

[Product Innovation]

PIC MCU Grows To 16 Bits, Adds A DSP

A mainstream 8-bit controller extends ISA to 16 bits and delivers 30 DSP MIPS.

[Ray Weiss](#)

ED Online ID #3919

[June 4, 2001](#)

Copyright © 2004 Penton Media, Inc., All rights reserved.

Printing of this document is for personal use only.

[For reprints of this or other articles, click here](#)

Core embedded applications increasingly demand more processing power, higher math capability, and faster processors. Driving these needs is a shift to more processing power to replace expensive sensors, to achieve closer motor control, and to support Internet, speech, and audio needs. One solution is faster, more powerful 16-bit controllers that also deliver DSP math power—and that's what the new dsPIC, the 16-bit MCU/DSP from Microchip Technology, is all about.

The dsPIC integrates a 16-bit microcontroller with a 16-bit DSP ([Fig. 1](#)) But instead of creating a totally new architecture, Microchip transformed the popular 8-bit PIC architecture's ISA into a 16-bit MCU/DSP, resulting in the dsPIC. It's upward-compatible to the PIC18xxxx. You can compile and assemble PIC18xxxx code and run it on the dsPIC. Engineers familiar with PICscan can easily step up to the new 16-bitter ([Fig. 2](#)). The dsPIC's designers did more than extend the 8-bit PIC ISA. They also built a new architecture with improved performance, programmability, C compilability, and RTOS execution.

Running at 30 MHz, the dsPIC delivers a peak 30 MIPS (25 MIPS typical). Designed for control, it incorporates advanced bit manipulation, fully interruptible operation, and prioritized interrupts. On the DSP side, it delivers a full 16-bit, single-cycle MAC DSP with dual 40-bit accumulators. This is a tuned architecture that minimizes hardware overhead, but provides the key resources. For example, it supports a full 16- by 6-bit register file, the W registers.

A DSP And μ C

One way to add processing power is by implementing a DSP. In fact, many applications, such as cellular phones and disk controllers, previously included both a microcontroller (μ C) and a DSP. (Now the trend is to a single chip.) Unfortunately, classic μ Cs and DSPs differ. The former control well, but lack math horsepower; the latter do math well, but lack many real-time control features. μ Cs routinely handle multiple interrupts with fairly low latencies. Conversely, DSPs tend to lock out interrupts when processing inner loops, or they can only perform limited interrupt processing there.

A DSP's main architectural difference is that it supports inner-loop MAC-type processing, typically used to build a series. This inner loop consists of some control and many MAC (or other accumulator

function) instructions. Each MAC operation—a multiply (a constant times a variable)—is a single-cycle operation (pipelined). An automatic fetch of the X and Y variables and an automatic loop control for single-cycle MAC execution support it. DSPs, which accomplish many operations in a single cycle, are the last of the CISC machines.

The dsPIC supports X and Y memory accessing, a set of MAC-type operations, and zero-overhead loop control. It functions as a DSP and an MCU, with both sharing the instruction load and decode logic. The MCU has its own adder and register set, but the DSP unit also operates with the MCU general register set, the W registers that also serve as the MCU's register set.

Generally, while either DSP or MCU instructions execute, the other lies idle. In the Euclidean Distance instructions, however, both the DSP and MAC arithmetic logic units (ALUs) are used. During DSP operations, results are stored in DSP result registers but can be explicitly written to W registers.

The DSP engine executes a MAC-type instruction in one cycle by prefetching the X and Y data values in the previous cycle and dropping them into the W registers for the next MAC cycle. In a sense, the dsPIC is pipelined.

The system supports 32 kwords of data memory, partitioned into X and Y data spaces. Y, a subset of X, is only used for DSP operations, while the X space is used for both MCU and DSP operations. The X and Y address generators supply X and Y data-access addresses for single-cycle MAC operations. In addition, the hardware implements DO (loop) and Repeat instructions to support inner-loop and multiple-instruction execution. These can run single-cycle MAC-class instructions using hardware register counters to monitor loop or instruction counts for zero-overhead looping.

MCUs normally shine at bit-manipulation, which is necessary for control and fast RTOS operation, whereas bit-manipulation instructions are a fairly new addition to most DSPs. The dsPIC has an outstanding set of bit-manipulation operations for both MCU operations and DSP operand scaling. For starters, the dsPIC extends the bit-manipulation features of the earlier PIC18xxxx.

But the dsPIC goes much farther in bit manipulation. It implements a set of Bit Find Operations that enable the software to find the first active bit in a data word. These powerful operations eliminate the need to search a word's bits, bit by bit, to find the first marker bit. They support scaling of the DSP operands and can find a scaled value to help set its exponent. They also support normalization of the accumulator values for efficient 16-bit word storage and help to speed up RTOS processing that involves bit checking, like interrupt polling, task switching, and I/O bit processing.

