Universe II (CA91C142) Fast Initialization Application Note

Visit our Website at http://www.tundra.com

603 March Rd., Kanata, Ontario, K2K 2M5
Tel: (613) 592-0714 • 1-800-267-7231 • Fax: (613) 592-1320

© 1999 Tundra Semiconductor Corporation

Document Number : 8091142.AN003.01

Printed in Canada

Information in this document is subject to change without notice

The information in this document is subject to change without notice and should not be construed as a commitment by Tundra Semiconductor Corporation. While reasonable precautions have been taken, Tundra Semiconductor Corporation assumes no responsibility for any errors that may appear in this document.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Tundra Semiconductor Corporation.

The acceptance of this document will be construed as an acceptance of the foregoing conditions.

Universe II (CA91C142)
Fast Initialization Application Note

This application note is intended to show the user an example initialization of the Universe II. The application note will show in detail the minimal steps that are required before the Universe II can be utilized on a board. It is divided into both hardware and software initialization aspects. This application note has the PCI bus as the bus that is utilized during initialization, however, the VMEbus could also have been used as the primary bus for initialization. This application note is intended to be used as a guide only.

HARDWARE INITIALIZATION

Normal Operating Mode

For Universe II normal system operation, and to set the JTAG[1] controller to a neutral state, the following Universe II pins should be connected as shown in the following table.

Table 1 - Pin Requirements for Normal Operating Mode

Pin Name	Pin Value	Pin Name	Pin Value
TMODE[2]	V_{SS}	VOCTL	V_{SS}
TMODE[1]	V_{SS}	TRST#	V_{SS}
TMODE[0]	V_{SS}	TCK	V_{dd}
PLL_TESTSEL	V_{SS}	TDI	V_{dd}
ENID	V_{SS}	TMS	V_{dd}
PLL_TESTOUT	N/C		

Reset and Power-up Options

On power up, the PWRRST# or the SYSRST* pins are driven low. Within two CLK64 periods after both SYSRST* and PWRRST# are negated, the Universe II latches the level on the power-up option pins on the rising edge of PWRRST# (Figure 1), and then negates VOE# one clock later. The power-up options are subsequently loaded into their respective registers several PCI clock periods after PWRRST#, SYSRST* and RST# have all been negated. As a default, all Universe II power-up option pins have an internal pull-down to ground. If other options are required other than the defaults, pins VA[31:1], REQ64#, BGIN[3]*, and VD[31:27] must be pulled-up to Vdd. Table 2.22, page 2-116 of the Universe II User Manual (Doc # 8091142.MD300.01) has a listing of the power up options. This table is indicated on the following page.

1. Joint Test Action Group. The JTAG Technical Subcommittee that developed the core of the IEEE Standard Test Access Port and Boundary - Scan Architecture (Std 1149.1-1990)

Option	***Register***	***Field***	***Default***	***Pins***
VMEbus Register Acess Slave Image	VRAI_CTL	EN	disabled	VA[31]
		VAS	A16	VA[30:29]
	VRAI_BS	BS	0x00	VA[28:21]
VMEbus CR/CSR slave image	VCSR_CTL	LAS[0]	memory	VA[20]
	VCSR_TO	TO	0x00	VA[19:15]
Auto-Id	MISC_STAT	DY4AUTO	disabled	VD[30]
	MISC_CTL	V64AUTO	disabled	VD[29]
	VINT_EN		0	
	VINT_STAT	SW_INT	0	
	VINT_MAP1	SW_INT	000	
Bi - mode	MISC_CTL	BI	disabled	VD[28]
Auto - Syscon detect	MISC_CTL	SYSCON	enabled	VBGIN[3]*
SYSFAIL* Assertion	VCSR_SET	SYSFAIL	asserted	VD[27]
	VCSR-CLR	SYSFAIL		
PCI Target Image	LSI0_CTL	EN	disabled	VA[13]
		LAS[0]	memory	VA[12]
		VAS	A16	VA[1:10]
	LSI0_BS	BS	0x0	VA[9:6]
	LSIO_BD	BD	0x0	VA[5:2]
PCI Register Access	PCI_BS0, PCI_BS1	SPACE	Table A.6 and A.7 of manual	VA[1]
PCI Bus Size	MISC_STAT	LCL_SIZE	32 bit	REQ64#
PCI CSR Image Space	PCI_CSR	BM	disabled	VA[14]

Section 2.10 of the Universe II User Manual outlines the resets and the power up options.

