Connecting the RC32355 to an SDRAM System

RC32355 Application Note AN-302

Notes

By Paul Snell

Revision History

June 8, 2001: Initial publication.

July 2, 2001: Revised Figure 2, Clock Buffer Connections.

Background

Revision ZB of The RC32355 does not meet the hold time required by PC100 and PC133 SDRAMs, SoDIMMs, and DIMMs, as listed in the RC32355 Device Errata sheet (item #8), when SYSCLKP (output clock from the RC32355) is used to drive the SDRAMs. This application note describes how to best connect SDRAMs to the RC32355 revision ZB to avoid the hold time issue. This solution also enables an easy migration to future revisions of the part that will have the hold time issue fixed. Finally, this application note also gives recommendations on how to terminate the SDRAM clock line (SYSCLKP) and where to best place the bypass capacitors around the supply pins for the SYSCLKP driver.

Programmable Skew Clock Driver

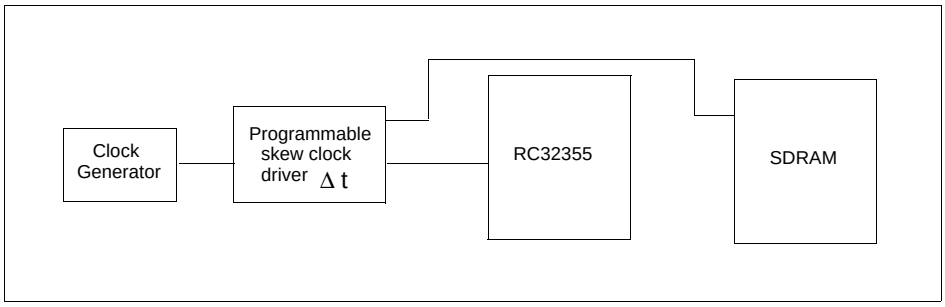

Figure 1 System Block Diagram with Programmable Skew Driver

For all systems based on the RC32355 revision ZB, it is recommended not to use the SDRAM output clock (SYSCLKP) from the RC32355, rather an externally generated clock must be used to drive the SDRAM system. This clock can be generated in two ways:

1. Use a programmable clock skew buffer to send a delayed version of the RC32355 masterclock, CLKP, to the SDRAMs.
2. Use an EPLD to generate a new clock for the SDRAM. In this case, the designer would most likely have to invert the clock normally available in the EPLD and then hardcode delays into the EPLD to fine-tune the SDRAM hold times.

This document focuses on the first solution.

IDT recommends the use of a programmable clock skew driver as shown in Figure 1. If the system input clock to the RC32355, CLKP, is considered to have zero skew, the buffer will induce some amount of skew to the clock for the SDRAM beyond that of the RC32355's system clock, CLKP. This newly derived SDRAM clock will be ahead of the RC32355's SDRAM output clock, SYSCLKP, yet behind the zero skew system clock, CLKP driven to the RC32355.

All of the recommendations below assume that the SDRAM clock line length as determined by the distance from the clock buffer to the SDRAM is approximately equal to the distance from the buffer to the RC32355 + the distance from the RC32355 to the SDRAM. For example, if there is four inches of trace from the buffer to the SDRAM, then we would expect 1" of trace from the buffer to the RC32355 plus another 3"

Notes

of trace from the RC32355 to the SDRAM. Or alternatively, 2" of trace from the buffer to the RC32355, plus another 2" of trace from the RC32355 to the SDRAM. Any additional trace length beyond this will add skew at approximately 185ps per inch of extra trace.

The return clock (SYSCLKINP) trace length from the SDRAM clock back to the RC32355 input clock is not critical. The skew from the buffer alone is sufficient to guarantee functionality. The further skew added by tapping the return clock off at the SDRAM and routing it back has no effect on the SDRAM hold time, and only adds further to the already generous margins for the RC32355 hold time and internal clock domain buffering (discussed below).

One of the many available buffers which will work well in this application is the IDT5V993A-2 programmable skew clock driver. It comes in a 28 pin QSOP. The recommended wiring for this part in this application is shown in figure 2. The clock driver output for the SDRAM is set for one time unit of skew, while the system clock for the CPU is programmed with no skew. At 50MHz, this single time unit of skew is 1.25ns. At 75 MHz, the skew drops to 0.83ns. At 1.25 ns, there will be 1.1 ns of hold time at the SDRAM. At 0.83ns, the hold time will go up to 1.57ns.

The optimal skew value is 1.0ns. This skew value was derived by carefully balancing three constraints: the hold time at the SDRAM, the hold time back to the CPU, and the margin between the input clock, SYSCLKP and the system clock, CLKP, clock domains.

This fix will work with both ZB and future revisions of silicon, as it in no way depends on the timing of the output clock (SYSCLKP). However, with the routing suggested at the end of this note, it will be possible to depopulate the clock buffer on the board using future revisions of silicon that have this issue resolved, and replace it with two zero ohm resistors, thereby saving the cost of the buffer on these production boards.