The DSP Engine

Tightly coupled to the MCU is the DSP engine, a minimal architecture implemented in logic and registers. An asynchronous logic implementation, it has no pipeline, and only the registers are clocked. Engine inputs come from the MCU's W registers, which are clocked into the W registers on the previous cycle.

The DSP engine has a fast 16- by 16-bit multiplier, 40-bit barrel shifter, 40-bit adder/subtractor, and two 40-bit accumulators (A and B). The engine handles both accumulator and MAC operations. Accumulator instructions include Add Accumulators, 16-bit signed Add, Load, Negate, Store, Arithmetic Shift, Store Rounded, and Subtract.

For MAC class operations, the instructions are Clear Accumulator, Euclidean Distance, Euclidean Distance Accumulate MAC, Move Special, Multiply, Multiply & Subtract, Square, and Square &

Accumulate. The accumulators have 8 leading guard bits to handle DSP overflow, underflow, and saturation. Only the accumulator and barrel shifter result registers are clocked.

The DSP engine executes the DSP instructions, as well as the MCU instructions that use the barrel shifter. The barrel shifter takes in 40-bit values. It supports a 15-bit shift to the right and a 16-bit shift to the left. Implemented in hardware multiplexers, the shifter completes a shift in one cycle. Longer shifts can be handled by the MSL instruction.

Designed for fast execution and low interrupt latencies, dsPIC instructions are prefetched during the last instruction and executed in the next cycle. Most instructions are packed into a 16-bit word and execute in a single cycle, but instructions that flush the prefetch buffer require two cycles. Such instructions include relative branches, relative calls, skips, and returns. Also, the few two-word instructions, like Call and Goto, take three cycles to execute—two for fetching and one for the pipeline flush.

The architecture builds on a 24-bit instruction word and a 16-bit datapath. Including 8 guard bits, the DSP engine handles 40-bit MAC-type operations. The dsPIC supports separate data and program memories, with up to 32 kwords (16 bits) of data and 4 Mwords (24 bits) of program memory. The initial dsPICs will have up to 32 kwords of on-chip flash memory. The DSP implements 94 instructions, including 19 DSP instructions, and it has 11 addressing modes. To save register contexts for interrupts and function call/returns in user memory, the hardware relies on a software stack (W15), which starts at lower memory and grows upward.

Two address generator units (AGUs), X and Y, support DSP operations. All instructions use the X AGU, which handles all addressing modes including DSP modulo and bit-reversal addressing. Only the DSP employs the Y generator, which supports modulo addressing. The data memory is partitioned between the X and Y data address spaces, and the Y space is a subset of the X space.

The system supports both byte (for PIC compatibility) and word addressing. But instructions and word data must be aligned on word boundaries. Addressing options include register direct and register indirect, with pre- or post-increment/decrement, register offset, or constant offset adjustments. Instructions can load literal, register, or memory values. The hardware supports three address instructions. Data and program memory are linear address spaces.

The dsPIC implements a windowed access to use the program code space for data. An access window opens up, reads from, and writes to the code space. These accesses take just one cycle for Repeat instructions (no instruction access), but otherwise take two cycles.

For multiple MAC and DSP operations, the hardware implements DO (loop) and Repeat instructions. DO sets up an execution loop with a count, a literal or register value. Repeat executes the following instruction *n* times, with the count set by a literal or register value. Both instructions are interruptible. The DO instruction is fairly robust, supporting nested loops and branches. The end of the loop address can be anywhere, as it's not restricted to a downstream.

The dsPIC instruction set is far from minimal. It's robust, giving compiler writers and programmers a lot of leeway to accomplish tasks, and tailored for easy C compilation.

The dsPIC provides a full set of 16 vectored interrupts—one Reset, seven nonmaskable traps, and eight prioritized general interrupts. All operations, both MCU and DSP, are interruptible. DSP operations are stopped, and the controlling register values stored, to restart the DSP function. Traps are ordered

into a fixed priority scheme

For Fast interrupts, the processor relies on shadowing, such as a fast backup for the W, PC, Data Table Page Address, Data Space Program Page Address, and Repeat Loop Counter registers. Basically, the hardware maintains an active and a shadow set of registers. On an interrupt, the hardware directly loads the shadow registers in parallel, eliminating the time to sequentially save the register context. Fast Interrupts, though, are only one level deep. Any more would lose register context.

Interrupt Latency for Fast Interrupts is one cycle. On the second clock cycle, the hardware executes the first instruction in the Interrupt Service Routine (ISR). But for standard interrupts—both one- and two-word instructions—interrupt latency is on the order of three clock cycles.

