Title of the Experiment: Hardware Trojan Attack I

Group Members' Names: Sujan Kumar Saha, Pankaj Bhowmik

Date: 09/24/2019

Abstract

This experiment is about Hardware trojan attack. Hardware trojans can be integrated in a system which does malicious activities (such as information leakage, information modification etc.) in the chip while in operation. In this experiment, we implemented combinational and sequential trojans and checked how those trojans are triggered and changed the important information (here the key of the des encryption and encrypted data). First, we implemented a given DES encryption module on MAX 10 FPGA on HAHA board and checked the encrypted data in RAM. In the second step, we implemented the combinational trojan and observed the changed output. In the third step, we implemented sequential trojan using finite state machine and without using finite state machine. We did not observe any change in the finite state machine implementation but observed a change in the encrypted data without using FSM.

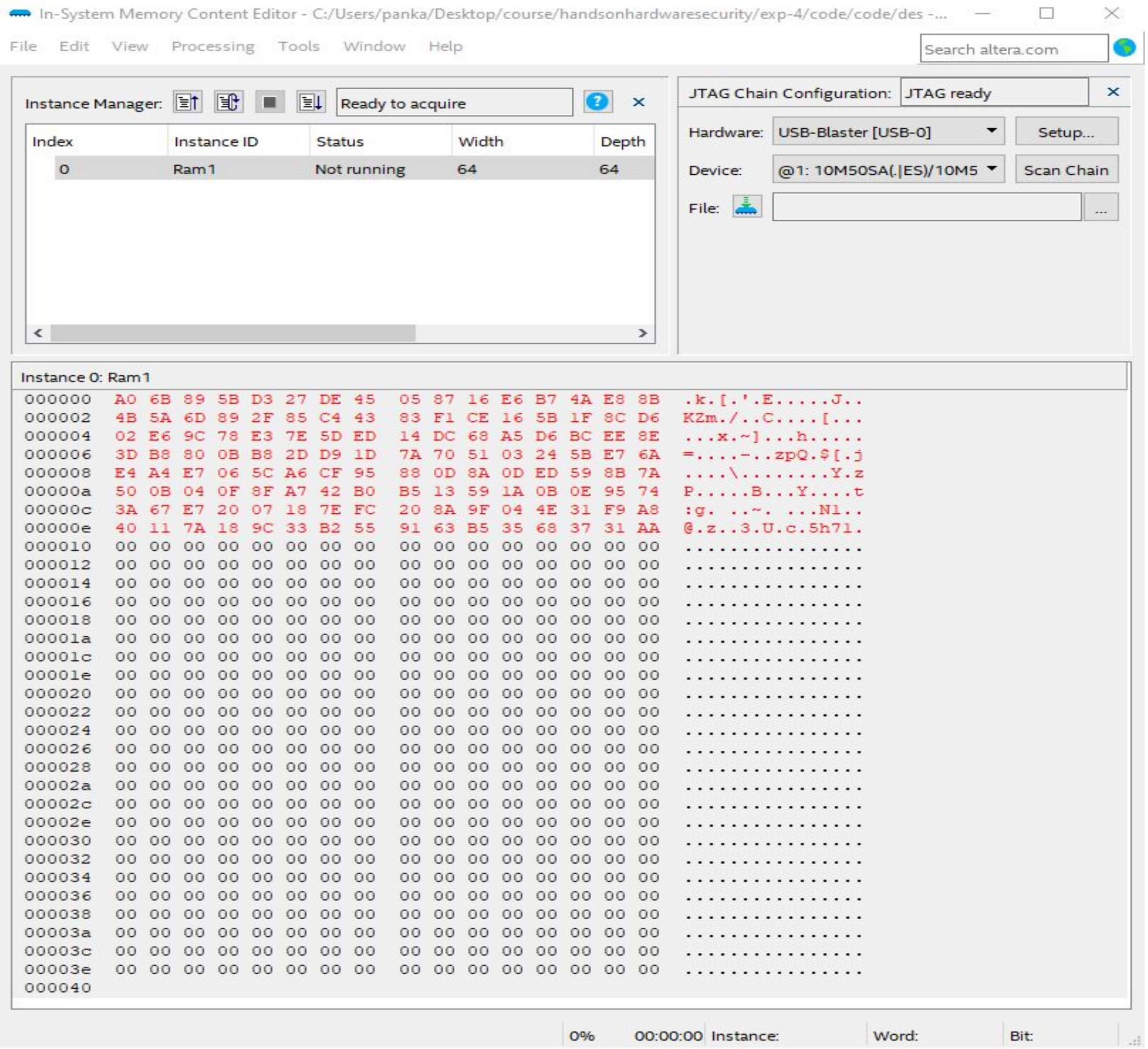
Experiment Details

Goals: The goal of this experiment is to implement a Hardware trojan and trigger that trojan at a certain condition.
Experimental Setup: We used Hardware Hacking (HaHa) board and Quartus software on a computer for this experiment

