

## tCAM IP Demo Instruction

1	Environment Setup.....	1
2	PC Setup.....	2
3	FPGA board setup.....	4
4	SignalTap setup.....	6
4.1	SignalTap operations .....	6
4.2	SignalTap trigger condition .....	8
4.2.1	To see Rule Initial Control signals.....	8
4.2.2	To see Rule Memory Interface signals.....	9
4.2.3	To see Input key and Search result signals.....	10
5	tCAMIP demo software .....	11
5.1	Demo software interface description.....	11
5.2	File format for Rule data and Key Data.....	12
5.3	Rule table creation.....	13
5.4	Key data table creation .....	14
5.5	Initialization rule data to FPGA Development board .....	15
5.6	Searching key data .....	16
5.7	Compare.....	18
6	Revision History .....	19

# tCAM IP Demo Instruction

Rev1.02 2-Jun-2023

This document describes the instruction to demonstrate the operation of tCAMIP on Arria10SoC development board. This demonstration uses tCAMIP demo software to communicate with development board via 1 Gigabits Ethernet for preparing rule table, initializing tCAMIP, sending keys and reading result data. User is also able to use SignalTap to see the operation of provided signal in FPGA.

## 1 Environment Setup

To operate tCAMIP demo, please prepare following test environment.

- 1) FPGA development boards (Arria10SoC development board)
- 2) Test PC with 1 Gigabits 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 “tCAMIPDemoPack.zip” that included Test Application named “tCAM-IP Demo.exe”, configuration file named “tCAMIPTest\_time\_limited.sof” and SignalTap file named “stp1.stp”.

(To download this file, please visit our web site at [www.design-gateway.com](http://www.design-gateway.com))

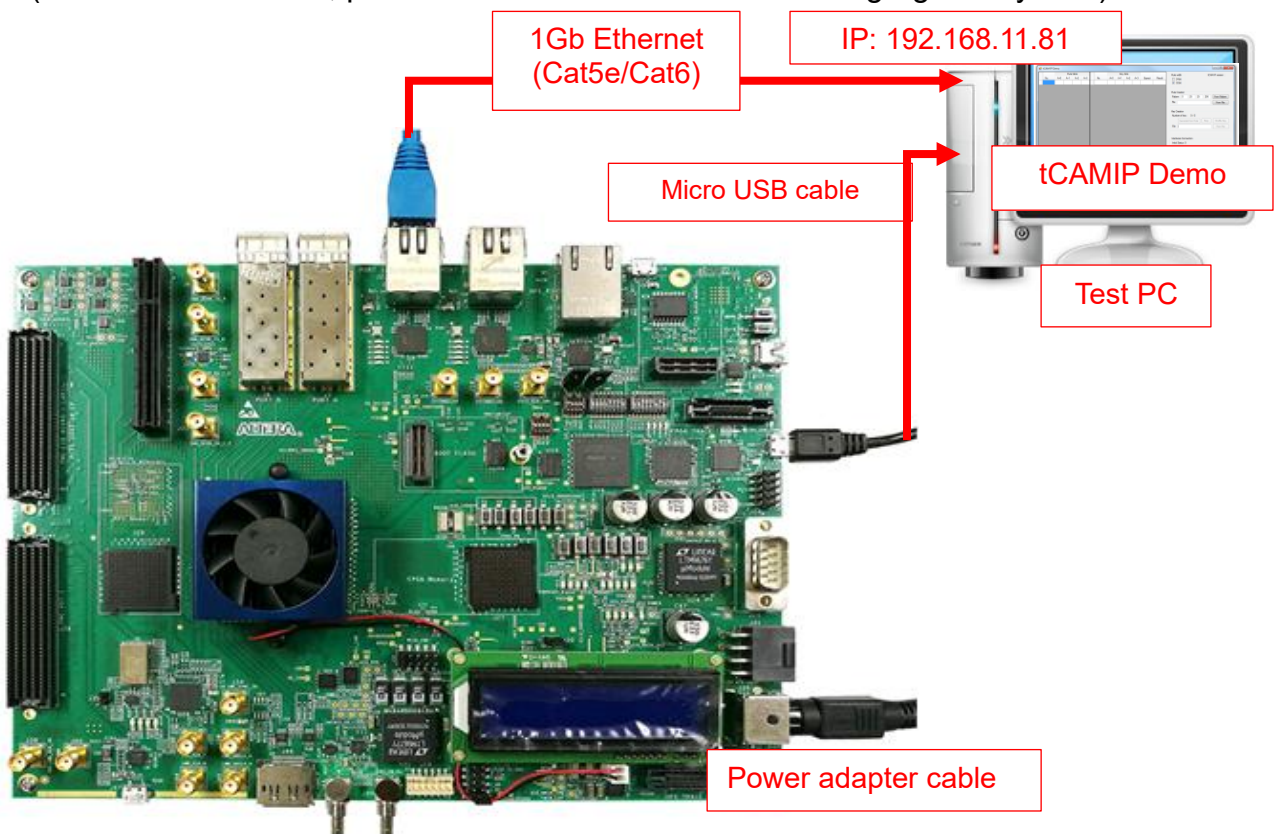


Figure 1-1 tCAMIP demo (FPGA<->PC) on Arria10SoC board

## 2 PC Setup

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

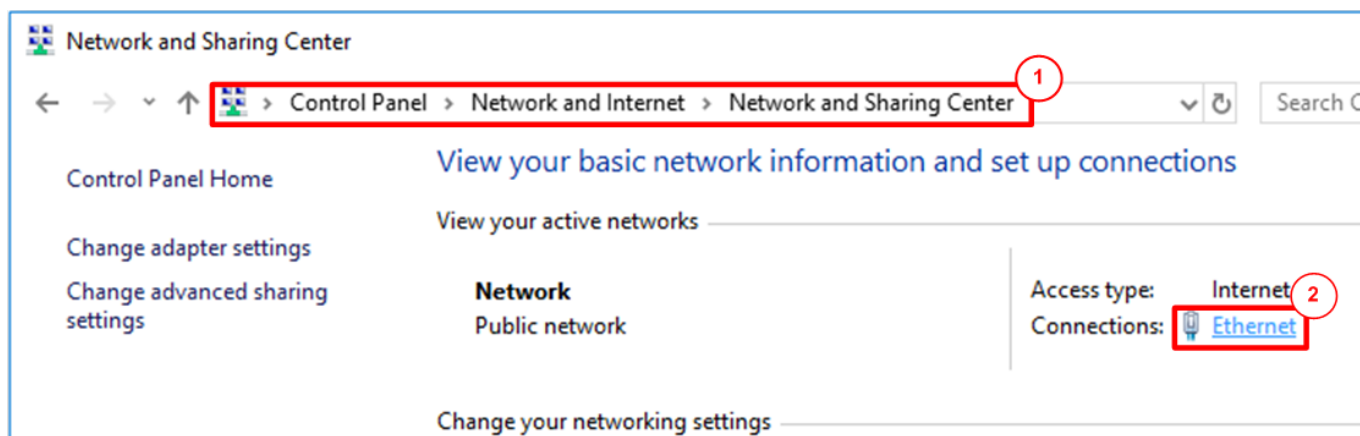


Figure 2-1 IPv4 setting

- 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.

