Convey White Paper

The Convey HC-1™ Computer

Architecture Overview

CONVEY computer™

The Convey HC-1 Computer

Architecture Overview

Contents

Introduction

For decades, the evolution of computer systems has been driven by the exponential increase in logic density predicted by Moore's Law. Performance has increased exponentially as clock rates increased, and soaring transistor counts are utilized in a wide variety of architectural innovations to increase performance per clock.

However, in recent years performance has begun to stagnate as power density—caused by increasing system complexity and increasing clock frequency—has become the limiting factor in design. In effect, the laws of physics have created a heat and power brick wall that is nearly impossible to circumvent with traditional processor and semiconductor architectures.

Many companies have turned to asymmetric or hybrid computing architectures that combine industry standard processors with specialized processors that focus on specific operations. Typically, these specific operations are those that represent a large component of an application. But specialized add-on processors are generally not tightly integrated with the host processors, presenting different architectural models that are notoriously difficult to program.

Convey hybrid-core computers overcome these challenges by tightly integrating an FPGA-based, reconfigurable coprocessor with an industry standard Intel® 64 processor. The coprocessor shares memory with the commodity processor, and can be dynamically reloaded with different instruction sets, called "personalities," targeted at specific workloads. A unified C/C++ and Fortran development environment generates both x86 and coprocessor code, allowing a single executable to utilize both elements.

The ability to support different instruction sets in a common hardware platform allows the implementation of new instruction sets in months instead of the years required to design and introduce a new microprocessor. The huge reduction in implementation time makes it practical to develop instruction sets tailored to specific applications and algorithms. Convey is developing a series of personalities for key application areas such as search/indexing, signal processing, financial analytics, image processing and finite element modeling.

The combination of an off-the-shelf development environment with custom hardware provides a system that offers enhanced performance without sacrificing the flexibility and ease of use of a general purpose system. When used as nodes in a high performance computing cluster, Convey systems deliver higher performance for a given number of nodes, providing substantially better performance per dollar or watt than conventional clusters.

CONVEY computer™

Convey System Architecture

Convey systems utilize a commodity two socket motherboard that includes an Intel® host processor and standard Intel I/O chipset, along with a reconfigurable coprocessor based on FPGA technology. The coprocessor can be dynamically reloaded with instructions that are optimized for different workloads. It also includes its own high bandwidth memory subsystem that logically shares a cache-coherent global memory space with the host processor.

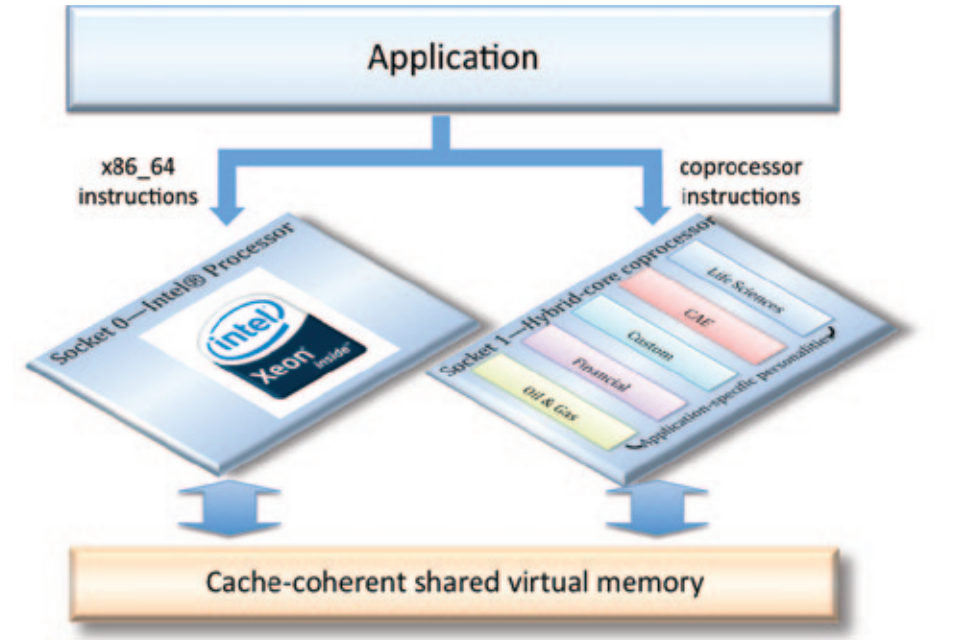

Figure 1. Hybrid-Core Computing

An application executable can contain both Intel and coprocessor instructions, and those instructions execute in the same virtual and physical address space (Figure 1). Coprocessor instructions can therefore be thought of as extensions to the Intel instruction set—they execute in the same address space and on the same data as x86 instructions.

