Contacts: Nancy B. Green
The William Baldwin Group
nbgreen@william-baldwin.com
+1 650-856-6192

Don Parkman
Cypress
dsp@cypress.com
+108-943-4885

Peter Riachi
HI-TECH Software
priachi@htsoft.com
+61 7 3722 7777

FOR RELEASE OCTOBER 2, 2007

NEW COMPILER TECHNOLOGY EFFECTIVELY DOUBLES PSoC® DEVICE MEMORY CAPACITY

Compiler Cuts Program Code Size as Much as 50%, Improves SRAM Utilization, Prevents Stack Overflows and Increases Execution Speed

San Jose, CA and Acacia Ridge, Queensland, Australia, Oct 2, 2007. . . Cypress Semiconductor Corporation and HI-TECH Software today announced new compiler technology that extends the memory capacity and performance of the dynamically configurable PSoC® mixed signal arrays. The new ANSI C compiler, "HI-TECH C PRO for the PSoC Mixed-Signal Array," exploits HI-TECH's "Omniscient Code Generation" (OCG) technology to radically reduce PSoC code size.

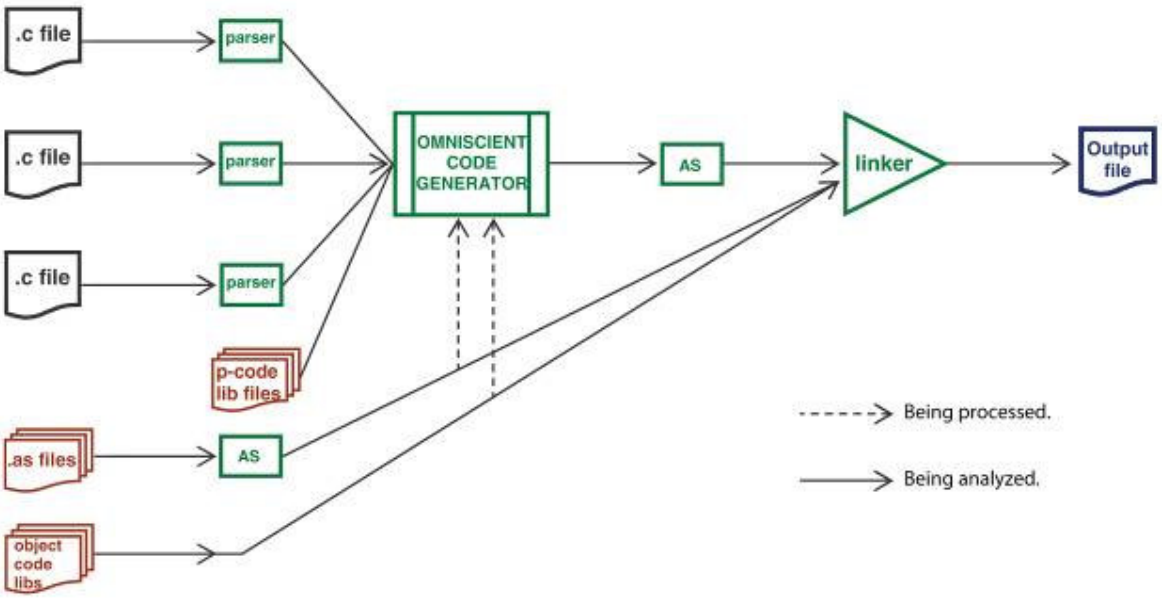
PSoC mixed-signal arrays integrate programmable analog and digital functionality with an 8-bit MCU core, up to 32 Kbytes of flash memory and up to 2 Kbytes of SRAM. They are widely used in cost-sensitive, space-limited consumer applications that include touch screen interfaces, motor control, and proximity detection, among others. PSoC devices can be dynamically reconfigured to execute multiple independent functions in the same silicon, thereby reducing component count, board space, and power consumption.