Part I:

Experiment Steps and results:

1. We have created a new project with the given code files and added RAM to complete the design. We generated bitstream and loaded the given code to MAX 10 FPGA on HaHa board. We opened the In-System Memory Content Editor and observed the DES encrypted 16 round data on the editor. The screenshot is given below.
2. The module "key_sel" is creating the key. Permutation is used to generate the round keys.
3. The module crp is doing the Fiestel function.
4. 693 total logic elements are used.


In-System Memory Content Editor - C:/Users/panka/Desktop/course/handsonhardwaresecurity/exp-4/code/code/des -...
File
Edit
View
Processing
Tools
Window
Help
Search altera.com
Setup...
Scan Chain

Instance Manager: Ready to acquire

Index	Instance ID	Status	Width	Depth
0	Ram1	Not running	64	64

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0]

Device: @1: 10M50SA(.|ES)/10M5

File:

Instance 0: Ram1

```
000000  A0 6B 89 5B D3 27 DE 45  05 87 16 E6 B7 4A E8 8B  .k.[.'.E.....J..
000002  4B 5A 6D 89 2F 85 C4 43  83 F1 CE 16 5B 1F 8C D6  KZm./..C....[...
000004  02 E6 9C 78 E3 7E 5D ED  14 DC 68 A5 D6 BC EE 8E  ...x.~]...h.....
000006  3D B8 80 0B B8 2D D9 1D  7A 70 51 03 24 5B E7 6A  =....-..zpQ.$[.j
000008  E4 A4 E7 06 5C A6 CF 95  88 0D 8A 0D ED 59 8B 7A  ....\........Y.z
00000a  50 0B 04 0F 8F A7 42 B0  B5 13 59 1A 0B 0E 95 74  P.....B...Y....t
00000c  3A 67 E7 20 07 18 7E FC  20 8A 9F 04 4E 31 F9 A8  :g. ..~. ...N1..
00000e  40 11 7A 18 9C 33 B2 55  91 63 B5 35 68 37 31 AA  @.z..3.U.c.5h71.
000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000012  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000014  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000016  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000018  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000022  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000024  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000026  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000028  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000032  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000034  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000036  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000038  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000040
```