Coprocessor instructions are grouped into sets (referred to as "personalities") that can be reloaded at runtime, allowing the system to present a customized set of instructions to each application. Each personality includes a base, or canonical set of instructions that are common to all personalities. The base set includes instructions that perform scalar operations on integer and floating point data, address computations, conditionals and branches, as well as miscellaneous control and status operations. A personality also includes a set of extended instructions that are designed for a particular workload. For example, the extended instructions for a personality designed for signal processing might implement a SIMD model and include vector instructions for 32-bit complex arithmetic.

The coprocessor has three major sets of components, referred to as the Application Engine Hub (AEH), the Memory Controllers (MCs), and the Application Engines (AEs) (Figure 2).

The AEH is the central hub for the coprocessor. It implements the interface to the host processor and to the Intel I/O chipset, fetches and decodes instructions, and executes Convey canonical instructions. It processes coherence and data requests from the host processor, and routing requests for addresses in coprocessor memory to the MCs.

Key Benefits of the Convey System Architecture

- Breaks the current power/ performance wall
- Significantly reduces support, power, and facilities costs
- Lowers system management costs by using industry-standard, Linux-based system management tools
- Reduces application porting and development efforts for high-performance applications

A cache for memory requests from the coprocessor to host memory reduces latency for remote references. Canonical instructions are executed in the AEH, while extended instructions are passed to the AEs for execution.

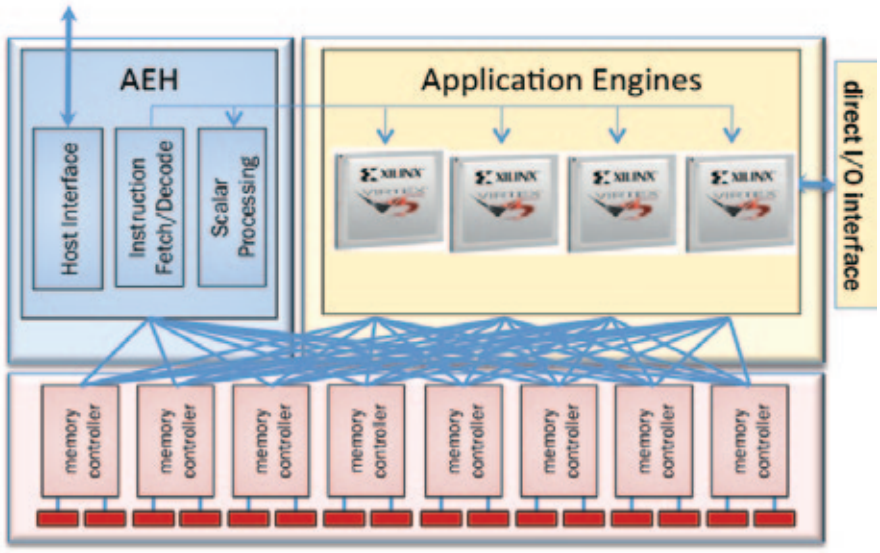

Figure 2. Convey Coprocessor

The Convey memory subsystem provides up to 80 GB/second of memory bandwidth to the Application Engines.

To support the bandwidth demands of the coprocessor, 8 Memory Controllers support a total of 16 DDR2 memory channels, which provide an aggregate of over 80GB/sec of bandwidth. The MCs translate virtual to physical addresses on behalf of the AEs, and include snoop filters to minimize snoop traffic to the host processor. The Memory Controllers support standard DIMMs as well as Convey designed Scatter-Gather DIMMs.

Most modern microprocessors transfer whole cache lines to and from memory, and waste bandwidth when executing programs that perform strided or random accesses to individual words. The degradation can be as much as 8x for an application that only uses 8 bytes from a 64-byte cache line. In contrast, the Convey memory system operates on individual words. The Scatter-Gather DIMMs are optimized for transfers of 8-byte bursts, and provide near peak bandwidth for non-sequential 8-byte accesses. The coprocessor therefore not only has a much higher peak bandwidth than is available to commodity processors, but also delivers a much higher percentage of that peak for non-sequential accesses.

