DG

SHA256-instruction-intel-en.doc

SHA256 IP Demo Instruction

1	1 Environment Setup	
2	2 PC Setup	2
3	3 FPGA board setup	4
4	4 SignalTap setup	6
	4.1 SignalTap operations	
	4.2 SignalTap trigger condition	
	4.2.1 To see set length signals timing	
	4.2.2 To see data transfer signals timing	9
	4.2.3 To see hash result signals timing	
5	5 SHA256IP demo software	
	5.1 Demo software interface description	11
	5.2 Hash function by input text	
	5.3 Speed test with 64-bit counter pattern data	
6	8 Revision History	



SHA256 IP Demo Instruction

Rev1.00 2-Jun-2023

This document describes the instruction to demonstrate the operation of SHA256IP on Arria10SoC development board. This demonstration uses SHA256IP demo software to communicate with development board via 1-Gb Ethernet for set length of data, send input text data, and read hash result. User is also able to use SignalTap to see the operation of provided signals in FPGA.

1 Environment Setup

To operate SHA256IP demo, please prepare following test environment.

- 1) FPGA development boards (Arria10SoC development board)
- 2) Test PC with 1-Gb Ethernet connection.
- 3) Micro USB cable for JTAG connection connecting between FPGA development board and PC
- 4) Ethernet cable (Cat5e or Cat6) for network connection between FPGA development board and PC
- 5) Quartus Prime for programming FPGA, installed on Test PC
- 6) File "SHA256IPDemoPack.zip" that included Test Application named "SHA256IP Demo", configuration file named "SHA256IPTest_time_limited.sof" and SignalTap file named "stp1.stp".

(To download this file, please visit our web site at www.design-gateway.com)

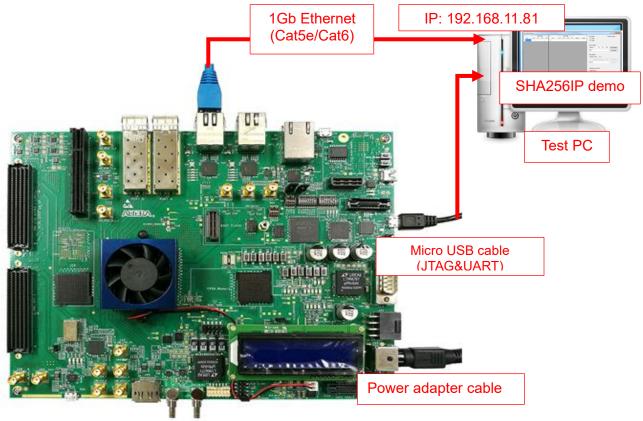


Figure 1-1 SHA256IP demo (FPGA<->PC) on Arria10SoC board



2 PC Setup

Before running demo, please check the network setting on PC. Ethernet setting is shown as follows.

💺 Network and Sharing Center		
🗧 🔶 👻 🛧 🔽 > Control Pa	nel > Network and Internet > Network a	and Sharing Center Search C
Control Panel Home	View your basic network infor	rmation and set up connections
Change adapter settings	View your active networks	
Change advanced sharing settings	Network Public network	Access type: Internet 2 Connections: <u>Ethernet</u>
	Change your networking settings	
	Figure 2-1 IPv4 settin	ng

- 1) Open Ethernet setting option from Control Panel -> Network and Internet -> Network and Sharing Center.
- 2) Click Ethernet icon which is used to connect with FPGA board.

Ethernet Status		× Ethernet Properties
Connection IPv4 Connectivity: IPv6 Connectivity: Media State: Duration: Speed:	No Internet access No network access Enabled 01:56:55 1.0 Gbps	Networking Connect using: 1-Gb LAN connection Intel(R) Ethemet Connection (7) I219-V Configure This connection uses the following items: Image: Client for Microsoft Networks Image: Client for Microsoft Networks Image: Client for Microsoft Networks
Details ActivitySent	· Received	
Bytes: 39,299,55	4 503,073,087	Description Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
	Close	OK Cance

Figure 2-2 Select IP address setting menu

- 3) Click Properties button in Ethernet Status window.
- 4) Select "TCP/IPv4".
- 5) Click Properties button in Ethernet Properties window.



