Managing High-Speed Clocks in High-Performance Satellite Systems

By Tim Meade, Standard Products Manager,
Aeroflex Colorado Springs

As spaceborn applications continue to demand increased bandwidth and performance in their electronic subsystems, the need to manage high-speed clock networks becomes a major factor in the success or failure of the system. Until recently, the absence of precision clock networking devices that are capable of operating in the harsh environment of outer space, has severely limited the effectiveness of these high performance systems. With the advent of the UT7R995 RadClock™, a PLL-based, eight output, programmable skewing clock buffer from Aeroflex Colorado Springs (Aeroflex), satellite systems no longer have to suffer performance limitations resulting from inadequate clock management.

Using a CompactPCI (cPCI) to SpaceWire Network Interface Card (NIC) as a practical example, this article demonstrates how the UT7R995 RadClock solves the clock management problem; regardless of the number of clocks in the network. This design example uses a pair of UT7R995 RadClocks and the 33.33MHz cPCI Bus Clock to generate and phase-align nine separate clock signals distributed at various distances across the NIC.

Application Requirements

The card plugs into a 6U (220mm x 160mm) cPCI chassis supporting the 33.33MHz/32-bit cPCI version 2.2 standard. The NIC manages data packet transmissions between the various cPCI agents and a spacecraft computer over a 133.33Mbps, full-duplex, serial bus running the SpaceWire link layer protocol. Based upon its architectural definition, the board includes five integrated circuits (ICs) requiring one or more clock sources. Table 1 lists each IC function and its corresponding clock requirements.

As Table 1 indicates, the NIC employs nine separate clocks in order to satisfy its operational requirements. The success or failure of this NIC rests largely on the quality of the clock management technique utilized in the design. This design example shows how to implement the entire clock network using only the 33.33MHz cPCI Bus Clock and two UT7R995 devices. The RadClocks ensure that each processing device receive its clocks at precisely the same time that the 33.33MHz cPCI Bus Clock reaches its destination receiver (2.5" +/-0.1" from the backplane connector) [1].

Table 1. NIC Clock Requirements

IC Function	Clock Requirements
PCI Master/Target	• 33.33 MHz PCI Clock • 66.67 MHz State Clock • 50 MHz Local Clock
SpaceWire Serial Interface	• 133.33 MHz SpaceWire Bus Clock • 50 MHz Local Clock
Local Control Hub	• 133.33 MHz State Clock • 50 MHz Local Clock
Ingress Data Flow Processor	• 25 MHz Master Clock
Egress Data Flow Processor	• 25 MHz Master Clock

Designing the Clock Network

The clock network design process breaks down into four simple steps:

1. Define the clock distribution scheme.
2. Enter the clock distribution scheme into the design schematic.
3. Floor plan the board content and hand route critical signals.
4. Perform signal integrity analysis on the post-routed clock signals.

"UT7R995 RadClock™ solves the clock management problem, regardless of the number of clocks in the network."

Prior to defining the clock distribution scheme, the designer must have a strong understanding of the RadClock's features and functions, as they are foundational to the successful clock network implementation. The following guidelines, which are derived from the RadClock's datasheet (http://www.aeroflex.com/RadClock), provide the designer with useful tips that, if considered throughout the design process, will significantly improve the likelihood of achieving "first-pass" design success.

- Banks 1 and 2 of the RadClock run at the same rate as its phase-locked loop (PLL)
- Bank 3 on the RadClock can operate at ¼, ½, or exactly the same rate as the PLL
- Bank 4 will supply an output clock that is ½ or exactly the same frequency as the PLL
- The reference clock input may be used directly or divided in half before driving the PLL

- Each output bank that is operating at the same rate as the PLL has independent skew control.
- Optimal designs will drive one receiver per RadClock output.
- Provisions to perform post-assembly configuration changes to the RadClock should be included.
- Whenever possible, use the slowest output clock to drive the RadClock's Feedback input.

Defining the Clock Distribution Scheme

After factoring the aforementioned guidelines into the design application requirements, the designer proceeds to define the clock distribution scheme. The first requirement to satisfy is the cPCI v2.2 demand for the 33.33MHz cPCI Bus Clock to drive exactly one load residing 2.5 inches from the backplane connector. This implies that the 33.33MHz cPCI Bus Clock cannot drive both the PCI Master/Target device and the RadClock. Since we plan to generate all clocks on the NIC from the RadClock, the 33.33MHz cPCI Bus Clock will necessarily drive the reference input on the first RadClock (RC1). The responsibility to supply an exact timing replica of the incoming cPCI Bus Clock to the PCI Master/Target device falls upon the RadClock. How to accomplish this clock-replication is explained later in the article, but first, we need to decide which clocks RC1 will be responsible for distributing throughout the NIC.