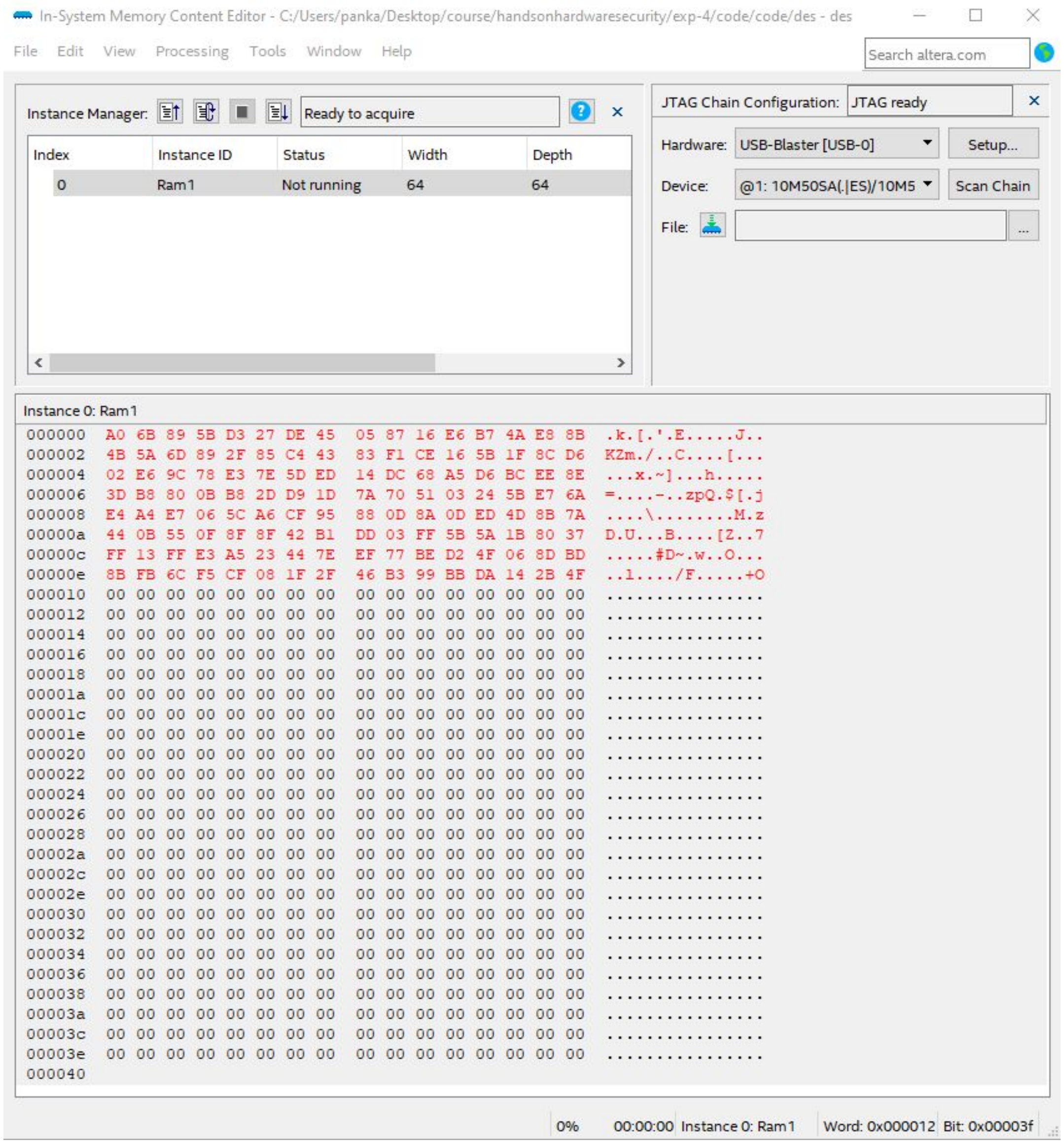
0% 00:00:00 Instance: Word: Bit:

Figure 1: The screenshot of DES encrypted data in the memory window

Part II:

Experiment Steps and results:

1. We have implemented a combinational Trojan in the des.v file. When the trojan is activated, the LSB of the key for DES is inverted. The trigger condition is if the least significant 4 bits of F function output is 4’b0110. We observed the change of the encrypted data in the memory window after 10 rounds. The screenshot is attached here.
2. We did the same for trigger condition 4’b1001. We observed the change of the encrypted data in the memory window after 1 round. The screenshot is attached here.
3. We have done the same for trigger condition 4’b1010. We did not observe any change of data for this trigger condition. That’s why no screenshot is attached.



In-System Memory Content Editor - C:/Users/panka/Desktop/course/handsonhardwaresecurity/exp-4/code/code/des - des
File
Edit
View
Processing
Tools
Window
Help
Search altera.com
Instance Manager:
Setup...
Scan Chain
...

Ready to acquire

Index	Instance ID	Status	Width	Depth
0	Ram1	Not running	64	64

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0]