 \times

SHA256-instruction-intel-en.doc

Internet Protocol Version 4 (TCP/IPv4) Properties

General	
You can get IP settings assigned autor this capability. Otherwise, you need to for the appropriate IP settings.	
Obtain an IP address automatical	ly
• Use the following IP address:	6a
IP address:	192.168.11.81
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	· · ·
Obtain DNS server address auton	natically
• Use the following DNS server add	resses:
Preferred DNS server:	
Alternate DNS server:	
Validate settings upon exit	Advanced
	OK Cancel

Figure 2-3 Set IP address

6) Set IP address = 192.168.11.81 and Subnet mask = 255.255.255.0. After that, click OK button to confirm IP address setting.



3 FPGA board setup

- 1) Make sure power switch is off and connect power supply to FPGA development board.
- 2) Connect USB cable between FPGA board and PC via micro USB
- 3) Connect CAT5e/CAT6 cable between PC and Ethernet connection of FPGA board. User must use the right port when FPGA board has two 1Gb Ethernet ports.

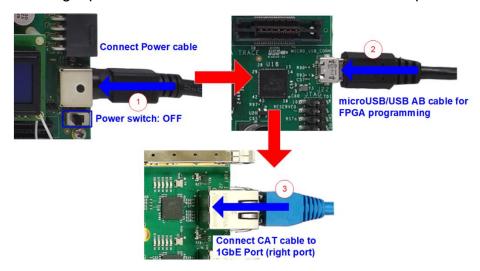


Figure 3-1 Power, Ethernet, and micro USB cable connection

- 4) Power on system.
- 5) Open QuartusII Programmer to program FPGA through USB-1 by following step.
 - a) Click "Hardware Setup..." to select USB-BlasterII.
 - b) Click "Auto Detect" and select FPGA device. (10AS066N3).
 - c) Select FPGA device icon.
 - d) Click "Change File" button, select SOF file in pop-up window, and click "open" button
 - e) Check "program"
 - f) Click "Start" button to program FPGA.
 - g) And wait until Progress status is equal to 100%

Programmer - D:/66.Projects/32.SH	A256/constructio	n/intel/SHA2	256IPTest - SI	HA256IPTes	t - [Cha	in1.cdf]*				_		\times
(a dit View Processing Tools W	/indow Help								g	Search a	altera.com	9
Aardware Setup USB-Blasterii [USI				Mode	JTAG			▼ Pr	ogress:			
Enable real-time ISP to allow backgroun	d programming whe	n available										
File	Device	Checksum	Usercode	Program/	Verify	Blank-	Examine	Security Bit	Erase	ISP CLAMP	IPS File	
output_files/SHA256 e>	10AS066N3F40 SOCVHPS	1D3CD247 00000000	FFFFFFFF <none></none>	Configure		Check						
Auto Detect D e>	5M2210Z 5M2210Z	00000000	<none> <none></none></none>									
X Delete	510122102	0000000	Shonez									
Add File												
Change File.												
Save File		\rightarrow	-	→	-							
Add Device			51122407									
10AS066N3F	40 SOCVH	irə	5M2210Z	5M2.	210Z							

Figure 3-2 FPGA Programmer



6) When configuration is completed, Quartus will show popup message of OpenCore Plus as shown in Figure 3-3 OpenCore Plus Status. Please do not press cancel button, because configuration in FPGA will stop running.

OpenCore Plus Status	×
Click Cancel to stop using OpenCore Plu	is IP.
Time remaining: unlimite	d
Cancel	

Figure 3-3 OpenCore Plus Status

- 7) When configuration is completed, user can check status LEDs on board as Figure 3-4 o LED#1 is always blink to show clock is working.
 - LED#2 is rstB signal. This LED#2 is related to hardware reset switch "S10".
 - LED#3 is "Connection on" status of TOE1G-IP. This LED is on when software open connection to board.
 - LED#4 is "Ready" status of TOE1G-IP. This LED is on when ethernet connection between PC and board is ready.

