UART Operation

Brief Description:

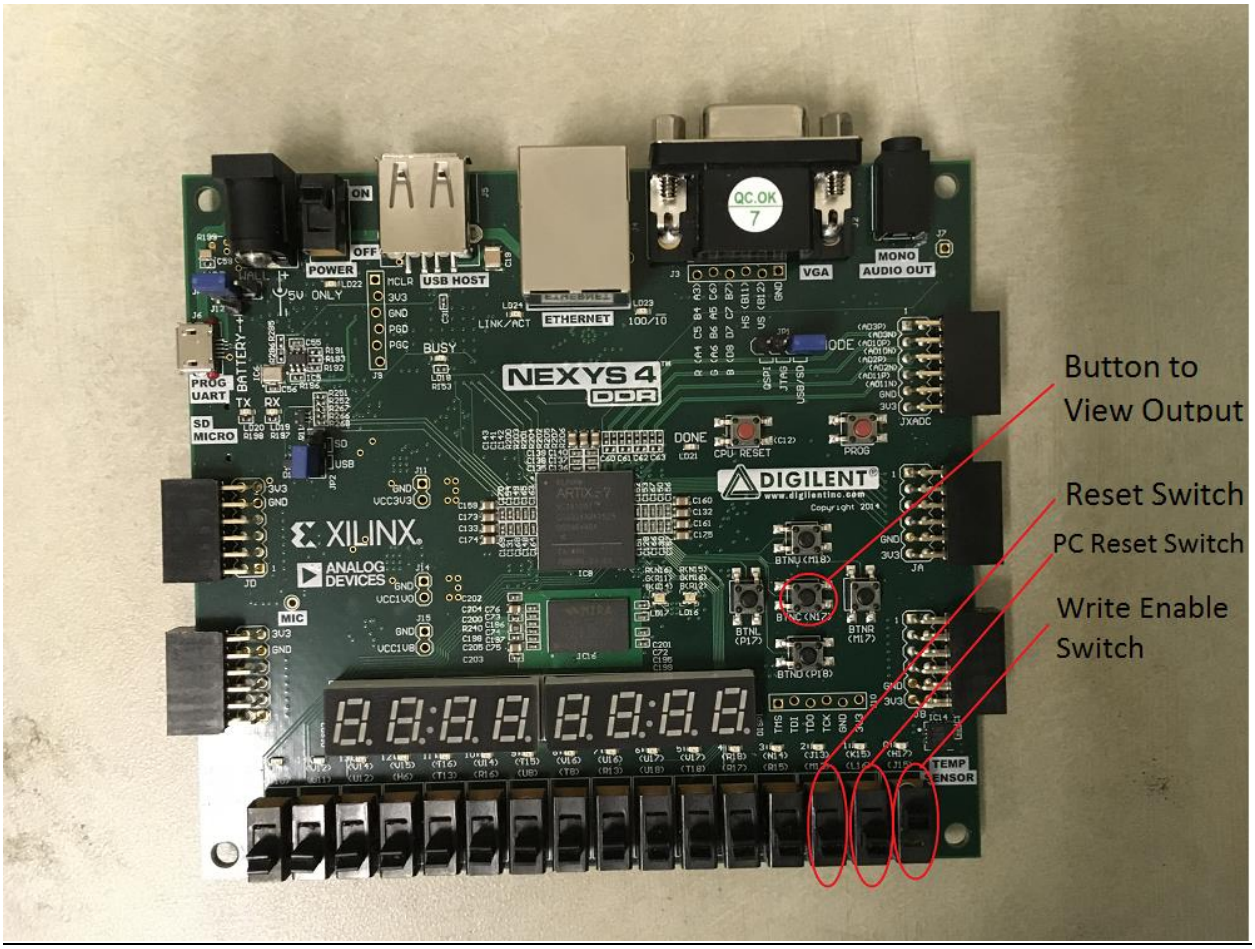


Fig 1. FPGA displaying the buttons and switches being used

The UART module (FPGA) and Software (Controller) is complete and working. We have verified the sample 2, Encryption, Decryption, and round key generation code using UART with the corresponding timing simulation results. The software has the feature to read the inputs such as "memory location" and "instruction" from a text file, and then send the 6 bytes of data over the UART (1 byte fixed "04" to indicate write operation, 1 byte Instruction memory address, 4 bytes of Instruction data).

