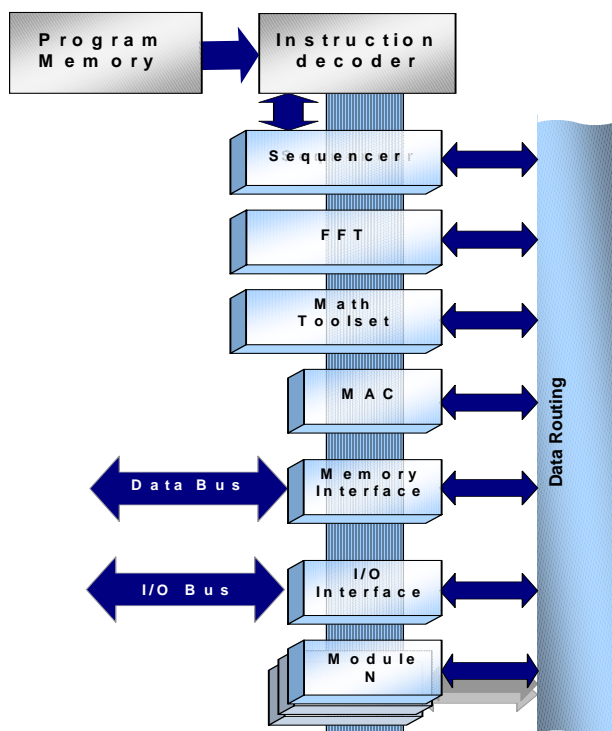


APE2 configurable DSP toolkit



APE2 is a customisable Digital Signal Processing technology developed by Cambridge Consultants. It is an Arithmetic Processing Engine for embedded DSP products and it comprises a complete toolkit so it is easily optimised for a specific application. APE2 is an adaptive data-path, low gate count, low power signal processor that is silicon proved in applications such as software defined radio.

As silicon processes for System on Chip ASICs evolve, more use is being made of DSP. APE2's flexibility speeds time to market by removing the need for custom hardware development and by introducing DSP programmability. APE2 provides a low risk solution with flexibility allowing its hardware to be precisely configured for each application.

The APE2 has a parallel modular structure, with modules selected from a library of processing elements. Any combination of modules can be selected to optimise the processor for a particular application. For example, dual Multiply-Accumulate (MAC) elements are easily created. Data is passed between modules using the data routing bus and its design allows the output of any module to be available at the input of any module.

High performance, low power

APE2 offers very low gate count of only 7,000 gates for a minimal 16-bit configuration. It has a scalable bus width allowing design flexibility.

Low power design is achieved by sleeping modules when not in use. Typical power consumption figures:

- 0.5 mW/MHz at 2.5V on 0.25 μm
- 60 μW /MHz at 1.25V on 0.18 μm

A recent implementation of APE2 produced a 20k gate DSP that was fabricated on 0.2 mm^2 of 0.18 μm CMOS, consuming under 50 μW .

APE2 has Very Long Instruction Word (VLIW) architecture for maximum flexibility. Coupled with code compression tools this minimises DSP program size.

Parallel execution produces high performance, and the unique data routing bus ensures all modules are kept fed with the data needed for non-stop signal processing.