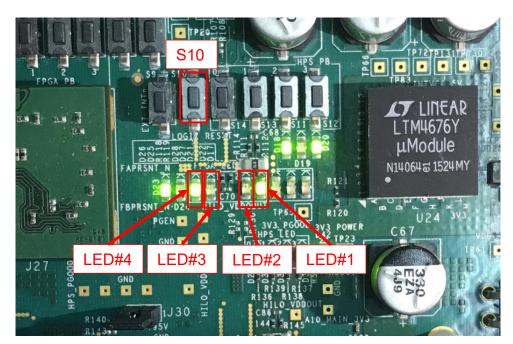


Figure 3-4 LEDs status on board



4 SignalTap setup

This designed block diagram of this demo is shown as in Figure 4-1. SignalTap is prepared to see all control signals between SHA256IP and user logics design.

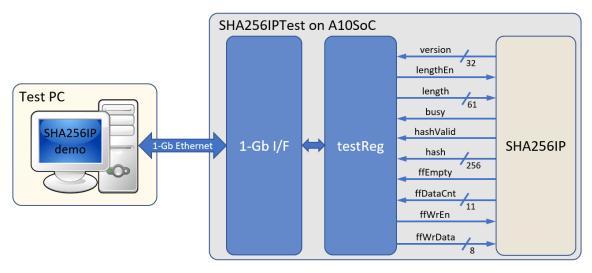


Figure 4-1 Demo environment block diagram

4.1 SignalTap operations

Step to use SignalTap II Logic Analyzer is as follows.

- a) Click File -> Open ..., then select file type to SignalTap II Logic Analyzer Files (*.stp)
- b) Select "stp1.stp", then click Open button as shown in Figure 4-2
- c) As in Figure 4-3, connect FPGA board by select Hardware to USB-BlasterII.
- d) Setup trigger condition to specify signals behavior. Sample of trigger condition and result is shown as in topic 4.2
- e) Click "Run Analysis" button, wait to capture signals from SHA256IP.
- f) The result will be shown, when do SignalTap detect signals same as trigger condition.

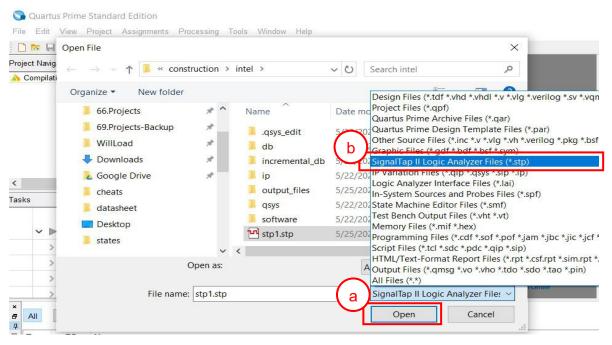


Figure 4-2 Open file "stp1.stp"



	Tap II Logic Analyz View Project Pr				ion/intel/SHA25	66IPTest - SHA2	56IPTest - [stp1	.stp]*								×
		• =	s <u>vv</u> indow	Пер						-			5	earch altera	.com	
	२ ल и 🔭 🕨	£3 😮								_ (c)					
Instar	r 🌂 💫 🛛	Ready	to acquire	1						×	Cha	in Configuratio	n: JTAG rea	dy		×
Instanc	ソー	Status	Enabled	LEs: 4349	Memory: 87552	Small: 0/185200	Medium: 9/2131	Large: 0/0			Hardware:	USB-Blaster	II (USB-11	-	Setup	
🔝 auto_	signaltap_0	Not running	\checkmark	4349 cells	87552 bits	0 blocks	9 blocks	0 blocks								
	~									ſ	Device:	@1: 10AS06	6H1(. ES)/10	AS066 -	Scan C	hain
	()										>> SOF	Manager: 🕌	i. U			
										1						
log: Trig	@ 2020/11/19 12:39:	08 (0:0:5.1 elaps	ed)					c	click to insert tim	ne bar						
Type Alias		Name			-32 -16	Q 16	32 48	64 8	80 96	112	128	144 16	0 176	192	208	224
	testReg:u_testReg	SHA256IP:u_SH	A256IP rstB													
-	testReg:u_testReg															
a	testReg:u_testR								000000000000000000000000000000000000000	008h						
*	testReg:u_testReg			/												-11
*	testReg:u_testReg testReg:u_testR	-														=1
	testReg:u testReg					+										
	testReg:u_testR				00h	ດແຫຼ່ງກາ				001	1					-11
	testReg:u testReg															
a			SHA256IPIf	fDataCnt[100]	000h)) (00	Ch))(000h					
Data Hierarchy D	Setup	d ×	Data Log:	P												×
	SHA256IPTest testReg:u_testReg SHA256IP:u_S	g	auto_sig													<u> </u>
🔝 auto_s	ignaitap_0													0%	00:0	0:00