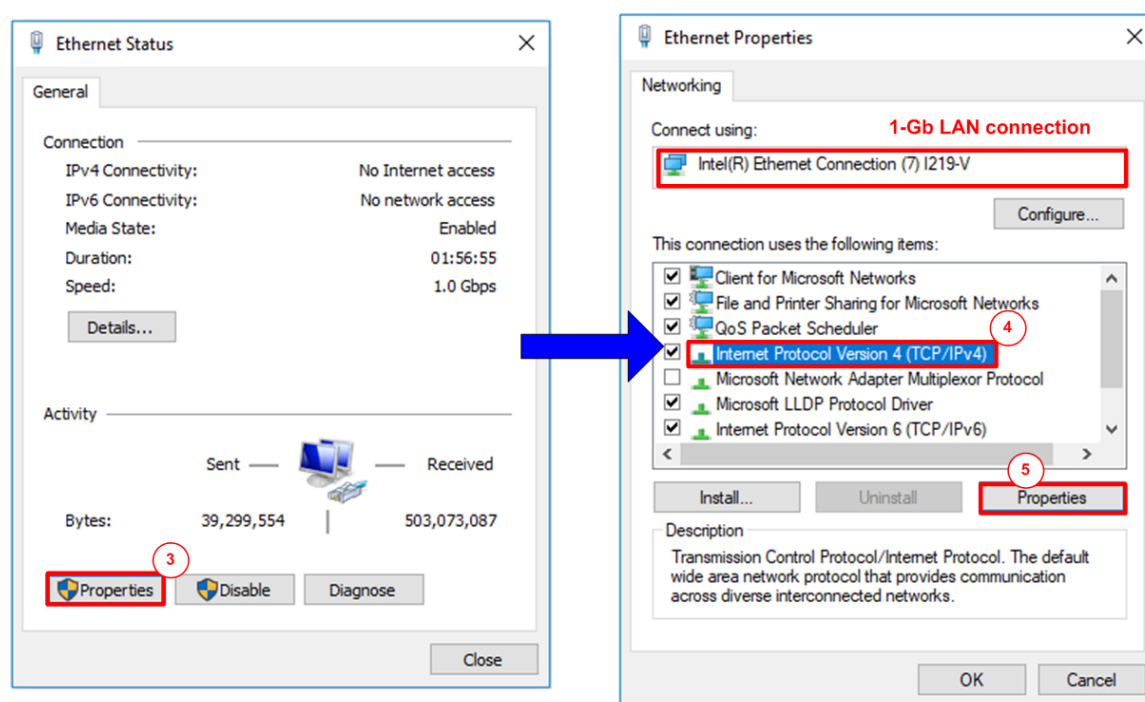


Figure 2-2 Select IP address setting menu

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

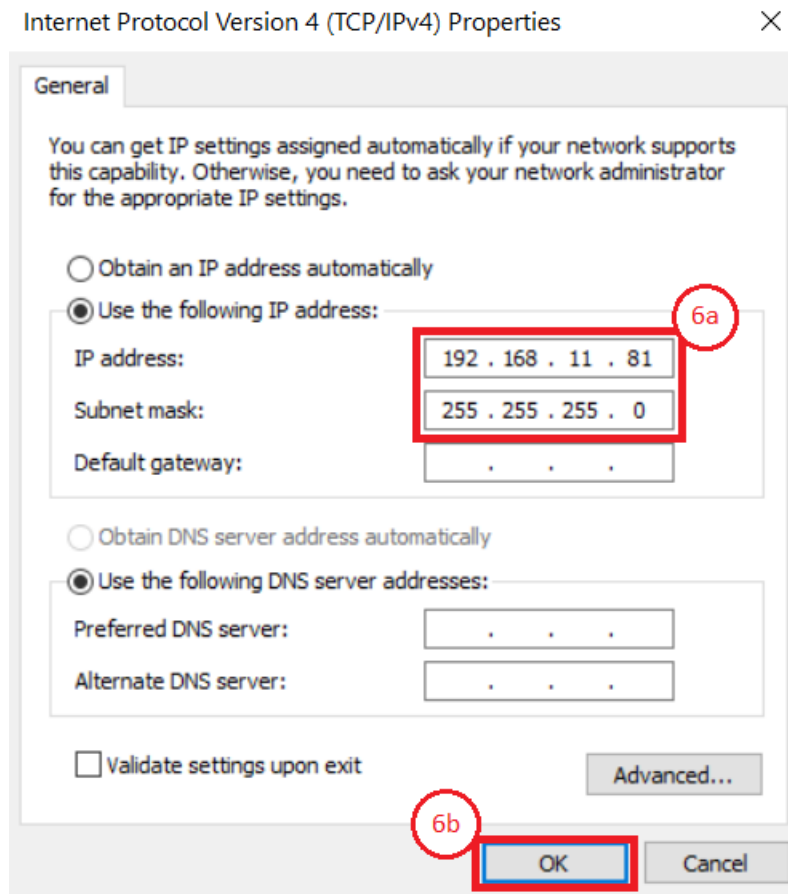


Figure 2-3 Set IP address

- 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 1 Gigabits Ethernet ports.

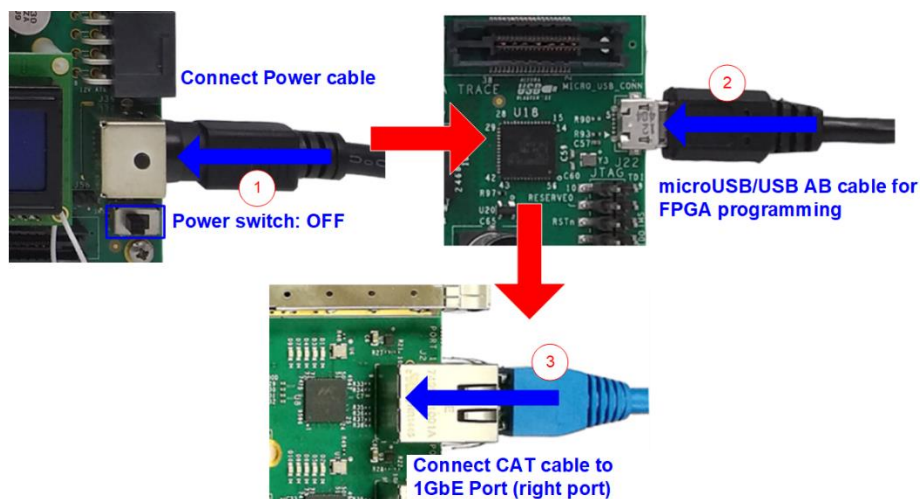


Figure 3-1 Power, Ethernet, and micro USB cable connection

- 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 [USB-1].
  - 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 and wait until Progress status is equal to 100%

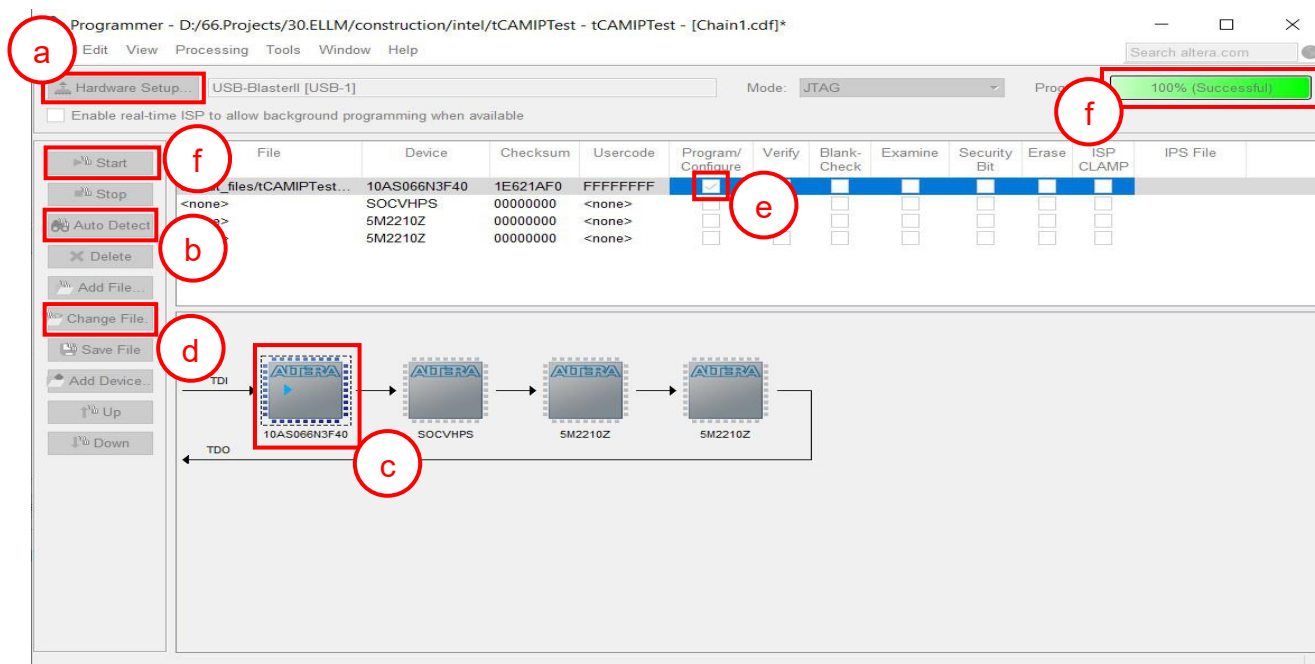


Figure 3-2 FPGA Programmer



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

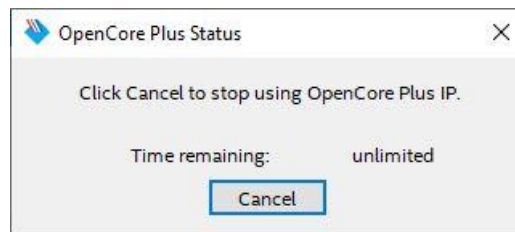


Figure 3-3 OpenCore Plus Status

- 7) When configuration is completed, user can check status LEDs on board as Figure 3-4
- LED0 is “Ready” status of TOE1G-IP. This LED is on when ethernet connection between PC and board is ready.
  - LED1 is “Connection on” status of TOE1G-IP. This LED is on when software open connection to board.
  - LED2 is show reset status and relate to hardware reset switch “S10”.
  - LED3 is always blink to show clock is working.