Price & Availability

dsPIC Beta sampling will begin in the fourth quarter of this year, with production in the first quarter of 2002. Cost is \$3 to \$9 in lots of 10,000. Microchip projects a 20-chip family by the first quarter of 2002.

Microchip Technology, 2355 W. Chandler Blvd., Chandler, AZ 85224; (480) 792-7200; www.microchip.com.

FUNCTION

- 24-bit instruction, 16-bit datapath
- Superset of 8-bit PIC18xxxx
- 16- by 16-bit general register set
- Full DSP capability
- Two 40-bit accumulators
- 16- by 16-bit multiplier
- 40-bit barrel shifter
- Dual X, Y accesses per clock cycle
- Zero-overhead MAC loops
- 4-kbyte data RAM
- 32-kbyte flash code memory

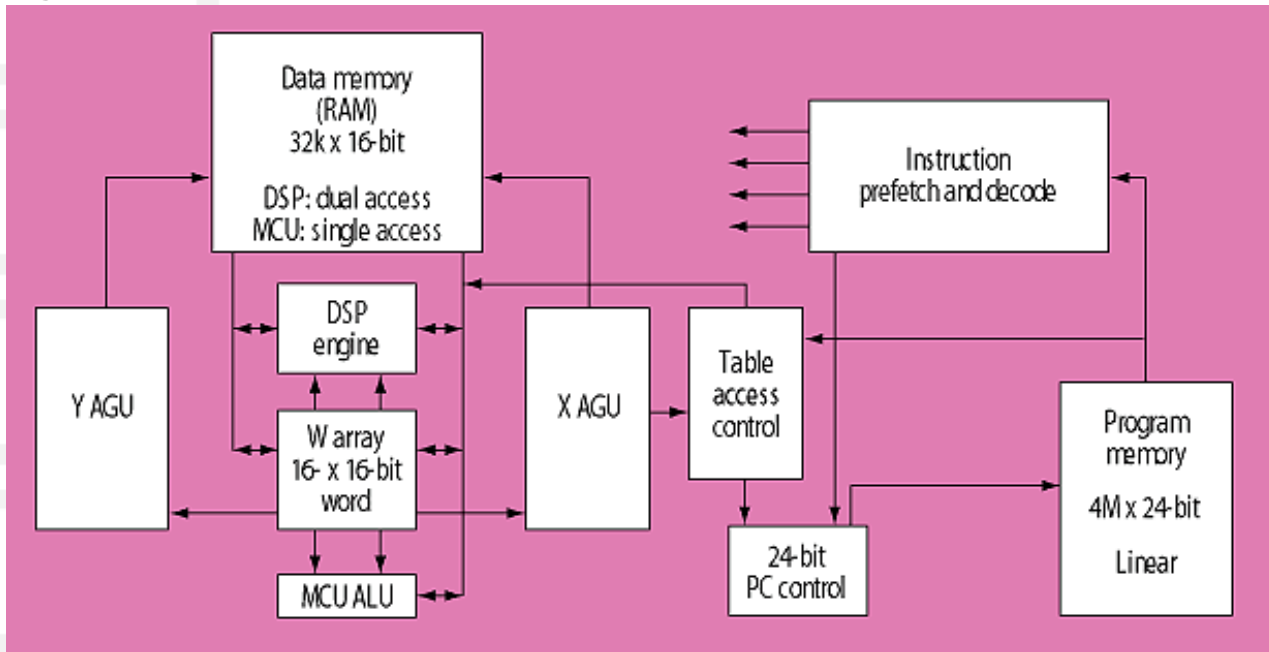
PERFORMANCE

- 30-MHz clock
- 30 MIPS peak, 25 MIPS typical
- Prefetch stage, execution stage
- Most instructions execute in one cycle
- Two-cycle execution for branches
- One-cycle 16- by 16-bit multiply
- DO loop instruction
- Repeat instruction