Figure 4-3 SignalTap II Logic Analyzer



4.2 SignalTap trigger condition

On demo running, user is able to use SignalTap to setup trigger condition and check the signal waveform after trigger is detected. The prepared SignalTap signals are separated to 3 parts as (1) Set length, (2) Data transfer and (3) Hash result, respectively.

4.2.1 <u>To see set length signals timing</u>

Figure 4-4 show trigger condition and Figure 4-5 show sample result from SignalTap when user press hash button in topic 5.2 or 5.3.

trigger: 2	020/11/19 12:34:59 #0	Lock mode: 🧧	CAllow all changes	•
	Node	Data Enable	Trigger Enable	Trigger Conditions
Type Alia:	Name	342	342	1 Basic AND ▼
*	testReg:u_testReg SHA256IP:u_SHA256IP rstB	\checkmark	\checkmark	
*	testReg:u_testReg SHA256IP:u_SHA256IP lengthEn	\checkmark	\checkmark	1
a	testReg:u_testReg SHA256IP:u_SHA256IP length[600]	\checkmark	\checkmark	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*	testReg:u_testReg SHA256IP:u_SHA256IP busy	\checkmark	\checkmark	
	testReg:u_testReg SHA256IP:u_SHA256IP hashValid	\checkmark	\checkmark	
a	testReg:u_testReg SHA256IP:u_SHA256IP hash[2550]	\checkmark	\checkmark	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
	testReg:u_testReg SHA256IP:u_SHA256IP ffEmpty	\checkmark	\checkmark	
a	testReg:u_testReg SHA256IP:u_SHA256IP ffWrData[70]	\checkmark	\checkmark	XXh
*	testReg:u_testReg SHA256IP:u_SHA256IP ffWrEn	\checkmark	\checkmark	
5	testReg:u_testReg SHA256IP:u_SHA256IP ffDataCnt[100]	\checkmark	\checkmark	XXXh

