NEX-IBM440GP
IBM440GP Disassembly Software Users Manual

Including these Software Support packages:
IBM440GP

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

TABLE OF FIGURES

TABLE OF TABLES

1.0 OVERVIEW

1.1 General Information

The NEX-IBM440GP disassembly software provides disassembly of acquired IBM440GP bus cycles. The NEX-IBM440GP support includes the NEX-DDRHS support, which is used to decode the DDR memory cycles which are needed to disassemble the Power PC instruction set for the IBM440GP. The TLA acquisition module and system must meet all requirements to acquire the DDR memory cycles. Please refer to the NEX-DDRHS manual for more information. The NEX- IBM440GP support is software only. Please see Section 3.0 of the NEX-DDRHS manual "CONNECTING to the NEX-DDRHS ADAPTER" for information on probing.

The IBM440GP provides full instruction decoding of the IBM 440 GP (PowerPC) instruction set. This manual assumes that the user is familiar with the IBM440GP processor specifications and the Tektronix TLA600/TLA700 Logic Analyzer. It is also expected that the user is familiar with Windows O.S. The TLA Application Software must be at V4.2 or later for the NEX-IBM440GP support to work properly.

2.0 SOFTWARE INSTALLATION

One 3½" diskettes has been included with the NEX-IBM440GP disassembly product. The NEX-IBM440GP software is loaded in the same method as other Windows programs. Place the NEX-IBM440GP 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 NEX-IBM440GP 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 either NEX-IBM440GP and click on **Okay**.

3.0 CONNECTING TO AN NEX-IBM440GP TARGET

3.1 General

As the IBM440GP utilizes DDR memory, connecting to a target is identical to connecting to DDR Memory. Please refer to the NEX-DDRHS manual for all connection information. Prior to using the NEX-IBM440GP, the support needs to be fully configured according to the NEX-DDRHS manual, which may include adjusting setup and hold times for DDR memory. Refer to the NEX-DDRHS manual, section 5.0 "CONFIGURING FOR READ / WRITE DATA ACQUISITION" for details on configuring DDR acquisition.

4.0 CLOCK SELECTION

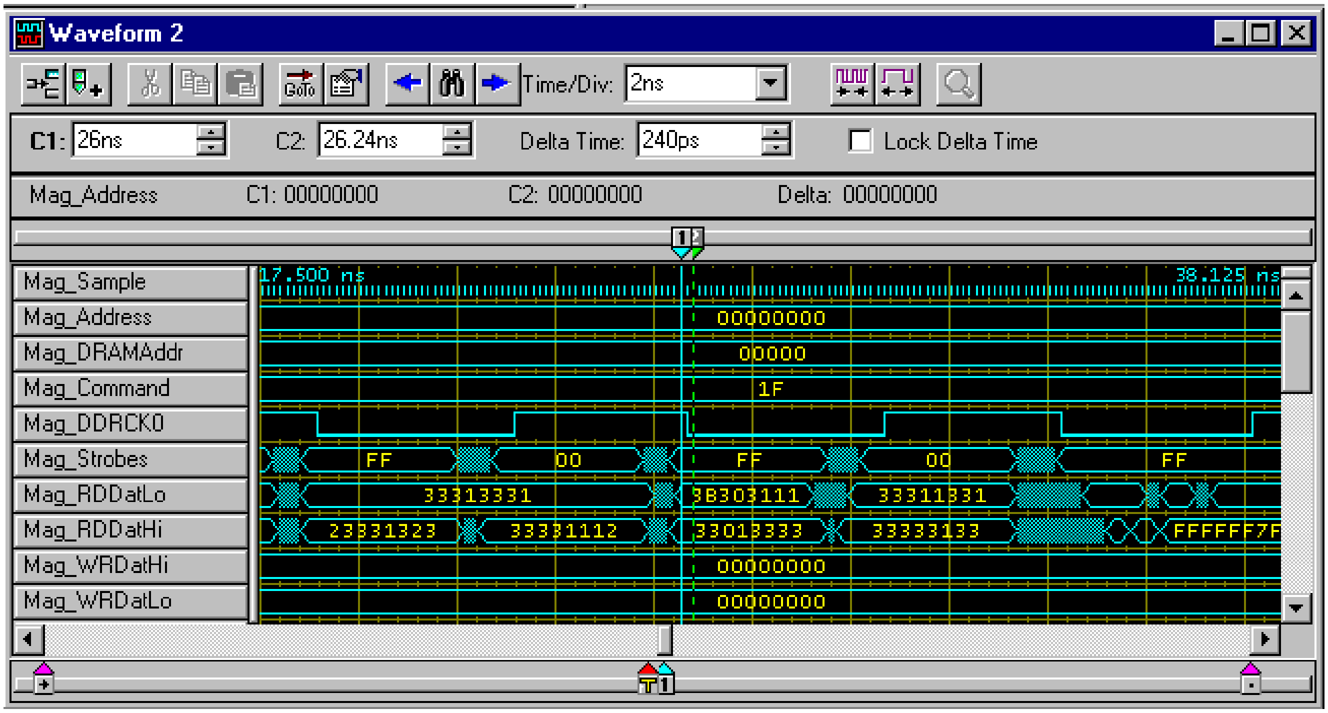
4.1 General Information

There are two clocking selections available when using the NEX-IBM440GP support package, Bus Cycles and Rising Edge of CLK. Each is explained in detail in section 4.0, “CLOCK SELECTION”, of the NEX-DDRHS manual.

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: IBM440GP and IBM440GP -MagniVu. The first will show the same data (from the same acquisition mode) as that shown in the Disassembly window, except in Timing format. The second selection, IBM440GP -MagniVu, will show all of the channels in 2GHz/8GHz 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.



