HET IDE Tutorials

User's Guide



Literature Number: SPNU485C May 2010-Revised August 2016



Contents

1	Installat	ion	6
	1.1	Installing the HET IDE	6
	1.2	Tutorial Installation	6
2	Tutorial	S	7
	2.1	Tutorial 1 — Getting Started	
	2.2	Tutorial 2 — Input Stimuli	14
	2.3	Tutorial 3 — Changing Device and Clock Configuration	24
	2.4	Tutorial 4 — Complex Breakpoints	29
	2.5	Tutorial 5 — Memory Triggers	32
	2.6	Tutorial 6 — Algorithm Library and XOR Configuration	39
	2.7	Tutorial 7 — Input VCDs	46
	2.8	Tutorial 8 — Input Stimuli From WaveFormer Pro	48
Revis	ion Histo	ory	51

2



List of Figures

1	Shortcut to Tutorials Installation (Read Only) Folder	. 7
2	Waveform Wizard	9
3	Waveform Wizard	10
4	Breakpoints	11
5	Memory Window After Step	11
6	Controls for 'Run For Loops'	12
7	Waveform at Completion of Tutorial 1	13
8	New Project Dialog - Initial Screen	14
9	New Project - Device Selection	15
10	New Project - Clock Frequency	15
11	Insert Instruction Tool	16
12	Insert Instruction Tool - Instruction List	16
13	Completed PCNT Wizard, Required Fields in Red (optional fields may be left blank)	17
14	Error Because C00 is Not Yet Defined - This is Normal	18
15	Completed CNT Instruction Wizard	19
16	Complete Program for Tutorial 2	19
17	Stimulus Creator	20
18	Pin Trigger to Drive Pin 1 High Periodically	21
19	Pin Trigger to Drive Pin 1 Low Periodically	22
20	Completed Stimulus Creator Showing Both Triggers	22
21	Result of Simulation	23
22	Zoom in of Simulation Result Waveform to Inspect Stimulus on watch_in[31:0]	23
23	Opening Project Properties Dialog	24
24	Changing the Device Configuration	25
25	Selecting a Device From Device Configuration List	25
26	Project Properties Dialog Showing Newly Selected Device	26
27	Changing the Clock Configuration to 110 MHz HET Clock and LR Clock Divider to 16	27
28	HET Clock Period Measurement	28
29	HET Loop Resolution Clock Period Measurement	29
30	Complex Breakpoints Tab	30
31	Configuring a Complex Breakpoint to Stop When Pin 7 is High	30
32	Review Complex Breakpoint Set for Pin 7 High	30
33	Complex Breakpoint Hit	31
34	Stopped When Pin 7 Changes to 1 (watch_in[31:0] Changes From 0x21 to 0xA1)	31
35	Memory Triggers Tab	32
36	Memory Trigger Editor	33
37	Waveform Wizard	34
38	Adding an Instruction Trace to the Waveforms Output by the HET Model	35
39	Adding a New Signal to the Waveform Viewer	36
40	Double Click to Edit the New Signal	36
41	Signal Properties Corresponding to Instruction Trace at Address 0x40	37
42	Result of Memory Write (corrupted program field) can be Seen After Simulation in HET IDE Memory Viewer.	38
43	Waveform at Completion of Tutorial 5 (with added instruction trace)	38
44	Tutorial 6 Starts With a Blank HET Program	39
45	Edit \rightarrow Insert Algorithm to Open the HET Algorithm Library	39
46	Select Standard Output Examples for PWM Algorithms	

3



47	Three Channel PWM Function From the Algorithm Library	40
48	Algorithm Library	41
49	Waveform Without XOR Configuration	
50	Expanding the watch_out[31:0] Bus to View Individual Signals	42
51	Expanded Waveform Shows Individual Signals for Pins 0,1,2	43
52	XOR Share Settings	44
53	Simulation Results Showing XOR Sharing of Pins 0,1 Output onto Pin 0	45
54	Selecting an Input VCD File From the Project Properties Dialog	46
55	Simulation Results - Stimulus From VCD File Appears in 'watch_in[31:0]' Trace	47
56	VCD File Used for Stimulus by Tutorial 7	47
57	Input Stimulus Drawn Directly in Synapticad Waveformer Pro	48
58	Simulation Waveform From Tutorial 8	49
59	Additional Dummy Stimulus Added to Work Rround Time Zero Issue	49
60	Moving the Dummy Stimulus to the 1 ns Scale Provides Perfect Tracking for Practical Purposes	50



5



This document contains instructions for the HET IDE tutorials. The HET IDE is a free integrated development environment for the programmable high-end timer module. It supports the HET, NHET, and N2HET timers from the TMS470M, TMS570, RM4xx, and RM5xx Hercules[™] MCU product lines. The HET IDE includes a code editor, assembler, simulator and Waveform Viewer and editor. The code editor has an 'Insert Instruction' wizard to assist in learning HET assembly syntax, as well as an algorithm library that allows you to simply copy and paste frequently used timing functions into your own custom program. The assembler produces files for simulation and also for linking into your final application. The simulator allows you to debug your program with control and visibility that is not available when running your program on silicon. You can also stream waveforms to and stimulus from an external waveform tool (Synapticad Waveformer Pro) so that the interaction of the timer and the external world can be modeled.

1 Installation

1.1 Installing the HET IDE

The HET IDE can be downloaded from http://www.ti.com/lit/zip/spnc016. Once you download this file, unzip and run the included installer. This tutorial document should be used with HET IDE version 03.05.01 or later.

1.2 Tutorial Installation

The tutorials are automatically installed when the HET IDE is installed. However, you must make a personal copy of the tutorial folder before beginning to work through the tutorials.

The HET IDE installer places the tutorials in the "%PROGRAMDATA%\Texas Instruments\Hercules\HET IDE\<version>\Tutorials" folder. This folder may be hidden, so the installer also places a shortcut to open the folder in windows explorer in the start menu (see Figure 1). The installation folder is read-only and is shared by all users, so make a personal copy of the tutorials folder in your own directory location before continuing. In the following instructions, your copy of the HET IDE tutorials is referred to with the path <your_copy>.



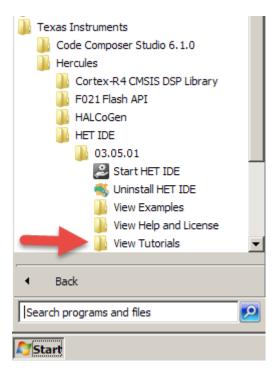


Figure 1. Shortcut to Tutorials Installation (Read Only) Folder

2 Tutorials

It is assumed that you have installed the HET IDE, as described in Section 1.1, and that you are working in your personal copy (<your_copy>) of the tutorials folder as described in Section 1.2

Also, between tutorials you should either close and restart the HET IDE, or use the RESTART button on the HET IDE toolbar to exit an existing simulation before moving on to the next tutorial.

- **NOTE:** Sometimes when changing projects or exiting the program, the Synapticad waveform viewer may present you with the option to "Save Files". It is very easy to miss this dialog and the HET IDE may appear unresponsive while this dialog is open. If the HET IDE appears unresponsive, switch the focus back to the Synapticad application and close this. Open the 'Save Files" dialog (Save None is a good choice for the tutorials). Then, the HET IDE will become responsive again.
- **NOTE:** If you have multiple projects open in the HET IDE, then your project must be made 'the active project' before you can assemble and load the project into the simulator. You can make any project 'the active project'; right click on the project name in the Project Explorer pane of the HET IDE and choose 'Set as Active Project' from the context menu. 'The active project' is always shown in **bold** face.
- **NOTE:** If the Assemble or Assemble and Load toolbar buttons are greyed out in the HET IDE, but the project is active and there are no simulations currently in progress, it sometimes helps to close the .het source file. Reopening the source file in the editor usually makes these buttons available again.

Tutorials

TEXAS INSTRUMENTS

www.ti.com

2.1 Tutorial 1 — Getting Started

2.1.1 Concepts Covered

This Tutorial covers the following topics:

- Opening a project
- Breakpoints
- Run a program
- Cycle count
- Waveform wizard
- View the waveform

2.1.2 Step-By-Step Instructions

- 1. Double click the "Start HET IDE" icon on the desktop, or from the start menu location shown in Figure 1.
- 2. Open the Project:
 - (a) Click Project \rightarrow Open Project to open the project.
 - (b) Open the *tut1_getting_started.prj* file from <your_copy>\Tutorials\tut1\tut1_overview\tut1_overview.prj
 - (c) Confirm that the project 'tut1_overview' is the active project. It should be listed in **bold** face in the Project Explorer pane of the HET IDE.
- 3. Build the Project:
 - (a) Click Debug \rightarrow Assemble.
 - (b) This step compiles the HET file and generates the result in the output window. Make sure that no errors are generated. If there are any errors, correct them and assemble it again.
- 4. Load an HET program:
 - (a) Click Debug \rightarrow Load HET Program to load the HET program into the RAM.
 - (b) You will see the program load into the simulator and Synapticad Waveformer Pro or Waveviewer Free should launch.
 - (c) Bring the HET IDE window into focus (in case Synapticad is currently in focus).



