STAR-Dundee SpaceWire Engineering Excellence

SPARCV8 SOFTWARE DEVELOPMENT ENVIRONMENT

The SPARCv8 Software Development Environment (SPARCv8-SDE) from STAR-Dundee is designed to support the development of software for the SPARC V8 series of processors, SoC devices and FPGA cores targeted for on-board spacecraft. SPARCv8-SDE consists of an all-in-one software development environment that provides great flexibility allowing generic application software development and, through a library of plugins, processor specific software.

Helping both hardware and software engineers at all stages of system development: evaluation, hardware prototyping, software development and debugging, SPARCv8-SDE provides the essential tools to help reduce development time and cost.

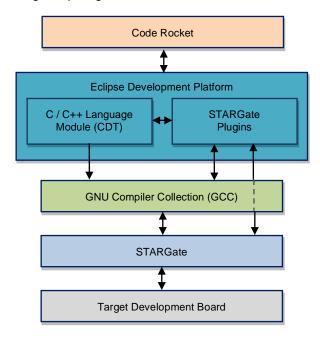
SPARCv8-SDE currently supports three major SPARC V8 based devices:

- **Microchip AT697E/F:** Radiation-Hardened SPARC[®] V8 architecture with throughput approaching 1MIPS per MHz. The AT697F is the latest release developed with ESA support to serve the increasing demand for on-board processing.
- Microchip AT7913E SpW-RTC: Single chip embedded system includes a general purpose LEON2-FT SPARC® V8 core that provides the mixed capability to effectively perform data handling at the platform level and powerful data processing at the payload level.
- Microchip ATF697FF: Radiation-Hardened SPARC[®] V8 architecture with Combined AT697F LEON2-FT 100Mhz Rad Hard Processor + ATF280F SRAM based FPGA (280K) in a single package.

In addition to the software packages, STAR-Dundee provides development boards containing target processors, and a complete range of SpaceWire interface, analysis, emulation, and test and development equipment.

Software Development Environment

SPARCV8-SDE is a suite of tools combining a powerful debugger, editor, code browser, configuration manager, and other tools into one integrated package.



Fully Featured IDE

The Eclipse IDE is used for the SPARCv8-SDE front end. Its wide adoption for software development of embedded systems due to a reputation for software quality and a vast support network, and its extensible nature make it an ideal choice.

Connection between the host computer and the target processor board over any of the range of supported debugging interfaces provides a smooth, responsive debugging experience. Supported interfaces include Debug UART over both USB and serial.

While debugging, Eclipse provides familiar views to the developer: Workspace for project navigation, Editor to allow inspection of code, variables and register values and Console Debug for code stepping.

Edit Source Refactor Navigate Search Project Bun	Window Help								
- 🗟 🗟 📓 😁 - 🕸 - O - 💁 - 💩	🍅 🖉 • [🍠 🗐 • 🖏 🖕 •	ф т				🔁 😳 Debug 🗟 Cit			
Debug IS		00: Variables [2]. *o Breakpoints III Registers mit Modules							
Bart ► 0 ■ 17 3, 9 (c B	1+ 🕱 🔍 Name								
328it Debug (RTC Application)	* 60- success	65- GA7455							
RTC Debugger (36/96/2011 12/21) (Suspended)	ed-start bankt			0					
Thread [0] (Suspended)	to- end bank3								
1 main() 328it.c:38 0x40001224									
sperc-elf-gdb (36/06/2011 12/21)									
and the second second second second									
			(m. 1						
8284.c 15			Cutine III D	Disassembly SS	Enter location here	• 🗞 🏠 🔂 * 🖻			
int main()				mov 1, %g1					
£			40001220:	st %g1, [%fp + -1	2]				
// Initialize success to true			40001224:	sethi %hi(0x402000	00), %g1				
int success = 1/				at \$01. [\$fp + -6					
			600012201	sethi %hi(0x403ffc	00), %g1				
// Initialize start and end memory pos		40001230:	or %g1, 0x3f0, %g1	1 0x403ffff0					
unsigned long start_bank1 = 0x40200000		400012341	at \$01, [\$fp + -4	1					
unsigned long end_bank1 = 0x403FFFF0;		400012381	1d [%fp + -8], %	00					
			4000123c:	1d [%fp + -4], 8	01				
// Test address range for bank 1			400012401	call 0x400011a0 <7	estAddressRange>				
success = TestAddressRange(start_bank1		400032441	500						
			400012481	mov koD, kgl					
// If test succeeded		400012401	at \$01, [\$fp + -1	2 1					
if(success)		400012501	1d [%fp + -12].	901					
(400032541	cmp %g1, 0					
// Output success			400012581	be 0x40001278 <mail< td=""><td>n+96></td><td></td></mail<>	n+96>				
printf("Test Succeeded\n");			4000125c:	500					
)		2	40001260:	seth1 %h1(0x400054	00), %g1				
// Else			400012641	or \$q1, 0x190, \$c0	1 0x40005590	< rodata start>			
else			40001268:	call 0x40001320 <r< td=""><td>uts></td><td></td></r<>	uts>				
(4000126c:	nop					
// Output failure			400012701	b 0x40001288 cmain	+112>				
printf("Test Failed\n");			400012741	500					
1			40001278:	seth1 %h1 (0x400054	00) 8-1				

