### **Trace Tutorial Overview**

The objective of this tutorial is to acquaint you with the basic use of the Trace System software. The Trace System software includes the following:

- The Trace Control The Trace Control allows you to set trace collection related options, such as synchronizing trace collection with target execution, whether to stall the CPU if the on-chip trace FIFOs become full, and so on.
- The Trace Display The Trace Display presents the collected trace data in columns and provides capabilities to search, bookmark, save and export the data.
- The CCStudio Breakpoint Manager and Event Sequencer These tools allow you to create Advanced Event Triggering actions that determine when and how to collect trace data, as well as what data to collect.

The following assumptions are made for this tutorial:

- This tutorial is based on using Code Composer Studio Version 3.3 using a Spectrum Digital 6416 DSK target board, with a TI XDS560T Trace Cable and a Blackhawk USB emulator.
- The Trace System user interface may vary depending upon your version of Code Composer Studio, target board, and emulator. The tutorial attempts to be as agnostic of these issues as possible.
- The tutorial assumes that you have already set up Code Composer Studio, the target board, and the emulator as described in the appropriate documentation.

Note: If you have previously run this tutorial or have used the Trace System software before running this tutorial, the screen shots in the tutorial may be different than what you see on your screen. This can be due to your having previously set options other than what the tutorial covers, and or hiding Trace Display columns, which is a setting preserved across Trace Display invocations.

### 

### Initializing the Target System

To initialize a target system:

- 1. Power the target board down and back up again.
- 2. Start Code Composer Studio.
- 3. Connect to the target by selecting Debug ® Connect.
- 4. Remove any existing breakpoints and reset the External Memory Interface by selecting GEL ® Resets ® Reset\_BreakPts\_and\_EMIF. (The exact menu item text may be different for specific devices).
- 5. Flush the on-chip cache by selecting GEL ® Resets ® Flush Cache. (The exact menu item text may be different for specific devices).
- Load the tutorial project located at CCStudioInstall \tutorial\trace\Device \tutorial\tutorial.pjt with Project 
   Open... (For example: CCStudioInstall\tutorial\trace\64x\tutorial\tutorial.pjt)
- 7. Rebuild the tutorial project by selecting Project ® Rebuild All.
- 8. Load the target program by selecting File ( Load Program... then browse to the Debug directory and select tutorial.out.
- 9. Click Open.



### How to Trace Your Target Application

The following steps demonstrate how to use the CCStudio Breakpoint Manager to program the on-chip AET hardware to collect Program Address and cycle accurate Timestamp information from your target application by using the Trace On Action.

1. When initialization is complete, start the Breakpoint Manager by selecting Tools ® XDS560 Trace ® Setup... or Debug ® Breakpoints. The Breakpoint Manager opens at the bottom of the CCStudio window.

<b>``</b> ■   % <b>B B</b>   ∾ ∩		in 3nn 9an \is   @ №	(1+1) (1 (1 = 1 / * %	- 7
torial.pjt 💽 Debug	- & 🕮 📇 🕘 🗶 -	• <u>\$</u>		
2 & <u>0</u> = = = = [	e d			
Files E GEL files Projects E D tutorial.pjt (Debu	00004DA0         c_int0	0: MVK.S2 MVKH.S2 AND.D2 MVK.S2 MVKH.S2 MVK.S2 MVK.S2 MVK.S1 ADDKPC.S2 MVKH.S1 MVKH.S1 MVKH.S1 NOP B.S2X ADDKPC.S2	0x47fc,SP 0x0000,SP -8,SP,SP 0x4fc0,DP 0x0000,DP 0x4ac0,B4 0x0000,B4 B4 0x4e80,A4 0x4e80,A4 0x4e00,A3 0x0000,A3 0x0000,A3 A3 0x4DE0 (PC+32 = 0x000	
New • 🔤 🎭 • 💊	- 📑 - 🗐 🚰 Columns 🐴 - 🗃 🛱			
Location	Condition Logical Action	Group	Name	
	L <sub>6</sub>			

2. Click New item dropdown list (black triangle next to the New icon in the toolbar) in the Breakpoint Manager and select Trace.

Hardware Breakpoint Chained Breakpoint Count	Condition Logica	al Action
Trace Watchdog Timer Watchpoint Hardware Breakpoint Software Breakpoint		
ALTED		ERR

3. The Property window for the Trace action should open automatically in the right side of the CCStudio window. If it does not automatically open, click on the Open Property Window icon (

<b>₽</b> ₽ % <b>₽</b> € ·					(= +=   #(= #(=   = =	
orial.pjt 🗾 🗾 🗖	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	• 🖉 🖄 🛗	X 0 8	<u>S</u>		
66°     Image: Constraint of the second	(Debug)	00004DA0 00004DA0 00004DA8 0787 00004DA8 07271 00004DB0 07000 00004DB8 02000 00004DB8 00004DB8 00004DC 00004DC 00004DC 00004DC 00004DC 0180 00004DC 0180 00004DC 0180 00004DC 0180 00004DC 0180 00004DC 0180 00004DB 00004D0 000000 00004D0 00000 00004D0 00000 000000 0000000 0000000 00000000	106A 19F2 502A 502A 502A 502A 3362 4028 4028 4162 5068 5028 5068 5028 5068 5028 5068 5028 5068 5028 5068 5028 5028 5028 5028 5028 5028 5028 502	MVK.S2 MVKH.S2 AND.D2 MVK.S2 MVK.S2 MVK.S2 MVK.S1 ADDKPC.S2 MVKH.S1 MVKH.S1 MVKH.S1 NOP B.S2X ADDKPC.S2	Oxi     Brewing       -8     Brewing       0xi     Physical Propert       0xi     Constant       0xi     Constant       0xi     Constant       0xi     Misc       0xi     Misc       0xi     Constant       0xi     Constant       0xi     Misc       0xi     Constant       0xi     Constant       0xi     Constant       0xi     Constant       0xi     Constant	rties Trace
New - 🍇 🍇 🗣 -	1152 55215		• 🗃 🖫 Action	Guarda	Mawa	
Location		ion Logical	Action	Group	Name Gr Trace	

4. In the Property window, expand the Type and What to Trace properties by clicking on the plus sign to the left of those properties. Under Actions, you can see that the default Trace Action is Trace On.



5. Make sure that the Program Address and Timing Stamp properties are checked under What to Trace, as well as the Enabled property. Finally, click on the Submit Changes icon in the Properties window toolbar to apply the changes.

E	Ph Submit Changes	
Ξ	Туре	Trace
	Actions	Trace On
Ξ	What to Trace	
	Program Address	🔽 True
	Timing Stamp	🔽 True
	Write Address	T False
	Write Data	T False
	Read Address	T False
	Read Data	☐ False
	PC with Data	T False
	Logical Properties	
	Group	Default Group
	Name	Trace
Ξ	Misc	
	Enable	🔽 Enabled

Note that the Code Composer Studio Status Bar updates with a Trace Initialization progress bar. All CCS operations are prevented while trace is initializing. When this progress bar disappears, the trace system is finished initializing and you may continue with CCS operations.

LE	C64xx XD5560 Emulator/CPU_1 : Trace Initializing

When initialization is complete, the Trace Display window automatically appears in front of the Code Composer Studio window. The Trace Display determines the correct target .out file for the loaded project (required to disassemble the collected trace data), programs the on-chip AET Hardware, initializes and attaches the receiver to the CPU, and creates a trace window for that CPU. The Trace Display's status bar changes to indicate that it is in the recording state Receiver recording trace data because the Trace On action starts recording data as soon as it is applied.

💀 Trace Display - [Blackhawk USB560m 🛛 - C6416 DSK F	mulator/TMS320C6400_0]		
File View Window Help			_ 8 ×
Stop Resume Query -> C Query	Filter Fields Save Dverlags	Bandwidth Info	
Program Load Ad Cycles (abs) [cyc] Di	sassembly Source		Trace Status
Tr	No trace data available now ace hardware is collecting the Trace buffer is empty.	trace.	
		Lς.	
Receiver recording trace data		Device Clock: 1000.000Mhz Trac	a Data: 222 200Mba
Receiver recording trace data Ready		One traceable CPU connect	

6. In Code Composer Studio, open the source file tutorial.c and set a software breakpoint at line 45, ccc++.

	int i;	
	<pre>for (i = 0; i &lt; 50; i++) {</pre>	
•	ccc++;	
1120	}	
	while(a)	
	xxx++;	
	yyy++;	
	cnt = 1;	
	<pre>VoidFunc();</pre>	
	cnt++;	
	VoidFuncWithArgs(a, &ret);	
	cnt++;	
6-10-1	ret = IntFunc():	
4		•

7. Run the target by selecting Debug ® Run.

When the target hits the breakpoint, the Trace System stops recording and the Trace Display appears automatically and displays data. This behavior is due to the default Synchronize Trace with Target Execution option. The exact columns that appear in the display depend upon the target device.

The same of the second se	Window Help			Fields Save Overlays Ba	indwidth Info	- 8 %
	lesume	Query  Cycles (abs) [cyc]	Query Filter.		Trace Status	
Flogram	LUQU AU	caries lanst inari	Disassembly		Start of trace	
					36410 01 01866	-
00004DA0	00004DA0		0	MVK.S2 boot.c(27)	PC collection on, T	`iming c
00004DA4	00004DA4		1	MVKH.S2		
00004DA8	00004DA8		2	AND.D2		
00004DAC	00004DAC		3	MVK.S2		
00004DB0	00004DB0		4	MVKH.S2		
00004DB4	00004DB4		5	MVK.S2		
00004DB8	00004DB8		6	MVKH.S2	Pipeline stall	
00004DB8	00004DB8	1	5	MVKH.S2		
00004DBC	00004DBC			B.S2		
00004DC0	00004DC0	1		MVK.S1		
00004DC4	00004DC4	1		ADDKPC.S		
00004DC8	00004DC8	2		MVKH.S1	Pipeline stall	
00004DC8	00004DC8	2	3	MVKH.S1		
00004AC0	00004AC0	2	4	OR.L1X autoinit.c(15	)	
00004AC4	00004AC4		6 F 1	STW.D2T1		
00004AC8	00004AC8		EE	NOP		
00004ACC	00004ACC			NOP		
00004AD0	00004AD0		i i	NOP		
00004AD4	00004AD4	2	5	STW.D1T1		
00004AD8	00004AD8		EE.	STDW.D2T		
00004ADC	00004ADC		- 11	OR.12		
00004AE0	00004AE0	2	6	OR.S1	Pipeline stall	
00004AE0	00004AE0	3		OR.S1	10	
00004AE4	00004AE4		EE.	CMPEQ.L1.		
00004AE8	00004AE8	3	6 [ AO]	BNOP.S2		
00004AEC	00004AEC			OR.11		
00004AF0	00004AF0			OR.S1		
00004AF4	00004AF4		1 [IA01	LDW.D1T1		
COOCTAL 4			[] []	್ಷ ಹೆಚ್ಚಿದ್ದ ಮಾಡಿದ್ದ ಮ		>
	dere be mennen a	nere standarder detrated w	and huffen to tool	6.JI	Deuter Clash, 1000 000Mbs Tures Date, 000 000Mbs	1000
and a state of the second second second second	que to user req	uest , New samples detected, T	race burrer 10,15%	- ruii	Device Clock: 1000.000Mhz Trace Data: 333.300Mhz Sa	ample 1 of 61
eady					One traceable CPU connected	

The status bar at the bottom of the Trace Display window indicates Receiver stopped due to user request. This is due to the default Synchronize Trace with Target Execution option.