Although adding reconfigurable functionality to the PSoC device does not increase the amount of silicon required, each reconfigurable function requires additional program code. In some applications, adding this additional functionality causes the program code to outgrow the on-chip flash memory. In addition, the larger software stack and variable requirements may push the limits of the on-chip SRAM, increasing the potential for undetected stack overflows. Until now, the only solutions to SRAM and flash memory constraints have been to: 1) limit the functionality of the end-product, 2) migrate the application to a more expensive PSoC device with more SRAM and flash memory, or 3) hand craft assembly language code to reduce the program, stack and variable sizes -- an exceptionally cumbersome and time-consuming task that severely restricts the portability of the program code. None of these alternatives is very attractive.

The HI-TECH C PRO complier for the PSoC Mixed-Signal Array uses the OCG technology to examine all program modules prior to compilation, optimizing pointers, registers and stack allocation, as well as eliminating redundant code. The compiler also frees up SRAM and reduces contention for the PSoC device's index register by compiling directly addressable, optimized function stacks for all non-recursive and non-re-entrant code. Since small code size executes in fewer cycles, the new compiler also increases PSoC device performance.

Unlike compilers that claim "global optimization" while operating only on individual program modules, Omniscient Code Generation, looks at every module in the entire program and optimizes across all program modules.

OCG whole-program compilation



PSoC device C language code, compiled using the OCG technology, is as much as 50% smaller than produced by competing compilers, effectively doubling the amount of program code that can be stored in the on-chip flash of any PSoC device. Since there is less code to execute, performance is also improved.

Comparative code sizes (bytes) for ts057.c				
Target Chip	**Compiler**	**OCG**	**Code Size**	**Δ%**
Cypress PSoC	ImageCraft	no	11957	
	PRO for the PSoC Mixed-Signal Array	yes	6129	-51.3%

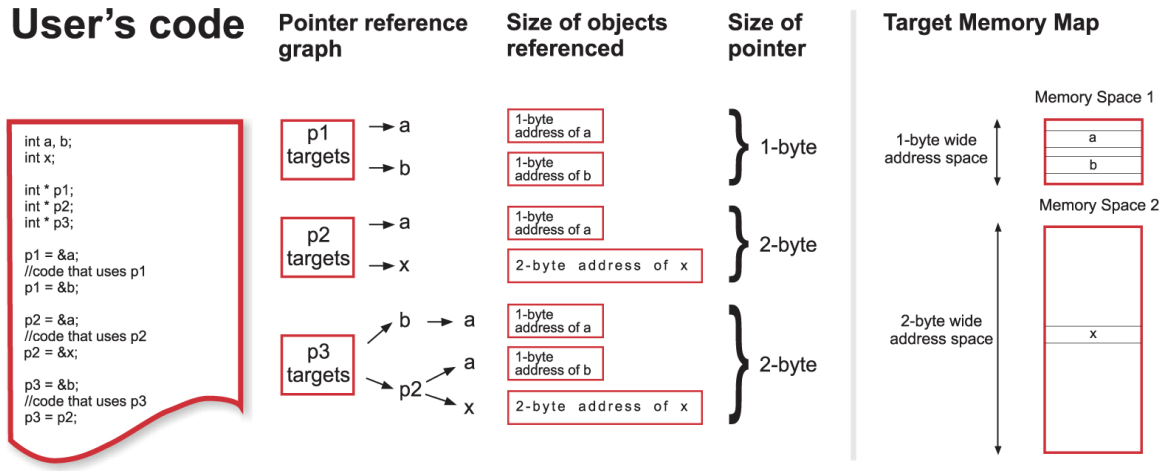
SRAM utilization is more efficient because OCG knows and allocates exactly the amount of memory required by variables and compiled stacks, even determining the range of pointer variables and allocating single-byte pointers whenever possible. In doing so the compiler

increases the available SRAM, sometimes by a significant percentage. In addition, by statically setting the stack to the maximum depth required for the program, OCG prevents potentially catastrophic stack overflows.

