# iReady Makes Silicon Internet Interface

## *Hardware-Only Approach Replaces CPU and Software With Verilog Models*

*by Jim Turley*

Sometimes being different is the key to success, and sometimes it's just—different. Startup company iReady hopes its unusual "Internet Tuner" will put it in the former category. Rather than using software to handle network drivers, protocol stacks, HTML decoding, and e-mail, iReady has developed a set of circuits that do it all in hardware.

iReady promises a simpler, less expensive way to deliver Internet capabilities with vastly lower power consumption. Its hardware-only approach eliminates the microprocessor, software, and much of the memory from embedded systems in exchange for about 100,000 gates of logic. While the iReady approach has advantages in some types of systems, it remains to be seen whether OEMs are willing to give up the comfort and flexibility of standard software and commit to a fixed hardware-only design.

As its initial licensee, the company recently signed Toshiba. The giant Japanese conglomerate will use iReady's technology internally and add the synthesizable hardware modules to its ASIC library to enable Internet features in mobile devices, toys, telephones, and other applications.

### Hardwired Logic Replaces CPU, OS, and Apps

The goal of iReady's nonstandard approach is to reduce the cost of integrating Internet capabilities into high-volume, low-cost consumer devices. The company maintains that Internet connectivity will (and should) be as simple and ubiquitous as AC power connectivity. To make it into the mainstream, Internet "tuners," like radio tuners, need to be cheap, reliable, foolproof, and manufacturable in high volume by anyone.

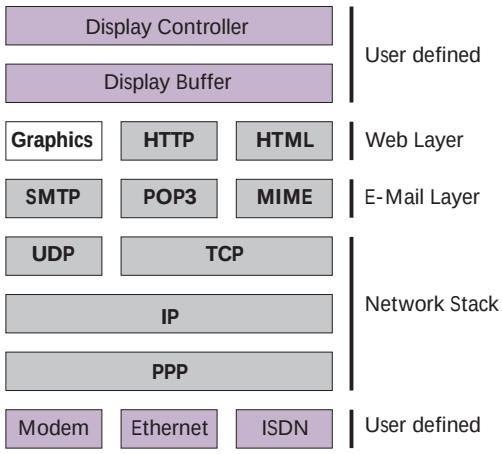


**Figure 1.** iReady's "hardware stack" includes several independent modules. The complete stack allows an entire e-mail and Web terminal to be built entirely in hardware. Shaded portions are available; open modules are in development.

In a normal computer system, making a connection to the Internet involves a small amount of hardware (a modem or network card) and a large amount of software, including an operating system, network drivers, transport-protocol stack, e-mail reader, Web browser, and possibly a Java virtual machine. As many PC users have found to their dismay, this software can easily usurp 16M or more of memory. A powerful 32-bit microprocessor is needed to churn through all this software and deliver acceptable performance.

For simpler devices that don't need the latest Web-browsing capabilities, the iReady design does away with all of the software layers and the microprocessor that runs them. In the company's view, Internet connectivity will become a widespread broadcast commodity—a standardized, expected feature not only in any PC, NC, or information appliance, but in industrial and consumer items as well. These latter devices might use the Internet to download only specific types of information rather than generalized Web pages.

### Modular Structure Allows Customization

As the diagram in Figure 1 shows, iReady's Internet tuner is not a monolithic block but a collection of logic modules. Customers are free to choose modules *à la carte*, integrating just the features required. A dedicated e-mail terminal, for example, might use the IP, TCP, and POP3 layers, while a Web phone might also include the low-level PPP transport layer as well as the HTTP and HTML parsing.

Each module communicates with its neighbors in a byte-serial manner. The PPP block, for example, examines the network-packet header. If it determines the data is an IP packet, it passes the data to the IP decoder for further processing. Similarly, data passes from the IP decoder to the TCP parser and then to the HTTP engine. Web pages are passed as ASCII strings to the HTML block, which interprets the tags, formats the data, and sends the output to a display buffer.

The physical connections at the low end are the OEM's responsibility. As the figure shows, the connection can be a modem, Ethernet card, ISDN interface, or some other medium. iReady does not supply the logic for these functions, although they could be integrated into the same ASIC as the rest of the Internet tuner.

