

Hi,

Problem description: JTAG connection failed, consistency, after every time, when stop the debugger on NIOS II hardware:

Problem messages: Here are the following error (Appendix 1- Error message):

Popup message:

‘Launching <project name> Nios II Hardware configuration’ has encountered a problem. Downloading ELF Process failed.

From the Console:

There are no JTAG cable available on your system which match the –cable option you provided (USB-Blaster on localhost (USB-1)).

Details:

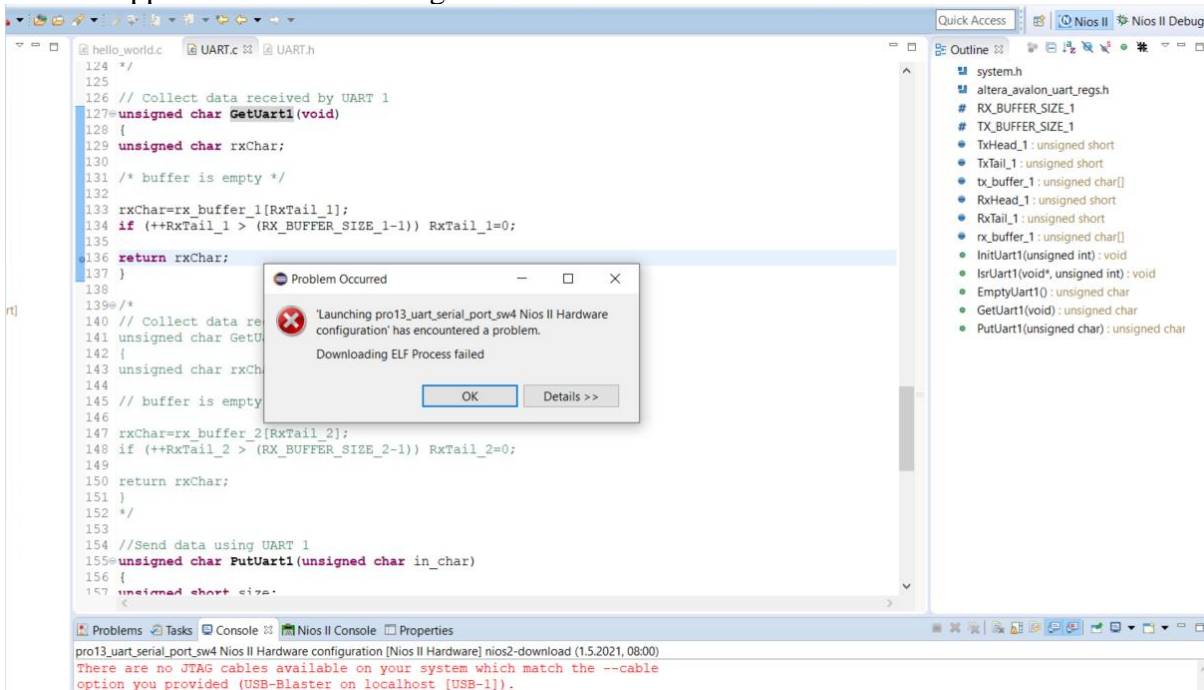
1. The only thing that can restore back the connection is reset to the computer (I’ve tried to close the Quartus and reopen it, disconnect and reconnect the JTAG cable, regenerate the Platform designer and burn the .sof and regenerate the eclipse project- the JTAG connection will not restore).
2. The positive thing is that the problem is consistency- it will happen after every time in the following order:
restart my computer → Open Quartus (18.1) → burn .sof (only one time because I after the JTAG disconnected, it doesn’t help if I disconnect the power to the EVM) → open Eclipse with the project → compile → Run as Nios II hardware → successfully debug the project (break points...) → stop debugging → change/ unchanged my C code → compile → Run as Nios II hardware → JTAG Error (as mentioned in the Problem message section) → reset computer → restart my computer → repeating the process (it will be available now)...
3. I’ve tried to increase the RAM CPU to maximum of 262144 and used the small C libraries (Appendix 3)- the JTAG connection keep disconnect will not restore.

This problem dramatically increases the overhead of the debugging process so I really need a solution for that.

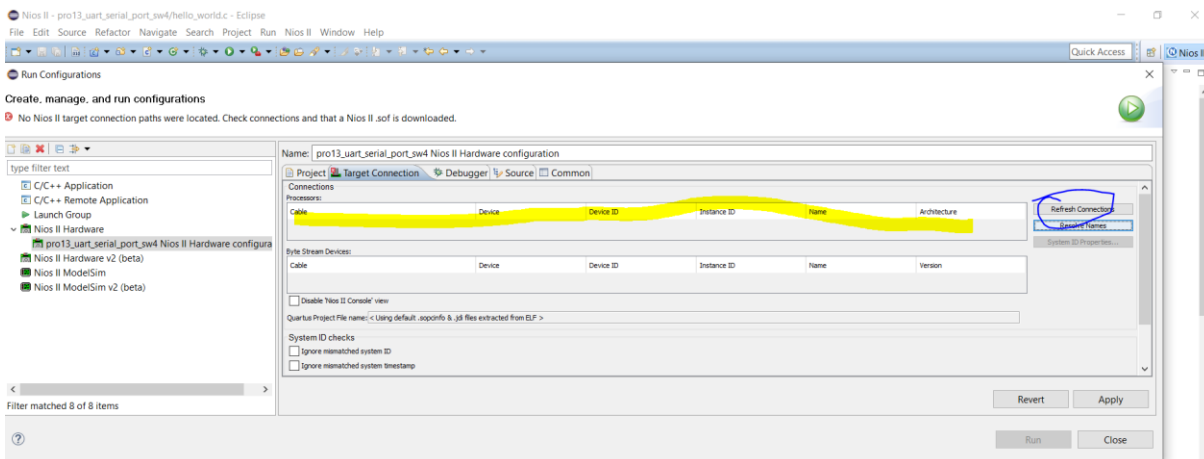
Thanks a lot,

Idan.

