6 page(s) will be printed. Back

Record: 1

Configurable computing. (cover story) By: Villasenor, John; Mangione-Smith, William H.. Scientific American, Jun97, Vol. 276 Issue 6, p66, 6p, 4 diagrams, 2c; (*AN* ***9709221451***)

Database: Academic Search Premier

CONFIGURABLE COMPUTING

Computing that modify their hardware circuits as they operate are opening a new era in computer design. Because they can filter data rapidly, they excel at pattern recognition, image processing and encryption

Computer designers face a constant struggle to find the right balance between speed and generality. They can build versatile chips that perform many different functions relatively slowly, or they can devise application-specific chips that do only a limited set of tasks but do them much more quickly. Microprocessors (such as the Intel Pentium or Motorola PowerPC chips commonly found in personal computers) are general purpose: programming instructions encoded in binary format can lead a microprocessor through virtually any logical or mathematical operation a programmer can conceive. The Intel Pentium, for example, was never designed specifically to execute either Microsoft Word or the computer game DOOM, but it can run both.

In contrast, custom hardware circuits, often known as application-specific integrated circuits (ASICs), provide precisely the functionality needed for a specific task. By carefully tuning each ASIC to a given job, the computer designer can produce a smaller, cheaper, faster chip that consumes less power than a programmable processor. A custom graphics chip for a PC, for instance, can draw lines or paint pictures on the screen 10 or 100 times as quickly as a general-purpose central processing unit can.

As designers make their choices between versatility and speed, they must also confront the issue of cost. A well-designed ASIC will solve the specific problem for which it was designed, but probably not a slightly modified problem introduced after the ASIC design is finished. Furthermore, even if a modified ASIC can be developed for the new problem, the original hardware circuits may be too highly customized to be reused in successive generations. As a result, the engineering effort required to design and build an ASIC must be amortized over a relatively small number of units.

A new development in integrated circuits offers a third option: large, fast, field-programmable gate arrays, or FPGAs--highly tuned hardware circuits that can be modified at almost any point during use. FPGAs consist of arrays of configurable logic blocks that implement the logical functions of gates. Logic gates are like switches with multiple inputs and a single output. They are used in digital circuits to perform basic binary operations such as AND, NAND, OR, NOR and XOR. In most hardware that is used in computing today, the logical functions of gates are fixed and cannot be modified. In FPGAs, however, both the logic functions performed within the logic blocks and the connections between the blocks can be altered by sending signals to the chip. These blocks are structurally similar to the gate arrays used in some ASICs, but whereas standard gate arrays are configured during manufacture, the configurable logic blocks in FPGAs can be rewired and reprogrammed repeatedly, long after the integrated circuit has left the factory.

The key that has opened the door to configurable computing is the design of new FPGAs that can be configured extremely quickly. The earliest field-programmable arrays required several seconds or more to change their connections--perfectly suitable for engineers who wanted to test alternative circuit designs or for companies that sold devices that might need occasional upgrading. Newer FPGAs can be configured in one millisecond, and we expect to see devices with configuration times as low as 100 microseconds within two years. Ultimately, computing devices may be able to adapt their hardware almost continuously in response to changes in the input data or processing environment.

There are many variations on FPGA design, but the basic structure consists of a large number of configurable logic blocks and a programmable grid of connections that can link those blocks in any pattern the designer chooses. Those FPGAs that are coarse "rained have a small number of powerful configurable logic blocks; those with a finer-grained structure have many simple blocks. A single element in a coarse-grained FPGA might be capable of adding or comparing two numbers. One block in a fine-grained device might be capable only of comparing two binary digits--in effect, it would be a single logic gate. A designer might choose to start with either a coarse-or fine-grained chip depending on the application at hand and the amount of time available for building complex subsystems from scratch.

Computing devices can make use of configurable elements in many different ways. The least demanding technique is to switch between functions on command--the hardware equivalent of quitting one program and then running another. Slow reconfiguration, on the order of several seconds, may well be acceptable in such an application. Faster programming times permit dynamic design swapping: a single FPGA performs a series of tasks in rapid succession, reconfiguring itself between each one. Such designs operate the chip in a time-sharing mode and swap between successive configurations so rapidly that it appears the FPGA is performing all its functions at once.

