PolarFire® SoC CPU Performance Benchmarking

Introduction

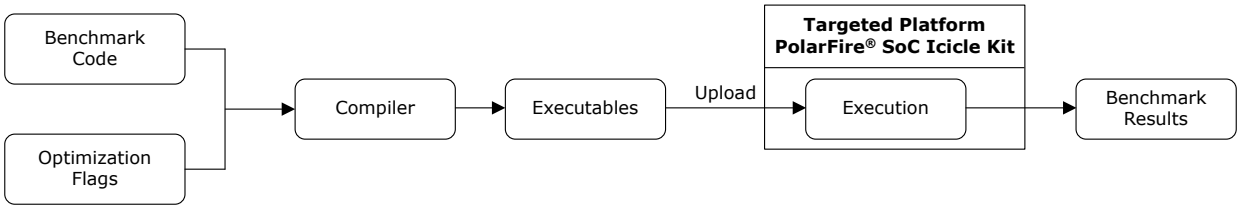
Microchip's PolarFire SoC FPGAs include the industry's RISC-V based Microprocessor Subsystem (MSS) and FPGA fabric that inherits all the features of the PolarFire family. The PolarFire SoC MSS includes 5x 64-bit RISC-V processor cores, AXI Switch, DDR Controller, Fabric Interface Controllers (FIC), and a rich set of peripherals. It also offers an unparalleled combination of low power consumption, thermal efficiency, and defence-grade security for smart, connected systems. It is the first SoC FPGA with a deterministic L2 memory subsystem enabling real-time applications. Built on the award-winning, mid-range, low power PolarFire FPGA architecture, PolarFire SoC devices deliver up to 50% lower power than alternative FPGAs, span from 25k to 460k logic elements, and feature 12.7G transceivers.

PolarFire SoC devices include L1 cache, L2 cache, DDR controller, and eNVM (embedded non-volatile memory) to execute user applications on U54 (Application Processor) cores. The L1 iCache can be used as ITIM (Instruction Tightly Integrated Memory) for code execution. The L2 Cache can be used as LIM (Loosely Integrated Memory) or Scratchpad. The external DDR memory such as LPDDR4 can also be interfaced using the DDR controller. The eNVM can be used for storing instructions and code execution. For more information, see PolarFire SoC FPGA MSS Technical Reference Manual.

This white paper describes CPU performance results based on the industry-standard performance benchmarking suites such as Dhrystone (Bare Metal), CoreMark (Bare Metal and Linux), and CoreMark-PRO (Linux) which are executed on the Application Processor Cores (U54, which uses RISC-V Architecture RV64GC) using the PolarFire SoC Icicle kit. The performance benchmarking suites are compiled with the GCC compiler on the RISC-V platform to evaluate the performance of embedded processors.

The following figure shows the execution methodology for the performance benchmarking suites. These performance suites are executed by applying different levels of optimization flags. All the computations are model-driven runtime (MDR), which uses all the aforementioned flags, trade-offs during the runtime to get the best performance results.

Figure 1. Execution Methodology for the Performance Benchmarking Suites



This white paper provides the summary of results after executing the benchmark applications from LIM, ITIM, eNVM, Scratchpad, and LPDDR4. The following table lists the system configuration common to all the performance benchmark suites.

Table 1. System Configuration

System Configuration	Description
Product and Architecture	Icicle kit, RISC-V
Platform	Dhrystone—Bare Metal
	CoreMark—Bare Metal and Linux
	CoreMark-PRO—Linux
CPU Core Frequency	600 MHz
External Memory Access	LPDDR4
LPDDR4 Frequency	800 MHz
Compiler	GCC
Toolchain for Linux	riscv64-oe-linux-gcc (v11.1.0)
Toolchain for Bare Metal	riscv64-unknown-elf-gcc (v8.3.0)

Attention: The benchmarking results depend on the tool chains, compiler flags such as optimization level, single core, or quad cores.