5. Configure the signals to be viewed in the waveform using Waveform Wizard of the HET IDE:
(a) In the HET IDE, click Tools → Waveform Wizard. Your screen should look like Figure 2.

	veFormer P		Described in a life	0-11	11Pa dava attala														_	
			ParameterLibs Vie																= 4	<u> </u>
	🧏 🛛 Veri		I 🚰 🛛 👬 🗤	1 Mii Mii Pi	and all 🔆] v] 🖻		Search											_ ₹,	
	t_signals.bt Signal Add B		lay Setup Sample					0 51												<u>×</u>
	Clock Add S		old Text Marker	HIGH LOW 1	TRI VAL INVal	쓰면 생산이 넣	EX 0 +	Q R												
3	5.40ns	35.40	ns Ons	2ns	4ns 6r	ns 8ns	10ns	12ns	14ns	16ns	18ns	20ns	22ns	24ns	26ns	28ns	30ns	32ns	34ns	36n:
0			utorials/tut1/tut1	L_overview)																
1		Ĺu.	View Tools Help																	
2		8	ъ s =	<i>©</i> ►	🐟 50 🛨	A 💀 🛛		лМ												
3	Program	memory s	ize 6 instructions																10	00.0000
5	tut1 over	view.het	Full Disassembly	C file He	ader file Parti	al Disassembly												Device	Stimulus cre	ator
6		bel Add							1	1								Simulator	N2HE	
7	A0	0x0	CNT			0 request=NOREC			comp=			ontrol=OFF								
8	A1 A2	0x10 0x20				0 request=NOREC 0 request=NOREC				=OFF en_pir =GE irq=O		ond_addr=2 ontrol=OFF		ion=PULSELO re ta=0	g=A irq=OH	data=6		Device Nar		70LS21
9	A3	0x30	ECM							Lore -					? ×	data=8		Num. of In	strs. 160	
10	A4 A5			brk=OFF brk=OFF	next=5 reqnum= next=0 regnum=	0 request=NOREC 0 request=NOREC 0 request=NOREC										data=4		⊡ 0	M 8	M
11							Clock	Pins Fields	Register	s Interna	Registers	Flags Ir	nstruction					M 1	9	M
14							HET Ck	ocks										M 2	M 10	M
							F HE	T Clock										M 3	11	M
							🔽 Hig	h Resolution Clock										₩ 4	12	2
							Con V	op Resolution Clock										M 5	₩ 13	M
																		 6	₩ 14	M
																		7	15	2
◀																				
Para	Output	Memory	1				Select a	ppropriate HET clock	<											
Adk nam	Label	Address	Program Field	Control Field	Data IR	Data HR Data F							Ol.				Internal			
ham	A0	0x0	0x00002c20	0x0000006	0	0 0x0000	X						Ok	C	ancel		Name		0	Contents
	A1 A2	0x10 0x20	0x00004100 0x00006c60	0x00404008 0x000000a		0 0x0000 0 0x0000				E-0x0 N				0x00000001						
	A2 A3	0x20	0x00008100	0x0040871a		0 0x0000				E-0x4 M				0x00000500 0x00000000						
	A4	0x40	0x0000aca0	0x0000000a		0 0x0000				E-0xc	HETOFF1			0x00000000						
	A5	0x50 0x60	0x00000100 0x00000000	0x0040c51c 0x00000000		0 0x0000 0 0x0000				E-0x10	HETOFF2 HETINTENAS			0x00000000 0x00000000						
		0x70	0x00000000	0x00000000	0	0 0x0000	0000				HETINTENAS			0,00000000						
		0x80 0x90	0x00000000 0x00000000	0x00000000 0x00000000		0 0x0000 0 0x0000				NHETGCR										
		0xa0	0x00000000	0x00000000		0 0x0000			-											
sim																				
sim																			1011	
											Not Conr	rected			5	ormulation I	nactive INS	u Lr	: 1 Col: 1	

Figure 2. Waveform Wizard

- (b) On the Waveform Wizard tab "Clock":
 - (i) Select HET Clock .
 - (ii) Select High Resolution Clock.
 - (iii) Select Loop Resolution Clock.
- (c) On the Waveform Wizard tab "Pins" :
 - (i) Select HET_0.
 - (ii) Select HET_5.
 - (iii) Select HET_7.
- (d) On the Waveform Wizard tab "Fields":
 - (i) Deselect PROGRAM.
 - (ii) Deselect CONTROL.
 - (iii) Select Data.
- (e) On the Waveform Wizard tab "Registers":
 - (i) Select NHETADDR.
 - (ii) Deselect all other registers.

- (f) On the Waveform Wizard tab "Internal Registers":
 - (i) Select A,B,T.
 - (ii) Deselect all other registers.
- (g) On the Waveform Wizard tab "Flags":
 - (i) Select Z.
 - (ii) Deselect all other flags.
- (h) On the Waveform Wizard tab "Instruction":
 - (i) Select address 0x0 from the Address drop-down list.
 - (ii) Select the PROGRAM, CONTROL, and DATA Instruction Fields.
 - (iii) Press the "Add" button. Make sure that you see address 0x0 listed in trace slot 0 ("SI. No").
 - (iv) Repeat steps i iii, but substitute addresses 0x20 and 0x40 for address 0x0.
 - (v) Confirm that you see address 0x20 in slot 1 and address 0x40 in slot 2.
- (i) Click OK.

	aveFormer P		ParameterLibs V	iew Options Windo	w Help		-	_								_1	미치
11.	Veri							e 🚑 🛛 Search								- 4 ,5) m
	et_signals.bt					110 1-		_],									×
Add	I Signal 🛛 Add B	Bus Di	elay Setup Sample	HIGH LOW TRI	VAL INVal	WHI WLO HE	× <u>+</u>	Q.F Q.R									
	I Clock Add S 35.40ns	35.40	old Text Marker	2ns 4ns			10ns		116ns	18ns 20ns	22ns	24ns 26ns	28ns	30ns	32ns	34ns	36n:
0			utorials/tut1/tu						. 1.010								10 și il I
1		Ĺ	View Tools Help					1									
2			ъ s	o? 🕨 🐟 🛛	50 🗄	A 👁 🕕		_ n M									
3 4	Program	n memory s	ize 6 instruction	5												10	0.0000
5	tut1_over	rview.het	Full Disassembly	C file Header f	ie Parti	al Disassembly									Device :	Stimulus crea	tor
6			ress Co												Simulator	N2HET	
7 8	A0		ON EC!			0 request=NOREQ 0 request=NOREQ				=OFF control=OFF pin_action=ON cond_addr=	max=6 data 2 pin=0 actio		data=6		Device Name		
9	A2	2 0x20) CN	T brk=OFF next+	=3 regnum=	0 request=NOREQ	angle_coun	t=OFF reg=B comp	=GE irq	=OFF control=OFF	max=10 data				Num. of Inst		OLD EX
10	A3			T brk=OFF next	=4 reqnum= =5 reqnum=	0 request=NOREQ 0 request=NOREQ	🚨 VCD s	election				<u>? ×</u>	data=8				
11	AS	5 0x50) ECI	MP brk=OFF next=	=0 reqnum=	0 request=NOREQ		Pins Fields Registe	rs Inte	mal Registers Flags	Instruction		data=4		0	8	M
12								Clocks				[M 1	9	M
								IET Clock							₩ 2	M 10	M
								ligh Resolution Clock							₩ 3	11	M
								.oop Resolution Clock							₩ 4	12	M
															5	M 13	
															6	14	N N
															7	15	IM.
Para		Memory]				Select	appropriate HET clock									
nan		Address		Control Field	Data LR	Data HR Data Fi					Ok	Cancel		Internal			
	A0	0x0 0x10	0x00002c20 0x00004100	0x0000006 0x00404008	-	0 0x00000 0 0x00000	200							Name		0	ontents
	A2	0x20	0x00006c60	0x0000000a	0	0 0x00000	000			NHETGCR NHETPFR		0x00000001 0x00000500					
	A3 A4	0x30 0x40	0x00008100 0x0000aca0	0x0040871a 0x0000000a		0 0x00000 0 0x00000				NHETADDR		0x00000000					
	A4 A5	0x40 0x50	0x0000aca0 0x00000100	0x0000000a		0 0x00000				NHETOFF1 NHETOFF2		0x00000000 0x00000000					
		0x60 0x70	0x00000000 0x00000000	0x00000000 0x00000000		0 0x00000				1 NHETINTENAS		0x00000000	-				
		0x80	0x00000000	0x00000000	0	0 0x00000	000		NHETGO	R							
		0x90 0xa0	0x00000000 0x00000000	0x00000000 0x00000000		0 0x00000 0 0x00000		-									
		- VAU		100000000	•	- 100000											
<u>sin</u>													N 1 6 1		0 [;	1011	
										Not Connected			simulation	nactive IN	S Ln:	1 Col: 1	

Figure 3. Waveform Wizard



- 6. Insert Break Points:
 - (a) Double click on Labels A1, A3, and A5 to insert break points.
 - (b) Confirm that your screen looks like Figure 4.

	HETIDE (F:/work/Tutorials/tuti/uti_overview) Project Edit Debug View Tools Help																							
Proje	ct E	Edit De	oug View Tools H	elp																				
P	Program memory size 6 instructions 100.000000 HHz, HR = 1, LR = 32																							
t	tut1_overview.het Ful Disassembly C file Header file Partial Disassembly C file Header file Partial Disassembly C																							
		Label	Address	Code		1							1						٦.	inulator	N2HET			
		A0	0x0	CNT	brk=OFF	next=1	regnum=0	request=NOREQ	angle_count=OFF	reg=A	comp=GE	irg=OFF	control=OFF	max=6	data=0				- °	mulator	NZHET			
	•	A1	0x10	ECMP	brk=OFF	next=2	regnum=0	request=NOREQ	hr_lr=LOW	angle_comp=OFF	control=OFF	en_pin_action=ON	cond_addr=2	2 pin=0	action=PULSELO	reg=A	irq=OFF da	ta=6	0	evice Name	TMS570L	S21XXZWT		
		A2	0x20	CNT	brk=OFF	next=3	regnum=0	request=NOREQ	angle_count=OFF	reg=B	comp=GE	irq=OFF	control=OFF	max=10	data=0					lum. of Inst	100			
	•	A3	0x30	ECMP	brk=OFF	next=4	reqnum=0	request=NOREQ	hr_lr=LOW	angle_comp=OFF	control=OFF	en_pin_action=ON	cond_addr=4	ipin=7	action=PULSEHI	reg=B	irq=OFF da	ata=8		un. or msu	5. [100			
		A4	0x40						angle_count=OFF	reg=T	comp=GE	irq=OFF	control=OFF	max=10	data=0					0	8	M 16	24	
	•	A5	0x50	ECMP	brk=OFF	next=0	reqnum=0	request=NOREQ	hr_lr=LOW	angle_comp=OFF	control=OFF	en_pin_action=ON	l cond_addr=6	ipin=5	action=PULSEHI	reg=T	irq=OFF da	ata=4						
																				1	9	17	25	
																				2	M 10	₩ 18	26	
																				M 3	M 11	M 19	27	

Figure 4. Breakpoints

- 7. Run the Program with different Run options:
 - (a) Click Debug \rightarrow Run (F5) to run the program. You should stop at the first breakpoint at label A1.
 - (b) Run two more times until reaching the breakpoint at label A5.
 - (c) Verify that the Data Fields of A0 (address 0x0), A2 (0x20) and A4 (0x40) all have the value 0x00000080 (Data LR = 1, Data HR = 0). You should see a memory window that matches Figure 5.