Waveform 2

Time/Div: 2ns

C1: 26ns C2: 26.24ns Delta Time: 240ps ☐ Lock Delta Time

Mag_Address C1: 00000000 C2: 00000000 Delta: 00000000

Figure 1- IBM440GP MagniVu Display on TLA700

6.0 USING THE DISASSEMBLY SOFTWARE

6.1 General

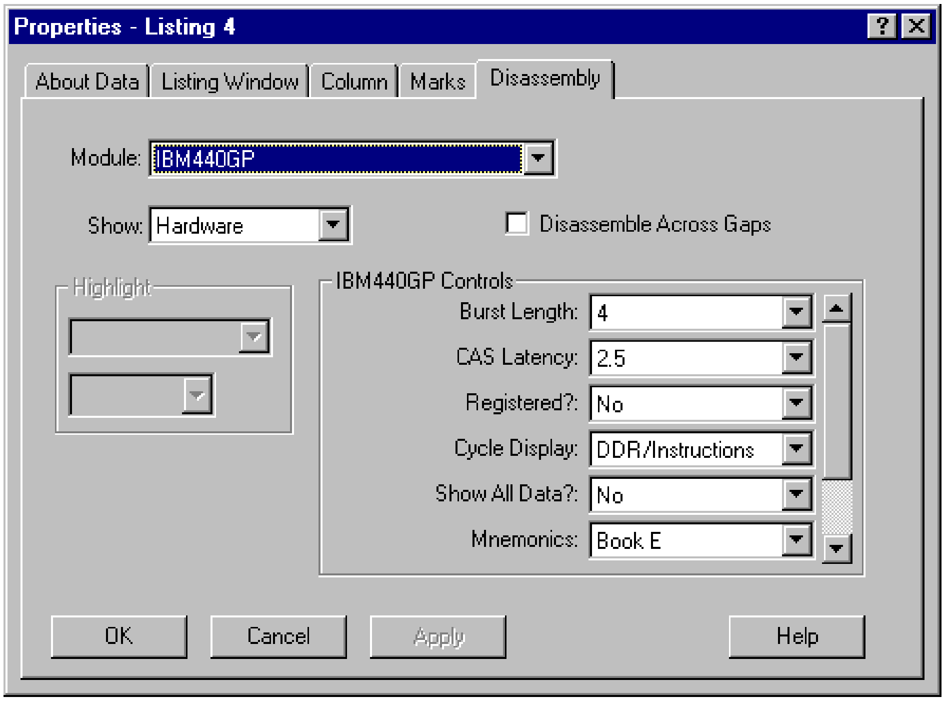
The NEX- IBM440GP support software acquires and decodes IBM440GP bus activity and displays the information as assembly language mnemonics (machine code) - see Figure 3. This permits the tracing of code execution for debug purposes. It is possible to filter the data display cycle types of interest to the software engineer (Figure 4). The user can choose to display the acquired data in Hardware, Software, Control Flow, or Subroutine modes.

A major feature of the NEX- IBM440GP software is its ability to intelligently acquire bus cycle information. By taking advantage of the data clocking power built in to the Tektronix Logic Analyzers the support software is able to acquire only the valid IBM440GP bus cycles and ignore Idle and Wait states. This means that the user is able to make optimum use of the acquisition card's memory and see more microprocessor bus cycles. For debug purposes the user also has the ability to override this function and acquire data on every Rising CLK Edge to permit the user to see all of the bus traffic including the Idle and Wait states. (See Section 4.2 Clocking Options in the NEX-DDRHS manual for further information.)

Every stored cycle (bus or clock edge, depending upon clocking selection) has a timestamp value stored with it. This time information, accurate to 500ps/125ps in the TLA600/700 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 Configuring the IBM440GP Disassembler

Because of the complexity and flexibility of the IBM440GP micro it is necessary to properly configure the support so the disassembler can process the data and display it accurately. The configuration menu can be accessed by moving to an IBM440GP List window, clicking on the display with the right mouse button, selecting **Properties** then left-clicking on the **Disassembly** tab. The menu shown in Figure 2 will then be displayed. Note the control box (named "IBM440GP Controls") that contains 8 select fields labeled "Burst Length", "Cas Latency", "Registered?", "Cycle Display", "Show All Data?", "Mnemonics", "Instruction Order", and "Configure Memory". The control box can be scrolled to view the lower select fields. Each select field is described in detail as follows:



Module: IBM440GP

Show: Hardware

□ Disassemble Across Gaps

IBM440GP Controls

Burst Length: 4

CAS Latency: 2.5

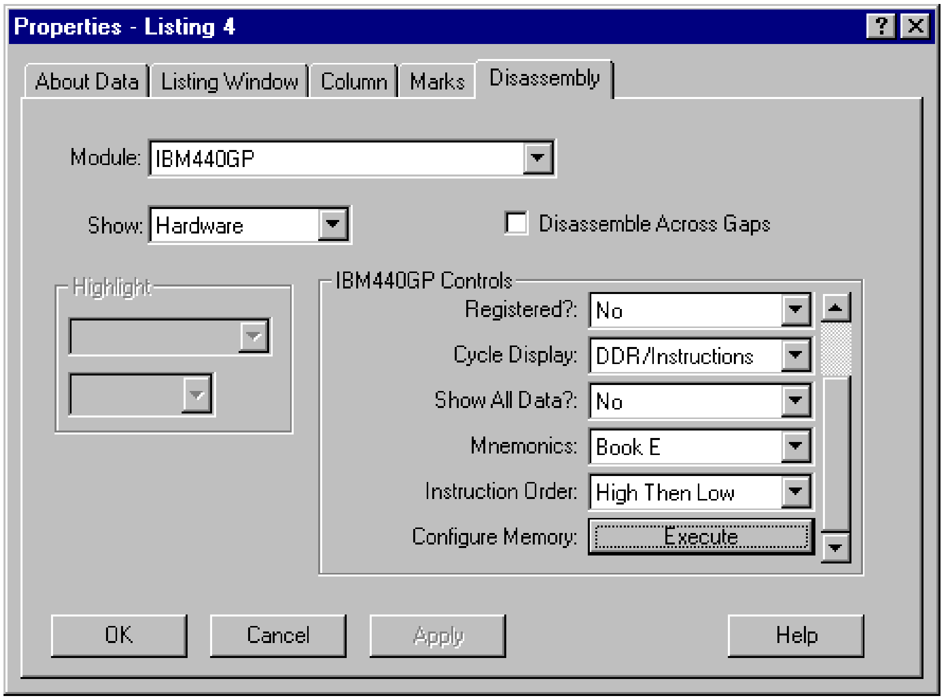
Registered?: No

Cycle Display: DDR/Instructions

Show All Data?: No

Mnemonics: Book E

Figure 2- IBM440GP Disassembly Controls



Module: IBM440GP

Show: Hardware

□ Disassemble Across Gaps

IBM440GP Controls

Registered?: No

Cycle Display: DDR/Instructions

Show All Data?: No

Mnemonics: Book E

Instruction Order: High Then Low

Configure Memory: Execute

Figure 3- Additional IBM440GP Disassembly Controls

Burst Length – This field permits specification of the Burst Size; which the DDR Memory is configured to use (usually set with a Mode Register Set (MRS) instruction).