Together the AEH and the MC's implement features that are present in all personalities. This ensures that important functions such as memory protection, access to coprocessor memory, and communication with the host processor are always available.

The Application Engines (AEs) are the heart of the coprocessor and implement the extended instructions that deliver performance for a personality. There are 4 AEs, connected to the AEH by a command bus that transfers opcodes and scalar operands, and connected via a network of point-to-point links to each of the memory controllers. Each AE instruction is passed to all four AEs. How they process the instructions depends on the personality.

While the clock rate for the FPGAs used to implement the coprocessor is lower than that of a commodity processor, each AE will have many functional units operating in parallel. This high degree of parallelism is the key to the coprocessor's performance. By implementing just those operations that are needed for a particular workload, many more of those units can be packed into each chip.

Personalities

A personality includes the precompiled FPGA bit files that implement a coprocessor instruction set, a description of the machine state model sufficient for the compiler to generate and schedule instructions, and an ID used by the application to load the correct image at runtime. A system can contain multiple personalities that can be dynamically loaded, but only one personality is loaded at any one time. Each personality supports the entire canonical instruction set, plus extended instructions that may be unique to that personality. Extended instructions are designed for particular workloads, and may include only the operations that represent the largest portion of the execution time for an application.

Personalities are the key to the Convey systems' performance and flexibility.

All personalities have some elements in common, however:

- Coprocessor execution is initiated and controlled via instructions, as defined by the Convey Instruction Set Architecture.
- All personalities use a common host interface to dispatch coprocessor instructions and return status. This interface uses shared memory and leverages the cache coherency protocol to minimize latency.
- Coprocessor instructions use virtual addresses and coherently share memory with the host processor. The host processor and I/O system can access coprocessor memory and the coprocessor can access host memory. The virtual memory implementation provides protection for process address spaces as in a conventional system.
- All personalities support the canonical instruction set, and the Convey compilers assume that the canonical instructions can be generated and executed.

These common elements ensure that compilers and other tools can be leveraged across multiple personalities, while still allowing customization for different workloads.

A personality therefore implements a computer architecture customized for a particular type of algorithm or workload. Figure 3 presents a diagram of the Convey SPvector personality, designed for signal processing algorithms such as FFTs.

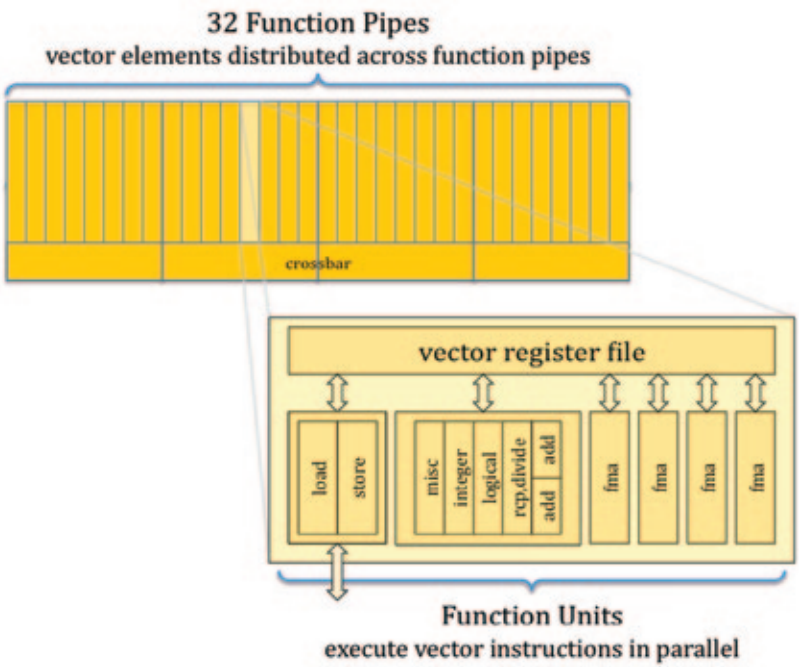

Figure 3. Convey SPvector Personality

This personality implements a vector register architecture optimized for single precision complex arithmetic. It includes multiple functional units and a large vector register set, supports out-of-order execution, and leverages the high sustained bandwidth of the word-oriented Convey coprocessor memory system. Convey compilers automatically generate both x86-64 and SPvector instructions from standard C, C++ and Fortran, and the resulting applications can be debugged using familiar tools such as gdb.