Using this approach, we have built a single-chip video transmission system that reconfigures itself four times per video frame. It thus requires only a quarter of the hardware that would be needed for a fixed ASIC. The FPGA first stores an incoming video signal m memory, then applies two different image-processing transformations and finally transforms itself into a modem to send the signal onward.

The most challenging and potentially most powerful form of configurable computing involves the hardware reconfiguring itself on the fly as it executes a task, refining its own programming for improved performance. An image-recognition chip might tune itself in response to a tentative identification of the object it is looking at: if an image contained a car or a truck, parts of the circuitry originally intended for tracking high-speed aircraft or slow-moving people could be reconfigured to focus instead on land vehicles. For some applications, such a radical departure from traditional computer design, in which the hardware is specified at the outset, could make for much faster and more versatile machines than are possible with either general-purpose microprocessors or custom chips.

Cutting Critical Hardware

One of the most promising applications for configurable computing involves pattern matching. Pattern matching is used in tasks such as handwriting recognition, face identification, database retrieval and automatic target recognition. A fundamental operation of pattern matching involves comparing an input set of bits (representing an image, a string of characters or other data) with a set of templates corresponding to the possible patterns to be recognized. The system declares recognition when the number of input bits that match bits in the template exceeds some threshold.

In the case of target recognition-a military application that drove some of our initial work--the greatest challenge is the rapid comparison of an input image to thousands of templates. A template could represent, for example, a front or side view of a specific type of vehicle. Each image typically contains thousands of pixels (picture elements), and a target could appear at any position within an image. To recognize targets fast enough for military applications, a system needs to perform comparisons at the rate of several trillion operations per second, because all the pixels in the input image must be compared with all the pixels in many templates.

With support from the Defense Advanced Research Projects Agency (DARPA), we have built a prototype recognition system with configurable hardware that achieves significant hardware savings by tuning itself to each template in turn. Many of the pixels in a typical template do not contribute to the matching results [see bottom illustration on opposite page], and so the configurable computing machine could simply omit them from its calculations. A convertional system could not easily pare itself down in a similar way, because the pixels to be ignored differ from template to template. One can go further in exploiting the flexibility of configurable machines by tuning the hardware to take advantage of similarities among templates. The configurable hardware can process a set of templates in parallel, using only one comparison unit for each pixel whose value is the same for templates in that set. For example, rather than having eight separate hardware circuits consider a certain pixel for eight different templates, a single circuit can consider the pixel and then propagate its outcome to the seven other templates.

Most recently, we have built a prototype encryption system (also funded by DARPA) that takes advantage of configurable hardware. An FPGA implements the Data Encryption Standard (DES), which uses 56-bit-long keys to encrypt 64-bit-long blocks of data. (A key in encryption is a number used to scramble or unscramble a confidential message.) DES encryption usually proceeds in two steps: subkey scheduling and data processing. In the first step, a set of rotations and permutations translates the 56-bit encryption key into a series of 16 subkeys. Each subkey then processes the data in a separate round; a full set of 16 rounds encrypts or decrypts each 64-bit block. When the computer deals concurrently with multiple users, each dialogue between users must have a distinct key, and the encryption hardware will change keys as parts of messages arrive for different users.

In many applications of DES, the encryption key remains constant while a long block of data passes through the data path. For example, if two people are communicating over a secure network, they exchange a secure encryption key once and then use that key throughout the duration of their dialogue to generate the subkeys for each round of encryption or decryption. Some ASICs are designed to handle only one kind of encryption algorithm, such as DES; others--such as programmable digital signal processors--are capable of implementing many encryption algorithms.

With a configurable chip, the software can calculate the subkey values once, and the data-processing circuitry can be optimized for those specific subkeys. This approach allows the subkey-scheduling hardware to be completely removed from the system. These savings have allowed us to implement the DES algorithm in a 13,000-gate FPGA, instead of the 25,000-gate circuit previously required. When the encryption key must be changed, the user can quickly specify a new circuit, customized to the new key, and download it to the FPGA.

The target-recognition and encryption prototypes we have built help illustrate the enormous flexibility that arises when the hardware in a computer can be customized to a diverse and changing set of external data. There are many other applications that could benefit from the ability to modify the computation hardware in this manner, including digital communications, design automation and digital filtering for radar.