After reception of the complete packet, the UART segregates and sends the address and the instruction, to be written on the instruction memory.

In this module, the address (first 2 Character) and the instruction data (8 characters), which is to be written on the Instruction memory is made available on a text file.

Here the location (path) of the file having hex codes is given as command line in the software. This can be done in 2 ways (shown in lower sections). The hex codes are read from the file and sent to FPGA via UART.

The software appends "04" in the beginning and sends the whole packet of 6 bytes. At the receiver end, the code is written in such a manner that when all the 6 byte data is received by the FPGA, the FPGA sends an acknowledgement indicating that the data is correctly received and written on the instruction memory.

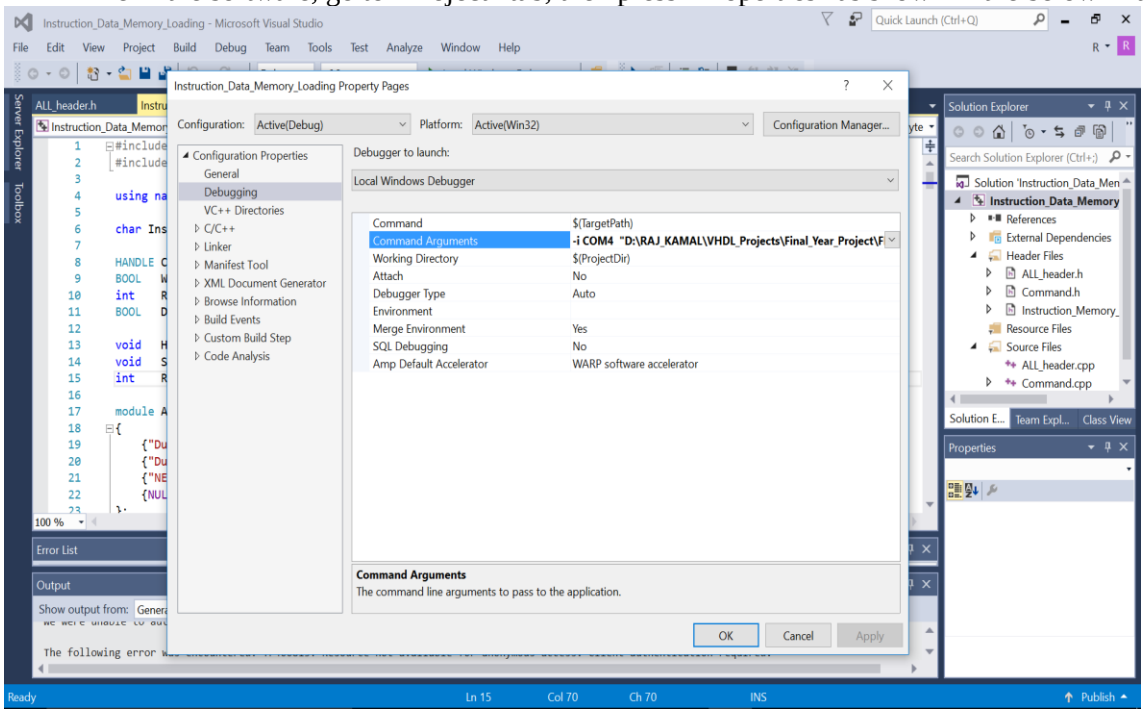
In FPGA, when 6 byte data (1 byte : x"04" to indicate write process, 1 byte for the address on which the instruction will be written on the data memory, 4 bytes of data which is the actual instruction) having address and the instruction is received, it enables ("1") a signal which acts as a clock for the processor and when the data is written on the memory, it lowers ("0") this signal, in turn acting as a clock. So, we first send all the instructions to FPGA, which get written on the instruction memory. Here we have implemented single stepping i.e. on a press of a button (center) on FPGA, a manual clock signal (as described above) is high and again when the button is pressed, the manual clock gets lowered.