	Label	Address	Program Field	Control Field	Data LR	Data HR		<u> </u>	
_	A0	0x0	0x00002c20	0x0000006	1	0	0x0000080		Address
	A1	0x10	0x00004100	0x00404008	6	0	0x00000300	_	
	A2	0x20	0x00006c60	0x0000000a	1	0	0x0000080		0x4
	A3	0x30	0x00008100	0x0040871a	8	0	0x00000400		0x8
	A4	0x40	0x0000aca0	0x0000000a	1	0	0x0000080		+··Oxc
	A5	0x50	0x00000100	0x0040c51c	4	0	0x00000200		
		0x60	0x0000000	0x00000000	0	0	0x00000000		+·0x14
		0x70	0x0000000	0x00000000	0	0	0x00000000		
		0x80	0x0000000	0x00000000	0	0	0x00000000		NHETGO
		0x90	0x0000000	0x00000000	0	0	0x00000000		
_		0xa0	0x0000000	0x0000000	0	0	0x0000000	•	
-									

Figure 5. Memory Window After Step 7c

- (d) Run the program an additional three times (press F5 or Debug \rightarrow Run three more times).
- (e) Remove all the break points by double clicking at the labels. All of the red breakpoint symbols in the second column of the "Full Disassembly" tab should be cleared.
- (f) Click Debug → Step Instruction (or press F11) six times and see that it executes each individual instruction at a time.

- (g) Enter 20 in edit box (after Run for Loops icon) in the tool bar (see item 'g' in Figure 6).
- (h) Select Debug \rightarrow Run For Loops from the menu, or Click the Run for Loops button (see item 'h' in Figure 6).

roject	Edit De	bug View Tools	Help			0				
È	86) L 8	* 0	e 🕨	a 20		A 🔉 II		л M	
Proc	aram men	ory size 6 instruc	tions		0					
tut1	overview.	het Full Disassen	ably I C	file Hea		Dential	Disconsciole			
un	_overview.i	net Fuil Disassen		піе і неа	ader file	Partia	Disassembly			
	Label	Address	Code		ader file	Partial		[
	_							angle_count=OF	F reg=A	comp
	_ Label	Address	Code	brk=OFF	next=1	reqnum=0			F reg=A angle_comp=OFF	· · ·
	Label	Address 0x0	Code CNT	brk=OFF brk=OFF	next=1 next=2	reqnum=0 reqnum=0	request=NOREQ		angle_comp=OFF	contr
	Label A0 A1	Address 0x0 0x10	Code CNT ECMP	brk=OFF brk=OFF brk=OFF	next=1 next=2 next=3	reqnum=0 reqnum=0 reqnum=0	request=NOREQ	hr_lr=LOW angle_count=OF	angle_comp=OFF	contr comp
	Label A0 A1 A2	Address 0x0 0x10 0x20	Code CNT ECMP CNT	brk=OFF brk=OFF brk=OFF brk=OFF	next=1 next=2 next=3 next=4	reqnum=0 reqnum=0 reqnum=0 reqnum=0	request=NOREQ request=NOREQ request=NOREQ request=NOREQ	hr_lr=LOW angle_count=OF	angle_comp=OFF F reg=B angle_comp=OFF	comp

Figure 6. Controls for 'Run For Loops'

- (i) Check cycle count at the bottom of the GUI window. It should read 736 cycles (from an initial value of 101 cycles).
- 8. Stopping Execution:
 - (a) Click Debug \rightarrow Pause to pause the execution (if enabled).
 - (b) Click Debug \rightarrow Stop to stop the execution or press the stop button.



- 9. View the Waveform:
 - (a) Switch to the Synapticad Waveformer Pro or Waveviewer Free Application.
 - (b) Press the Zoom-Full Button (Magnifier with "F" beside it). Your screen should look like Figure 7.

<u>Ø</u>	Wav	veF	ormer Pro	
File	e In	npo	rt/Export Edit Bus Parame	eterLibs View Options Window Help
	lu l	<mark>] (</mark>	Verilog 🔽 🊰	# #!!! \# \# ># #!! 🛠 #!!! ☞ 🖬 🕼 🎒 Search
E	he	t_s	ignals.btim *	
			al Add Bus Delay Setu k Add Spacer Hold Tex	
	0.	.00	Ops 0.000ps	Ous 1us 2us 3us 4us 5us 6us 7us 8us
0	•	•	drive[31:0]	
1		•	het.het_clock	
2		•	et.high_resolution_clock	
3		•	et.loop_resolution_clock	
4	•		het.loop_count	
5		-	het.ir_a[31:0]	
6		•	het.ir_b[31:0]	(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aa
7		•	het.ir_t[31:0]	(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aaa.)(aa
8		•	het.NHETADDR	
9		•	het.flag_z	
1	0	•	het.flag_x	
1	1		watch_in[31:0]	<u> </u>
1	2		watch_out[31:0]	
				1

Figure 7. Waveform at Completion of Tutorial 1



Tutorials

2.2 Tutorial 2 — Input Stimuli

This tutorial covers the following features:

- Creating repetitive input stimulus using the built-in Stimulus Creator
- Using the Insert Instruction wizard to add instructions to a program.
- 1. Open the HET IDE.
- Create a new project, called my_tut2. From the HET IDE file menu, select Project → New Project. When the New Project dialog appears (see Figure 8), change the project name to 'my_tut2' and make sure the project path is pointing to a folder where you can read or write.

😢 New Project			<u>?</u> ×
Create HET Proje	ect		
Project name	my_tut2	[
Project path	F:/work/Tutorials		
Project directory	F:/work/Tutorials/my_tut2		
	< Back	Next >	Cancel

Figure 8. New Project Dialog - Initial Screen

3. Press the Next button to bring up a selection of HET devices. For this project, select the TMS570LS31XXZWT device (see Figure 9).

Device	Туре	HET Instructions	Pin Configuration
MISS70LS10XXX	INFIET	120	0,1,2,3,4,3,0,7,0,9,10
MS570LS11XXPGE	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
MS570LS11XXZWT	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
MS570LS12XXPGE	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
MS570LS12XXZWT	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
MS570LS20XXX	NHET	128	0,1,2,3,4,5,6,7,8,9,10
MS570LS20XXX	NHET	128	0,1,2,3,4,5,6,7,8,9,10
MS570LS21XXPGE	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
MS570LS21XXZWT	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
MS570LS31XXPGE	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
MS570LS31XXZWT	N2HET	160	0,1,2,3,4,5,6,7,8,9,10
•			

Figure 9. New Project - Device Selection

 Press the Next button again to bring up the Clock Frequency dialog. Set the clock frequency to 90 MHz (see Figure 10).

😕 New Project		? ×
Clock Frequency Settin	igs	
Clock frequency 90	MHz	
	< Back Next >	Cancel

Figure 10. New Project - Clock Frequency



- Tutorials
 - 5. Press Next and your project will be created with an empty HET assembly source file named my_tut2.het. To begin adding code, press the Insert Instruction toolbar button (see Figure 11).

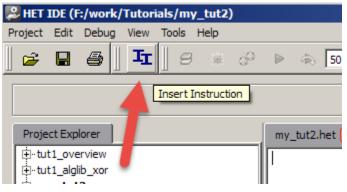


Figure 11. Insert Instruction Tool

 Add an input capture instruction - PCNT. Select PCNT from the list of instructions that appears (see Figure 12) and press Insert.

Instruction	Description	<u> </u>
ECMP	Equality Compare	
ECNT	Event Count	
MCMP	Magnitude Compare	
MOV32	Move 32	
MOV64	Move 64	
OR	Logical Or	
PCNT	Period/pulse Count	
PWCNT	Pulse Width Count	
RADM64	Register Add Move 64	
RCNT	Ratio Count	
SBB	Subtract With Borrow	
SCMP	Sequence Compare	
SCNT	Step Count	
SHFT	Shift	
SUB	Subtraction	-
•		
	bing in the editor the instruction name action in the table and clicking on "Inser	t" button. nsert

Figure 12. Insert Instruction Tool - Instruction List



7. After inserting the PCNT instruction, a wizard appears to help you fill out the required (red) and optional parameters for PCNT. In this case, the time between the rising and falling edges on pin 1 is captured. Therefore, select 'type' as 'RISE2FALL' and 'pin' as '1'. Create a label for this instruction so that other instructions can reference it, (label the instruction P01). Put a default value in the data field by setting period and data both to 0. Finally, after this instruction is finished, branch back to the beginning of the program. Even though the instruction has not been coded yet, label it C00; enter 'C00' in the 'next' field. All HET programs must branch back to the first instruction (at address 0) before the end of the loop resolution period, or an instruction overflow will occur. Your completed PCNT wizard will look like Figure 13. Note that many of the optional fields are left blank. This is ok, as the assembler assumes default values for all unspecified optional fields.

PCNT		<u>? ×</u>
Code Label	Pot	
hr_lr		Specifies the HIGH/LOW resolution.
brk		Defines the software breakpoint.
next	C00	Gives the Program address of next instruction in the program flow.
regnum		Gives the number of the request line(0, 1, 27) to trigger either HTU or DMA.
request		Allows to select type of request(NOREQ,GENREQ,QUIET).
irq	_	Maintains the control field of the remote instruction.
type	RISE2FALL	Determines the type of the counter that is to be implemented.
pin	1	Gives the pin number on which the action occurs.
control	•	When SET, clears the immediate data field when it is read.
prv	•	Gives the initial value(0 or 1)defining the previous bit.
period	0	Specifies the 25/20 bit initial count value for the data field.
data	0	Specifies the 25/20 bit initial count value for the data field.
Field names p	rovided in red color are n	nandatory
		Ok Cancel

Figure 13. Completed PCNT Wizard, Required Fields in Red (optional fields may be left blank)



Tutorials

- www.ti.com
- 8. Press OK on the PCNT wizard. Note that an error message appears (see Figure 14). After every instruction insertion through the wizard, the HET IDE tries to assemble your program to make sure that the instruction you just inserted is free from errors. In this case, the error is expected. The PCNT instruction was told to branch to the instruction labeled "C00' next, but that instruction was not yet entered so the assembly will fail. This is OK so select "Yes".