By configuring RC1 to multiply the incoming 33.33MHz cPCI Bus Clock by four, its PLL operates at 133.33MHz. Considering the guidelines mentioned previously, banks 1 and 2 of RC1 supply four output clocks operating at the nominal PLL frequency of 133.33MHz. Enabling bank 3 to operate at ¼ the PLL rate permits RC1 to supply two clocks running at 33.33MHz. Finally, configuring bank 4 to operate at ½ the PLL rate results in the generation of two 66.67MHz output clocks. Based on this configuration, RC1 will source the following clocks to the various devices on the NIC:

- RC1 – Bank 1Q0 drives the 133.33MHz serial bus clock on the SpaceWire Interface
- RC1 – Bank 1Q1 is not used
- RC1 – Bank 2Q0 drives the 133.33MHz state clock on the Local Control Hub
- RC1 – Bank 2Q1 is not used
- RC1 – Bank 3Q0 drives the 33.33MHz Reference clock on the second RadClock (RC2)

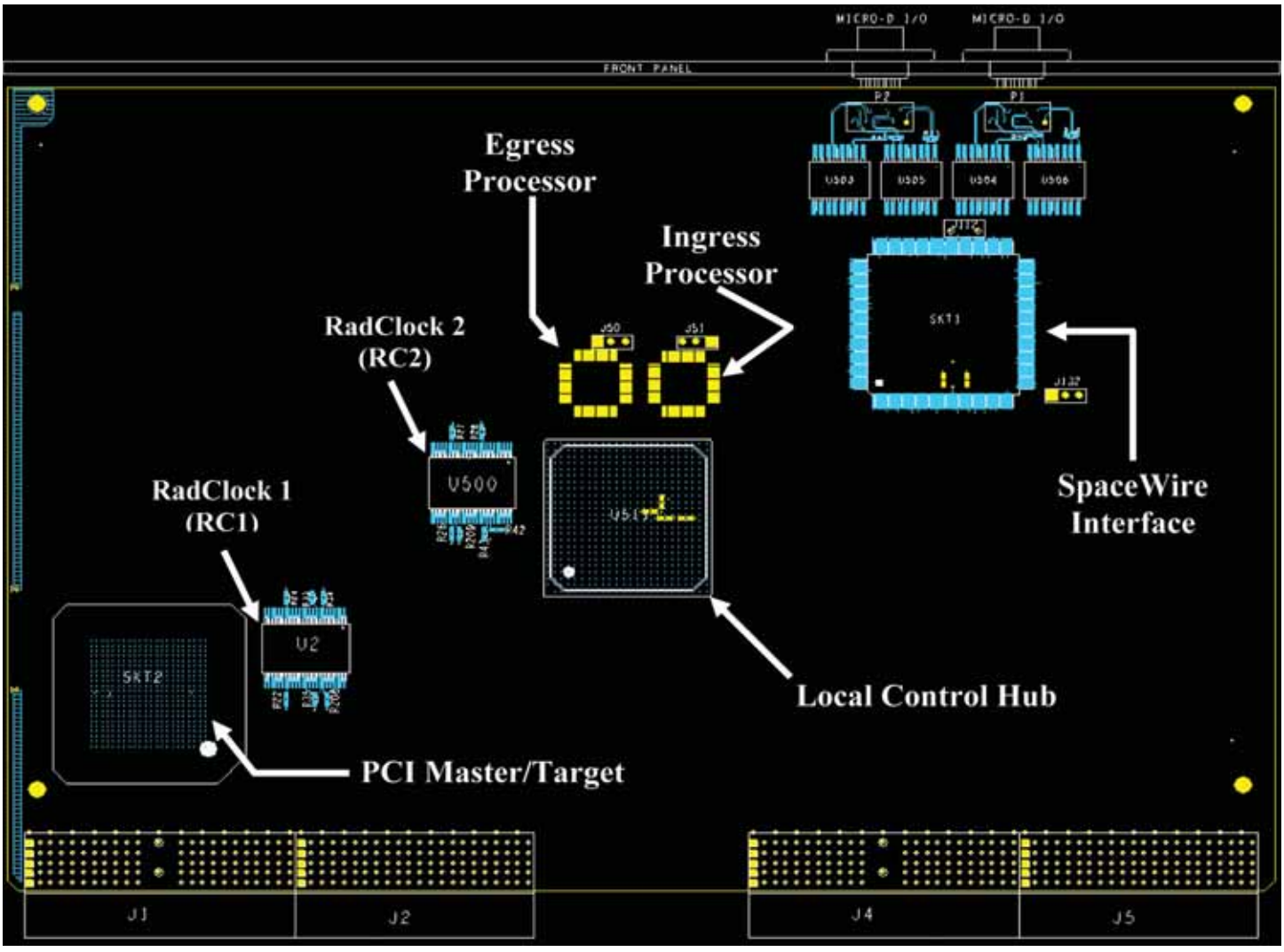