The SPvector personality is more general-purpose and is applicable to many scientific applications.

This vector model is applicable to a wide variety of applications that are loop intensive and perform the same set of operations on a large array of data. For instance, Convey is developing a personality optimized for financial analytics applications that uses the same basic structure and datapaths as the SPvector personality, but replaces the functional units with ones appropriate for codes that use double precision arithmetic, and make heavy use of intrinsics (Figure 4).

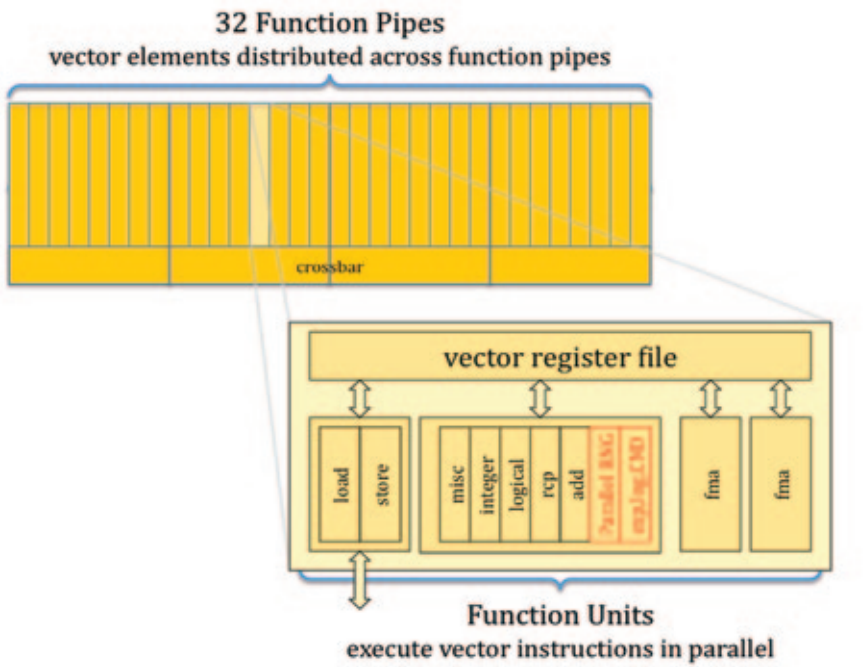

Figure 4. Convey Financial Analytic Personality

In this personality, pairs of single precision units are replaced with double precision units, and specialized units are added that support heavily used intrinsics, parallel random number generation, and computation of the cumulative normal distribution.

This approach isn't limited to just floating point applications. The SPvector and FINvector personalities already implement vector integer and bit operations; by replacing the floating point units with units devoted to these types of operations the coprocessor can achieve very high levels of parallelism and very high throughput for integer and bit operation intensive applications.

An even higher level of specialization is possible, however. Convey supports a personality development kit that allows customers to incorporate their own logic designs in a personality. The PDK allows the customer to leverage the coprocessor infrastructure, and operate on shared memory using virtual addresses. Since these personalities replace a particular routine, they are specific to that algorithm and are referred to as "procedural" personalities. Novel architectures can be implemented that coexist with x86-64 instructions in a process address space, leading to much higher application performance.

