



NEX-PXA2XX

PXA2XX Disassembly Software Users Manual

Including these Software Support packages:
PXAX2XX PXAX2XXEV PXAX2XXWR

Copyright © 2008 Nexus Technology, Inc. All rights reserved.

Contents of this publication may not be reproduced in any form without the written permission of Nexus Technology, Inc.

Brand and product names used throughout this manual are the trademarks of their respective holders.

Warranty Terms and License Agreement

For warranty terms, refer to the Terms and Conditions of Sale document that was included in the product shipment. The Software License Agreement is displayed during installation. A hardcopy of that agreement may be obtained from Nexus Technology.

All Nexus Technology products to which this manual refers are subject to the Terms and Conditions of Sale document and the Software License Agreement, as appropriate.

Compliance with WEEE and RoHS Directives

This product is subject to European Union regulations on Waste Electrical and Electronics Equipment. Return to Nexus Technology for recycle at end of life. Costs associated with the return to Nexus Technology are the responsibility of the sender.

TABLE OF CONTENTS

1.0 OVERVIEW	5
1.1 General Information.....	5
2.0 SOFTWARE INSTALLATION.....	5
3.0 CONNECTING TO A PXA210/PXA250 TARGET	6
3.1 Intel BBPXA2XX Evaluation Board.....	6
3.2 Wind River PXA2XX Development Board	6
3.3 Custom Target Connections	6
4.0 CLOCK SELECTION.....	10
4.1 General Information.....	10
5.0 VIEWING DATA.....	11
5.1 Viewing Timing Data on the TLA700.....	11
6.0 USING THE DISASSEMBLY SOFTWARE.....	12
6.1 General.....	12
6.2 PXA2XX Modes.....	13
6.3 "Burst Size"	13
6.4 "Read Latency"	13
6.5 "Cycle Display"	13
6.6 "Configure Memory"	14
6.7 Disassembly Using the TLA700.....	14
6.8 Help! The Disassembler's Confused (or: Using Mark Opcode)	16
6.9 Instruction Decoding/Addressing Modes Supported.....	17
6.10 PXA2XX Instructions Supported	18
7.0 THE NEX-PXA2XX CONTROL GROUP SYMBOL TABLE	20
APPENDIX A - PXA2XXEV Mictor Pin Assignments	21
APPENDIX A - PXA2XXEV Mictor Pin Assignments (cont'd.)	22
APPENDIX B - PXA2XXWR Mictor Pin Assignments.....	23
APPENDIX B - PXA2XXWR Mictor Pin Assignments (cont'd.).....	24
APPENDIX C - PXA2XX Mictor Pin Assignments.....	25
APPENDIX C - PXA2XX Mictor Pin Assignments (cont'd.).....	26
APPENDIX D - Address Translation and Address Maps	27
APPENDIX E - Necessary Signals for Clocking	30
APPENDIX F - Support	31
APPENDIX G - References.....	32

TABLE OF FIGURES

Figure 1- Clocking Mode choices	10
Figure 2- Burst Read Length choices	10
Figure 3- Burst Write Length choices	11
Figure 4- PXA2XX MagniVu Display on TLA700	12
Figure 5- Disassembly Modes in the Properties-Disassembly Window	13
Figure 6- PXA2XX Disassembly	15
Figure 7- Disassembly Display Filter Window	16
Figure 8- Mark Opcode Window.....	17
Figure 9- PXA2XX Address Map Selection Dialog	27
Figure 10- Sdram Address Maps Display.....	28
Figure 11- Sa-1111 Address Maps Display.....	28

TABLE OF TABLES

Table 1- PXA2XXEV TLA600/700 Channel Grouping	7
Table 2- PXA2XXWR TLA600/700 Channel Grouping	8
Table 3- PXA2XX TLA600/700 Channel Grouping.....	9
Table 4- NEX-PXA2XX Control Symbol Table.....	20
Table 5- PXA2XXEV 'A' Mictor Probe (J38)	21
Table 6- PXA2XXEV 'C' Mictor Probe (J28)	21
Table 7- PXA2XXEV 'D' Mictor Probe (J34)	22
Table 8- PXA2XXWR 'A' Mictor Probe (J13).....	23
Table 9- PXA2XXWR 'C' Mictor Probe (J14)	23
Table 10- PXA2XXWR 'D' Mictor Probe (JP12).....	24
Table 11- PXA2XX 'A' Mictor Probe.....	25
Table 12- PXA2XX 'C' Mictor Probe.....	25
Table 13- PXA2XX 'D' Mictor Probe.....	26
Table 14- PXA2XX Custom Clocking Required Signals.....	30

1.0 OVERVIEW

1.1 General Information

The NEX-PXA2XX disassembly software provides disassembly of acquired Intel XScale PXA210 and PXA250 memory bus using a TLA600/TLA700 with a 136-channel acquisition module. A 100MHz TLA acquisition card is sufficient unless the core clock frequency is expected to approach or exceed 200MHz. The NEX-PXA2XX support is software only. Please see Section 3.0 “Connecting to an PXA2XX Target” for information on probing.

NOTE: Three versions of the NEX-PXA2XX support exist. The NEX-PXA2XX support is generic support for any target, given that the Mictor connections specified are followed. The NEX-PXA2XXev support is configured to match the Mictor layout for the **Intel BBPXA2XX Evaluation Board**. The NEX-PXA2XXWR support is configured to match the Mictor layout for the **Wind River PXA2XX Development Board**. Except for the Mictor connections and signal maps, this manual describes all three supports.

The NEX-PXA2XX provides full instruction decoding of ARM, through Architecture v4. Thumb instructions are not supported at this time.

This manual assumes that the user is familiar with the PXA210/PXA250 processor specification and the Tektronix TLA600/TLA700 Logic Analyzers. It is also expected that the user is familiar with Windows 2000. The TLA700 Application must also be at V1.1 or later for the NEX-PXA2XX support to work properly.

2.0 SOFTWARE INSTALLATION

One 3½” diskette has been included with the NEX-PXA2XX disassembly product. The NEX-PXA2XX software is loaded in the same method as other Windows programs. Place the NEX-PXA2XX Install disk in the floppy drive of the TLA700. Select **Control Panel** and run **Add/Remove Programs**, choose **Install**, **Next**, then **Finish**. Add/Remove will then run SETUP.EXE on the floppy and install the Selected support in its proper place on the hard disk.

To load PXA2XX support into the TLA700, first select the desired Logic Analyzer card in the Setup screen, select Load Support Package from the File pull-down, then choose PXA2XX and click on **Okay**.

3.0 CONNECTING TO A PXA210/PXA250 TARGET

3.1 Intel BBPXA2XX Evaluation Board

The Intel BBPXA2XX Evaluation Board has Mictor connectors placed on it for easy access to the PXA memory signals. The TLA P6434s should be connected to the BBPXA2XX board as follows:

- 'A' Group Mictor to J38
- 'D' Group Mictor to J34
- 'C' Group Mictor to J28

Refer to Table 1 for a list of the acquired signals and how they are grouped. The NEX-PXA2XXEV version of the support is configured to work with the BBPXA2XX Evaluation Board when connected this way.

Refer to Appendix A for the complete Mictor layout for the NEX-PXA2XXEV support.

3.2 Wind River PXA2XX Development Board

The Wind River PXA2XX Development Board has Mictor connectors placed on it for easy access to the PXA memory signals. The TLA P6434s should be connected to the BBPXA2XX board as follows:

- 'A' Group Mictor to J13
- 'D' Group Mictor to JP12
- 'C' Group Mictor to J14

Refer to Table 2 for a list of the acquired signals and how they are grouped. The NEX-PXA2XXWR version of the support is configured to work with the Wind River PXA2XX Development Board when connected this way.

Refer to Appendix B for the complete Mictor layout for the NEX-PXA2XXWR support.

3.3 Custom Target Connections

To connect to a custom target, Mictor layouts on the target are the preferred method. Table 3 lists the recommended signal arrangements and how they are grouped. The NEX-PXA2XX version of the support is configured to work with custom targets when connected this way.

Refer to Appendix C for the complete Mictor layout for the NEX-PXA2XX support.