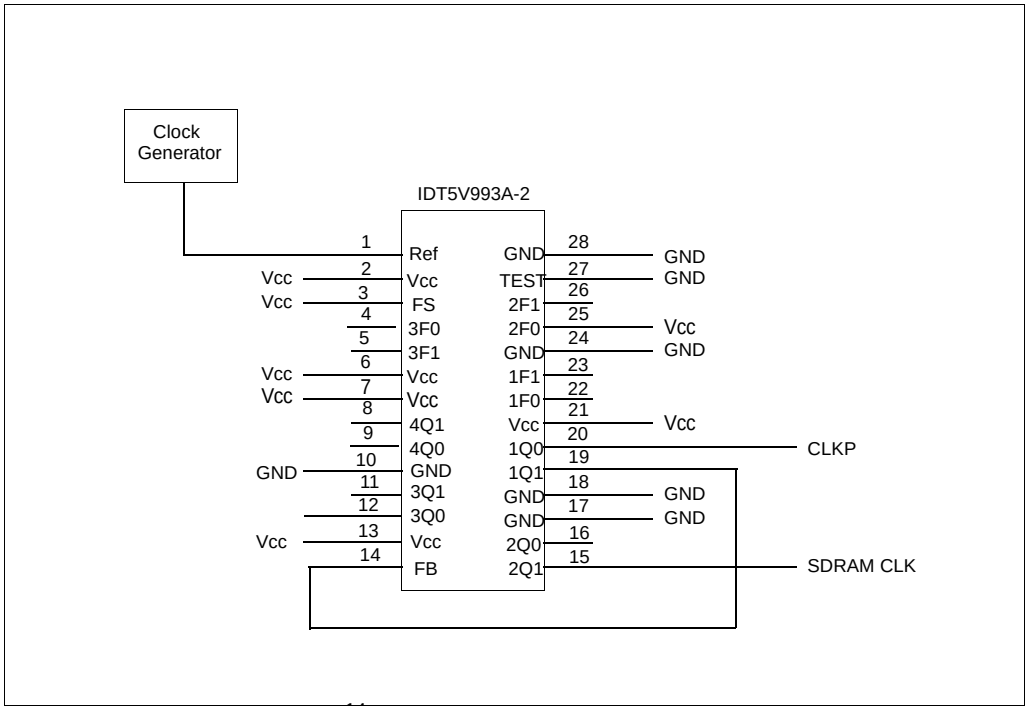

Figure 2 Clock Buffer Connections

The routing method depicted in Figure 3 features two sets of resistor pads. There is one optional resistor located under the body of the 993A clock buffer. There is a second resistor, which is not optional, that can be stuffed such that the SDRAM either derives its clock from the 993A buffer, or is driven off of the RC32355 SDRAM output clock SYSCLKP. Once a new revision of silicon with this issue resolved is available, it will be possible to depopulate the 993A, stuff the resistor under it, and move the other resistor to its alternate location.

Notes

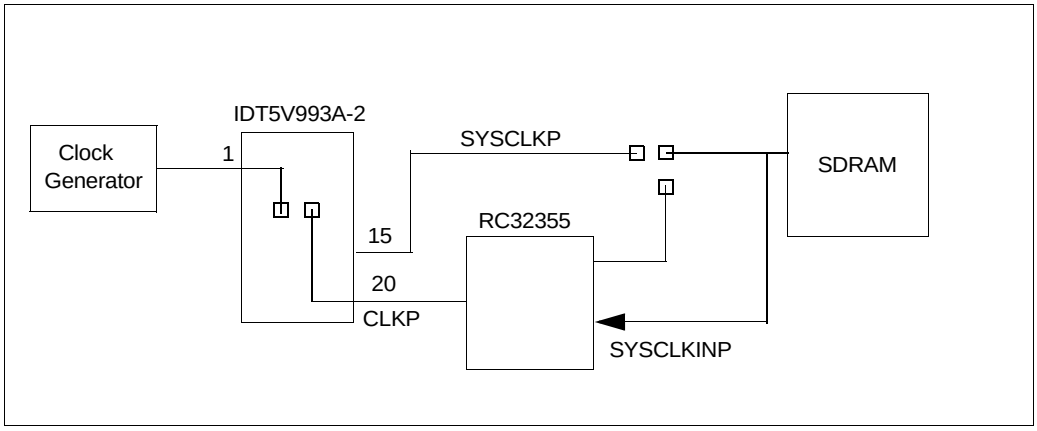

Figure 3 Suggested Routing

SDRAM Clock Recommendation for Termination

The proper termination for this clock consists of a series resistor located near the SYSCLKP driver pin (or at the output of the 993A buffer for boards that use clock buffers), as well as a parallel termination located at or near the point where SYSCKLINP enters the RC32355. The recommended value for the series resistor is 33 ohms. This value is selected assuming a board trace resistance of 50 ohms. If this value is larger, then the correct resistor value will be the actual trace resistance - 50 ohms + 33 ohms.

The recommended value for the parallel termination is 20pF in series with 50 ohms. The parallel termination resistor value should match the board trace resistance, typically between 50 and 65 ohms.

These are only general recommendations. The correct values for both the capacitor and resistor are board-specific. The actual capacitive values for the parallel termination may need to be tuned on the final board to produce the best possible clock waveform at the SDRAM chips.

Bypass Capacitors

The RC32355 requires adequate bypass capacitance to compensate for localized power fluctuations. The exact capacitor values will depend on the frequency content of the power supply noise on the board. But in general, values of 0.1uF work well. A good rule of thumb is to place four capacitors on each side of the chip. One at each corner, and one each a third of the way along the chip.

It is also recommended to add capacitive bypassing specifically for the clock driver (SYSCLKP). This bypassing consists of one 0.1uF capacitor which is intended to bridge the Vcc IO and Vss pin combination that feeds the SYSCLKP driver. In the PQFP package, SYSCLKP is pin 183. In this case, the intent is to bridge between pin 184, Vss, and pin 180, Vcc IO. This will maximize the drive strength of the output clock by ensuring there is adequate current to supply the driver. (For ZB silicon, this issue is irrelevant because the SYSCLKP should not be used. However, to ease the migration to future revisions of silicon, it is advisable to follow the recommendation.)

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.