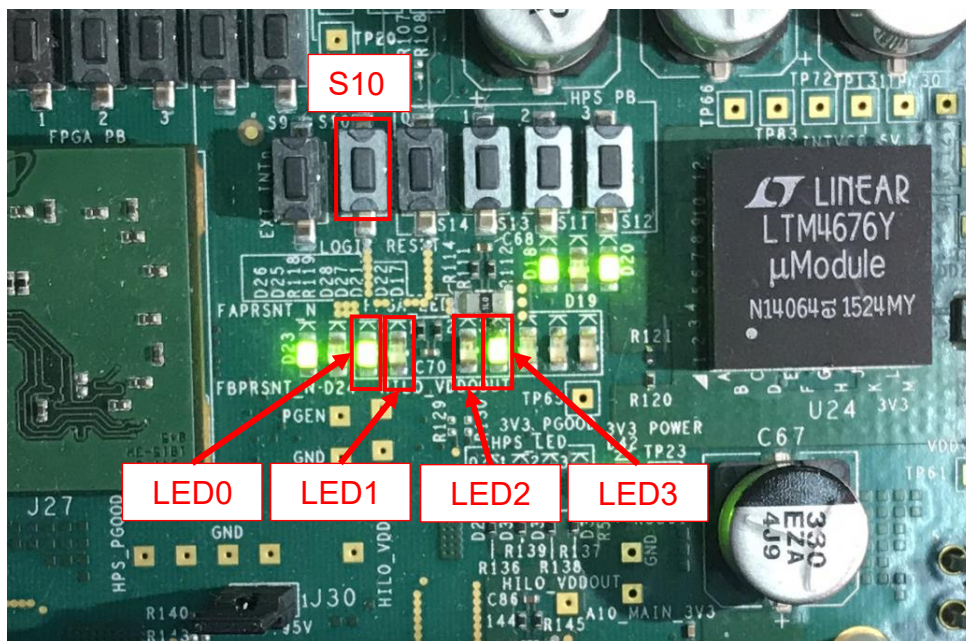


Figure 3-4 LEDs status on board

## 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 tCAMIP and user logics design.

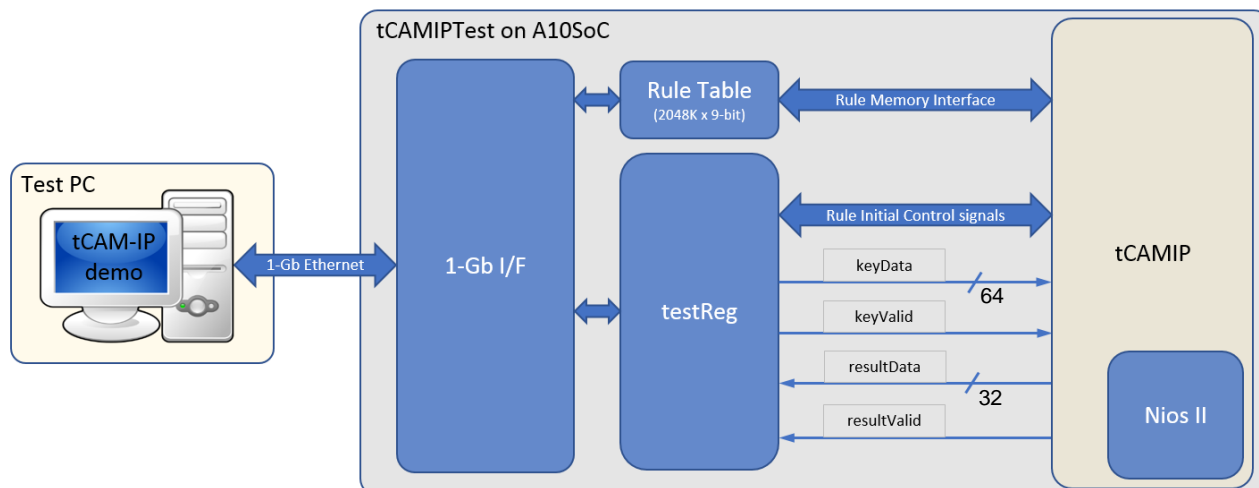


Figure 4-1 Demo environment block diagram

### 4.1 SignalTap operations

Step to use SignalTap II Logic Analyzer is as follows.

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

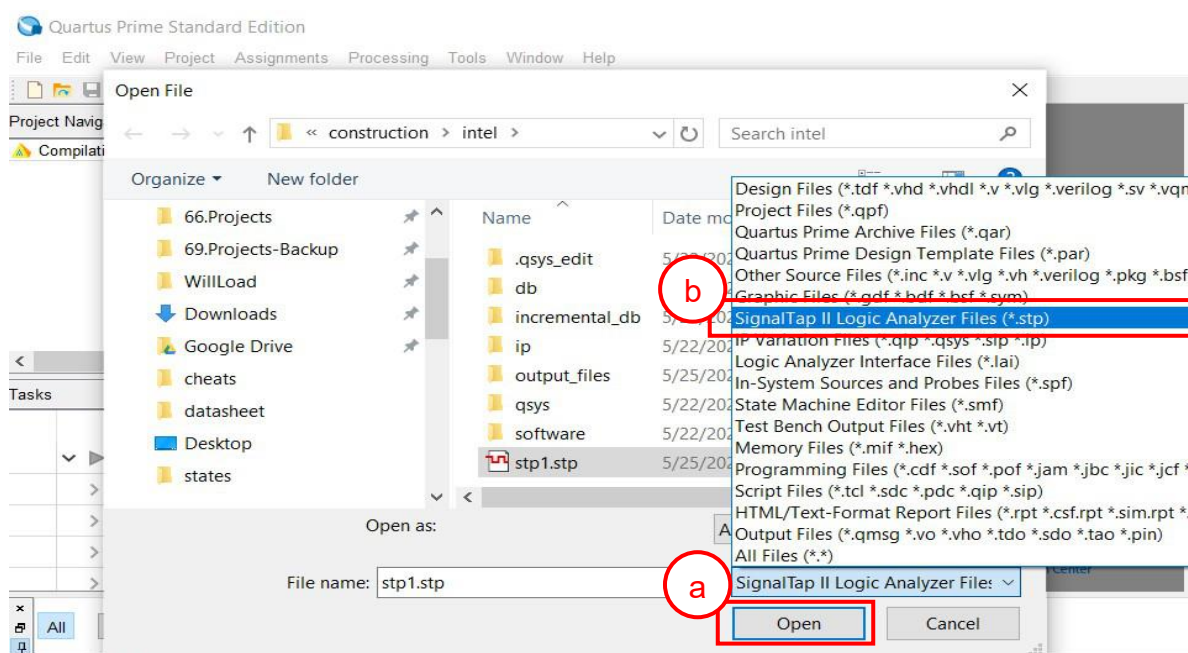


Figure 4-2 Open file "stp1.stp"

The screenshot displays the SignalTap II Logic Analyzer software interface. At the top, the title bar reads "SignalTap II Logic Analyzer - D:/66.Projects/30.ELLM/construction/intel/tCAMIPTest - tCAMIPTest - [stp1.stp]\*". The menu bar includes File, Edit, View, Project, Processing, Tools, Window, and Help. A toolbar contains various icons for file operations and analysis. Below the toolbar, a table lists instances, with "auto\_signaltap\_0" selected. To the right, the "JTAG Chain Configuration" window is open, showing hardware as "USB-BlasterII [USB-1]" and device as "@1: 10AS066H(112)2ES(3)3E2(3)".

