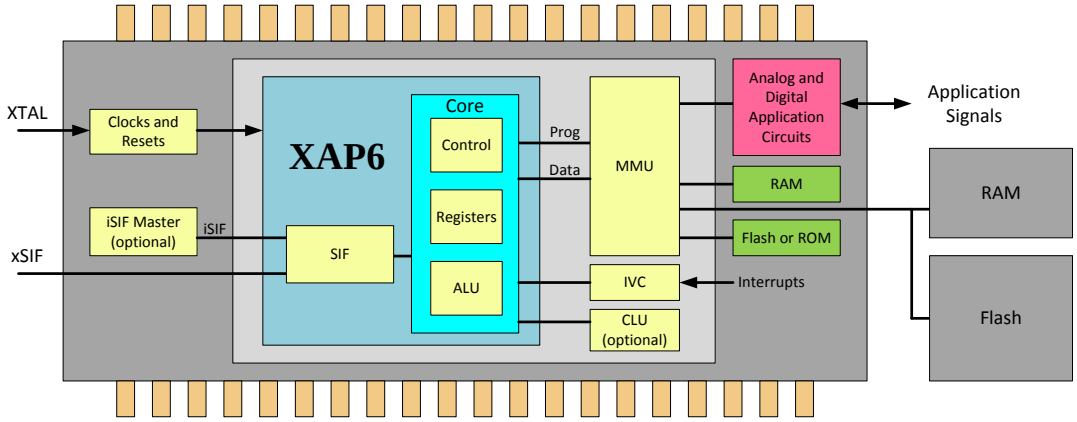
XAP6 processor core



XAP6 is a 32-bit processor IP core available as Verilog RTL. It offers advanced computing functions at the lowest possible cost and energy consumption.

XAP6 addresses the requirements of high volume and cost sensitive chip designs such as sensor nodes and wireless products, e.g. WiFi, Bluetooth, ZigBee, GPS, RFID and NFC. Applications for XAP6 include security, cryptography, automotive, communications, energy, medical devices, industrial, retail, metering, consumer and many other products.

The XAP6 architecture is very similar to that of the 16-bit XAP4 and XAP5 processors. Like them, it is optimised for low energy and is ideal for battery-powered systems.

Silicon designers can customise the IVC (Interrupt Vector Controller), MMU (Memory Management Unit) and CLU (Custom Logic Unit) as required for their IC.

Cambridge Consultants has developed and licensed the XAP® processor family since 1994. There are now over two billion XAPs in silicon.

XAP6a performance

- 32-bit, 4 GByte processor
- 8 general purpose registers
- Dedicated registers: Program Counter, 2 Stack Pointers, Global Pointer, Vector Pointer, 4 Breakpoints
- User mode and 3 privileged modes: Trusted, Supervisor, Interrupt
- 32 interrupts including four NMI
- 32 exception vectors
- Interrupt latency of 10 to 19 cycles
- Von Neumann architecture with little-endian organisation
- Very high code density
- 30k gates (NAND2x1) inc SIF Debug
- 1.36 DMIPS/MHz from 32-bit memory
- 272 DMIPS at 200 MHz*
- Low power
- High efficiency and code density
- xSIF serial debug interface
- xIDE Integrated Debug Environment
- GNU tools (GCC, Binutils, GDB)
- Soft core, Verilog RTL delivery

* Performance on 90 nm CMOS

Power, Performance and Area for other silicon processes available on request.

Low cost, low energy

XAP6 is a 32-bit processor core offering high performance and low energy consumption in a very small die area. Its modern architecture and software tools enable memory system design to be optimised for highly efficient systems on a chip.

XAP6 has a 32-bit internal bus between the registers and the ALU. It provides separate 32-bit Program and Data buses to the MMU that the user can customise.

XAP6 features very high density program code using a run-time mix of 16, 32 and 48-bit instructions. Its 3-stage pipeline architecture maximises instruction throughput at low clock speeds, enabling many programs to run directly from Flash memory, consuming less energy while still enabling low interrupt latency. This makes the XAP6 ideal for real-time applications.

Advanced computing

XAP6 brings advanced computing functions to low cost, low energy chip designs. Many of the functions support high reliability and security.

The processor runs programs using execution modes that isolate user tasks from privileged code such as the operating system and interrupt services. This protects critical code and data against unintended corruption and guarantees a real-time system's performance.

XAP6 has good support for re-locatable code and data. Programs start up quickly and can execute in-place from non-volatile memory such as Flash. Multiple programs, or instances of the same program, can co-exist in memory facilitating reliable Flash updates and ease of remote software upgrades.

