Pegasus

DATA SHEET

Pegasus Parallel Processing™ software is a complete environment for creating real-time Digital Signal Processing (DSP) applications.

Pegasus automatically generates application code for DSP chips and for communication with a host computer.

Pegasus defines a framework that greatly eases the extension of standard features. Fully custom components can also be added.

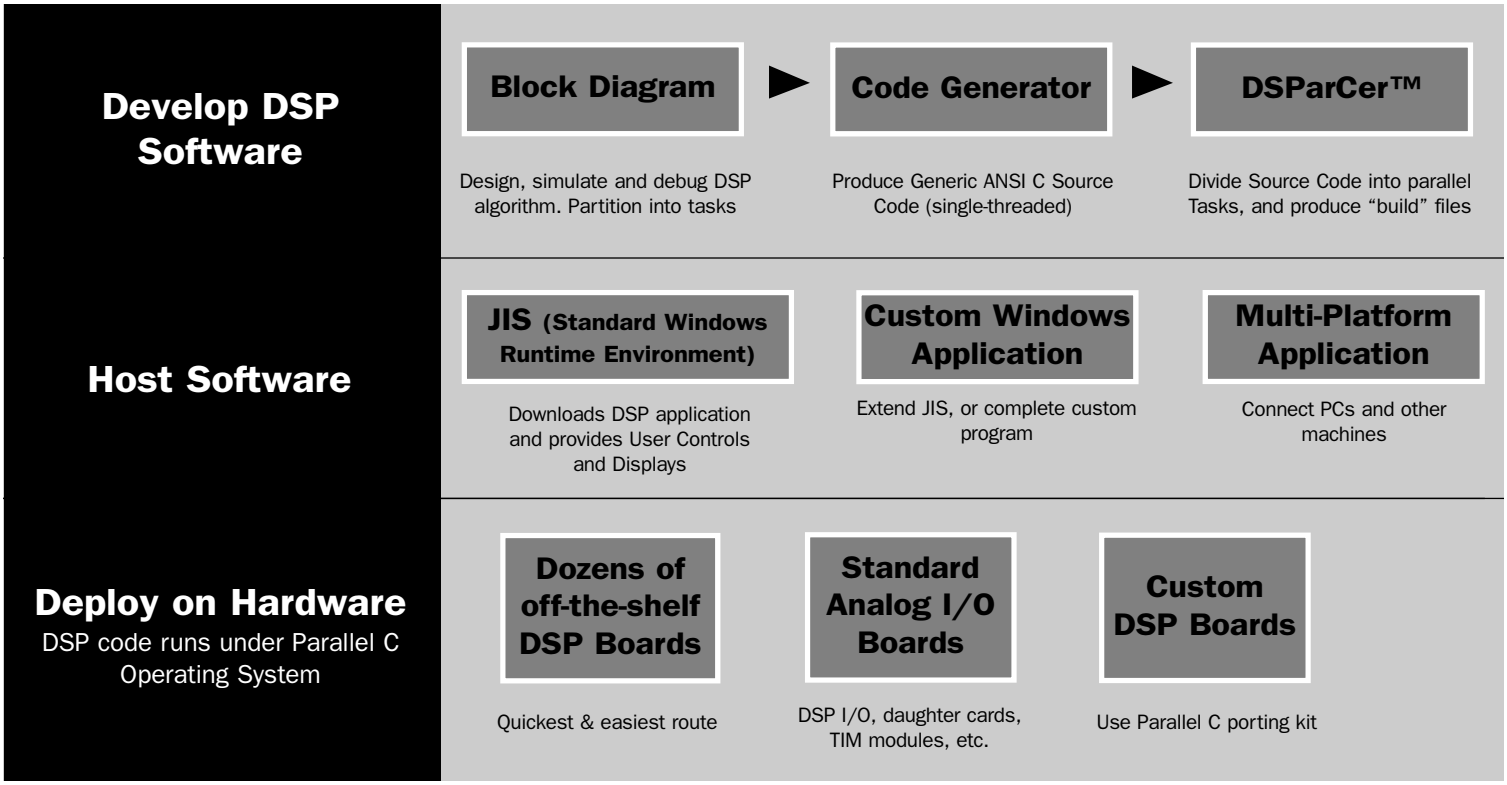
JOVIAN SYSTEMS, INC

Software for Parallel Processing DSP Design

The fast and efficient way to create Digital Signal Processing applications

- Produces complete applications that run on systems containing off-the shelf DSP boards or on custom DSP and I/O hardware.
- Includes TCP/IP communication blocks to make systems that are distributed across the Internet.
- Graphical Block Diagram front-end is used to implement processing algorithms and simulate them.
- Over 300 DSP and general purpose function blocks are included.
- DSP application code runs under an Operating System that is efficient, easy to use, and compact. OS provides real-time pre-emptive multi-tasking capability.
- Easily re-map tasks to different numbers and types of DSPs – develop code on one hardware configuration, deploy on another.
- DSP application code is compiled, not interpreted.
- Uses standard DSP C compiler, assembler, and linker.
- Automatically generates C Source Code for the DSP application program (can be customized, but usually not necessary).
- Generated code is based on user-modifiable template files (better than changing the source code directly – changes are preserved if code is regenerated).
- Code is automatically optimized for efficient parallel operation.
- Block Wizard creates custom function blocks for simulation and for automatic inclusion in DSP run-time applications.
- Existing C source code easily incorporated using custom blocks, or by substituting for automatically generated block code.
- Use Jovian's standard run-time host environment to run the DSP application programs. This automatically produces applications that include the same user interface as the simulation.
- Custom host-side application programs can be made using host extension kit, or a fully-custom host application can be created.
- Hardware independence: supports over 40 of the most popular ISA, PCI, and VME boards based on the TMS320C4x, SHARC, and DEC Alpha chips. Applications can run on stand-alone DSP hardware (no host computer).
- PC to VME cross-platform development supported.