Group Name	Signal Name	TLA input	Group Name	Signal Name	TLA input
DRAMAddr (Hex)	ADDR[25]	A3:1	Data (Hex)	DATA[31]	D3:7
	ADDR[24]	A3:0		DATA[30]	D3:6
	ADDR[23]	A2:7		DATA[29]	D3:5
	ADDR[22]	A2:6		DATA[28]	D3:4
	ADDR[21]	A2:5		DATA[27]	D3:3
	ADDR[20]	A2:4		DATA[26]	D3:2
	ADDR[19]	A2:3		DATA[25]	D3:1
	ADDR[18]	A2:2		DATA[24]	D3:0
	ADDR[17]	A2:1		DATA[23]	D2:7
	ADDR[16]	A2:0		DATA[22]	D2:6
	ADDR[15]	A1:7		DATA[21]	D2:5
	ADDR[14]	A1:6		DATA[20]	D2:4
	ADDR[13]	A1:5		DATA[19]	D2:3
	ADDR[12]	A1:4		DATA[18]	D2:2
	ADDR[11]	A1:3		DATA[17]	D2:1
	ADDR[10]	A1:2		DATA[16]	D2:0
	ADDR[09]	A1:1		DATA[15]	D1:7
	ADDR[08]	A1:0		DATA[14]	D1:6
	ADDR[07]	A0:7		DATA[13]	D1:5
	ADDR[06]	A0:6		DATA[12]	D1:4
	ADDR[05]	A0:5		DATA[11]	D1:3
	ADDR[04]	A0:4		DATA[10]	D1:2
	ADDR[03]	A0:3		DATA[09]	D1:1
	ADDR[02]	A0:2		DATA[08]	D1:0
	ADDR[01]	A0:1		DATA[07]	D1:7
ADDR[00]	A0:0	DATA[06]	D0:6		
Control (Symbol)	nSDCS[3]	C3:5	DATA[05]	D0:5	
	nSDCS[2]	C3:4	DATA[04]	D0:4	
	nSDCS[1]	C3:3	DATA[03]	D0:3	
	nSDCS[0]	C3:2	DATA[02]	D0:2	
	nSDRAS	C2:5	DATA[01]	D0:1	
	nSDCAS	C2:4	DATA[00]	D0:0	
	nWE	C1:1	Bytes (Hex)	DQM[3]	C3:1
Misc (Bin)	SDCKE[1]	C3:6		DQM[2]	C3:0
	SDCKE[0]	A3:4		DQM[1]	C2:7
	nSDCLK[1]	C3:6;CK3		DQM[0]	C2:6
Ungrouped	nSDCLK[2]	A3:5	Ungrouped	nCS[5]	C1:7
	PWM[1]	A3:3		nCS[4]	C1:6
	PWM[0]	A3:2		nCS[3]	C1:5
	nSDCLK[0]	C2:3		nCS[2]	C1:4
	PREG#	C2:2		nCS[1]	C1:3
	PSKTSEL#	C2:1		nCS[0]	C1:2
	PWAIT#	C2:0		RD/WR#	C0:7
	PCE[2]	C0:2		POE#	C0:6
	PCE[1]	C0:1		PWE#	C0:5
	IOIS !6#	C0:0		PIOW#	C0:4
				PIOR#	C0:3

Table 1- PXA2XXEV TLA600/700 Channel Grouping

Group Name	Signal Name	TLA input	Group Name	Signal Name	TLA input
HiData (Hex)	DQ63	D1:7	LoData (Hex)	DQ31	D3:7
	DQ 62	D1:6		DQ30	D3:6
	DQ 61	D1:5		DQ29	D3:5
	DQ 60	D1:4		DQ28	D3:4
	DQ 59	D1:3		DQ27	D3:3
	DQ 58	D1:2		DQ26	D3:2
	DQ 57	D1:1		DQ25	D3:1
	DQ 56	D1:0		DQ24	D3:0
	DQ 55	D1:7		DQ23	D2:7
	DQ54	D0:6		DQ22	D2:6
	DQ53	D0:5		DQ21	D2:5
	DQ52	D0:4		DQ20	D2:4
	DQ51	D0:3		DQ19	D2:3
	DQ50	D0:2		DQ18	D2:2
	DQ49	D0:1		DQ17	D2:1
	DQ48	D9:0		DQ16	D2:0
	DQ47	E1:7		DQ15	E3:7
	DQ46	E1:6		DQ14	E3:6
	DQ45	E1:5		DQ13	E3:5
	DQ44	E1:4		DQ12	E3:4
	DQ43	E1:3		DQ11	E3:3
	DQ42	E1:2		DQ10	E3:2
	DQ41	E1:1		DQ9	E3:1
	DQ40	E1:0		DQ8	E3:0
	DQ39	E0:7		DQ7	E2:7
	DQ38	E0:6		DQ6	E2:6
	DQ37	E0:5		DQ5	E2:5
	DQ36	E0:4		DQ4	E2:4
	DQ35	E0:3		DQ3	E2:3
	DQ34	E0:2		DQ2	E2:2
	DQ33	E0:1		DQ1	E2:1
	DQ32	E0:0		DQ0	E2:0
MaskBits (Hex)	SDQM7	C3:7	DRAMAddr	SBA1	A2:1
	SDQM6	C3:6		SBA0	A2:0
	SDQM5	C3:5		SA12	C1:4
	SDQM4	C3:4		SA11	C1:3
	SDQM3	C3:3		SA10	C1:2
	SDQM2	C3:2		SA9	C1:1
	SDQM1	C3:1		SA8	C1:0
	SDQM0	C3:0		SA7	C0:7
ChekBits	SCB7	C2:7		SA6	C0:6
	SCB6	C2:6		SA5	C0:5
	SCB5	C2:5		SA4	C0:4
	SCB4	C2:4		SA3	C0:3
	SCB3	C2:3		SA2	C0:2
	SCB2	C1:7		SA1	C0:1
	SCB1	C1:6		SA0	C0:0
	SCB0	C1:5			
Misc (Bin)	DCLK2	Clock:2	Control (Sym)	SCE1#	A2:1
Ungrouped	ABORT	A2:6		SCE0#	A2:2
	DVALID	A2:5		SRAS#	C1:5
				SCAS#	C1:6
			SWE#	C1:7	

Table 2- PXA2XXWR TLA600/700 Channel Grouping

Group Name	Signal Name	TLA input	Group Name	Signal Name	TLA input
HiData (Hex)	DQ63	D1:7	LoData (Hex)	DQ31	D3:7
	DQ 62	D1:6		DQ30	D3:6
	DQ 61	D1:5		DQ29	D3:5
	DQ 60	D1:4		DQ28	D3:4
	DQ 59	D1:3		DQ27	D3:3
	DQ 58	D1:2		DQ26	D3:2
	DQ 57	D1:1		DQ25	D3:1
	DQ 56	D1:0		DQ24	D3:0
	DQ 55	D1:7		DQ23	D2:7
	DQ54	D0:6		DQ22	D2:6
	DQ53	D0:5		DQ21	D2:5
	DQ52	D0:4		DQ20	D2:4
	DQ51	D0:3		DQ19	D2:3
	DQ50	D0:2		DQ18	D2:2
	DQ49	D0:1		DQ17	D2:1
	DQ48	D9:0		DQ16	D2:0
	DQ47	E1:7		DQ15	E3:7
	DQ46	E1:6		DQ14	E3:6
	DQ45	E1:5		DQ13	E3:5
	DQ44	E1:4		DQ12	E3:4
	DQ43	E1:3		DQ11	E3:3
	DQ42	E1:2		DQ10	E3:2
	DQ41	E1:1		DQ9	E3:1
	DQ40	E1:0		DQ8	E3:0
	DQ39	E0:7		DQ7	E2:7
	DQ38	E0:6		DQ6	E2:6
	DQ37	E0:5		DQ5	E2:5
	DQ36	E0:4		DQ4	E2:4
	DQ35	E0:3		DQ3	E2:3
	DQ34	E0:2		DQ2	E2:2
	DQ33	E0:1		DQ1	E2:1
	DQ32	E0:0		DQ0	E2:0
MaskBits (Hex)	SDQM7	C3:7	DRAMAddr	SBA1	A2:1
	SDQM6	C3:6		SBA0	A2:0
	SDQM5	C3:5		SA12	C1:4
	SDQM4	C3:4		SA11	C1:3
	SDQM3	C3:3		SA10	C1:2
	SDQM2	C3:2		SA9	C1:1
	SDQM1	C3:1		SA8	C1:0
	SDQM0	C3:0		SA7	C0:7
ChekBits	SCB7	C2:7		SA6	C0:6
	SCB6	C2:6		SA5	C0:5
	SCB5	C2:5		SA4	C0:4
	SCB4	C2:4		SA3	C0:3
	SCB3	C2:3		SA2	C0:2
	SCB2	C1:7		SA1	C0:1
	SCB1	C1:6	SA0	C0:0	
	SCB0	C1:5			
Misc (Bin)	DCLK2	Clock:2	Control (Sym)	SCE1#	A2:1
Ungrouped	ABORT	A2:6		SCE0#	A2:2
	DVALID	A2:5		SRAS#	C1:5
				SCAS#	C1:6
			SWE#	C1:7	

Table 3- PXA2XX TLA600/700 Channel Grouping

4.0 CLOCK SELECTION

4.1 General Information

There are no clocking options available when using the NEX-PXA2XXEV or the NEX-PXA2XXWR support packages. By default the software is configured to acquired data on every rising edge of SDCK.

The NEX-PXA2XX supports several clocking options. Figure 1 shows the Clocking Mode choices, Rising Edge of Clock (acquires all cycles) or Selective Clocking (acquires only Read and Write cycles).