# ◄►

### Hiding and Reordering Trace Columns

By default, the Trace Display shows the columns of trace data for the most typical debug use cases. It is possible that your debug use case will require collecting and analyzing additional types of trace data. The Trace Display allows you to hide, add, or reorder the columns that are displayed.

1. To hide columns, click on the Fields... button.

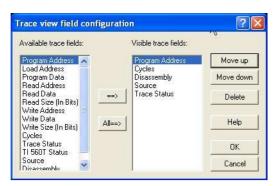
race view field conf		?
Available trace fields:	Visible trace fields:	48
Program Address 🔼	Program Address	Move up
Load Address Program Data Bead Address	Load Address Cycles Disassembly	Move down
Read Data Read Size (In Bits)	==> Disassembly Source Trace Status	Delete
Write Address	indee status	
Write Size (In Bits) Cycles	All==>	Help
Trace Status TI 560T Status		OK
Source		Cancel

Note: The available trace fields list varies based on the target device.

Since the tutorial project does not use software overlays, this field is not necessary for analysis of the trace data. Remove the Load Address field by selecting the Load Address field under the right hand pane of the dialog box (Visible trace fields), and then clicking on the Delete button. The dialog should now resemble the following:

race view field config	uration	?
Available trace fields:	Visible trace fields:	
Program Address	Program Address Cycles	Move up
Program Data	Disassembly	Move down
Read Address Read Data Read Size (In Bits) Write Address	==> Source Trace Status	Delete
Write Data		Help
Trace Status		OK
Source		Cancel

3. You can reorder the fields by selecting the fields in the right hand pane and clicking the Move up or Move down buttons so that the columns appear in the order that makes analysis of the trace data easiest for you. You can add fields by selecting them in the "Available trace fields" list, and then clicking on the => button to move them to the "Visible trace fields" list. For this tutorial, you should not add or reorder the fields being displayed; so be sure that the fields dialog appears as below when you have finished experimenting with adding and reordering fields.



4. Click OK to dismiss the Fields dialog.

The Trace Display columns should now appear as below. (The data may differ.)

	File View Window H	Query V	C Query	Fields Save Overlays	Bandwidth Info	_ 8 :
Start of trace           00004DA0         0         MVK.52         boot.c(27)         PC collection on, Timing collection           00004DA4         1         MVK.52         PC collection on, Timing collection           00004DA6         2         AND.D2         PC collection on, Timing collection           00004DA6         3         MVK.52         PC collection on, Timing collection           00004DB0         4         MVK.52         PC collection on, Timing collection           00004DB1         5         MVK.52         PC collection on, Timing collection           00004DB2         4         MVK.52         PC collection           00004DB3         5         MVK.52         PC collection           00004DC4         16         B.52         PC collection           00004DC4         18         ADDRFC.5.         PC collection           00004AC0         24         OR.11X         autoinit.cc(15)           00004AC0         24         OR.11X         autoinit.cc(15)           00004AC0         11         NOP         PC collection           00004AC0         11         NOP         PC collection           00004AC0         11         NOP         PC collection           00004AC0 <t< th=""><th></th><th></th><th></th><th></th><th></th><th></th></t<>						
00004DA0         0         MVK.S2         boot.c(27)         PC collection on. Timing collection           00004DA4         1         MVK.S2         MVK.S2         MVK.S2           00004DAC         3         MVK.S2         MVK.S2         MVK.S2           00004DB0         4         MVKH.S2         MVK.S2         MVK.S2           00004DB0         6         MVKH.S2         MVK.S2         MVK.S2           00004DB8         6         MVKH.S2         MVK.S2         MVK.S2           00004DC0         16         B.S2         MVKH.S1         MVKH.S1           00004DC4         18         ADDKPC.S         MVKH.S1         MVKH.S1           00004AC0         24         OR.I1X         autoinit.c(15)         MVKH.S1           00004AC4         11         STW.D271         MVC         MVKH.S2           00004AC4         11         STW.D271         MVC         MVKH.S1           00004AC0         24         OR.I1X         autoinit.c(15)         MVKH.S2           00004AC4         11         STW.D271         MVC         MVC           00004AC5         11         NOP         MVC         MVC           00004AC6         11         NOP <td< th=""><th>Flugrann   Cycles ja</th><th></th><th>scilluly</th><th>JULIC</th><th></th><th>N</th></td<>	Flugrann   Cycles ja		scilluly	JULIC		N
00004DA4       1       MVKH.S2         00004DAC       3       MVK.S2         00004DB0       4       MVKH.S2         00004DB4       5       MVK.S2         00004DB5       6       MVKH.S2         00004DB6       6       MVKH.S2         00004DB7       16       B.S2         00004DC4       18       ADDKPC.S.         00004DC8       21       MVKH.S1         00004AC4       1       STW.DZTI         00004AC4       1       STW.DZTI         00004AC4       11       NOP         00004AC4       11       NOP         00004AD5       11       NOP         00004AC4       11       NOP         00004AD5       11       NOP         00004AD6       12       STW.DITI         00004AD6       12       OR.L2         00004AD7       10       OR.L2         00004AD6       25       STW.DITI         00004AD6       12       OR.L2         00004AD6       13       OR.S1         00004AE7       11       OMPC.S2         00004AE8       36       IAO         00004AE8       36						
00004DB0       4       MVK.S2         00004DB8       6       MVKH.S2         00004DB8       15       MVKH.S2         00004DC0       16       B.S2         00004DC0       17       MVK.S1         00004DC3       21       MVKH.S1         00004DC4       18       ADDKPC.S         00004DC5       23       MVKH.S1         00004AC0       24       OR.L1X         00004AC3       11       STW.D2T1         00004AC4       18       STW.D2T1         00004AC5       11       NOP         00004AC6       11       NOP         00004AC6       11       NOP         00004AC6       11       NOP         00004AD4       25       STW.D1T1         00004AD5       11       OR.L2         00004AD6       15       OR.S1         00004AE0       35       OR.S1         00004AE0       35       OR.S1         00004AE6       11       OP.S2         00004AE6       14       OP.S1         00004AE7       11       OP.S1         00004AE7       11       OP.S1		0			PC collection on,	Timing collecti
00004DB0       4       MVK.S2         00004DB8       6       MVKH.S2         00004DB8       15       MVKH.S2         00004DC0       16       B.S2         00004DC0       17       MVK.S1         00004DC3       21       MVKH.S1         00004DC4       18       ADDKPC.S         00004DC5       23       MVKH.S1         00004AC0       24       OR.L1X         00004AC3       11       STW.D2T1         00004AC4       18       STW.D2T1         00004AC5       11       NOP         00004AC6       11       NOP         00004AC6       11       NOP         00004AC6       11       NOP         00004AD4       25       STW.D1T1         00004AD5       11       OR.L2         00004AD6       15       OR.S1         00004AE0       35       OR.S1         00004AE0       35       OR.S1         00004AE6       11       OP.S2         00004AE6       14       OP.S1         00004AE7       11       OP.S1         00004AE7       11       OP.S1		1				
00004DB0       4       MVK.S2         00004DB8       6       MVKH.S2         00004DB8       15       MVKH.S2         00004DC0       16       B.S2         00004DC0       17       MVK.S1         00004DC3       21       MVKH.S1         00004DC4       18       ADDKPC.S         00004DC5       23       MVKH.S1         00004AC0       24       OR.L1X         00004AC3       11       STW.D2T1         00004AC4       18       STW.D2T1         00004AC5       11       NOP         00004AC6       11       NOP         00004AC6       11       NOP         00004AC6       11       NOP         00004AD4       25       STW.D1T1         00004AD5       11       OR.L2         00004AD6       15       OR.S1         00004AE0       35       OR.S1         00004AE0       35       OR.S1         00004AE6       11       OP.S2         00004AE6       14       OP.S1         00004AE7       11       OP.S1         00004AE7       11       OP.S1		2				
00004DB4         5         MVK.52         Pipeline stall           00004DB8         6         MVKH.52         Pipeline stall           00004DB6         15         MVKH.52         Pipeline stall           00004DC0         17         MVK.51         Pipeline stall           00004DC3         21         MVKH.51         Pipeline stall           00004C4         18         ADDKPC.S.         Pipeline stall           00004C3         23         MVKH.S1         Pipeline stall           00004AC0         24         OR.LIX         autoinit.c(15)           00004AC3         11         NOP         Pipeline stall           00004AC4         11         STW.D2T1         Pipeline stall           00004AC4         11         NOP         Pipeline stall           00004AC5         11         NOP         Pipeline stall           00004AD0         11         NOP         Pipeline stall           00004AD5         11         OR.L2         Pipeline stall           00004AE0         25         GR S1         Pipeline stall           00004AE0         35         OR S1         Pipeline stall           00004AE4         11         CMPEQ.L1         Pipeline stall     <		3				
00004DB8       15       MVKH.S2         00004DC0       16       B.S2         00004DC0       17       MVK.S1         00004DC3       18       ADDKPC.S         00004DC4       18       ADDKPC.S         00004DC5       21       MVKH.S1         00004AC0       24       OR.L1X       autoinit.c(15)         00004AC4       11       STW.D2T1         00004AC5       11       NOP         00004AC6       11       NOP         00004AC7       11       NOP         00004AC8       11       NOP         00004AC4       10       NOP         00004AD4       25       STW.D1T1         00004AD4       25       STW.D1T1         00004AD4       25       OR.S1       Pipeline stall         00004AD6       35       OR.S1       Pipeline stall         00004AE0       36       [ A0]       BNOP.S2       OULL         00004AE6       11       OR.S1          00004AE7       11       OR.S1          00004AE0       36       [ A0]       BNOP.S2          00004AE7       11       OR.S1		4				
00004DB8       15       MVKH.S2         00004DC0       16       B.S2         00004DC0       17       MVK.S1         00004DC3       18       ADDKPC.S         00004DC4       18       ADDKPC.S         00004DC5       21       MVKH.S1         00004AC0       24       OR.L1X       autoinit.c(15)         00004AC4       11       STW.D2T1         00004AC5       11       NOP         00004AC6       11       NOP         00004AC7       11       NOP         00004AC8       11       NOP         00004AC4       10       NOP         00004AD4       25       STW.D1T1         00004AD4       25       STW.D1T1         00004AD4       25       OR.S1       Pipeline stall         00004AD6       35       OR.S1       Pipeline stall         00004AE0       36       [ A0]       BNOP.S2       OULL         00004AE6       11       OR.S1          00004AE7       11       OR.S1          00004AE0       36       [ A0]       BNOP.S2          00004AE7       11       OR.S1		5			<b>D</b> <sup>2</sup> <b>1</b> <sup>2</sup> <b>1 1</b>	
00004DEC       16       B.S2         00004DC0       17       MVK.S1         00004DC3       21       MVKH.S1         00004DC3       23       MVKH.S1         00004DC3       23       MVKH.S1         00004AC0       24       OR_LIX       autoinit.c(15)         00004AC2       11       NOP         00004AC3       11       NOP         00004AC4       11       NOP         00004AC5       11       NOP         00004AC6       11       NOP         00004AD0       11       NOP         00004AD3       5       STW.DITI         00004AD4       25       STW.DITI         00004AD6       11       OR.L2         00004AD6       11       OR.L2         00004AD6       11       OR.S1         00004AE0       35       OR.S1         00004AE0       36       [ AO] BNOP.S2         00004AEC       11       OR.S1         00004AEC       11       OR.S1         00004AEC       11       OR.S1         00004AE0       11       OR.S1					Pipeline stall	
00004DC0     17     MVK.S1       00004DC4     18     ADDKPC.S       00004DC8     21     MVKH.S1       00004AC0     24     OR.L1X       00004AC4     11     STW.D2T1       00004AC4     11     NOP       00004AC4     11     NOP       00004AC4     11     NOP       00004AC5     11     NOP       00004AC6     11     NOP       00004AC0     11     NOP       00004AC0     11     NOP       00004AD0     11     NOP       00004AD0     11     OR       00004AE0     26     OR       00004AE0     35     OR       00004AE0     11     OP       00004AE0     11     OP <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td></t<>						
00004DC4       18       ADDKPC.S       Pipeline stall         00004DC8       21       MVKH.S1       Pipeline stall         00004AC0       23       MVKH.S1       autoinit.c(15)         00004AC4       11       STW.D2T1       autoinit.c(15)         00004AC3       11       NOP          00004AC4       11       NOP          00004AC4       11       NOP          00004AC5       11       NOP          00004AD4       25       STW.D1T1          00004AD4       25       OR.S1          00004AE0       35       OR.S1          00004AE4       11       CMPEQ.L1          00004AE4       11       OP.S1          00004AF0       11           00004AE4       11       OR.S1						
00004DC8         21         MVKH.S1         Pipeline stall           00004DC3         23         MVKH.S1            00004AC0         24         OR.LIX            00004AC0         24         OR.LIX            00004AC4         11         STW.D2T1            00004AC2         11         NOP            00004ACC         11         NOP            00004AD0         11         OR.L2            00004AD0         26         OR.S1            00004AE0         35         OR.S1            00004AE0         36         [ AO] BNOP.S2            00004AEC         11         OR.S1            00004AEC         11         OR.S1            00004AE0         11         OR.S1            00004AE0         11						
00004DC8         23         MVKH.S1           00004AC0         24         OR.L1X         autoinit.c(15)           00004AC4         II         STW.D2T1           00004AC4         III         NOP           00004AC4         III         NOP           00004AD0         III         NOP           00004AD0         III         NOP           00004AD0         III         OR.L2           00004AD0         III         OR.L2           00004AE0         26         OR.S1           00004AE0         35         OR.S1           00004AE4         III         CMPEQ.L1           00004AE5         36         [ A0]           00004AE0         35         OR.S1           00004AE0         36         [ A0]           00004AE0         10         OR.S1           00004AEC         III         OR.S1           00004AE0         IIII         OR.S1					Disslins stall	
00004AC0       24       OR.L1X       autoinit.c(15)         00004AC4                STW.D2T1         00004AC8                NOP         00004AC0                NOP         00004AC0                NOP         00004AC4       25       STW.D1T1         00004AD4       25       STW.D1T1         00004AD4       25       STW.D2T         00004AD5                OR.L2         00004AE0       35       OR.S1         00004AE0       35       OR.S1         00004AE4                CMPEQ.L1         00004AE5       36       [ A0]         00004AE6                OR.S1         00004AE7                OR.S1         00004AE0       36       [ A0]         00004AE6                OR.S1         00004AE7                OR.S1         00004AE7                OR.S1			WITTER CT		riperine stari	
00004AC4     II     STW.D2T1       00004AC3     II     NOP       00004ACC     II     NOP       00004AD0     II     NOP       00004AD0     II     NOP       00004AD6     II     STW.D1T1       00004AD6     II     OT.L2       00004AE0     26     OR.S1       00004AE0     35     OR.S1       00004AE4     II     CMPEQ.L1       00004AE5     36     IAOI       00004AE6     II     OR.S2       00004AE6     II     OR.S1       00004AE6     II     OR.S1       00004AE6     II     OR.S1				autoinit o(15)		
00004AC8     I     NOP       00004ACC     II     NOP       00004AD0     II     NOP       00004AD4     25     STW.DITI       00004AD4     25     STW.DITI       00004AD5     II     STDW.D2T       00004AD6     II     OR.L2       00004AE0     26     OR.S1       00004AE4     II     CMPEQ.L1       00004AE5     36     [ A0]       00004AE6     II     OR.S2       00004AE7     II     OR.S1       00004AE6     II     OR.S1       00004AE0     II     OR.S1		24		autornit.c(15)		
00004ACC         NOP           00004AD0         I           00004AD4         25           00004AD4         25           00004AD4         25           00004AD4         25           00004AD4         25           00004AD4         25           00004AD5         II           00004AD6         II           00004AE0         26           00004AE0         35           00004AE0         35           00004AE4         II           00004AE5         36           00004AE6         II           00004AE7         II           00004AE8         36           II         OR.S1           00004AE6         II           00004AE7         II           00004AE8         36           II         OR.S1		E F	NOD			
00004AD0         I         NOP           00004AD4         25         STW.DITI           00004AD8         I         STW.DITI           00004AD6         II         OR.L2           00004ADC         II         OR.L2           00004AE0         26         OR.S1           00004AE0         35         OR.S1           00004AE4         II         CMPEQ.L1           00004AE5         36         [ A0]           00004AEC         II         OR.S1           00004AE0         06         07.51		1	NOD			
00004AD4     25     STW.DIT1       00004AD8            STDW.D2T       00004ADC            OR.L2       00004AE0     26     OR.S1       00004AE0     35     OR.S1       00004AE4            CMPEQ.L1       00004AE8     36     [ A0]       00004AE0            OR.L1       00004AE0            OR.L1       00004AE4            OR.L1		(19) (19)	(1997) S.S.S.			
00004AD8            STDW.D2T       00004ADC            OR.12       00004AE0     26     OR.S1       00004AE0     35     OR.S1       00004AE0     35     OR.S1       00004AE4            CMPEQ.L1       00004AE5     36     [ A0]       00004AEC            OR.S1       00004AEC            OR.S1       00004AE0            OR.S1		25				
00004ADC         I         OR.L2         OR.           00004AEO         26         OR.S1         Pipeline stall           00004AEO         35         OR.S1         OR.S1           00004AEO         35         OR.S1         OR.S1           00004AEA         II         CMPEQ.L1         OR.S2           00004AEC         36         [ A0]         BNOP.S2         OR.L1           00004AEC         II         OR.S1         OR.S1         OR.S1		20 11				
00004AE0         26         OR.S1         Pipeline stall           00004AE0         35         OR.S1            00004AE4                    CMPEQ.L1         00004AE8           00004AE6         36         [ A0] BNOP.S2            00004AE7                    OR.L1            00004AE7                    OR.S1			OD TO			
00004AE0         35         OR.S1            00004AE4                    CMPEQ.L1            00004AE8         36         [A0]         ENOP.S2            00004AEC                    OR.L1            00004AF0                    OR.S1		26			Pipeline stall	
00004AE4                    CMPEQ_I1           00004AE8         36         [ A0]         BNOF.S2           00004AEC                    OR.L1            00004AF0                    OR.S1						
00004AE8         36         [ A0 ]         BNOP.S2            00004AEC                    OR.L1            00004AF0                    OR.S1						
00004AEC    OR.L1 00004AF0    OR.S1		36				
00004AF0    OR.S1	00004AEC	11				
	00004AF0	11				
						>
	ady				One traceable CPU connected	