Cas Latency – This field permits specification of the Latency used by the DDR Memory (usually set with a Mode Register Set (MRS) instruction).

Registered? – This field allows specification of whether the DDR Memory is running registered mode (usually set with a Mode Register Set (MRS) instruction).

Cycle Display – This field allows setting what types of cycles are displayed. The choices are DDR Only, DDR/Instructions and Instructions Only.

Show All Data? – This field allows setting whether irrelevant data and address values are displayed or suppressed.

Mnemonics – This field allows the user to choose to view the disassembled instructions as Book E mnemonics or Extended mnemonics. See Appendix A for further information.

Instruction Order – This field allows setting which order the Instructions are processed, which accounts for "Big-Endian" versus "Little Endian" byte ordering.

Configure Memory – This field allows selection of the SDRAM address to linear address mapping. Clicking the left mouse button on this control will bring up a selection dialog, which will display the available address maps. The maps provided are those listed in Section 17.7.3, "Physical Address to Memory Address Mapping" and shown in Table 17-12 and 17-13 of the "PPC440GP Embedded Processor User's Manual" from IBM. For more details on Address Translation, refer to Appendix B.

6.3 Disassembly Using the TLA700

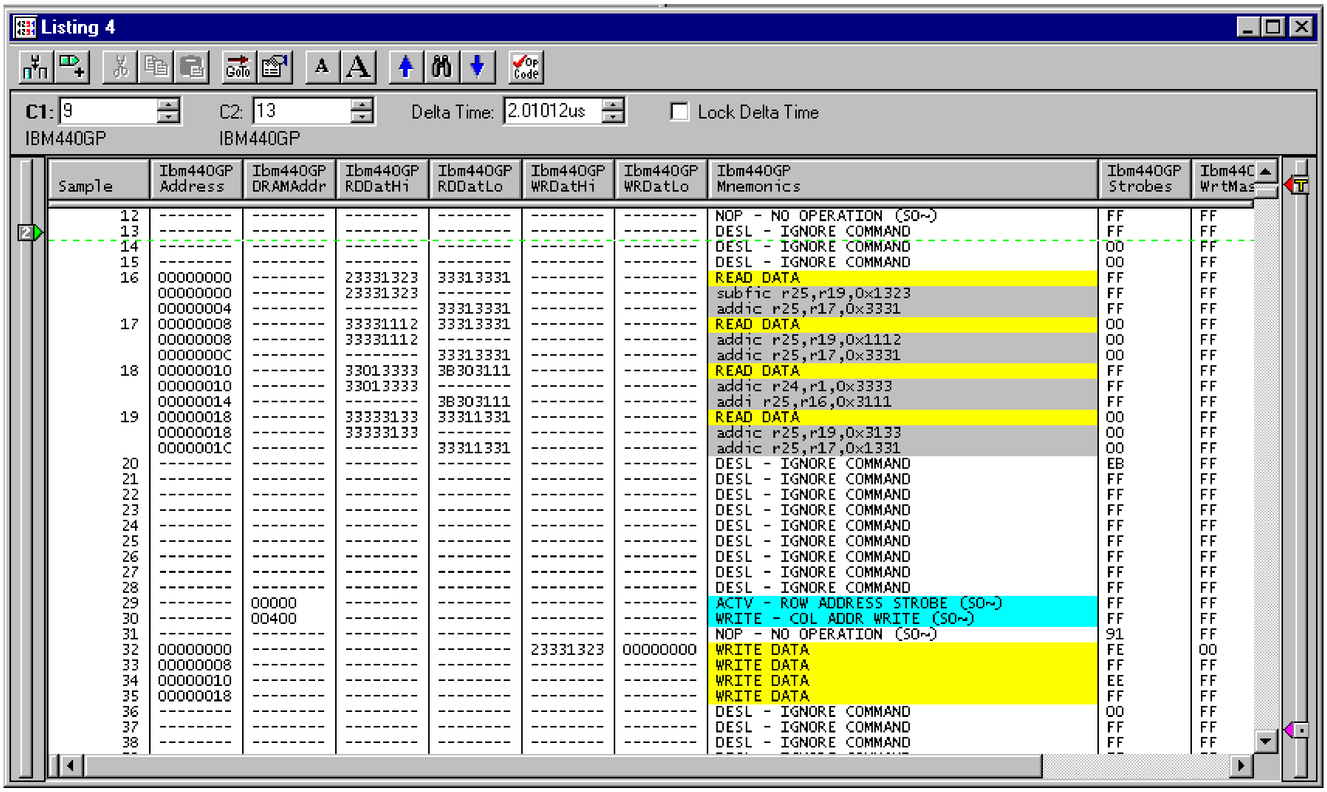
The TLA700, since it is a Windows program, has the same type of user interface as other Windows-based applications. In the Disassembly Listing window, a tool bar at the top of the window contains buttons that allow the user to modify the display. These buttons, from left to right, perform the following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