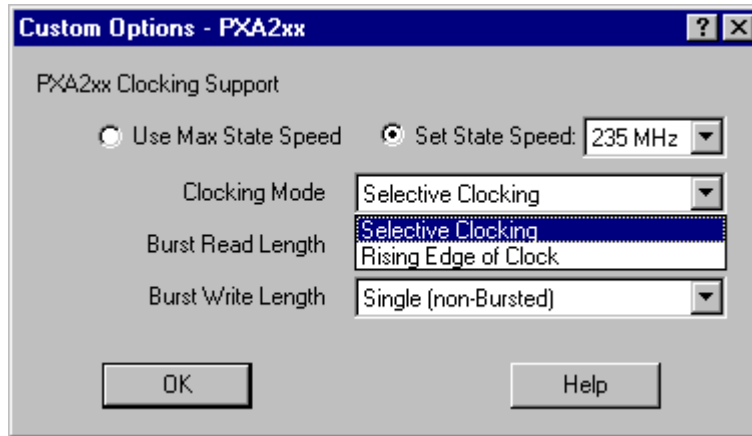


Figure 1- Clocking Mode choices

Figure 2 shows the Burst Read Length choices, 1/2/4/8 (for limited bursts) or Full Page (for variable burst lengths).

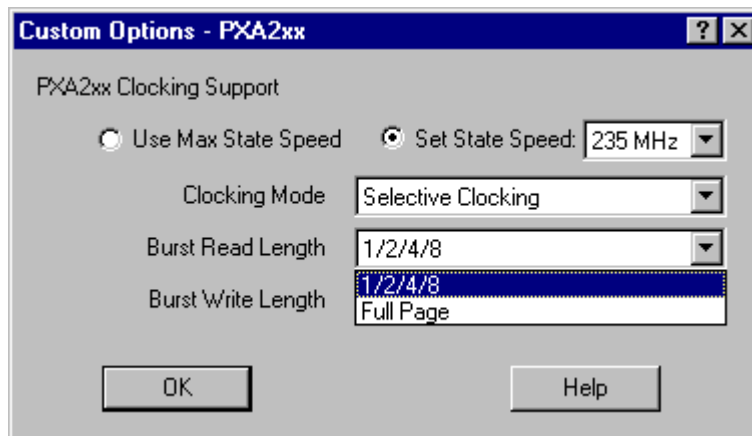


Figure 2- Burst Read Length choices

Figure 3 shows the Burst Write Length choices, Single (when running Multi-Burst Single Write) or Same Length as Reads (Multi-Burst).

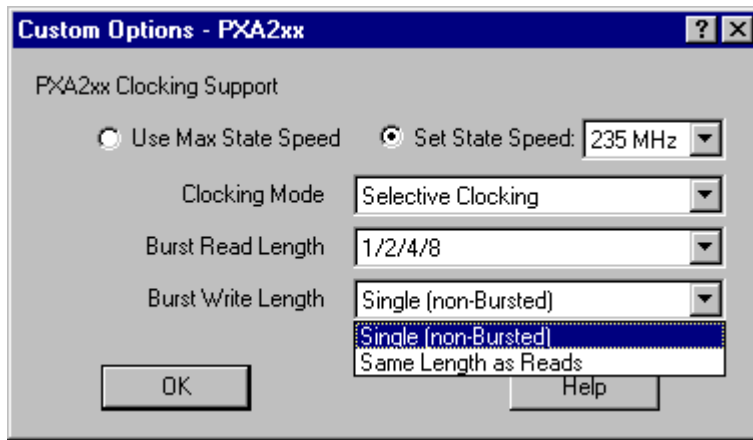


Figure 3- Burst Write Length choices

5.0 VIEWING DATA

5.1 Viewing Timing Data on the TLA700

By default, the TLA700 will display an acquisition in the Disassembly mode. However, the same data can be displayed in Timing form by adding a Waveform Display window. This is done by clicking on the Window pull-down, selecting New Data Window, clicking on Waveform Window Type, then choosing the Data Source. Two choices are presented: PXA2XX and PXA2XX-MagniVu. The first will show the exact same data (same acquisition mode) as that shown in the Disassembly window, except in Timing format. The second selection, PXA2XX-MagniVu, will show all of the channels in 2GHz MagniVu mode, so that edge relationships can be examined at the module's trigger point. With either selection, all channels can be viewed by scrolling down the window. Refer to the TLA700 System User's Manual for additional information on formatting the Waveform display.

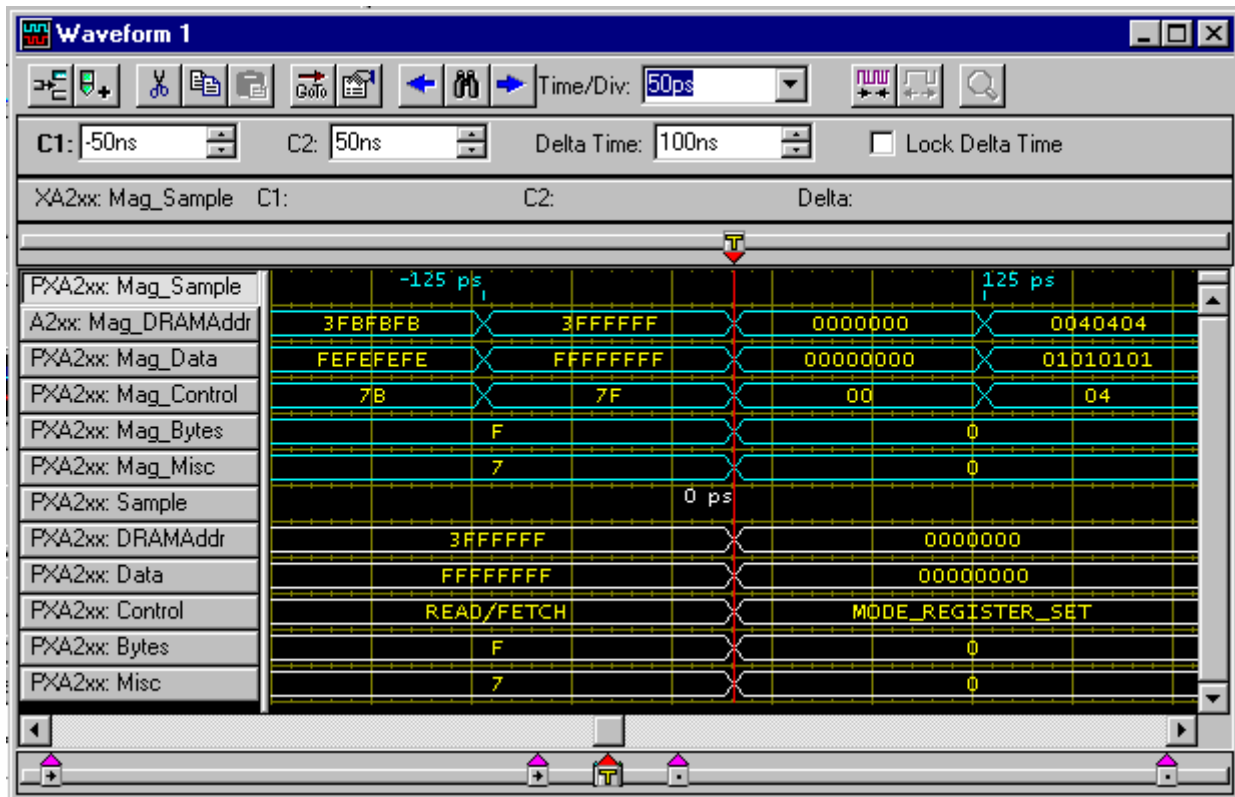


Figure 4- PXA2XX MagniVu Display on TLA700

6.0 USING THE DISASSEMBLY SOFTWARE

NOTE: For the PXA2XX support to acquire all cycles in proper order, instruction prefetch caching **MUST BE DISABLED**. Please refer to the Intel PXA250 and PXA210 Application Processors Developer’s Manual for information on disabling the instruction and data caches.

6.1 General

The NEX-PXA2XX support software acquires and decodes PXA2XX bus activity and displays the information as assembly language mnemonics (machine code) - see Figure 6. This permits the tracing of code execution for debug purposes. It is possible to filter the data display cycle types of interest to the software engineer (Figure 7). The user can choose to display the acquired data in Hardware, Software, Control Flow, or Subroutine modes.

Every stored cycle has a timestamp value stored with it. This time information, accurate to 500ps in the TLA7L/M/N/P/Q series, and accurate to 125ps in the TLA7AA/B series, permits precise measurements of microprocessor bus activity. Because of the design of Tektronix Logic Analyzers there is no need to worry about trading off acquisition memory depth when making these measurements, as the timestamp memory is separate from the acquisition memory.

6.2 PXA2XX Modes

To support variations in target configurations, several modes have been designed into the PXA2XX support. These are "Burst Size", "Read Latency", "Cycle Display", and "Configure Memory". Figure 5 shows the Properties-Disassembly window, with these modes. Each mode is explained below.