Figure 4-4 Trigger setup for set length signals

📌 SignalTap II Logic An	alyzer - D:/66.Pro	jects/32.SH	A256/construc	tion/intel/SHA2	56IPTest -	SHA256IPT	est - [stp	1.stp]*								_		×
File Edit View Project	Processing Too	ls Window	Help												Se	arch alter	a.com	- 6
層 📒 🤈 C 继 🛤	► 😫 😮																	
Instance Manager: 🍡 👂) 🔳 🛄 Read	ly to acquire									×	JTAG Cha	ain Configu	uration:	JTAG read	y		×
Instance	Status	Enabled	LEs: 4349	Memory: 87552	2 Small: 0/*	18520(Medi	um: 9/213	1 Large: 0/0				Hardware	LISE BI	actorii []	ISB-11	•	Setu	-
🕄 auto_signaltap_0	Not running	\checkmark	4349 cells	87552 bits	0 blocks	9 blo	cks	0 blocks				Taruware	. 030-06	asteni (O	30-1 <u>j</u>	•	Jett	p
												Device:	@1: 10/	AS066H1	1(. ES)/10A	S066 -	Scan	Chain
												>> SO	F Manage	r. 🚢				
		5.04							15 A									
log: Trig @ 2020/11/19 12	2:30:06 (0:0:29.2 ela Name			-6 -4	-2	0 2	4	6	click to	insert time 10	e bar 12	. 14	16	18	20	22	24	26
	Reg SHA256IP:u S		1	-04 .	۲ 4	Y , 4	. 1	, Ч.,	Ч.	ης 1	14	. 17 .	ι ^ρ .	ιp	, 40 ,	44 ,	24	20
	Reg SHA256IP:u_SI																	
	estReg SHA256IP:u			0000000000	000000h	X .				00	00000	000000008	h					_
	Reg SHA256IP:u_SI	HA256IP bus	y y															_
👗 testReg:u_test	Reg SHA256IP:u_S	HA256IP has	hValid															
Image: TestReg:u_t	estReg SHA256IP:u	SHA256IP	nash[2550]			6A09E6	67BB67A	E853C6EF37	2A54FF5	3A510E52	27F9B	05688C1F8	3D9AB5B	E0CD19	h			_
	Reg SHA256IP:u_SI																	_
	estReg SHA256IP:u									00h								
	Reg SHA256IP:u_S																	
E testReg:u_t	estReg SHA256IP:u	_SHA256IPlf	fDataCnt[100]							000h								
				<														>
ጆ Data 🛛 🐺 Setup																		
Hierarchy Display:	×Г	Data Log:	P															×
SHA256IPTest		auto si																
✓ ✓ ➤ testReg:u test	stRea	[<u>11]</u> 2010_31	gnanap_o															
SHA256IP																		
🕄 auto_signaltap_0																		
																0%	00:	00:00
			Figure	4-5 Sar	mple	resul	t for	set le	engt	h sig	gna	als						



4.2.2 To see data transfer signals timing

Figure 4-6 show trigger condition and Figure 4-7 show sample result from SignalTap user press hash button in topic 5.2 or 5.3.

trigger: 20	20/11/19 12:34:59 #0	Lock mode: 🧧	CAllow all changes	•
	Node	Data Enable	Trigger Enable	Trigger Conditions
Type Alias	Name	342	342	1 🔽 Basic AND 🔹
*	testReg:u_testReg SHA256IP:u_SHA256IP rstB	\checkmark	\checkmark	
*	testReg:u_testReg SHA256IP:u_SHA256IP lengthEn	\checkmark	\checkmark	
5	testReg:u_testReg SHA256IP:u_SHA256IP length[600]	\checkmark	\checkmark	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*	testReg:u_testReg SHA256IP:u_SHA256IP busy	\checkmark	\checkmark	
*	testReg:u_testReg SHA256IP:u_SHA256IP hashValid	\checkmark	\checkmark	
\	testReg:u_testReg SHA256IP:u_SHA256IP hash[2550]	\checkmark	\checkmark	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
*	testReg:u_testReg SHA256IP:u_SHA256IP ffEmpty	\checkmark	\checkmark	
\	testReg:u_testReg SHA256IP:u_SHA256IP ffWrData[70]	\checkmark	\checkmark	XXh
*	testReg:u_testReg SHA256IP:u_SHA256IP ffWrEn	\checkmark	\checkmark	1
5	⊡ testReg:u_testReg SHA256IP:u_SHA256IP ffDataCnt[100]	\checkmark	\checkmark	XXXh

