

Lab 05

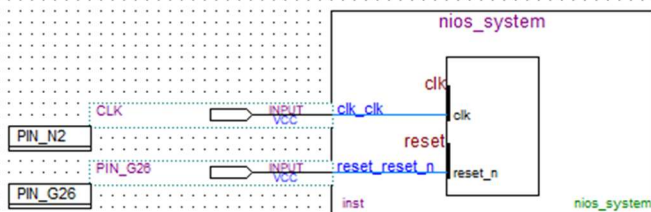
Josh Bussis

Engr 220L - A

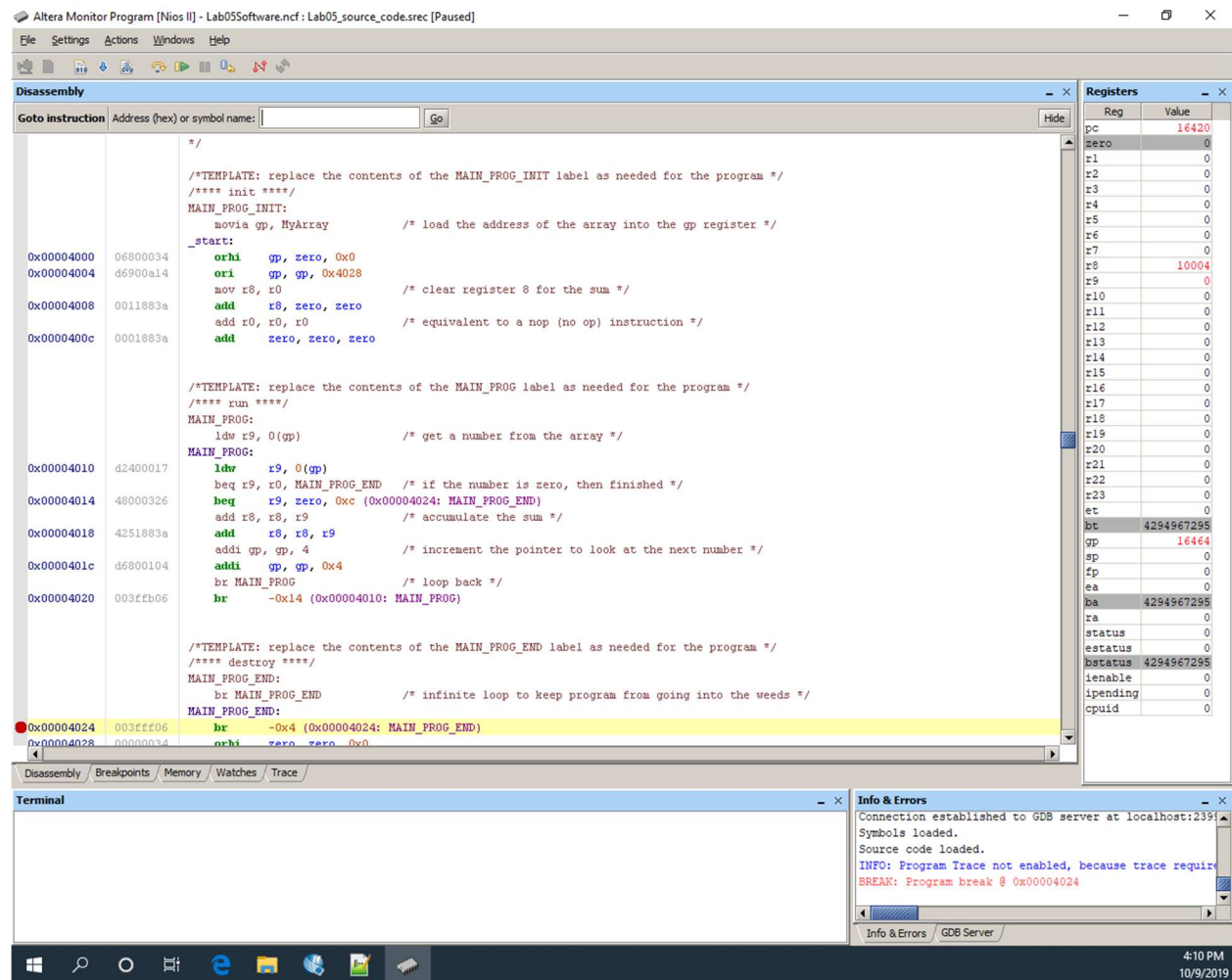
10/09/2019

Part 1:

System Contents Address Map Clock Settings Project Settings Instance Parameters System Inspector HDL Example Generation							
Use	Connections	Name	Description	Export	Clock	Base	End
<input checked="" type="checkbox"/>		<div>clk_0</div> <div>clk_in</div> <div>clk_in_reset</div> <div>clk</div> <div>clk_reset</div>	<div>Clock Source</div> <div>Clock Input</div> <div>Reset Input</div> <div>Clock Output</div> <div>Reset Output</div>	<div>clk</div> <div>reset</div> <div>Double-click to export</div> <div>Double-click to export</div>	clk_0		
<input checked="" type="checkbox"/>		<div>nios2_qsys_0</div> <div>clk</div> <div>reset_n</div> <div>data_master</div> <div>instruction_master</div> <div>jtag_debug_module_re...</div> <div>jtag_debug_module</div> <div>custom_instruction_m...</div>	<div>Nios II Processor</div> <div>Clock Input</div> <div>Reset Input</div> <div>Avalon Memory Mapped Master</div> <div>Avalon Memory Mapped Master</div> <div>Reset Output</div> <div>Avalon Memory Mapped Slave</div> <div>Custom Instruction Master</div>	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div>	clk_0		IRQ 0 IRQ 31
<input checked="" type="checkbox"/>		<div>onchip_memory2_0</div> <div>clk1</div> <div>s1</div> <div>reset1</div>	<div>On-Chip Memory (RAM or ROM)</div> <div>Clock Input</div> <div>Avalon Memory Mapped Slave</div> <div>Reset Input</div>	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div>	clk_0	0x8800	0x8fff
					clk1	0x4000	0x7fff



Part 2:



Movia explanation: The movia command was broken down to two different basic functions. The first function, orhi, shifted the address found at 0x0 16 bits left into the gp register, giving a starting point for the program. Then, the ori, added the contents (through “or-ing” the contents with zeros) of 0x4028 to the address in the gp register, thus giving the address for the array to the pointer.

Register Name	How used in the program	Final Value
r8	Stored the sum of the array items	10004
r9	Stored the current item of the array	0
PC	Program Counter: held the address of the current instruction	16420
gp	Pointer for the array	16464

Altera Monitor Program [Nios II] - Lab05Software.ncf : Lab05_source_code.srec [Paused]

File Settings Actions Windows Help

Memory

Goto memory address Address (hex): Go Query All Devices Refresh Memory Hide

	+0x0	+0x4	+0x8	+0xc
0x00003fe0	0	0	0	0
0x00003ff0	0	0	0	0
0x00004000	109051956	3599763988	1148986	100410
0x00004010	3527409687	1207960358	1112639546	3598713092
0x00004020	4193030	4194054	52	377
0x00004030	136	2011	23	872
0x00004040	1003	1	97	5432
0x00004050	0	0	0	0
0x00004060	0	0	0	0
0x00004070	0	0	0	0
0x00004080	0	0	0	0
0x00004090	0	0	0	0
0x000040a0	0	0	0	0
0x000040b0	0	0	0	0
0x000040c0	0	0	0	0
0x000040d0	0	0	0	0

Registers

Reg	Value
pc	16420
zero	0
r1	0
r2	0
r3	0
r4	0
r5	0
r6	0
r7	0
r8	10004
r9	0
r10	0
r11	0
r12	0
r13	0
r14	0
r15	0
r16	0
r17	0
r18	0
r19	0

Disassembly Breakpoints Memory Watches Trace

Terminal

Info & Errors

Connection established to GDB server at localhost:2396
 Symbols loaded.
 Source code loaded.
 INFO: Program Trace not enabled, because trace requires
 BREAK: Program break @ 0x00004024

Info & Errors GDB Server

Explanation: This window shows the contents of RAM around the memory value of 0x4000. The values originally stored in the array can clearly be seen starting at memory location 0x4020, offset 0x8. From this point on, until the first zero value, are the values of the array. The values before this point were probably the values of the instructions for the program to execute or other memory locations used to store data.