SCORE

***S**afety **C**ritical, **O**bject-oriented **R**eal-time **E**mbedded*

FEATURE SHEET

DDC-I

Safety Critical Software Solutions
for Mission Critical Systems

Superior Features

Key Features of the SCORE® / OpenArbor™ Product Line

- Tight code generation on 1750A, C4x & 80x86
  - Smaller code size
- Less code elaboration vs. GNAT
  - Faster application start-ups (see detail below)
- Selective Ada linking
  - Removes significant percentage of Ada run-time and root library code for **smaller executable**
- No dynamic or self modifying code
  - GNAT is less safe than SCORE (below)

KEY COMPONENTS

- Ada-95 compiler
- C-90 compiler with some C-99 extensions
- Embedded C++ compiler
- Fortran-77 compiler
- Mixed Language Debugger
- Targets: PowerPC, 80x86, Pentium, Mil-STD-1750A, Texas Instruments C4x, native
- GNU assemblers and linkers
- Other utilities (disassemblers, archivers, etc)
- Run-time libraries for all languages

Flexible Coding for Cost Savings

Less Code Elaboration

The SCORE Ada root library contains next to no library unit elaboration code; GNAT is equally good at a number of units, but there are also language constructs (such as tagged types) where GNAT has elaboration code and SCORE doesn't (due to clever handling In the Ada linker). SCORE's approach is best because it reduces code size and allows selective linking to do a better job. The elaboration code must be executed before the main program can start. The more elaboration code that must be executed, the longer the period of time before the program actually starts performing its intended functions.

No Dynamic or Self Modifying Code

In connection with nested subprograms GNAT creates executable code on the stack. SCORE never uses dynamic or self-modifying code. The GNAT approach requires the stack to be mapped as executable (against the spirit of e.g. ARINC 653 and other coding standards) and makes it more difficult to document code coverage. GNAT's approach is inherently unsafe and unsuited for safety critical applications.

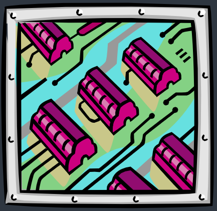
More Key Features...

- SCORE Ada code is more multi-thread safe than GNAT
    - No global variables (see below)
- SCORE C run-time is compatible with GCC
- SCORE Ada uses an Ada library database
    - GNAT creates 2 files from compilation
    - SCORE approach is more efficient
- Debugging of optimized code
    - No need to turn optimizations off to debug
- Optimization always on!
    - The optimized code is continually tested
- Enabling DWARF debug information generation does not change executable code
- SCORE includes integrated simulators for PPC, 1750A and C4x
- VectorCast integration for module testing

Multi-Thread Safe

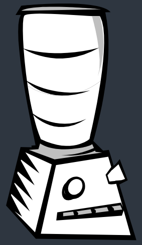
In a multi-thread environment, global variables are at risk of being modified by more than one thread. Applications (and run-times) have to be designed with this possibility in mind. In a multi-thread environment, an analysis must be performed to ensure that no other thread might tamper with a critical piece of global data. The lack of global variables in the SCORE run-time means that the user does not need to worry (and ask for an analysis) of our run-time. The SCORE root library contains almost no global variables. It would appear that GNAT depends on a little more variable data (e.g. for the tagged types mentioned above).

SCORE's approach eases documentation and testing, and ensures multi-thread safety in the vendor code (with GNAT's code it may be more difficult to prove safety). Of course, a programmer can use global variables, for example thread-A periodically reads a global variable that thread-B writes to. However, if thread-C were to write to the same global variable, then it may be a problem since it may be significant where the data came from or the two threads might interfere with one another. User's have to handle this situation or prevent it from happening.



BARE BOARD SUPPORT

Configurable run-times support the language environment on targets with no real-time operating system (RTOS).



MIXED LANGUAGE SUPPORT

You can compile and debug any or all of the following languages:
Ada, C Embedded C++ and/or Fortran.

The DDC-I Advantage

With over 30 years of experience supplying complex COTS and custom development solutions, DDC-I provides long-term, strategic advantages to an ever changing safety critical industry. Boasting founding and committed management, our staff offers a level of experience and service that is unmatched in the industry. Our customers have direct access to DDC-I's highly experienced engineers, which ensures the success of our customer's development efforts.

4545 E. Shea Blvd. #210
Phoenix, AZ 85028
P 602-275-7172
F 602-252-6054
www.ddci.com

DDC-I
Safety Critical Software Solutions
for Mission Critical Systems