SPARCv8-SDE provides target specific custom plugins for the Eclipse environment for each supported processor to add views that allow access to all aspects of the target hardware, such as the device internal register view. Register values can quickly be visualised, and individual bits clearly identified and edited graphically.

🖻 🐣 🖻		Conne	cted	 				
Name	Address	Value	-		16 000000000	8 0001011	1111111	Memory Config 1 0x80000000
Memory Config 2 Memory Config 3 AHB Falling Address AHB Status Cache Control Power Down Write Protext 1 Write Protext 2 Leon Configuration Timer 1 Reload	0x80000000 0x80000004 0x80000006 0x8000000C 0x80000010 0x80000014 0x80000018 0x8000001C 0x80000012 0x80000024 0x80000040 0x80000048	272630527 4096 3355443455 2147483660 138 1351712783 0 0 1349688977 4294001757 500000 2						PROM Read Watates = 15. Non watates = value *2 PROM Web Watates = 15. Non watates = value *2 PROM Web ~2 undered PROM Web ~2 undered (0 branke = 0 (0 branke = 0 (0 branke = 0 (0 branke = 0 (0 Web ~2 undered Asysc Bas Ready = 0 PROM Bia Ready = 0

The trace buffer shows operations that have taken place on the hardware. It can be used in conjunction with break points to step through a program when debugging a problem or as a source of post-mortem information when a program has crashed unexpectedly.

æ											Connected
Line	Time	Info	Address	Opcode	Op	Data	Time	Info	Address	Data	*
451	0000001166		0x4000119C	0x92102000	clr	0.00000000					
452							0000001170	R b 0x0 0x1	0x40001710	0x9DE3BFA0	
453	0000001171		0x400011A0	0.694102000	clr	0.000000000					
454							0000001172	R Sb 0x0 0x1	0x40001714	0x91D02001	
455	0000001173		0x400011.A4	0x40000158	call	0:400011A4					
456							0000001174	R Sb 0x0 0x1	0x40001718	0x80100001	
457	0000001175		0x400011.AS	0x01000000	nop	0.00000000					
458							0000001176	R Sb 0x0 0x1	0s4000171C	0x81E80000	
459	0000001178		0x40001710	0x9DE3BFA0	Save	0x40FFFF60					-
460							0000001182	R b 0x0 0x1	0x40001720	0x81C3E008	

Source code views of the trace buffer and Code Rocket views are available as well as the assembly language view, simplifying understanding of the program execution flow and speeding up debugging. Various configuration options are integrated inside Eclipse that provide direct control over hardware settings and compilation options.

reate, manage, and re STAR SDE tools launch co	
Yine filter test yern R SDE Tools TGAR SDE Tools TGAR SDE Tools	Sime STAR SDE Tools Configuration Image: Connection O Smith Protect O Smith Protect O Visit Protect PRoduct Statistics Engine Example Engine Statist Protect PROMW Width Statistics Statist Statistics Statist Connection Example Enging Example Connection Example Enging Example Enging Example Enging Statistics Enging Example Enging
+	Apply Reyert

Code Rocket

Code Rocket, from Rapid Quality Systems Ltd., is a software design tool that provides abstract pseudocode and flowchart visualisations of algorithms. A plugin for the Eclipse IDE allows the developer to design and visualise methods on demand.