◀►

### Scrolling and Source Code Display Synchronization

The trace tutorial program contains one C file, tutorial.c, which is not located in the tutorial project directory. In the Trace Display you will see that under the Source column, only the source file name and line number, "tutorial.c(36)", appear for the main() function instructions.

😽 Trace Display - [Bla	ickhawk USB560m	n - C6416 DSI	< Emulator/TMS	320C6400_0]				
File View Window I	Help							- 8
Start Resume	Query	💽 🗆 🗖 Qu	ery Filter	Fields Save	Overlays_	Bandwidth Info		
Program Cycles	abs) [cyc]	Disassembly		Source		Т	race Status	
00004E24		[] [!B0]	LDW.D1T1					
00004E28	418		MVK.L1					
00004E2C	419	[!B0]	ADD.L1					
00004E30	420		NOP					
00004E34	421		OR.L2X	args_main.c	(36)			
00004800	422		STW.D2T2	tutorial.c(:	36)			
00004804	423		NOP	2				
00004808	425		MVK.L1	tutorial.C(	37)			
0000480C	426		STW.D2T1					
00004810	427		NOP					
00004814	429		MVK.L1	tutorial.c(3	38)			
00004818	430		STW.D2T1		and the			
0000481C	431		NOP					
00004820	433		MVK.L2	tutorial.c(	(2)			
00004824	434		STW.D2T2	010000.0000000000000000000000000000000				
00004828	435		NOP					
0000482C	437		MVK.S2	tutorial.c(	(2)			
00004830	438		CMPLT.L2					
00004834	439	[!B0]	BNOP.S1					
00004834	444	[   B0 ]	BNOP S1			F	ipeline stall	
00004834	450	[ BO]	BNOP S1				xecution continued	
00004838	451	[]	LDH. D2T2		4)			
0000483C	452		NOD	outorier.et.				
0000483C	454		NOP			F	xecution continued	
00004000	455						mulation halt	
	400						iming disabled during	emulation
						24	Iming disabled during	Cardiotion .
						E	nd of trace	
								>
eceiver stopped due to use	r request , New sample	es detected, Trac	e buffer 0.09% full			Device Clock: 100	0.000Mhz Trace Data: 333.300Mhz	Sample 589 of 6
eady						One traceable	CPU connected	-

To inform the Trace Display where the tutorial.c file is located, you must add the directory to the Trace Display's Source Directory List.

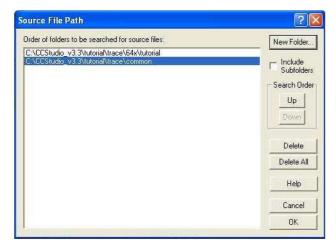
1. Right click in the Trace Window and select Source ® Set Source Directory... to bring up the Source File Path Dialog.

2 🛛
New Folder
Up Down Delete
Delete All
Cancel

2. Click on the New Folder... button and navigate to and select the common directory one level up from the tutorial project directory.

Browse For Folder	2 🛛
Please select folder to be added to source file path	
🕀 🧰 sim27xx	A
🗄 🛅 sim28xx	
🗄 🛅 sim54xx	
🗈 🛅 sim55xx	
🗄 🫅 sim62xx	
🗄 🛅 sim64xx	
🗄 🛅 simarm7	
🖃 🧰 trace	
🖃 🧰 64×	
🗉 🧰 tutorial	
⊞ 🧰 <u>64x+</u>	
🖃 🖾 common	
regdata.lst	
C) tutorial.c	
🕀 🧰 traceold	
💽 DosRun.bat	
options.txt	
sdvr_inst.dll	
osdvr_inst.exe	
🗄 🧫 clearcase	
🗷 🧰 compressed	
🗉 🧰 ctools	S
	OK Cancel

3. Click OK to dismiss the dialog. The Source File Path dialog should now contain both the tutorial project directory and the common directory as shown below.



4. Click OK to dismiss the Source File Path dialog.

The Trace window should now show the source code lines for the main() function in tutorial.c as seen below, rather than just the file name and line numbers.