SO MANY DSP CHIPS,
SO LITTLE TIME



BLOCK DIAGRAM DESIGN & SIMULATION ENVIRONMENT

The Block Diagram program runs under Windows and provides two fundamental functions of the Pegasus system: Algorithm Design and Algorithm Simulation.

Algorithms are designed by graphically placing blocks on a worksheet and connecting them together with lines that determine the data flow. Parameters for the block functions are entered through the diagram. Once the block diagram is put together, you can run the diagram to test the algorithm's operation.

The data for this "simulation" can be input from a hardware interface, from the Internet, from a disk file, or can be generated by blocks in the diagram. "User control" blocks can be included in the diagram (knobs, sliders, data tables, etc.)

The data output from the algorithm can be directed to hardware, to the Internet, to a disk file, or can be shown to the user through display blocks placed in the diagram. This simulation runs on the PC, and will be slower than the ultimate application running on the DSPs.

After successful simulation, the rest of the Pegasus components are used to generate an application that runs on the DSPs. Special Pegasus blocks in the diagram allow the developer to determine which functions map onto which DSP. This partitioning determines which tasks run in parallel, and which run sequentially. The DSP application can communicate with a host PC, but can be made to be entirely stand-alone.

User controls or displays in the block diagram will automatically appear in the standard JIS run-time application (code necessary to communicate between the host and DSPs is automatically inserted). The rest of the diagram is automatically translated into code that runs on the DSPs.

Data that comes from or goes to hardware during simulation can be made to use DSP I/O directly, or can still be routed through the host.

Block Diagram additional features:

- Large library of pre-built blocks for DSP, arithmetic, and general purpose functions.
- Hierarchical design allows assembling "primitive" blocks into "macro" blocks (also called hierarchy blocks or sub worksheets).
- Custom blocks are easily added using Block Wizard.
- DLL-based blocks allow using exactly the same host-side controls and displays in simulation and at run-time.

CODE GENERATOR

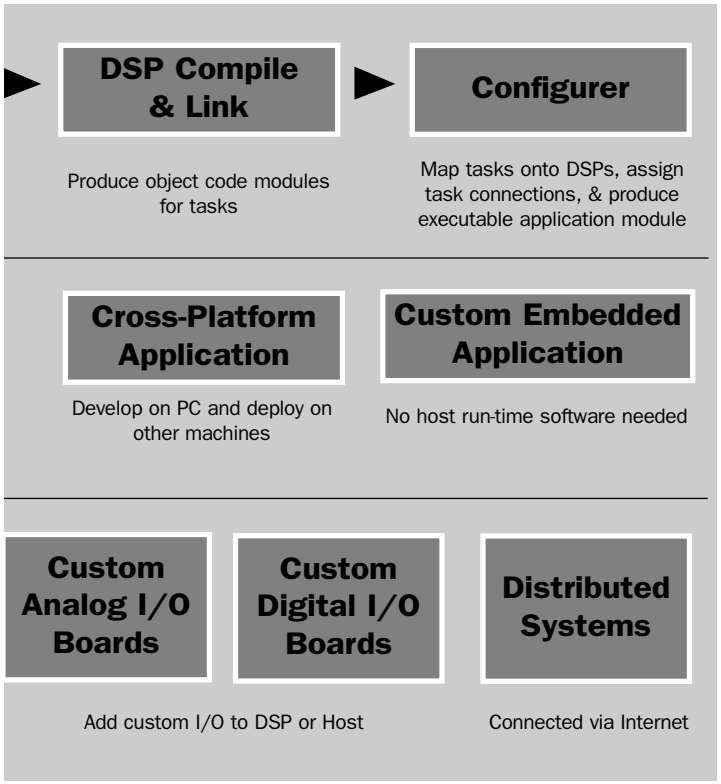
The Code Generator produces a single-threaded ANSI C compatible source code representation of the Block Diagram.