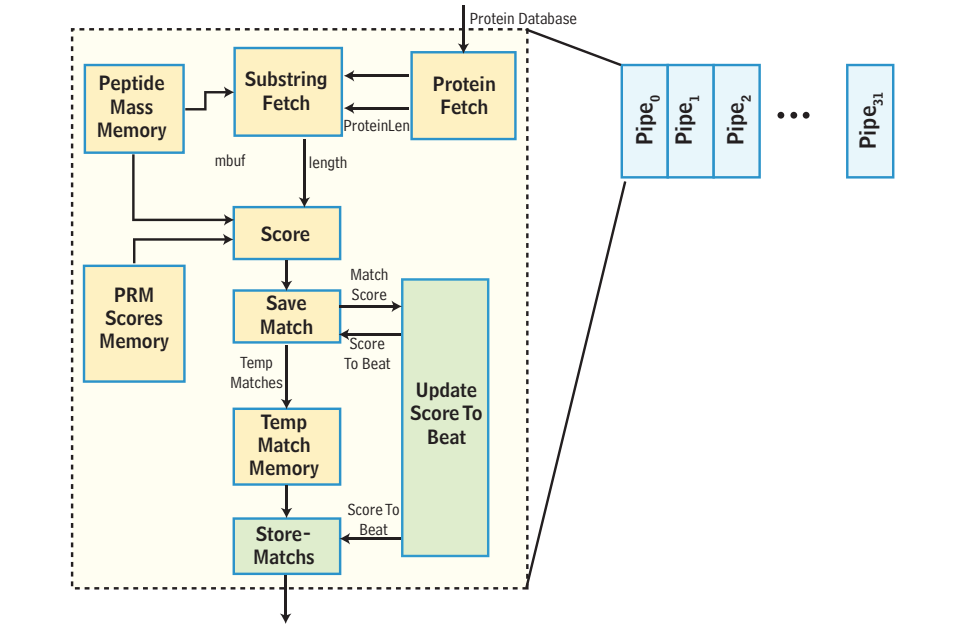

Figure 5. InsPect Procedural Personality

The InsPect personality is procedural—an entire algorithm implemented in hardware.

Figure 5 presents a diagram of a procedural personality that implements the search function in the InsPecT mass spectrometry bioinformatics application from the University of California, San Diego. The entire search algorithm is implemented as a pipelined set of logic blocks, which fetch samples from memory, compare them to a protein database, and accumulate scores for the best matches. The core logic is replicated 32 times and executed in parallel on the coprocessor. Since the coprocessor infrastructure handles virtual-to-physical translation, the pipes operate on virtual addresses passed by the main application. This example demonstrates how complex functions can be implemented entirely in hardware, yet execute in the context of a Linux process address space.

Programming Environment

The operating system for the Convey platform is based on the industry standard Linux operating system, with extensions to support the high performance features of the coprocessor. Application binaries built for a standard Intel 64 system running Linux® can run unmodified on the host processor. Applications can access the coprocessor either by linking in Convey math libraries with coprocessor versions of common algorithms or by recompiling with the Convey compilers.

The Convey compilers use a dual-path optimization system that generates both Intel 64 and coprocessor instructions from standard ANSI C/C++ or Fortran. A common front end and high level optimizer translate source files into the compiler's intermediate language and perform architecture independent optimizations.

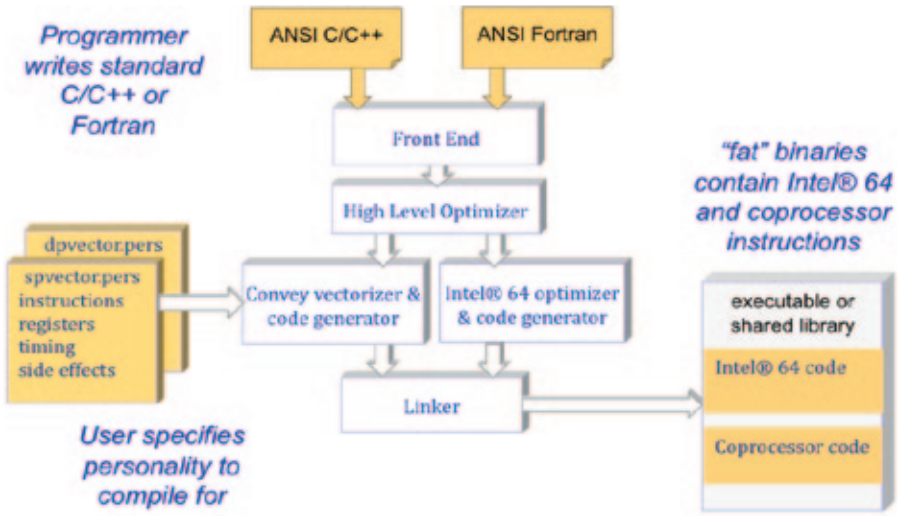

Figure 6. Convey Compilers

Code regions that appear suitable for execution on the coprocessor are cloned and tagged, with one copy of the code passed to the Intel 64 optimizer and code generator, and the other copy passed to the Convey vectorizer and code generator. The resulting executable contains both the Intel 64 and coprocessor code, along with a runtime test to determine which to execute based on a profitability test of which is likely to be faster.

The Convey Compilers go to great lengths to discover opportunities for inserting coprocessor instructions.

A run-time profitability test determines the optimal hardware on which to execute the generated code.