Trace Display - [Bla File View Window		n - C6416 DSI	K Emulator/TMS	3200.6400	_0]					
Start Resume	Query	▼> F Qu	ery Filter	Fields	Save 0	Iverlays	Bandwidth Info	u.]		
Program Cycles	(abs) [cyc]	Disassembly	/	Source				Trace State	us	
00004E24		[!B0]	LDW.D1T1							
00004E28	418		MVK.L1							
00004E2C	419	[!B0]	ADD.L1							
00004E30	420		NOP							
00004E34	421		OR.L2X	args_ma	in.c(36	)				
00004800	422		STW.D2T2	{						
00004804	423		NOP							
00004808	425		MVK.L1	int a =	10:					
0000480C	426		STW D2T1		82					
00004810	427		NOP							
00004814	429			int cnt	= 0.					
00004818	430		STV.D2T1		1.5					
0000481C	431		NOP							
00004820	433		MVK.L2		for (i	= 0 · i	. < 50;			
00004824	434		STW.D2T2		0.000.000					
00004828	435		NOP							
0000482C	437		MVK.S2		for (i	= 0 · i	< 50:			
00004830	438		CMPLT.L2		101 (1					
00004834	439	[!B0]	BNOP.S1							
00004834	444	[   B0]	BNOP S1					Pipeline	stall	
00004834	450	[ ! B0 ]	BNOP S1						continued	
00004838	451	[:00]	LDH. D2T2			aaa++	93	LACCUCIU	r continued	
0000483C	452		NOR			aaatt	1			
0000483C	454							Property in a	n continued	
00004830	454		NOP					Emulation		
	400								isabled during	
								liming d.	isabied during	emutation
								End of th	race	
										>
ceiver stopped due to use	er request , New sample	es detected, Trac	e buffer 0.09% full				Device Clock: 10	100.000Mhz Tr	ace Data: 333.300Mh	z Sample 589 of 6
							One tracea	ole CPU connec	ted	

As you begin to scroll through the trace window, the Trace Display opens a separate window for the tutorial.c source file. You may need to select Window ® Tile to see both the trace window and the source window in the Trace Display.

The highlighted line in the source code window will change to match the highlighted source line in the trace window as you continue to scroll in either direction. For example:

Initial scrolling results in:

	Kildwk Osbodoolii - Ci	6416 DSK Emulator/TMS32	20C6400_0	
e View Window Help				
C:\CCStudio_v3_3\	tutorial\trace\commo	n\tutorial c		
*/				
*/ oid main()				
ard marn()				
nt a = 10;				
it cnt = 0;				
t ret; t i:				
.с. д.				
for (i =	0; i < 50; i++)			
{	and a second			
	(aa++; ;cc++;			
}	200++;			
while(a)				
ł				
i x	xx++;			
anda Antonio (1996) - 1970 - 1980 - 1980 - 1980	na salah karang karang karang sa	THE PROPERTY AND A		
	na salah karang karang karang sa	ator/TMS320C6400_0		
	na salah karang karang karang sa		Fields Save Overlays Bandwidth Info	
Blackhawk USB560 Start Resume rogram Cycles	om - C6416 DSK Emul Query - (abs) (cyc) Dis	C Query Filter	Source Trace Status	
Blackhawk USB560 Start Resume rogram Cycles 00004814	0m - C6416 DSK Emul Query (abs) [cyc] Dis 429	assembly		
Blackhawk USB560 Start Resume rogram Cycles 00004818	m - C6416 DSK Emut Query → (abs) [cyc] Dis 429 430	assembly MVK L1 STW. D2T1	Source Trace Status	
Blackhawk USB560           Start         Resume           rogram         Cycles           00004814         00004818           00004818         00004816	0m - C6416 DSK Emul Query (abs) [cyc] Dis 429 430 431	Query Filter sassembly MVK.L1 STV.D2T1 NOP	Source Trace Status int cnt = 0;	
Blackhawk USB560 Start Resume rogram Cycles 00004814 00004816 00004820	0m - C6416 DSK Emul Query (abs) [cyc] Dis 429 430 431 433	A Query Filter assembly MVK. L1 STW. D2T1 NOP MVK. L2	Source Trace Status	
Blackhawk USB560 Start Resume rogram Cycles 00004814 00004818 00004820 00004824	m - C6416 DSK Emul Query ▼ [abs] [cyc] Dis 429 430 431 433 434	A Query Filter Sassembly MVK.L1 STW.D2T1 NOP MVK.L2 STW.D2T2 NOP	Source Trace Status int cnt = 0;	
Blackhawk USB560 Start Resume	0m - C6416 DSK Emul Query (abs) [cyc] Dis 429 430 431 433	A Query Filter Sassembly MVK.L1 STW.D2T1 NOP MVK.L2 STW.D2T2 NOP	Source         Trace Status           int cnt = 0;         for (i = 0; i < 50;	
Blackhawk USB560           Start         Resume           rogram         Cycles           00004814         00004818           00004820         00004820           00004824         00004828           00004828         00004828           00004830         00004820	m - C6416 DSK Emul Query (abs) [cyc] Dis 429 430 431 433 434 434 435 437 438	Autor Contraction	Source Trace Status int cnt = 0;	
Blackhawk USB560 Start Resume  rogram Cycles 00004814 00004812 00004820 00004824 00004824 00004828 00004826	m - C6416 DSK Emul Query (1) [abs] [cyc] Dis 429 430 431 433 434 434 435	A Query Filter Sassembly MVK.L1 STW.D2T1 NOP MVK.L2 STW.D2T2 NOP MVK.S2	Source         Trace Status           int cnt = 0;         for (i = 0; i < 50;	
Blackhawk USB560           Start         Resume           rogram         Cycles           00004814         0004814           00004814         0004812           00004820         00004824           00004824         00004828           00004820         00004824           00004820         00004828           00004820         00004828	m - C6416 DSK Emul Query (abs) [cyc] Dis 429 430 431 433 434 434 435 437 438	Autor Contraction	Source         Trace Status           int cnt = 0;         for (i = 0; i < 50;	
Blackhawk USB560 Stat Resume rogram Cycles 00004814 00004816 00004820 00004820 00004824 00004828 00004828 00004830 00004834	0m - C6416 DSK Emul Query (abs) (cyc) Dis 429 430 431 433 434 435 435 437 438 439	Query         Filter           assembly         MVK.L1           STW.D2T1         NOP           MVK.L2         STW.D2T2           STW.D2T2         NOP           MVK.S2         CMPLT.L2           [1B0]         BNOP.S1	Source         Trace Status           int cnt = 0;         for (i = 0; i < 50;	
Blackhawk USB560 Stat Resume rogram Cycles 10004814 10004818 10004820 10004820 10004820 10004824 10004828 10004830 10004834	0m - C6416 DSK Emul Query (abs) (cyc) Dis 429 430 431 433 434 435 435 437 438 439	Query Filter assembly MVK.L1 STW.D2T1 NOP MVK.L2 STW.D2T2 NOP MVK.S2 CMPLT.L2 [1B0] BNOP.S1	Source         Trace Status           int cnt = 0;         for (i = 0; i < 50;	

Second scrolling results in:

	andwik OSDODOUII	- C6416 DSK Emulator/TMS3	20C6400_0		
View Window Help					
C:\CCStudio_v3.3\t	utorial\trace\com	mon\tutorial.c			2
•/					
oid main()					
nt a = 10:					
nt cnt = 0;					
nt ret;					1
nt i;					1
for (i = )	); i < 50; i+4	+)			
{		91 91			
	aa++; cc++;				
}	26++;				
while(a)					
{					
X	xx++;				
and a second	A STATE OF A	mulator/TMS320C6400_0			
and a second	m - C6416 DSK E	mulator/TMS320C6400_0 ▼ >	Fields Save Øverlays Bandwidt	h Info	
Blackhawk USB560	n - C6416 DSK Er		Fields Save Overlays Bandwidt	h Info	
Blackhawk USB560 Start Resume Program Cycles ( 00004828	m - C6416 DSK En Query abs] [cyc] 435	Query Filter  Disassembly  NOP	Source	Trace Status	
Blackhawk USB560 Start Resume rogram Cycles ( 00004828 0000482C	m - C6416 DSK Er Query abs) [cyc] 435 437	Query      Filter Disassembly  NOP MVK.S2	<b>Source</b> for (i = 0; i < 50;	Trace Status	
Blackhawk USB560           Start         Resume           trogram         Cycles (           00004828         00004826           00004820         00004830	m - C6416 DSK En Query abs) [cyc] 435 435 438	C Query      Filter  Disassembly  NOP  MVK.S2  CMPLT.L2	for (i = 0; i < 50;	Trace Status	
Blackhawk USB560           Start         Resume           Program         Cycles           00004828         00004820           00004820         00004830           00004834         00004834	m - C6416 DSK En Query abs] [cyc] 435 437 438 439	C Query      Filter  Disassembly  NOP  MVK.S2  CMPLT.L2  [180] BNOP.S1	<b>Source</b> for (i = 0; i < 50;	Trace Status	
Blackhawk USB560 Start Resume rogram Cycles ( 00004828 0000482C 00004834 00004834	m - C6416 DSK Er Query abs] [cyc] 435 437 438 439 439 444	Query Filter      Disassembly     NOP     MVK.S2     CMPLT.L2.     [180] BNOP.S1     [180] BNOP.S1	<b>Source</b> for (i = 0; i < 50;	Trace Status	
Blackhawk USB560 Start Resume Program Cycles (	m - C6416 DSK En Query abs] [cyc] 435 437 438 439	C Query      Filter  Disassembly  NOP  MVK.S2  CMPLT.L2  [180] BNOP.S1	for (i = 0; i < 50;	Trace Status	
Blackhawk USB560           Start         Resume           Irogram         Cycles (           00004828         00004820           00004820         00004834           00004834         00004834           00004834         00004834           00004838         00004838	m - C6416 DSK Er Query abs) [cyc] 435 437 438 439 439 439 444 450 451	Query Filter      Disassembly     NOP     MVK.S2     CMPLT.L2     [180] BNOP.S1     [180] BNOP.S1     [180] BNOP.S1     LDH.D2T2     NOP	for (i = 0; i < 50; aaa++;	Trace Status Pipeline stall Execution continued	
Blackhawk USB560 Start Resume rogram Cycles ( 00004828 00004820 00004834 00004834 00004834 00004834 00004838 00004836	m - C6416 DSK Er Query abs [cyc] 435 437 438 439 443 439 444 451	▼	<pre>Source for (i = 0; i &lt; 50;</pre>	Trace Status	
Blackhawk USB560           Start         Resume           rogram         Cycles (           00004828         00004830           00004830         00004834           00004834         00004834           00004834         00004834           00004836         00004834           00004836         00004834           00004836         00004836	m - C6416 DSK E Query abs] [cyc] 435 437 438 439 434 439 444 450 451 452 454	Query         Filter           Disassembly         NOP           MVK.S2         CMPIT.L2           [!B0]         BNOP.S1           [!B0]         BNOP.S1           [!B0]         BNOP.S1           [IB0]         DNOP.S1           NOP         NOP           NOP         NOP	<pre>Source for (i = 0; i &lt; 50;</pre>	Fipeline stall Execution continued	
Blackhawk USB560 Start Resume Program Cycles ( 00004828 00004830 00004834 00004834 00004834 00004834 00004838 00004836 00004836	m - C6416 DSK E Query abs] [cyc] 435 437 438 439 434 439 444 450 451 452 454	Query Filter      Disassembly          NOP         MVK.S2         CMPLT.L2      [180] BNOP.S1     [180] BN	<pre>Source for (i = 0; i &lt; 50;</pre>	Trace Status Pipeline stall Execution continued	

Note: As you scroll through the Trace data, the corresponding source file opens in a separate window in the Trace Display. The highlighted line in the source code window changes to match the highlighted source line in the Trace Display must know the location of the source file. Source files located in the Code Composer Studio project directory are found automatically by the Trace Display. If the Trace Display does not know the location of the source file, the Trace Display will not open a source window, and instead will show only the source file name and line number in the Source column. As an example, this behavior is not true for the code at the beginning of the Trace Tutorial program, which is the C Run-Time Library. The source files for the C Run-Time Library are distributed as an archive file in the CCStudioInstall/Device/cgtools/lib/ directory titled rts.src. You can use the Code Generation archiver tool to extract the source files for the C Run-Time Library.