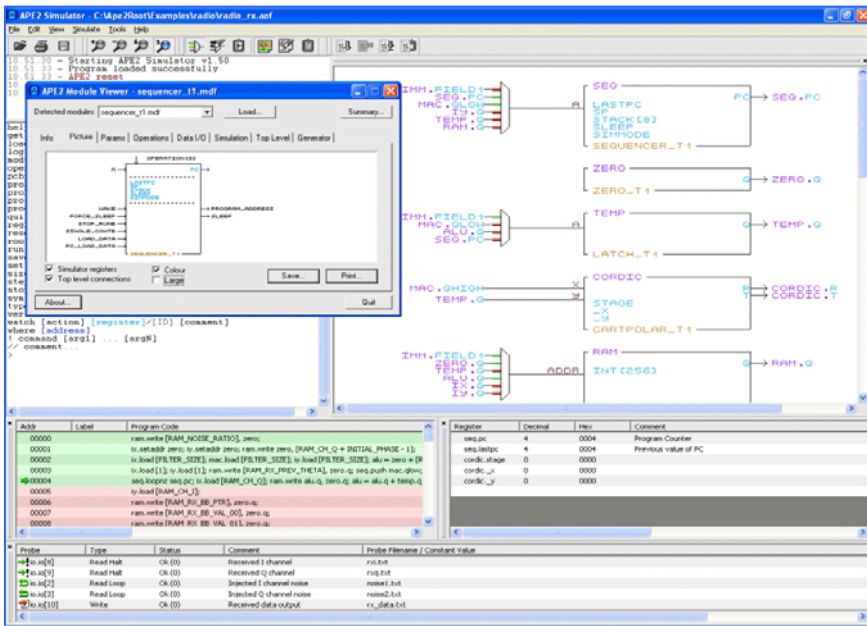
DSP developers' toolkit

DSP developers using the APE2 toolkit can code and observe their signal processing algorithms running within their configured APE2 simulation. The toolkit includes an assembler, a simulator, a hardware generator outputting an RTL net list, a code compression tool to minimise ROM footprint and it has a graphical user interface.

The toolkit is provided with licenses to embed APE2 in an ASIC. It supports hardware selection, code generation for any hardware configuration, design optimisation including code compression, and software simulation. The toolkit automatically generates Verilog RTL including ROM decompression hardware. Developers can have complete control and flexibility over their DSP design, or Cambridge Consultants can produce custom designs on request.

The functional flexibility opened up by the configurability of this DSP hardware and data routing bus allows APE2 designers to reduce NRE cost for products needing different capabilities in different markets. For example, one DSP core design could be programmed to support different wireless data modulation schemes. Or, with some flexibility designed into the core's hardware configuration, designers could easily use the DSP to add functionality to products over time, thereby reaping time to market and cost reduction benefits.

Alternatively, designers can use the optimisation facilities built into the tool suite to make APE2 designs as economic as possible, by identifying and eliminating unused or seldom used resources such as data paths, to design an implementation that is highly application specific.



Module selection

The APE2 can be configured to provide optimal performance for a particular application. This is achieved by selecting the hardware modules that will form the processor, both from a library of existing modules and, if desired, any user-designed modules.

Options

A rich library of standard modules is available. These IP modules have been developed and used in a range of projects. They include modules for:

- multiplies and shifts
- ratios and reciprocals
- sin, cos, tan, arcsin, arccos, arctan
- square root, exponents, logarithms
- FFT radix 4 butterfly and address generation
- co-ordinate transforms
- vector magnitude
- access to memories and hardware registers

Flexibility

The APE2 has no core structure. Designers have complete freedom to include only the modules required for a particular application.

Customise

Additionally, custom modules can be included within an APE2 design. These modules can be created by Cambridge Consultants to meet specific application requirements.

Optimise

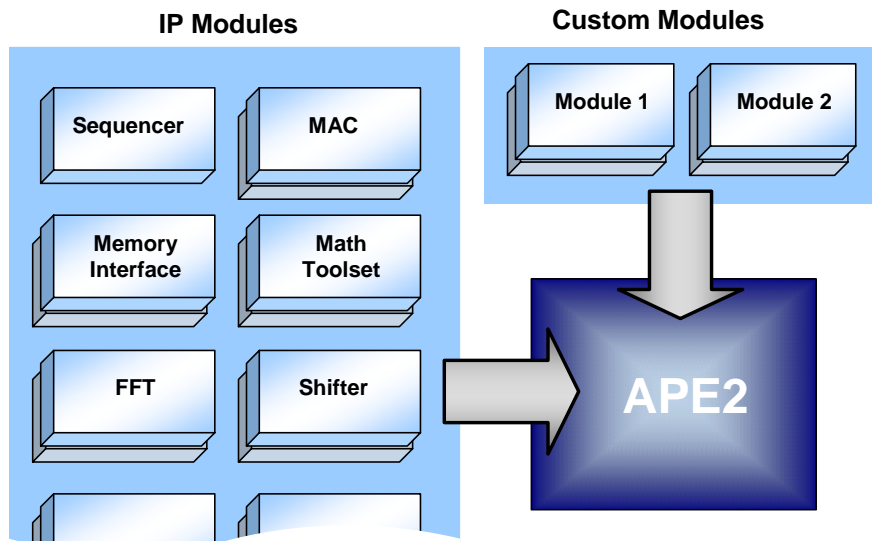
Any number of modules can be selected to form an APE2 design. Modules are selected from the toolkit and this selection can easily be changed during development if the requirements change. The toolkit provides usage statistics to enable the designer to reach an optimal solution. This ability to change the structure of the DSP allows designers to make tradeoffs between silicon area, processing speed and power consumption.

