

## SSRAM\_FLASH user guide

Target		Clock Settings			
Device Family: Stratix IV		Name	Source	MHz	
		clk_50	External	50.0	
		system_clk	pll_0.c0	83.333333	
		ssram_clk	pll_0.c1	83.333333	

  

Use	Connections	Module Name	Description	Clock	Base	End	Tags	IRQ
<input checked="" type="checkbox"/>		cpu_0	Nios II Processor	system_clk				
		instruction_master	Avalon Memory Mapped Master	system_clk				
		data_master	Avalon Memory Mapped Master	system_clk				
		jtag_debug_module	Avalon Memory Mapped Slave	system_clk	0x00001000	0x000017ff		
<input checked="" type="checkbox"/>		onchip_memory2_0	On-Chip Memory (RAM or ROM)	system_clk	0x00800000	0x008f9fff		
		s1	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		sc_inf_inst	sc_inf	system_clk	0x00000000	0x000003ff		
		nios	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		jtag_avalon_master_0	JTAG to Avalon Master Bridge	system_clk				
		master	Avalon Memory Mapped Master	system_clk				
<input checked="" type="checkbox"/>		pll_0	PLL	clk_50	0x00000420	0x0000043f		
		s1	Avalon Memory Mapped Slave	clk_50				
<input checked="" type="checkbox"/>		led_out	PIO (Parallel I/O)	system_clk	0x00000490	0x0000049f		
		s1	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		dipsw_in	PIO (Parallel I/O)	system_clk	0x000004a0	0x000004af		
		s1	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		pb_in	PIO (Parallel I/O)	system_clk	0x000004c0	0x000004cf		
		s1	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		lcd_0	Character LCD	system_clk	0x000004b0	0x000004bf		
		control_slave	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		jtag_uart_0	JTAG UART	system_clk	0x000004d0	0x000004d7		
		avalon_jtag_slave	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		time_stamp_timer	Interval Timer	system_clk	0x00000440	0x0000045f		
		s1	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		system_clk_timer	Interval Timer	system_clk	0x00000460	0x0000047f		
		s1	Avalon Memory Mapped Slave	system_clk				
<input checked="" type="checkbox"/>		tri_state_bridge_0	Avalon-MM Tristate Bridge	system_clk				
		avalon_slave	Avalon Memory Mapped Slave	system_clk				
		tristate_master	Avalon Memory Mapped Tristate Master	system_clk				
<input checked="" type="checkbox"/>		cfi_flash_0	Flash Memory Interface (CFI)	system_clk	0x04000000	0x07ffffff		
		s1	Avalon Memory Mapped Tristate Slave	system_clk				
<input checked="" type="checkbox"/>		ssram_0	Cypress CY7C1380C SSRAM	ssram_clk	0x00400000	0x005fffff		
		s1	Avalon Memory Mapped Tristate Slave	ssram_clk				
<input checked="" type="checkbox"/>		max2_inf	Cypress CY7C1380C SSRAM	ssram_clk	0x00600000	0x006fffff		
		s1	Avalon Memory Mapped Tristate Slave	ssram_clk				
<input checked="" type="checkbox"/>		temp	PIO (Parallel I/O)	clk_50	0x00000400	0x0000040f		
		s1	Avalon Memory Mapped Slave	clk_50				
<input checked="" type="checkbox"/>		volt	PIO (Parallel I/O)	clk_50	0x00000410	0x0000041f		
		s1	Avalon Memory Mapped Slave	clk_50				

This project is just a connection to a lot of components.  
 There are 2 Avalon masters, CPU and JTAG-Avalon-Memory mapped Master.  
 Below them, there are

Component	instance name	Start address	End address
On chip memory		0x00800000	0x008f9fff
System Console Interface	(sc_inf_inst)	0x00000000	0x000003ff
PLL	(pll_0)	0x420	0x43f
LED interface	(led_out)	0x490	0x49f
Dipswitch interface	(dipsw_in)	0x4a0	0x4af
Push Button interface	(pb_in)	0x4c0	0x4cf
LCD interface	(lcd_0)	0x4b0	0x4bf
Temperature control	(temp)	0x400	0x40f
12V Voltage interface	(volt)	0x410	0x41f
CFI Flash interface	(cfi_flash_0)	0x04000000	0x07ffffff
SSRAM interface	(ssram_0)	0x00400000	0x005fffff

MAX-II register interface (max2\_inf) 0x00600000 0x006ffff

I recommend to use System Console.

The system console can be called from SOPC Builder or command line.

Here are the sample commands to use

1. set nios [ lindex [ get\_service\_paths master ] 1 ]
2. open\_service master \$nios
3. master\_read\_memory \$nios 0x00400 4
4. master\_write\_memory \$nios 0x0040C [ list 0x54 ]

3 and 4 is accessing the component.

Master\_read\_memory \$nios [target\_read\_address] [byte\_to\_read]

So, the example is reading address 0x400 for 4 bytes.

master\_write\_memory \$nios [target\_write\_address] [list write\_data]

write\_data is the hex data to write.

If you want to write 32 bits data, it will be [list 0x00 0x01 0x02 0x03]

For more information about the system console, please check online document for the system console.

```
System Console
(INCLUDING device programming or simulation files), and any
associated documentation or information is expressly subject
to the terms and conditions of the Altera Program License
Subscription Agreement, Altera MegaCore Function License
Agreement, or other applicable license agreements, including,
without limitation, that your use is for the sole purpose of
programming logic devices manufactured by Altera and sold by
Altera or its authorized distributors. Please refer to the
applicable agreement for further details.
-----
Welcome to Altera's System Console

This Tcl console provides access to the hardware modules instantiated in your
FPGA. You can use System Console for all of the following purposes:

* To start, stop, or step a Nios II processor
* To read or write Avalon Memory-Mapped (Avalon-MM) slaves using special
  masters
* To sample the SOPC system clock as well as system reset signal
* To run JTAG loopback tests to analyze board noise problems
* To shift arbitrary instruction register and data register values to
  instantiated system level debug (SLD) nodes

In addition, the directory $QUARTUS_ROOTDIR/sopc_builder/system_console_macros
contains Tcl files that provide miscellaneous utilities and examples of how to
access the functionality provided. You can include those macros in your
scripts by issuing Tcl source commands.
-----

% set nios [ lindex [ get_service_paths master ] 1 ]
/connection/USB-Blaster [USB-0] /EP4SGX230/[MFG:110 ID:132 INST:0 VER:1]

% open_service master $nios

% master_read_memory $nios 0x00400 4
0x25 0x00 0x00 0x00

%
```