Figure 1: cPCI-to-SpaceWire NIC Component Placement

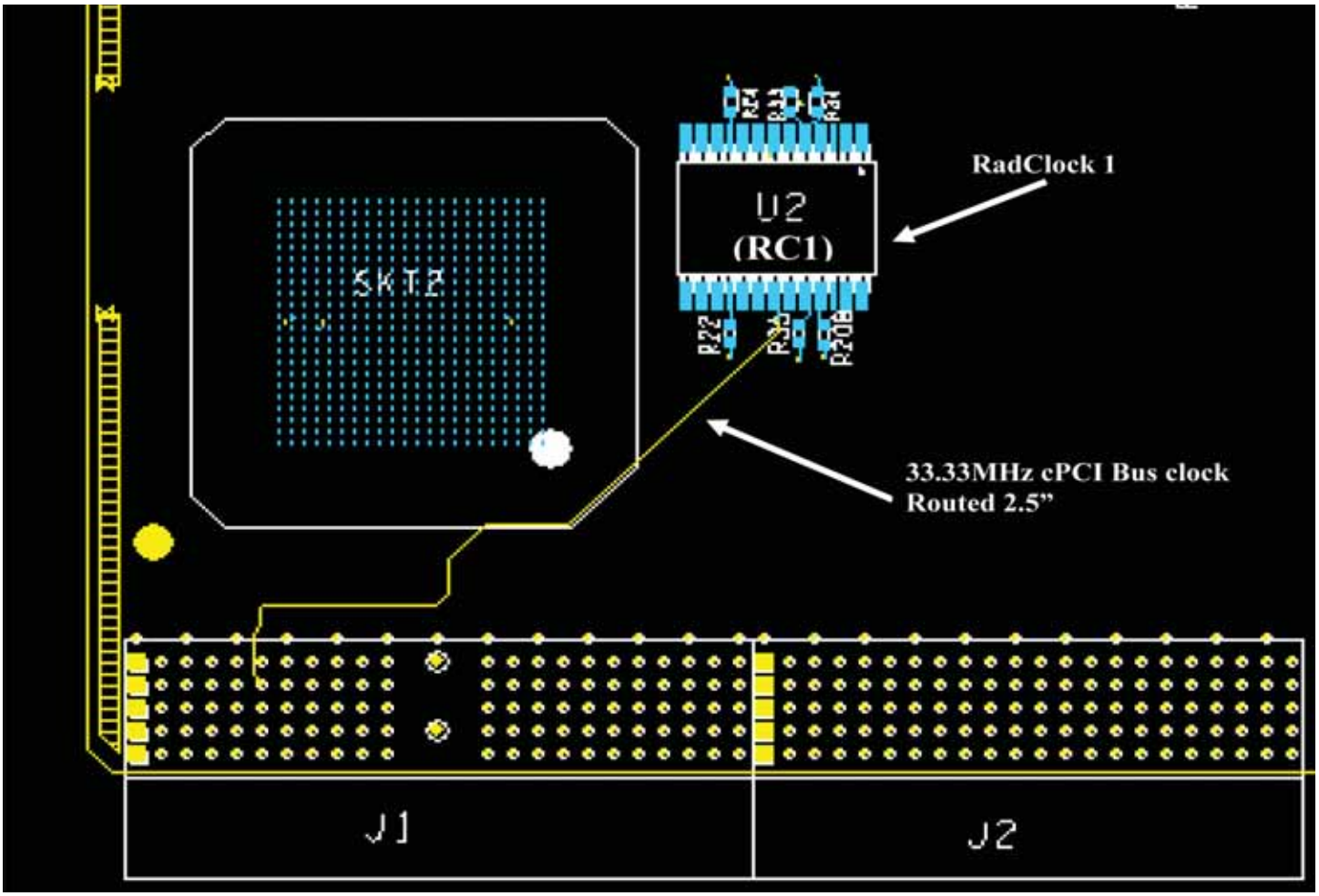

Figure 2: Routing for 33.33MHz cPCI Bus Clock

- RC1 – Bank 3Q1 drives the 33.33MHz PCI clock on the PCI Master/Target device along with the Feedback input on RC1
- RC1 – Bank 4Q0 drives the 66.67MHz state clock on the PCI Master/Target device
- RC1 – Bank 4Q1 is not used

Although optimal designs drive one receiver per RadClock output (see the guidelines in the previous section), the need to ensure the divided 33.33MHz output bank is completely in phase with the 33.33MHz cPCI Bus Clock takes priority. Therefore the tradeoff is made, on RC1, to drive its feedback input as well as the 33.33MHz PCI Clock on the PCI Master Target device using one of the outputs on bank 3; thereby dedicating the remaining bank 3 output on RC1 to drive a 33.33MHz clock to the reference clock input on the second RadClock (RC2) in the network.