Table of Contents

1. Dhrystone

The industry standard Dhrystone v2.2 is executed on one of the RISC-V CPU cores of the PolarFire SoC target platform. The Dhrystone suite remains intact with the application even after changing the wrapper code that allows different levels of execution timings on the PolarFire SoC Icicle kit. Dhrystone suite is adapted to get the processor performance, which plays an important role in the non-numeric system programming and emulates the processor usage with the help of certain common set of programs. The benchmarking value shows the iterations of the program executed by the processor at a particular time frame and normalizes to per MHz of CPU clock frequency.

The Dhrystone program gets the comparative performance differences between the external and internal memory settings along with the system settings. The Dhrystone suite features a well-structured code base, which takes full advantage of the wrapper code resulting in optimal biased configuration to get the best performance results.

Following are the key features of Dhrystone:

- Well-suited for embedded CPUs
- Provides great comparison of the compiler performances
- Performs synthetic benchmarks to compare different system architectures
- Has a very small binary footprint, which determines the CPU's benchmarking quickly

For more information about Dhrystone, see en.wikipedia.org/wiki/Dhrystone.

1.1 Dhrystone on Bare Metal

To execute a benchmarking test suite, it must be self-contained to allow its execution on the hardware directly. This is necessary at the start of an SoC design to evaluate and establish correlations of expected integer performance.

1.1.1 Results

The following table lists the Dhrystone benchmarking results for one Application Processor Core U54 running Bare Metal using the PolarFire SoC Icicle Kit.

Table 1-1. Dhrystone Benchmarking Results on Bare Metal

Dhrystone	Bare Metal					
Memory Section	LIM (1.8 MB)	ITIM (28 KB)	eNVM (128 KB)	eNVM (128 KB)	Scratchpad mem (512 KB)	LPDDR4
Code (main loop is run from)	LIM	ITIM	eNVM	eNVM	Scratchpad	LPDDR4
Stack located in	LIM	Scratchpad	LIM	Scratchpad	Scratchpad	LPDDR4
Dhrystone Result (DMIPS/MHz)	**0.58**	**1.58**	**0.47**	**0.975**	**1.714**	**1.714**
Code Size	Full image size is around 70 KB. Dhrystone main loop is less than 2 KB.					Full image size is around 63 KB
Compiler Flags	-pipe -fno-builtin-printf -mexplicit-relocs -fpredictive-commoning -fprofile-report -fstack-usage -fmem-report -flto-report -fipa-pta -fipa-cp-clone -fdevirtualize-speculatively -fdevirtualize-at-ltrans -fira-region=all -fira-hoist-pressure -fira-loop-pressure -fno-stack-limit -fno-inline -fno-common -falign-functions=4 -ffunction-sections -fdata-sections					

2. CoreMark

CoreMark is a benchmarking suite that is intended for the performance of the CPU, when running with specific workloads. The internal CPU architecture is independent of the on-chip memory and peripherals. CoreMark focuses on processor core and cache memory read/write by utilizing the CPU pipeline architecture. The advantage of CoreMark is its size, which allows CoreMark to easily fit in a processor's memory and makes it suitable to get the performance outcome. To evaluate processor performance, CoreMark uses about 20k size of code, which includes I/O computations. All the computations are made at run time. CoreMark mainly focuses on basic read/write operations and integer operations. CoreMark uses realistic workloads. The CoreMark workload comprises of several commonly used algorithms, including:

- Matrix manipulation, to exercise common math operations.
- Linked list manipulation, to exercise the common use of pointers.
- State machine operation, to exercise data-dependent branches.
- Cyclic Redundancy Check (CRC) is a common function in the embedded systems.

The outcome of the CoreMark execution is as follows:

- The four bits of workload tested are matrix manipulation, linked lists, state machines, and CRCs. The output of each stage of the compilers is intended to get the best outcome of the performance results.
- PolarFire SoC device can load the entire test in the cache, DDR, and Flash memory.

The GCC on the RISC-V gives a good performance with -O2 optimization compiler flag and without a compressed extension (RVC).

For more information about CoreMark, see EEMBC.

2.1 CoreMark Benchmarking Results using Signed Index

The CoreMark benchmarking results are captured for one Application Processor Core U54 for both Bare Metal and Linux. The following table lists the CoreMark benchmarking results when signed index is used as loop counter. RISC-V is designed to be more efficient in handling the common case loop index variables in the form of “(signed) int”. Hence, a typedef in the header file was modified to be a signed data type (ee_u32).