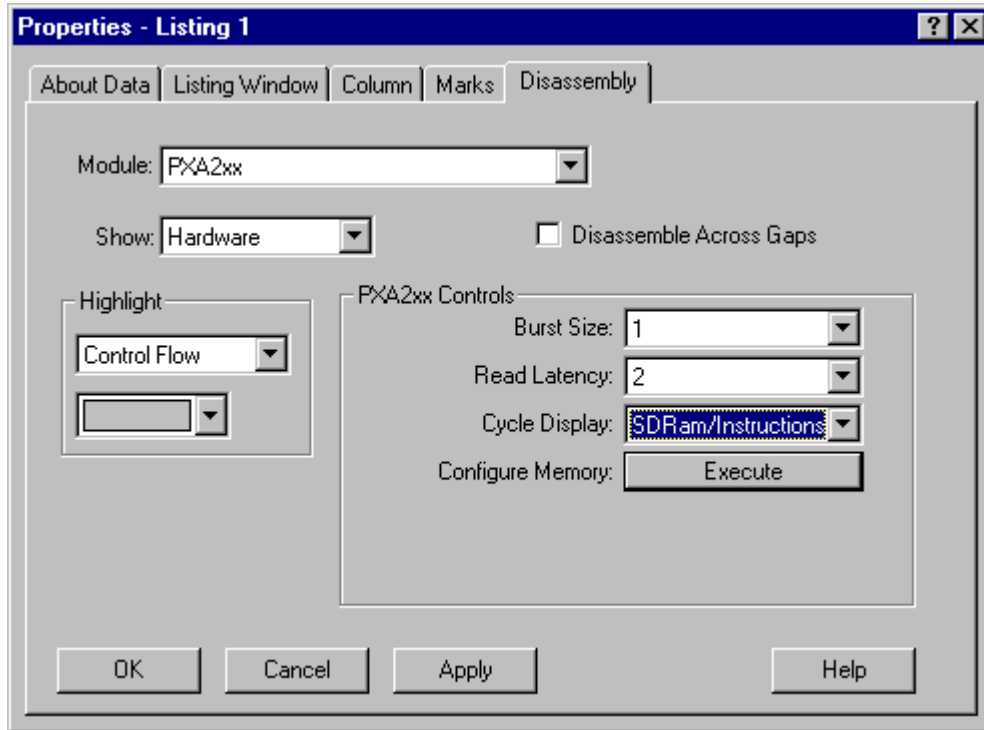


Figure 5- Disassembly Modes in the Properties-Disassembly Window

6.3 "Burst Size"

This sets the how many locations are read for each read cycle. Valid choices are 1, 2, 4 and 8.

6.4 "Read Latency"

This sets the delay, in clock cycles, from the Read command until the first piece of valid Read data is available. This value must be set properly for all valid Read Data to be displayed. Valid choices are 2 (default) or 3 cycles.

6.5 "Cycle Display"

This selection enables the user to view only Instructions, SDRAM cycles and Instructions, or SDRAM cycles only. The default selection displays Instructions only.

6.6 "Configure Memory"

The PXA2XX supports two different sets of address maps, SA-1111 and Normal Bank Addressing. This select field permits the user to specify the precise address map used, so that accurate address reconstruction is possible. Refer to the Intel PXA250 and PXA210 Application Processors Developer's Manual Section 6.6.5.1 "SDRAM Addressing Modes" for further information. The NEX-PXA2XX support uses this information to reconstruct the physical memory address from the RAS and CAS cycles and display it in the PXA2XX Address group. Refer to Appendix D for more information on Address Maps.

6.7 Disassembly Using the TLA700

The TLA700, since it is a Windows program, has the same type of user interface as other Windows-based applications. In the Disassembly Listing window, a tool bar at the top of the window contains buttons that allow the user to modify the display. These buttons, from left to right, perform the following functions:

- Add Column - Adds a column to the display
- Add Mark - Adds a user mark to the display
- Cut - (may be grayed out) - Cuts the selection to the Clipboard
- Copy - (may be grayed out) - Copies the selection to the Clipboard
- Paste - (may be grayed out) - Inserts the contents of the Clipboard
- Go To - Moves the display to the item of interest
- Properties - Edits the current Listing Display properties
- Smaller Font - Decreases the displayed font size
- Larger Font - Increases the displayed font size
- Search Backward - Moves to a previous data match
- Define Search - Define data to be matched
- Search Forward - Moves to the next data match
- Mark Opcode - Permits placing an opcode mark

The format (or display properties) of each displayed column can be changed by putting the mouse cursor on the heading of the column, clicking the left mouse button to select that column, clicking the right mouse button to bring up the editing dialog, then selecting Properties. The column to be modified can also be selected by clicking on the Column tab, selecting the column of interest in the Column field, then making any desired modifications to that display column. The modification or selections possible will vary from column to column.

Two display columns of particular interest are the Timestamp and Mnemonics columns. Timestamp shows a time value associated with the acquisition. By default, Timestamp shows the time from System Trigger. Clicking on the From window in the Timestamp Reference field shows all available selections: Absolute (from when the Logic Analyzer was started), Previous (the time from the present sequence to the previous displayed one), and three selections that permit time to be displayed from different reference points: System Trigger, Cursor 1 Current Position, and Cursor 2 Current Position. Selecting the desired mode with the mouse, and then clicking the left mouse button, will make the selection the present Timestamp display mode.

Listing 1

C1: 2046 C2: 2050 Delta Time: 16ns Lock Delta Time

PXA2xx PXA2xx

Sample	PXA2xx DRAMAddr	PXA2xx Data	PXA2xx Mnemonics	Control	Byt
39	0000494	27272727	STRCS/HS R2, [R7, -R7 LSR #OE]!	READ/FETCH(S3, S1)	2
47	00004B4	2F2F2F2F	SWICS/HS #2F2F2F	READ/FETCH(S3)	2
55	10004D4	37373737	UNDEFINED	READ/FETCH(S1)	3
71	0000514	47474747	STRMIB R4, [R7, -R7 ASR #OE]	READ/FETCH(S3, S1, S0)	4
79	0A02534	4F4F4F4F	SWIMI #4F4F4F	READ/FETCH(S3, S0)	4
87	1404554	57575757	UNDEFINED	READ/FETCH(S1, S0)	5
95	1E06574	5F5F5F5F	SWIPL #5F5F5F	READ/FETCH(S0)	5
103	0000594	67676767	STRVSB R6, [R7, -R7 ROR #OE]!	READ/FETCH(S3, S1)	6
111	0A0A5B4	6F6F6F6F	SWIVS #6F6F6F	READ/FETCH(S3)	6
119	14045D4	77777777	UNDEFINED	READ/FETCH(S1)	7
263	0000414	07070707	STREQ R0, [R7, -R7 LSL #OE]	READ/FETCH(S3, S1, S0)	0
271	0A02434	0F0F0F0F	SWIEQ #0F0F0F	READ/FETCH(S3, S0)	0
279	1404454	17171717	UNDEFINED	READ/FETCH(S1, S0)	1
287	1E06474	1F1F1F1F	SWINE #1F1F1F	READ/FETCH(S0)	1
295	0000494	27272727	STRCS/HS R2, [R7, -R7 LSR #OE]!	READ/FETCH(S3, S1)	2
303	0A0A4B4	2F2F2F2F	SWICS/HS #2F2F2F	READ/FETCH(S3)	2
311	14044D4	37373737	UNDEFINED	READ/FETCH(S1)	3
327	0000514	47474747	STRMIB R4, [R7, -R7 ASR #OE]	READ/FETCH(S3, S1, S0)	4
335	0A02534	4F4F4F4F	SWIMI #4F4F4F	READ/FETCH(S3, S0)	4
343	1404554	57575757	UNDEFINED	READ/FETCH(S1, S0)	5
351	1E06574	5F5F5F5F	SWIPL #5F5F5F	READ/FETCH(S0)	5
359	0000594	67676767	STRVSB R6, [R7, -R7 ROR #OE]!	READ/FETCH(S3, S1)	6
367	0A0A5B4	6F6F6F6F	SWIVS #6F6F6F	READ/FETCH(S3)	6
375	14045D4	77777777	UNDEFINED	READ/FETCH(S1)	7
519	0000414	07070707	STREQ R0, [R7, -R7 LSL #OE]	READ/FETCH(S3, S1, S0)	0
527	0A02434	0F0F0F0F	SWIEQ #0F0F0F	READ/FETCH(S3, S0)	0
535	1404454	17171717	UNDEFINED	READ/FETCH(S1, S0)	1
543	1E06474	1F1F1F1F	SWINE #1F1F1F	READ/FETCH(S0)	1
551	0000494	27272727	STRCS/HS R2, [R7, -R7 LSR #OE]!	READ/FETCH(S3, S1)	2
559	0A0A4B4	2F2F2F2F	SWICS/HS #2F2F2F	READ/FETCH(S3)	2
567	14044D4	37373737	UNDEFINED	READ/FETCH(S1)	3
583	0000514	47474747	STRMIB R4, [R7, -R7 ASR #OE]	READ/FETCH(S3, S1, S0)	4

Figure 6- PXA2XX Disassembly

The other column of interest is the Mnemonics column, where the PXA2XX disassembly information is displayed. As mentioned previously, it is possible to filter the PXA2XX instructions that are displayed. This is done via selections made in the Disassembly tab of the Properties window (see Figure 7). By default the display is in Hardware mode, where all bus cycles are displayed (Memory Reads, Memory Writes, Instructions, etc.). Other choices are: Software (only executed instructions are displayed), Control Flow (display of instructions affecting code flow such as Jumps, Branches, etc.), and Subroutine (only instructions such as Calls, Returns, etc. are displayed).