Each block of code is represented by a "utility" function and an "initialization" function. Arrays are used to transfer data between blocks.

Except for user control and display code, this source code could be compiled or cross-compiled for other non-DSP devices and would run as a single execution thread.

Code Generator additional features:

- Code Generator's output is the starting point for generating the multiple tasks that will form the DSP application.
- Supports real or complex data.
- Generates code for standard blocks and for custom blocks



added using Block Wizard.

- The code generated for the DSP can be the same as that used in the simulation, or can be entirely different.
- Can contain calls to DSP assembly language functions.
- Generated source code is completely independent of the simulation environment and does not require any sort of interpreter.

DSPARCER™

DSParCer starts with the source code produced by the Code Generator and processes it into a "parallelized" form.

DSParCer features:

- Translates single-threaded code into multiple tasks mapped onto one or more processors, based on user-entered information in Block Diagram.
- Automatic "temporal" optimization rearranges code to execute as much of a task as possible without waiting for input data.
- Creates threads for data movement between tasks.
- Inserts tasks and threads that implement an Asynchronous or Synchronous data passing mechanism between the DSP and host computer.
- Removes potential task communication deadlock conditions by placing task input and output code in separate threads.
- Produces makefile, linker command files, and Parallel C configuration file to automatically build DSP applications.
- C source code substitution for task blocks lets you use alternate code in place of the automatically generated code.
- Task substitution for task blocks swaps pre-built Parallel C tasks in place of the auto generated tasks.
- Optional assembly language optimized block substitution automatically substitutes hand-optimized assembly code for popular signal processing blocks.

JOVIAN INFORMATION SERVER (JIS)

JIS is Jovian's standard Windows run-time environment software for the host.

JIS features:

- Downloads DSP applications to the DSP hardware.
- Provides Graphical user interface (user controls that send data to the DSP, and data displays using data read from the DSPs).
- Provides window management for user controls and displays (tile, cascade, store custom layouts, etc.).
- Uses the Parallel C Windows Server to implement hardware independent communications between the PC and DSPs.
- Implements a general mechanism for User Controls, allowing custom controls to be added easily.
- Extension kit allows addition of custom GUI features.

BLOCK WIZARD

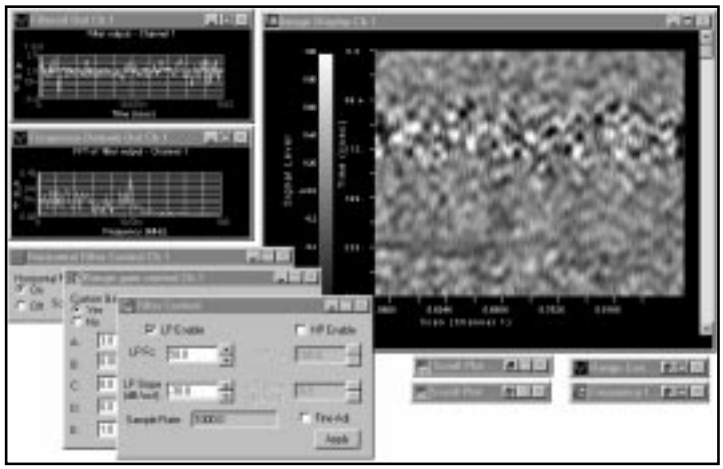
Block Wizard produces custom blocks that can be used in the same way as the standard blocks that come with Pegasus. These blocks can be "learned" into the Code Generator to automatically produce code for the DSP run-time application.

The source code produced for DSP run-time can be the same as the code used in the simulation, or can be different (library calls or calls to assembly language routines can also be inserted). For routines that do more than just crunch data, this allows for differences in executing on the Host or on the DSP – different ways to control I/O hardware, for example.

User controls or displays can be used in both the simulation and in the host-side run-time application program.

Block Wizard additional features:

- Prompts for standard and custom parameters for the block.
- Produces source code for the block that compiles into a DLL.
- Produces project build files for Microsoft Visual C++.
- Automatically writes all "boilerplate" code for the DLL. Users do not have to be DLL-knowledgeable, but just "fill in the blanks" with custom data processing code.
- Produces user interface parameter entry dialog box which can be easily modified.



