Cahners
MICROPROCESSOR REPORT
www.MPRonline.com
THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

INFINEON TARGETS 3G WITH CARMEL2000

Application-Specific PowerPlugs Accelerate Carmel Core

Jennifer Eyre and Jeff Bier, BDTI {7/17/00-02}

Processor vendors everywhere want a piece of the third-generation ("3G") wireless market, which many expect to be the next DSP "killer app." Today's 2G digital cellular phones are consuming hundreds of millions of 16-bit DSPs and 32-bit MCUs. Third-generation devices, however, will have performance requirements far beyond the capability of even today's fastest DSPs, including application-layer tasks like video decompression. And, of course, this blazing performance will have to be matched with extremely strong energy efficiency and reasonable cost. At **Embedded Processor Forum 2000**, Amnon Rom, Vice President of VLSI Design at Infineon, explained how it will upgrade its Carmel DSP core to address the demands of 3G wireless applications through the addition of tightly coupled application-specific hardware accelerators.

Carmel is a VLIW-based, 16-bit fixed-point DSP core introduced by Infineon (formerly Siemens Semiconductor) in 1998 (see *MPR 12/28/98-04*, "Carmel Enables Customizable DSP"). The processor uses a mixed-width 24/48-bit instruction set and can execute one 48-bit instruction, one 24-bit instruction, or two 24-bit instructions per cycle. In addition, Carmel allows the user to combine up to six standard instructions as part of a single configurable long instruction word, or CLIW. Each CLIW instruction is stored only once, in a small, dedicated memory. The advantage of this approach is that you can get high parallelism where you need it (in speed-critical inner loops) without paying the penalty of high program memory use everywhere else—a price often exacted by other VLIW architectures. Keeping memory use in hand is particularly important in portable applications like 3G wireless, since extra memory is costly in terms of both space and power. In addition to using the core in its own application-specific standard products, Infineon offers the design for license to OEMs and other semiconductor vendors.

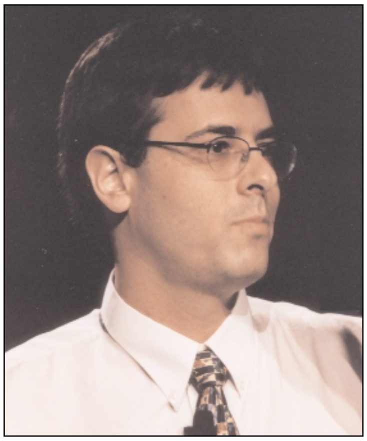
With Carmel2000, Infineon takes the concept of "parallelism where you need it" a step further, allowing SoC designers to create hardware accelerators that can be tightly integrated with the basic Carmel architecture to speed up execution of inner loops even more. These accelerators, called PowerPlugs, can be thought of as an extension of the CLIW concept; in addition to configurable instructions, you can customize the core itself to allow more parallelism where needed. This approach blurs the line between custom hardware and programmable processors in an attempt to combine the best of both worlds, and Infineon is not alone in exploring this avenue. Indeed, both ARC and Tensilica allow designers even more latitude in customizing their processor cores (see *MPR 7/8/96-03*, "ARC Core Gives ASICs Programmability" and *MPR 3/8/99-02*, "Tensilica CPU Bends to Designers' Will").

Carmel's PowerPlugs are closely integrated with the underlying Carmel core, which views them as additional execution units in the datapaths. PowerPlugs can also include their own registers and memory, which are accessed using the base core's memory buses. Unlike for multicore or coprocessor solutions, an SoC designer using Carmel2000 does not have to define interfaces or figure out how to pass data to the PowerPlugs; these interfaces are predefined. The base Carmel core handles data address generation, wait states, and data memory read/write controls.

For More Information

For more information check out Infineon's Web site at: *www.carmeldsp.com*.

PowerPlugs are defined at the register-transfer level and can be implemented and integrated with the core via logic synthesis or physical design. Infineon hinted that it may offer a library of predefined PowerPlugs, but for the moment it is committing only to offering a MAC PowerPlug that is identical to the MAC units on the base Carmel core. SoC designers can create their own application-specific PowerPlugs, using RTL to define the hardware and a C description of the PowerPlug instruction functionality (for simulation). For example, Infineon envisions MPEG-4 PowerPlugs that can enable real-time MPEG video decoding, as well as Viterbi PowerPlugs to accelerate physical-layer processing for wireless communications.