🔎 HET Si	imulator	×				
?	Following error(s) occured while parsing the existing source code:					
	NHET Assembler Release 1.7 Texas Instruments Incorporated.					
	Texas Instruments Incorporated. PASS 1					
	PASS 2 *** ERROR! EOF: E0300: The following symbols are undefined: C00					
	1 Error, No Warnings					
	Errors in Source - Assembler Aborted					
	Couldn't parse new instruction. Do you want to add it without parsing?					
	Yes No					

Figure 14. Error Because C00 is Not Yet Defined - This is Normal

9. Now, create the instruction for label C00. Use the Insert Instruction tool again, but select CNT from the instruction list instead of PCNT. The CNT instruction wizard should appear.



10. Populate the CNT instruction wizard (see Figure 15). The label should be C00. The next field could be blank, in which case the assembler will choose the next instruction in the program. In this case, it is explicitly specified as P01. Note that the next instruction is not *required* to be sequential and can be any instruction. We want the counter to count between 0, 1, 2, ... 10 and then roll over to begin at 0 again. So, a max value of 10 and an initial data value of 0 is placed in the wizard. Also, the count should be placed in register A. When you've completed populating the CNT instruction wizard, press OK.

CNT		<u>?×</u>
Code Label	C00	
brk	•	Defines the software breakpoint.
next	P01	Gives the Program address of next instruction in the program flow.
reqnum		Gives the number of the request line(0,1,27) to trigger either HTU or DMA.
request	•	Allows to select type of request(NOREQ,GENREQ,QUIET).
angle_count	•	A value of ON increments the counter value only if NewAngle Flag is SET.OFF increments counter value each time CNT is executed.
reg	A	Tells the internal register that is used for data comparison and storage.
comp	•	EQ resets counter when it is equal to max count,GE resets counter when it is greater or equal to max count.
irq	•	If set as ON an interrupt is generated when the edge state is satisfied.
control	OFF 💌	When SET, dears the immediate data field when it is read.
max	10	Specifies the 25/20 bit integer value that defines the maximum count value allowed in the data field.
data	0	Specifies the 25/20 bit integer value serving as a counter.
Field names pr	ovided in red color are ma	andatory
		Ok Cancel

Figure 15. Completed CNT Instruction Wizard

11. You should now have a complete program consisting of two instructions (see Figure 16). Confirm that C00 is the first instruction in the program, if not swap C00 and P01 using the editor. This program periodically counts between 0...10 and uses this counter as timebase to measure the interval between rising and falling edges of pin 1.

😂 HET IDE (F:/work/Tutorials/my	_tut2)
Project Edit Debug View Tools	Help
] 🛎 🖬 🖨 🛛 🎞 🛛 S	🗰 🕫 🕨 🚋 🏚 🗇 II 🔳 🔯 🗍 🎵 🕅
Project Explorer	my_tut2.het 🗵
-tut1_overview -tut1_alglib_xor -my_tut2 -my_tut2.het	<pre>CON CNT { next=P01,reg=A,control=OFF,max=10,data=0}; P01 PCNT { next=C00,type=RISE2FALL,pin=1,period=0,data=0};</pre>

Figure 16. Complete Program for Tutorial 2

12. Assemble and Load your new program into the HET IDE's simulator.



- Tutorials
 - 13. To test a program that performs an input capture function, stimulus on the capture pin must be provided during simulation. There are several ways to provide stimulus; in this tutorial, the built-in Stimulus Creator is used. When the simulator starts, find the Stimulus Creator tab in the upper right hand pane of the HET IDE window (see Figure 17).

	90.000000 MHz, HR = 1, LR = 32									
	Device	Stimulus creator	Memory Tri	ggers C · ·						
ł		Add	Del	Edit						
		Γ	Sort by time							
L	Trigger	Pin	V	alue						
L										

Figure 17. Stimulus Creator



14. Press the "Add" button to add input stimulus. A Pin Trigger to drive pin 1 high has been set up, starting at 250 ns into the simulation and repeating every 100 ns. To accomplish this, configure the pin trigger dialog (see Figure 18). Pay close attention to the units; the time unit defaults to 'ps' so be sure to change it to 'ns' in both places. When you have configured this trigger, press OK.

📙 Set Pin Triggers	?)
Trigger	
Time 250	NS 💌
C Cycle	
Action On	
Pin	1
Value	
C 0	
● 1	
C Log_X : Unknown State	
C Log_Z : High Impedance	
Repeat	
After	
Time 100	NS
C Cycle	
	Ok Cancel

Figure 18. Pin Trigger to Drive Pin 1 High Periodically



15. Press "Add" a second time, to add a second pin trigger. Configure the trigger to drive pin 1 low again, starting at 300 ns into the simulation and repeating every 100 ns. The completed dialog for this trigger is shown in Figure 19

🕹 Set Pin Triggers 🔶 🤶
Trigger
© Time 300 NS 💌
C Cycle
Action On
Pin 1
Value
· • •
C 1
C Log_X : Unknown State
C Log_Z : High Impedance
Repeat
After
Interimentation ● Time ■ 100 ■ NS ■
C Cyde
Ok Cancel

Figure 19. Pin Trigger to Drive Pin 1 Low Periodically

16. At this point, your stimulus creator should look like Figure 20. The combination of both pin triggers with repetitive periods, is to drive pin 1 high at 250 ns, low at 300 ns, and then continually repeat. This creates a toggling input stimulus on pin 1 with a period of 100 ns.

90.000000 MHz, HR = 1, LR = 32							2
	Device Stimulus creator Memory Triggers C						
-	Add			Del Edit			
	☐ Sort by time						
	Trigger		Pin	Value	Repeat		
		: 300 NS : 250 NS	1 1	Low High	Time : 10 Time : 10		

Figure 20. Completed Stimulus Creator Showing Both Triggers

Tutorials



17. Run the simulation for 50 loops, then switch to the Waveform Viewer and zoom so that the full simulation result is visible. You should see a waveform similar to Figure 21.

Γ	Ø: w	/aveF	ormer Pro							
	File	Impo	t/Export Edit Bus Parame	terLibs View Options Window Help						
	Image: Store and Hide Stores									
			Add Bus Delay Setup Add Spacer Hold Text							
		7.33		0us 1us 2us 3us 4us 5us 6us 7us 8us 9us 10us 11us 12us 13us 14us 15us 16us 17us 1						
	0	-	drive[31:0]							
HET IDE	1	•	het.het_clock							
ject Ec	2	•	et.high_resolution_clock							
2 G	3	-	et.loop_resolution_clock							
	4	-	het.loop_count							
Progran	5	•	het.ir_a[31:0]							
my_tut2	6	•	het.ir_b[31:0]							
	7	•	het.ir_t[31:0]							
> c	8	-	het.NHETADDR							
P	9	•	het.flag_z							
	10	•	het.flag_x							
	11	-	watch_in[31:0]							
	12	-	watch_out[31:0]	300000000						



18. Zoom in so that you can see transitions on the watch_in[31:0] bus. You will see that Pin 1 is toggling every 100 ns, just as it was configured to do through the stimulus creator. (The value of the bus toggles between 0x02 and 0x00). A later tutorial explains how to expand the bus and show the signals individually.

E het_signals.btim *						
		Add Spacer Hold Tex				
9	150.1	Ons 100.0ns	800ns 850ns 900ns 950ns 1.00us 1.05us 1.10us 1.15us	1.20us 1.25us		
-1	•	hetm.drive_0.in				
0	-	drive[31:0]				
1	-					
2						
3		et.loop_resolution_clock	Λ			
4	-	het.loop_count	0000002	0000003		
5	-	het.ir_a[31:0]	00000002 (00.)	0000000		
6	•	het.ir_b[31:0]				
7	•	het.ir_t[31:0]				
8	•	het.NHETADDR	0 × 1 ×	0		
9	•	het.flag_z				
10	•	het.flag_x				
11	-	watch_in[31:0]	22 / 00000000 / 0000002 / 00000000 / 0000002 / 00000000	00000000 0000002		
12	-	watch_out[31:0]	Next State: 0000000			

Figure 22. Zoom in of Simulation Result Waveform to Inspect Stimulus on watch_in[31:0]



Tutorials

2.3 Tutorial 3 — Changing Device and Clock Configuration

This tutorial covers the following features:

- Changing Device configuration
- Clock configuration
- 1. Open the HET GUI.
- 2. Open the Project:
 - (a) Click Project \rightarrow Open Project to open the project.
 - (b) Open the *tut3_config.prj* file from <your_copy>/Tutorials/tut3_config.
- 3. Change Device Configuration:
 - (a) Right click on tut3_config in Project explorer and select project properties (see Figure 23).

		ork/Tutorials/tut3_con	fig)		
8×F	Project Edit Deb	ug View Tools Help			
	🖻 🖥 🎒		ð?	⊳	الله الله الم
×	Project Explorer			tut2	2_input_vcd.h
	tut1_overview tut1_alglib_xc tut2_input_st tut2_input_vc tut2_wavefor tut2_wavefor tut3_config	or imuli id m_editor Set HET File		A0 A1 A2 A3 A4 A5	CNT {reg ECMP { r CNT {reg ECMP { r

Figure 23. Opening Project Properties Dialog



(b) When the project properties dialog opens, make sure Device is selected in the left pane. Then, click on the "Change Device" button (see Figure 24).

tut3_config Settings			?×
Device Clock Stimuli VCD XOR AND SHARE	Name of device Type Number of instructions Pin configuration	TMS570LS21XXZWT NHET 160 0,1,2,3,4,5,6,7, 8,9,10,11,12,13,14, 16,17,18,19,20,21,2 24,25,26,27,28,29,3 Change Device	22,23,
	Note: This device exists	in e database but n	nodified. Cancel

Figure 24. Changing the Device Configuration

(c) A list of supported Hercules devices will appear. Select the RM48LXXXZWT with N2HET and 160 words of instruction memory (see Figure 25). Then press OK.

Device	HET Instructions	Pin Configuration	Type 🔺
RM44LXXXZWT	160	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,1	N2HET
RM46LXXXPGE	160	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1	N2HET
RM46LXXXZWT	160	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1	N2HET
RM48LXXXPGE	160	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1	N2HET
RM48LXXXZWT	160	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,1	N2HET
RM57L843ZWT	256	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1	N2HET
TMS470MF03107PZ	64	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,	HET
TMS470MF04207PZ	64	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,	HET
TMS470MF06607PZ	128	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 24, 25,	HET 🗸
•			

Figure 25. Selecting a Device From Device Configuration List

(d) The project properties dialog still shows the previous device. Press OK to apply the change and close the project properties dialog.