Designers can organise their memory with custom MMU designs that secure XAP systems against hacking or reverse engineering. Exceptions are triggered by illegal memory accesses.

Reduced memory cost

XAP6 has a regular, orthogonal, load-store instruction set with instructions available in 16, 32 or 48-bit forms for different argument sizes. The linker automatically selects the shortest form it can use for each instruction. This carefully designed instruction set is optimised for XAP6's register layout and GNU GCC C compiler.

The resulting high code density means that programs execute quicker and use less energy.

Programs are easy to write or port to XAP6 with its single flat memory map, unaligned data access and byte addressing. Applications can be written entirely from C without the need for assembly language.

XAP6 can address up to 4 GBytes of memory for programs and data.

Fast real-time performance

XAP6 switches rapidly between modes and tasks in response to interrupts or exceptions that can nest to any depth. Switching is atomic with processor hardware storing and retrieving live data and state, to and from the stacks that SP0 and SP1 point to. A common convention is for SP0 to point to one persistent stack, while SP1 switches between separate stacks for each task or thread.

Privileged mode is required to access all of the processor's instructions and memory. Potential damage by User mode tasks is constrained because they can only use a subset of the instructions and memory in the system.

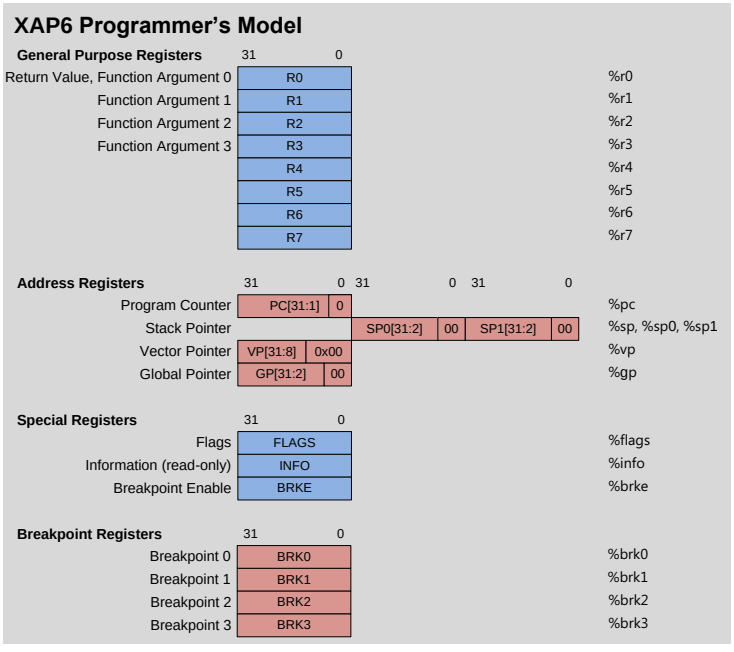
Task switching is fast, secure and deterministic. This maximises pre-emptive RTOS performance and reliability. RTOS ports for XAP6 include ThreadX, CMX, FreeRTOS and Micrium uC/OS-II.

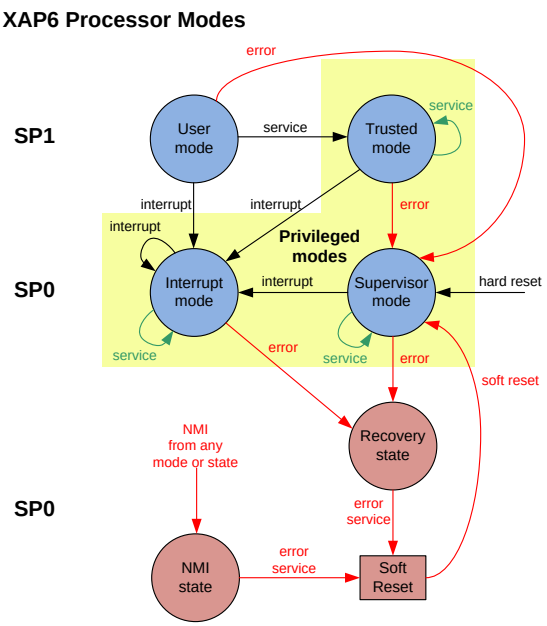
XAP6's event handling provides a fast real-time interrupt response without any penalty for nested interrupts. Interrupt entry latency is between 10 and 19 clock cycles (depends on the completion time for the interrupted instruction). Longer instructions, e.g. multiply, divide and block copy can be interrupted. XAP6 has 28 maskable and 4 non-maskable interrupts. The IVC hardware allows software assignment for 16 interrupt priority levels.

