## **RiscFree**<sup>TM</sup> **SDK** IDE, Compiler and Debugger

9 • 🗊 🕼 🐘 💭 🖬	<b>a</b> ( ) B	- 00 🖷	1 3. 2	a.e.  10 ≡	12. 4	ଧାଣାର•୦• <b>୦</b> •ା <u>୫</u> ୭୫୫୫ <mark>୭</mark> ୫•ା					Q 18 4	4
Debug 🗙 🍋 Project Explore			0	1 1 1 =	0	Function Flow View X				X E 0 53	1 K & & -   9 & K &   # + 8 =	
sifive_sum [Ashling RISC-V H	ardware De	bugging	đ			13:14:48.044010 13:14:48.044020 13:14:48.044030 13:14:48.044040 13:14:4	044050 13	14:48.044060	13:14:48.044070 13:14:48.044080	13:14:48.044090 13:	14-48-044100 13-14-48-044110 13-14-48-044120	
✓ ∰ sifive_sum.elf						Call Stack		- 14 14		• ••••••••••••••••••••••••••••••••••••		
✓ 🧬 Thread #1 [TAP 1 (SiFi			ended : Break	(point)				man.				
main() at sum.c:12						Runtime ontrollect	110	Long Long	MemoryBeadWrite1		Memon/ReadWrite?	
ash-riscv-gdb-server.exe						3 MemoryReadWrite1 MemoryReadWrite2			SontArravAscending		SontArrayDescending	
iscv64-unknown-elf-gdt	LEX E					A Sort/maxAvending Sort/Ama/Decending	and the second se	1000		COMPANY OF THE OWNER OF	and county county can be can a county of the second second	
						s com contempor and an and an and an and a second and a second and an an	a de la caractería de la c					-
🕫 Variables 🚿 Expressi 强 P	eriphe_	11 Regist										. v.
			20	A 8 0 1	* 8	¢						>
Name	1	alue				(i) sum.c × (i) start.S		Trace :				
Ht General Purpose Registers						Write Reg Test(); /* Code modify registers and gdb-server read back */		Core No				-
IIII zero	0	0x0 0x80000c9e				<pre>ErrRet = FinishedWriteRegisterTest();</pre>		Core No		Disassembly	Source Line	
1111 58								0	0x80000d90 fc f4 0d a3	sb a5,-37(s0)	a[3] = a[2] + 1	- 1
lill sp	0x80002+68					//Writing global variables to watch		0	0x80000d94 fe c4 27 83	1w a5,-20(50)	ulVariable3 = ulVariable4;	
IIII gp	0x800030c0					gCharNatch = 5; gShortNatch = 10;		0	0x800000d98 fe f4 24 23 0x800000d9c fe c4 27 83		ulVariable4++;	
iiii tp						gIntHatch = 10;		0	0x80000ds2 fe 24 27 83 0x80000da0 07 85	1W A5,-20(80) addi a5,a5,1	ulvariable4++;	
1111 #D	0	1000				WatchPointTesti();		0	0x80000da2 fe f4 26 23	aw a520(a0)		
iiii t1	0	bb44			× -	WatchPointTestFinished();		0	0x80000da6 fd b4 07 83		a[5] = a[3] + 1;	
4					>	return NO_ERROR;		0	0x80000das 0f f7 f7 93	zext.b a5,a5		- 10
Trace Profiling View ×						}		0	0x80000dae 07 85	addi a5,a5,1		
	100 100	1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0.00	a la	a /**		0	0x80000db0 Of #7 #7 93	zext.b a5.a5		
	@ -EI	~	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	DI ACTI MARIE NO.	1.0	* Testing Runtime control functionalities		0	0x80000db4 07 e2	sili a5,a5,0x18		
<sup>0</sup> type filter text					1	*/		0	0x80000db6 87 e1	srai a5,a5,0x18		
Name (location)	c	C.11.	Time/Call			<pre>static void RuntimeControlTest(void)</pre>		0	Ox80000db8 fc f4 0e 23	sb a5,-36(s0)		
	13073	Cano	nine) car		0.0%	{ //Step OverTest		0	0x80000dbc fe c4 27 83	1w a5,-20(s0)	ulVariable3 = ulVariable4;	
<ul> <li>Summary</li> <li>GetMinimumsIndex</li> </ul>	9767	64	152.609us			SoftwareBreakpoint1();		0	0x80000dc0 fe f4 24 23		•••	
> SwapEntry	1788	64	27.937us		1685	SoftwareBreakpoint2();		0	0x80000dc4 fe c4 27 83		ulVariable4++;	
<ul> <li>SortArrayAscending</li> </ul>	521	2	260,500us					0	0x80000dc8 07 85	addi a5,a5,1		
> SortArrayDescending	518	2	259.000us			<pre>//Step Into NemoryReadWrite1();</pre>		0	0x80000dca fe f4 26 23			
> memopy	172	2	85.000us	1		NemoryReadWrite2();		0	0x80000dce fd c4 07 83		a[5] = a[4] + 1;	-
> WatchPointTest1	116	1	116.000us		89%	)		0	0x80000dd2 0f f7 f7 93 0x80000dd6 07 85	zext.b a5,a5 add1 a5,a5,1		-
> main	39	0			1%			0	0x80000dd8 0f f7 f7 93			-
> MemoryReadWrite1	28	2	14.000us		21%	Function: main		0	0x800000ddg 07 e2	slli a5,a5,0x18		
> MemoryReadWrite2	28	2	14.000us		21%	Engineer: Nikolay Chokoey			0x800000dde 87 el	srai a5, a5, 0x18		
> SoftwareBreakpoint1	18	2	9.000us	0.		Tarvit	~		OACCOULTE OF EL	, d5, 0410		. *
> SoftwareBreakpoint2	16	2	8.000us					¢.				,
> RuntimeControlTest	13	1	13.000us	0.		🖸 Console 🗙 🖹 Problems 🔘 Executables 🛛 🔳 💥 🐩 🐘 🔝 🥥 🗩 💭 😁	· · - 0	1 Memor	y Browser ×			
> HardwareBreakpoint1	9	1	9.000us			ifive_sum (Ashling RISC-V Hardware Debugging)		0x80000	00		<ul> <li>Go New Ta</li> </ul>	ah.
> FinishedWriteRegisterTest	8	1	8.000us			Ashling GDB Server for RISC-V (ash-riscv-gdb-server).	~	1111	<traditional> ×</traditional>		1 (	11
> HardwareSreakpoint2	8	1	8.000us	- 0.		/23.3.1, 17-Oct-2023, (c)Ashling Microsystems Ltd 2023.				0000/000 00016247 01	414432 11418882 088800622 00010001	
> MemoryReadWrite3	8	1	8.000us	- 0.		Initializing connection		0x800000				<u> </u>
> Nop_Reg_Test	8	1	8.000us 8.000us	0.	1076 C	checking for an active debug connection using the selected debug probe (SN: 1880101):		0x80000C			878593 86C18513 06132341 27870480	
> Write_Reg_Test	8	1	a.000us	- 0.	0.0	Connected to target device SiFive-E using Opella-XD (1880101) via JTAG at 1.00MHz.		0×80000C			293769 A423872A 47158451 84518823	
						(diskware: v1.6.2, 05-May-2023, firmware: v1.6.5, 05-Sep-2023). Info : Active Harts Detected : 1		0x800000				100
						Info : [0] System architecture : RV32		0x80000C				
					1	Info : [0] Debug version : v0.13		0x800000			414452 11418882 00000002 85354781 820141 C6221141 47810888 44328538	
					1	Info : [0] Number of hardware breakpoints available : 4		0x50000C			83FEF4 12238421 A783FEF4 26238441	
¢					3	TATA I (B) BURDER AT ARABASE BUTTERY: 15	>				SUPERA 12238421 A/SUPERA 26238441	~
									itable Smart Insert	128:4:3541		