(e) Repeat step 3a, opening the project properties dialog again. This time you should see that your change to the device name has taken effect (see Figure 26).

😢 tut3_config Settings			? X
Device Clock Stimuli VCD XOR AND SHARE	Name of device Type Number of instructions Pin configuration	RM48LXXXZWT NHET 160 0,1,2,3,4,5,6,7, 8,9,10,11,12,13,14,15, 16,17,18,19,20,21,22,23, 24,25,26,27,28,29,30,31, Change Device in the database.	
		Ok Cance	el

Figure 26. Project Properties Dialog Showing Newly Selected Device

- 4. Change the Clock Configuration:
 - (a) With the project properties still open from step 3e, select "Clock" in the left hand pane.
 - (b) Change the HET Clock Frequency to 110 MHz (9.09 ns period) (see Figure 27).
 - (c) Change the Loop Resolution Prescale to 16 (see Figure 27). Leave the HR Prescaler at 1. With these settings, the Loop resolution clock period should be 16 * 9.09 ns = 145.5 ns.
 - (d) Press the OK button to close the dialog.

😢 tut3_config Settings		? ×
Device Clock Stimuli VCD XOR AND SHARE	HET Clock frequency 110.000 Loop Resolution Prescaler 16 HR Prescaler 1 Number of HET Cycles in a Loop Resolution Period : 16 HET Prescaler HET (3 bits) Loop resolution of HET Cycles in a Loop Resolution Period : 16 HET Cycles in a Loop Resolution Period : 16 Het Prescaler (3 bits) Loop resolution of HET Cycles in a Loop Resolution of HET Cycles in a Loop Resolution of HET Cycles in a Loop Resolution Period : 16	•
	Ok Cancel	

Figure 27. Changing the Clock Configuration to 110 MHz HET Clock and LR Clock Divider to 16

- (e) Press the "Assemble and Load" toolbar button to load the program into the simulator. Then, run the simulation for 20 loops.
- (f) Switch to the Waveformer window to inspect the clock frequencies. You can measure the clock period (1/frequency) in the Waveformer with these steps:
 - (i) Left click on one edge of the clock signal in the waveform display so that a green marker appears on the edge.
 - (ii) Move the mouse to hover over the next edge of the same direction (rise or fall).
 - (iii) Read the delta time from the Waveformer display; it will be listed in blue above the signal list.

- 💁 WaveFormer Pro File Import/Export Edit Bus ParameterLibs Options Window View Help ALL 😵 JTT HIDE HIDE Sea 12 Verilog -4# ▶# 📙 🕼 🦣 🖨 Ē 1 **₩**# 11.11 het_signals.btim = Add Signal Add Bus Setup Sample + 🔍 F LOW WLO HIGH TRI VAL INVal WHI HEX Add Clock Add Spacer Hold Text Marker Q.R. 0 300.0ns 9.09ns 300ns 350ns 400ns 450ns k/ 0 drive[31:0] g 1 het.het_clock 2 et.high resolution clock 3 et.loop resolution clock ry 4 het.loop_count 00000001 5 t het.ir_a[31:0] 00000001 Add 6 het.ir_b[31:0] 00000001)x() 7 het.ir_t[31:0] 00000080)x1 8 het.NHETADDR 2 3 4 5 0)x2 9)x3 het.flag_z)x4 10 het.flag_x)x5 11 watch_in[31:0] 12 watch_out[31:0]
- (g) Using the technique described in step 4f, verify that the HET clock matches your setting of 110 MHz (9.09 ns period) (see Figure 28).

Figure 28. HET Clock Period Measurement

- 🚆 het_sig Add Signal (Add Bus Delay Setup Sample Hold Text Marker <mark>Q</mark>,F Q,R LOW **Q**+ TRI VAL INVal HEX win Add Clock Add Spacer 100ns 281.8ns 145.4ns |150ns 200ns 250ns 300ns 350ns 40 [31:0] 0 -1 het.het Л лллл 2 • et.high_resolution_clock 3 • et.loop_resolution_clock 4 het.loop_count 00000001 00000002 5 het.ir_a[31:0] 00000001 00000002 6 00000001 het.ir_b[31:0] 00000002 7 het.ir t[31:0] 00000080 0000010 8 het.NHETADDR 2345 0 1 2 3 4 5 0 9 het.flag_z 10 het.flag_x 11 watch_in[31:0] 00000000 12 watch_out[31:0]
- (h) Using the technique described in step 4f, verify that the HET loop resolution clock prescaler is 16 (145.5 ns) by measuring the loop resolution clock period (see Figure 29).

Figure 29. HET Loop Resolution Clock Period Measurement

2.4 Tutorial 4 — Complex Breakpoints

This tutorial covers the following features:

- Complex breakpoints
- 1. Open the HET GUI.
- 2. Open the Project:
 - (a) Click Project \rightarrow Open Project to open the project.
 - (b) Open the *tut1_overview.prj* file from <your_copy>/Tutorials/tut1_overview folder .

NOTE: Do not use the project in the tut4_complex_breakpoint folder.

- 3. Assemble the program (Debug \rightarrow Assemble).
- 4. Load the HET program (Debug \rightarrow Load HET Program).



Tutorials

5. Select the "Complex Breakpoints" tab in the top right pane of the HET IDE (see Figure 30).

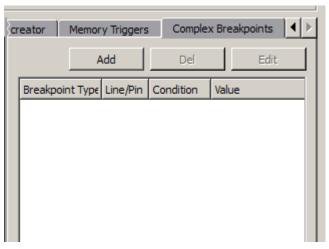


Figure 30. Complex Breakpoints Tab

- 6. Press the "Add" button to open the Set complex Breakpoints dialog.
- 7. Select a "Pin" type breakpoint, then choose Pin Number '7' and a Level of '1' (see Figure 31).

	?
7	.
Ok	Cancel
	7

Figure 31. Configuring a Complex Breakpoint to Stop When Pin 7 is High

8. Press 'Ok' and confirm that the breakpoint is displayed in the Complex Breakpoints tab. You should see Breakpoint Type listed as "Pin", Line/Pin as 7, and Value as 1 (see Figure 32).

reator	Memory Triggers		Complex Breakpoints			
R	Add Del		Del	Edit		
Breakp	oint Type	Line/Pin	Condition	Value		
Pin		7		1		



9. Run the program.



10. The program should stop with a message "Hit a complex breakpoint at pin 7" (see Figure 33).

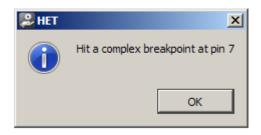


Figure 33. Complex Breakpoint Hit

11. Switch to the Waveform Viewer. Zoom full, then zoom out once, so that the end of the simulation can be seen within the waveform window. Hover the mouse pointer over the very last edge of the watch_in[31:0] bus. You will see that the value has changed from 0x21 to 0xA1 exactly where it was stopped with the breakpoint. The difference between 0x21 and 0xA1 is that bit 7 is set. Bit 7 contains the value for pin 7. This shows that the breakpoint has stopped the HET simulation when bit 7 became 1, as it was configured (see Figure 34).

i h	et_s	ignals.btim *	
		al Add Bus Delay Setu < Add Spacer Hold Tex	
	1.30	9us 1.309us	Ons 200ns 400ns 600ns 800ns 1.0us 1.2us 1.4us
0	-	drive[31:0]	
1	-	het.het_clock	
2	-	et.high_resolution_clock	
3	-	et.loop_resolution_clock	
4	-	het.loop_count	(00000001)(0000002)(00000003)(0000004)(00000005)(0000006)(0000007)(0000008)
5	-	het.ir_a[31:0]	(00000001)(0000002)(00000003)(0000004)(00000005)(0000006)(00000000)(00000001)
6	-	het.ir_b[31:0]	(00000001)/ 00000002)/ 00000003)/ 00000004)/ 00000005)/ 00000006)/ 00000007)/00000008
7	-	het.ir_t[31:0]	00000080 % 00000100 % 00000180 % 00000200 % 00000280 % 00000300 % 00000380 %00000400
8	-	het.NHETADDR	
9	-	het.flag_z	
10	-	het.flag_x	
11	-	watch_in[31:0]	00000000 00000000 000000000000000000000
12	-	watch_out[31:0]	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 34. Stopped When Pin 7 Changes to 1 (watch_in[31:0] Changes From 0x21 to 0xA1)

12. Using the "DEL" button in the Complex Breakpoints tab, Delete the complex breakpoint for Pin 7.



2.5 Tutorial 5 — Memory Triggers

This tutorial covers the following features:

- Memory triggers
- Tracing the waveform
- 1. Open the HET GUI.
- 2. Open the Project:
 - (a) Click Project \rightarrow Open Project to open the project.
 - (b) Open the *tut5_mem_trig.prj* file from <your_copy>/Tutorials/tut5_mem_trig.
 - (c) Outside the HET IDE, open the folder <your_copy>/Tutorials/tut5_mem_trig in Windows Explorer
 - (d) Delete the *'het_signals.btim'* file; it is outdated. The HET IDE automatically refreshes this file with a clean copy before starting a simulation.
- 3. Switch back to the HET IDE.
- 4. Assemble the program (Debug \rightarrow Assemble).
- 5. Load the HET program (Debug \rightarrow Load HET Program).
- 6. Set Memory Triggers:
 - (a) Select the Memory Triggers tab from the top-right pane of the HET IDE window (see Figure 35).

	100.000	0000 MHz,	, HR = 1, LR = 32
Device	Stimulus creator	Memory	Triggers C
	Add	Del	Edit
Trigger	Address		Value

Figure 35. Memory Triggers Tab

- (b) Click Add. The Memory Trigger window appears (see Figure 36).
 - (i) In the Trigger pane, enter value as 2500 and change units to NS by clicking the arrow.
 - (ii) In the Action pane, enter the address to be modified as 0x40, and enter the value to be written as 13.

