UFT3G Register Write Sequences

Abstract

This document discusses important timing considerations that must be considered when writing to the UFT3G devices (8T49N28x, 8T49N24x, and 8T49N1012) using I2C or SPI. The UFT3G devices are very flexible and support in-system programming. Whether configuring the device for on-the-fly frequency changes, or programming the part at boot up, it is important to provide the necessary idle time between writes in order to ensure that I2C writes are successful. Though many of these devices support the SPI protocol, the scope of this document will be limited to the I2C protocol; however, the same principles and idle times apply for the SPI protocol. In addition, this document does not discuss the setup and hold timings associated with each protocol because they are covered in the device datasheets.

Contents

Figures

Tables

1. I2C Write Sequences

One important consideration is the type of I2C writes that the system is using. It is very common for systems to use single-byte writes, as described in Figure 1. The system's I2C master will write individual registers with each command starting with a Start command and terminating with a Stop command.

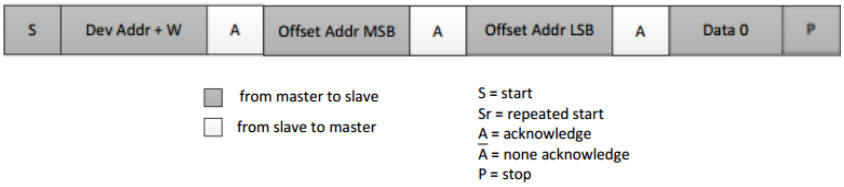

Figure 1. Single-Byte Write

Systems may also support I2C block writes, also referred to as "Sequential Write" in the datasheets and shown in Figure 2.

Figure 2. Block Write

2. Timing Considerations

The first consideration is that changes to the PLL feedback dividers or DSM_ORD will affect the I2C timings, which are derived from the PLL (PLL0 for devices with more than one PLL). Under normal operation, whenever changes are made to the feedback dividers or DSM_ORD of this PLL it triggers a re-calibration that will momentarily disrupt the I2C timing. The calibration time is dependent on the APLL phase-frequency-detector frequency, which is directly derived from the crystal.

Table 1. Time Taken for PLL Calibration and Lock with Different Crystal Frequency and DLCNT Settings

Crystal Frequency (MHz)	APLL PFD Frequency (MHz)	DLCNT Setting	Calibration Time t_{cal} (ms)
10	20	0	1248
		1	594
25	50	0	504
		1	236
40	80	0	312
		1	149
50	100	0	250
		1	119
62.5	125	0	200
		1	97

In addition, if the DPLL calibration is disabled, though changes to the feedback divider will not cause any re-calibration, they will still cause a momentary change in frequency that also disrupts the I2C timings for about 350µS. Any I2C transactions issued during these "disruptions" run the risk of corruption, so it's recommended to provide the idle times mentioned.

2.1 Timing Terms

- t_{cal}: The calibration time shown in Table 1.
- $I2C_{wait}$: The idle time from the completion of an I2C sequence (I2C Stop command) to the start of the next I2C sequence (I2C Start command).

2.2 Timing Recommendations

With the Write Sequence and Timing Considerations in mind, the following recommendations ensure deterministic I2C sequences:

Back-to-back single-byte writes that don't modify the primary PLL feedback dividers

Recommended $I2C_{wait}$=160nS. (See Appendix for the calculations).

Back-to-back single-byte writes that modify the primary PLL feedback dividers:

1. If the PLL is running under normal operation (DPLL calibration is enabled), allow t_{cal} wait time for calibration each time a feedback divider setting is modified. This may be impractical in a system since this may add seconds of idle time for a single configuration. Taking that into account, the calibration circuit may be disabled. See next point for the idle time recommendation.
2. If the DPLL's calibration circuit is disabled, allow $I2C_{wait}$=350uS of idle time.

2.3 System Implementation

The following recommendations describe how this is implemented in a system:

Step A: After reset, allow this amount of time for the I2C bus to be available:

- 150mS if not using an EEPROM
- 200ms if using an EEPROM

Step B: Use the sequence for the case below that matches the system's implementation.

Case 1: Single-byte-writes, DPLL calibration disabled

1. Disable DPLL0 and I2Cwait = 350µS

   8T49N28x: Write 0x05 to register 0x0B8

   8T49N24x: Write 0x05 to register 0x70

   8T49N1012: Write 0x03 to register 0xA2
2. Write back-to-back byte writes with I2Cwait = 160nS idle time between writes.

   If writing to the primary PLL feedback dividers, I2Cwait = 350uS between writes.
3. Enable DPLL0.

   8T49N28x: Write 0x00 to register 0x0B8

   8T49N24x: Write 0x00 to register 0x70

   8T49N1012: Write 0x00 to register 0xA2
4. Calibration takes place at the end of the write, so I2Cwait = tcal mS (from Table 1) before the next I2C command.

Case 2: Single-byte-writes, DPLL calibration enabled

1. Write back-to-back byte writes with I2Cwait = 160nS idle time between writes.
2. If modifying the primary PLL feedback dividers or DSM_ORD, $I2C_{wait} = t_{cal}$

Case 3: Block write

1. Disable DPLL0 and $I2C_{wait}$ = 350uS.
   8T49N28x: write 0x05 to register 0x0B8
   8T49N24x: write 0x05 to register 0x70
   8T49N1012: write 0x03 to register 0xA2
2. Write all the registers in a single command and $I2C_{wait}$ = 350uS.
   8T49N28x: write from register 0x00 to 0xCB
   8T49N24x: write from register 0x00 to 0x7B
   8T49N1012: write from register 0x00 to 0xA2
3. If the DPLL0 was not cleared in step 2, then enable DPLL0 as follows:
   8T49N28x: write 0x00 to register 0x0B8
   8T49N24x: write 0x00 to register 0x70
   8T49N1012: write 0x00 to register 0xA2
4. Calibration takes place at the end of the write, so wait t_{cal} before the next I2C command.

3. Appendix

Idle Time Calculations:

One T cycle = VCO period * 64.

For a device with a VCO minimum frequency of 3GHz, that is 6 * (64 * 1/3GHz) = 2 * 64 * 1nS = 128nS.

For a device with a VCO minimum frequency of 2.4GHz, that is 6 * (64 * 1/2.4GHz) = 2 * 64 * 1nS = 160nS.

4. Revision History

Revision	Date	Description
1.1	Jul.17.20	Updated the last sentence in the Appendix.
1.0	Jun.2.20	Initial release.

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.