Note that when data is suppressed in this fashion that Timestamp information (in Previous form) will be updated to show the time between displayed cycles.

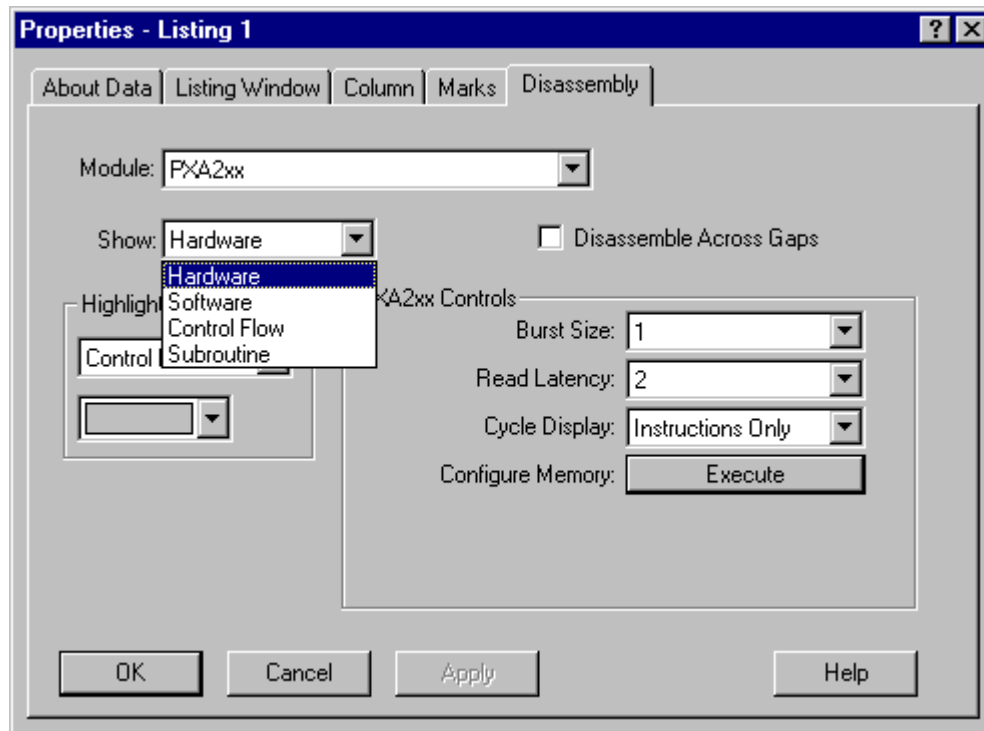


Figure 7- Disassembly Display Filter Window

6.8 Help! The Disassembler's Confused (or: Using Mark Opcode)

Because the PXA2XX does not have a signal to denote an opcode fetch, it is difficult to distinguish between a Read cycle and an Opcode Fetch. The disassembly software does the best it can to figure this out, but it assumes that every Read from contiguous memory cycles is a Fetch, and this may not always be the case. When this happens the user can Mark an Opcode (a Sample or Cycle in the List display) to help the disassembler re-synchronize. To do this, the user first moves the mouse cursor to the cycle that is to be marked. Click on the right mouse button to bring up the menu selections, then click on Mark Opcode. Another window will appear (see Figure 8) which permits selecting the type of cycle that is to be marked (Opcode Fetch, Memory Read, etc.). Select the desired cycle and then click on Okay. To remove an existing Mark, select Undo Mark and then click on OK

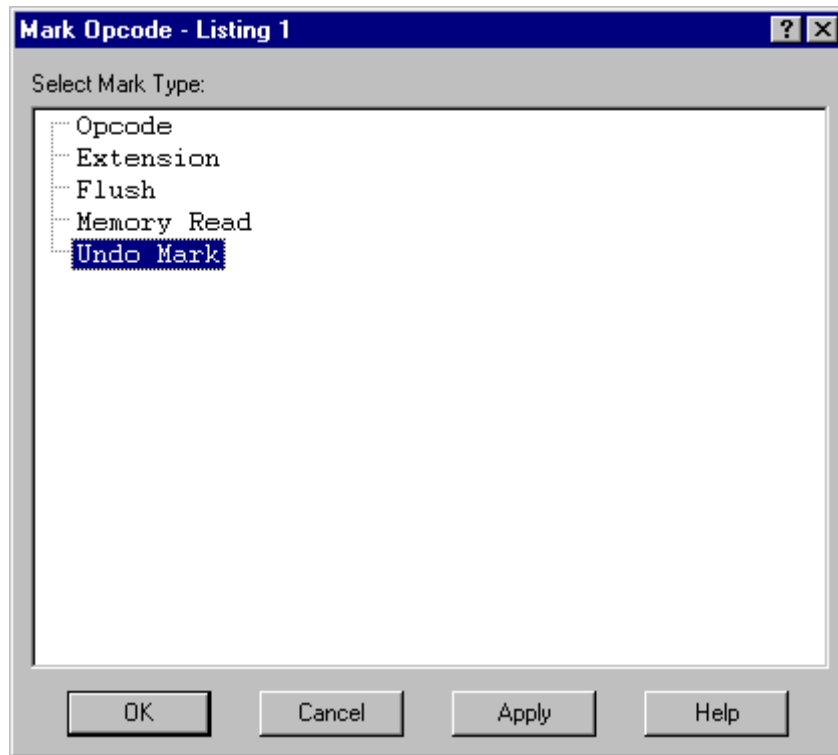


Figure 8- Mark Opcode Window

6.9 Instruction Decoding/Addressing Modes Supported

The following lists the particular feature sets that the NEX-PXA2XX disassembler packages supports.

IMPORTANT: The Thumb instruction set is not supported at this time.

Architecture v4 Level Instructions and Addressing modes

- Load data read cycle detection
- LDM and LDC multiple load read cycle detection
- Branch Prefetch Instruction flush detection
- Mark-Opcode support

All five addressing modes:

Addressing Mode 1

- Shifter operands
- Immediate
- Register
- Logical shift left by immediate
- Logical shift left by register
- Logical shift right by immediate
- Logical shift right by register

- Arithmetic shift right by immediate
- Arithmetic shift right by register
- Rotate right by immediate
- Rotate right by register
- Rotate right with extend

Addressing Mode 2

- Immediate offset
- Register offset
- Scaled register offset
- Immediate pre-indexed
- Register pre-indexed
- Scaled register pre-indexed
- Immediate post-indexed
- Register post-indexed
- Scaled register post-indexed

Addressing Mode 3

- Immediate offset
- Register offset
- Immediate pre-indexed
- Register pre-indexed
- Immediate post-indexed
- Register post-indexed

Addressing Mode 4

- Increment after
- Increment before
- Decrement after
- Decrement before

Addressing Mode 5

- Immediate offset
- Immediate pre-indexed
- Immediate post-indexed

6.10 PXA2XX Instructions Supported

ADC	{<cond>}{S} Rd, Rn, <shifter_operand>
ADD	{<cond>}{S} Rd, Rn, <shifter_operand>
AND	{<cond>}{S} Rd, Rn, <shifter_operand>
B{L}	{<cond>} <target address>
BIC	{<cond>}{S} Rd, Rn, <shifter_operand>