😢 Edit Memory triggers	5				<u>?</u> ×
Trigger					
 Time 	2500			NS 💌	[
C Cycle					
🔿 Pin Change	0		∇	Level 0 💌	
C Interrupt	0		$\overline{\nabla}$		
-Action					
Address to be modi	fied: 0x	40			
Value to be written:	:	0xd			
			Ok	Cance	

Figure 36. Memory Trigger Editor

(iii) Click OK.



- (c) Open Waveform Wizard (see Figure 37).
 - (i) Go to Tools \rightarrow Waveform wizard.

😕 VCD selection	<u>?</u> ×
Clock Pins Fields Registers Internal Registers Flags Instruction	
F HET Clock	
 High Resolution Clock Loop Resolution Clock 	
Select appropriate HET clock	
Ok Cancel	

Figure 37. Waveform Wizard



(ii) Go to the Instructions tab. Select 0x40 in the address drop-down list, check Program in instructions fields, and click add (see Figure 38).

🚨 VCD sele	ection						?	x
Clock	Pins Fields	Registers	Internal Registe	ers Flags	Instruction	1		
Address	0x	40	•				Add	
Instruct	tion Fields							
	DGRAM							
	NTROL							
	ΓA							
		2	C 1					
SI.No A		Program TRACE	Control	Data				
	X10	INACE						
1								E
_							Delete	E
Enable th	he instructions to vi	ew in the VCD(N	4aximum 10 inst	ructions can be	selected.)			
					Ok		Cancel	1
							Curreer	1

Figure 38. Adding an Instruction Trace to the Waveforms Output by the HET Model

(iii) Make sure that you see SI. No 0 populated with address 0x40 (see Figure 38), then click on OK.



7. Switch focus to the Synapticad Waveform Viewer and press the Add Signal button (see Figure 39).

-	
🖉 Wa	veFormer Pro
File	mport/Export Edit Bus ParameterLibs View O
על∥	🔀 🛛 Verilog 💽 🚰 🗍 🏰 🗤 Ka
E.h	t_signals.btim
	Signal Add Bus Delay Setup Sample Ck Add Spacer Hold Text Marker HIGH
1	.0ps 390.0ps Ons 2ns
0	drive[31:0]
1	het.het_clock
2	et.high_resolution_clock
3	et.loop_resolution_clock
4	 het.loop_count
5	het.ir_a[31:0]

Figure 39. Adding a New Signal to the Waveform Viewer

8. The newly added signal appears at the bottom of the signal list (see Figure 40).

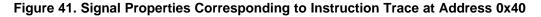
9		het.flag_z	
10		het.flag_x	
11	-	watch_in[31:0]	
12	-	watch_out[31:0]	
13	-	SIG0	_
		Name: SIG0	

Figure 40. Double Click to Edit the New Signal



- 9. Double click on the newly added SIG0 to open the Signal Properties Dialog. Then make the changes shown in Figure 41.
 - (a) Change the signal name to 'het.mem_0x40_program'. This name must be entered exactly, as it must match the name exported by the HET simulator.
 - (b) Change the signal type to Watch.
 - (c) Change the signal direction to Input.
 - (d) Change the signal to a 32-bit Bus, with MSB=31 and LSB=0.
 - (e) Press OK to save the changes.

Signal Properties
Name: het.mem_0x40_program 🚺 🗖 Active Low
Simulate Once Analog Props Grid Lines
O Drive O Simulate O Watch O Compare
Equation Entry Verilog VHDL
Type: Boolean Eqn 💌 ex. (SIG1 and SIG2) delay 5
Clock: Unclocked 💌 Edge/Level: pos 💌
Set: Not Used 💌 Clear: Not Used 💌
Clock Enable: Not Used 💌 Advanced Register
Use Waveform from Library
Wfm Eqn 8ns=Z (5=1 5=0)*5 9=H 9=L 5=V 5=X
Label Egn Hex(Inc(0,2,5))
Export Signal Dire3n: input
Analog Display Size Ratio: 1
Signal Type: 4_state
Radix: hex Bus MSB: 31 LSB: 0
☐ Falling Edge Sensitive ☐ Rising Edge Sensitive
OK Cancel Apply Prev Next



- 10. Switch focus back to the HET IDE.
- 11. Run the program for 20 loops.



Tutorials

12. In the Program field of the Memory window at address 0x40, change to 13 (0xD) when time>=2500 ns (see Figure 42).

Label	Address	Program Field	Control Field	Data LR	Data HR	Data Field	
A0	0x0	0x00002c20	0x0000006	0	0	0x00000000	
A1	0x10	0x00004100	0x00404008	6	0	0x00000300	
A2	0x20	0x00006c60	0x0000000a	5	0	0x00000280	
A3	0x30	0x00008100	0x0040871a	8	0	0x00000400	
44	0x40	0x000000d	0x0000000a	5	0	0x00000280	
A5	0x50	0x00000100	0x0040c51c	4	0	0x00000200	
	0x60	0x00000000	0000000	0	0	0x00000000	
	0x70	0x00000000	0x0u. 0000	0	0	0x00000000	
	0x80	0x00000000	0x00000000	0	0	0x00000000	
	0x90	0x00000000	0x00000000	0	0	0x00000000	
	0xa0	0x00000000	0x00000000	0	0	0x00000000	

Figure 42. Result of Memory Write (corrupted program field) can be Seen After Simulation in HET IDE Memory Viewer

- 13. Switch focus to Synpaticad Waveform Viewer to see the resulting waveform with trace.
- 14. After zooming, your waveform should appear like the one shown in Figure 43. Note that the memory trigger, that you setup to over-write the instruction at address 0x40, can be seen taking effect in the trace of HET.mem_0x40_program[31:0] at 2500 ns (2.5 μs), which matches the trigger time you configured. The value 13 is displayed in hexadecimal format in the waveform listing as 0xD.

🙆 Way	veFormer Pro			<u>_ 0 ×</u>
File In	nport/Export	Edit Bus ParameterLibs View	Options Window Help	
יען ₪	Verilo		# 4# ># ## 和 茨 ## 🛎 🖬 🗿 🏨 🖨 Search	▼ 4 ₈₃ ▶ ₈₃
🚍 het	t_signals.btir	1*		×
	ignal Add Bu: lock Add Spa			
	2.500us	0.000ps	Dus 1us 2us 3us 4us 5us 6us 7us 7us 8us	9us 10us
0 -	-	drive[31:0]		•
1	►	het.het_clock		
2	•	het.high_resolution_clock		
3	•	het.loop_resolution_clock		
4 -	-	het.loop_count	(000./000./000./000./000./000./000./000	00.\000.\000.\000.\000.\000.\000.\000.
3 4 - 5 6	•	het.ir_a[31:0]	0000000	1
6	►	het.ir_b[31:0]		
7	•	het.ir_t[31:0]		
8	•	het.NHETADDR		0 0 0 0 0 0 0 0
9	•	het.flag_z		
10	•	het.flag_x		
11 -	-	watch_in[31:0]	0000 (0000
	-	watch_out[31:0]	xxxxxxx [bx:]	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
13	🕨 he	.mem_0x40_program[31:0]	0000AF43	000000D
			Prev State: 0000AF43 Next State: 000000D	
1				
1				

Figure 43. Waveform at Completion of Tutorial 5 (with added instruction trace)

15. A complete list of the signal names that can be exported by the HET simulator and traced with the Synapticad Waveform Viewer can be found in appendices of the *HET IDE User's Guide* (SPNU483). Remember to match the signal names exactly, as the HET simulation model has already been compiled and its port names are fixed.



2.6 Tutorial 6 — Algorithm Library and XOR Configuration

This tutorial covers the following features:

- Inserting Algorithm from Algorithm library
- Selecting Algorithm family
- Parameter values selection
- XOR configuration
- 1. Open the HET GUI.
- 2. Open the Project:
 - (a) Click Project \rightarrow Open Project to open the project.
 - (b) Open the *tut6_alg_lib.prj* file from <your_copy>/Tutorials/tut6_alg_lib.
 - (c) Outside the HET IDE, open the folder <your_copy>/Tutorials/tut6_alg_lib in Windows Explorer
 - (d) Delete the *het_signals.btim* file; it is outdated. The HET IDE automatically refreshes this file with a clean copy before starting a simulation.
- 3. Switch focus back to the HET IDE. You should see that the source file for this tutorial begins with an empty file (except for the initial comment). Functionality is added by using the Algorithm Library.

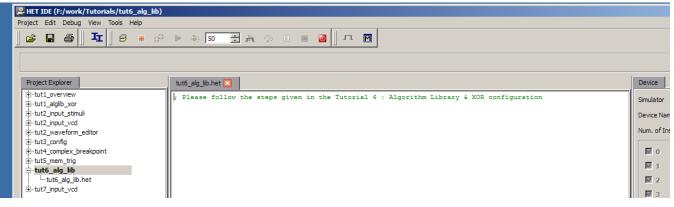


Figure 44. Tutorial 6 Starts With a Blank HET Program

- 4. Insert new algorithm:
 - (a) Click Edit \rightarrow Insert Algorithm (see Figure 45).

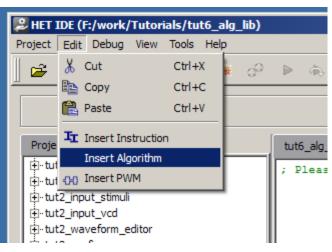


Figure 45. Edit \rightarrow Insert Algorithm to Open the HET Algorithm Library

Tutorials



Tutorials

(b) Select Standard Output Examples (see Figure 46).

😕 HET Algorithm Library					
Algorithm Families	Standard Input Examples 🔻				
Algorithms	Standard Input Examples Standard Output Examples				
Input Pulse/Period measurements with PCNT					

Figure 46. Select Standard Output Examples for PWM Algorithms

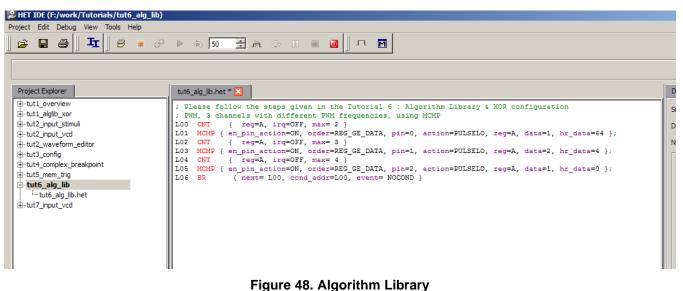
(c) Select PWM, three channels with different frequencies, using MCMP (see Figure 47).