Library.



### Searching Trace Data

You can also search through the data in the Trace Display by clicking on the Query... button to bring up the Trace Search Query dialog.

- 1. Maximize the Trace Window by clicking on the Maximize icon in the upper right corner. Then click on the Home button on your keyboard to scroll to the beginning of the trace data.
- 2. Select the Trace Status column in the left hand pane and click the Field contains string A radio button.
- 3. Enter the text Pipeline stall into the A = text edit box, then click OK.

Program Address Load Address Program Data Read Address Read Data Read Size (In Bits) Write Address Write Data Write Size (In Bits) Cycles	: (any) : (any) : (any)	Field condition C Don't care Field == A (Field & Mask) == A (Field & Mask) >= A & (Field & Mask) <= 8 (Field contains string A
Trace Status TI 560T Status Source Disassembly	: (any) : contains "Pipeline stall" : (any) : (any) : (any)	A = Pipeline stall

The Trace Display now highlights in light gray the rows that match the specified Query text.

C:\CCStudio_v3.3\tu	torial\trace\cor	nmon\tutorial.c			
*/					
oid main()					
nt = 10;					
nt cnt = 0;					
nt ret;					
nt i;					
for $(i = 0)$	; i < 50; i+	+)			
{					
	α++; ≥++;				
}	377,				
while(a)					
{					
XXX	7 L L ·				
	ai (,				
and a second		Emulator/TNS220C6400_0			
Blackhawk USB560m	- C6416 DSK I	Emulator/TMS320C6400_0	and the second second		
		imulator/TMS320C6400_0 ▼ > ▼ Query □ Filter	Fields Save Overlays Bandwidth Info		
Blackhawk USB560m	- C6416 DSK I		Fields Save Overlays Bandwidth Info	Trace Status	
Blackhawk USB560m Start Resume Program Cycles (a 00004828	- C6416 DSK I Query bs) [cyc] 435		Source		
Blackhawk USB560m           Start         Resume           Program         Cycles (a           00004828         0000482C	- C6416 DSK I Query bs] [cyc] 435 437	✓ V Query Filter Disassembly NOP MVK.S2	Source		
Blackhawk USB560m           Start         Resume           Program         Cycles (a           00004828         00004820           00004830         00004830	- C6416 DSK I Query bs) [cyc] 435 437 438	▼ > ▼ Query  Filter Disassembly NOP MVK.S2 CMPLT 152	<b>Source</b> for (i = 0; i < 50;		
Blackhawk USB560m           Start         Resume           Program         Cycles (a)           00004828         0000482C           00004830         00004834	- C6416 DSK I Query bs] [cyc] 435 437 438 439	▼ > ▼ Query	<b>Source</b> for (i = 0; i < 50;	Trace Status	
Blackhawk USB560m           Start         Resume           Program         Cycles (a           00004828         00004820           00004830         0004834	- C6416 DSK I Query bs] [cyc] 435 437 438 439 444	Ouery Filter      Disassembly      NOP      NVK.S3      CMPIT M22      [180] BNOP S1      [180] BNOP S1      [180] BNOP S1	Source for (i = 0; i < 50;	Trace Status Pipeline stall	
Blackhawk USB560m           Start         Resume           Program         Cycles (a)           00004828         0000482C           00004830         00004834	- C6416 DSK I Query bs] [cyc] 435 437 438 439		Source	Trace Status	
Blackhawk USB560m Stat Resume Program Cycles [a 00004828 00004820 00004834 00004834 00004834 00004834 00004834 00004834 00004832	- C6416 DSK I Query bs] [cyc] 435 437 438 439 444 450 451	▼ > ▼ Query	Source for (i = 0; i < 50; aaa++;	Trace Status Pipeline stall	
Blackhawk USB560m           Statt         Resume           Program         Cycles [a           00004828         00004820           00004820         00004830           00004834         00004834           00004834         00004838	- C6416 DSK I Query bsi [cyc] 435 437 438 439 449 445 451	▼ > ▼ Query	Source for (i = 0; i < 50; aaa++;	Trace Status Pipeline stall	
Blackhawk USB560m Start Resume Program Cycles [a 00004828 00004828 00004830 00004834 00004834 00004834 00004834 00004834 00004834	- C6416 DSK I Query bs] [cyc] 435 437 438 439 444 450 451	V Query Filter      Disassembly      NOP     MVK.S2     CMPIT VS2     (HB0] BNOP S1     (HB0] BNOP S1     (HB0] BNOP S1     (HB0] BNOP S1     (HB0] DNOP S1     (HB0) DNOP S1     (HB	Source for (i = 0; i < 50; aaa++;	Trace Status Pipeline stall Execution continued	
Blackhawk USB560m           Start         Resume           Program         Cycles [a           00004828         00004820           00004820         00004830           00004834         00004834           00004834         00004834           00004834         00004834           00004836         00004836           00004837         00004836	- C6416 DSK I Query bs) [cyc] 435 437 438 439 444 450 451 450 451 452 454	▼ > ▼ Query Filter Disassembly NOP MVK.S3 CMPLT 322. [180] BNOP.S1 [180] BNOP.S1 [180] BNOP.S1 [180] BNOP.S1 LDH.D2T2 NOP NOP	Source for (i = 0; i < 50; aaa++;	Trace Status Pipeline stall Execution continued Execution continued	
Blackhawk USB560m           Start         Resume           Program         Cycles [a           00004828         00004820           00004820         00004830           00004834         00004834           00004834         00004834           00004834         00004834           00004836         00004836           00004837         00004836	- C6416 DSK I Query bs) [cyc] 435 437 438 439 444 450 451 450 451 452 454	V Query Filter      Disassembly      NOP     MVK.S2     CMPIT VS2     (HB0] BNOP S1     (HB0] BNOP S1     (HB0] BNOP S1     (HB0] BNOP S1     (HB0] DNOP S1     (HB0) DNOP S1     (HB	Source for (i = 0; i < 50; aaa++;	Trace Status Pipeline stall Execution continued	

4. You may need to click on the Home key and scroll down to see more than one Pipeline stall in the Trace Display window.

	khawk USB560m - C641	6 DSK Emulator/TMS32	2006400_0		
ile View Window Help					
C:\CCStudio_v3.3\t	utorial\trace\common\tr	utorial.c			
{ int a = 10; int cnt = 0; int ret; int i;					^
{ c } while(a) {	0; i < 50; i++) aa++; cc++; xx++;				
у у	yy++;				
cnt = 1;	yy++;				~
cnt = 1;	yy++: m   - C6416 DSK Emulato	r/TMS320C6400_0			
cnt = 1;		r/TMS320C6400_0 ▼ Query	Fields Save Overlays	Bandwidth Info	
cnt = 1; Blackhawk USB560 Start Resume	m - C6416 DSK Emulato		Fields Save Overlays Source	Bandwidth Info	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles 00004DB0	m - C6416 DSK Emulato Query > [abs] [cyc] Disas 4	Query     Filter sembly MVKH.S2			
cnt = 1; Blackhawk USB560 Start Resume Program Cycles   00004DB0 00004DB4	m - C6416 DSK Emulato <u>Query</u> [abs] [cyc] Disas 4 5	✓ Query         ✓ Filter           sembly         MVKH.S2           MVK.S2		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles   00004DB0 00004DB4 00004DB8	m - C6416 DSK Emulato <ul> <li>Query</li> <li>Query</li> <li>Disas</li> <li>4 5 6</li> </ul>	✔         Query         Filter           sembly         MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2         MVKH.S2			
cnt = 1; Blackhawk USB560 Start Resume Program Cycles   00004DB0 00004DB4 00004DB8 00004DB8	m - C6416 DSK Emulato < Query > > [abs] [cyc] Disas 4 5 6 15	♥ Query Filter sembly MVKH.S2 MVK.S2 MVKH.S2 MVKH.S2		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles   00004DB0 00004DB4 00004DB8 00004DB8 00004DBC	m - C6416 DSK Emulato < Query > [abs] [cyc] Disas 4 5 6 15 16	© Query Г Filter sembly MVKH.S2 MVKH.S2 MVKH.S2 MVKH.S2 B.S2		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles 00004DB0 00004DB4 00004DB8 00004DB8 00004DBC 00004DC0	m - C6416 DSK Emulato Query > [abs] [cyc] Disas 4 5 6 15 16 17	☑ Query         Filter           sembly         MVKH.S2           MVK.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles   00004DB0 00004DB8 00004DB8 00004DB8 00004DC4	m - C6416 DSK Emulato Query >> [abs] [cyc] Disas 4 5 6 15 16 17 18	☑ @uery         ☐ Filter           sembly         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         B.S2           MVK.S1         ADDKPC.S.		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles 00004DB0 00004DB4 00004DB8 00004DB8 00004DB8 00004DBC 00004DC0	m - C6416 DSK Emulato <ul> <li>Query</li> <li>Query</li> <li>Disas</li> <li>4</li> <li>5</li> <li>6</li> <li>15</li> <li>16</li> <li>17</li> <li>18</li> <li>21</li> </ul>	☑ Query         ☐ Filter           sembly         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         B.S2           MVK.S1         ADDKPC.S.		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles   00004DB0 00004DB8 00004DB8 00004DB8 00004DC0 00004DC4	m - C6416 DSK Emulato Query >> [abs] [cyc] Disas 4 5 6 15 16 17 18	☑ @uery         ☐ Filter           sembly         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         B.S2           MVK.S1         ADDKPC.S.		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles   00004DB4 00004DB4 00004DB8 00004DB8 00004DBC 00004DC0 00004DC4 00004DC8	m - C6416 DSK Emulato <ul> <li>Query</li> <li>Query</li> <li>Disas</li> <li>4</li> <li>5</li> <li>6</li> <li>15</li> <li>16</li> <li>17</li> <li>18</li> <li>21</li> </ul>	☑ Query         ☐ Filter           sembly         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         B.S2           MVK.S1         ADDKPC.S.		Trace Status	
cnt = 1; Blackhawk USB560 Start Resume Program Cycles 00004DB0 00004DB4 00004DB4 00004DB8 00004DB8 00004DC0 00004DC0 00004DC4 00004DC8 00004DC8	m - C6416 DSK Emulato <ul> <li>Query</li> <li>Query</li> <li>Disas</li> <li>4</li> <li>5</li> <li>6</li> <li>15</li> <li>16</li> <li>17</li> <li>18</li> <li>21</li> </ul>	☑ Quety         Filter           sembly         MVKH S2           MVKH S2         MVKH S2           MVKH S2         MVKH S2           MVKH S2         MVKH S1           ADDKPC S         MVKH S1           MVKH S1         MVKH S1	Source	Trace Status	