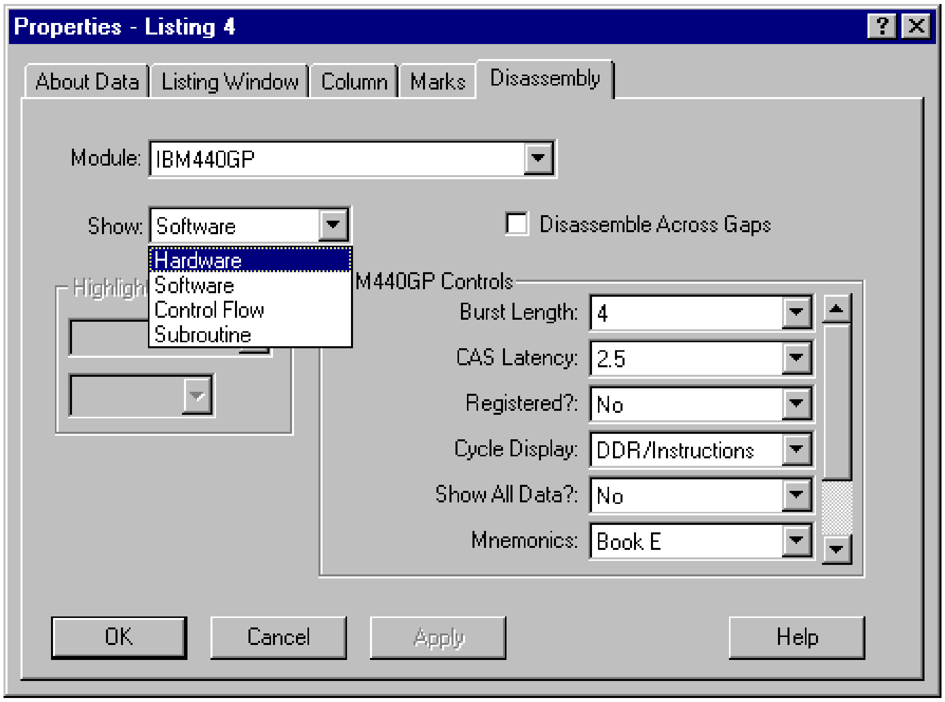
Listing 4

C1: 9 C2: 13 Delta Time: 2.01012us □ Lock Delta Time
IBM440GP IBM440GP

Sample	Ibm440GP Address	Ibm440GP DRAMAddr	Ibm440GP RDDatHi	Ibm440GP RDDatLo	Ibm440GP WRDatHi	Ibm440GP WRDatLo	Ibm440GP Mnemonics	Ibm440GP Strobes	Ibm440C WrtMas
12	--------	--------	--------	--------	--------	--------	NOP - NO OPERATION (SO~)	FF	FF
13	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
14	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	00	FF
15	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	00	FF
16	00000000	--------	23331323	33313331	--------	--------	READ DATA	FF	FF
	00000000	--------	23331323	--------	--------	--------	subfic r25,r19,0x1323	FF	FF
	00000004	--------	--------	33313331	--------	--------	addic r25,r17,0x3331	FF	FF
17	00000008	--------	33331112	33313331	--------	--------	READ DATA	00	FF
	00000008	--------	33331112	--------	--------	--------	addic r25,r19,0x1112	00	FF
	0000000C	--------	--------	33313331	--------	--------	addic r25,r17,0x3331	00	FF
18	00000010	--------	33013333	3B303111	--------	--------	READ DATA	FF	FF
	00000010	--------	33013333	--------	--------	--------	addic r24,r1,0x3333	FF	FF
	00000014	--------	--------	3B303111	--------	--------	addi r25,r16,0x3111	FF	FF
19	00000018	--------	33333133	33311331	--------	--------	READ DATA	00	FF
	00000018	--------	33333133	--------	--------	--------	addic r25,r19,0x3133	00	FF
	0000001C	--------	--------	33311331	--------	--------	addic r25,r17,0x1331	00	FF
20	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	EB	FF
21	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
22	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
23	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
24	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
25	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
26	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
27	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
28	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
29	--------	00000	--------	--------	--------	--------	ACTV - ROW ADDRESS STROBE (SO~)	FF	FF
30	--------	00400	--------	--------	--------	--------	WRITE - COL ADDR WRITE (SO~)	FF	FF
31	--------	--------	--------	--------	--------	--------	NOP - NO OPERATION (SO~)	91	FF
32	00000000	--------	--------	--------	23331323	00000000	WRITE DATA	FE	00
33	00000008	--------	--------	--------	--------	--------	WRITE DATA	FF	FF
34	00000010	--------	--------	--------	--------	--------	WRITE DATA	EE	FF
35	00000018	--------	--------	--------	--------	--------	WRITE DATA	FF	FF
36	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	00	FF
37	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF
38	--------	--------	--------	--------	--------	--------	DESL - IGNORE COMMAND	FF	FF

Figure 4- IBM440GP Disassembly

The other column of interest is the Mnemonics column, where the IBM440GP disassembly information is displayed. As mentioned previously, it is possible to filter the IBM440GP instructions that are displayed. This is done via selections made in the Disassembly tab of the Properties window (see Figure 4). 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 active SDRAM cycles and 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). The mnemonics display is further controlled by the "Cycle Display" select field; which allows display of only DDR cycle information, only IBM440GP Instruction information or both.



Module: IBM440GP

Show: Software

□ Disassemble Across Gaps

Burst Length: 4

CAS Latency: 2.5

Registered?: No

Cycle Display: DDR/Instructions

Show All Data?: No

Mnemonics: Book E

Figure 5- Disassembly Display Filter Window

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.

7.0 THE IBM440GP 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 is provided for this purpose for the Command group for the NEX-IBM440GP (IBM440GP_Cmd). As the memory cycles for the IBM440GP are, in fact, DDR memory cycles; this Symbol Table is the same as shown in Table 3 in section 6.1 (Viewing State Data on the TLA – General) of the NEX-DDRHS manual.

APPENDIX A – Support for Extended Mnemonics

The IBM440GP defines two sets of instruction codes defined in Section 28.5 of the IBM PPC440GP Embedded Processor User's Manual Preliminary. The Book E opcodes are the ones that are displayed by default in the disassembly listing. The IBM440GP disassembler also supports the Extended Mnemonics for the PowerPC family as listed in Section 28.5, subject to the qualifications detailed in this appendix:
The actual opcode binary between the two mnemonics is unchanged, simply the mnemonic text that is used to represent the opcode. Because there is no way for the IBM440GP software to know which opcodes the user prefers to view a select field has been added to allow switching from one to the other. Refer to Section 6.1 for information on how this switching is done.

The following Extended Mnemonics documented in Section 28.5 are supported:

Table 28-4 "Extended Mnemonics for addi"

la	addi RT,RA,D
li	addi RT,0,IM
subi	addi RT,RA,-IM

Table 28-5 "Extended Mnemonics for addic"

subic	addic RT,RA,-M

Table 28-6 "Extended Mnemonics for addic."

subic.	addic. RT,RA,-M

Table 28-7 "Extended Mnemonics for addis"

lis	addis RT,0,IM
subis	addis RT,RA,-IM

Table 28-8 "Extended Mnemonics for bc,bca,bcl,bcla"