BX	{<cond>} Rm	
CDP	{<cond>} p<cp#>, <opcode_1>, CRd, CRn, CRm, <opcode_2>	
CMN	{<cond>} Rn, <shifter_operand>	
CMP	{<cond>} Rn, <shifter_operand>	
EOR	{<cond>} {S} Rd, Rn, <shifter_operand>	
LDC	{<cond>} p<cp_num>, CRd, <addressing_mode>	
LDM	{<cond>} <addressing_mode> Rn{!}, <registers>	LDM(1)
LDM	{<cond>} <addressing_mode> Rn, <registers>^	LDM(2)
LDM	{<cond>} <addressing_mode> Rn{!}, <registers_and_pc>^	
LDM(3)		
LDR	{<cond>} Rd, <addressing_mode>	
LDR	{<cond>} B Rd, <addressing_mode>	
LDR	{<cond>} BT Rd, <post_indexed_addressing_mode>	
LDR	{<cond>} H Rd, <addressing_mode>	
LDR	{<cond>} SB Rd, <addressing_mode>	
LDR	{<cond>} SH Rd, <addressing_mode>	
LDR	{<cond>} T Rd, <post_indexed_addressing_mode>	
MCR	{<cond>} p<cp#>, <opcode_1>, Rd, CRn, CRm, <opcode_2>	
MLA	{<cond>} {<S>} Rd, Rm, Rs, Rn	
MOV	{<cond>} {S} Rd, <shifter_operand>	
MRC	{<cond>} p<cp#>, <opcode_1>, Rd, CRn, CRm, <opcode_2>	
MRS	{<cond>} Rd, CPSR	
MRS	{<cond>} Rd, SPSR	
MSR	{<cond>} Rd, CPSR_f, #32bit immediate	
MSR	{<cond>} Rd, CPSR_<fields>, Rm	
MSR	{<cond>} Rd, SPSR_f, #32bit immediate	
MSR	{<cond>} Rd, SPSR_<fields>, Rm	
MUL	{<cond>} {<S>} Rd, Rm, Rs	
MVN	{<cond>} {<S>} Rd, <shifter_operand>	
ORR	{<cond>} {S} Rd, Rn, <shifter_operand>	
RSB	{<cond>} {S} Rd, Rn, <shifter_operand>	
RSC	{<cond>} {S} Rd, Rn, <shifter_operand>	
SBC	{<cond>} {S} Rd, Rn, <shifter_operand>	
SMLAL	{<cond>} {<S>} RdLo, RdHi, Rm, Rs	
SMULL	{<cond>} {<S>} RdLo, RdHi, Rm, Rs	
STC	{<cond>} p<cp_num>, CRd, <addressing_mode>	
STM	{<cond>} <addressing_mode> Rn{!}, <registers>	STM(1)
STM	{<cond>} <addressing_mode> Rn{!}, <registers>^	STM(2)
STR	{<cond>} Rd, <addressing_mode>	
STR	{<cond>} B Rd, <addressing_mode>	
STR	{<cond>} BT Rd, <post_indexed_addressing_mode>	
STR	{<cond>} H Rd, <addressing_mode>	
STR	{<cond>} T Rd, <post_indexed_addressing_mode>	
SUB	{<cond>} {<S>} Rd, Rn, <shifter_operand>	
SWI	{<cond>} <24_bit_immediate>	
SWP	{<cond>} Rd, Rm, [Rn]	
SWP	{<cond>} B Rd, Rm, [Rn]	
TEQ	{<cond>} Rn, <shifter_operand>	

TST {<cond>} Rn, <shifter_operand>
UMLAL {<cond>}{<S>} RdLo, RdHi, Rm, Rs
UMULL {<cond>}{<S>} RdLo, RdHi, Rm, Rs

7.0 THE NEX-PXA2XX CONTROL GROUP SYMBOL TABLE

The use of Symbol Tables when displaying state data and defining a trigger enables the user to quickly determine the type of bus cycle that occurred or is desired. A symbol table for the Control group (PXA2XX_Ctrl, Table 4) has been provided to quickly show the type of bus transaction acquired when viewing data in Listing display. This same symbol table can be used in the Trigger area of the TLA to easily define the sort of bus cycle that is to be triggered on.

Pattern	TLA700 Symbols	Meaning
11xxx	DESL – IGNORE_COMMAND—DATA?	Idle or Data Cycle
10111	NOP—NO_OPERATION_(S0~)	
01111	NOP—NO_OPERATION_(S1~)	
10110	BST—BURST_STOP_(S0~)	
01110	BST—BURST_STOP_(S1~)	
10101	READ—COL_ADDR_READ_(S0~)	
01101	READ—COL_ADDR_READ_(S1~)	
10100	WRITE—COL_ADDR_WRITE_(S0~)	
01100	WRITE—COL_ADDR_WRITE_(S1~)	
10011	ACTV—ROW_ADDRESS_STROBE_(S0~)	
01011	ACTV—ROW_ADDRESS_STROBE_(S1~)	
10010	PRE—PRECHARGE_SELECT_BANK_(S0~)	
01010	PRE—PRECHARGE_SELECT_BANK_(S1~)	
10001	REF—REFRESH_(S0~)	
01001	REF—REFRESH_(S1~)	
10000	MRS—MODE_REGISTER_SET_(S0~)	
01000	MRS—MODE_REGISTER_SET_(S1~)	

Table 4- NEX-PXA2XX Control Symbol Table

Signals, from left to right: SCE1#, SCE0#, SRAS#, SCAS#, SWE#

APPENDIX A - PXA2XXEV Mictor Pin Assignments

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	CK0		6	36	CK1	
7	4	A3:7		8	35	A1:7	ADDR[15]
9	5	A3:6		10	34	A1:6	ADDR[14]
11	6	A3:5	SDCLK[2]	12	33	A1:5	ADDR[13]
13	7	A3:4	SDCKE[0]	14	32	A1:4	ADDR[12]
15	8	A3:3	PWM[1]	16	31	A1:3	ADDR[11]
17	9	A3:2	PWM[0]	18	30	A1:2	ADDR[10]
19	10	A3:1	ADDR[25]	20	29	A1:1	ADDR[09]
21	11	A3:0	ADDR[24]	22	28	A1:0	ADDR[08]
23	12	A2:7	ADDR[23]	24	27	A0:7	ADDR[07]
25	13	A2:6	ADDR[22]	26	26	A0:6	ADDR[06]
27	14	A2:5	ADDR[21]	28	25	A0:5	ADDR[05]
29	15	A2:4	ADDR[20]	30	24	A0:4	ADDR[04]
31	16	A2:3	ADDR[19]	32	23	A0:3	ADDR[03]
33	17	A2:2	ADDR[18]	34	22	A0:2	ADDR[02]
35	18	A2:1	ADDR[17]	36	21	A0:1	ADDR[01]
37	19	A2:0	ADDR[16]	38	20	A0:0	ADDR[00]

Table 5- PXA2XXEV 'A' Mictor Probe (J38)

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	CK3	SDCLK[1]	6	36	Q1	
7	4	C3:7	SDCLK[1]	8	35	C1:7	CS[5]
9	5	C3:6	SDCKE[1]	10	34	C1:6	CS[4]
11	6	C3:5	SDCS[3]#	12	33	C1:5	CS[3]
13	7	C3:4	SDCS[2]#	14	32	C1:4	CS[2]
15	8	C3:3	SDCS[1]#	16	31	C1:3	CS[1]
17	9	C3:2	SDCS[0]#	18	30	C1:2	CS[0]
19	10	C3:1	DQM[3]	20	29	C1:1	WE#
21	11	C3:0	DQM[2]	22	28	C1:0	OE#
23	12	C2:7	DQM[1]	24	27	C0:7	RD/WR#
25	13	C2:6	DQM[0]	26	26	C0:6	POE#
27	14	C2:5	SDRAS	28	25	C0:5	PWE#
29	15	C2:4	SDCAS	30	24	C0:4	PIOW#
31	16	C2:3	SDCLK[0]	32	23	C0:3	PIOR#
33	17	C2:2	PREG#	34	22	C0:2	PCE[2]
35	18	C2:1	PSKTSEL#	36	21	C0:1	PCE[1]
37	19	C2:0	PWAIT#	38	20	C0:0	IOIS 16#

Table 6- PXA2XXEV 'C' Mictor Probe (J28)

APPENDIX A - PXA2XXEV Mictor Pin Assignments (cont'd.)

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	Q0		6	36	CK2	
7	4	D3:7	DATA[31]	8	35	D1:7	DATA[15]
9	5	D3:6	DATA[30]	10	34	D1:6	DATA[14]
11	6	D3:5	DATA[29]	12	33	D1:5	DATA[13]
13	7	D3:4	DATA[28]	14	32	D1:4	DATA[12]
15	8	D3:3	DATA[27]	16	31	D1:3	DATA[11]
17	9	D3:2	DATA[26]	18	30	D1:2	DATA[10]
19	10	D3:1	DATA[25]	20	29	D1:1	DATA[09]
21	11	D3:0	DATA[24]	22	28	D1:0	DATA[08]
23	12	D2:7	DATA[23]	24	27	D0:7	DATA[07]
25	13	D2:6	DATA[22]	26	26	D0:6	DATA[06]
27	14	D2:5	DATA[21]	28	25	D0:5	DATA[05]
29	15	D2:4	DATA[20]	30	24	D0:4	DATA[04]
31	16	D2:3	DATA[19]	32	23	D0:3	DATA[03]
33	17	D2:2	DATA[18]	34	22	D0:2	DATA[02]
35	18	D2:1	DATA[17]	36	21	D0:1	DATA[01]
37	19	D2:0	DATA[16]	38	20	D0:0	DATA[00]

Table 7- PXA2XXEV 'D' Mictor Probe (J34)