Device: @1: 10M50SA(.|ES)/10M5

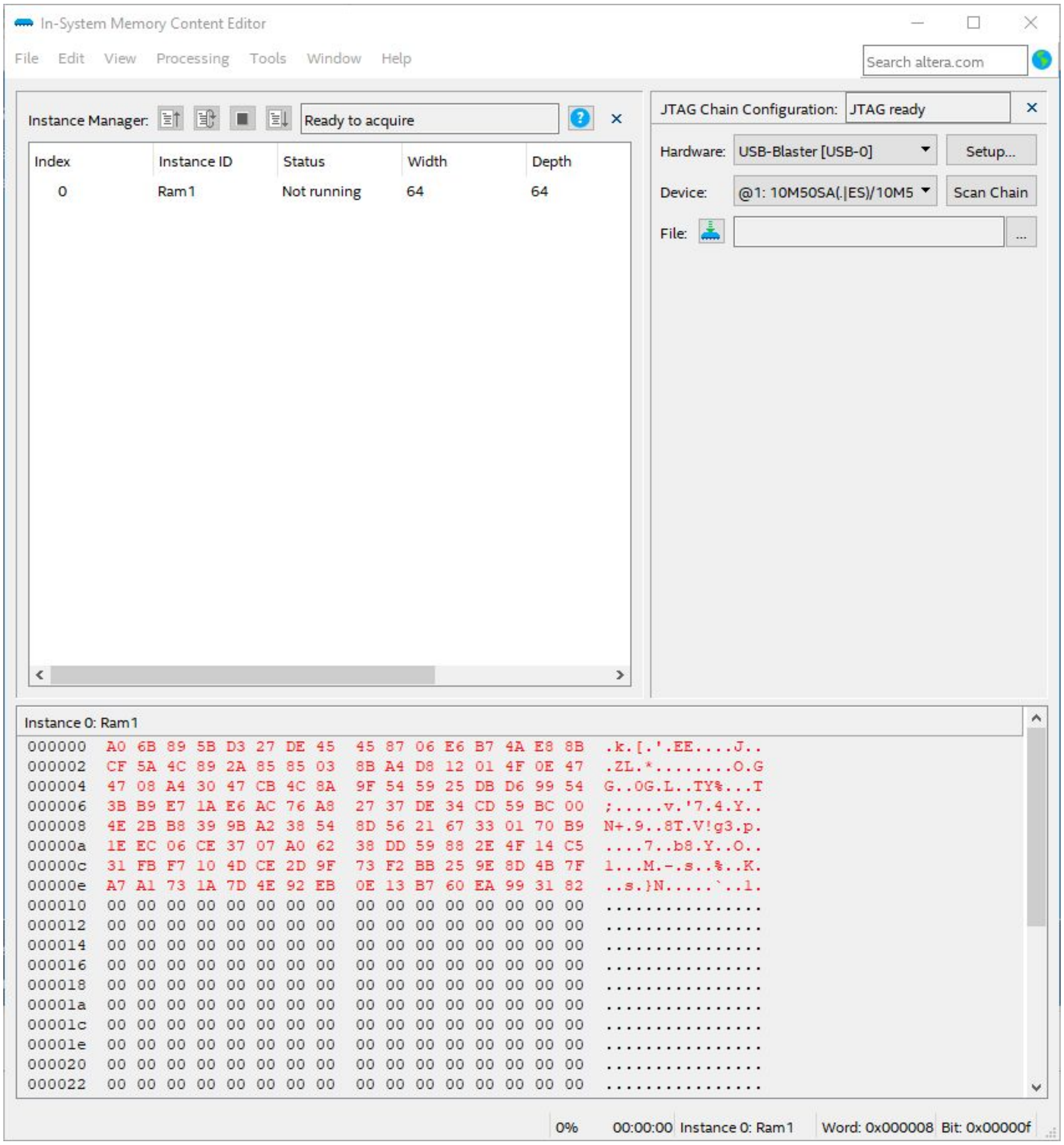
File:

Instance 0: Ram1

```
000000  A0 6B 89 5B D3 27 DE 45  05 87 16 E6 B7 4A E8 8B  .k.[.'.E.....J..
000002  4B 5A 6D 89 2F 85 C4 43  83 F1 CE 16 5B 1F 8C D6  KZm./..C....[...
000004  02 E6 9C 78 E3 7E 5D ED  14 DC 68 A5 D6 BC EE 8E  ...x.~]...h.....
000006  3D B8 80 0B B8 2D D9 1D  7A 70 51 03 24 5B E7 6A  =....-..zpQ.$[.j
000008  E4 A4 E7 06 5C A6 CF 95  88 0D 8A 0D ED 4D 8B 7A  ....\........M.z
00000a  44 0B 55 0F 8F 8F 42 B1  DD 03 FF 5B 5A 1B 80 37  D.U...B....[Z..7
00000c  FF 13 FF E3 A5 23 44 7E  EF 77 BE D2 4F 06 8D BD  .....#D~.w..O...
00000e  8B FB 6C F5 CF 08 1F 2F  46 B3 99 BB DA 14 2B 4F  ..l..../F.....+O
000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000012  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000014  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000016  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000018  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000022  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000024  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000026  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000028  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000032  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000034  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000036  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000038  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000040
```