The Future of Configurable Computing

Configurable computing is still an extremely young field. Although Gerald Estrin of the University of California at Los Angeles proposed configurable computing in the late 1960s, the first demonstrations did not occur until a few years ago, and current FPGAs, with up to 100,000 logic elements, still do not come close to exploiting the full possibilities of the technique. Future FPGAs will be much larger; as with many other integrated circuits, the number of elements on a single FPGA has doubled roughly every 18 months. Before the decade is out, we expect to see FPGAs that have a million logic elements. Such chips will have much broader application, including highly complex communications and signal-processing algorithms.

Academic researchers and manufacturers are overcoming numerous other design limitations that have hindered the adoption of configurable computing. Not all computations can be implemented efficiently with today's FPGAs: they are well suited to algorithms composed of bit-level operations, such as pattern matching and integer arithmetic, but they are ill suited to numeric operations, such as high-precision multiplication or floating-point calculations. Dedicated multiplier circuits such as those used in microprocessor and digital signal chips can be optimized to perform more efficiently than multiplier circuits constructed from configurable logic blocks in an FPGA. Furthermore, FPGAs currently provide very little on-chip memory for storage of intermediate results in computations; thus, many configurable computing applications require large external memories. The transfer of data to and from the FPGA increases power consumption and may slow down the computations.

Fortunately, researchers are developing advanced FPGAs that contain memory, arithmetic processing units and other special-purpose blocks of circuitry. Andre DeHon and Thomas F. Knight, Jr., of the Massachusetts Institute of Technology have proposed an FPGA that stores multiple configurations in a series of memory banks. In a single clock cycle, which is on the order of tens or hundreds of nanoseconds, the chip could swap one configuration for another configuration without erasing partially processed data.

Meanwhile Brad L. Hutchings of Brigham Young University has used configurable computing to build a dynamic instruction set computer (DISC), which effectively marries a microprocessor to an FPGA and demonstrates the potential of automatic reconfiguration using stored configurations. As a program runs, the FPGA requests reconfiguration if the designated circuit is not resident. DISC allows a designer to create and store a large number of circuit configurations and activate them much as a programmer would initiate a call to a software subroutine in a microprocessor.

The Colt Group, led by Peter M. Athanas of Virginia Polytechnic Institute and State University, is investigating a run-time reconfiguration technique called Wormhole that lends itself to distributed computing. The unit of computing is a stream of data that creates custom logic as it moves through the reconfigurable hardware.

John Wawrzynek and his colleagues at the University of California at Berkeley are developing systems that combine a microprocessor and an FPGA. Special compiler software would take ordinary program code and automatically generate a combination of machine instructions and FPGA configurations for the fastest overall performance. This approach takes advantage of opportunities to integrate a processor and an FPGA on a single chip.

FPGAs will never replace microprocessors for general-purpose computing tasks, but the concept of configurable computing is likely to play a growing role in the development of high-performance computing systems. The computing power that FPGAs offer will make them the devices of choice for applications involving algorithms in which rapid adaptation to the input is required.

In addition, the line between programmable processors and FPGAs will become less distinct: future generations of FPGAs will include functions such as increased local memory and dedicated multipliers that are standard features of today's microprocessors; there are also next-generation microprocessors under development whose hardware supports limited amounts of FPGA-like reconfiguration. Indeed, just as computers connected to the Internet can now automatically download special-purpose software components to perform particular tasks, future machines might download new hardware configurations as they are needed. Computing devices 10 years from now will include a strong mix of software-programmable hardware and hardware-configurable logic.

Programmable Circuitry

Programmable circuits (blue lines) in a field-programmable gate array (FPGA) can be created or removed by sending electronic signals to gates in the logic elements. A built-in grid of circuits arranged in columns and rows allows the designer to connect a logic element to other logic elements or to an external memory or microprocessor. The logic elements are grouped in blocks that perform basic binary operations such as AND, OR and NOT; several firms, including Xilinx and Altera, have developed devices with the capability of 100,000 equivalent gates.

John Villasenor and William H. Mangione-Smith

PHOTOS (COLOR): RECONFIGURABLE LOGIC DEVICE (right) has circuits that can be partially or completely changed while it is operating. Scanning electron micrographs above (magnification 3,000 x) show three fixed-circuit metal layers of a flip-flop cell in a field-programmable gate array. The flip-flop cell contains four memory cells with switches (bottom, colored areas) that can be opened or closed electronically to produce a conductive path.

