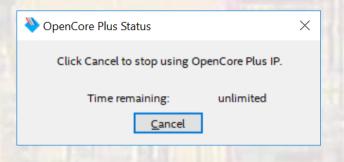
- No spaces allowed in paths or file names
- No simulation can be selected
- The Time-Limited warning box must be left open

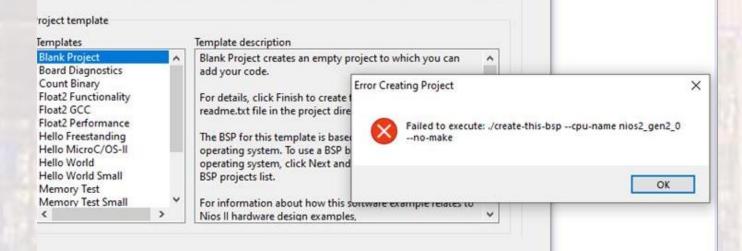


- To debug your C code
 - Select "debug as" instead of "run as"
 - You have access to variables, registers, stepping, breakpoints, ...

- Can't find ... (you built the BSP without the DE10 programmed and the time-limited window open)
 - Select the Target Connection tab
 - Scroll to the far right
 - Select refresh connections

Run Configurations				×					
Create, manage, and run co (3) [Target Connection]: No Nios II	2	s were located. Check connections a	and that a Nios II .sof is do	wnloaded.	- toka				
Image: Second	Project Targ		~	≥l∕nios_pixel_sw\ ∨					
ıs II ModelSim ıs II ModelSim v2 (beta)	File system ELF file na	me:		Name: nios_pixel_sw Nios II Hardware configuration Project Target Connection Debugger Common Source					
				e Device	ID Instance ID	Name	Architecture	Refresh Connections Resolve Names System ID Properties	
Filter matched 8 of 8 items			Re <u>v</u> ert	e Device	ID Instance ID	Name	Version		
?			Rur						

Cannot generate BSP



- You failed to correct the path name when generating the platform designer HDP file
 - Remove the qip file from the project
 - Re-generate HDL with the corrected path
 - Compile, program
 - Open Eclipse ...

© ti

- ELF errors
 - Executable and Linkable Format
 - Contains the executable portions of your code along with information as to how to link, order and debug the sections

MOST ELF errors are CLOCK errors

- 1. Look at your Quartus RTL
 - 1. Make sure you compiled the correct TOP LEVEL design
 - 2. Make sure your pin NAMES are correct (capitalization MATTERS in pin names)
- 2. Check the pin planner and look for any pins that have not been assigned (they will appear white)
- 3. Check your Platform Designer (qsys) design
 - 1. Make sure all blocks have a clock and reset
 - 2. Check PLL frequencies (and phase shifts if using external DRAM)

- NIOS Pixel specific issues
 - My processor starts (prints out "entered main" or something similar) but nothing is going to the VGA display
 - 1. The notes clearly say that alt_up_pixel_buffer_dma_draw() draws to the back buffer, which means it is not going directly to the VGA
 - 2. Since the pixel buffer is in the external SDRAM but your code is in the FPGA SRAM you probably have a connection issue with the SDRAM
 - 1. Make sure the PLL has the correct phase offset
 - 2. Make sure the PLL drives the SDRAM CLK input

-- continued --

- NIOS Pixel specific issues
 - My processor starts (prints out "entered main" or something similar but nothing is going to the VGA display)
 - 3. Edit the BSP and open the Linker Script tab.
 - 1. .txt and the exceptions should point to on-chip-memory
 - 2. All other sections should point to the SDRAM

If not – recheck the wiring on the Pixel DMA buffer and the front and back buffer memory addresses

© ti

- NIOS Pixel specific issues
 - My processor does nothing (no elf error)
 - Make sure you checked the "small C library" and "reduced device driver" boxes in the BSP editor
 - Watch for "processor OK" and "processor paused" in the Nios console window
 - Check the memory allocation during the build project

CDT Build Console [nios_pixel_sw]	
<pre>nios2-elf-g++ -T'/nios_pixel_sw_bsp//linker.x' -msys-crt0='/nios_pixel_sw_bsp//obj/HAL/st</pre>	rc, ^
nios2-elf-insert nios_pixel_sw_elf =-thread_model halcpu_name hros2_gen2_0gsys trues:	im
<pre>Info: (nios_pixel_sw.elf) 16 KBytes program size (code + initialized data).</pre>	
Info: 65534 KBytes free for stack + heap.	
Info: Creating nios_pixel_sw.objdamp	
<pre>nios2-elf-objdumpdisassemblesymsall-headersource nios_pixel_sw.elf >nios_pixel_sw</pre>	.0}
[nios_pixel_sw build complete]	
15:07:18 Build Finished (took 19s.296ms)	~
<	>

- NIOS Pixel specific issues
 - My display has a fixed colorful pattern
 - SDRAM is in it's default state you are not writing to it or it is not connected properly
 - My display has a fixed colorful pattern except the top has some ransom flickering pixels
 - Your program is running out of SDRAM and overlapping your pixel buffer
 - This usually is the result of a connection error between the pixel buffer and the SDRAM controller

© ti