0% 00:00:00 Instance 0: Ram1 Word: 0x000012 Bit: 0x00003f

Figure 2: The screenshot of DES encrypted data when combinational trojan is triggered at 4’b0110



Index	Instance ID	Status	Width	Depth
0	Ram1	Not running	64	64

Instance 0: Ram1

```
000000  A0 6B 89 5B D3 27 DE 45  45 87 06 E6 B7 4A E8 8B  .k.[.'.EE....J..
000002  CF 5A 4C 89 2A 85 85 03  8B A4 D8 12 01 4F 0E 47  .ZL.*........O.G
000004  47 08 A4 30 47 CB 4C 8A  9F 54 59 25 DB D6 99 54  G..0G.L..TY%...T
000006  3B B9 E7 1A E6 AC 76 A8  27 37 DE 34 CD 59 BC 00  ;.....v.'7.4.Y..
000008  4E 2B B8 39 9B A2 38 54  8D 56 21 67 33 01 70 B9  N+.9..8T.V!g3.p.
00000a  1E EC 06 CE 37 07 A0 62  38 DD 59 88 2E 4F 14 C5  ....7..b8.Y..O..
00000c  31 FB F7 10 4D CE 2D 9F  73 F2 BB 25 9E 8D 4B 7F  1...M.-.s..%..K.
00000e  A7 A1 73 1A 7D 4E 92 EB  0E 13 B7 60 EA 99 31 82  ..s.}N.....`..1.
000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000012  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000014  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000016  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000018  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000022  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

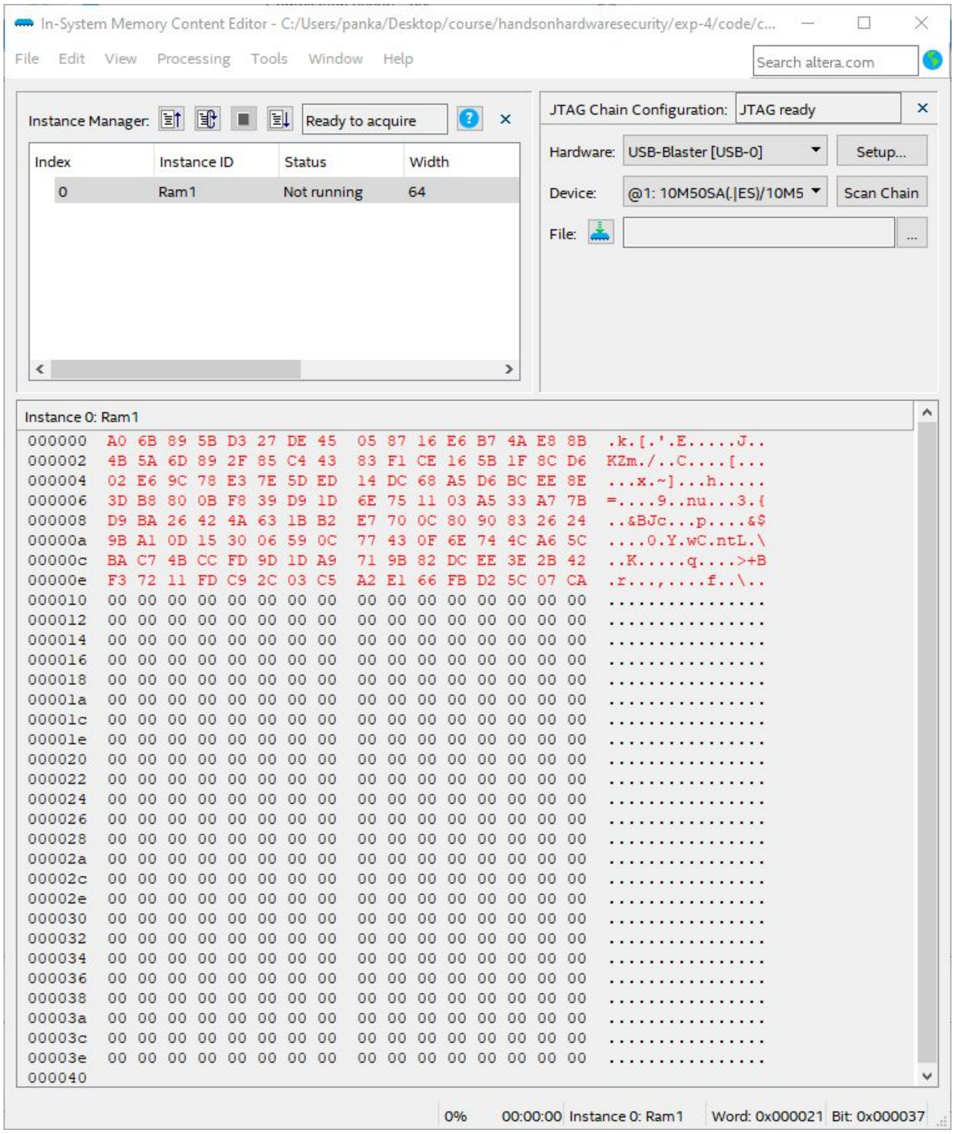
0%   00:00:00   Instance 0: Ram1   Word: 0x000008   Bit: 0x00000f

Figure 3: The screenshot of DES encrypted data when combinational trojan is triggered at 4’b1001

Part III:

Experiment Steps and results:

1. We have implemented a sequential Trojan in the des.v file by using a finite state machine with four states. We used one 2-bit register for current state, one 2-bit register for next state, three 2-bit register for state values and one 56 bit register for intermediate key value. When the trojan is activated, the LSB of the key for DES is inverted. The trigger condition is if the least significant 2 bits of F function output is 2'b01, 2'b10 and 2'b11 sequentially one after another. We did not observe any change in the encrypted data.
2. We did the same for trigger condition of 2'b11, 2'b01 and 2'b00 but did not observe any change in encrypted data.
3. We implemented the sequential trojan in the des.v file in a different way without using finite state machine. That modification is commented in the attached file. By using that implementation, we observed the change in the encrypted data after 8 rounds. The screenshot is attached here.


In-System Memory Content Editor - C:/Users/panka/Desktop/course/handsonhardwaresecurity/exp-4/code/c...
File
Edit
View
Processing
Tools
Window
Help
Search altera.com
Setup...
Scan Chain
...

Instance Manager: Ready to acquire

Index	Instance ID	Status	Width
0	Ram1	Not running	64

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0]

Device: @1: 10M50SA(.|ES)/10M5

File:

Instance 0: Ram1