Appendix 1- Error message:



Appendix 2- JTAG lost connection:



Appendix 3- enable C small libraries:

The screenshot shows the 'BSP Editor - settings.bsp' window. The 'Settings' tab is active, and the 'Common' section is expanded. Under the 'lib' category, the following options are checked:

- enable_small_c_library
- enable_gprof
- enable_reduced_device_drivers
- enable_sim_optimize

The 'lib linker' section contains the following settings:

- enable_exception_stack:
- exception_stack_size: 1024
- exception_stack_memory_region_name: onchip_memory_0
- enable_interrupt_stack:
- interrupt_stack_size: 1024
- interrupt_stack_memory_region_name: onchip_memory_0

The 'lib make' section contains the following settings:

- bsp_flags_debug: -g
- bsp_flags_optimization: -Ov

The 'Information' pane at the bottom shows a list of messages, including:

- Mapped module: 'tag_uart_0' to use the default driver version.
- Finished loading drivers from executable report.
- Loading BSP settings from settings file.
- Finished loading BSP builder system info file 'C:\ps13_uart_peripheral_port\software\ps13_uart_peripheral_port\bsp\settings.bsp'.
- Generated BSP file in 'C:\ps13_uart_peripheral_port\software\ps13_uart_peripheral_port\ps13_uart_peripheral_port\ps13_uart_peripheral_port\bsp\settings.bsp'.
- Default memory regions will not be persisted in BSP Settings File.
- Generated file 'C:\ps13_uart_peripheral_port\software\ps13_uart_peripheral_port\ps13_uart_peripheral_port\ps13_uart_peripheral_port\ps13_uart_peripheral_port\bsp\settings.bsp'.
- Mapped section 'exception' to memory region 'onchip_memory_0'.
- Mapped section 'entry' to memory region 'test'.
- Added interrupt controller device driver for 'CPU' to all_sys_int0 in all_sys_int.c.
- Added device driver for 'uart' to all_sys_int0 in all_sys_int.c.
- Added device driver for 'tag_uart_0' to all_sys_int0 in all_sys_int.c.
- Finished generating BSP file. Total time taken = 2 seconds.

Buttons for 'Generate' and 'Exit' are visible at the bottom right of the window.

Appendix 4- Platform Designer schematic:

The screenshot displays the Platform Designer interface for a system named 'pro13_uart_serial_port'. The main window shows a schematic diagram with components like 'CPU', 'onchip_memory2_0', and 'uart_0' connected to a common clock source 'clk_0'. A table below the schematic lists the system components and their properties.

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags	Opcode Name
		clk_0	Clock Source	clk	exported					
		clk_in_reset	Clock Input	clk						
		clk	Reset Input	clk						
		clk_reset	Clock Output	clk						
		clk_reset	Reset Output	clk						
		CPU	Non-IP Processor	clk						
		reset	Reset Input	clk						
		data_master	Avion Memory Mapped Master	clk						
		instruction_master	Avion Memory Mapped Master	clk						
		irq	Interrupt Receiver	clk				IRQ 0	IRQ 31	
		debug_reset_request	Reset Output	clk						
		debug_mem_slave	Avion Memory Mapped Slave	clk						
		custom_instruction_master	Custom Instruction Master	clk						
		onchip_memory2_0	On-Chip Memory (RAM or ROM)	clk						
		clk	Clock Input	clk						
		st	Avion Memory Mapped Slave	clk		0x0000_0000	0x0000_ffff			
		reset	Reset Input	clk						
		uart_0	UART (RS-232 Serial Port) Intel FPGA IP	clk						
		clk	Clock Input	clk						
		reset	Reset Input	clk						
		avion_tag_slave	Avion Memory Mapped Slave	clk		0x0004_1040	0x0004_1047			
		irq	Interrupt Sender	clk						
		uart	UART (RS-232 Serial Port) Intel	clk						
		clk	Clock Input	clk						
		reset	Reset Input	clk						
		st	Avion Memory Mapped Slave	clk		0x0004_1000	0x0004_101f			
		external_connection	Conduit	clk						
		irq	Interrupt Sender	clk						
		LED	LED (Parallel I/O) Intel FPGA IP	clk						
		clk	Clock Input	clk						
		reset	Reset Input	clk						
		st	Avion Memory Mapped Slave	clk		0x0004_1030	0x0004_103f			

Messages:

- 2 Info Messages
- pro13_uart_serial_port.Switch: PFD inputs are not hardened in test bench. Undefined values will be read from PFD inputs during simulation.
- pro13_uart_serial_port_uart_0: IFAG UART IP input clock need to be at least double (2x) the operating frequency of IFAG TCK on board.