Table 2-1. CoreMark Benchmarking Results when Signed Index is used as Loop Counter

CoreMark	Bare Metal						Linux
Memory Section	LIM (1.8 MB)	ITIM (28 KB)	eNVM (128 KB)	eNVM (128 KB)	Scratchpad mem (512 KB)	LPDDR4	LPDDR4
Code (main loop is run from)	LIM	ITIM	eNVM	eNVM	Scratchpad	LPDDR4	LPDDR4
Stack located in	LIM	Scratchpad	LIM	Scratchpad	Scratchpad	LPDDR4	LPDDR4
CoreMark Result (CoreMark/MHz)	**0.950**	**3.128**	**0.950**	**3.030**	**3.128**	**3.128**	**3.125**
Code Size	Full application image size is 70 KB. CoreMark code size is less than 20 KB.					Full image size is around 50 KB	

.....continued		
CoreMark	**Bare Metal**	**Linux**
Compiler Flags for Bare Metal	-Wno-maybe-uninitialized -fno-common -funroll-loops -finline-functions -falign-functions=16 -falign-jumps=4 -falign-loops=4 -finline-limit=1000 -fno-if-conversion2 -fselective-scheduling -fno-tree-dominator-opts	
Compiler Flags for Linux	-O3 -O2 -DPERFORMANCE_RUN=1 -DHAS_STDIO -DHAS_TIME_H -DUSE_CLOCK -fno-common -funroll-loops -finline-functions -falign-functions=16 -falign-jumps=4 -falign-loops=4 -finline-limit=1000 -fno-if-conversion2 -fselective-scheduling -fno-tree-dominator-opts -lpthread -DHAS_FLOAT=0 -mtune=sifive-7-series -lrt	

Note:
CoreMark application chooses heap or stack during its run. Here, it is configured for stack.

2.2 CoreMark Benchmarking Results using Unsigned Index

The following table lists the CoreMark Benchmarking results using default header file (with no modifications to the typedef unsigned int ee_u32).

Table 2-2. CoreMark Benchmarking Results when Unsigned Index is used as Loop Counter

CoreMark	**Bare Metal**						**Linux**
Memory Section	LIM (1.8 MB)	ITIM (28 KB)	eNVM (128 KB)	eNVM (128 KB)	Scratchpad mem (512 KB)	LPDDR4	LPDDR4
Code (main loop is run from)	LIM	ITIM	eNVM	eNVM	Scratchpad	LPDDR4	LPDDR4
Stack located in	LIM	Scratchpad	LIM	Scratchpad	Scratchpad	LPDDR4	LPDDR4
CoreMark Result (CoreMark/MHz)	**0.92**	**2.645**	**0.916**	**2.568**	**2.65**	**2.65**	**2.56**
Code Size	Full application image size is 70 KB. CoreMark code size is less than 20 KB.					Full image size is around 50 KB	
Compiler Flags for Bare Metal	-Wno-maybe-uninitialized -fno-common -funroll-loops -finline-functions -falign-functions=16 -falign-jumps=4 -falign-loops=4 -finline-limit=1000 -fno-if-conversion2 -fselective-scheduling -fno-tree-dominator-opts						
Compiler Flags for Linux	-O3 -O2 -DPERFORMANCE_RUN=1 -DHAS_STDIO -DHAS_TIME_H -DUSE_CLOCK -fno-common -funroll-loops -finline-functions -falign-functions=16 -falign-jumps=4 -falign-loops=4 -finline-limit=1000 -fno-if-conversion2 -fselective-scheduling -fno-tree-dominator-opts -lpthread -DHAS_FLOAT=0 -mtune=sifive-7-series -lrt						

Note:
CoreMark application chooses heap or stack during its run. Here, it is configured for stack.

3. CoreMark-PRO

CoreMark-PRO is a comprehensive, advanced processor benchmark working with CoreMark. When a CoreMark stresses the CPU pipeline, CoreMark-PRO tests the entire processor and memory subsystem, adding support for the following:

- Multi-core and a combination of integer
- Floating-point workloads
- Data sets for using larger memory subsystems

Data sets, depending on the task, ranging from 42 kB to 3 MB, stresses the multi-core devices that the code can launch multiple threads, with parallelism in the workload.