With a 33.33MHz reference clock, RC2 is configured to supply the 50MHz local and 25MHz processor clocks to the various components on the NIC. These frequencies are generated on RC2 by first dividing the 33.33MHz reference clock received from RC1, and then multiplying the input clock by a factor of three. Thus, the PLL will operate at a rate of 50MHz and the output distribution scheme for RC2 is defined using a similar approach to the one previously described for RC1.

Based on the PLL running at 50MHz, RC2 satisfies the remaining clock requirements for the NIC (see Table 1) by using three of its output banks to supply the 50MHz local clocks to the various devices on the board and configuring bank 4 to provide the 25MHz processor clocks. Because the 50MHz and 25MHz clocks generated by RC2 are uneven multiples of the 33.33MHz reference clock, 1.5x and 0.75x respectively, it is not possible to connect the lowest speed output clock to the feedback input on RC2 and still maintain the desired PLL rate of 50MHz. Therefore, the resulting clock distribution scheme for RC2 follows:

- RC2 – Bank 1Q0 drives the 50MHz local clock to the Local Control Hub
- RC2 – Bank 1Q1 drives a 50MHz clock to the Feedback input on RC2
- RC2 – Bank 2Q0 drives the 50MHz local clock on the PCI Master/Target device
- RC2 – Bank 2Q1 is not used
- RC2 – Bank 3Q0 drives the 50MHz local clock on the SpaceWire Interface
- RC2 – Bank 3Q1 is not used
- RC2 – Bank 4Q0 drives the 25MHz master clock on the Ingress Data Flow Processor
- RC2 – Bank 4Q1 drives the 25MHz master clock on the Egress Data Flow Processor

With the clock frequencies defined and the source/destination relationships identified, architecting the clock distribution scheme is complete. The next step is to translate the clock network distribution architecture into the design schematic.

Clock Network Design Schematic Entry

Translating the clock distribution scheme into the design schematic is a straightforward process requiring minimal guidance; however, as it pertains to the implementation of the clock network, this design example focuses on some subtle, but important, aspects of the schematic entry process. Considering these issues throughout the schematic entry portion of the design maximizes the probability of design success while minimizing the iterations through the cyclical process associated with converting the schematic design into a physical design.

Firstly, the designer should avoid locking the RadClock's into their expected configurations as suggested by the clock distribution scheme. Instead, the final configuration should be defined after layout and, preferably, at the hardware level. Finalizing the RadClock configuration at the end of the development process affords the designer maximum flexibility during layout and provides an opportunity to adjust clock edges after the board has been assembled.

Table 2. Frequency Range and MF Selection

MF	Nominal PLL Frequency Range (f_{NOM})
32	24 to 50 MHz
16	48 to 100 MHz
8	96 to 200 MHz

For prototype and proof of concept designs, an eight-position, single-pole-triple-throw (SP3T) DIP switch is useful in supporting "real-time" configuration changes in hardware. Engineering and flight models, however, require a permanent or "hands-free" method to set the RadClock configuration. Permanent configurations can be implemented by installing provisional bias resistors during assembly or by interfacing the RadClock configuration pins to an array of tri-stateable outputs on an FPGA or microcontroller. Because the RadClock uses ternary (three-level) instead of binary (two-level) inputs, the ability to float the configuration inputs, in addition to driving them high or low, is necessary to achieving all configuration options offered by the RadClock.

Regardless of the configuration flexibility envisioned for each RadClock, enter the solution into the schematic along with signal termination for each output clock, which brings us to our second point. Although signal integrity analysis is most applicable to a design following board layout, performing IBIS or spice simulations during the schematic entry phase of the design helps in choosing a termination technique and impedance values that best suit the application. IBIS models for the RadClock may be freely downloaded at the Aeroflex website (www.aeroflex.com/RadClock).

Once the schematic level design is complete, and all necessary simulations performed, it is time to proceed with the layout portion of the design. Keep in mind that the schematic entry and layout are integral and cyclical processes to one another. After components have been placed and signals routed, some changes to the schematic may be necessary. Most notably, the reconfiguration of the RadClock may be necessary if sufficient flexibility is not already designed into the post assembly version of the NIC.

Layout and Routing the Clock Network

The designer begins the layout process by porting the netlist containing the entire NIC design, including the clock network, into the layout tool of choice. Assuming that all component symbols are available in the layout tool's library, the board form factor and layer stack-up are defined; then the imported netlist should directly translate into the working canvas with all nets connected to corresponding device pins.

The layout continues by first placing all components into a logical location on the board. The goal of component placement

$$\text{Equation 1.}\quad t_U = \frac{1}{(f_{NOM} * MF)}$$

$$\text{Equation 2.}\quad L_{1t_U} = \frac{t_U}{\text{trace prop. delay}}$$

is to situate each device onto the board in such a way that the majority of signals have short, direct paths between their source and destination devices. Special care should be given to the placement of devices that use differential signaling or those that drive signals with particularly fast edge rates (e.g. less than 2ns). Figure 1 shows the top-level component placement of the devices used on this cPCI-to-SpaceWire NIC.