The main area shows a data log for a trigger event at 2020/08/18 15:45:48. The log table has columns for Type, Alias, Name, and a time axis from -8 to 8. The data is displayed as a waveform with hexadecimal values. A red box highlights the data log area, and a red circle 'f' is placed near the bottom right of the log.

Type	Alias	Name	Value
		tCAMIP_u_tCAMIP[keyValid]	
		tCAMIP_u_tCAMIP[keyData][63..0]	00000000h 01000000h 02000000h 03000000h 04000000h 05000000h 06000000h FF000000h 00100000h 01010000h 02010000h 03010000h 04010000h 05010000h 06010000h
		tCAMIP_u_tCAMIP[resultValid]	
		tCAMIP_u_tCAMIP[resultData][31..0]	00000001h 00000002h 00000003h 00000004h 00000005h 00000006h 00000007h 00000008h

At the bottom left, the "Hierarchy Display" shows a tree view with "tCAMIPTest" expanded to "tCAMIP\_u\_tCAMIP". The "Data Log" section shows "auto\_signaltap\_0".

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) Rule Initial control signals, (2) Rule Memory interface signals and (3) Input keys & search result signal respectively.

### 4.2.1 To see Rule Initial Control signals

Figure 4-4 show trigger condition and Figure 4-5 show sample result from SignalTap when user do initial rule table in topic 5.5

trigger: 2020/08/18 14:51:37 #1		Lock mode:  Allow all changes			
Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	186	186	1 <input checked="" type="checkbox"/> Basic AND
		tCAMIP:u_tCAMIP ruleWidth[15..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXh
		tCAMIP:u_tCAMIP ruleCount[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleInit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
		tCAMIP:u_tCAMIP ruleBusy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP ruleAddr[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleRdReq	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP ruleRdValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP ruleData[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleStatusCode[3..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Xh
		tCAMIP:u_tCAMIP keyValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP keyData[63..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP resultValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP resultData[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh

Figure 4-4 Trigger setup for rule initial control signals

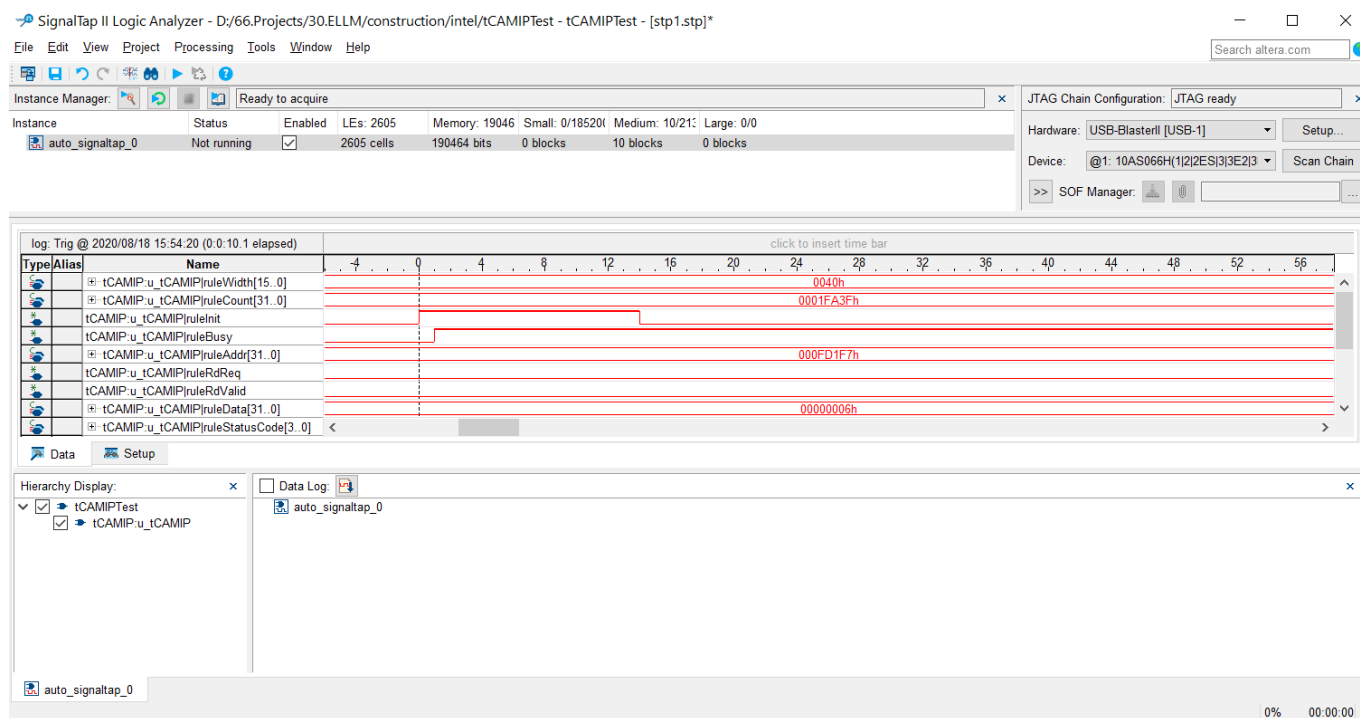


Figure 4-5 Sample result for rule initial control signals

### 4.2.2 To see Rule Memory Interface signals

Figure 4-6 show trigger condition and Figure 4-7 show sample result from SignalTap when user do initial rule table in topic 5.5

trigger: 2020/08/18 15:54:10 #0		Lock mode:  Allow all changes			
Type	Alias	Node Name	Data Enable	Trigger Enable	Trigger Conditions
			186	186	1 <input checked="" type="checkbox"/> Basic AND
		tCAMIP:u_tCAMIP ruleWidth[15..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXh
		tCAMIP:u_tCAMIP ruleCount[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleInit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP ruleBusy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP ruleAddr[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleRdReq	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
		tCAMIP:u_tCAMIP ruleRdValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP ruleData[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleStatusCode[3..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Xh
		tCAMIP:u_tCAMIP keyValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP keyData[63..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP resultValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		tCAMIP:u_tCAMIP resultData[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh

Figure 4-6 Trigger setup for rule memory signals

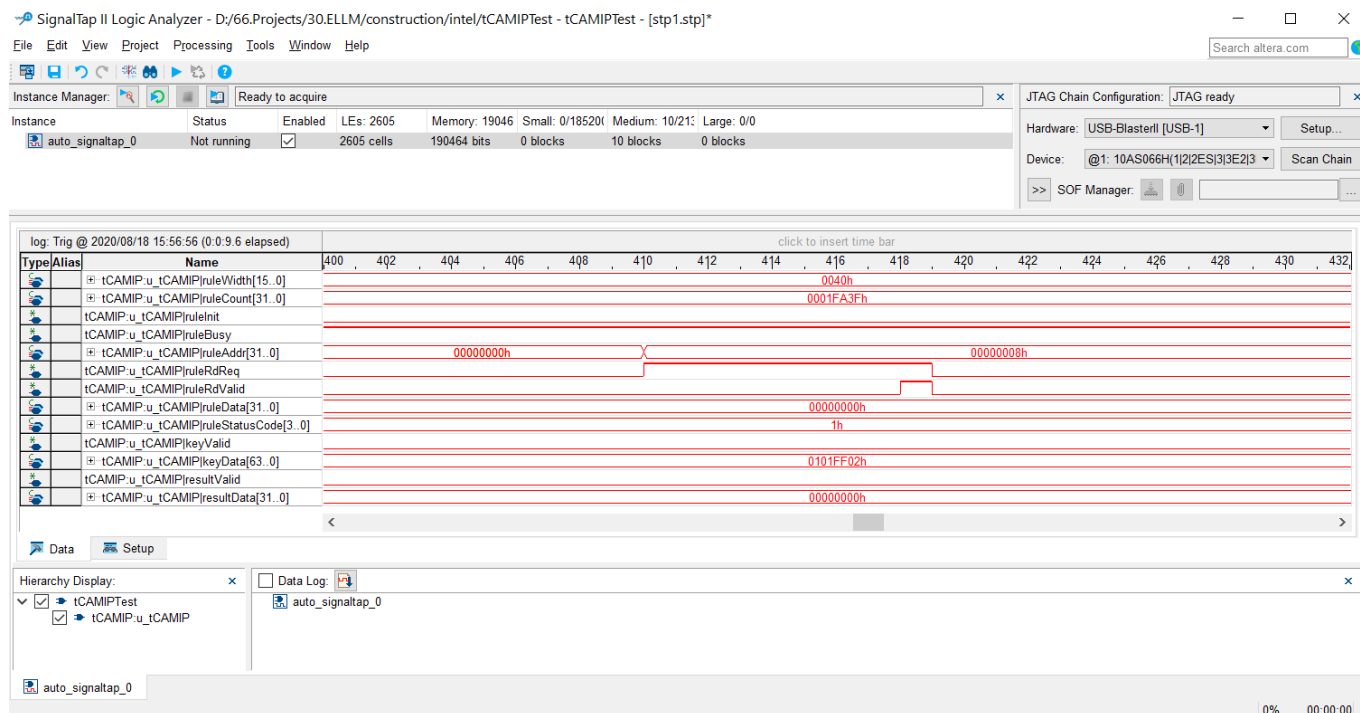


Figure 4-7 Sample result for rule memory signals

### 4.2.3 To see Input key and Search result signals

Figure 4-8 show trigger condition and Figure 4-9 show sample result from SignalTap when user do initial rule table in topic 5.6

trigger: 2020/08/18 15:58:58 #0		Lock mode:	Allow all changes		
Type	Alias	Name	Data Enable	Trigger Enable	Trigger Conditions
		tCAMIP:u_tCAMIP ruleWidth[15..0]	186	186	1 <input checked="" type="checkbox"/> Basic AND
		tCAMIP:u_tCAMIP ruleCount[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXh
		tCAMIP:u_tCAMIP ruleInit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleBusy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		tCAMIP:u_tCAMIP ruleAddr[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleRdReq	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		tCAMIP:u_tCAMIP ruleRdValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		tCAMIP:u_tCAMIP ruleData[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP ruleStatusCode[3..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Xh
		tCAMIP:u_tCAMIP keyValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		tCAMIP:u_tCAMIP keyData[63..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh
		tCAMIP:u_tCAMIP resultValid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		tCAMIP:u_tCAMIP resultData[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXh

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

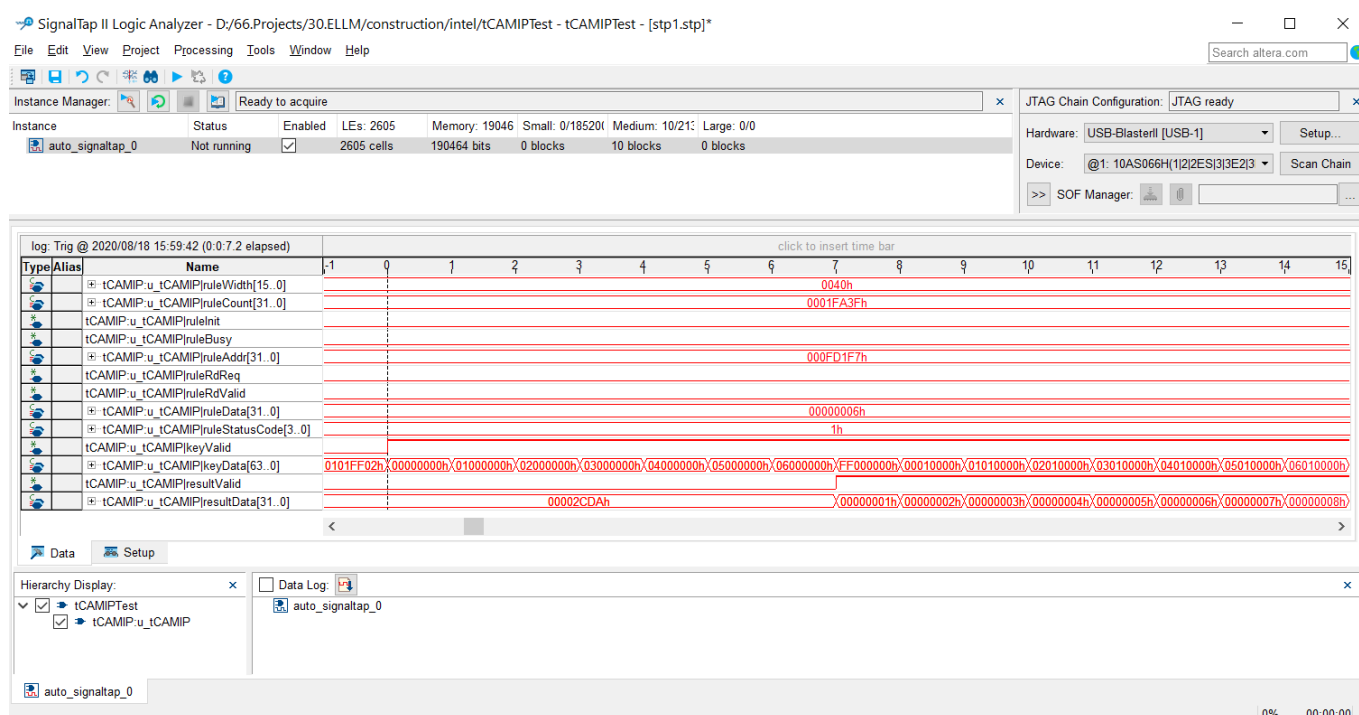


Figure 4-9 Sample result for input key and searching result

## 5 tCAMIP demo software

tCAMIP demo software is used for preparing and sending rule and key pattern data to A10SoC board via 1 Gigabits Ethernet connection and getting the result of searching from tCAMIP.

The main features of tCAMIP demo software are as following.

- 1) Rule creation.
- 2) Key creation.
- 3) Initialization rule data table to FPGA development board.
- 4) Search key data and display result.
- 5) Compare expect data of key data searching to result data of key data searching.