## RiscFree Features

- IDE with full source & project creation, editing, build & debug support.
- Works with Windows and Linux Hosts.
- RiscFree includes a single-shot installer that installs & automatically configures all the component tools to work "out-of-the-box".
- Automatic source-code formatting, syntax colouring and function folding.
- Integrated GCC and/or LLVM compiler toolchains.
- Project wizards, templates and examples.
- Ashling hardware debug & trace probe support is fully integrated into the *RiscFree* debugger allowing hardware-based debug & easy setup, capture & display of trace, profiling & analysis data.
- Integrated Code Coverage and Profiling support and views.
- Heterogeneous (e.g., Arm + RISC-V + ARC) & homogeneous debug & trace support for multi-core SoCs sharing a single debug interface (accessed via probe) is provided.
- On-chip and off-chip trace/debug analytics support.
- ROM or RAM based debugging support (e.g., hardware breakpoints for flash-based support).
- Integrated QEMU ISA simulator for 32-bit & 64-bit cores. Support for other industry standard instruction & cycle accurate simulators.
- High-level register viewer.
- Integrated RTOS (e.g., FreeRTOS or Zephyr) & OS (e.g., Linux) debug support.
- Creation of customisable *RiscFree* user-interface "Views" (Windows) using Python scripts.
- Fully scriptable debug interface using Python or GDB syntax.
- Support for SoC wide breakpoints i.e., a single breakpoint can halt all cores in active debug launches.
- Core dump debugging allows post-crash debugging and analysis using a core dump file.
- **RiscFree** supports core-specific software breakpoints in shared code i.e., only halt if a specified core is running for common code shared between multiple cores.
- RiscFree has a dedicated Global Variables view to browse and show global variables in the application (ELF).

Product	Order Code
RiscFree SDK (supports RISC-V, Arm and Synopsys ARC & ARC-V)	RF-GENERAL
RiscFree SDK (supports MIPS RISC-V ISA P8700 & I8500 CPUs with GCC toolchain)	RF-MIPS

## www.ashling.com