Performance

APE2 modules consist of high-speed arithmetic hardware such as the single cycle Multiply Accumulate (MAC) module. Designers can specify any number of MAC blocks to execute in parallel. Efficient addressing modes with auto-incrementing pointers and circular buffers are provided. The sequencer module implements zero overhead program loops, and function calls.

Specialised modules

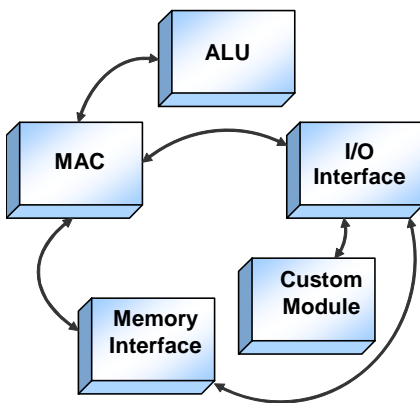
Specialised modules such as a radix 4 FFT block are available. This can perform a 64-point complex FFT in 768 clock cycles.



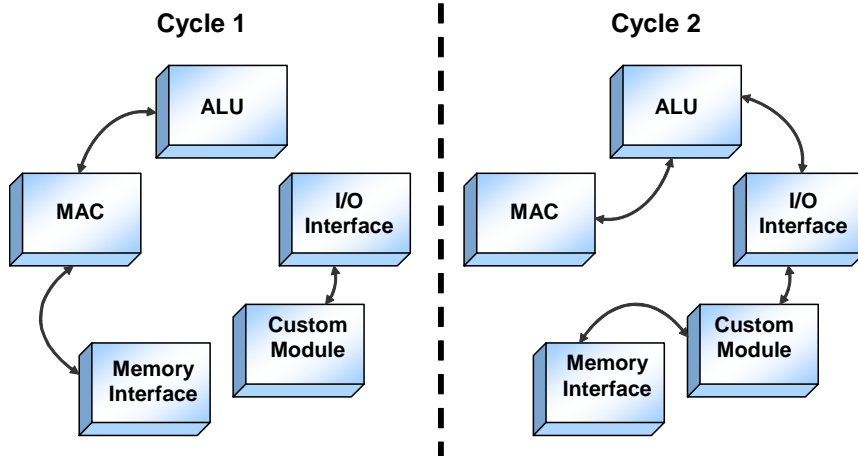
Data Routing

APE2 features an innovative data routing bus that optimises its DSP performance. Designers are able to select what data paths are available within each APE2 design. The output of any module can be made available at the input of any module. However to optimise a design for a particular application it is possible to reduce the data paths. The toolkit gives information on connection usage so seldom- or un-used connections can be seen.

Connection



The key advantage of the data routing bus is flexibility. Data routing can be changed on a per instruction basis. This routing ensures maximum data bandwidth. Processing performance is optimised as the data routing bus ensures all modules are fed with data. The toolkit generates data routing, allowing designers to focus on algorithms in the knowledge that the data will be available to each processing module.



Software Toolkit

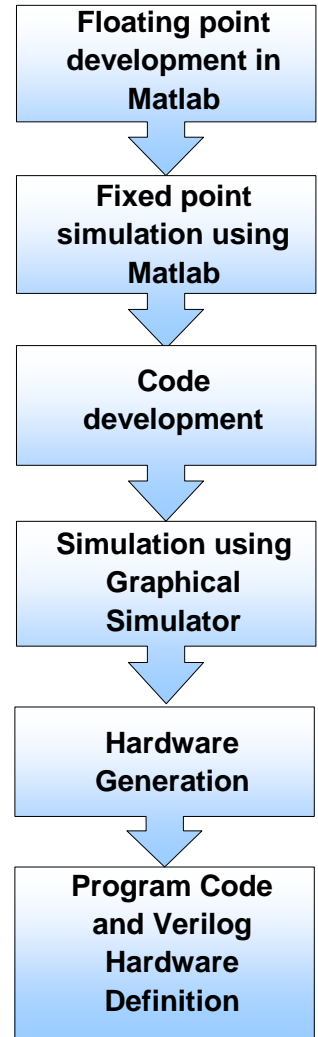
APE2's toolkit simplifies migration from prototype algorithms to finalised hardware and software. Each stage of the development process is supported, from coding algorithms to hardware generation.

Design

The selection of hardware modules that will be used for a design is defined in software. Additionally the connections between the modules are defined. These definitions can be changed at any time, and all of the tools will react accordingly. This maximises flexibility, allowing design changes to be made at any point in the development process.