The CoreMark-PRO contains five integer workloads and four popular floating-point workloads. The integer workloads include JPEG compression, ZIP compression, an XML parser, the SHA-256 Secure Hash Algorithm (SHA), and a more memory-intensive version of the original CoreMark. The floating-point workloads include a Fast Fourier Transform (FFT), a linear algebra routine derived from LINPACK, a greatly improved version of the Livermore loops benchmark, and a neural net algorithm to evaluate patterns.

The CoreMark-PRO execution utilizes the MultiBench comprehensive suite to evaluate the performance of PolarFire SoC device. The MultiBench comprehensive suite utilizes Multi Instance Test Harness (MITH), which provides a framework to coordinate scalability analysis. The MultiBench framework evaluates the performance of scalability, Single-core, Multi-core, memory bandwidth, OS scheduling support, and compiler benchmarking.

For more information about CoreMark-PRO, see EEMBC.

3.1 CoreMark-PRO on Linux

To execute the CoreMark-PRO on Linux platform for optimal benchmarking results on PolarFire SoC Icicle kit, run each workload with single/multi workers, configure the number of workers to get the better scaling for each workload. These values are obtained by passing workloads using XCMD flags as XCMD='-c4 -w4 -v1 -i4 ' to the make command, which in turn utilizes the Perl script to get the values as listed in Table 3-1.

3.1.1 Results

The following table lists the CoreMark-PRO benchmarking results using four application U54 cores with L2 cache size configured to 1.5 MB. CoreMark-PRO provides a score (Iterations/Sec) for Multi-core and Single-core with a scaling factor. The scaling factor depends on the workload. For more information about the workloads and scaling factor, see EEMBC CoreMark-PRO User Guide.

Table 3-1. CoreMark-PRO Benchmarking Results on Linux

Workload	Multi-Core (Iterations/sec)	Single-core (Iterations/sec)	Scaling
core	0.35	0.09	3.89
loops-all-mid-10K-sp	0.65	0.20	3.25
nnet_test	0.93	0.28	3.32
parser-125k	8.47	2.34	3.62
zip-test	21.74	5.99	3.63
linear_alg-mid-100x100-sp	18.66	4.85	3.85
sha-test	26.53	8.00	3.32
radix2-big-64k	42.93	20.48	2.10
cjpeg-rose7-preset	47.17	12.21	3.86
Results	**802.42**	**237.68**	**3.38**

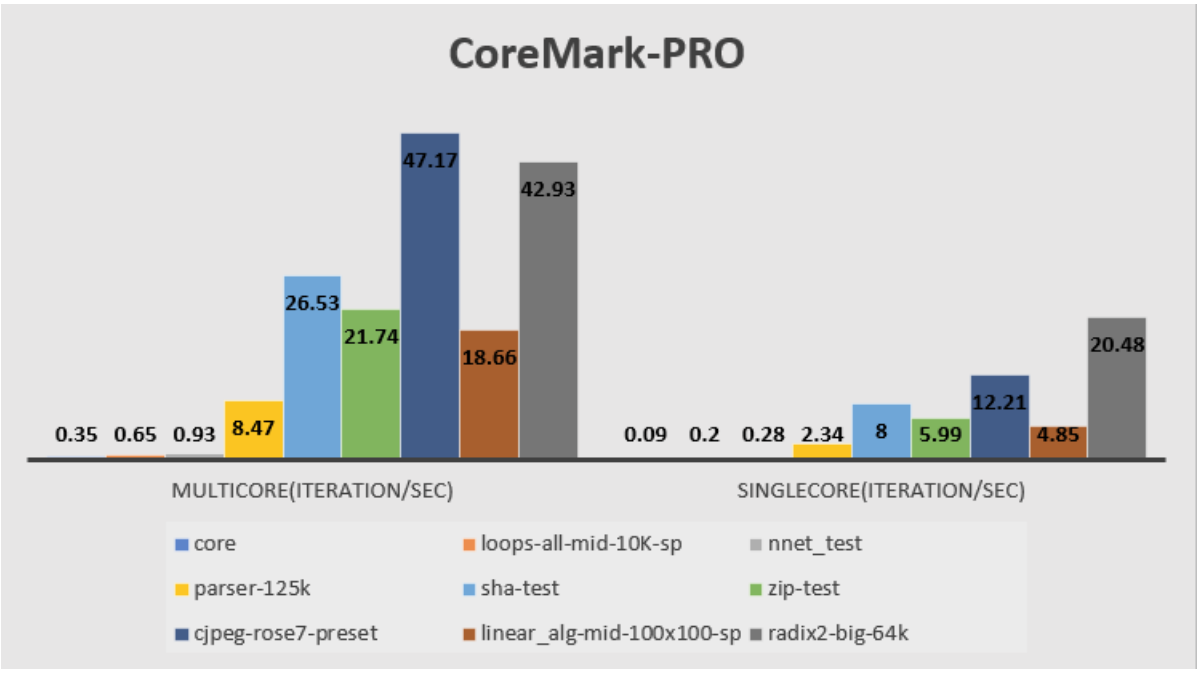
The following figure shows the CoreMark-PRO execution.