NOTE: where duplicate extended mnemonics exist, they are shown in a merged form.
For example: bge(nl)

bdnz	bc 16,0,target
bdnza	bca 16,0,target
bdnzl	bcl 16,0,target
bdnzla	bcla 16,0,target
bdnzf	bc 0,cr_bit,target
bdnzfa	bca 0,cr_bit,target
bdnzfl	bcl 0,cr_bit,target
bdnzfla	bcla 0,cr_bit,target
bdnzt	bc 8,cr_bit,target
bdnzta	bca 8,cr_bit,target
bdnztl	bcl 8,cr_bit,target
bdnztla	bcla 8,cr_bit,target
bdz	bc 18,0,target
bdza	bca 18,0,target
bdzl	bcl 18,0,target

bdzla	bcla 18,0,target
bdzf	bc 2,cr_bit,target
bdzfa	bca 2,cr_bit,target
bdzfl	bcl 2,cr_bit,target
bdzfla	bcla 2,cr_bit,target
bdzt	bc 10,cr_bit,target
bdzta	bca 10,cr_bit,target
bdztl	bcl 10,cr_bit,target
bdztla	bcla 10,cr_bit,target
beq	bc 12,4*cr_field+2,target
beqa	bca 12,4*cr_field+2,target
beql	bcl 12,4*cr_field+2,target
beqla	bcla 12,4*cr_field+2,target
bf	bc 4,cr_bit,target
bfa	bca 4,cr_bit,target
bfl	bcl 4,cr_bit,target
bfla	bcla 4,cr_bit,target
bge(nl)	bc 4,4*cr_field+0,target
bge(nl)a	bca 4,4*cr_field+0,target
bge(nl)l	bcl 4,4*cr_field+0,target
bge(nl)la	bcla 4,4*cr_field+0,target
bgt	bc 12,4*cr_field+1,target
bgta	bca 12,4*cr_field+1,target
bgtl	bcl 12,4*cr_field+1,target
bgtla	bcla 12,4*cr_field+1,target
ble(ng)	bc 4,4*cr_field+1,target
ble(ng)a	bca 4,4*cr_field+1,target
ble(ng)l	bcl 4,4*cr_field+1,target
ble(ng)la	bcla 4,4*cr_field+1,target
blt	bc 12,4*cr_field+0,target
blta	bca 12,4*cr_field+0,target
bltl	bcl 12,4*cr_field+0,target
bltla	bcla 12,4*cr_field+0,target
bne	bc 4,4*cr_field+2,target
bnea	bca 4,4*cr_field+2,target
bnel	bcl 4,4*cr_field+2,target
bnela	bcla 4,4*cr_field+2,target
bngla	bcla 4,4*cr_field+1,target
bnlla	bcla 4,4*cr_field+0,target
bns(u)	bc 4,4*cr_field+3,target
bns(u)a	bca 4,4*cr_field+3,target
bns(u)l	bcl 4,4*cr_field+3,target
bns(u)la	bcla 4,4*cr_field+3,target
bnul	bcl 4,4*cr_field+3,target
bnula	bcla 4,4*cr_field+3,target
bso(un)	bc 12,4*cr_field+3,target
bso(un)a	bca 12,4*cr_field+3,target
bso(un)l	bcl 12,4*cr_field+3,target

bso(un)la	bcla 12,4*cr_field+3,target
bt	bc 12,cr_bit,target
bta	bca 12,cr_bit,target
btl	bcl 12,cr_bit,target
btla	bcla 12,cr_bit,target
bunl	bcl 12,4*cr_field+3,target
bunla	bcla 12,4*cr_field+3,target

Table 28-9 "Extended Mnemonics for bcctr, bcctrl"

NOTE: where duplicate extended mnemonics exist, they are shown in a merged form.
For example: bge(nl)ctr

bctr	bcctr 20,0
bctrl	bcctrl 20,0
beqctr	bcctr 12,4*cr_field+2
beqctrl	bcctrl 12,4*cr_field+2
bfctr	bcctr 4,cr_bit
bfctrl	bcctrl 4,cr_bit
bge(nl)ctr	bcctr 4,4*cr_field+0
bge(nl)ctrl	bcctrl 4,4*cr_field+0
bgtctr	bcctr 12,4*cr_field+1
bgtctrl	bcctrl 12,4*cr_field+1
ble(ng)ctr	bcctr 4,4*cr_field+1
ble(ng)ctrl	bcctrl 4,4*cr_field+1
bltctr	bcctr 12,4*cr_field+0
bltctrl	bcctrl 12,4*cr_field+0
bnectr	bcctr 4,4*cr_field+2
bnectrl	bcctrl 4,4*cr_field+2
bngctr	bcctr 4,4*cr_field+1
bngctrl	bcctrl 4,4*cr_field+1
bnlctrl	bcctrl 4,4*cr_field+0
bns(u)ctr	bcctr 4,4*cr_field+3
bns(u)ctrl	bcctrl 4,4*cr_field+3
bso(un)ctr	bcctr 12,4*cr_field+3
bso(un)ctrl	bcctrl 12,4*cr_field+3
btctr	bcctr 12,cr_bit
btctrl	bcctrl 12,cr_bit

Table 28-10 "Extended Mnemonics for bclr,bclrl"

NOTE: where duplicate extended mnemonics exist, they are shown in a merged form.
For example: bge(nl)lr

blr	bclr 20,0
blrl	bclrl 20,0
bdnzlr	bclr 16,0
bdnzlrl	bclrl 16,0
bdnzflr	bclr 0,cr_bit
bdnzflrl	bclrl 0,cr_bit
bdnztlr	bclr 8,cr_bit
bdnztlrl	bclrl 8,cr_bit

bdzlr	bclr 18,0
bdzlrl	bclrl 18,0
bdzflr	bclr 2,cr_bit
bdzflrl	bclrl 2,cr_bit
bdztlr	bclr 10,0
bdztlrl	bclrl 10,0
beqlr	bclr 12,4*cr_field+2
beqlrl	bclrl 12,4*cr_field+2
bflr	bclr 4,cr_bit
bflrl	bclrl 4,cr_bit
bge(nl)lr	bclr 4,4*cr_field+0
bge(nl)lrl	bclrl 4,4*cr_field+0
bgtlr	bclr 12,4*cr_field+1
bgtlrl	bclrl 12,4*cr_field+1
ble(ng)lr	bclr 4,4*cr_field+1
ble(ng)lrl	bclrl 4,4*cr_field+1
bltlr	bclr 12,4*cr_field+0
bltlrl	bclrl 12,4*cr_field+0
bnelr	bclr 4,4*cr_field+2
bnelrl	bclrl 4,4*cr_field+2
bns(u)lr	bclr 4,4*cr_field+3
bns(u)lrl	bclrl 4,4*cr_field+3
bso(un)lr	bclr 12,4*cr_field+3
bso(un)lrl	bclrl 12,4*cr_field+3
btlr	bclr 12,cr_bit
btlrl	bclrl 12,cr_bit

