hearing from you and helping you in your projects. Please find our contact information below.

V. Contact Information

SpinCore Technologies, Inc.

4631 NW 53rd Avenue, Suite 103

Gainesville, Florida 32653, USA

Phone: +1-352-271-7383

Fax: +1-352-371-8679

Website: http://www.spincore.com

VI. Document Information

Document Title:	PulseBlasterESR QuadCore 500 (Turbo)
Document Number:	DA-46
File Name:	PBESR_QuadCore_Manual
Document Location:	S:\Product_Folders\Manuals\PBESR_QuadCore
Original Document Created:	Chris Hett, 2009/05/19
Revision History:	2009/05/28 – Chris Hett – Added SP18 photo to front page 2010/07/30 – Leonid Tsakh – Fixed dates, updated address.