Once all components have been placed on the board design canvas and rotated to maximize routing channels, it is time to start the hand routing of critical signals. Beginning with the cPCI requirement that the 33.33MHz cPCI Bus Clock travel 2.5 inches from the backplane, then it makes sense that RadClock RC1 will be placed near enough to cPCI connector J1 to ensure the clock can be routed the required distance (see Figure 2).

After routing the cPCI Bus Clock a distance of 2.5 inches from the backplane connector, we continue to hand-route the rest of the clock signals. However, before routing the remainder of the clock network, the designer must understand how the RadClock's PLL, in conjunction with its programmable skew control and the feedback trace length ensures that all clocks on the NIC are phase aligned. This phenomenon is achievable because the PLL preemptively drives its output clock such that it returns to the feedback input directly in phase with the reference clock. Therefore, by matching the length of the feedback trace (including its distance through the termination resistor) on RC1 to the 2.5" cPCI Bus Clock trace, the designer is assured that all outputs from the RadClock naturally depart the device at the exact same time the 33.33MHz cPCI Bus Clock enters the board through the backplane connector J1. Then, by applying this same logic, the designer can anticipate that all outputs from the RadClock, which connect

to their destination receivers on 2.5" traces, arrive at the same time the 33.33MHz cPCI Bus clock reaches the reference input on RC1. If a clock requires a trace longer than 2.5" to reach its destination, then the designer leverages the programmable skew feature on the RadClock to compensate for the additional trace delay.

To ensure that the programmable skew control successfully "zeros-out" trace delay, the designer first calculates the resolution of the RadClock's skew control and correlates this resolution with the electrical propagation delay expected for the signal routing. The programmable skew resolution, or time unit (t_U), is determined from Equation 1 by calculating the inverse of the nominal PLL operating frequency multiplied by the constant of proportionality (MF) listed in Table 2. Once the designer determines the skew resolution for each RadClock, correlating the value to a specific trace length is a simple conversion process of dividing the calculated time unit by the electrical propagation delay of the trace, as shown in Equation 2.

Applying this analysis to the application at hand, where the PLL on RC1 operates at 133.33MHz and the corresponding MF value taken from Table 2 is eight, the resulting time unit for RC1 is 1/(133.33e6 * 8) = 938ps. Solving Equation 2 with a given trace delay of 180ps/inch for the clock signals routed on an inner trace layer of this FR-4 PWB, the designer finds that a time unit of 938ps translates to a trace length of 5.2". Using the same process on the second RadClock (RC2), whose PLL is configured to operate at 50MHz, the designer calculates a time unit of 625ps (t_U=1/(50e6 * 32)) correlating to 3.5" of routing on an internal trace whose propagation delay is 180ps/inch.

After correlating the trace lengths with the associated skewing time unit for each RadClock, the designer can route the remainder of the clock network by first routing all clocks that can reach their destination receivers on 2.5" traces. Then the designer routes the remaining clock traces so they traverse a length of 2.5" plus an integer multiple in trace length corresponding to the skewing time unit of the driving RadClock. Lastly, using the RadClock's programmable skew control, the designer configures the RadClock outputs to lead the feedback clock by the number of time units necessary to "zero-out" the additional trace delays. Table 3 shows a summary of the clock routing associated with the cPCI-to-SpaceWire NIC demonstrated in this article. The physical routing of each clock is depicted in Figure 3.

Once the board is completely routed, the designer should perform a final signal integrity analysis of the clock network using the extracted models of the NIC's physical implementation. Post-layout signal integrity analysis establishes confidence in clock quality while simultaneously helping to identify cross-talk problems that may occur between the clocks and nearby signal traces. Signal integrity analysis only helps identify problems; it does not prevent them from occurring. To prevent signal integrity problems from crippling a design, a number of high-speed layout techniques should be employed by the designer. The following section of this article presents some helpful design and layout considerations that will maximize the likelihood that the design will successfully pass the post-layout signal integrity analysis and perform as planned in the end system.