A switch (1st switch from the right) acting as "write enable" signal for UART should be turned on during the entire duration, when we want the UART module to receive and write the data on the memory. If this switch is OFF, the UART will receive all the data and send acknowledgment but it will not write the data on the instruction memory.

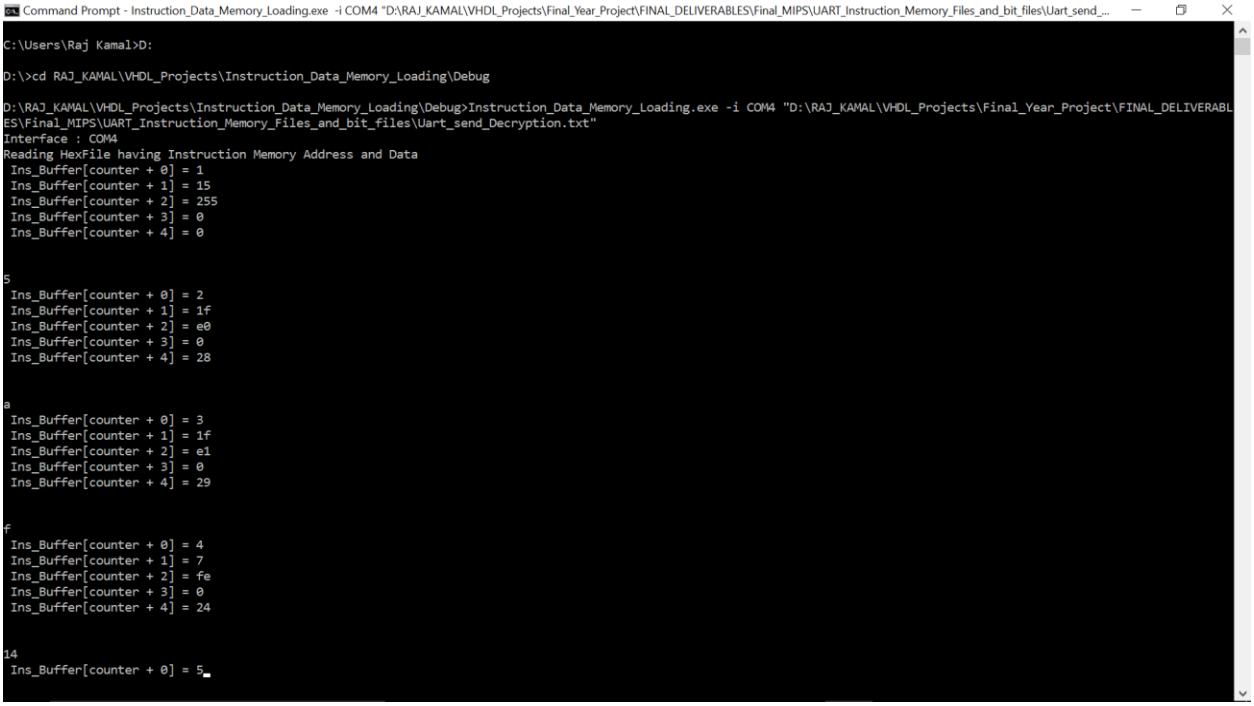
Also, we have used 2 switches to indicate system reset (3rd switch from right) and a PC reset (2nd Switch from Right). On a system reset the Instruction Memory and data memory are eased (filled with "Zeros"). Please refer Fig. 1.

Steps to follow

1) Write the bit file ("UART_Sample_2.bit", "UART_Round_key.bit", "UART_Encryption.bit", "UART_Decryption.bit") on the FPGA using Adept.
2) Switch ON ("1") the Reset switch (3rd from right) and the PC reset switch (2nd switch from right), and the Write Enable switch (1st Switch from right).
3) The software can be used in 2 ways. First method.
   a) Directly from the software (Visual Studio). Write the location (path) of the corresponding HEX file in the software, go to "Project" tab, then press "Properties" as shown in the below image.



b) In the “Command Arguments” write **“**-i COM Port being used, location of HEX file”. For example“iCOM4“D:\RAJ_KAMAL\VHDL_Projects\Final_Year_Project\FINAL_DELIVERABLES\Final_MIPS\UART_Instruction_Memory_Files_and_bit_files\Uart_send_Decryption.txt”.
c) Compile the code using “F7” on Visual Studio.
d) Execute using “F5” on Visual Studio.
e) Initially it will display all the data which will be sent through UART.