ON-CHIP PERIPHERALS

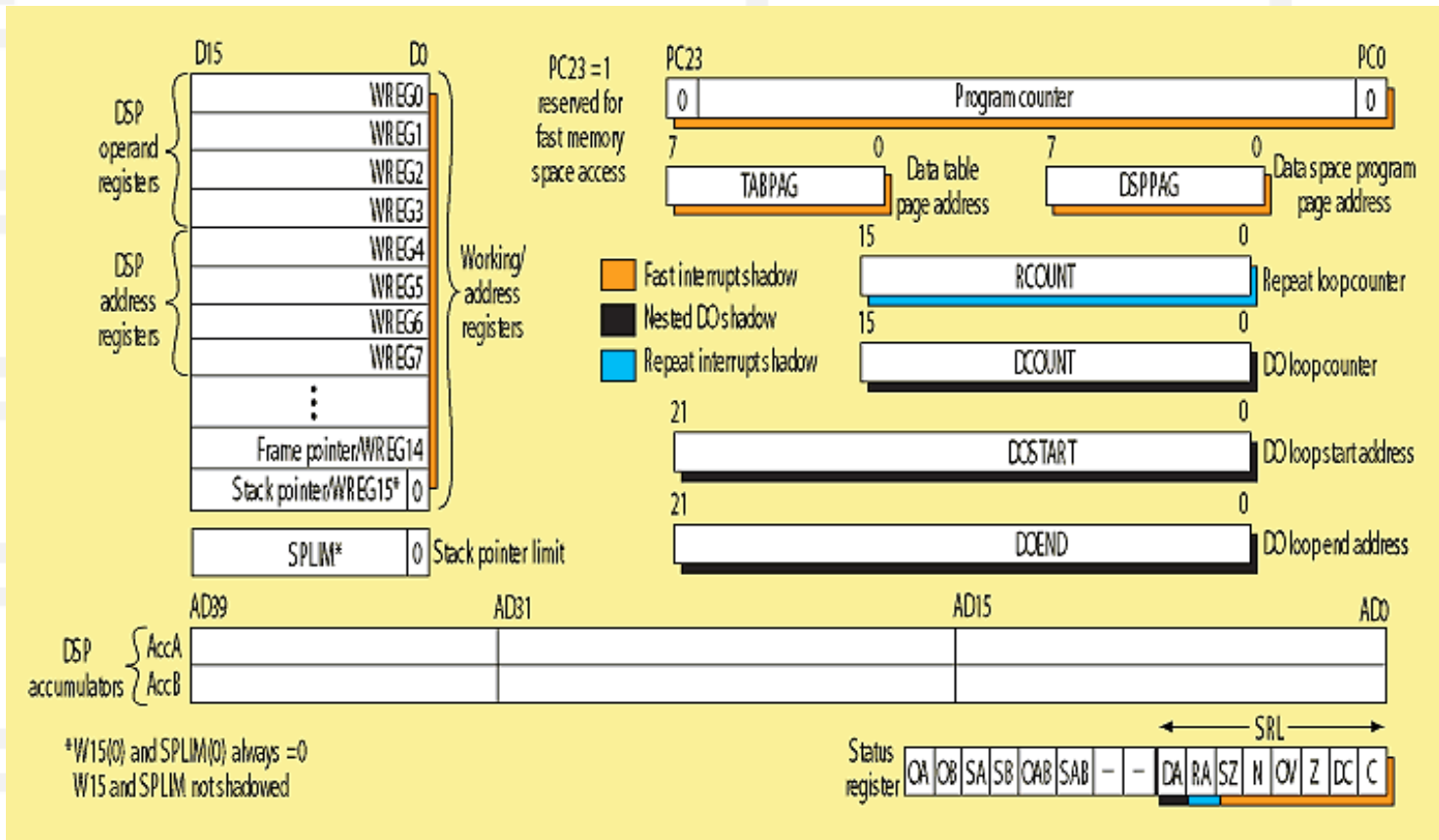
- Four 16-bit timer/counters
- Serial SPI port (8-, 18-bit transfers)
- I²C serial port (100 kHz, 400 kHz, 1 MHz)
- UART, CAN bus interface
- 12-bit ADC (100 ksamples/s, 16-word sample buffer)
- PWM, quadrature encoder

- Codec interface
- Oscillator and power management
- Dynamic flash memory loading

Figure 1

1. The dsPIC integrates a 16-bit MCU with a 16-bit DSP. The DSP executes a MAC in a single cycle. The design includes both X and Y address generator units for DSP addressing.

Figure 2



2. Programmers can work with a set of 16 general-purpose registers to hold DSP operands and address values for DSP processing.

RESOURCES



Discover
Circuits



Power Design
365



Featured
Vendors



E-Cards



Free Design
Resources

Part Finder



Electronic
Design China



Ideas for
Design



Basics of
Design



Events



Design FAQs

brought to you by:



Job Board

PlanetEE Marketplace

[MLC NAND Flash Memory from Toshiba](#)

Get the facts on MLC vs. SLC NAND in a free MLC Application Note

[National's Power ICs](#)

Free samples, High-Voltage Online Seminar, WEBENCH Design

[Atmel \(R\): total security solutions](#)

Learn about our products and qualify for a free development kit.

[DigiKey.com - Unparalleled Service](#)

Access more than 400,000 parts from over 300 vendors



Planet EE Network Home | [Contact Us](#) | [Editorial Calendar](#) | [Media Kit](#) | [Site Feedback & Bugs](#)
Copyright © 2005 Penton Media, Inc., All rights reserved. [Legal](#) | [Privacy](#)