Figure 3-1. CoreMark-PRO Execution

```
WORKLOAD RESULTS TABLE

                                                  MultiCore SingleCore
Workload Name                                     (iter/s)  (iter/s)    Scaling
------------------------------------------------- --------- ---------- ---------
cjpeg-rose7-preset                                   47.17      12.21       3.86
core                                                  0.35       0.09       3.89
linear_alg-mid-100x100-sp                            18.66       4.85       3.85
loops-all-mid-10k-sp                                  0.65       0.20       3.25
nnet_test                                             0.93       0.28       3.32
parser-125k                                           8.47       2.34       3.62
radix2-big-64k                                       42.93      20.48       2.10
sha-test                                             26.53       8.00       3.32
zip-test                                             21.74       5.99       3.63

MARK RESULTS TABLE

Mark Name                                         MultiCore SingleCore    Scaling
------------------------------------------------- --------- ---------- ---------
CoreMark-PRO                                        802.42     237.68       3.38
make[2]: Leaving directory '/home/root/coremark-pro'
make[1]: Leaving directory '/home/root/coremark-pro'
root@icicle-kit-es:~/coremark-pro#
root@icicle-kit-es:~/coremark-pro#
root@icicle-kit-es:~/coremark-pro#
root@icicle-kit-es:~/coremark-pro# lscpu
Architecture:        riscv64
Byte Order:          Little Endian
CPU(s):              4
On-line CPU(s) list: 0-3
Thread(s) per core:  4
Core(s) per socket:  1
Socket(s):           1
CPU MHz:             0.00
L1d cache:           32 KiB
L1i cache:           32 KiB
root@icicle-kit-es:~/coremark-pro#
root@icicle-kit-es:~/coremark-pro#
root@icicle-kit-es:~/coremark-pro#
root@icicle-kit-es:~/coremark-pro#
```

The following figure shows the representation of CoreMark-PRO benchmarking results.

Figure 3-2. Comparison Between Mutli-core and Single-core



4. Summary of PolarFire SoC Benchmarking Results

The following table lists the summary of results of benchmarking suites. The best performance results are achieved by running the application code from the Scratchpad or LPDDR4.

Table 4-1. Summary of PolarFire SoC Benchmarking Results

Benchmarks	Platform	Result
Dhrystone	Bare Metal	1.714 DMIPS/MHz
CoreMark	Bare Metal	3.128 CoreMarks/MHz
	Linux	3.125 CoreMarks/MHz
CoreMark-PRO	Linux	Multi-Core 802.42 (Iterations/sec)
		Single-Core 237.68 (Iterations/sec)

5. Conclusion

This white paper describes the toolchain, compiler and optimization flags, and the process for running benchmarks such as Dhrystone, CoreMark, and CoreMark-PRO using the PolarFire SoC Icicle kit.

The best performance results are achieved when code is executed from L2 Scratchpad memory or LPDDR4 memory. When executing applications from LIM, the instructions and data are not cached. LIM is intended for executing applications with deterministic behavior.

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-8845-3

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820