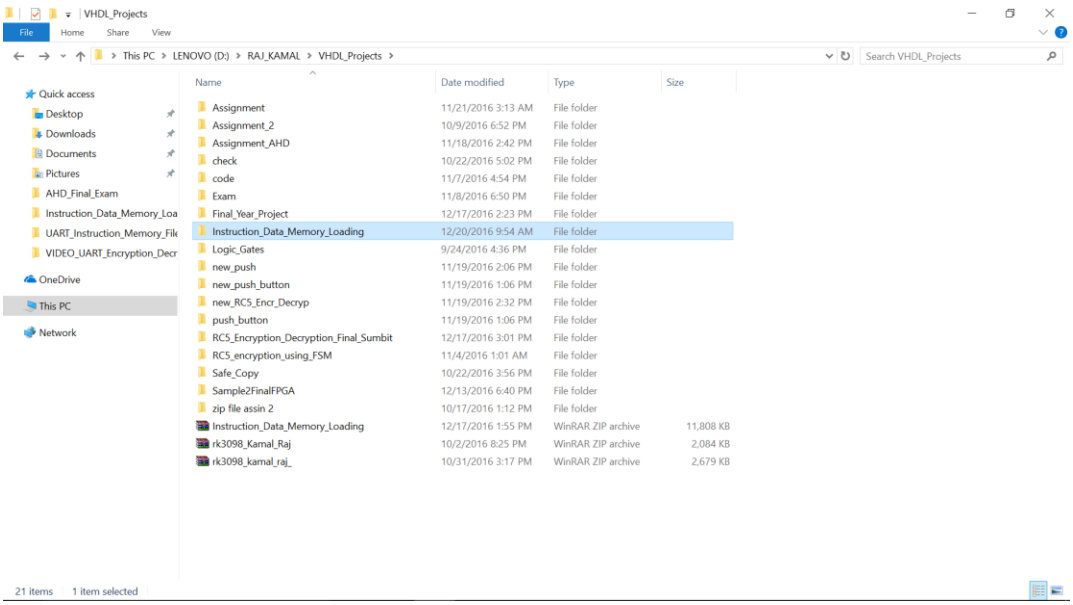
f) Next, the packets are sent and an acknowledgment is received (Shown as Data Received = “1”).