Rich feature set

XAP6 has a rich set of instructions and features:

- Multiply/Divide/Remainder. 16x16 mult in 1 clock. 32x32 mult in 3 clocks. 32/32 div, rem, divrem in 32 to 35 clocks.
- Signal processing instructions.
- Endian swap. Instructions to rearrange bits or bytes between big and little-endian form.
- Custom Logic Unit. Customised instructions can be added to the processor, executed by custom hardware connected directly to the XAP core. A set of instruct-ion templates, which preserve compiler performance, pass data directly to and from XAP6's general-purpose registers.
- Sleep mode. An instruction to stop the processor between tasks for minimum stand-by energy consumption. A wake-up signal restarts execution.
- iSIF internal interface. Enables XAP6 to act as a slave to an on-chip master processor or state machine. Used if XAP is part of an IP sub-system in an SoC or for saving registers prior to powering down the core to stop energy loss by leakage current.





Flexible hardware

The standard XAP6 hardware is a single processor core including iSIF interface and xSIF debug, which is used unmodified in a chip design. A Memory Management Unit and an Interrupt Vector Controller are also provided and these are modified by the chip designer to achieve the required architecture, i.e. memory map with i/o registers and interrupt structure. XAP6 can be extended with an optional Custom Logic Unit.

The XAP6 core has separate physical buses for program and data. They use the same logical address space with all vectors, instructions, data and i/o registers appearing in a single 32 GByte Von Neumann address map. This can support many on-chip and off-chip memory configurations. A typical XAP6 chip design might have an internal 32-bit flash, an internal 32-bit RAM and an external 1-bit or 4-bit flash memory. The silicon designer can decide whether the MMU (Memory Management Unit) connects to the memories with a shared bus or separate buses (enabling simultaneous program and data accesses).

This flexibility enables the silicon designer to optimise the chip design to best exploit the on-chip and off-chip memory technologies available. This leads to systems with lower unit cost and energy consumption.

The silicon designer also has great flexibility to customise the IVC (Interrupt Vector Controller) to meet the exact interrupt requirements of the application.

If the application requires custom instructions for DSP or Security functions, the silicon designer can develop a CLU (Custom Logic Unit) to map new instructions to the pre-allocated CLU instructions already included in the XAP6 instruction set and software tools. This can be useful for confidentiality, as the custom use of these instructions is not publicly documented.

Execute from Flash memory

XAP6 is very efficient at executing programs stored in non-volatile memory such as OTP ROM or NOR Flash that is embedded on-chip or connected off-chip, perhaps in a System in Package (SiP).

Flash memory is large and offers a very low cost per bit, but has slow access time. XAP6's design gains optimum performance from Flash without compromising its capability to run fast from RAM or ROM.

The MMU can be customised to operate with parallel Flash, or with off-chip SPI or QSPI serial Flash.

XAP6's Vector Pointer holds the address where a program's event vectors are stored, including reset. In-place execution of programs can then be started from wherever they are located in memory. This means that programs start up immediately without having to first copy vectors or instructions to another memory.

Multiple instances of programs can co-exist in memory. This provides a safe mechanism for Flash program updates whereby a download can complete safely and be checked for errors before changing the VP start value and restarting the system.

XAP6 has a Global Pointer (GP). Multiple instances of a program can be run from a single copy in memory. A different GP value is used for each instance. Each GP value points to a different RAM region for the global variables for each instance. i.e XAP6 can support position independent data, enabling dynamic relocation of program RAM. Multiple copies of a single piece of application code can share a single flash image, and execute simultaneously using different areas of RAM.

Secure and reliable

High security systems, resilient to hacking or reverse engineering, can be designed with XAP6. The processor's software modes, event handling, deterministic behaviour and configurable MMU facilitate various techniques that support a secure software kernel managing isolated tasks in a trusted and high-availability system.

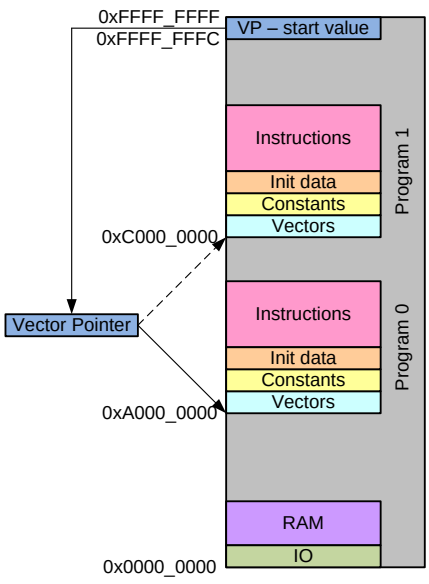
For example a section of code held in ROM may be used to access security keys stored in memory. By gating addresses in the MMU with the Program Counter value, it is possible to prevent access to the key memory by anything other then the trusted ROM routine. It is also possible to prevent reading and disassembly of the ROM code if it is made accessible only when the Program Counter executes from it.