Figure 4-6 Trigger setup for data transfer

	ITap II Logic Analy View Project F				tion/ir	ntel/SH	A256II	PTest - S	SHA256I	PTest - [st	p1.stp]*							Se	earch alter	Com	×
· 🗃 📘	うで ※ 80 🕨	K																			
	lanager: 🍡 🔊		dy to acquire											×	JTAG Ch	ain Confic	ouration	: JTAG read	dv		×
Instance		Status		LEs: 4349	Men	nory: 87	552 SI	mall: 0/1	8520(M	edium: 9/21	131 Large	0/0			-		·		-		
	signaltap 0	Not running		4349 cells		2 bits		blocks		olocks	0 bloc				Hardware	: USB-E	Blasterli	[USB-1]	•	Setu	Jb
															Device:	@1: 10	0AS066	iH1(. ES)/10/	AS066 🔻	Scan	Chain
																	=				
															>> \$0	F Manag	er: 🚠				
log: Trig	g @ 2020/11/19 12:3	2:03 (0:0:5.1 elap	psed)									cli	ck to inse	rt time bar							
Type Alia		Nam			ļ 9		2.	4.	6	β.	10	12 .	14	16	18 2	20 2	22 ,	24 2	6 28	3 _ 3	30
	testReg:u_testRe																				
	testReg:u_testRe				_																
a	testReg:u_test											0	00000000	0000008h							
	testReg:u_testRe			1																	
	testReg:u_testRe				_																
a	testReg:u_test	tReg SHA256IP:u	J_SHA256IP I	hash[2550]					6A09	E667BB67	AE853C6E	F372A5	4FF53A5	10E527F9	B05688C1F8	33D9AB58	BE0CD	19h			
-	testReg:u_testRe	g SHA256IP:u_S	HA256IP ffEn	npty																	
a	testReg:u_test	tReg SHA256IP:u	u_SHA256IP f	ffWrData[70]	61	h_X62h	i∕ 63⊦	<u>1)(64h)</u>	<u>31h</u>	(<u>32h</u>)(3	33h) 34h	_X				00h	1				
-	testReg:u_testRe						Ľ														
a		tReg SHA256IP:u	SHA256IP	ffDataCnt[100]			000	h)		004h		_X	008h	_X			00Ch			
					<																>
Data	a 👼 Setup																				
pa Data	a setup																				
Hierarchy	Display:	×	Data Log:	P																	×
	SHA256IPTest testReg:u testR		<table-of-contents> auto_si</table-of-contents>	gnaltap_0																	
	SHA256IP:u																				
🔝 auto	_signaltap_0																				
																			0%	00	:00:00
																			070		

Figure 4-7 Sample result for data transfer signals



4.2.3 To see hash result signals timing

Figure 4-8 show trigger condition and Figure 4-9 show sample result from SignalTap when user press hash button in topic 5.2 or 5.3.

trigge	er: 20	20/11/19 12:34:59 #0	Lock mode: 🧧	Allow all changes	-
		Node	Data Enable	Trigger Enable	Trigger Conditions
Type /	Alias	Name	342	342	1 Basic AND ▼
*		testReg:u_testReg SHA256IP:u_SHA256IP rstB	\checkmark	\checkmark	
*		testReg:u_testReg SHA256IP:u_SHA256IP lengthEn	\checkmark	\checkmark	
5		testReg:u_testReg SHA256IP:u_SHA256IP length[600]	\checkmark	\checkmark	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*		testReg:u_testReg SHA256IP:u_SHA256IP busy	\checkmark	\checkmark	
*		testReg:u_testReg SHA256IP:u_SHA256IP hashValid	\checkmark	\checkmark	1
5		testReg:u_testReg SHA256IP:u_SHA256IP hash[2550]	\checkmark	\checkmark	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
*		testReg:u_testReg SHA256IP:u_SHA256IP ffEmpty	\checkmark	\checkmark	
5		testReg:u_testReg SHA256IP:u_SHA256IP ffWrData[70]	\checkmark	\checkmark	XXh
*		testReg:u_testReg SHA256IP:u_SHA256IP ffWrEn	\checkmark	\checkmark	
5		testReg:u_testReg SHA256IP:u_SHA256IP ffDataCnt[100]	\checkmark	\checkmark	XXXh