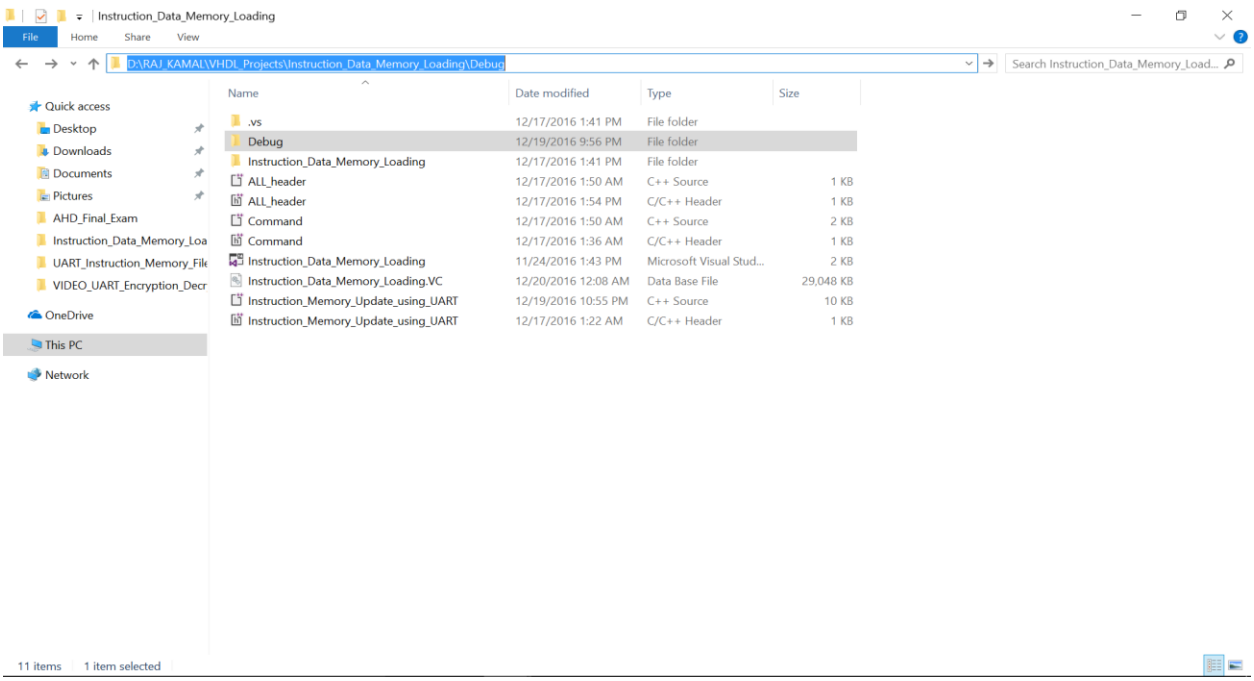
g) Once all the packets are sent, A message “UART Operation Complete” is displayed on the screen.

4) Second method (**Better**). **Using Command Prompt.**

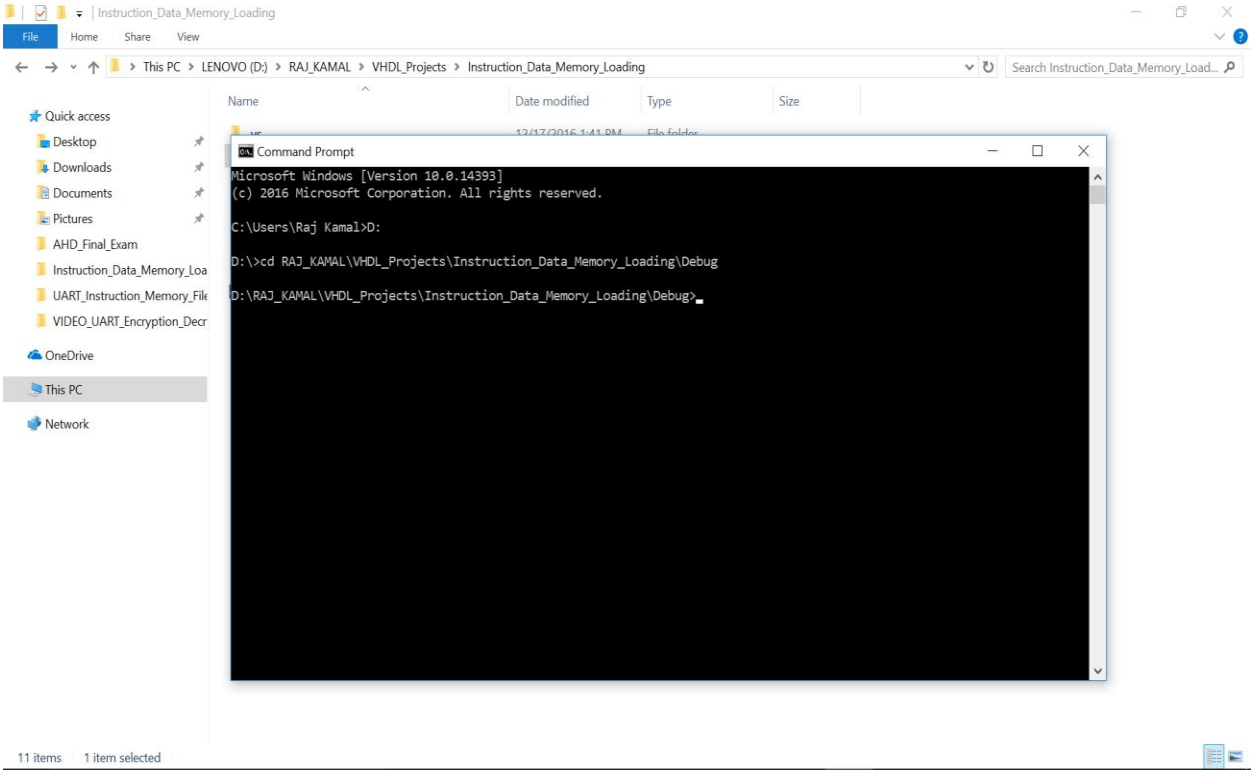
a) Open command prompt.