Table 28-11 "Extended Mnemonics for cmp"

cmpw	cmp BF,0.RA,RB

Table 28-12 "Extended Mnemonics for cmpi"

cmpwi	cmpi BF,0,RA,IM

Table 28-13 "Extended Mnemonics for cmpl"

cmplw	cmpl BF,0,RA,RB

Table 28-14 "Extended Mnemonics for cmpli"

cmplwi	cmpli BF,0,RA,IM

Table 28-15 "Extended Mnemonics for creqv"

crset	creqv bx,bx,bx

Table 28-16 "Extended Mnemonics for crnor"

crnot	crnor bx,by,by

Table 28-17 "Extended Mnemonics for cror"

crmove	cror bx,by,by

Table 28-18 "Extended Mnemonics for crxor"

crclr	crxor bx,bx,bx

Table 28-19 "Extended Mnemonics for mfspr"

mtccr0	mfspr RT, 0x3B3
mfcsrr0	mfspr RT, 0x03A
mfcsrr1	mfspr RT, 0x03B
mfctr	mfspr RT, 0x009
mfdac1	mfspr RT, 0x13C
mfdac2	mfspr RT, 0x13D
mfdbcr0	mfspr RT, 0x134
mfdbcr1	mfspr RT, 0x135
mfdbcr2	mfspr RT, 0x136
mfdbdr	mfspr RT, 0x3F3
mfdbsr	mfspr RT, 0x130
mfdcdbtrh	mfspr RT, 0x39D
mfdcdbtrl	mfspr RT, 0x39C
mfdear	mfspr RT, 0x03D
mfdec	mfspr RT, 0x016
mfdecar	mfspr RT, 0x036
mfdnv0	mfspr RT, 0x390
mfdnv1	mfspr RT, 0x391
mfdnv2	mfspr RT, 0x392
mfdnv3	mfspr RT, 0x393
mfdtv0	mfspr RT, 0x394
mfdtv1	mfspr RT, 0x395
mfdtv2	mfspr RT, 0x396
mfdtv3	mfspr RT, 0x397
mfdvc1	mfspr RT, 0x13E
mfdvc2	mfspr RT, 0x13F
mfdvlim	mfspr RT, 0x398
mfesr	mfspr RT, 0x03E
mfiac1	mfspr RT, 0x138
mfiac2	mfspr RT, 0x139
mfiac3	mfspr RT, 0x13A
mfiac4	mfspr RT, 0x13B
mficdbdr	mfspr RT, 0x3D3
mficdbtrh	mfspr RT, 0x39F
mficdbtrl	mfspr RT, 0x39E
mfinv0	mfspr RT, 0x370
mfinv1	mfspr RT, 0x371
mfinv2	mfspr RT, 0x372
mfinv3	mfspr RT, 0x373
mfitv0	mfspr RT, 0x374
mfitv1	mfspr RT, 0x375
mfitv2	mfspr RT, 0x376
mfitv3	mfspr RT, 0x377
mfivlim	mfspr RT, 0x399

mfivor0	mfspr RT, 0x190
mfivor1	mfspr RT, 0x191
mfivor2	mfspr RT, 0x192
mfivor3	mfspr RT, 0x193
mfivor4	mfspr RT, 0x194
mfivor5	mfspr RT, 0x195
mfivor6	mfspr RT, 0x196
mfivor7	mfspr RT, 0x197
mfivor8	mfspr RT, 0x198
mfivor9	mfspr RT, 0x199
mfivor10	mfspr RT, 0x19A
mfivor11	mfspr RT, 0x19B
mfivor12	mfspr RT, 0x19C
mfivor13	mfspr RT, 0x19D
mfivor14	mfspr RT, 0x19E
mfivor15	mfspr RT, 0x19F
mfivpr	mfspr RT, 0x03F
mflr	mfspr RT, 0x008
mfmmucr	mfspr RT, 0x3B2
mfpid	mfspr RT, 0x030
mfpir	mfspr RT, 0x11E
mfpvr	mfspr RT, 0x11F
mfrstcfg	mfspr RT, 0x39B
mfsprg0	mfspr RT, 0x110
mfsprg1	mfspr RT, 0x111
mfsprg2	mfspr RT, 0x112
mfsprg3	mfspr RT, 0x113
mfsprg4	mfspr RT, 0x104; mfspr rt, 0x114
mfsprg5	mfspr RT, 0x105; mfspr RT, 0x115
mfsprg6	mfspr RT, 0x106; mfspr RT, 0x116
mfsprg7	mfspr RT, 0x107; mfspr RT, 0x117
mfsrr0	mfspr RT, 0x01A
mfsrr1	mfspr RT, 0x01B
mftbl	mfspr RT, 0x10C; mfspr RT, 0x11C
mftbu	mfspr RT, 0x10D; mfspr RT, 0x11D
mftcr	mfspr RT, 0x154
mftsr	mfspr RT, 0x150
mfusprg0	mfspr RT, 0x100
mfxer	mfspr RT, 0x001

Table 28-20 "Extended Mnemonics for mtcrf"

mtcr	mtcrf 0xFF,RS

Table 28-21 "Extended Mnemonics for mtspr"

mtccr0	mtspr RT, 0x3B3
mtcsrr0	mtspr RT, 0x03A
mtcsrr1	mtspr RT, 0x03B
mtctr	mtspr RT, 0x009