Table 3. Clock Routing Summary

Clock Source	Clock Destination	Frequency (MHz)	Trace Length (inches)	Associated Time Unit (t_U)	RadClock Skew Setting
Backplane (J1)	RC1 – Reference	33.33	2.5"	N/A	N/A
RC1 – Bank 1Q0	SpaceWire Interface	133.33	12.9"	938 ps	-2 t_U
RC1 – Bank 2Q0	Local Control Hub	133.33	7.7"	938 ps	-1 t_U
RC1 – Bank 3Q0	RC2 – Reference	33.33	1.7"	938 ps	0 t_U
RC1 – Bank 3Q1	PCI Master/Target	33.33	2.5"	938 ps	0 t_U
RC1 – Bank 3Q1	RC1 – Feedback	33.33	2.5"	938 ps	0 t_U
RC1 – Bank 4Q0	PCI Master/Target	66.67	2.5"	938 ps	0 t_U
RC2 – Bank 1Q0	Local Control Hub	50	2.5"	625 ps	0 t_U
RC2 – Bank 1Q1	RC2 – Feedback	50	1.7"	625 ps	0 t_U
RC2 – Bank 2Q0	PCI Master/Target	50	6"	625 ps	-1 t_U
RC2 – Bank 3Q0	SpaceWire Interface	50	9.5"	625 ps	-2 t_U
RC2 – Bank 4Q0	Ingress Data Flow Processor	25	2.5"	625 ps	0 t_U
RC2 – Bank 4Q1	Egress Data Flow Processor	25	2.5"	625 ps	0 t_U

Maximizing Signal Integrity

Throughout the schematic entry and layout portions of this NIC design, consideration was given to the signal integrity of all high-speed signals (e.g. signals with edge rates < 2ns). To begin with, any trace having a propagation delay greater than one-third the signal's edge rate was terminated. For example, an internal trace within this FR-4 PWB having an edge rate of 2ns would require termination if the corresponding trace length (L) is more than 3.7 inches (e.g., L = (1/3 * 2ns) / 180ps per inch). Applying this analysis to the RadClock, whose output edge rates are 500ps on average, it