b) Go to the location having the application i.e. “Instruction_Data_Memory_Loading.exe” and copy the path for the folder having this application. This can be found in “Instruction_Data_Memory_Loading” folder -> “Debug” Folder, as shown in the below images.



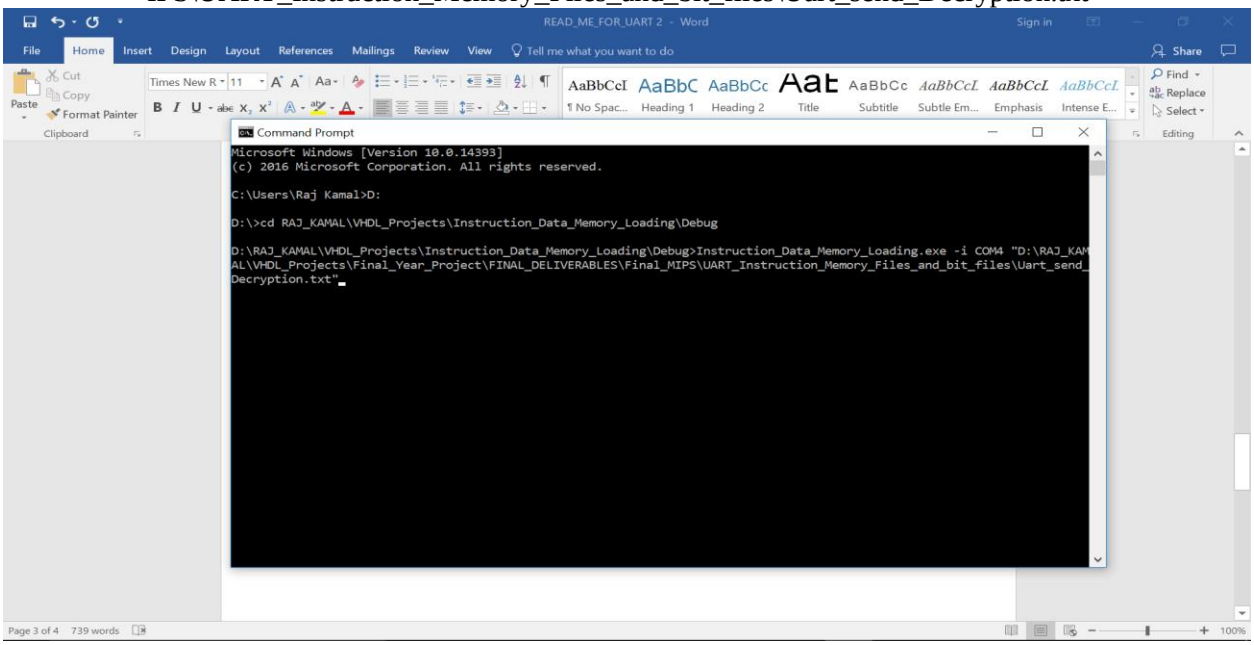


c) Change the current directory to above copied path using the command "**cd PATH**" on the command prompt.
Example:

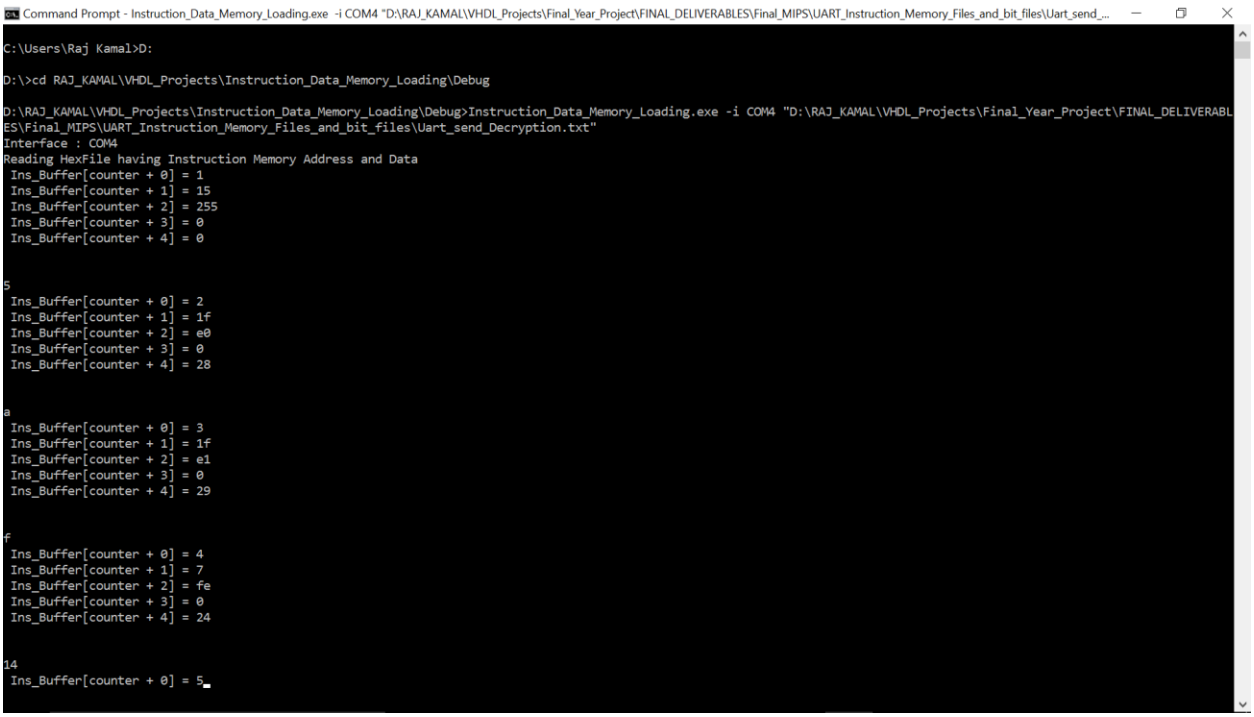


d) Similarly copy the hex file path. For example, if I wish to do Decryption, copy the path for "Uart_send_Decryption.txt" file.

e) Now write the following on command prompt.
Instruction_Data_Memory_Loading.exe -i COM_Port_being_Used "Text_File_Path",
Example:
Instruction_Data_Memory_Loading.exe -i COM4
"D:\RAJ_KAMAL\VHDL_Projects\Final_Year_Project\FINAL_DELIVERABLES\Final_MIPS\UART_Instruction_Memory_Files_and_bit_files\Uart_send_Decryption.txt"



f) Press Enter. The process will start.
g) Initially it will display all the data which will be sent through UART.



h) Next, the packets are sent and an acknowledgment is received for each packet (Shown as Data Received = “1”).

i) Once all the packets are sent, A message “UART Operation Complete” is displayed on the screen.

5) Now on FPGA module, **Switch ON and OFF the PC reset switch (2nd switch from Right). Caution: Do not touch the RESET switch (3rd From Right). It will wipe out the entire Instruction memory just written, and you will have to do the whole process again (as I had to do in the video when reset switch was accidently pressed).**

6) Lower down (Switch OFF i.e. '0') the Write Enable switch (1st Switch from Right).
7) Now on a press of button (**Center**), we can see the output of ALU on the seven-segment display.
8) If you reset the "PC Reset", it is assigned a value x"00000000", and we can again execute all the instructions from beginning.

This completes the UART writing and executing the instructions written on memory on the 32-bit single cycle MIPS processor.