mtdac1	mtspr RT, 0x13C
mtdac2	mtspr RT, 0x13D
mtdbcr0	mtspr RT, 0x134
mtdbcr1	mtspr RT, 0x135
mtdbcr2	mtspr RT, 0x136
mtdbdr	mtspr RT, 0x3F3
mtdbsr	mtspr RT, 0x130
mtdcdbtrh	mtspr RT, 0x39D
mtdcdbtrl	mtspr RT, 0x39C
mtdear	mtspr RT, 0x03D
mtdec	mtspr RT, 0x016
mtdecar	mtspr RT, 0x036
mtdnv0	mtspr RT, 0x390
mtdnv1	mtspr RT, 0x391
mtdnv2	mtspr RT, 0x392
mtdnv3	mtspr RT, 0x393
mtdtv0	mtspr RT, 0x394
mtdtv1	mtspr RT, 0x395
mtdtv2	mtspr RT, 0x396
mtdtv3	mtspr RT, 0x397
mtdvc1	mtspr RT, 0x13E
mtdvc2	mtspr RT, 0x13F
mtdvlim	mtspr RT, 0x398
mtesr	mtspr RT, 0x03E
mtiac1	mtspr RT, 0x138
mtiac2	mtspr RT, 0x139
mtiac3	mtspr RT, 0x13A
mtiac4	mtspr RT, 0x13B
mticdbdr	mtspr RT, 0x3D3
mticdbtrh	mtspr RT, 0x39F
mticdbtrl	mtspr RT, 0x39E
mtinv0	mtspr RT, 0x370
mtinv1	mtspr RT, 0x371
mtinv2	mtspr RT, 0x372
mtinv3	mtspr RT, 0x373
mtitv0	mtspr RT, 0x374
mtitv1	mtspr RT, 0x375
mtitv2	mtspr RT, 0x376
mtitv3	mtspr RT, 0x377
mtivlim	mtspr RT, 0x399
mtivor0	mtspr RT, 0x190
mtivor1	mtspr RT, 0x191
mtivor2	mtspr RT, 0x192
mtivor3	mtspr RT, 0x193
mtivor4	mtspr RT, 0x194
mtivor5	mtspr RT, 0x195
mtivor6	mtspr RT, 0x196
mtivor7	mtspr RT, 0x197

mtivor8	mtspr RT, 0x198
mtivor9	mtspr RT, 0x199
mtivor10	mtspr RT, 0x19A
mtivor11	mtspr RT, 0x19B
mtivor12	mtspr RT, 0x19C
mtivor13	mtspr RT, 0x19D
mtivor14	mtspr RT, 0x19E
mtivor15	mtspr RT, 0x19F
mtivpr	mtspr RT, 0x03F
mtlr	mtspr RT, 0x008
mtmmucr	mtspr RT, 0x3B2
mtpid	mtspr RT, 0x030
mtpir	mtspr RT, 0x11E
mtpvr	mtspr RT, 0x11F
mtrstcfg	mtspr RT, 0x39B
mtsprg0	mtspr RT, 0x110
mtsprg1	mtspr RT, 0x111
mtsprg2	mtspr RT, 0x112
mtsprg3	mtspr RT, 0x113
mtsprg4	mtspr RT, 0x104; mtspr rt, 0x114
mtsprg5	mtspr RT, 0x105; mtspr RT, 0x115
mtsprg6	mtspr RT, 0x106; mtspr RT, 0x116
mtsprg7	mtspr RT, 0x107; mtspr RT, 0x117
mtsrr0	mtspr RT, 0x01A
mtsrr1	mtspr RT, 0x01B
mttbl	mtspr RT, 0x10C; mtspr RT, 0x11C
mttbu	mtspr RT, 0x10D; mtspr RT, 0x11D
mttcr	mtspr RT, 0x154
mttsr	mtspr RT, 0x150
mtusprg0	mtspr RT, 0x100
mtxer	mtspr RT, 0x001

Table 28-22 "Extended Mnemonics for nor,nor."

not	nor RA,RS,RS
not.	nor. RA,RS,RS

Table 28-23 "Extended Mnemonics for or,or."

mr	or RT,RS,RS
mr.	or. RT,RS,RS

Table 28-24 "Extended Mnemonics for ori"

nop	ori 0,0,0

Table 28-25 "Extended Mnemonics for rlwimi,rlwimi."

NOTE: the specification of left shift or right shift encoding for the rlwimi instruction is indeterminate; therefore all variants are shown as ins(lr)wi or ins(lr)wi.

Ins(lr)wi	rlwimi RA,RS,32-b,b,b+n-1

Ins(lr)wi.	rlwimi. RA,RS,32-b,b,b+n-1
Ins(lr)wi	rlwimi RA,RS,32-b-n,b,b+n-1
Ins(lr)wi.	rlwimi. RA,RS,32-b-n,b,b+n-1

Table 28-26 "Extended Mnemonics for rlwinm,rlwinm."

NOTE: the specification of left shift or right shift encoding for the rotlwi and rotrwi mnemonics are indeterminate; therefore they are shown as rot(lr)wi or rot(lr)wi. Also, the slwi and srwi mnemonics are indistinguishable from the extlwi and extrwi; therefore they are displayed as extlwi(slwi) and extrwi(srwi).

clrlwi	rlwinm RA,RS,0,n,31
clrlwi.	rlwinm. RA,RS,0,n,31
clrlswi	rlwinm RA,RS,n,b-n,31-n
clrlswi.	rlwinm. RA,RS,n,b-n,31-n
clrrwi	rlwinm RA,RS,0,0,31-n
clrrwi.	rlwinm. RA,RS,0,0,31-n
extlwi(slwi)	rlwinm RA,RS,b,0,n-1
extlwi(slwi).	rlwinm. RA,RS,b,0,n-1
extrwi(srwi)	rlwinm RA,RS,b+n,32-n,31
extrwi(srwi).	rlwinm. RA,RS,b+n,32-n,31
rot(lr)wi	rlwinm RA,RS,n,0,31
rot(lr)wi.	rlwinm. RA,RS,n,0,31
rot(lr)wi	rlwinm RA,RS,32-n,0,31
rot(lr)wi.	rlwinm. RA,RS,32-n,0,31
extlwi(slwi)	rlwinm RA,RS,n,0,31-n
extlwi(slwi).	rlwinm. RA,RS,n,0,31-n
extrwi(srwi)	rlwinm RA,RS,32-n,n,31
extrwi(srwi).	rlwinm. RA,RS,32-n,n,31

Table 28-27 "Extended Mnemonics for rlwnm,rlwnm."

rotlw	rlwnm RA,RS,RB,0,31
rotlw.	rlwnm. RA,RS,RB,0,31

Table 28-28 "Extended Mnemonics for subf,subf.,subfo,subfo."

sub	subf RT,RB,RA
sub.	subf. RT,RB,RA
subo	subfo RT,RB,RA
subo.	subfo. RT,RB,RA

Table 28-29 "Extended Mnemonics for subfc, subfc.,subfco,subfco."

subc	subfc RT,RB,RA
subc.	subfc. RT,RB,RA
subco	subfco RT,RB,RA
subco.	subfco. RT,RB,RA