😕 HET Algorithm Library			<u>?</u> ×
Algorithm Families	Standard Output Examples 💌		
Algorithms	Parameters		
Simple PWM PWM with synchronous duty cycle update at compare ma PWM with synchronous duty cycle update at period end PWM, 1 channel, using MCMP PWM, 4 channel with same PWM frequency using MCMP PWM, 3 channels with different PWM frequencies, using PWM, 1 channel with buffer field to modulate PWM period Frequency Modulated Output Output pulse generation with PWCNT	data=1, hr_data=64 }; L02 CNT { reg=A, irq=OFF, L03 MCMP { en_pin_action=0 data=2, hr_data=4 }; L04 CNT { reg=A, irq=OFF,	, max = 2 } N, order =REG_GE_[, max = 3 } N, order =REG_GE_[, max = 4 } N, order =REG_GE_[DATA, pin=0, action=PULSELO, reg=A, DATA, pin=1, action=PULSELO, reg=A, DATA, pin=2, action=PULSELO, reg=A,
			Insert Cancel

Figure 47. Three Channel PWM Function From the Algorithm Library

- (d) Click Insert.
- (e) Close the Algorithm Library window.

(f) You should see that the three channel PWM algorithm has been inserted into your HET assembler program (see Figure 48).



- (g) Close the HET Algorithm Library wizard.
- (h) Save the HET file.
- 5. Assemble the program.
- 6. Load the HET program.
- 7. Run the program for 50 loops.
- 8. View the Waveform and use the Magnifer + F to zoom to show the full waveform. Your display should look like Figure 49. There is clearly activity on the HET output pins (watch_out[31:0]), but it is difficult to see if the activity is on pins 0,1, and 2 as expected.

🔮 WaveFormer Pro
File Import/Export Edit Bus ParameterLibs View Options Window Help
het_signals.btim *
Add Signal Add Bus Delay Setup Sample High Low TRI VAL INVa WH WLO HEX Q+ QF Add Clock Add Spacer Hold Text Marker
1.595us 0.000ps 0us 1us 2us 3us 4us 5us 6us 7us 8us 9us 1us 1us 1us 1us 1us 14us 14us 15us 16u
0 drive[31:0]
1 het.het_clock
2 Dethigh resolution_clock
3 🖝 et.loop_resolution_clock
4 - het loop_count corder (or for for for for for for for for for f
5 🕨 het.ir_a[31:0] 🚾 (oc.)oc. oc. (o. (o.). (o.). (oc. (oc. (oc.) oc. (oc.)
6 het.ir_b[31:0]
7 🕨 het.ir_t[31:0]
8 🕨 het.NHETADDR 🗾 60 60 60 60 60 60 60 60 60 60 60 60 60
9 het.flag_z
10 het flag_x
11 🗨 watch_in[31:0] (00000000_0c/)) (oc_looo) (oc_l) (oc_looo) (oc_l) (oc_looo) (oc_l) (oc_looo) (oc_l) (oc_looo) (oc_loc)
12 • watch_out[31:0] 0000000 (c;)()(c; (c))(c; (c))(c; (c))(c; (c))(c) (c) (c) (c) (c) (c) (c) (c) (c)





Tutorials

9. Expand the 'watch_out[31:0]' bus by right clicking on the signal name and selecting 'Show Bus Member Signals' (see Figure 50).

0 w	See WaveFormer Pro						
		ort/Export Edit Bus Parame	eterLibs View Options Window Help				
] b u	🛛 💽 📝 🔄 🚰 📲 👯 🖛 🖛 👫 🏗 😻 🗍 🕮 🖬						
Eh	🚍 het_signals.btim *						
Add Signal Add Bus Delay Setup Sample Add Clock Add Spacer Hold Text Marker							
	50.9	10ns -1.544us	Dus 1us				
0	-	drive[31:0]					
1	•	het.het_clock					
2	•	et.high_resolution_clock					
3	•	et.loop_resolution_clock					
4	-	het.loop_count	oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)				
5	-	het.ir_a[31:0]	OC.,OC. OI. OI. OI. OI. OI. OI. OI. OI. OI. OI				
6	-	het.ir_b[31:0]					
7	-	het.ir_t[31:0]					
8	•	het.NHETADDR					
9	•	het.flag_z					
10	•	het.flag_x					
11	-	watch_in[31:0]	<u>, 20(, 20), 20), 20, 20, 20, 20, 20, 20, 20, 20, 10, 20, 000000</u>				
12	-		VVVVVVV / / - / - / / - / / - /				
			ected Signal(s)				
			:: Signal(s) <-> Clock(s)				
			Selected Signal(s)				
			ected Signal(s)				
		Show O	nly These Signal(s)				
			rameters on Selected Signal(s)				
			arameters on Selected Signal(s)				
			t: Group Bus <-> Virtual Bus				
			s Member Signals				
			us Member Signals				
		Fut wa					

Figure 50. Expanding the watch_out[31:0] Bus to View Individual Signals



10. You should now see the individual signals for HET ports 0, 1, and 2 (see Figure 51).

💇 Wa	🙅 WaveFormer Pro							
File	File Import/Export Edit Bus ParameterLibs View Options Window Help							
של∐	🛛 📭 🌠 📝 🖉 🔄 🔐 🖓 🖓 🖓 🖓 🖓 🖓 🖓 🖓 🖓 🕅							
⊟ h	et_s	gnals.btim *						
		Add Bus Delay Setur						
		Add Spacer Hold Text	t Marker					
	1.05	2us - <mark>542.9ns</mark>	0us 1us 2us 3us 4us 5us 6us					
0	-	drive[31:0]						
1	-	het.het_clock						
2	•	et.high_resolution_clock						
3	•	et.loop_resolution_clock						
4	-	het.loop_count	(oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)					
5	•	het.ir_a[31:0]	OC.)OC. O(. O. O(. O(. O(. O(. O(. O). O).))), O(. O(. O(. O). O).					
6	•	het.ir_b[31:0]						
7	•	het.ir_t[31:0]						
8	•	het.NHETADDR						
9	•	het.flag_z						
10	•	het.flag_x						
11	-	watch_in[31:0]	<u>0000000</u>					
12	-	hetm.watch_0.out						
13	-	hetm.watch_1.out						
14	-	hetm.watch 2.out						
15	-	hetm watch 3 out						

Figure 51. Expanded Waveform Shows Individual Signals for Pins 0,1,2

11. Restart the Simulator.



12. Right click on the project. Right click to open the project properties dialog. Select XOR from the left pane, and check 0_1 in Pins to XOR (see Figure 52). Click OK to apply the change and close the project properties dialog.

😕 tut6_alg_lib Settings			<u>? x</u>
Device Clock Stimuli VCD XOR AND SHARE	Pins to XOR	 16_17 18_19 20_21 22_23 24_25 26_27 28_29 30_31 	
1		Ok Ca	incel

Figure 52. XOR Share Settings

- 13. Click OK.
- 14. Assemble the program.
- 15. Load the HET program.
- 16. Run the program for 50 loops.
- 17. Stop the program and switch to the Waveform Viewer to see the results. Your simulation results should match Figure 53. Compare this result to the result shown in Figure 52. Specifically, note how the individual signals from HET pins 0 and 1 from Figure 52 have been combined into a single output on HET pin 0 in Figure 53. This shows how the XOR share function operates. In addition to simply applying the 'XOR' logical function to two different pins, the XOR share feature of the HET allows the same physical pin to be updated twice during a single loop resolution period: once by operating on HET pin 0 and again by operating on HET pin 1. Without XOR (or AND) sharing, a program is restricted to one transition per pin during each loop resolution period.

TEXAS INSTRUMENTS

www.ti.com

	WaveFormer Pro						
File	File Import/Export Edit Bus ParameterLibs View Options Window Help						
լթ	🗽 🔆 🖉 🔽 🚰 📲 🚓 🖓 🖓 🎇 🌾 🎆 🏭 🎋 📲 🎆 🖉 🖉 🕼 🖨 🖉 📴 🚱						
E. h	🚍 het_signals.btim *						
		al Add Bus Delay Setup K Add Spacer Hold Text	Sample HIGH LOW TRI VAL INVal WLD HEX Q+ Q,F I Marker Marker Marker Marker Marker Q,R				
	16.0	Ous 14.41us	0us 1us 2us 3us 4us 5us 6us 7us 8us 9us 10us				
0	-	drive[31:0]					
1	•	het.het_clock					
2	•	et.high_resolution_clock					
3	•	et.loop_resolution_clock					
4	-	het.loop_count	(ac.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)oc.)o				
5	•	het.ir_a[31:0]	(ac.)ac. a. a. a. a. a. a. a. a				
6	•	het.ir_b[31:0]					
7	•	het.ir_t[31:0]					
8	•	het.NHETADDR					
9	•	het.flag_z					
10	•	het.flag_x					
11	-	watch_in[31:0]	(0000000); ;; ; ; ; ; ; ; ; ; ; ; ; ; ;				
12	-	hetm.watch_0.out					
13	-	hetm.watch_1.out					
14	-	hetm.watch_2.out					
15		hetm.watch_3.out					
16	-	hetm.watch_4.out					
		netrit.water_4.00t					

Figure 53. Simulation Results Showing XOR Sharing of Pins 0,1 Output onto Pin 0

Tutorials



2.7 Tutorial 7 — Input VCDs

This tutorial covers the following features:

- Input stimuli from a VCD file
 - **NOTE:** To use VCD files as stimulus, the HET IDE must launch the Waveviewer Free application, not Waveformer Pro or any other premium license. If you have acquired a Waveformer Pro (or other premium) license, then you can try Tutorial 8 instead. Alternatively, you can open an instance of Synapticad first from outside the HET IDE and open or create a new file so that you occupy the license file and force the HET IDE to launch the Waveviewer Free application instead. If you work in an environment where there are multiple floating licenses, you may want to change the license server path or license file that Synapticad uses temporarily so that the premium license is not found.
- 1. Open the HET GUI.
- 2. Open the Project:
 - (a) Click Project \rightarrow Open Project to open the project.
 - (b) Open the tut2_input_vcd.prj file from <your_copy> /Tutorials/tut2_input_vcd. (Do not use the files in the tut7_input_vcd folder.)
- 3. Selecting the VCD file:
 - (a) Right click on the project.
 - (b) Go to Project properties.
 - (c) Select Stimuli VCD.
 - (d) Open the folder to select the VCD file.
 - (e) Select the file from <HET installation directory>/tutorials/tut7/inputVCD.vcd.