Call Graphs and Pointer Reference Graphs Based on All Program Modules. In preparation for compilation, the OCG Compiler creates partially compiled code libraries for every module in the program. It then searches all of these for all library functions that are referenced anywhere in the program and creates a "call graph". Once the call graph is complete, any functions that are never called are removed, and the compiler identifies and marks any functions that are called re-entrantly, such as those from both main-line code and interrupt functions.

The OCG compiler also creates a pointer reference graph of every pointer in every program module. Since the compiler knows the size of every object addressed by any pointer, it can detect any accesses past the end of a statically allocated object and alert the programmer to a possible invalid memory access. Pointers that are never initialized are detected and appropriate warnings issued at compile time. In addition the compiler provides the developer with a valuable debugging aid, a report showing all pointers and their targets.

The compiler also identifies incompatible declarations of variables or objects that exist in different source files and alerts the user.



Frees-up SRAM by Optimizing Variables and Stacks. Conventional compilers generally make worst case assumptions about the sizes of variables. This can lead to the under utilization of SRAM resources. Since the compiler knows the size of all variables, it allocates only the amount of memory required to store each one, freeing up 10% or more of the available SRAM, depending on the application.

Like many low-cost 8-bit programmable devices, the PSoC mixed-signal array uses the same SRAM space for both software function stacks and data variables. If insufficient space in the

SRAM is allocated to accommodate the maximum depth of the dynamic stack, stack can overflow into data variable space which could cause the program to crash.

Most functions are non-re-entrant and non-recursive and may be implemented with a more predictable, static compiled stack. The compiler's OCG technology looks at all the program modules, identifies all non-re-entrant and non-recursive functions, and compiles an optimally sized function stack with exactly enough memory to accommodate each function's maximum depth. Because the call graph for all functions has already been determined, functions with are executed at different times can share the same SRAM space for their static compiled stack. This feature reduces stack space to the absolute minimum required and frees up more SRAM for data. A compiled static stack also reduces the likelihood of stack overflows that occur when a dynamic stack expands into the data variables' space in the SRAM.

Re-entrant functions must be handled in a different fashion which includes assigning them to dynamic stack space for storage of local variables, or managing the re-entrant calls so they cannot overwrite existing data. The OCG compiler achieves this in a way that is completely transparent to the developer, and without requiring non-standard extensions to the language.

Eliminates Contention in PSoC Index Register. Accessing data that is already on the PSoC device's dynamic stack requires that the contents of the stack pointer, which is used solely to push and pop registers, be transferred to the index register, which is then used to access the data. Since PSoC devices contain only one index register, the dynamic stack creates contention for this scarce resource. In addition, if the stack pointer changes or the index register has to be reused for any other purpose, the stack pointer will have to be saved and then re-copied to the index register. With between 4 and 10 clock cycles required each time the pointer is copied from or written to the index register, an application with frequent accesses to the dynamic stack, could end up with much more program code than is actually required for system functionality.