### Modest Gate Count, Memory Requirements

To be integrated economically, the iReady logic must be small enough that it does not dominate the cost of the ASIC. As Table 1 shows, the gate count of each of the logic layers is large but not overwhelming, numbering a few tens of

thousands of gates. In a typical 0.5-micron ASIC process, the complete 110,000-gate Web stack would fill about 22 $mm^2$. While 100,000-gate ASICs are not trivial, they are not the daunting undertakings they were a few years ago. Adding 100,000 gates to a moderately sized ASIC is probably still more economical than adding a microprocessor and support logic. This gate count is also within reach of big FPGAs, such as Lucent's Orca series.

The only part left out of this equation is memory. Apart from physical-medium attachments, the iReady logic also requires some external memory to operate. Along with the gate-count budget, Table 1 lists the memory requirements of each logic layer. In the case of the network stack (PPP, IP, TCP, and UDP layers), the 6K memory allocation includes three network sockets. Additional sockets, up to 16 total, require another 3K per socket.

Given iReady's—and other vendors'—push toward integration, it would be natural to merge this memory onto the same ASIC device as the logic. Compared to a microprocessor, the design's modest clock rate would not tax even the mediocre logic performance of the current crop of merged DRAM/logic processes (see MPR 8/4/97, p. 19).

There's much more logic than memory in iReady's design; 86K of DRAM would need very little space in a 0.25-micron ASIC process. Thus, a logic-optimized process would be better suited than a modified memory process. LSI Logic's G10 (0.35-micron) or G11 (0.25-micron) processes would be ideally suited but are still expensive, being much faster than required to meet iReady's modest data rates.

## Clock Speed Is in the Low Kilohertz

To run without CPU intervention, iReady's entire hardware stack is designed as a byte-serial state machine. This has the advantage of allowing the same design to work at various data rates while keeping power consumption to a minimum.

The logic is all clocked and synchronous. When no data is being received (or transmitted), the user can slow the clock or stop it to save power. When data is being transferred, the logic must run at a minimum frequency to meet memory-bandwidth requirements. There is no advantage to running the clock faster than necessary.

Table 2 gives typical minimum clock rates for standard bit rates. As the table shows, the logic needs to switch at only 40–50% of the rate of the incoming data under ideal circumstances, using synchronous SRAMs for buffer memory.

| | Gates | Power* | Memory |
|---|---|---|---|
| **Network stack** | 34,000 | 131 µW | 6K |
| **E-mail** | 28,000 | 58 µW | 16K |
| **Web browsing** | 48,000 | 174 µW | 64K |
| **Total** | 110,000 | 363 µW | 86K |

**Table 1.** The three major layers of the iReady hardware stack total about 110,000 gates of synthesized logic and use about 86K of memory (assuming three TCP sockets). *power measured at 28.8 kbps. (Source: iReady)

The tables also highlight the iReady logic's dependence on memory speed. Internal frequencies must be almost twice as fast when the logic is using asynchronous memory instead of synchronous memory. This is because of the more complicated state machine required to deal with an asynchronous interface.

The logic design is the same regardless of the transmission data rate. The same circuit can handle 14.4-kbps modems and 100-Mbit/s Fast Ethernet transmissions (given the appropriate interface) just by increasing its clock rate. The only requirement is that the manufacturing process must be fast enough to accommodate the expected data rate.

The moderate gate count and modest clock rates allow customers to prototype their designs in FPGAs, such as those from Altera or Xilinx. iReady itself has done this and encourages customers to do the same.

## Power Usage Is Essentially Zero

As Table 2 shows, the power consumption of the complete iReady hardware stack is quite low, just fractions of a milliwatt at 28.8 kbps. These figures assume a conservative 0.8-micron manufacturing process and a 5-V power supply. Obviously, lowering the voltage or moving to a more modern fab process would reduce power ever further.

Such extremely low power consumption is iReady's strongest argument in favor of its hardware-only approach. It is at least two orders of magnitude lower than any microprocessor capable of doing comparable work. Toshiba's new 1904AF MIPS processor, for instance, consumes a mere 35 mW at 20 MHz (see MPR 8/4/97, p. 9), but can barely handle a 14.4-kbps soft modem, much less TCP/IP and HTML. Even the fast but low-power SA-110 burns nearly 1 W at 233 MHz, 2,500 times the power requirement of the iReady tuner.