MICHAEL MUSTACCHI

Amnon Rom, VP of VLSI Design at Infineon, describes Carmel2000's PowerPlugs at the Forum.

The original Carmel core has been renamed Carmel1000 to distinguish it from the new core. The key differences between Carmel1000 and Carmel2000 are a doubling of CLIW memory size (from 1K x 96 to 2K x 96), some additional shadow registers, and, most significantly, hooks for PowerPlugs. Both base Carmel cores have two datapaths, each with a MAC unit and an ALU. One of the two datapaths also has a shifter and exponent unit. CLIW instructions for Carmel are positional and take the form:

```
cliw name (op1, op2, op3, op4)      // cliw ref and operands
{
 MAC || ALU || MAC || ALU || Mov || Mov // cliw definition
}
```

where the double bar indicates that the instructions will be executed in parallel. The 48-bit reference line is stored in program memory each time the CLIW instruction is invoked, but the 96-bit definition is stored only once.

In Carmel2000, a PowerPlug can be activated using any one of the first four slots in the CLIW word. Thus, the total number of instructions that can be specified in parallel is not increased, but the programmer can substitute a PowerPlug instruction for one of the MAC or ALU instructions where needed to increase performance. For example, by adding two MAC PowerPlugs (a configuration Infineon will implement in a Carmel2000 development chip), a CLIW instruction can implement a quad-MAC block FIR filter inner loop as follows:

```
cliw blkfir_innerloop (r0– –, r1– –, r4++)
 {
   a0 += *ma3 * *ma1
  || plug1 pa0 += *ma3 * *ma2
  || a1 += *ma3 * ff1
  || plug3 pa0 += *ma3 * pff1, pff1 = *ma2
  || ff1 = *ma1
 }
```

where ma1:ma3 are registers specified using the operand list in the CLIW reference line (in this case, registers r0, r1, and r4). In this example, the two ALU instructions have been replaced by PowerPlug MAC instructions, indicated by plug1 and plug3. PowerPlugs can be activated only by CLIWs.

As might be expected from the example above, Carmel2000 is fully backward compatible with Carmel1000. Infineon claims that Carmel development tools, including the compiler, will provide full support for PowerPlug modules. It is unlikely, however, that the compiler will be able to generate CLIWs that use custom PowerPlugs.

Infineon expects the first Carmel2000 development chip to be available in the first quarter of 2001. Including the two additional MAC units, the core is less than 4mm^2 in 0.18 micron, according to Infineon. Carmel is expected to execute at up to 300MHz in this fabrication process. If the development chip achieves 300MHz, it will likely provide DSP performance comparable to the 300MHz quad-MAC StarCore SC140 but less than that of TI's recently announced 'C64xx executing at its initial target clock speed of 600MHz. TI isn't targeting 3G appliances with the 'C64xx, however; for such applications TI intends to use its more energy efficient 'C55xx core. Also at EPF, Motorola announced its plan to offer a chip for 3G wireless appliances based on the SC140 core.

Interestingly, although Infineon's introduction of Carmel2000 focused on the architecture's suitability for 3G wireless, representatives from Infineon, TI, and Motorola all acknowledged at the recent Forum that even their latest cores cannot provide adequate performance to implement all the features of a 3G appliance without the help of additional hardware accelerators. And while PowerPlugs can provide a significant performance boost, 3G requires more than just computational power—it also requires extremely high data bandwidth. Carmel2000's data bandwidth is constrained by the memory architecture of the original core. Although Carmel's data bandwidth is high compared with that of most other DSP processors (Carmel can read four independent 16-bit data words/cycle), it will likely prove insufficient for the bandwidth demands of a full 3G application. A Carmel2000 with PowerPlugs that contain their own memory will be able to make more efficient use of that bandwidth, but it most likely won't be able to meet the roughly order-of-magnitude increase required by 3G. So, although Carmel2000 is a step closer to a single-core 3G solution, it appears we will have to wait a while longer before we see processors capable of implementing a complete 3G terminal without the aid of powerful off-core coprocessors. ◇