

Ashling announces *RiscFree*[™] C/C++ SDK support for Lattice RISC-V MCU CPU Soft IP Cores

Limerick, Ireland – July 14, 2023 – <u>Ashling</u> today announced its *RiscFree* SDK has been added to the <u>Lattice Semiconductor</u> RISC-V[®] MC CPU soft IP support ecosystem.

RiscFree is Ashling's SDK including an IDE, compiler and debugger, which now provides software development and debug support for Lattice's RISC-V MCU CPU soft IP. Since its introduction, Ashling's *RiscFree* SDK has been steadily building market share within the embedded tools market and is particularly strong in the RISC-V market where its ease-of-use, broad functionality, plug-in architecture and real-time trace support have made it the go-to-choice for 32-bit and 64-bit RISC core software development.

"We are pleased to announce support for Lattice's RISC-V MCU CPU soft IP cores and that our RiscFree is now part of Lattice's tools eco-system for development and debugging. We're looking forward to enhanced collaboration, particularly between our engineering teams, to ensure that RiscFree is positioned to exploit all additional debugging and analysis capabilities planned for future MCU CPU IP devices" said **Hugh O'Keeffe, CEO of Ashling**.

"Enabling our customers to innovate without constraints is a top priority at Lattice. We are excited to welcome Ashling to our growing partner ecosystem and to open up even more ways for our customers to achieve their design goals with our power efficient FPGAs" said Karl Wachswender, Principal System Architect at Lattice Semiconductor.

Ashling *RiscFree* SDK support for Lattice's MCU CPU soft IP includes:

- IDE with full source & project creation, editing, build & debug support
- **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 & function folding
- Integrated compiler toolchain
- Integrated QEMU ISA simulator with support for other industry standard instruction & cycle accurate simulators
- High-level RISC-V register viewer
- Integrated RTOS (e.g. FreeRTOS or Zephyr) debug support.
- Project wizards, templates & examples

workspace - lattice_riscv_		2							-	o x
File Edit Source Refact	tor Navigate	Search Project Run W	/indow L	Linux Custom Views Help						
i 📑 👻 🖩 👘 🗎 👹 i 👹 i	🖻 🐺 🔍	🕨 II 🔳 🕅 🔍 🖓 .1	ଥ i→ ≡୍	≅ ∞ 🕹 💩 🐐 • 🔕 • 🎥 🛷 • 🍠 🖢 • 🖗 • 🏷	=> 💝	• ⇒ • ⊡			Q	i 🖻 🖬 🛃
🎄 Debug 🗙 🔓 Project E	xplorer	📄 🙀 i> 🕴 🗖		main.c X	• 🗆	1999 Registers $ imes$	- 0	(x)= V 🛛 🗙	ଙ୍କୁ E 📲	🖌 М 🗖 В
✓						1	1 🖂 🗂 🖻		2. 1	
✓ [™] lattice_riscv_mc_cpu.elf				printf("Hello RISC-V World from CR00103!\r\n")	;		/alue	Name	Type	Value
✓ IP Thread #1 (Suspended : Signal : 0:Signal 0)				LED_SET(ALL_OFF);		✓ ₩ General Regist		(×)= ids		1 '\001
main() at main.c:64 0x26e)x()	(4)- 103	unito_t	1 (001
penocd.exe				while (true) {)x25e			
📓 riscv64-unknown-elf-gdb.exe				LED_SET(LED_ON(idx));			x1390			
📰 Disassembly 🗙 💁 Bre	akpoints	-		if (++idx == LED COUNT) {						
Enter location he		8 A 5 B 🖬 🖻	8	idx = 0;			_			
			ö	}	Name : sp		Name :			
00000262: b3 17 fa 00000266: b3 87 f9		a5,s4,a5 a5,s3,a5				Hex:0x1390		Det	ails:1	·\001'
00000268: D3 37 19		a5,1028(s2)				4	Þ	4		Þ
0000026e: 83 47 04		a5,0(s0)		Console 🗙 🖏 Progress 👔 Problems 🕥 Executabl 🔛 Debugger	- 6	3 👖 Memory 🗙	1010 1010 📑 🛒	->t; 🛄	5 6	• 8 - E
00000272: 85 07	add	i a5,a5,1		- X ½ B I P F F - 9	- =	🗸 Monitor 👍 💥 🍇	g 🖳 gpio0_inst: 0x8	400 V -	New Rend	lerings
00000274: 93 f7 f7	Of zex	.b a5,a5	lattic	ice riscv mc cpu [GDB OpenOCD Debugging]		 0x26e 		-		
00000278: e3 9e 97	fc bne	a5,s1,0x254 <m< td=""><td></td><td>vice List:</td><td></td><td><pre>vx20e vx20e v</pre></td><td>Register</td><td></td><td></td><td>Value</td></m<>		vice List:		<pre>vx20e vx20e v</pre>	Register			Value
65		idx = 0;		LFD2NX-40:0x310F104300000000000000000000000000000000	100000	 cpuo_inst gpio0 inst 	✓ ☐ gpio0_inst		00008400	
0000027c: 23 00 04	100 sb	zero,0(s0)		fo : clock speed 1000 kHz			✓ 1818 RD_DAT	-		0x000000FD
00000280: el bf		0x258 <main+88< td=""><td></td><td>fo : JTAG tap: fpga_spinal.bridge tap/device found:</td><td>0x10(</td><td></td><td>0.00 ml = 0.00 ml = 0.</td><td></td><td></td><td>0xFD</td></main+88<>		fo : JTAG tap: fpga_spinal.bridge tap/device found:	0x10(0.00 ml = 0.			0xFD
				fo : Listening on port 3333 for gdb connections arted by GNU MCU Eclipse		> 1010 SET_DA				
Launch Configurations	🔒 Peripherals	× 🖾 🎿 🖻 🖇 🗖		fo : Listening on port 6666 for tcl connections			> 1919 CLEAR			
Peripheral	Address	Description		fo : Listening on port 4444 for telnet connections			> 1010 DIRECT	_		
🔽 🔜 cpu0_inst	0xFFFF0000			fo : accepting 'gdb' connection on tcp/3333			> 1111 INT TYP	-		
	🔒 gpio0_inst 0x00008400		sen	semihosting is enabled			> 1010 INT ME	-		
Result = 200000000000000000000000000000000000				=== VexRiscv registers			> 1111 INT_STA			
) x0 (/32)			> 1010 INT_EN	ABLE_RE 0x	00008420	0x0000000
4) —	<u> </u>			> 1010 INT_SET	REG 0	00008424	0x00000000
							1			

Ashling's RiscFree SDK Debug View

For more information on Ashling's *RiscFree* see: <u>https://www.ashling.com/ashling-riscv/</u> and for details on Lattice's MCU CPU soft IP see:

https://www.latticesemi.com/products/designsoftwareandip/intellectualproperty/ipcore/ipcores04/risc vmccpu

About Ashling

Ashling have been a leading provider of Embedded Development Tools & Services since 1982 with design centres in Limerick Ireland and Kochi India and sales and support offices in Europe, Asia Pacific, the Middle East and America. Visit <u>www.ashling.com</u> for more details.

About RISC-V

The RISC-V open architecture ISA is under the governance of the RISC-V International. Visit <u>https://riscv.org</u> for more details.

Ashling Contact

Contact info@ashling.com

All trademarks, logos and brand names are the property of their respective owners.