 In the Trace Display window, click on the > button next to the Query drop down box. The first matching line is highlighted in yellow. Clicking the > button highlights the next match, and clicking the < button highlights the previous match.</li>

and the second second second second		6 DSK Emulator/TMS320C	8100_0	
File View Window Help				
C:\CCStudio_v3.3\tuto	rial\trace\common\tr	utorial.c		
{				^
int $a = 10;$ int cnt = 0;				
int ret;				
int i:				
	i < 50; i++)			
{ aaa+	++;			
ccc+	H+:			
} while(a)				
{	LE 7			
yyy+				
cnt = 1;				×
🔲 Blackhawk USB560m -	- C6416 DSK Emulato	r/TMS320C6400_0		
	< Query ->	Query Filter Fie	elds Save Overlays Bandwidth Info	1
Start Resume		It goory I Ther Th	eius Save Overlays Bariuwiu(mini)	
		sembly Sc		race Status
Program Cycles (abs 00004DB0	s) [cyc] Disas 4	sembly Sc MVKH.S2		
Program Cycles (abs 00004DB0 00004DB4		sembly Sc MVKH.S2 MVK.S2	purce 1	race Status
Program Cycles (abs 00004DB0	s) [cyc] Disas 4 5 6 15	sembly Sc MVKH.S2	purce 1	
Program         Cycles [abs           00004DB0         00004DB4           00004DB4         00004DB8           00004DB8         00004DB8           00004DB8         00004DBC	s) [cyc] Disas 4 5 6 15 16	sembly Sc MVKH.S2 MVK.S2 MVKH.S2 MVKH.S2 B.S2	purce 1	race Status
Program         Cycles [abs           00004DB0         00004DB4           00004DB8         00004DB8           00004DB8         00004DB8           00004DBC         00004DC0	s) [cyc] Disas 4 5 6 15 16 17	sembly Sc MVKH.S2 MVK.S2 MVKH.S2 MVKH.S2 B.S2 MVK.S1	purce 1	race Status
Program         Cycles [abs           00004DB0         00004DB4           00004DB8         00004DB8           00004DB8         00004DBC           00004DBC         00004DBC           00004DBC         00004DC0           00004DC4         00004DC4	s) [cyc] Disas 4 5 6 15 16 17 18	sembly         Sc           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2	purce 1	race Status
Program         Cycles [abs           00004DB0         00004DB4           00004DB4         00004DB8           00004DB8         00004DBC           00004DBC         00004DC0           00004DC4         00004DC8           00004DC8         00004DC8	s) [cyc] Disas 4 5 6 15 16 17 18 21	sembly         Sc           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S1           MVK.S1         ADDKPC.S.           MVKH.S1         MVKH.S1	purce 1	race Status
Program         Cycles [abs           00004DB0         00004DB4           00004DB8         00004DB8           00004DB8         00004DBC           00004DBC         00004DBC           00004DBC         00004DC4	s) [cyc] Disas 4 5 6 15 16 17 18	sembly         Sc           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2           MVKH.S2         MVKH.S2	purce 1	race Status ♪ Pipeline stall
Program         Cycles [abs           00004DB0         00004DB4           00004DB4         00004DB8           00004DB8         00004DBC           00004DBC         00004DC0           00004DC4         00004DC8           00004DC8         00004DC8	s) [cyc] Disas 4 5 6 15 16 17 18 21 23	sembly         Sc           MVKH.S2         MVK.S2           MVKH.S2         MVKH.S2           B.S2         MVK.S1           MVK.S1         MVKH.S1           MVKH.S1         MVKH.S1	purce 1	race Status

# ◀►

### Filtering Trace Data

While it is often useful to be able to search for data in this find-next/previous manner, it is also useful to find all the places that match a particular set of search criteria and display only those lines. This batch searching capability is called Filtering.

- 1. Click on the checkbox next to the Query... to turn off the Query highlighting.
- 2. Next, click on the Filter... button and set up the same search criteria that we did in the Trace Search Query dialog in the Trace Filter Query dialog:
- a. Select the Trace Status column in the left hand pane and click the Field contains string A radio button.
- b. Enter the text Pipeline stall into the A = text edit box.

Program Address Load Address Program Data Read Address Read Data Read Size (In Bits) Write Address Write Address Write Data Write Size (In Bits) Cycles Trace Status	: (any) : (any)	Field condition         C       Don't care         C       Field == A         C       (Field & Mask) == A         C       (Field & Mask) >= A & & (Field & Mask) <= B         Image: Field contains string A
Ti 560T Status Source Disassembly	: contents in pointe stair : (any) : (any) : (any)	A = Pipeline stall

3. Click OK.

The Trace Display now shows only those lines that match the Filter search criteria. The currently selected row is still highlighted.

		n - C6416 DSK Emulator/TMS3	20C6400_0		
ile View Window H	Help				
C:\CCStudio_v3	3.3\tutorial\trace\c	ommon\tutorial.c			<b>_ </b>
{ int a = 10; int cnt = 0; int ret; int i;					^
for (i {	= 0; i < 50; i aaa++;	++)			
} while(- {	ccc++;				
<b>N</b> .0	<b>XXX++</b> ;				
cnt =	yyy++; 1;				~
Blackhawk USB	1; 560m - C6416 DSK	Emulator/TMS320C6400_0	- Par Alexander		
	1; 560m - C6416 DSK	Emulator/TMS320C6400_0	Fields Save Overlays	Bandwidth Info	
Blackhawk USB	1; 560m - C6416 DSK	▼	Fields Save Overlays	Bandwidth Info	
Blackhawk USB Start Resum Program Cycl 00004DB8	1 ; 560m - C6416 DSK e Query les [abs] [cyc]	Query      Filter     Disassembly     MVKH.S2		Trace Status Pipeline stall	
Blackhawk USB Start Resum Program Cycl 00004DB8 00004DC8	1 ; 560m - C6416 DSK e Query les (abs) [cyc]	Cuery      Filter     Disassembly     MVKH. S2     MVKH. S1		Trace Status Pipeline stall Pipeline stall	
Blackhawk USB Start Resum Program Cycl 00004DB8 00004DC8 00004DC8	1 ; 560m - C6416 DSK ie Query les (abs) [cyc] 2 2 2 2 2	✓      ✓		Trace Status Pipeline stall Pipeline stall Pipeline stall Pipeline stall	
Blackhawk USB           Start         Resum           Program         Cycl           00004DB8         00004DC8           00004DC8         00004DC8           00004DC8         00004DC8           00004DC8         00004DC8	1 ; 560m - C6416 DSK ie Query les (abs) [cyc] 2 2 2	Ouery         Filter           Disassembly         6           MVKH.S2         1           MVKH.S1         6           MVKH.S1         6		Trace Status Pipeline stall Pipeline stall Pipeline stall Pipeline stall	
Blackhawk USB           Start         Resum           Program         Cycl           00004D28         00004D28           00004D28         00004D28           00004D28         00004D28           00004D28         00004D28           00004D28         00004D28	1 ; 560m - C6416 DSK ie Query les [abs] [cyc] 2 2 4 4	Query         Filter           Disassembly         MVKH.S2           MVKH.S1         MVKH.S1           MVKH.S1         MVKH.S1           MVKH.S1         MVKH.S1	Source	Trace Status Pipeline stall	
Blackhawk USB           Start         Resum           Program         Cycl           00004D88         00004D28           00004D28         00004A20           00004A20         00004A20           00004B0C         00004B0C	1 ; 560m - C6416 DSK e Query les (abs) [cyc] 2 2 2 4 6	▼         Query         ▼         Filter           Disassembly         MVKH.S2         1           1         MVKH.S1         1           1         MVKH.S1         6           0         NOP         0	Source	Trace Status           Pipeline stall	
Blackhawk USB           Start         Resum           Program         Cycl           00004D28         00004D28           00004D28         00004ADC8           00004ADC8         00004ADC           00004ADC8         00004ADC           00004ADC8         00004ADC           00004ADC8         00004ADC           00004ADC8         00004ADC           00004ADC2         00004ADC	1 ; 560m - C6416 DSK le Query les (abs) [cyc] 2 2 4 4 6 6 6	Query         Filter           Disassembly         6           MYKH.S2         1           MYKH.S1         1           MYKH.S1         6           OR         0           NOP         0           2         OR.L1	autoinit.c(20) autoinit.c(21)	Trace Status Pipeline stall Pipeline	
Blackhawk USB           Start         Resum           Program         Cycl           00004D58         00004D58           00004D58         00004D58           00004D58         00004D58           00004D58         00004D58           00004D58         00004D58           00004450         00004450           00004815         00004820           00004252         00004520	1 ; 560m - C6416 DSK ie Query les [abs] [cyc] 2 2 2 4 6 6 6 6 6 6	Query         ✔         Filter           Disassembly         6         MVKH.S2           1         MVKH.S1         1           1         MVKH.S1         6           6         OR S1         6           6         NOP         0           0         NOP         2           2         OR L1         5           5         [ A0] LDB D2T1         5	autoinit.c(20) autoinit.c(21) memcpy64.asm(105)	Trace Status Pipeline stall Pipeline	
Blackhawk USB           Start         Resum           Program         Cycl           00004D28         00004D28           00004D28         00004AD28           00004AD28         00004AD2           00004AD28         00004AD2           00004AD28         00004AD2           00004AD28         00004AD2           00004AD28         00004AD2           00004AD20         00004AD2	1 ; 560m - C6416 DSK le Query les (abs) [cyc] 2 2 4 4 6 6 6	Query         ✔         Filter           Disassembly         6         MVKH.S2           1         MVKH.S1         1           1         MVKH.S1         6           6         OR S1         6           6         NOP         0           0         NOP         2           2         OR L1         5           5         [ A0] LDB D2T1         5	autoinit.c(20) autoinit.c(21)	Trace Status Pipeline stall Pipeline	
Blackhawk USB           Start         Resum           Program         Cycl           00004D58         00004D58           00004D58         00004D58           00004D58         00004D58           00004D58         00004D58           00004D58         00004D58           00004450         00004450           0000481C         000048120           00004252         00004252	1 ; 560m - C6416 DSK ie Query les [abs] [cyc] 2 2 2 4 6 6 6 6 6 6	Query         ✔         Filter           Disassembly         6         MVKH.S2           1         MVKH.S1         1           1         MVKH.S1         6           6         OR S1         6           6         NOP         0           0         NOP         2           2         OR L1         5           5         [ A0] LDB D2T1         5	autoinit.c(20) autoinit.c(21) memcpy64.asm(105)	Trace Status Pipeline stall Pipeline	
Blackhowk USB           Start         Resum           Program         Cycl           00004D88         00004D88           00004D28         00004AE0           00004AE0         00004AE0           00004AE0         00004AE0           00004AE1C         00004AE1C           00004AE20         00004AE1C           00004AE1C         00004AE1C           00004AE20         00004AE1C           00004AE20         00004AE1C	1 ; 560m - C6416 DSK ie Query les (abs) [cyc] 2 2 2 2 4 6 6 6 9	Query         ✔         Filter           Disassembly         6         MVKH.S2           1         MVKH.S1         1           1         MVKH.S1         6           6         OR S1         6           6         NOP         0           0         NOP         2           2         OR L1         5           5         [ A0] LDB D2T1         5	autoinit.c(20) autoinit.c(21) memcpy64.asm(105) autoinit.c(29)	Trace Status Pipeline stall Pipeline	
Blackhawk USB           Start         Resum           Program         Cycl           00004DB8         00004DC8           00004DC8         00004AE0           00004AE0         00004AE0           00004AE1         00004BIC           00004AE20         00004AE1C           00004AE20         00004AE1C           00004AE20         00004AE1C           00004AE34         00004AE34	1 ; 560m - C6416 DSK ie Query les (abs) [cyc] 2 2 2 2 4 6 6 6 9	Query         Filter           Disassembly         MVKH.S2           6         MVKH.S1           1         MVKH.S1           6         OR.S1           6         NOP           0         NOP           2         OR.L1           5         [.A0]           6         NOP	autoinit.c(20) autoinit.c(21) memcpy64.asm(105) autoinit.c(29)	Trace Status         Pipeline stall	

4. You can turn off filtering by clicking the checkbox to the left of the Filter... icon. Note that the currently selected sample in the filtered view remains selected in the unfiltered view. This is a useful way to find a particular occurrence of a filtered sample (such as the third Pipeline stall), and then see the surrounding code once filtering is turned off.