Utt7_input_vcd Settings		<u>?</u> ×
Device Clock Stimuli VCD XOR AND SHARE	Select stimuli VCD File F:/work/Tutorials/tut2/tut2_input_vcd/inputVCD.vcd Clear Selection	
	Ok Cancel	

Figure 54. Selecting an Input VCD File From the Project Properties Dialog

- (f) Click on OK.
- 4. Assemble and Load the program.

- 5. Run the program for 100 loops.
- 6. Switch to the Waveform Viewer Free and inspect the results. Your screen (after zooming) should look like Figure 55.

	🤐 WaveViewer Free					
Fi	File Import/Export Edit Bus ParameterLibs View Options Window Help					
	A	uto R	um 🔤 🐼 🚺 👬 👯	Kar 🗛 Mar Stil 🛠 🛛 📅 🖉 🖨 🎒 🏥 🎒 🚺 Search		
	📑 he	et_si	gnals.btim			
		Signa Clock	Add Bus Delay Setu Add Spacer Hold Tex			
	6	67.8	8us 67.88us	0us 5us 10us 15us 20us 25us 30us		
	0	-	drive[31:0]			
	1	-	het.het_clock			
	2	-	et.high_resolution_clock			
	3	-	et.loop_resolution_clock			
	4	-	het.loop_count			
	5	-	het.ir_a[31:0]			
	6	-	het.ir_b[31:0]			
	7	-	het.ir_t[31:0]			
	8	-	het.NHETADDR			
	9	-	het.flag_z			
	10	-	het.flag_x			
	11	-	watch_in[31:0]	<u>(0000000)/000000.) 0000000 / 0000001 /oc./oc./ 0000000F /oc.</u>		
	12	-	watch_out[31:0]			

Figure 55. Simulation Results - Stimulus From VCD File Appears in 'watch_in[31:0]' Trace

7. Open the *inputVCD.vcd* file directly in Waveviewer Free and compare the stimulus with what you saw during your simulation. They match, however, note that in the VCD file, the signals that are being driven are shown individually rather than as a bus. Figure 56 shows the VCD file as viewed in Synapticad.

See WaveViewer Free					
File Import/Export Edit Bus Para	neterLibs View Options Window Help				
📗 Auto Run 🖿 😿 🛛 🔐 🗄	[Ka 4a ▶a 35 ‱ ∰ \$2 \$2]] 119]] 😂 🖬 🕼 🖨]] Search				
inputVCD.vcd *					
12.14us 12.14us	0us 5us 10us 15us 20us 25us 30us 35us 40us 45us 50us				
0 🛥 hetm.drive_0.ir					
1 🛥 hetm.drive_1.ir					
2 🛥 hetm.drive_2.ir					
3 🛥 hetm.drive_3.ir					
A A hater watch 0 is					

Figure 56. VCD File Used for Stimulus by Tutorial 7



Tutorials

2.8 Tutorial 8 — Input Stimuli From WaveFormer Pro

This tutorial covers the following features:

- Input stimuli VCD in synaptiCAD.with license
- Generating input in WaveFormer Pro

NOTE: For this tutorial, you must be able to access a Waveformer Pro license through the HET IDE. If you only have access to Waveviewer Free, complete Tutorial 7 instead.

- 1. Open the HET GUI.
- 2. Open the Project:
 - (a) Click Project \rightarrow Open Project to open the project.
 - (b) Open the *tut2_waveform_editor.prj* file from <your_copy> /Tutorials/tut2_waveform_editor. (*Do not use the files in the tut8_input_waveform_syncad_license folder.*)
- 3. Assemble and Load the project.
 - (a) Open Waveformer Pro. Make sure that Waveformer Pro is launched (see NOTE above.)
- 4. Edit waveforms in the SynaptiCAD WaveFormer:
 - (a) Zoom Out to see 100 µs.
 - (b) Select the 'drive[31:0]' signal.
 - (c) Create a pulses on all pins:
 - (i) Click on ~20 μ s inside the confines of the drive[31:0] row.
 - (ii) Click again at 40-50 µs.
 - (iii) Keep clicking at distance of 10 μs. Note that the Waveformer Pro tool automatically alternates the level between high and low as you draw. When complete, your stimulus should look like Figure 57.

WaveFormer Pro			
File Import/Export Edit Bus ParameterLibs View Options Window Help			
Verilog 📝 📗 🗤 HIDE	a 4#)# 雅 疑] 🎬] 😂 🖬 🗿 🔝 🥌] Search		
🚍 het_signals.btim *	x		
Add Signal Add Bus Delay Setup Sample Hig Add Clock Add Spacer Hold Text Marker			
3.932us -105.8us	10us 10us 20us 30us 40us 150us 160us 70us 80us 90us 100us 110us 120us 13		
0 🛥 drive[31:0]	FFFFFFF 1 00000000 1 FFFFFFF 1 00000000 1 FFFFFFF 1 00000000		
1 het.het_clock			
2 het.high_resolution_clock			
3 het.loop_resolution_clock			
4 het.ir_a[31:0]			
5 • het.ir_b[31:0]			
6 het.ir_t[31:0]			
7 het.NHETADDR			
8 🕨 het.flag_z			
9 het.flag_x			
10 - watch_in[31:0]			
11 - watch_out[31:0]			

Figure 57. Input Stimulus Drawn Directly in Synapticad Waveformer Pro

- (iv) Save the file
- 5. Run the simulation for 100 loops.



6. The resulting waveform is shown in Figure 58. Note how the stimulus from the drive[31:0] bus appears on the watch_in[31:0] bus during the simulation run. Also note how the initial state of the drive[31:0] bus is dropped from the simulation. To work around this issue, you can create an additional dummy set of transitions in the drive waveform before the HET program begins executing, as shown in Figure 59. The extra transitions are drawn on a scale of 1 µs so that they are visible in the screen capture; in practice the dummy transitions can be drawn at the 1 ns scale. Figure 60 shows a simulation where the dummy transactions are within 2 ns of time zero so that they do not affect the simulation results because they are complete before the first HET clock.

0	VaveF	ormer Pro			
File	Import/Export Edit Bus ParameterLibs View Options Window Help				
Newlog If With Kar Aar Aar Aar Aar Aar Aar Aar Aar Aar A					
	🛱 het_signals.btim				
		I Add Bus Delay Setup Sample High Add Spacer Hold Text Marker			
	5	.284us -104.4us	0us 10us 20us 30us 40us 50us 60us 70us 80us 90us 10us 110us 110us 120us		
0	-	drive[31:0]	FFFFFFF 1 00000000 X FFFFFFF 1 00000000 XFFFFFFF 1 00000000 XFFFFFFF 1 00000000 XFFFFFFF		
1	-	het.het_clock			
2	-	het.high_resolution_clock			
3	-	het.loop_resolution_clock			
4	-	het.ir_a[31:0]			
5	-	het.ir_b[31:0]			
6	-	het.ir_t[31:0]			
7	-	het.NHETADDR			
8	-	het.flag_z			
9	-	het.flag_x			
10	-	watch_in[31:0]	00000000 X FFFFFFF X 00000000 X FFFFFFF X 00000000		
11	-	watch_out[31:0]	XXXXXXXXX		
			1		



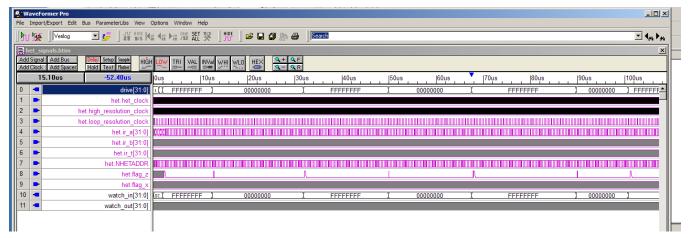


Figure 59. Additional Dummy Stimulus Added to Work Rround Time Zero Issue



Tutorials

Ø. v	Vave	Former Pro			
	File Import/Export Edit Bus ParameterLibs View Options Window Help				
<u></u>]₽] 🗽 🔆 Verilog 🔄 🌽 ff ∰ K# 4# № 號 💱 Iff 🛎 🖬 🧔 🏨 🖨 Search				
	≓ het_signals.btim *				
		nal Add Bus Delay Setup Sample High ck Add Spacer Hold Text Marker			
		27.09us 0.000ps	0us 10us 20us 30us 40us 50us 60us 70us 80us 90us 100us 110us 120us 130us		
0	-	drive[31:0]	FFFFFFFX 00000000 X FFFFFFFFX 00000000 X FFFFFFFF		
1		het.het_clock			
2		het.high_resolution_clock			
3		het.loop_resolution_clock			
4		het.ir_a[31:0]			
5		het.ir_b[31:0]			
6		het.ir_t[31:0]			
7		het.NHETADDR			
8		het.flag_z			
9		het.flag_x			
10	-	watch_in[31:0]	FFFFFFFX 00000000 X FFFFFFF X 00000000 X FFFFFFFF		
11	-	watch_out[31:0]			

Figure 60. Moving the Dummy Stimulus to the 1 ns Scale Provides Perfect Tracking for Practical Purposes



Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Ch	Changes from B Revision (October 2015) to C Revision		
•	Updates were made in Section 2.4.	29	

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have *not* been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products Audio Amplifiers Data Converters DLP® Products	www.ti.com/audio amplifier.ti.com dataconverter.ti.com www.dlp.com	Applications Automotive and Transportation Communications and Telecom Computers and Peripherals Consumer Electronics	www.ti.com/automotive www.ti.com/communications www.ti.com/computers www.ti.com/consumer-apps	
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy	
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial	
Interface	interface.ti.com	Medical	www.ti.com/medical	
Logic	logic.ti.com	Security	www.ti.com/security	
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense	
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video	
RFID	www.ti-rfid.com			
OMAP Applications Processors	www.ti.com/omap	TI E2E Community	e2e.ti.com	
Wireless Connectivity	www.ti.com/wirelessconnectivity			

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2016, Texas Instruments Incorporated