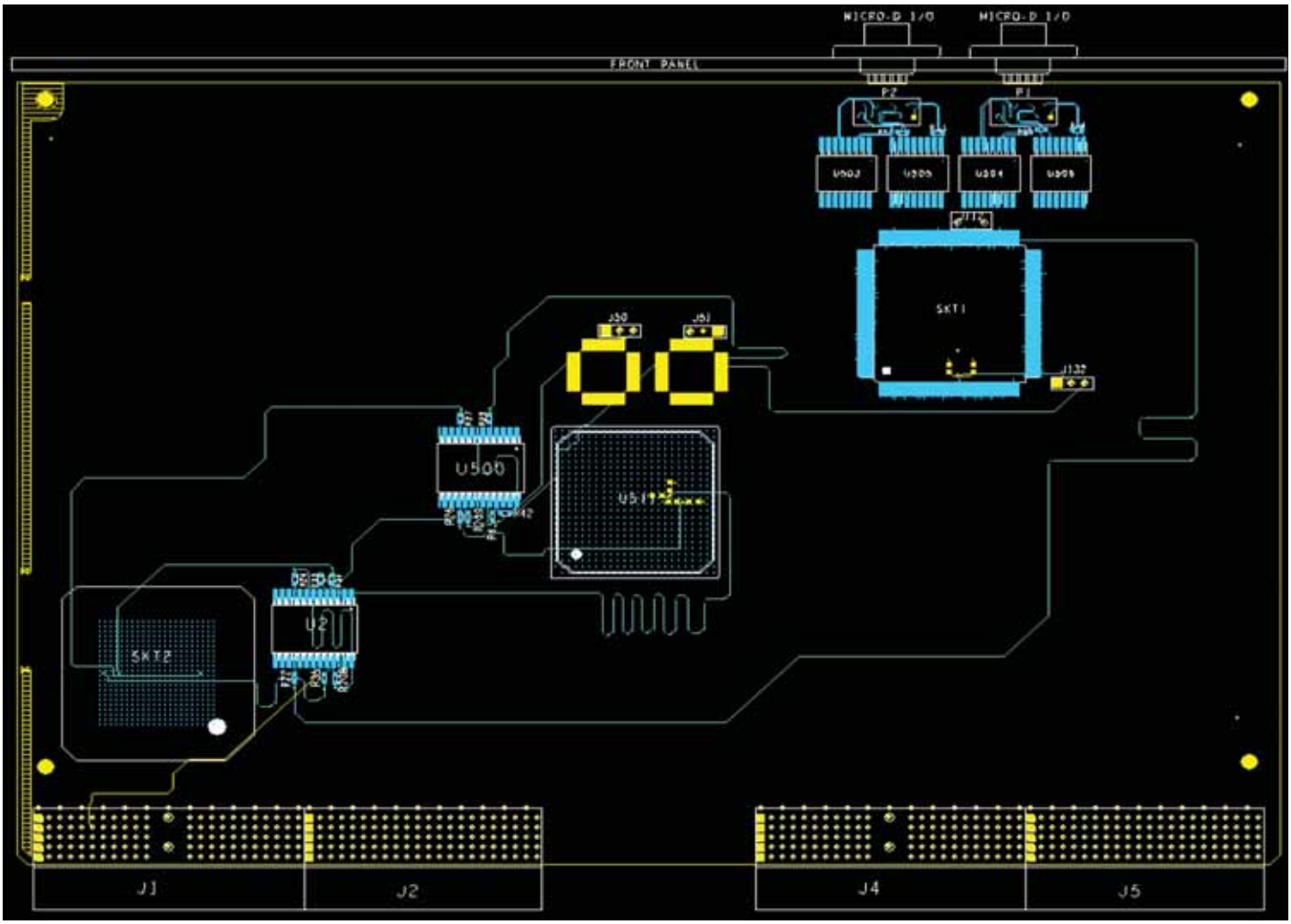

Figure 3: Final Clock Routing

is determined that any trace longer than 0.9 inches must be terminated. Referencing Table 3, the shortest RadClock output trace is 1.7 inches. Therefore, all RadClock outputs on this NIC are terminated.

In addition to simply terminating the critical and high-speed signals, like the RadClock outputs, particular care was taken in routing the traces themselves. The objective when routing a high-speed signal trace is to minimize any variation in the desired characteristic impedance of the trace. Unlike the minimal impact that common routing tools like vias and "serpentine" traces exert on slow signals, their impact, if not implemented properly, can significantly affect high-speed signals.

Although vias are an essential tool in any layout, they present local impedance minima and are inherently capacitive. Because the via (a.k.a. plated through-hole) has less impedance than the signal trace, a negative reflection propagates back to the source when the signal encounters a via [2]. Additionally, the excessive capacitance associated with a via causes signal loss by attenuating high-frequency harmonics from the signal [3]. Signal losses are particularly exacerbated if a via connects inner trace layers and leaves a "stub" extending to the surface of the board.

In general, to minimize the effects of vias on high-speed signals, the designer should group all vias as close to the driver and/or receiver as possible. This creates "lumped" impedances at the two end points of the net. If plated through-holes are used to connect inner trace layers of very high-speed signals, then buried vias, blind vias, or back-drilled vias are potential candidates. However, from a manufacturing and reliability standpoint, back-drilling is the most promising solution to removing via stubs.

In addition to the judicial use of plated through-holes, the designer must be aware of the effects that signal routing and board stack-up have on the integrity of high-speed signals. Firstly, all bends in the trace should be less than 45° or, better yet, rounded. The reason that traces should be rounded as much as possible is to maintain a constant characteristic impedance along the entire length of trace. A 90° bend, for example, has 40% more trace width at the bend. This additional trace width reduces the characteristic impedance and therefore propagates a negative reflection back to the signal source [4].

Secondly, sandwiching high-speed traces between power and ground planes will help to establish more controllable characteristic impedances, ensures more

consistent propagation delays, reduces the potential for cross-talk, and limits electro-magnetic emissions from the PCB. If high-speed signals are routed with signal planes above and/or below the high-speed trace, the traces should be routed perpendicular to each other in order to minimize cross-talk between the traces. To mitigate lateral cross-talk between signal traces, a ground-shielded guard-trace is an effective solution. Additionally, using conservative spacing rules like the "three-width (3W)" rule will help minimize conducted noise between signal traces [5].

Lastly, performing post layout signal integrity simulations using IBIS or spice models of the critical nets on the board is a must. These post-layout simulations help identify potential areas of concern and reduce the likelihood that a signal integrity issue will arise to cause a functional problem in the hardware. After the board has been completely routed and relevant signal integrity analysis has been performed, it is time to release the gerbers and proceed with circuit card assembly and board checkout. If everything goes as planned, we will have a board where all devices receive clocks that appear to travel exactly 2.5" from the backplane connector, thus creating a comprehensive clock network that is completely phase aligned and synchronized to the 33.33MHz cPCI Bus Clock while complying with the required timing of the cPCI v2.2 bus standard.

The Proof is in the Pudding

Aeroflex Colorado Springs put theory to practice by taking the cPCI-to-SpaceWire NIC discussed in this article from the virtual world of electronic design automation to the lab. After completing the layout, fabricating the PWB, and assembling the NIC, Aeroflex tested the hardware by exercising the assembly through full-functional operation in a real application environment and analyzed all clocks with a 7GHz (20GS/s) Tektronix TDS7704B Digital Phosphor Oscilloscope. Using Tektronix P7260 (6GHz bandwidth) probes, Aeroflex measured all clocks at their respective destination receivers while referencing them to the 33.33MHz cPCI Bus Clock located at the reference input of RadClock RC1. As indicated by the composite summary of clock measurements shown in Figure 4, all clocks on the NIC arrive at their respective destination receivers in phase with the 33.33MHz cPCI Bus Clock located at the reference input on RC1; thus demonstrating that the UT7R995 RadClock is an ideal solution to the clock management problem in high-performance satellite systems!