PHOTO (COLOR): Programmable Circuitry

DIAGRAMS: CONFIGURABLE COMPUTING architectures combine elements of general-purpose computing and application-specific integrated circuits (ASICs). The general-purpose processor operates with fixed circuits that perform multiple tasks under the control of software. An ASIC contains circuits specialized to a particular task and thus needs little or no software to instruct it. The configurable computer can execute software commands that alter its FPGA circuits as needed to perform a variety of Jobs.

DIAGRAM: PATTERN MATCHING is one of the most promising applications for configurable computing. FPGAs are particularly well suited for pattern matching because they can carry out comparisons using simple additions and shifts of individual bits. Pattern matching begins with a candidate image and the target image for comparison (left). The images are converted into binary templates, and the FPGA compares bits in the candidate image with those in the target template using a logical AND circuit. Another part of the device is configured as an adder to count the total number of pixels that are "on" (represented by 1's) in both templates. The higher the number, the better the match.

DIAGRAM: PROTOTYPE VIDEO COMMUNICATIONS SYSTEM uses a single FPGA to perform four functions that typically require separate chips. A memory chip stores the four circuit configurations and loads them sequentially into the FPGA. Initially, the FPGA's circuits are configured to acquire digitized video data. The chip is then rapidly reconfigured to transform the video information into a compressed form and reconfigured again to prepare it for transmission. Finally, the FPGA circuits are reconfigured to modulate and transmit the video information. At the receiver, the four configurations are applied in reverse order to demodulate the data, uncompress the image and then send it to a digital-to-analog convert" so it can be displayed on a television screen. The prototype is capable of processing eight frames per second.

DIAGRAM: HYBRID-ARCHITECTURE COMPUTER combines a general-purpose microprocessor and reconfigurable FPGA chips. The dynamic instruction set computer (DISC), built at Brigham Young University to demonstrate mixed architecture advantages, consists of two FPGAs and memory on a circuit board connected to a personal computer. The controller FPGA loads circuit configurations stored in the memory onto the processor FPGA in response to the requests of the operating program. If the memory does not contain a requested circuit, the processor FPGA sends a request to the PC host, which then loads the configuration for the desired circuit.

Further Reading

A DYNAMIC INSTRUCTION SET COMPUTER. M. J. Wirthlin and B. L. Hutchings in Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, April 1995. IEEE Computer Society Press, 1995.

A FIRST GENERATION DPGA IMPLEMENTATION. E. Tau, D. Chen, I. Eslick, J. Brown and A. DeHon in Proceedings of the Third Canadian Workshop on Field-Programmable Devices, pages 138-143; May 1995.

CONFIGURABLE COMPUTING SOLUTIONS FOR AUTOMATIC TARGET RECOGNITION. John Villasenor, Brian Schoner, Kang-Ngee Chia, Charles Zapata, Hea Joung Kim, Chris Jones, Shane Lansing and Bill Mangione-Smith in 4th IEEE Symposium on FPGAs for Custom Computing Machines (FCCM '96) April 1996. Available from http://www.computer.org/cspress/catalog/pr07548.htm

WORMHOLE RUN-TIME RECONFIGURATION. Ray Bittner and Peter Athanas in FPGA '97: ACM/SIGDA International Symposium on Field Programmable Gate Arrays. Special Interest Group on Design Automation (SIGDA), ACM, 1997.

~~~~~~~~

by John Villasenor and William H. Mangione-Smith
~~~~~~~~

JOHN VILLASENOR and WILLIAM MANGIONE-SMITH are on the faculty of the electrical engineering department at the University of California, Los Angeles. Villasenor received a Ph.D. in electrical engineering from Stanford University in 1989 and was with the Jet Propulsion Laboratory from 1990 to 1992. He moved to U.C.L.A. in 1992 and currently conducts research on image and video communications and on configurable computing. Mangione-Smith received a Ph.D. in computer engineering from the University of Michigan in 1991 and then worked for Motorola until 1995, becoming the systems architect for a second-generation personal wireless communicator. His research interests include low-power and high-performance computing engines, processors, systems architecture and systems software.

Source: Scientific American, Jun97, Vol. 276 Issue 6, p66, 6p
Item: 9709221451