

**A** change of name, a change of management and an injection of finance appear to have worked wonders for Ubicom. Where previously, as Scenix, it developed configurable microcontrollers aimed at such applications as lawn sprinkler controls, today Ubicom is developing network processors for use in leading edge data communications equipment.

Kevin Gee, senior product manager, noted the company's basic business model is to sell chips to major manufacturers of consumer networking gear, including Netgear and D-Link. "But we also do networking stacks and operating systems, releasing this as source code. Our software development kits allow customers to build turnkey home routers out of the box."

Gee said it has partners who take the IP5000 and add more features, VoIP for example. "Slim Devices, with the Squeeze-

**A different hardware and software approach is boosting network processor efficiency.**

**By Graham Pitcher.**

box, is one example – and the company has just been bought by Logitech."

### Ahead of the game

Ubicom believes it has anticipated market developments with its products. "In the past," said Gee, "home networking gear has been designed to move data – email and web pages – around at low rates from one part of the house to another. These are not time critical nor bandwidth hungry."

But times are changing; absolute bandwidth is growing by at least an order

of magnitude. PHY rates are approaching 300Mbit/s, said Gee, translating to a real world data rate of 150Mbit/s.

The reason for this is a change in the type of data being transmitted and the diversity of applications. "We now have VoIP, online games being played on Xbox, Playstation and pcs, video on demand, youtube and Windows Media Center streaming video to the Xbox. All have stringent timing requirements – jitter and packet loss, for example. It's a whole different world to 5Mbit/s," Gee noted.

The problem now is that many cpu architectures are not up to the job – at least in Gee's view. "So we've approached home networking with a clean slate, taking a new hardware and software slant."

That approach supports 10 way multithreading in hardware, a new instruction set, 'lots' of on chip memory and flexible I/O.

Gee said the instruction set had been created specifically for networking. "Our philosophy is not to reuse data," he claimed, and gave the example of moving a bit to highlight the different approach. "On a risc cpu, there's an instruction to move data to a register, where there will be some 'number crunching' requiring

# Nice threads!





two or three instructions. Then, once you're done, the data gets moved back to off chip memory. That takes around five op codes.

### Optimised architecture

"The problem is that architecture was optimised from the point of view of loading data, doing lots of operations on it and then moving it back out – general purpose computing, in other words. For networking, you only want to do one thing."

Ubicom's approach loads data, manipulates it and loads it back to memory using only one instruction. "If you want to increment a bit," Gee continued, "you use one op code and that's a 5:1 efficiency ratio."

He claimed this also brings efficien-

time; and non real time. The former have fixed schedules to service, for example, I/O or dsp functions. The latter execute in a round robin format, servicing general purpose code. "Having these two types of thread gives determinism," Gee believed, "and allows you to use the cpu continually because non real time threads 'fill in the blanks'."

There is 192k of sram on chip, which Gee claims is 'large' in comparison to competitive devices. This is broken down into 16k instruction and 8k data caches. "Because we have fewer cache misses, we don't need big caches."

Even with this design, Ubicom has found it necessary to speed specific tasks and it does this using an approach it calls

roughly equivalent; a 270MHz part will handle 250Mbit/s."

Now, he sees other companies beginning to use the same approach. One particular competitor could handle 0.24 instructions/clock cycle with a previous product and has boosted that to 0.61 in its latest range. "But we're already at 72% overall efficiency," he claimed.


Software plays an equally important role and Gee says the greatest efficiency will be obtained when the implementation matches the work being done. "Networking is inherently event driven," he continued, "and you don't know when the next packet will arrive. Neither do you know what to do with it."

He claimed the 'traditional' approach is a tasking/blocking model, a feature used in Linux, Windows and VxWorks, amongst others. "This uses one task for every logical event. What's dangerous is that tasks can spawn tasks."

He gave handling a TCP request as an example. "One task will handle this request and every time you get to do other work, you need a context switch. This means scratch information has to be flushed and replaced."

Gee is convinced this load/reload process isn't doing anything useful. "You have all these tasks waking up and going to sleep and this consumes resources. Just doing context switching can consume 10% of a cpu's resources."

He contrasts this with the Ubicom approach of run to completion. "Once an event occurs, we do all the work to process that event before doing anything else. And, because these events happen within a thread, we never have to do context switching. This allows us to do 'interesting' things."

Amongst these is the ability to group tasks with similar timescales into threads and provide fine grained control. "We'll measure our system latency in microseconds," Gee concluded, "where competitors measure theirs in milliseconds." 



cies in component count. "Normally, these operations require two chips and two memory chips. We use just one processor and one memory chip."

With 10 threads, Ubicom can allocate tasks to individual threads. "If you need to do VoIP, you can have a codec running in the thread that services it at deterministic intervals," he expanded.

Each thread is also provided with its own bank of registers, so users can switch between threads with zero context switching penalty. "This reduces cache misses," Gee observed.

Threads take two forms: hard real

Fastpath. "Where we use the memory subsystem to manipulate high performance, low latency data, we use hardware and software techniques to move that data through as quickly as possible."

Fastpath also provides quality of service functionality. "We separate data and control planes," Gee explained, "and, for routers, the data plane is the Fastpath and when data is in the Fastpath, we don't copy it and use the most efficient code we can."

Routing brings an efficiency penalty, said Gee. "Generally, you can take a cpu's clock speed and divide by two or three to get its routing performance. So, you'll need a 500MHz cpu to handle 150Mbit/s. Our routing performance is



**Left:** Ubicom's processors are used in D-Link's DIR-655 wireless router.

**Above:** Software development kits allow customers to build turnkey home routers out of the box.