Distributed Clocks Measured at their Destination Receivers

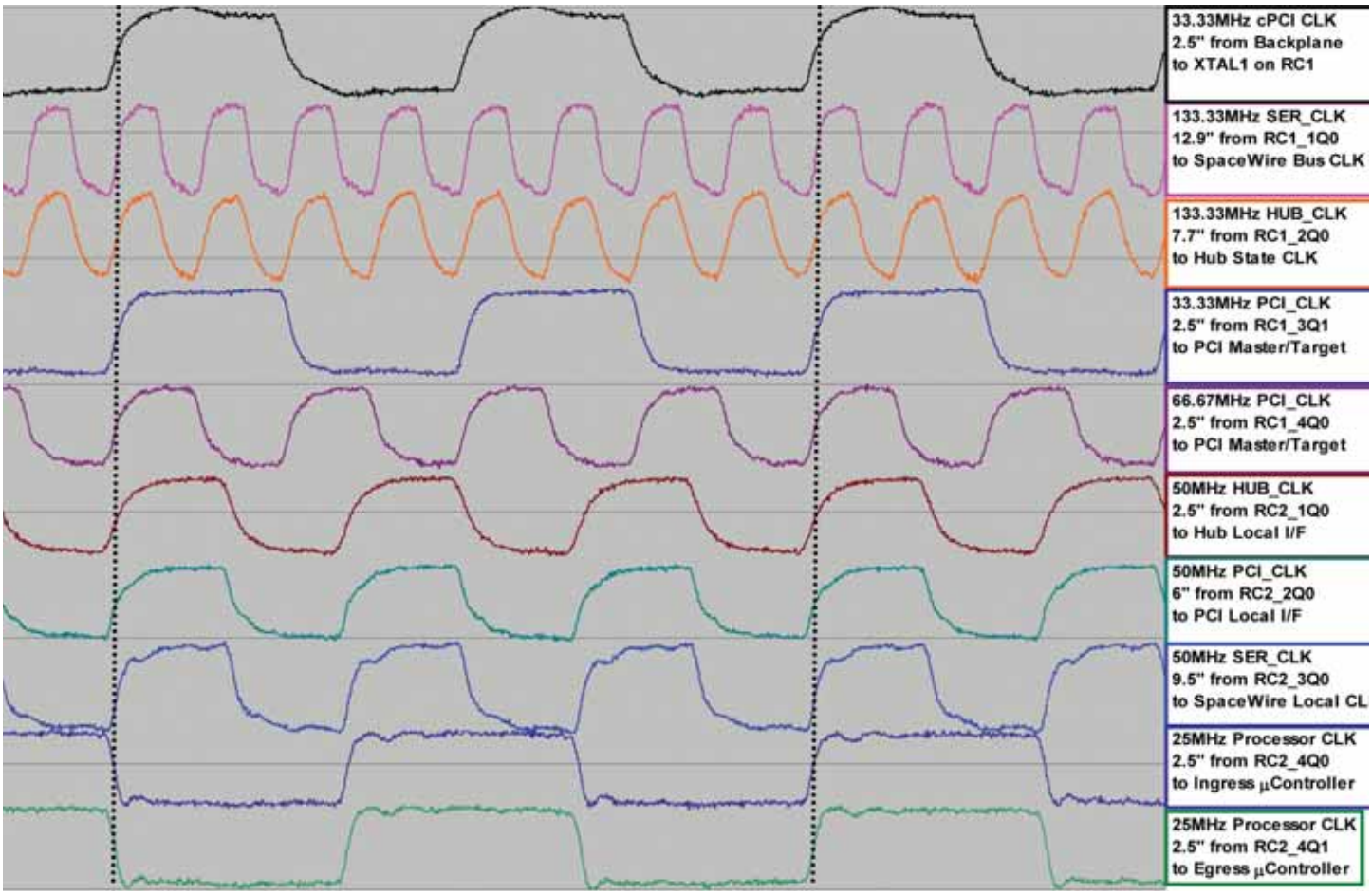

Figure 4: Oscilloscope Measurements of All Network Clocks on the NIC

References

[1] PICMG CompactPCI® Core Specification, Version 2.0, Revision 3.0, 10/1/1999, p.19

[2] Douglas Brooks, "The Effects of Vias on PCB Traces," Xtera Design, 10/20/2004, p.2-3

[3] Karl Dietz, "Fine Lines in High Yield (Part XCVI): Back-drilling of Plated Throughholes on Backplanes," CIRCUITREE, September 2003, p.16

[4] Dr. Eric Bogatin, "Riddle Me This," PRINTED CIRCUIT DESIGN & MANUFACTURE, January 2006, p.28

[5] Zulki Khan, "Key Design Techniques Reduce EMI/RFI Effect in Military Applications," COTS Journal, December 2005, p.43-44

Tim Meade
Standard Products Manager
Aeroflex Colorado Springs
(800) 645-8862
www.aeroflex.com/RadClock