The Convey optimizer and code generator automatically identify code that can be vectorized, and generates coprocessor instructions and interface calls that dispatch them to the coprocessor (Figure 7). The compiler examines each loop in the program and performs a dependency analysis to determine whether iterations can be executed in parallel. Where it is safe to do so, the compiler translates scalar operations into vector operations from those supported by the target personality. Use of hardware features such as non-unit-stride and indexed vector loads, operations under mask, and reduction instructions enable transformation of many loops that would otherwise not be vectorizable. Higher level transformations such as inlining, loop interchange, and loop distribution also enable vectorization of loops that would otherwise be inefficient or impossible to vectorize.

```
$ cat saxpy.f
      parameter (N=100000)
      common a,b,c
      real*4 a(N),b(N),c(N),s

      s = 3.0
      do i=1,N
       a(i) = 2.0
       b(i) = 2.0
      enddo

      do i=1,N
       c(i) = a(i)+s*b(i)
      enddo

      write(6,1000) N
1000  format(i8,' iterations')

      end
$ cnyf90 -mcny_auto_vector -mcny_sig=sp saxpy.f
VECTORIZED STMT in MAIN__0 at 6
VECTORIZED STMT in MAIN__1 at 11
```

Figure 7. Example of automatic translation for SPvector personality.

The compiler also automatically inserts the calls and interface code required to allocate the coprocessor, load the appropriate personality, and control execution. The host processor initiates execution on the coprocessor by writing the address of the coprocessor code to be executed and any associated parameters to a command block in memory. When the coprocessor sees the command block updated, it fetches the first instruction at the specified address and begins execution. This mechanism leverages the cache coherency hardware to minimize latency. Since the coprocessor and host processor share the same view of memory, parameters can be in the form of scalar values or pointers to data structures in memory. When the coprocessor completes execution, it updates a status block. The host processor can either spin wait on this status block or execute other code while waiting for the coprocessor.

This low latency mechanism to transfer control to the coprocessor, along with the coherent shared memory architecture, allow the compiler and other tools to treat code executing on the x86 host processor and coprocessor as part of one seamless architecture. The strengths of both architectures can be applied to the parts of a program where they are most effective.

Custom Personalities

Many emerging applications in fields such as bioinformatics and data mining incorporate algorithms that don't fit classic architectural models like vector processing. Convey provides a Personality Development Kit (PDK) that allows custom instructions to be implemented to support such applications. Custom instructions might be as simple as a single instruction that instructs the AEs to process a large in-memory data structure.

Customers can create their own personalities with the Personality Development Kit.

The PDK includes logic blocks that implement the interfaces between the AEs and the other components of the coprocessor, tools to package bitfiles produced by the Xilinx® FPGA development tools into a personality, a simulator for debugging, and system and compiler APIs to allow execution of the user-defined instructions. The custom logic is loaded only in the AEs, however the system interface, virtual memory, and canonical instruction set implementation in the AEH and Memory Controllers is unchanged.

An architected set of instructions transfer data between the canonical register set in the AEH and the AEs, and initiate execution by the custom logic. This allows the user to develop FPGA logic that executes within a process address space using virtual memory addresses. The FPGA logic therefore can operate directly on the same data structures created by the application code running on the host processor. Other processes and system memory are protected from invalid accesses by the virtual memory protection system.

The ability to create new instructions that can be loaded dynamically allows the implementation of highly parallel instructions that are specific to algorithms and data structures not well served by classic scalar and vector architectures. As these applications evolve, the instruction set architecture can evolve along with new instructions.

Summary

The Convey HC-1 hybrid-core computer provides increased scalability and cost effectiveness by delivering higher performance per node for compute intensive workloads. It leverages high density programmable logic, allowing the creation of multiple specialized architectures optimized for specific workloads. These specialized architectures are integrated into an industry standard Intel 64 system—leveraging commodity components and allowing easy integration into an existing environment.

The Convey systems maximize productivity by delivering prebuilt personalities for important applications and a unified development environment based on standard ANSI C/C++ and Fortran. Users can create new personalities, allowing new instruction sets to be innovated for emerging applications.

CONVEY computer™

Convey Computer Corporation
1302 E. Collins Boulevard
Richardson, Texas 75081
Phone: 214.666.6024 Fax: 214.576.9848
Toll Free: 866.338.1768
www.conveycomputer.com

November 2008

Convey Computer Corporation
1302 E. Collins Boulevard
Richardson, Texas 75081
Phone: 214.666.6024 Fax: 214.576.9848
Toll Free: 866.338.1768
www.conveycomputer.com

November 2008