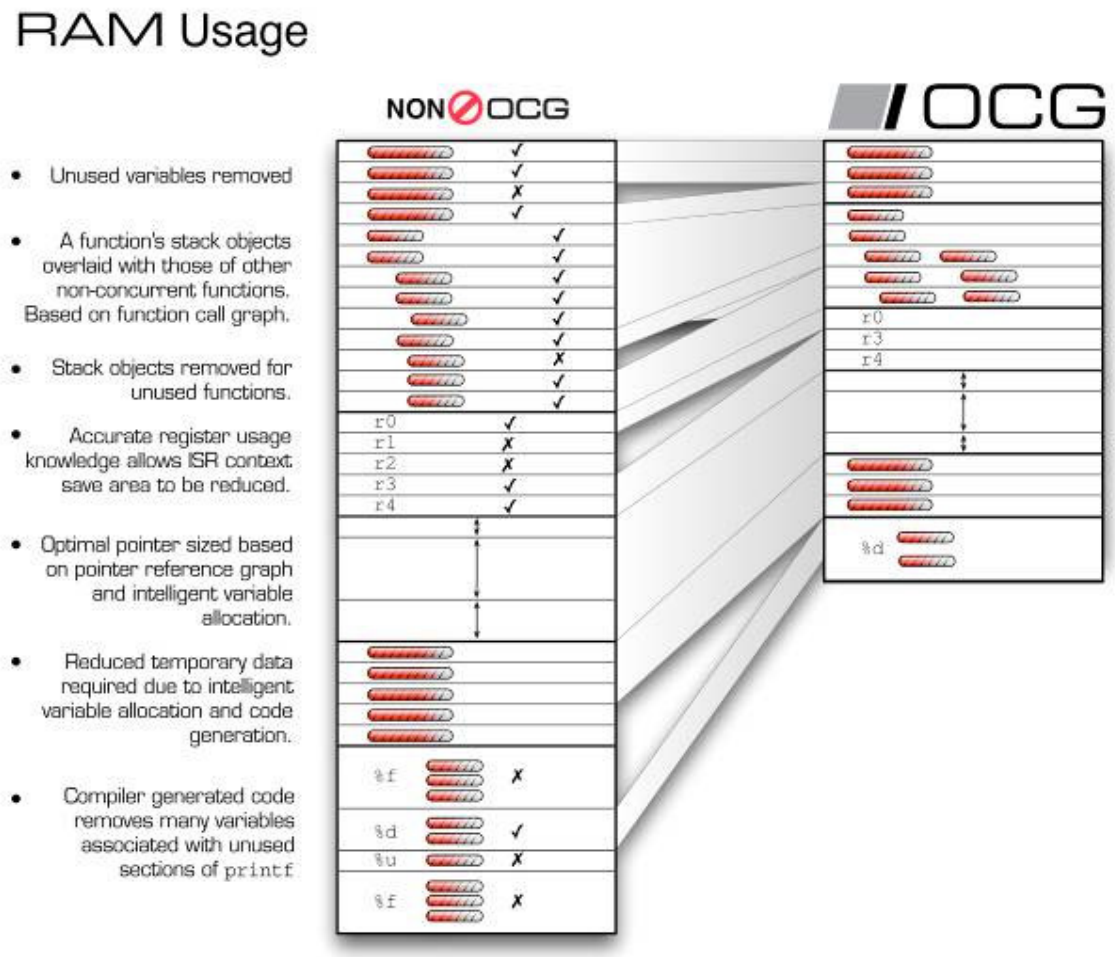
Since the static software stack compiled is directly addressable, without using the stack pointer or index register, both code size and the number of cycles required to execute the program are reduced. With the OCG compiler the conventional stack can be reserved for the small number of re-entrant or recursive functions in a program. With this Compiler the developer avoids using the PSoC device's dynamic stack an average of 99% of the time, eliminating most of the redundant code required to manage a dynamic stack.

Optimizes PSoC Memory Usage. The PSoC mixed-signal array has a paged SRAM architecture, in which only 256 bytes of SRAM are addressable at any one time. Accessing any other memory page requires the page select register (PSR) to be reset. Each PSR reset takes three bytes of code and 12 cycles to execute. If a data from a page that is already in use (e.g. for an interrupt routine) must be written to another memory page, additional program code and more clock cycles

are required. For example, Page0 is automatically selected for Interrupt routines. If the interrupt routine requires access to a variable in any page other than Page0, the PSR must be saved, the memory access mode changed, and the PSR loaded with the other page address. Afterwards the PSR must be restored to its state prior to the interrupt.

Each time this happens, a total of 12-Bytes of program code and 50 clock cycles are added to program execution. In an extreme case, sub-optimal allocation of variables to the memory pages could easily double the program size *and* the number of cycles required to execute the routine.

Omniscient Code Generation technology determines all the variables being used, when they are used and the context in which they are used. It can therefore allocate data variables to the SRAM memory pages to minimize the number of PSR updates required. For example, the compiler allocates variables accessed from interrupt routines in an SRAM page that is constantly addressable during the interrupt, rather than one which requires a page select register update before access.