De Lift Source Refector Savigate Search Project Ban Window Help					
1-202 1 1- 0-0-0- 00 JU-0-0-00-0	-			E O Debus	2 C/C+
Pseudocode Editor	- 0	Reschart Editor 33			-0-
20124		Assignment O Sub	-Procedure D For D White	Provent	Uvel . Br
Initialize access to true Desilative many and mak access Tert access access the back 1 of each access for a constraint access tert Budg B		00124	main Instalise success to true		7 (0- 0- 22
a teste Il	- 0		Initialise start and		
// Initialise start and end memory manigoed long start bank1 = SEANI2 BANK1 START/	休息		end memory		
<pre>uniqued long end_bask1 = RARD2_RARD2_RARD2_RARD2 //Tors Schere single Sch East //Tors Schere single Sch East //III Cont Purchased (international international)</pre>		no	Test address tange for bank 1 If test succeeded	yes	
// Output success		Output failure		Output success	
printf("Test Succeeded(u");		T	return 0;	T	
Console (In Table) Debume (O Executable (In Debug (I))) (In Table) Debug (I) (In Table) Debug (I) (In Table) Debug (I) (In Table) Debug (II (II (II (In Table) Debug (II (In Table) De			END		

The Code Rocket toolset seamlessly integrates with existing development processes to improve developer productivity and code understanding. When designing new algorithms, boiler plate code is automatically generated from abstract pseudocode and flowcharts. This allows the developer to focus on designing high quality algorithms without being distracted by implementation details which can be filled in later. The abstract design visualisations can be useful when discussing processes with non-technical stakeholders or when developers are working with unfamiliar code. Similarly, the Code Rocket views can help during debugging as they are automatically populated and highlighted when stepping through the code. Automated detailed design documentation can be produced at the touch of a button in both HTML and Word formats, ensuring that documentation is always up-to-date.

GNU Compiler Collection

SPARCV8-SDE uses a version of the industry standard GNU Compiler Collection (GCC) which is tailored for use with LEON2 and LEON2-FT processors, and which supports all peripheral features.

This includes the compilers, linkers and debuggers, and a number of other tools.

A Board Support Package (BSP) is provided to support development of C and C++ programs.

STARGate

STARGate is the platform abstraction layer that provides a suite of tightly coupled plugins to expose interfaces and functionality essential for developers to make efficient use of target device capabilities when implementing complex instrument control applications.

STARGate provides connections over all supported communication interfaces available on the target platform, for example RS232 serial link, USB and SpaceWire.

Applications

SPARCv8-SDE can be used to develop a wide range of applications such as instrument data processing and control software.

It allows development and integration of software and hardware for spacecraft instruments or payloads:

- In early stages, in conjunction with STAR-Dundee development units, when there is little or no access to final hardware.
- When actual hardware (flight or test) becomes available.

SPARCV8-SDE is designed to resolve problems much earlier in the development process and provides for extensive system testing prior to flight platform integration.

Specifications

Software:

• Operates under Windows (Windows 10, 8, 7, Vista, XP).

Plugins:

- Microchip AT697E/F/FF:
 - Views: Register bit level view, Trace buffer, Cache mapping; instruction and data.
- Microchip AT7913E:
 - Views: Register bit level, Trace buffer, Cache mapping: instruction and data.
 - Peripherals: CAN, RS232, PCI, USB, GPIO, 64KByte EDAC protected memory, SpaceWire, FIFO interface, ADC/DAC.

PROM programming support:

Allows programming on-board, non-volatile memory over various board and processor supported interfaces. Supports the following standards: Common Flash Interface (CFI) and Scalable Command Set.

Further Information

For more information about STAR-Dundee development boards see: <u>https://www.star-dundee.com/products/spacewire-rtc</u>

For more information on Microchip radiation hardened processors see:

- https://www.microchip.com/wwwproducts/en/AT697F
- <u>https://www.microchip.com/wwwproducts/en/AT7913E</u>
- <u>https://www.microchip.com/wwwproducts/en/ATF697FF</u>

STAR-Dundee STAR House, 166 Nethergate Dundee, DD1 4EE, Scotland, UK Tel: +44 1382 201 755 Fax: +44 1382 388 838 Email: enquiries@star-dundee.com Web: www.star-dundee.com

STAR-Dundee