APPENDIX B - PXA2XXWR Mictor Pin Assignments

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	CK0	MICTOR_CK	6	36	CK1	/SDCAS
7	4	A3:7	/SDCS0	8	35	A1:7	MA[15]
9	5	A3:6	/SDRAS	10	34	A1:6	MA[14]
11	6	A3:5	/SDCAS	12	33	A1:5	MA[13]
13	7	A3:4		14	32	A1:4	MA[12]
15	8	A3:3		16	31	A1:3	MA[11]
17	9	A3:2		18	30	A1:2	MA[10]
19	10	A3:1	MA[25]	20	29	A1:1	MA[09]
21	11	A3:0	MA[24]	22	28	A1:0	MA[08]
23	12	A2:7	MA[23]	24	27	A0:7	MA[07]
25	13	A2:6	MA[22]	26	26	A0:6	MA[06]
27	14	A2:5	MA[21]	28	25	A0:5	MA[05]
29	15	A2:4	MA[20]	30	24	A0:4	MA[04]
31	16	A2:3	MA[19]	32	23	A0:3	MA[03]
33	17	A2:2	MA[18]	34	22	A0:2	MA[02]
35	18	A2:1	MA[17]	36	21	A0:1	MA[01]
37	19	A2:0	MA[16]	38	20	A0:0	MA[00]

Table 8- PXA2XXWR 'A' Mictor Probe (J13)

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	CK3	MICTOR_CK	6	36	Q1	/SDCAS
7	4	C3:7	MD[31]	8	35	C1:7	MD[15]
9	5	C3:6	MD[30]	10	34	C1:6	MD[14]
11	6	C3:5	MD[29]	12	33	C1:5	MD[13]
13	7	C3:4	MD[28]	14	32	C1:4	MD[12]
15	8	C3:3	MD[27]	16	31	C1:3	MD[11]
17	9	C3:2	MD[26]	18	30	C1:2	MD[10]
19	10	C3:1	MD[25]	20	29	C1:1	MD[09]
21	11	C3:0	MD[24]	22	28	C1:0	MD[08]
23	12	C2:7	MD[23]	24	27	C0:7	MD[07]
25	13	C2:6	MD[22]	26	26	C0:6	MD[06]
27	14	C2:5	MD[21]	28	25	C0:5	MD[05]
29	15	C2:4	MD[20]	30	24	C0:4	MD[04]
31	16	C2:3	MD[19]	32	23	C0:3	MD[03]
33	17	C2:2	MD[18]	34	22	C0:2	MD[02]
35	18	C2:1	MD[17]	36	21	C0:1	MD[01]
37	19	C2:0	MD[16]	38	20	C0:0	MD[00]

Table 9- PXA2XXWR 'C' Mictor Probe (J14)

APPENDIX B - PXA2XXWR Mictor Pin Assignments (cont'd.)

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	Q0		6	36	CK2	
7	4	D3:7	/WE	8	35	D1:7	/SDCAS
9	5	D3:6	/OE	10	34	D1:6	/SDRAS
11	6	D3:5	RDY	12	33	D1:5	MBREQ
13	7	D3:4	/RESETOUT	14	32	D1:4	MBGNT
15	8	D3:3		16	31	D1:3	
17	9	D3:2		18	30	D1:2	/POE
19	10	D3:1	SDCKE0	20	29	D1:1	/PWE
21	11	D3:0	MICTOR_CK	22	28	D1:0	/PIOW
23	12	D2:7		24	27	D0:7	/PIOR
25	13	D2:6	SDCLK0	26	26	D0:6	/PCE2
27	14	D2:5	/PCI_CS5	28	25	D0:5	/PCE1
29	15	D2:4	/PCI_CS4	30	24	D0:4	
31	16	D2:3		32	23	D0:3	/PWAIT
33	17	D2:2	/UART_CS2	34	22	D0:2	/PIOIS16
35	18	D2:1	/LED_CS1	36	21	D0:1	
37	19	D2:0	/FLASH_CS0	38	20	D0:0	/PREG

Table 10- PXA2XXWR 'D' Mictor Probe (JP12)

APPENDIX C - PXA2XX Mictor Pin Assignments

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	CK0	SDCLKx	6	36	CK1	SDCLKx
7	4	A3:7	MA[25]	8	35	A1:7	MA[09]
9	5	A3:6	MA[24]	10	34	A1:6	MA[08]
11	6	A3:5	MA[23]	12	33	A1:5	MA[07]
13	7	A3:4	MA[22]	14	32	A1:4	MA[06]
15	8	A3:3	MA[21]	16	31	A1:3	MA[05]
17	9	A3:2	MA[20]	18	30	A1:2	MA[04]
19	10	A3:1	MA[19]	20	29	A1:1	MA[03]
21	11	A3:0	MA[18]	22	28	A1:0	MA[02]
23	12	A2:7	MA[17]	24	27	A0:7	MA[01]
25	13	A2:6	MA[16]	26	26	A0:6	MA[00]
27	14	A2:5	MA[15]	28	25	A0:5	
29	15	A2:4	MA[14]	30	24	A0:4	
31	16	A2:3	MA[13]	32	23	A0:3	
33	17	A2:2	MA[12]	34	22	A0:2	
35	18	A2:1	MA[11]	36	21	A0:1	
37	19	A2:0	MA[10]	38	20	A0:0	

Table 11- PXA2XX 'A' Mictor Probe

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	CK3	nSDCS1	6	36	Q1	nSDCS0
7	4	C3:7	SDCKE1	8	35	C1:7	MD[15]
9	5	C3:6	SDCKE0	10	34	C1:6	MD[14]
11	6	C3:5		12	33	C1:5	MD[13]
13	7	C3:4		14	32	C1:4	MD[12]
15	8	C3:3		16	31	C1:3	MD[11]
17	9	C3:2		18	30	C1:2	MD[10]
19	10	C3:1		20	29	C1:1	MD[09]
21	11	C3:0		22	28	C1:0	MD[08]
23	12	C2:7	DQM3 ¹	24	27	C0:7	MD[07]
25	13	C2:6	DQM2 ¹	26	26	C0:6	MD[06]
27	14	C2:5	DQM1	28	25	C0:5	MD[05]
29	15	C2:4	DQM0	30	24	C0:4	MD[04]
31	16	C2:3	nOE	32	23	C0:3	MD[03]
33	17	C2:2	nWE	34	22	C0:2	MD[02]
35	18	C2:1	nSDCAS	36	21	C0:1	MD[01]
37	19	C2:0	nSDRAS	38	20	C0:0	MD[00]

Table 12- PXA2XX 'C' Mictor Probe

APPENDIX C - PXA2XX Mictor Pin Assignments (cont'd.)

Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name	Amp Mictor Pin #	Tek Mictor Pin #	TLA700 Channel	PXA2XX Signal Name
5	3	Q0	nSDCS3	6	36	CK2	nSDCS2
7	4	D3:7	²	8	35	D1:7	MD[31] ¹
9	5	D3:6	²	10	34	D1:6	MD[30] ¹
11	6	D3:5	²	12	33	D1:5	MD[29] ¹
13	7	D3:4	²	14	32	D1:4	MD[28] ¹
15	8	D3:3	²	16	31	D1:3	MD[27] ¹
17	9	D3:2	²	18	30	D1:2	MD[26] ¹
19	10	D3:1	²	20	29	D1:1	MD[25] ¹
21	11	D3:0	²	22	28	D1:0	MD[24] ¹
23	12	D2:7	²	24	27	D0:7	MD[23] ¹
25	13	D2:6	²	26	26	D0:6	MD[22] ¹
27	14	D2:5	²	28	25	D0:5	MD[21] ¹
29	15	D2:4	²	30	24	D0:4	MD[20] ¹
31	16	D2:3	²	32	23	D0:3	MD[19] ¹
33	17	D2:2	²	34	22	D0:2	MD[18] ¹
35	18	D2:1	²	36	21	D0:1	MD[17] ¹
37	19	D2:0	²	38	20	D0:0	MD[16] ¹

Table 13- PXA2XX 'D' Mictor Probe

Notes:

- 1) PXA250 only
- 2) Do not use (recommended)

APPENDIX D - Address Translation and Address Maps

The SDRAM Address lines, which are present in the acquisition from the memory bus, consist of the Row and Column addresses during the Row Address Strobe (RAS) and Column Address Strobe (CAS) cycles. This represents the way a linear address is translated to utilize Dynamic RAM. Although a complete explanation of SDRAM addressing is beyond the scope of this manual; this appendix will cover the impact of this address translation on the support package and the manner in which it is handled in the disassembler controls.

The exact correspondence of linear address bits to SDRAM Row and Column address bits will vary, depending on the *physical* memory arrangement used by the target. SDRAM component sizes, quantities and banks used in the memory arrangement will affect the mapping of Row and Column address bits to the linear address seen in an assembler application. For example, “Intel® PXA250 and PXA210 Application Processors Developer’s Manual” specifies memory maps which are supported by the PXA2XX in Section 6.6.5.1. To support these various memory arrangements, this support provides an Address Map for each configuration. Each of these Address Maps is saved as a discrete file in a subdirectory of the support directory, with an extension of *.nam*. For the PXA2XX support, this directory is:

C:\Program Files\TLA 700\Supports\PXA2XX\Maps

(This assumes that the Tektronix TLA software was installed on the C: drive under Program Files).