### 5.1 Demo software interface description

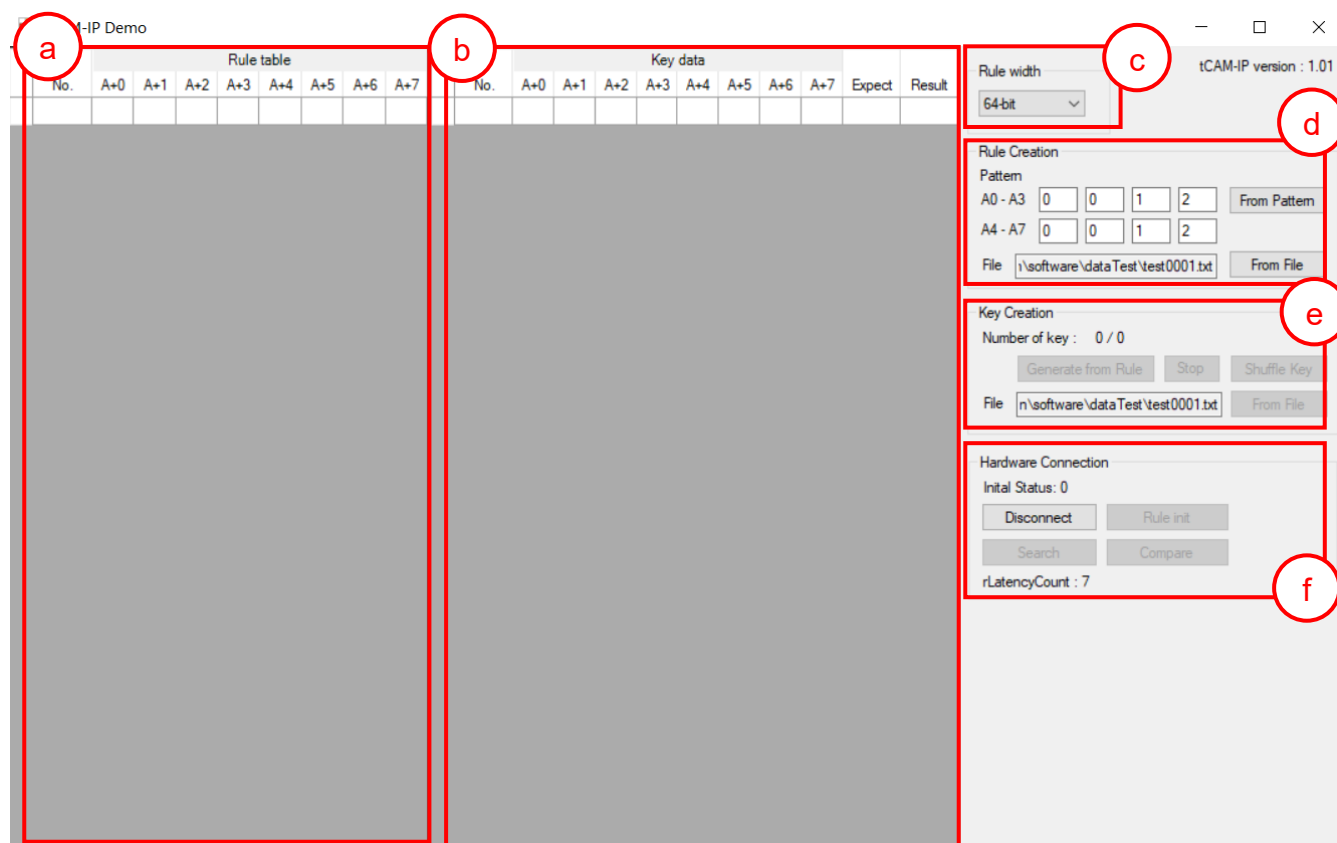


Figure 5-1 Software interface

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

- a) Rule table is the user rule data display.
- b) Key data table is the user key data display.
- c) Rule width, Users can select data size modes.
- d) Rule Creation, Users can generate rule data table pattern or load rule data table pattern from the file.
- e) Key Creation, Users can generate key data from rule table or load key data table pattern from the file.
- f) Hardware Connection, Users can communicate with FPGA development board in this part. Which consists of connection with board, rule Initialize, key data search, result compare.

## 5.2 File format for Rule data and Key Data

User can prepare Rule data or Key data file for this demo. The file format is shown as Figure 5-2 and Figure 5-3 for rule width 32-bit and 64-bit respectively.

- 1) The first line is header of file for specific number of byte data in this file like “A+0, A+1, A+2, A+3” for 32-bit file format.
- 2) Next lines are data. The valid range of data is 0-255, others number and ‘x’ is defined to don’t care value.

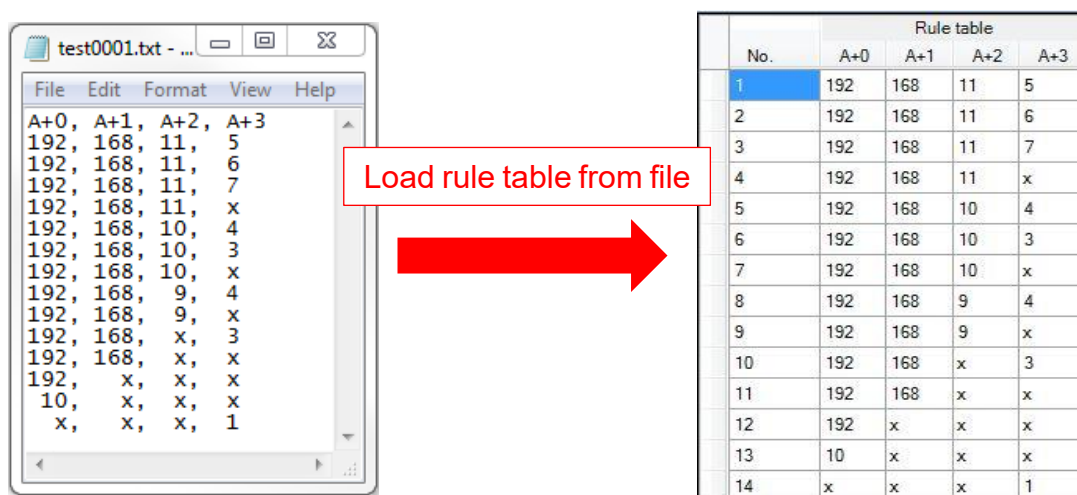


Figure 5-2 Example of load rule file 32-bit

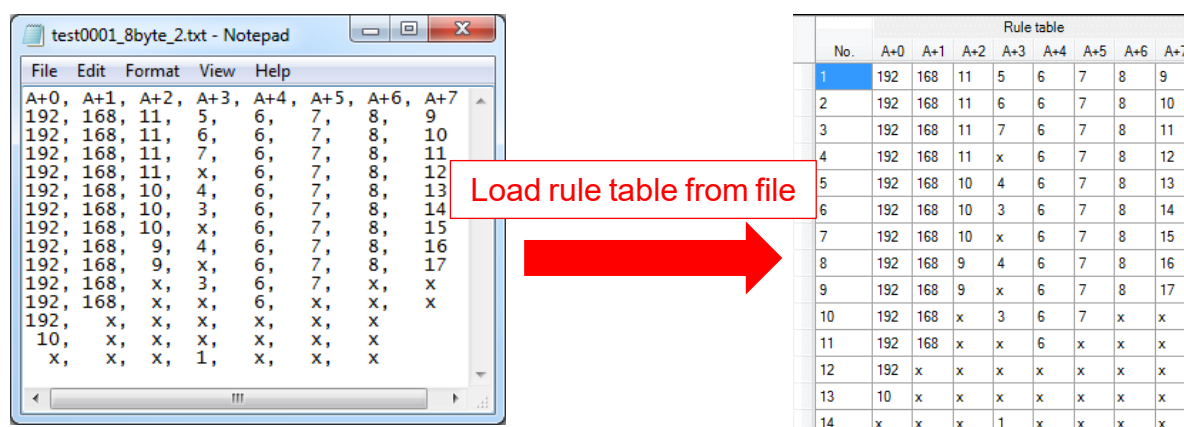
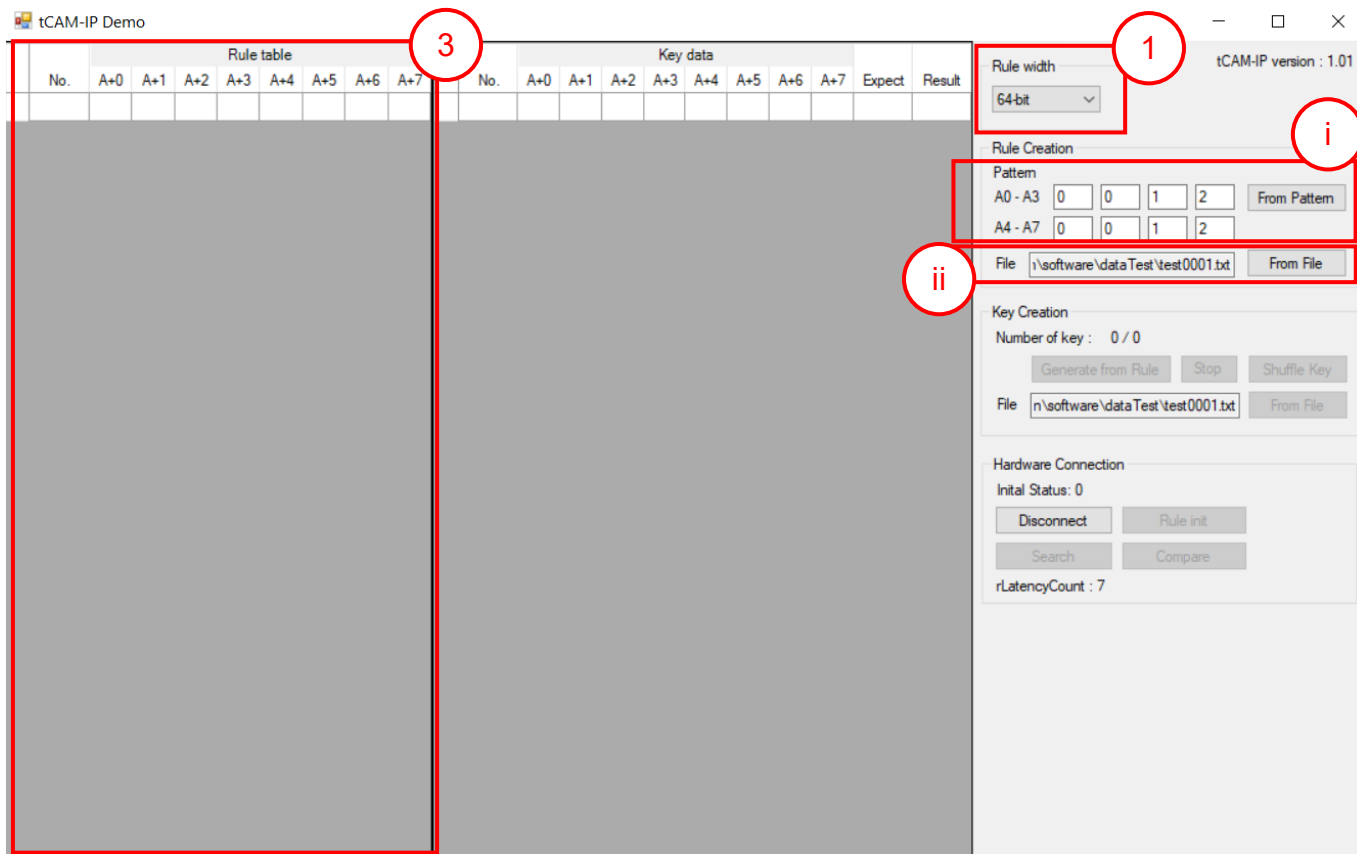