Figure 4-8 Trigger setup for input key and searching result

% SignalTap II Logic Analyzer - D:/66.Projects/32.SHA256/constru	tion/intel/SHA256IPTest - SHA256IPTest - [stp1.stp]*	- 🗆 X
File Edit View Project Processing Tools Window Help		Search altera.com
🖷 目 つ (** 👪 🕨 😫 🔒		
Instance Manager: 🍡 👂 🔳 🛅 Ready to acquire		× JTAG Chain Configuration: JTAG ready ×
Instance Status Enabled LEs: 4349 auto_signaltap_0 Not running 4349 cells	Memory: 87552 Small: 0/18520(Medium: 9/2131 Large: 0/0 87552 bits 0 blocks 9 blocks 0 blocks	Hardware: USB-Blasteril [USB-1] Setup Device: @1:10AS066H1(JES)/10AS066 • Scan Chain >> SOF Manager:
log: Trig @ 2020/11/19 12:35:11 (0:0:11.7 elapsed)	click to insert time bar	
Type Alias Name	-1 φ 1	2 3
testReg u_testRegISHA256IP u_SHA256IPInstB testReg u_testRegISHA256IP u_SHA256IPInstB testReg u_testRegISHA256IP u_SHA256IPInstpht60.0] testReg u_testRegISHA256IP u_SHA256IPInsthValid testReg u_testRegISHA256IP u_SHA256IPInsthValid	00000000000000000000000000000000000000	6659EC5450B69AEh X
Data msselup Hierarchy Display: × ✓ > SHA256IPTest ✓ > testReg u_testReg ✓ > SHA256IP.u_SHA256IP		×
auto_signaltap_0		0% 00:00:00

Figure 4-9 Sample result for input key and searching result



5 SHA256IP demo software

SHA256IP demo software is used for show hash function that compute by SHA256IP in A10SoC board via 1-Gb Ethernet connection.

5.1 Demo software interface description

	🖳 SHA256IP Demo	—	\times
a	SHA256IP (timeout): 1.00 Connect		
	Hash function by input text		
\frown	hash		
(b	input text abcd1234		
	length : hash		
	Speed test with 64-bit counter pattern data		
C	Speed Test with 64-bit Counter data size (bytes): 3200000 3200000	0	
Ċ	counter pattern		
	length : hash		

Figure 5-1 Software interface

Figure 5-1 shows SHA256IP demo software and the description is shown as below.

- a) Connect button is used for open connection to A10SoC board via 1Gb-Ethernet.
- b) This section is hash functional test by input text.
- c) This section is speed test function with fixed data pattern by 64-bit counter pattern.



5.2 Hash function by input text

User can input text data and press hash button, then software will transfer input text to SHA256IP and get the hash result back to show in "length : hash" text box. Figure 5-2 shows example of hash function by input text.

🖳 SHA256IP D	emo				_		\times
SHA256IP (timeou	ıt): 1.00						
Disconnect							
Hash function by	input text						
hash							
input text	abcd1234						
length : hash	8 bytes : E9CEE7	1AB932FDE863338D(08BE4DE9DFE39EA049BDA	FB342CE659	EC5450B	69AE	
Speed test with	64-bit counter patte	m data					
Speed Test wit	h 64-bit Counter	data size (bytes):	32000000	32000000			
counter pattern							
length : hash							



5.3 Speed test with 64-bit counter pattern data

User can input data size (bytes unit), and press "Speed test with 64-bit Counter" button. Then software will show pattern of data in counter pattern text box and send command to A10SoC board to start generate test pattern data send to SHA256IP. After hash function is finished, software will get the result hash value and show in "length : hash" text box. Then software will popup message to show time of hash function operation. Figure 5-3 shows example of speed test with 64-bit counter pattern data.

💀 SHA256IP D	emo			-	- 🗆	\times
SHA256IP (timeo	ut): 1.00		1			
Connect						×
Hash function b	y input text			data size (bytes time (seconds):		
input text	abcd1234				1.1	
length : hash	8 bytes : E9CEE7	1AB932FDE863338D	08BE4DE9DFE39EA049BI		OK	
Speed test with	64-bit counter patte	m data	1			
Speed Test wit	th 64-bit Counter	data size (bytes):	32000000	32000000		
counter pattern	hex : 000000000	000000 0100000000	000000 0200000000000000	0 increment until	FF083D000	0000000
length : hash	h 32000000 bytes : 1CA554E6F0817062B6B4765BFF7F52A425811534A636D112157934704156FE15				E15	

Figure 5-3 Example of speed test with 64-bit counter pattern data



6 Revision History

Revision	Date	Description	
1.00	26-Jan-2021	Initial version release	