High availability and high reliability systems are designed through use of XAP's software execution modes combined with the MMU design.

XAP6's low power consumption and powerful instruction set enables it to implement cryptography algorithms such as AES-128 and ECC-256p at low energy.

Further security can be engineered by introducing a Custom Logic Unit containing cryptography hardware, and using 'secret' CLU instructions to access it.

XAP6 Vector Pointer and Memory Map



Instruction set summary

The XAP6 processor has over 140 distinct instructions as follows:

- Load and Store: many address modes, atomic swap (semaphores)
- Push and Pop: to and from registers, push immediate, function return pop
- Move: immediate and register with many address modes
- ALU operations: add, subtract, multiply, divide, remainder, and, or, xor
- CLU instructions: direct transfer to and from registers
- Compares: 8, 16 and 32-bit with signed and unsigned result flags
- Branches: conditional and unconditional
- Shifts and Rotates: left, signed right, unsigned right, with carry, rotate
- Block memory: copy, store
- DSP: absolute, msbit, bit flip, byte flip
- Single bit: operations on Flag bits
- Miscellaneous: system calls, break, soft reset, sleep, SIF, version, flags registers, address registers, breakpoint registers

Tools and support

xIDE is a fully-featured Integrated Development Environment for XAP. It includes a source-level debugger and XAP6 instruction set simulator. Python script support automates tasks and can extend simulation to model other parts of a chip design.

XAP6's GNU tool-chain (GCC, Binutils) integrates with xIDE, making the assembler, linker and C compiler easy to use. XAP6 can also be debugged with GNU GDB.

We can provide an evaluation copy of xIDE for XAP6 that enables pre-silicon assessment of a program's code size and execution time.

Application software development can be simplified with versions of xIDE extended with software plug-ins that interface to a specific chip. These versions of xIDE can carry a licensee's brand identity and be distributed under licence.

This all makes the XAP6 attractive to professional programming teams who need to support a range of chips and applications over a long product life cycle.

Right first time

Cambridge Consultants' patented xSIF interface provides a non-invasive debug facility capable of fully controlling the processor core and acquiring data from inside the processor and any of its memory-mapped peripherals. This allows a full investigation of chip behaviour (except where prohibited by the MMU's hardware security design).

xSIF uses a high-speed four-wire serial interface with handshake for fast debug access. Development PCs connect via an intelligent SIF pod. xSIF can also run over JTAG when used for manufacturing test.

Cambridge Consultants develops tools such as xIDE and xSIF that combine to reduce risk and time-scale for projects using XAP6.

Designers using these tools should get their ASIC right-first-time, even with programs in on-chip ROM. Our engineers use them to good effect; delivering projects on-time and in-budget. For added flexibility, XAP6 designers can include a ROM patch system whereby selected function calls are redirected to code held in RAM.

Complete deliverables

The XAP6a is a 3-stage pipeline processor core supplied in Verilog RTL for synthesis and layout in ASIC or FPGA. It comes with synthesis scripts and a test bench including a complete instruction test program.

The XAP6a delivers 1.36 Dhrystone MIPS/MHz. On 90nm CMOS it can be clocked at 200 MHz to deliver 272 DMIPS. For slower low power systems XAP6a can execute code directly from Flash memory to deliver over 21 DMIPS at 16 MHz. XAP6a synthesis can be optimised for area or for energy consumption.

XAP6a can be synthesised to any silicon process from 700nm to 20nm. XAP6a can be used on CMOS, SiGe or BiCMOS processes.

XAP6 has very high code density allowing you to use smaller memories on and off-chip. The flexible MMU makes it easy to integrate XAP6a with the latest parallel and serial, volatile and non-volatile off-chip memory technologies.

At just 30k gates XAP6a combines small size with high performance.

Visit our web site to apply for a free 30-day trial copy of the xIDE software tools for XAP processors.

XAP

www.CambridgeConsultants.com/xap
xap@CambridgeConsultants.com

Ref: ASICs-SB-017 v1.5, 24 January 2015

Cambridge Consultants

xap@CambridgeConsultants.com
www.CambridgeConsultants.com/xap

Cambridge Consultants Ltd
Science Park
Milton Road
Cambridge
England CB4 0DW

Tel: +44 (0)1223 420024
Fax: +44 (0)1223 423373

Cambridge Consultants Inc
745 Atlantic Avenue, Floor 6
Boston MA 02111
USA

Tel: +1 617 532 4700
Fax: +1 617 532 4747