```
000000  A0 6B 89 5B D3 27 DE 45  05 87 16 E6 B7 4A E8 8B  .k.[.'.E.....J..
000002  4B 5A 6D 89 2F 85 C4 43  83 F1 CE 16 5B 1F 8C D6  KZm./..C....[...
000004  02 E6 9C 78 E3 7E 5D ED  14 DC 68 A5 D6 BC EE 8E  ...x.~]...h.....
000006  3D B8 80 0B F8 39 D9 1D  6E 75 11 03 A5 33 A7 7B  =....9..nu...3.{
000008  D9 BA 26 42 4A 63 1B B2  E7 70 0C 80 90 83 26 24  ..&BJc...p....&$
00000a  9B A1 0D 15 30 06 59 0C  77 43 0F 6E 74 4C A6 5C  ....0.Y.wC.ntL.\
00000c  BA C7 4B CC FD 9D 1D A9  71 9B 82 DC EE 3E 2B 42  ..K.....q....>+B
00000e  F3 72 11 FD C9 2C 03 C5  A2 E1 66 FB D2 5C 07 CA  .r...,....f..\..
000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000012  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000014  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000016  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000018  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00001e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000022  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000024  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000026  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000028  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00002e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000032  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000034  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000036  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000038  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003a  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003c  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00003e  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000040
```

0% 00:00:00 Instance 0: Ram1 Word: 0x000021 Bit: 0x000037

Figure 4: The screenshot of DES encrypted data when sequential trojan is triggered at 2'b01->2'b10->2'b11

Summary:

We implemented hardware trojan attacks in the system and observed the malicious activities performed by the trojans. We faced had to figure out the clock pin for the system. We used pin 88 for our system. We also used pin 30 for reset. Unfortunately, pin 30 was high and we were not getting 16 round data. We were getting one round data. With the help of our TA, we figured out that mistake and got 16 round data using reset pin low. Overall, this experiment is a good learning for us to know how malicious activities can be inserted in a system.