Screen shot showing custom user controls on a Radar application created with Pegasus.

DSP OPERATING SYSTEM

The Parallel C Operating System (OS) provides the runtime environment on the DSP boards. Ultimately, all the other Pegasus components are used to produce a DSP program that runs on the DSP(s) under control of Parallel C. From the programmer's point of view, Parallel C is implemented as a set of functions that extend the standard C Compiler for the DSP. The standard DSP compiler and linker are used to produce independent object modules for each task.

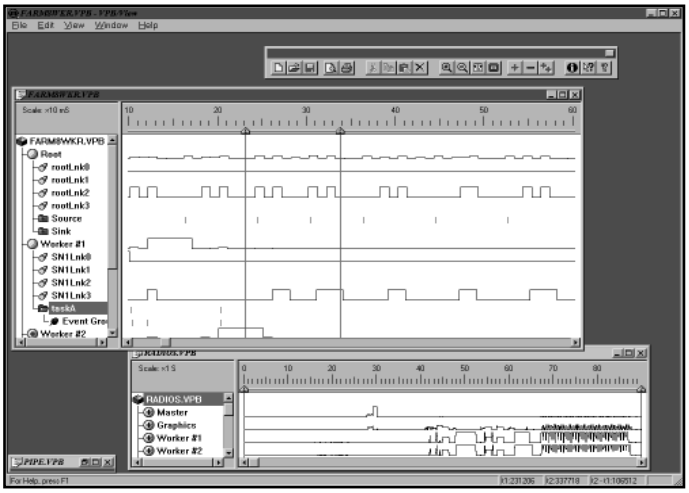
Run-time DSP application programs are produced by a configuration program (a configuration "script" is automatically produced by DSParCer). The configurer produces a downloadable application file by binding user tasks with a microkernel and with code that implements an underlying inter-task communication mechanism.

Parallel C additional features:

- Basic execution unit is a "task" (corresponds to a task block defined in Block Diagram).
- Number of tasks limited only by available memory.
- Tasks operate as "Communicating Sequential Processes" connected via unidirectional channels.
- Up to 256 input channels and 256 output channels for each task (each channel handles a stream of data).
- Tasks can be multi-threaded.
- Task threads can be dynamically created at run-time.
- Task and thread synchronization via semaphores or via message passing over channels.
- Small, efficient, distributed microkernel.
- Pre-emptive scheduling based on priority and timeslicing.
- Threads can be defined as "urgent" to bypass priority pre-empting and timeslicing (this "shuts off" the OS, reducing overhead to zero!).
- DSP application code is ROM-able.
- I/O port connection mapping and task-to-processor mapping is independent of task's processing algorithm (controlled by configuration file generated by DSParCer).
- Can develop on single-processor hardware, and later map to multiple DSPs.
- Virtual communication channels provide simple task connections.
- Physical communication channels allow connecting I/O devices or optimizing data flow.
- Provides robust host/DSP communications independent of the actual DSP board used.
- Host Server allows DSP to use I/O functions from the C Library (printf, scanf, etc.)
- Loader for DOS, Windows, Solaris, and VxWorks provides JIS independent operation.

OPTIONAL SOFTWARE

- VPB timing and event analysis software.
- Block Extension Kit for custom GUI capability.
- Optimized DSP assembly language function libraries.
- DSParCer Optimizer for automatic assembly library use.
- Parallel C Debugger Support Kit for use with JTAG emulators.
- JIS OEM Run-time Source Kit for complete customization.



Sophisticated real-time event tracing using VPB

CUSTOM DSP HARDWARE SUPPORT

Supporting a fully custom DSP board requires that the Parallel C operating system be ported to that board. This process can be done by the customer, or on a paid consulting basis. Call for details.

SYSTEM AND SOFTWARE CONSULTING AND TRAINING

Call for information on application-specific consulting or for training classes.

DEVELOPMENT SYSTEM REQUIREMENTS

- 133+ MHz Pentium system with Windows 95 or NT.
- 32+ MB RAM, 100+ MB free disk space.
- Parallel C compatible ISA or PCI bus DSP board with one or more DSP chips and at least 128Kwords memory per DSP (512Kwords suggested), or VME DSP card with supported interface to PC.
- Microsoft Visual C++, Version 5.0 or above.

JOVIAN SYSTEMS, INC
70 Tower Office Park
Woburn, MA 01801 USA
781-937-6300 • info@jovian.com • www.jovian.com