Table 28-30 "Extended Mnemonics for tw"

trap	tw 31,0,0
tweq	tw 4,RA,RB

twge	tw 12,RA,RB
twgt	tw 8,RA,RB
twle	tw 20,RA,RB
twlge	tw 5,RA,RB
twlgt	tw 1,RA,RB
twlle	tw 6,RA,RB
twllt	tw 2,RA,RB
twlng	tw 6,RA,RB
twlnl	tw 5,RA,RB
twlt	tw 16,RA,RB
twne	tw 24,RA,RB
twng	tw 20,RA,RB
twnl	tw 12,RA,RB

Table 28-31 "Extended Mnemonics for twi"

tweqi	twi 4,RA,IM
twgei	twi 12,RA,IM
twgti	twi 8,RA,IM
twlei	twi 20,RA,IM
twlgei	twi 5,RA,IM
twlgti	twi 1,RA,IM
twllei	twi 6,RA,IM
twllti	twi 2,RA,IM
twlngi	twi 6,RA,IM
twlnli	twi 5,RA,IM
twlti	twi 16,RA,IM
twnei	twi 24,RA,IM
twngi	twi 20,RA,IM
twnli	twi 12,RA,IM

APPENDIX B – Address Translation and Address Maps

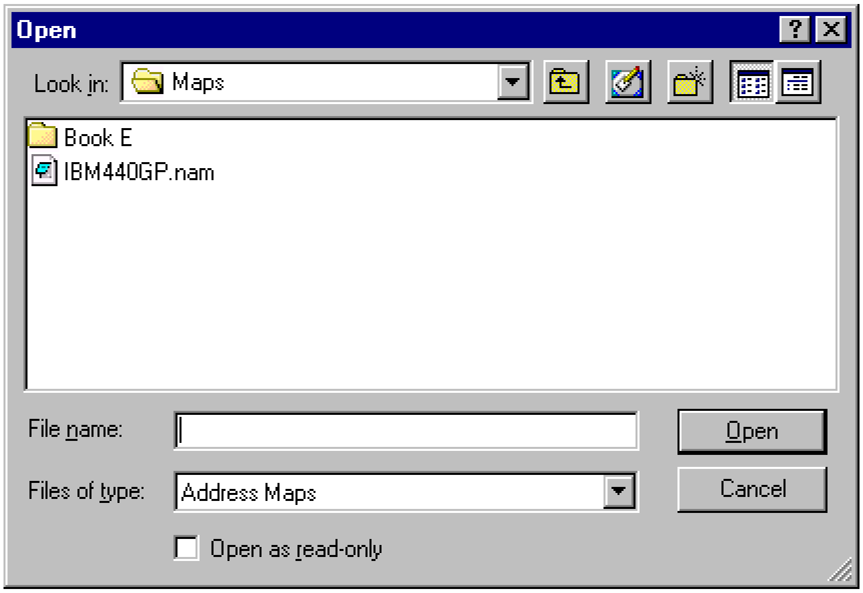
The SDRAM Address lines, which are present in the acquisition from the memory bus, consist of the Row and Column addresses during the Row Address Strobe (RAS) and Column Address Strobe (CAS) cycles. This represents the way a linear address is translated to utilize Dynamic RAM. Although a complete explanation 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, the IBM PPC440GP Embedded Processor User's Manual Preliminary identifies 8 supported memory arrangements in Table 17-12 (32-bit) and 17-13 (64-bit). 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 IBM440GP support, this directory is:

C:\Program Files\TLA 700\Supports\Ibm440GP\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 3), the following dialog will be displayed:



Look in: Maps

Book E

IBM440GP.nam

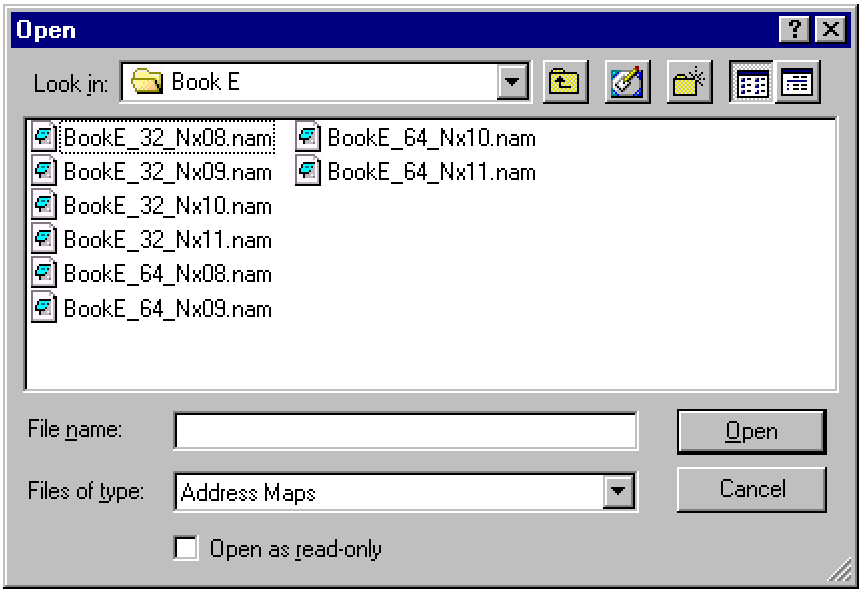
File name:

Files of type: Address Maps

☐ Open as read-only

Figure 6- Address Maps Directory 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, IBM440GP.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 "Book E" subdirectory, which will display the dialog below:



Look in: Book E

BookE_32_Nx08.nam
BookE_32_Nx09.nam
BookE_32_Nx10.nam
BookE_32_Nx11.nam
BookE_64_Nx08.nam
BookE_64_Nx09.nam
BookE_64_Nx10.nam
BookE_64_Nx11.nam

File name:

Files of type: Address Maps

□ Open as read-only

Figure 7- Available Address Maps Display

This shows the available 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 C - 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 D - References

Tektronix TLA700 System User’s Manual

Tektronix TLA700 Module User’s Manual

Tektronix P6434 Mass Termination Probe Instruction Manual

Tektronix P6810, P6860 and P6880 Logic Analyzer Probes Instruction Manual

IBM PPC440GP Embedded Processor User’s Manual Preliminary.
Document SA14-2519-11; 03/2002; Seventh Preliminary Edition.
Copyright International Business Machines, Inc. 2000,2001.

Book E: Enhanced PowerPC Architecture"
Version 0.90, March 23, 2000.
Copyright International Business Machines, Inc. 1993,2000.