Power consumption this low is essentially zero. Any useful device will consume more energy in its LCD display, power conversion, modem, or RF interface than the iReady logic would require (at least, at moderate data rates). Intel's 386EX uses more power with its clock stopped than the iReady logic uses when it's running.

The power consumption of the iReady tuner may approach the lowest theoretically possible, but that alone may not be enough to win over customers in every case. The power consumption of many wireless devices is dominated by the

| Bit Rate | Clock Speed (typical) | |
|---|---|---|
| | Sync SRAM | Async SRAM |
| **14.4 Kbit/s** | 6 KHz | 12 KHz |
| **28.8 Kbit/s** | 12 KHz | 24 KHz |
| **128 Kbit/s** | 50 KHz | 90 KHz |
| **10 Mbit/s** | 5 MHz | 9 MHz |
| **100 Mbit/s** | 40 MHz | 70 MHz |

**Table 2.** Although the logic design doesn't change, the fabrication process must be able to accommodate certain minimum clock rates for various standard data rates.

### For More Information

iReady is currently licensing Verilog models of its i-1000 logic blocks; the i-2000 collection is expected to be available in 1Q98.

For more information, contact iReady (San Jose, Calif.) at 408.325.8720 or visit *www.ireadyco.com*.

RF, not the digital, circuits. Further reducing the digital component has little payoff. The iReady approach is thus better suited for "all-digital" systems, such as PHS and PCS (popular in Asia and Europe and growing in popularity in North America). Larger devices, such as television set-top boxes, have access to AC power, so power consumption is less of a factor. Even small, handheld devices can (and do) draw power from the telephone line. For these devices, low power consumption is a benefit, but it is not always the most important one.

## Hard Logic Eliminates Possibility of Upgrades

One concern potential customers may have with iReady's solid-state approach is the loss of flexibility. With no software, the system cannot be upgraded, either in the field or during production. Every feature, from network protocols to HTML tags, is fixed in silicon. With some Internet standards, such as HTML, undergoing frequent revision, users may be nervous about committing to a fixed set of features.

The HTML standard, for example, currently stands at revision 3.2, which iReady supports in its HTML module. But that standard changes regularly, with new revisions historically coming every few months. The HTML module is one of iReady's most complex. It would take the company several months to revise it and several more for an OEM's revised ASIC to enter production. By that time, a new HTML standard may be adopted, and the cycle starts anew.

On the other hand, iReady is not aiming at WebTV replacements but at simpler devices that rely on low-level Internet standards to communicate among themselves. Networked vending machines, for example, or portable wireless terminals with small text displays are closer to iReady's target, and such devices aren't dependent on the latest bells and whistles in the newest HTML definition.

iReady believes that the applications for its technology fall within the more traditional consumer-electronics space, where features and capabilities are limited and well defined. Such devices tend to have stable features for specific activities. Production runs are generally large enough—and product cycles short enough—that committing to a particular feature set over the product's life is not an issue. And, unlike computers, consumer-electronics products aren't updated; in that market, it's a recall.

## Four Logic Bundles Planned

iReady wants to address certain applications with preconfigured "bundles" of its logic blocks, targeting, say, fax makers, printer manufacturers, or cell-phone producers. The company also offers four modular bundles of its logic. As Table 3 shows, each collection is a superset of the previous ones, intended to serve particular markets.

The first collection, dubbed i-1000, is currently available. It includes the TCP/IP protocol stack, POP3 and SMTP for e-mail, the HTTP engine, and an HTML decoder for interpreting Web pages (text only; with no graphics filters, the HTML decoder simply discards any image tags). The i-1000 package is aimed at devices with unsophisticated displays, such as cellular telephones, or with no displays, such as networked vending machines.

After starting development in February 1996, iReady received working silicon of its network-level (PPP, IP, UDP, and TCP) modules at the end of 1996; the e-mail and HTML blocks arrived in July of this year. The company expects the next package, the i-2000, to be available in 1Q98; the others will follow, with no fixed schedule announced. The i-2000 will include support for more sophisticated displays, including JPEG and GIF filters.