When the “Execute” button is clicked in the disassembly controls (reference Figure 7), the following dialog will be displayed:

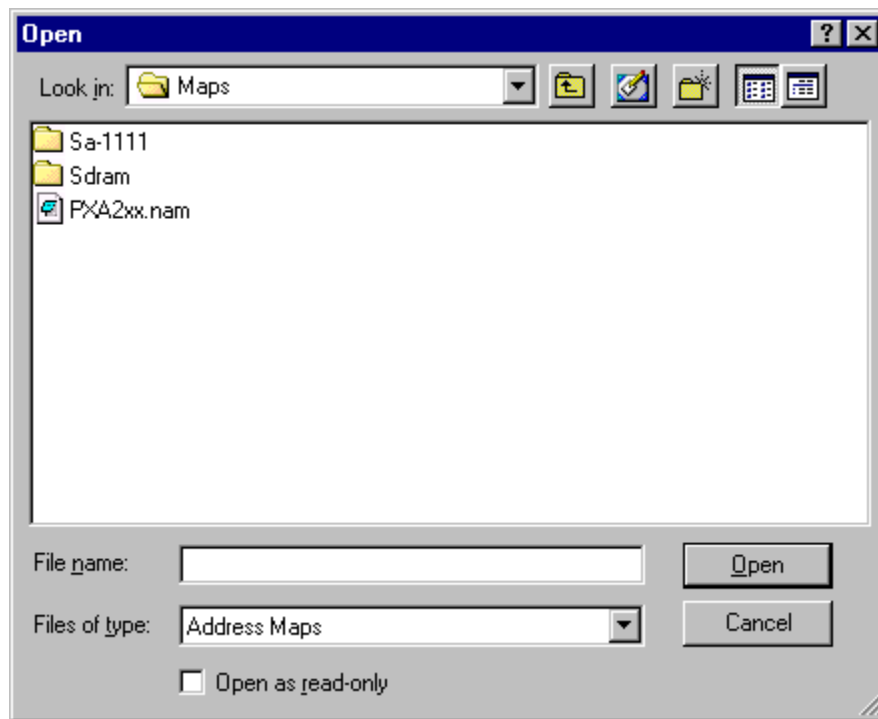


Figure 9- PXA2XX Address Map Selection Dialog

This dialog displays the contents of the Address Maps subdirectory. Notice there is a file in this directory, the name of which is *the support name and a .nam extension* (in this example, PXA2XX.nam). This is the *active* map that the support is using to reconstruct the linear address

during disassembly. To change this map you can open the “Sdram” subdirectory, which will display the dialog below:

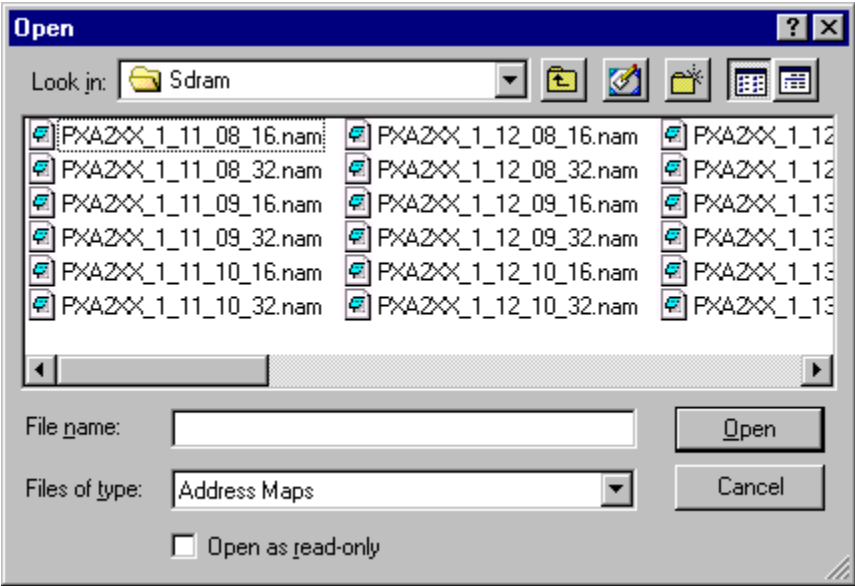


Figure 10- Sdram Address Maps Display

This dialog shows the available SDRAM Address Maps. You may also navigate to the “Sa-1111” directory, which will display the dialog below:

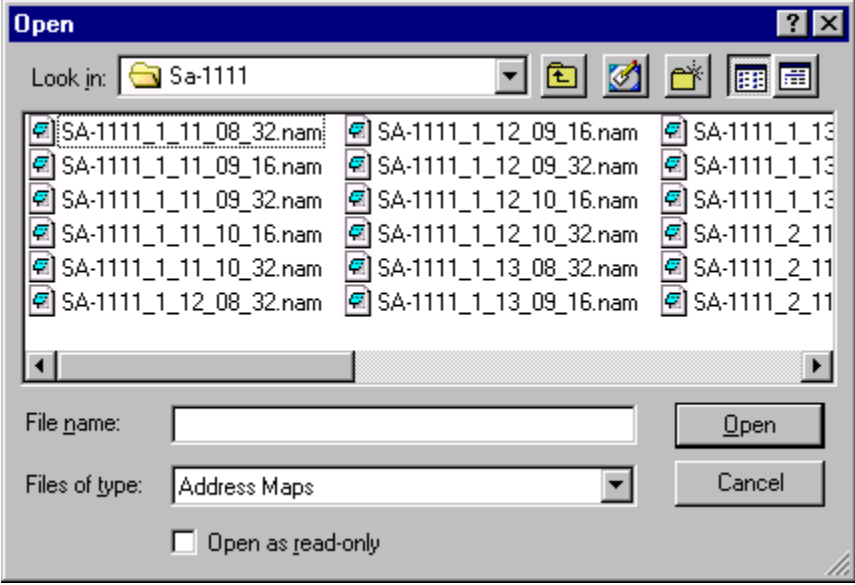


Figure 11- Sa-1111 Address Maps Display

This dialog shows the available SA-1111 Address Maps.

To select a new map, double click the left mouse button on the file icon or single click the left mouse button then single click the left mouse button on the “Open” button. This will return you to the disassembly controls display. When the changes are accepted, the map selected will be copied to the working map in the Maps subdirectory (into the file with the support name in it). The selected map is now the *active* map; which will be used by the support, until another map is selected. Selecting the map that matches the target’s memory arrangement insures that the disassembly display will show the correct linear address.

APPENDIX E - Necessary Signals for Clocking

The NEX-PXA2XXEV and NEX-PXA2XXWR support clocks in data on every rising edge of SDCLK. The PXA2XX supports custom clocking based on the following signals being present:

Signal	Channel
SDCLK	CK0
SDRAS	C2:0
SDCAS	C2:1
nWE	C2:2
SDCS0	Q1
SDCS1	CK3
SDCS2	CK2
SDCS3	Q0

Table 14- PXA2XX Custom Clocking Required Signals

This makes it possible for the support to only acquire meaningful cycles that are relevant to memory access, including RAS, CAS, Read Burst and Write Burst cycles. This will allow more meaningful data to be acquired for each acquisition taken.

APPENDIX F - Support

About Nexus Technology, Inc.



Established in 1991, Nexus Technology, Inc. is dedicated to developing, marketing, and supporting Bus Analysis applications for Tektronix Logic Analyzers.

We can be reached at:

Nexus Technology, Inc.
78 Northeastern Blvd. #2
Nashua, NH 03062

TEL: 877-595-8116
FAX: 877-595-8118

Web site: <http://www.nexustechnology.com>

Support Contact Information

Technical Support	techsupport@nexustechnology.com
General Information	support@nexustechnology.com
Quote Requests	quotes@nexustechnology.com

We will try to respond within one business day.

If Problems Are Found

Document the problem and e-mail the information to us. If at all possible please forward a Saved System Setup (with acquired data) that shows the problem. Do not send a text listing alone as that does not contain enough data for analysis. To prevent corruption during the mailing process it is strongly suggested that the Setup be zipped before transmission.

APPENDIX G - References

Tektronix TLA700 System User's Manual

Tektronix TLA700 Module User's Manual

Tektronix P6434 Mass Termination Probe Instruction Manual

“Advanced RISC Machines Architectural Reference Manual”

Edited by Dave Jaggard

Document Number ARM DD1 01008

ISBN 0-13-736299-4

© Advanced RISC Machines Ltd. (ARM) 1996

Prentice Hall, New York, NY

PXA250 and PXA210 Application Processors Developer's Manual

© Intel Corporation, February 2002

Order Number: 278522-001