Code Generation

In the initial stages the simple intuitive assembly language allows rapid coding of an algorithm. The assembly language uses mathematical operators to make translation from algorithm to code as clear as possible. The assembler provides usage statistics so that the hardware performance can be adjusted to meet the algorithm requirements. The assembler will produce code for any hardware configuration, and the instruction word is determined at assembly time. The assembler produces binary object code that can be used in a HDL environment.



Simulation

The toolkit provides a number of simulation environments. All of the variants simulate the execution of the APE2 code without the need for hardware.

The graphical simulator allows the designer to step through code execution, and examine the state of individual hardware modules, registers and memory contents. A visual representation of the APE2 is provided with signal activity indicated, even during execution. The simulator provides a flexible range of methods for setting breakpoints. All register values, and RAM contents can be saved to or read from files. Altogether these features provide powerful tools for the designer.

Integration

Other variants of the simulator are available to enhance the development process. A command line version of the simulator allows batch processing of data with the design. A Matlab MEX based simulator allows performance comparison in Matlab. The simulator can also be used as a plug-in to the XAP software developers' toolkit. This is ideal for testing the APE2 as a coprocessor for the XAP.

Hardware Generation

The toolkit automatically generates Verilog source code from the user design. The generated source code will only include the user-defined modules, connections, and interfaces. The modules are automatically scaled to match the data bus width.

Code Compression

The toolkit provides support for implementation of code compression. This tool will compress the program code, and produce a Verilog hardware block to decompress the code. The user can control the level of compression and the longest combinatorial path in the decompression hardware. The compression of code is loss-less and has no impact on the execution performance of the APE2.

APE2 as a XAP[®] coprocessor

APE2 is ideal for use as a co-processor with a main processor core. XAP processor cores from Cambridge Consultants can be interfaced with APE2 to form a complete platform for data intensive processing. The APE2 can be called to perform execution tasks from any point in the XAP program. Complex mathematical operations are off-loaded onto the APE2 to increase system performance. The APE2 has interfaces that allow it to share memory. Additionally the XAP tools offer a method to quickly integrate the APE2. The two processors can typically run at 60 MHz on 0.18 μm CMOS in a total of just 30k gates.

How it works

- XAP uses APE2 to run DSP subroutines
- XAP loads data into shared RAM, starts APE2 at required subroutine, the APE2 will then interrupt the XAP when the subroutine completes.
- Data can be exchanged in a dedicated RAM, or via main memory. The APE2 will have priority during its subroutine execution.
- APE2 code is either fixed in ROM, or loaded into RAM by XAP e.g. from Flash memory.

Toolkit integration

The XAP integrated software development and debug environment, xIDE, can be extended by a software plug-in to simulate the APE2 in parallel with the XAP. This allows complete simulation of the interaction of the two processors. DSP subroutines are developed in the APE2 toolkit, and then assembled to be called from within the XAP's C program.

Coprocessor applications

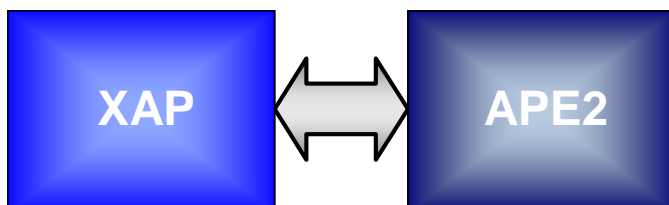
The combination of a RISC style processor and a dedicated DSP provides a low cost, low power solution for a range of applications.

- Radio. In radio applications the combination allows the XAP to handle the protocol layers, with the APE2 performing the DSP functions. The XAP can be used to handle framing of data and CRC, while the APE2 performs filtering and clock recovery.
- Sensors. The XAP can be used to provide an interface for the device while the APE2 is used to process input data streams.
- The APE2 can be used as a coprocessor in computationally intensive tasks such as audio processing, video processing, data filtering, data sorting, correlation and convolution.

More Information

Learn more about APE, XAP and other ASIC products and services www.CambridgeConsultants.com/ASIC

For further information send email to ASIC@CambridgeConsultants.com



XAP is a registered trademark and xIDE and APE are trademarks of Cambridge Consultants Ltd. All other trademarks herein are acknowledged.

© 2008 Cambridge Consultants Ltd and Cambridge Consultants Inc

Ref: ASICs-SB-008 v1.2