💀 Trace Display - Blackhawk USB560m	n - C6416 DSK Emulator/TMS32	20C6400_0		
File View Window Help				
C:\CCStudio_v3.3\tutorial\trace\co	mmon\tutorial.c			
{ int a = 10; int cnt = 0; int ret; int ret;				
<pre>for (i = 0; i &lt; 50; i {</pre>	++)			
cnt = 1;				~
Blackhawk USB560m - C6416 DSK	Emulator/TMS320C6400_0			
Start Resume Query	▼ / □ Query □ Filter	Fields Save Overlays	Bandwidth Info	
Program Cycles (abs) [cyc]	Disassembly	Source	Trace Status	<u>^</u>
00004DB0 5				( <u>1311</u> )
00004DB4 6 00004DB8 7	5 MVK.S2 7 MVKH.S2		Pipeline stall	
00004DB8 16			riperne star	
00004DBC 17				
00004DC0 18	MVK.S1			
00004DC4 19				
00004DC8 22			Pipeline stall	(2008)
00004DC8 24	1 MVKH.S1			M
<				>
Receiver stopped due to user request , New sam	ples detected, Trace buffer 0.09% full	Devi	ce Clock: 1000.000Mhz Trace Data: 333.300Mhz	Sample 10 of 618
Ready			One traceable CPU connected	1



### More Complex Filtering

Filtering is useful to determine if a particular Trace Status value, say Pipeline stalls (see Filtering Trace Data), has occurred at all in the collected trace data. By setting more than one field's search criteria, you can narrow down your search. For example, you could search for all Pipelines stalls while the Program Address value is in the range of the start and end addresses of a particular function.

To demonstrate more complex Filtering criteria, let us assume we are only interested in Pipeline stalls that occur in the Program Address range 0x4B50 - 0x4B70.

- 1. Click on the Filter button to bring up the Trace Filter Query dialog again.
- 2. Select the Program Address column in the left hand pane.
- 3. Click on (Field & Mask) >= A && (Field & Mask) <= B, and enter the value 0x4B50 for A and 0x4B70 for B, then click OK.

Program Address     (0x4b50) <= Value	<ul> <li>C Don't care</li> <li>C Field == A</li> <li>C (Field &amp; Mask) == A</li> <li>(Field &amp; Mask) &gt;= A &amp; (Field &amp; Mask) &lt;= B</li> <li>C Field contains string A</li> </ul>
---------------------------------------	---

The Trace Display should now appear as below.

Start	Resume Query	▼ > □ Query IV Filter.	Fields Save Overlays	Bandwidth Info	
Program	Cycles (abs) [cyc]	Disassembly	Source	Trace Status	1
00004B54	69	NOP .	autoinit c(29)	Pipeline stall	
00004B54	96	NOP .	autoinit.c(29)	Pipeline stall	and a second second second
00004B54	123	NOP .	autoinit.c(29)	Pipeline stall	
00004B54	150	NOP .	autoinit.c(29)	Pipeline stall	
00004B54	177	NOP .	autoinit.c(29)	Pipeline stall	
00004B54	204	NOP .	autoinit.c(29)	Pipeline stall	en en en en en en el
00004B54	231	NOP .	autoinit.c(29)	Pipeline stall	
00004B54	258	NOP .	autoinit.c(29)	Pipeline stall	
00004B54	285	NOP .	autoinit.c(29)	Pipeline stall	
00004B64	316	[ A0] BNOP.S1 .		Pipeline stall	
					2

Tip: Another useful search strategy is to filter for the Trace Status rows that contain "Interrupt", then turn off filtering by un-checking the Filter checkbox to see where each interrupt occurred. This is helpful to verify that interrupts are occurring at the correct priority level. This is also a good way to detect if your Interrupt Service Table Pointer Register contains the correct address by noting where the interrupt branches to when it does occur.

4. Un-check the checkbox next to the Filter... button to turn off all filtering now.

The Trace Display format should now appear as below. (Your data may differ.)

File View Window Help						
Start Resume < Qu	iery 💌 🗾	C Query Filter	Fields Save Overlays	Bandwidth Info		
Program Cycles (abs) (cyc	c] Disass	sembly	Source	Tr	ace Status	
00004B54	69	NOP	autoinit c(29)		peline stall	
00004B54	70	NOP		Ex	ecution continued	
00004B20	71	OR.L1	autoinit.c(21)			
00004B24	[.[.	OR.S1				
00004B28	11	OR.L2X				6
,00004B2C	EL.	ADDKPC.S				
00004CC0	72		memcpy64.asm(101)			
00004CC4		AND.S1		N		
00004CC8	ff	MVK.L1		3		
00004CCC			memcpy64.asm(105)	18.0		
00004CD0	] ] ]	A0] MVK.S1				
00004CD4	a second a literative	OR.L1X				
00004CD8	74 [		memcpy64.asm(109)			
00004CDC	] ]]	A1] LDB.D1T1				
00004CE0	e les	SHRU.S2X				
00004CE4	AND THE ADDRESS OF TH	AND.S1				
00004CE8	75 [		memcpy64.asm(114)			
00004CEC	E E	OR.D1				
00004CF0	1.100	OR.S1X				
00004CF4	anne L'El Area	MVK.L2				
00004CF8	76 [		memcpy64.asm(119)			
00004CFC	1 II	B0] SUB.S2				
00004D00	S12555 0101 9035	CMPGT.L2				
00004D04	77 [		memcpy64.asm(123)			
00004D08	[	B0] LDNDW.D2				
00004D0C	] ]]	B0] SUB.L2				
00004D10	78 [		memcpy64.asm(127)			
00004D14	1 ]]	A0] MVK.S2				
00004D18		OR.L2X				
00004D1C	79 [	A1] STB.D1T2	memcpy64.asm(131)			
						>
eiver stopped due to user request , f	New samples detection	ed, Trace buffer 0.09% full		Device Clock:	1000.000Mhz Trace Data: 333.300Mhz	Sample 96 of 60
ady				One trace	able CPU connected	and the second s

#### Saving and Exporting Trace Data

You can save the collected trace data in different formats for subsequent offline analysis or re-opening in the Trace Display.

1. Click on the Save button in the Trace Display.

Save Trace Samples to a File	? 🛛
Format Fo	with the Trace Display:
Help	Save Cancel

The trace data can be saved in three different file formats:

- Comma Separated Value (.csv) This format is used for importing into spreadsheets and with scripting languages such as Perl or Python. This format cannot be reopened by the Trace Display.
- Text (.ttd) This format is useful for parsing with scripting languages such as Perl or Python. This format can be reopened by the Trace Display.
- Binary (.tdf) This format contains the encoded trace data as it comes out of the receiver and all of the information necessary to reopen the file with the Trace Display.

There are many options in the Save File dialog that allow you to save only the trace data that is important to you rather than saving all of the trace data, which can become quite sizeable.

When saving in binary format, any Trace Display options you have set at the time you save your file are preserved in the file and restored when you re-open the file. For instance, if you have your data filtered, the filter search criteria are restored when you re-open the file.

2. Save the file in binary format as tutorial.tdf.

You can re-open this file using the Trace Display after finishing the Trace Tutorial to explore more of the Trace Display's capabilities.

# ◀►

#### Using Trace System Options

Now that we have seen some of the features of the Trace Display, let us examine a few of the more important Trace System options.

- Using the Synchronize Trace With Target Execution Option
- Restarting and Resuming Trace Collection
- Choosing the Trace Buffer Type
- Changing the Trace System Receiver

### ∢►

### Using the Synchronize Trace With Target Execution Option

By default, when the target hits a breakpoint, the Trace System stops recording and the Trace Display appears automatically and displays data. This behavior is due to the default Synchronize Trace with Target Execution option. The exact columns that appear in the display depend upon the target device.

To see this option in action, select  $\mathsf{Debug} \to \mathsf{Run}$  to run to the breakpoint again

When the breakpoint is hit again, the Trace Display updates automatically, and contains all and only the code executed in the second iteration of the loop between "ccc++;" statements. This is because we have asked the Receiver to synchronize with target execution, and this option forces the Trace Receiver to automatically clear its buffer and restart collection when the target is run. As long as you keep hitting the Run button in Code Composer Studio, the Trace System will clear the Receiver's buffer and record only what is between the loop iterations.

For example:

le View Window Help	JOIN - CUATU	DSK Emulator/TMS32	20C6400_0				
e view window Heip							
C:\CCStudio_v3.3\tutorial\trace	\common\tute	orial.c					
and the second							
nt a = 10; nt cnt = 0;							
it ret;							
ti;							
for $(i = 0; i < 50;$	544X						
{	144)						
aaa++;							
ccc++;							
while(a)							N
{							R
xxx++; yyy++;							
239112							
cnt = 1;							
	SK Forulator (	TUE22004400 0					
Blackhawk USB560m - C6416 E							
	<u>•</u> <u>&gt;</u> 1	Query Filter	Fields Save	Overlays	Bandwidth Info.		
Blackhawk USB560m - C6416 D Start Resume Query		Query Filter	Fields Save	Overlays		_ Trace Status	
Blackhawk USB560m - C6416 D Start Resume Query	<u>•</u> <u>&gt;</u> 1	Query Filter		Overlays		<u></u>	
Blackhawk USB560m - C6416 D Start Resume Query rogram Cycles (abs) (cyc)	<u>•</u> <u>&gt;</u> 1	Query Filter		<u>Cverlays</u>	· ·	<b>Trace Status</b> Start of trace	
Blackhawk USB560m - C6416 D Start Resume Query rogram Cycles (abs) [cyc] 00004840 00004844	<mark>. →</mark>	Query Filter mbly ADD. L2 STH. D2T2		Overlays	· ·	<b>Trace Status</b> Start of trace	
Blackhawk USB560m         - C6416 D           Start         Resume         Query           rogram         Cycles [abs] [cyc]           00004840         00004844           00004848         00004848	<mark>. →</mark>	ADD. L2 STH. D2T2NOF				<b>Trace Status</b> Start of trace	m, Timing collect
Blackhawk USB560m - C6416 D           Start         Resume         Query           rogram         Cycles [abs] [cyc]           00004840         00004844           00004844         00004844           0000484C         0000484C	Disasse	ADD 12 STH D2T2 NOP LDB D2T2				<b>Trace Status</b> Start of trace	
Blackhawk USB560m         - C6416 D           Start         Resume         Query           Program         Cycles [abs] [cyc]           00004840         00004844           00004848         00004848           00004850         0	Disasse	ADD. L2 STH. D2T2 NOP LDB. D2T2 NOP				<b>Trace Status</b> Start of trace	
Blackhawk USB560m         - C6416 D           Start         Resume         Query           rogram         Cycles [abs] [cyc]           00004840         00004844           00004844         00004844           00004845         00004845	Disasse	ADD. L2 STH. D2T2 NOP LDB. D2T2 NOP NOP ADD. L2				<b>Trace Status</b> Start of trace	
Blackhawk USB560m         - C6416 D           Start         Resume         Query           rogram         Cycles [abs] [cyc]           00004840         00004844           00004844         00004844           00004845         00004845           00004854         00004854	Disasse	ADD. L2 STH. D2T2 NOP LDB. D2T2 NOP				<b>Trace Status</b> Start of trace	m, Timing collect
Blackhawk USB560m - C6416 D           Start         Resume         Query           rogram         Cycles [abs] [cyc]           00004840         00004840           00004844         00004844           00004845         00004850           00004850         00004858	Disasse     D     D     1     2     4     5     9     10	ADD. 1.2 STH. D2T2 NOP LDB. D2T2 NOP ADD. 1.2 STB. D2T2 NOP			3	Trace Status Start of trace PC collection c	
Blackhawk USB560m         - C6416 D           Start         Resume         Query           rogram         Cycles [abs] [cyc]           00004840         00004844           00004844         00004844           00004845         00004845	Disasse     D     D     1     2     4     5     9     10	ADD. 1.2 STH. D2T2 NOP LDB. D2T2 NOP ADD. 1.2 STB. D2T2 NOP			; vice Clock: 1000.	Trace Status Start of trace PC collection c	m, Timing collect

# ◀►

### **Restarting and Resuming Trace Collection**

We can modify the default Trace behavior so that the Trace system appends data to the existing Receiver buffer rather than clearing it.