Figure 5-3 Example of load rule file 64-bit



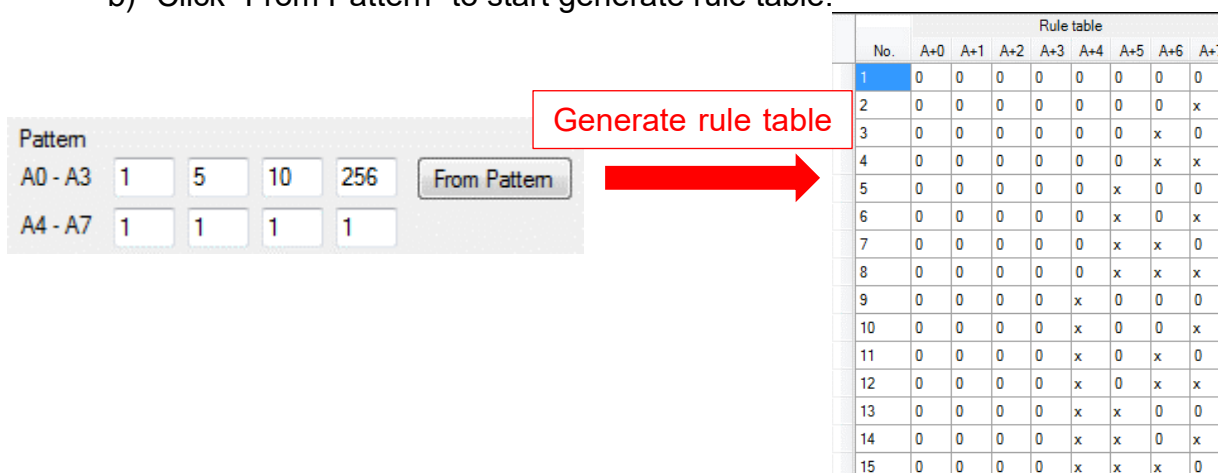
### 5.3 Rule table creation

To create the rule table, the step is shown as in Figure 5-4.



**Figure 5-4 Rule creation process**

- 1) Select Rule width mode for setup data size.
- 2) Create rule table, the user can select 2 modes as follows.
  - i) Generate rule data table from pattern as show in Figure 5-5.
    - a) Fill out valid range (0-256) of each byte in “Pattern”. Byte of pattern position starts “A+0” left to right.
    - b) Click “From Pattern” to start generate rule table.



**Figure 5-5 Example rule data table pattern.**

- ii) Load rule data table from the file by click “From File” button to browse and load rule data file.
- 3) When rule is created, rule data show in rule table as shown in Figure 5-5.

### 5.4 Key data table creation

To create key table, the step is shown as in Figure 5-6.

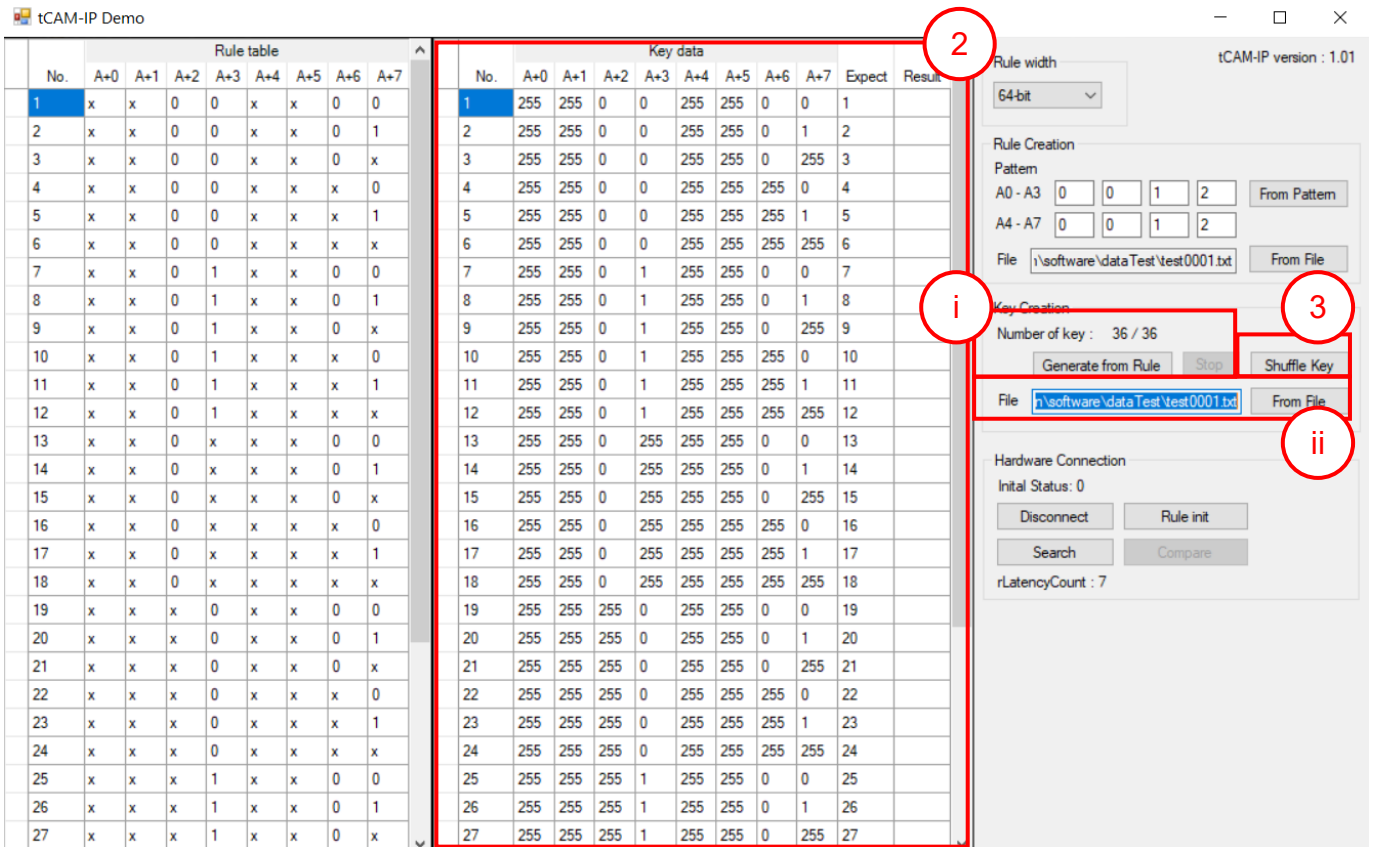


Figure 5-6 Key creation

- 1) Create Key data, the user can select 2 modes as follows
  - i) Generate key data from rule table.
    - a) Click “Generate from Rule” to start generate key data.
    - b) “Number of key: X / Y” is progress status by X is progress number of key and Y is total number of keys from all combination of rules. (generating time depends on the amount of key data)
    - c) “Stop” button is used for stop key generating.
  - ii) Load key data from the file by click “From File” button to browse and load rule data file.
- 2) When key data is created, the expect value will be generated automatically.
- 3) “Shuffle Key” button is used to shuffle each key in key table randomly.

## 5.5 Initialization rule data to FPGA Development board

The step to initialize rule data is following as below and Figure 5-7.