OCG compilation algorithms take into account each and every instance of a variable, plus each instance of an assignment of a pointer value to a pointer (either directly, via function return,

function parameter passing, or indirectly via another pointer) and builds a data reference graph, known as a Pointer Reference Graph. When complete, the pointer reference graph identifies the set of *all* objects that can possibly be referenced by each pointer. This information is used to determine which memory area and page each pointer will be required to access. Any conflicting declarations of the same object from different modules can be detected and an informative error message issued to the user. Any variables never referenced can also be determined and deleted.

The compiler also defines a set of address spaces for each pointer variable that is optimally efficient for the PSoC, without any specific direction from the programmer.

HI-TECH's OCG-enabled compiler has intelligence about the complete set of used variables and pointers across all program modules *and* their frequency of use. It also knows exactly how big the stack must be and where it will be located before the code is generated. It allocates the most frequently used variables into the most readily accessible RAM and puts less frequently used data into the paged RAM. This results in improved code density by eliminating the need for the program code to set a page select register, and can achieve substantial performance improvements.

Seamless Integration with PSoC Designer™v4.4. HI-TECH's Compiler with OCG technology has been seamlessly integrated into Cypress' PSoC Designer integrated development environment (IDE) for PSoC mixed-signal arrays.

Demo download and an extended evaluation period. A fully functional 45 day trial version of the Compiler can be downloaded, free of charge, at HI-TECH's website http://www.cypress.htsoft.com.

Price and availability. The Compiler is available now for $1,195.00 U.S. if purchased before 31 March 2008, after which it will be $1,495 U.S.. It includes, at no extra cost, HI-TECH Software's 12 month access to updates and technical support, as well as a 30 day money back guarantee. Multi-user and educational user discounts are available.

###

About HI-TECH Software. HI-TECH Software is a world class provider of development tools for embedded systems, offering compilers, RTOS and an Eclipse based IDE (HI-TIDE) for 8-, 16-, and 32-bit microcontroller and DSP chip architectures.

Founded by Clyde Stubbs, in 1984 in Brisbane, Australia, HI-TECH Software has an extensive network of distributors around the globe.

About the PSoC Family

PSoC devices are configurable mixed signal arrays that integrate a fast 8-bit microcontroller with many peripheral functions typically found in an embedded design." PSoC devices provide the advantages of an ASIC without the ASIC NRE or turn-around time. A single PSoC device can integrate as many as 100 peripheral functions with a microcontroller, saving customers design time, board space and power consumption. Customers can save from 5 cents to as much as $10 in system costs. Easy to use development tools enable designers to select configurable library elements to provide analog functions such as amplifiers, ADCs, DACs, filters and comparators and digital functions such as timers, counters, PWMs, SPI and UARTs. The PSoC family's analog features include rail-to-rail inputs, programmable gain amplifiers and up to 14-bit ADCs with exceptionally low noise, input leakage and voltage offset. PSoC devices include up to 32KB of Flash memory, 2KB of SRAM, an 8x8 multiplier with 32-bit accumulator, power and sleep monitoring circuits, and hardware I2C communications.

All PSoC devices are dynamically reconfigurable, enabling designers to create new system functions on-the-fly. Designers can achieve far greater than 100 percent utilization of the die, in many cases, by reconfiguring the same silicon for different functions at different times. Learn more about PSoC products at www.cypress.com and receive free online training at www.cypress.com/psoctraining.

About Cypress

Cypress delivers high-performance, mixed-signal, programmable solutions that provide customers with rapid time-to-market and exceptional system value. Cypress offerings include the PSoC® mixed-signal array, USB controllers, general-purpose programmable clocks and memories. Cypress also offers wired and wireless connectivity solutions ranging from its WirelessUSB™ radio system-on-chip, to West Bridge™ and EZ-USB® FX2LP controllers that enhance connectivity and performance in multimedia handsets. Cypress serves numerous markets including consumer, computation, data communications, automotive, industrial, and solar power. Cypress trades on the NYSE under the ticker symbol CY. Visit Cypress online at www.cypress.com.

# # #