- 1. In Code Composer Studio, select Tools ® XDS560 Trace ® Control... to bring up the Trace Control window once again.
- 2. Turn off Synchronize Trace with Target Execution and click Apply.

💀 Trace System Control	
Blackhawk USB560-M - TCI6482 Emulator/C6400PLUS_0 Select Pins 12-PIN Trace(EMU11:0) Sample Configuration Synchronize Trace with target execution Trace Buffer Type C Circular buffer Stop on buffer full Advanced	Select EBM Size 256K Bytes Settings Save Settings Load Settings Default Settings
About Help Receiver OK	Cancel Apply

- 3. When Trace is done reinitializing, click OK to dismiss the Trace Collection dialog.
- 4. Select Debug ® Run.
- 5. When the breakpoint is hit, go to the Trace Display and click on the Stop button.
- 6. The Trace Display contains just the one iteration of the loop.
- 7. Click the Resume button rather than the Start button.
- 8. Repeat the Run/Stop/Resume steps two or three more times.

Each time the display contains more data, having appended each iteration of the loop because we have been using the Resume button rather than the Start button. Use Start when you want the Receiver to clear its buffer before you run the target again. Use the Resume button when you want to append to the existing Receiver buffer.



### Choosing the Trace Buffer Type

There are two choices for the Trace Buffer Type in the Trace Control: Circular Buffer and Stop on Buffer Full. The default is Circular Buffer.

In Stop on Buffer Full, recording stops automatically when enough trace data is collected to fill the trace buffer. Recording can stop before the trace buffer is full if the target halts, an End Trace Action trigger occurs, or you hit the Stop button in the Trace Display.

In Circular Buffering, the buffer wraps and new trace data is stored starting at the beginning of the trace buffer when enough trace data is collected to fill the buffer, so that older trace

data is continually overwritten. Recording does not stop until the target stops executing, an End Trace Action trigger occurs, or you hit the Stop button in the Trace Display.

In Circular Buffering, if the buffer has not wrapped and recording stops, the data in the buffer contains everything from the time recording started until recording stopped. This is equivalent to Stop on Buffer Full. However, in Circular Buffering, if the buffer has wrapped and recording stops, the data in the buffer no longer contains information from when recording started, but only data backwards in time from when recording stopped.

Thus, in Circular Buffering the buffer usually wraps, which allows you to look backward from the point you stopped recording, while Stop on Buffer Full allows you to look forward in time from the point you started recording.

Circular Buffering would be good to use to determine the program flow that led to a problem at a known program location where the problem occurred. To perform this function, you would first program a Trace On action, set a breakpoint in your target application at the problem location, and then run your target application. When execution halts on the breakpoint, trace recording will be stopped. Trace Display will then show your target application actions from the time you hit the Stop button backwards in time, and allowing you to observe the problem path and discover the root cause of the problem.

Stop on Buffer Full would be good to use when your target application runs outside of the program image space defined in your .out/COFF file, soon after a certain function is called. In this case, you would set a breakpoint at that function, run your target application until it hits the breakpoint, then program a Trace On action and continue executing your target application from that breakpoint. When the trace buffer is full, recording will stop automatically, and Trace Display will show as much information as allowed by the trace data buffer capacity, starting from the breakpoint on. You can see exactly where your program flow started to run outside of the program image space because the Trace Status column will show the message *Bad PC*. If the trace data buffer capacity to capture the problem, you may use it to find a point further in the target application program flow to set another breakpoint so that you can capture the problem.

### 

#### **Changing the Trace System Receiver**

Trace assumes you are using a TI XDS560T receiver. You can change the receiver by following these steps:

1. Bring up the Trace Control window by selecting the Tools  $\circledast$  XDS560 Trace  $\circledast$  Control... menu item.

💀 Trace System Control	×
Blackhawk USB560-M - TCI6482 Emulator/C6400PLUS_0	
Select Pins 12-PIN Trace[EMU11:0]	Select EBM Size
Sample Configuration Synchronize Trace with target execution Trace Buffer Type Circular buffer C Stop on buffer full	Settings Save Settings Load Settings
Advanced	Default Settings
About Help Receiver OK	Cancel Apply

2. Click on the Receiver button. Highlight the desired receiver name and click OK.

Select Rec	eiver				
Blackhawk I	JSB560-M	- TCI64	482 Emul	lator/C640	OPLUS_0
None 560 Trace ETB	Pod				~
	OK		Can	cel	

3. Click Apply in the Trace Control window. The Trace System presents a dialog and a progress indicator in the Code Composer Studio status bar while it is initializing the Trace System.

# ◀►

### Summary of Trace Jobs

This tutorial used only one of the Trace actions, Trace On, and one type of trace, called Standard Trace, to demonstrate basic Trace System capability.

The other type of trace is called Event Trace, it allows you collect one of four categories of events:

- Memory event tracing records cache events, such as L1P Read Miss, L1D Write, and so on.
- Stall event tracing records the reason for the stall.
- Predication event tracing records whether an instruction executed (predicated true or false).
- External event tracing records events you program your target to generate. See the Interrupt Controller Spec for your target to learn how to program External Events. For example, for a C64x+ target, see the TMS320C64x+ DSP Megamodule Reference Guide (SPRU871E).

Note that the Trace On action starts recording trace immediately. All of the other Trace actions start recording when a trigger condition is true. The most frequently used trigger event is when the PC reaches a specified location or is within a specified range. Other triggers are possible, such as data memory accesses and events.

For more information about Event Trace and trace triggers, see the Trace System online help.

The following are other Trace actions with a brief description of what they do and some example use cases.

Trace On – The Trace On action turns trace on immediately. This action is useful for when you first start debug and do not know where in your program flow a problem may have occurred. See Choosing the Trace Buffer Type for examples.

- Start Trace and End Trace The Start and End Trace actions allow you to start and end trace at any program address location or within any program address range. The Start and End Trace actions also allow data memory triggers or event triggers. Thus, you can start or end collecting trace data whenever a write occurs to a data address range. Or you can start or end collecting trace data whenever a write occurs to a data address range. Or you can start or end collecting trace data whenever a write occurs to a data address range. Or you can start or end collecting trace data whenever a write occurs to a data address range. Or you can start or end collecting trace data whenever a write occurs to a data address range. Or you can start or end collecting trace data whenever a write occurs to a data address range. The start frant Trace actions follow the program beyond function call boundaries. You can have more than one Start/End trace action. As with Trace On actions, knowing approximately where the problem occurs helps assure that you can capture the problem within the trace data buffer you collect. Start/End trace is useful for debugging stack corruption problems since the captured program flow will show you where function returns do not return to the correct calling function.
- Trace In Range The Trace In Range action allows you to specify a particular program address range to trace, for example, a particular function or range of functions contiguous in memory (such as ISRs). The Trace In Range action does not trace program flow outside of the program range start/end addresses. The Trace in Range action is useful when you can limit the problem to a specific program address range. This is particularly useful in debugging Interrupt Service Routines, which otherwise can not be debugged by setting breakpoints. Collecting Time Stamp information for ISRs is useful for determining if you are exceeding the time in which the interrupt must be serviced.
- Don't Trace In Range The Don't Trace In Range action allows you to specify a particular program address range not to trace, such as a particular function or range of functions contiguous in memory (such as ISRs). The Don't Trace In Range action helps keep Interrupt Service Routine code from using space in the trace Receiver's buffer when the problem is known not to be in the ISR.
- End All Trace The End All Trace action turns off all tracing. That is, the trigger for a Start Trace action will not cause Trace to start recording again after an End All Trace trigger has occurred. The End All Trace action is useful if you have more than one Start Trace action and wish to turn off all tracing at a particular program address location or range in your program flow. If you explicitly hit the Start button in the Trace Display or have the Synchronize Trace with Target Execution Trace Control option on, then recording will start again when the trigger for a Start Trace action occurs.
- Trace Variable The Trace Variable action allows you to specify a variable name or absolute data address for which trace collection should occur. The Trace Variable action is useful when you know that a specific variable is being corrupted. Turning on the Collect PC w/Data option shows you the code that corrupted the data. It is important to note that you should not specify stack variables since their address is program-flow dependent. Another good example use of Trace Variable is to set a breakpoint beyond the initialization of a DSP data table and then add a Trace Variable action with the range of the data table to see how its values are changing. You can then export the data to a spreadsheet for graphing.
- Store Sample The Store Sample action records trace data for the current cycle whenever the trigger for the action is true. The Store Sample action is useful since it allows data memory, or event triggers, which some other Trace action do not. Thus, you can collect trace data whenever a write occurs to a data address range, not just one address as in the Trace Variable action. Or you can collect trace data whenever a particular event occurs, for example a cache hit or miss, or an interrupt acknowledge.
- Don't Store Sample The Don't Store Sample action prevents recording trace data for the current cycle whenever the trigger for the action is true. The Don't Store Sample action takes precedence over the Store Sample action. The general use of the Don't Store Sample action would be to prevent Store Sample actions from taking place under certain trigger conditions, such as eliminating a sub-range of a data address range (e.g., do not record data for accesses to array elements 6-10).
- Insert Trace Marker The Insert Trace Marker action records a special marker within the trace data whenever the trigger for the action is true. This marker is identified in the Trace Status column of the Trace Display with the message Trace Marker. Like the Store Sample action, the Insert Trace Marker action allows data memory, or event triggers. Thus, the Insert Trace Marker action can be used as an indication that you have reached a particular location in your program flow, that a write access to a particular data memory range has occurred, or a specific event has occurred, such as a cache hit or miss, a CPU stall, and so on.

All Trace actions allow you to specify what trace information to collect: Program Address, Write Address, Read Address, and so on. The more types of data you ask to collect, the more likely you will exceed the ability of the Trace Hardware to collect and transmit all of the data off the chip. You can prevent this by choosing the option in the Trace Control Advanced Settings... dialog to stall the CPU to allow the Trace Hardware time to export all data. Be aware that stalling the CPU has an impact on the real time aspect of your program.

There are many other Trace System options in both the Trace Control and the Trace Display which are meant to help you debug and optimize your target code. Please explore these options and use the online help, which explains the options and their potential uses. For example, the Event Sequencer includes trace actions that, combined with the AET state machine HW capabilities, provide you with more precise control over trace collection in complex program flow situations.