- 1) Please make sure that LED0 (TOE1G ready) is on (please refer LEDs on board in Figure 3-4), before click “Connect” button.
- 2) When connection is completed, the text of the button will change to “Disconnect” and LED1 (open connection status) is turn on. The button “Rule init”, “Search” and “Compare” will enable.
- 3) On SignalTap, setup trigger condition when ruleInit = '1' and other signals are don't care. Then press “Run Analysis” button.
- 4) Click “Rule init” button to initialize data of rule table. Initialization duration time depends on the amount of key data.

“Initial Status: X” is active status for Rule transfer, Rule verify, Rule initial and Status code. By X is running number of rule table, time count of Rule initial status and status code which shows that process is still running)

- When ruleInit signal = '1', SignalTap will show all interface signals of tCAMIP.
  - In case to see the difference trigger point, user can change trigger condition in SignalTap on step 3).
- 5) When initialization is completed, the message box will popup “Rule initial completed”. In case initialization is incomplete, “Initial Status:” is shown error message of status code.

The screenshot shows the tCAM-IP Demo interface. It consists of three main panels:

- Rule table:** A table with columns No., A+0, A+1, A+2, A+3, A+4, A+5, A+6, A+7. It contains 27 rows of binary data.
- Key data:** A table with columns No., A+0, A+1, A+2, A+3, A+4, A+5, A+6, A+7, Expect, Result. It contains 27 rows of binary data.
- Control Panel:** Located on the right, it includes:
  - Rule width: 64-bit
  - Rule Creation section with Pattern (A0-A3, A4-A7) and File input fields.
  - Key Creation section with Number of key (36 / 36) and buttons for Generate from Rule, Stop, and Shuffle Key.
  - Hardware Connection section with Initial Status: 0 and buttons for Disconnect (circled in red with '1'), Rule init (circled in red with '4'), Search, and Compare.
  - LatencyCount: 7

Figure 5-7 Rule initialization

### 5.6 Searching key data

To search key data by the hardware of tCAMIP demo, the step is following as below and Figure 5-8.

- 1) On SignalTap, setup trigger condition when keyValid = '1' and other signals are don't care. Then press "Run Analysis" button.
- 2) Click "Search" button to run key data searching.
- 3) When search is completed, the message box will popup "Search completed"
  - SignalTap will show all signals interface of tCAMIP as shown in Figure 5-9
  - In case to see the different trigger point, user can change trigger condition in SignalTap on step 1).
- 4) Result of key search will be displayed to 'Result' column. "rLantencyCount: X" show number of clocks between first key data to first result data.

Rule table									Key data										
No.	A+0	A+1	A+2	A+3	A+4	A+5	A+6	A+7	No.	A+0	A+1	A+2	A+3	A+4	A+5	A+6	A+7	Expect	Result
1	x	x	0	0	x	x	0	0	1	255	255	0	0	255	255	0	0	1	1
2	x	x	0	0	x	x	0	1	2	255	255	0	0	255	255	0	1	2	2
3	x	x	0	0	x	x	0	x	3	255	255	0	0	255	255	0	255	3	3
4	x	x	0	0	x	x	x	0	4	255	255	0	0	255	255	255	0	4	4
5	x	x	0	0	x	x	x	1	5	255	255	0	0	255	255	255	1	5	5
6	x	x	0	0	x	x	x	x	6	255	255	0	0	255	255	255	255	6	6
7	x	x	0	1	x	x	0	0	7	255	255	0	1	255	255	0	0	7	7
8	x	x	0	1	x	x	0	1	8	255	255	0	1	255	255	0	1	8	8
9	x	x	0	1	x	x	0	x	9	255	255	0	1	255	255	0	255	9	9
10	x	x	0	1	x	x	x	0	10	255	255	0	1	255	255	255	0	10	10
11	x	x	0	1	x	x	x	1	11	255	255	0	1	255	255	255	1	11	11
12	x	x	0	1	x	x	x	x	12	255	255	0	1	255	255	255	255	12	12
13	x	x	0	x	x	x	0	0	13	255	255	0	255	255	255	0	0	13	13
14	x	x	0	x	x	x	0	1	14	255	255	0	255	255	255	0	1	14	14
15	x	x	0	x	x	x	0	x	15	255	255	0	255	255	255	0	255	15	15
16	x	x	0	x	x	x	x	0	16	255	255	0	255	255	255	255	0	16	16
17	x	x	0	x	x	x	x	1	17	255	255	0	255	255	255	255	1	17	17
18	x	x	0	x	x	x	x	x	18	255	255	0	255	255	255	255	255	18	18
19	x	x	x	0	x	x	0	0	19	255	255	255	0	255	255	0	0	19	19
20	x	x	x	0	x	x	0	1	20	255	255	255	0	255	255	0	1	20	20
21	x	x	x	0	x	x	0	x	21	255	255	255	0	255	255	0	255	21	21
22	x	x	x	0	x	x	x	0	22	255	255	255	0	255	255	255	0	22	22
23	x	x	x	0	x	x	x	1	23	255	255	255	0	255	255	255	1	23	23
24	x	x	x	0	x	x	x	x	24	255	255	255	0	255	255	255	255	24	24
25	x	x	x	1	x	x	0	0	25	255	255	255	1	255	255	0	0	25	25
26	x	x	x	1	x	x	0	1	26	255	255	255	1	255	255	0	1	26	26
27	x	x	x	1	x	x	0	x	27	255	255	255	1	255	255	0	255	27	27

Figure 5-8 Search key data

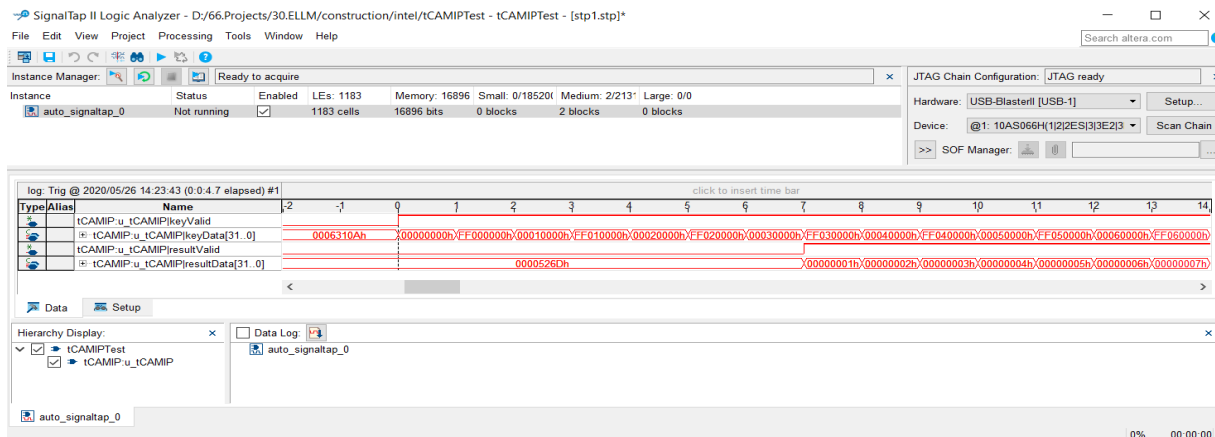


Figure 5-9 Sample of SignalTap result



## 5.7 Compare

To compare between the expected value (from software) and result from hardware searching, the step is as shown in Figure 5-10.

- 1) Click “Compare” button to compare expect data and result data of key data.
- 2) When comparing is completed, if compare have mismatch. The message box will popup “Completed with X mismatch found”, By X is number of mismatches found.

The screenshot shows the tCAM-IP Demo interface. It consists of three main panels:

- Rule table:** A table with columns No., A+0, A+1, A+2, A+3, A+4, A+5, A+6, A+7. It contains 27 rows of binary data.
- Key data:** A table with columns No., A+0, A+1, A+2, A+3, A+4, A+5, A+6, A+7, Expect, Result. It contains 27 rows of hexadecimal and binary data.
- Hardware Connection:** A control panel with buttons for Disconnect, Rule init, Search, and Compare. The Compare button is highlighted with a red box and a circled '1'. The status code is 1 (initial completed).

Figure 5-10 Step to compare data.

## 6 Revision History

Revision	Date	Description
1.02	5-Mar-2021	Revise
1.01	25-Aug-2020	Support up to 64-bit width of rule
1.00	4-Jun-2020	Initial version release