Toshiba and iReady are both members of the Virtual Socket Initiative (VSI) consortium (see MPR 9/16/96, p. 15), a group advocating the exchange of mutually compatible ASIC modules. Toshiba plans to offer a VSI-compliant version of the i-1000 package later this year, to go with its VSI-compliant MIPS core (used in the 1904AF microprocessor; see MPR 8/4/97, p. 9). This would theoretically allow ASIC customers to add the Internet tuner to other hard macros without first synthesizing the entire design.

| i-1000 | i-2000 | i-3000 | i-4000 |
|---|---|---|---|
| | | **Streaming video** | **Java** |
| | | **Telephony** | Streaming video |
| | **JPEG, GIF** | JPEG, GIF | Telephony |
| HTTP, HTML 3.2 | HTTP, HTML 3.2 | HTTP, HTML 3.2 | JPEG, GIF |
| SMTP, POP3 | SMTP, POP3, **IMAP4** | SMTP, POP3, IMAP4 | HTTP, HTML 3.2 |
| TCP, IP, PPP | TCP, IP, PPP | TCP, IP, PPP | SMTP, POP3, IMAP4 |
| | | | TCP, IP, PPP |

**Table 3.** iReady plans four progressively elaborate collections of its logic macros. Currently, the i-1000 suite, for systems with simple display needs, is the only one available. The i-2000, due in 1Q98, adds JPEG and GIF decompression.

## Company Seeks Four Benefits

The iReady design technique makes the most sense for devices that will feature Internet connectivity as a standard function. The great majority of consumer-electronics items are built around custom silicon, and in these cases, integrating iReady's logic into the circuit might avoid a complete redesign of the product into a microprocessor-based system.

Slightly more exotic appliances, such as text pagers and "feature phones" with

LCD displays and multiple lines, generally use 8-bit microcontrollers in addition to their custom logic. Upgrading these devices with e-mail capabilities would involve a significant processor upgrade and software effort, with a commensurate increase in RAM and ROM capacity. Adding hardwired e-mail features would probably be less expensive than upgrading the processor and memory, and less risky than developing and debugging new software, assuming iReady already had the modules available.

The company touts four advantages: low cost, low power, high performance, and ease of use. Although it's difficult to make objective comparisons, iReady's cost target seems to be met, assuming the customer already has significant ASIC development under way. The cost of the extra gates, plus iReady's royalty—a few dollars in volume—is likely to be less than the cost of a microprocessor, its associated memory, and a possible royalty for an operating system or application software.

The low-power claim certainly holds true. At less than one-third of a milliwatt, there's no conceivable combination of 8-, 16-, or 32-bit microprocessor and memory that would even come close. For battery-powered toys, terminals, and telephones, this is a powerful argument in iReady's favor.

## Overturning the Conventional Wisdom

iReady's unusual approach to what seemed a well-understood and well-handled function—providing Internet connectivity, e-mail, Web browsing, etc.—has potential. In a desktop computer, where a high-end microprocessor, lots of memory, and lots of software are already available, adding a little more software to run a Web browser is entirely logical. But to break into low-cost, high-volume devices that are operated almost unconsciously, something cheaper and simpler is needed.

That need can be met by cheaper, simpler (than Pentium) processors, of which there are many. Software developers like WebTV, Navio, Teknema (see MPR 7/8/96, p. 10), and Diba (see MPR 8/25/97, p. 4), have demonstrated cheaper, simpler software. Obviously, television-based Web browsers can be sold for less than $200 using conventional means, but that's a far cry from the "no cost" level needed to make Internet capabilities standard in industrial and consumer applications. For the time being, iReady must concentrate on applications where even a moderately priced microprocessor is too expensive or too power-hungry, and where Internet connectivity would have an immediate payoff.

Ironically, iReady's hardware implementation comes just as software-only modems become the norm, because they're cheaper than dedicated hardware. But soft modems assume a powerful CPU running the end application. In such systems, adding a little more software is an inexpensive way to add valuable features. Only in systems with no such high-end processor do hardwired features make sense. But in the end, iReady's intriguing technology and fascinating engineering challenge may just turn out to be—different. MPR