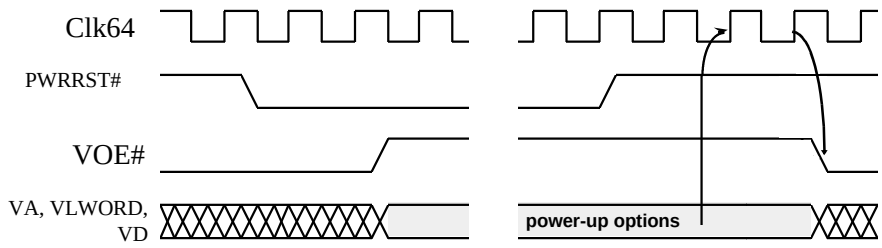

Figure 1 - Reset and Power-up Options Timing

A typical application is to pull-up pin VA[1] on power-up in order to access the PCI Registers using memory cycles from another PCI Master on the PCI bus. All other options can be set to a particular users needs.

SOFTWARE INITIALIZATION

Programming Universe II Registers

The following assumes that the user will be programming the Universe II from the local (PCI bus) side. The first registers that must be programmed in the Universe II are the PCI_BS0 and PCI_BS1 registers. These registers specifies the base address of the Universe II register space as seen from the PCI bus. One of these two registers determines the mapping into I/O space and the other into Memory, this is determined by the SPACE bit set during Power-up (See Power-up Options Table 2.22, page 2-116 of the Universe II User Manual 1998) .

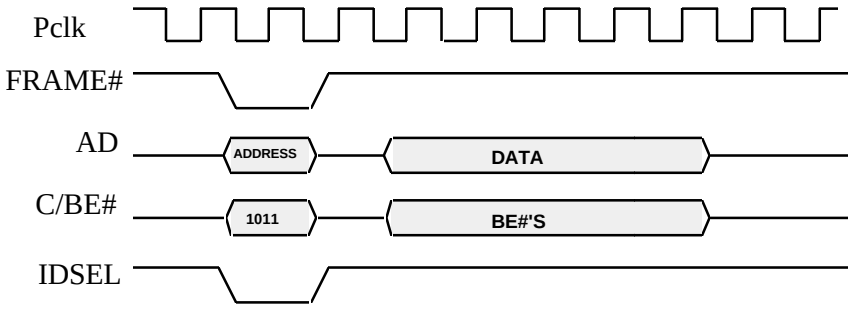

Figure 2 - Configuration Cycle with IDSEL

The second register that should be accessed is the PCI_CSR register (offset 0x004) which contains the PCI configuration space control and status registers. In the simplest form, this register can be programmed with 0x0000_0007, which will bring up the Universe II as a PCI bus master, and enable the Universe II to respond to memory and I/O cycles as a slave. This register can be accessed using configuration or memory cycles (as set by PCI_BSx). The next register that should be set is the LMISC register (offset 0x184). This register contains the value for the coupled window (CWT) timer. The CWT timer by default is set to release the VMEbus immediately after the completion of the first PCI slave channel coupled cycled.

PROGRAMMING LOCAL AND VME SLAVE IMAGES

When programming either the Local or the VME slave images (These are defined by the LSIx_xxx and VSIx_xxx registers, they allow the user to define a specific mapping of either PCI memory space (and/or I/O space) and VME memory space respectively), the most important thing to keep in mind is that all of the registers for a particular image should be programmed before the actual image is enabled. The best way to do this is to program the image base, bound, and translation offset registers first, and then write and enable the control register of the image.

This following example will demonstrate the steps required to program PCI Target Image 0 (LSI0) to do the following:

- Address Base 32'h2000_0000
- Address Bound 32'h4000_0000
- LSI0 to respond to PCI I/O cycles
- PCI address will be translated by 32'h0100_0000 on the VMEbus
- VME address and data size will be A32/D32
- VME AM code to be Program Supervisory
- VMEbus cycle type to be a combination of single and block transfer cycles
- Decoupled PCI transactions (posted write enabled)
- PCI Bus Memory Space

The first register that should be programmed is LSI0_BS (offset 104). Using memory cycles, this register should be loaded with a value of 0x2000_0000. As can be seen, the lower 12 bits of the register are reserved[2] . The LSI0_BD register (offset 0x108) should be programmed with 0x4000_0000, and the LSI0_TO registers (offset 0x10C) should be programmed with 0x0100_0000. The last step in programming the image is enabling the control register. The control register (LSI0_CTL/ offset 0x100) should be programmed with 0xC082_5101. Now local slave image 0 is programmed and ready to respond to I/O cycles.
The exact same procedure should be followed for programming VME slave images.

2. A PCI cycle that falls within this particular slave image will have the lower 12 bits of it's PCI address directly propagated as the lower 12 bits of the VME address. The upper 20 bits of the VMEbus address is determined by adding the value in the LSI0_TO register to the upper 20 bits of the PCI bus address.