

32. Caspi, E., Chu, M., Huang, R., Yeh, J., Markovskiy, Y., Wawrzynek, J., and André DeHon, A., Stream computations organized for reconfigurable execution (SCORE), in *10th Intl Conference on Field Programmable Logic and Applications, LNCS*, Hartenstein, R.W. and Gruenbacher, H., Eds., p. 1896, 2000 also http://www.cs.berkeley.edu/projects/brass/documents/score_tutorial.html.
33. Callahan, T.J., Hauser, J.R., and Wawrzynek, J., The Garp architecture and C compiler, *IEEE Computer*, 33, 4, 62, 2000.
34. Callahan, T.J. and Wawrzynek, J., Instruction level parallelism for reconfigurable computing, in *8th International Workshop Field-Programmable Logic and Applications, LNCS*, Hartenstein, R. and Keevalik, A., Eds., p. 1482, 1998.
35. Tsu, W. et al., in *Proc. 7th International Symposium on Field Programmable Gate Arrays*, Trimberger, S., Ed., 1999.
36. Diessel, O. and Milne, G., Compiling process algebraic descriptions into reconfigurable logic, in *Proc 15th IPDPS Workshops, LNCS 1800*, Rolim, J. et al., Eds., p. 916, 2000.

21.2 Using Configurable Computing Systems

Danny F. Newport and Don Bouldin

21.2.1 Definitions

AQ1

Configurable computing systems: Systems that use reprogrammable logic components, typically field-programmable gate arrays (FPGAs), to implement a specialized instruction set and arithmetic units to improve the performance of a particular application. These systems can be reconfigured, enabling the same hardware resource to be reused depending on its interaction with external components, data dependencies, or algorithm requirements.

Configuration time: Time required to program an FPGA or configurable computing system with a given configuration. This time varies from hundreds of nanoseconds to seconds, depending on the system and the FPGAs that are used in the system.

Field-programmable gate array: Integrated circuit containing arrays of logic blocks and programmable interconnect between these blocks. The logic blocks can be configured to implement simple or complex logical functions and can be changed as required. Example functions are registers, adders, and multipliers. The programmable interconnect permits the construction of even more complex functions or systems.

FPGA: Acronym for field-programmable gate array.

Reconfigurable computing systems: Alternate term for configurable computing systems. This term is usually used to indicate that the system can be reconfigured at any time for some desired function.

21.2.2 Introduction

Configurable computing systems use reprogrammable logic components, which are now capable of providing more than 10 million logic gates on a single chip. These systems can be reconfigured at runtime, enabling the same hardware resource to be reused depending on its interaction with external components, data dependencies, or algorithm requirements.

In essence, a specialized instruction set and arithmetic units can be configured as desired by an application designer on an as-needed basis to achieve optimal performance. The location of configurable components within a computing system is one of the keys to achieve maximum efficiency and performance. A variety of architectures driven by the location of configurable components are described in a later section.

21.2.3 Configurable Components

Before describing various architectures of configurable computing systems, an understanding of the internal structure of FPGAs, the major component of a configurable computing system is necessary.

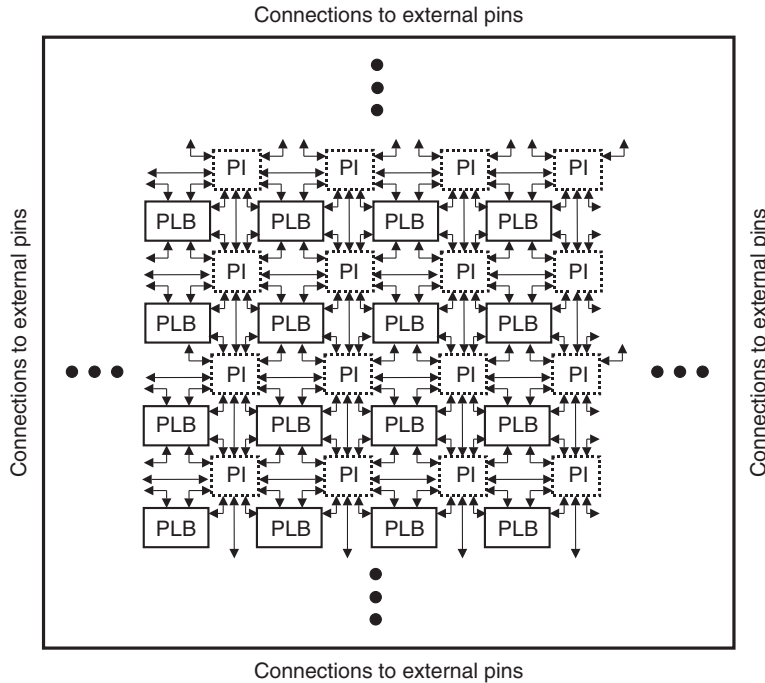


FIGURE 21.7 Basic internal structure of an FPGA.

In 2006, the top two FPGA vendors were Altera Corporation and Xilinx, Inc. The FPGA products from these, and other vendors, differ in their internal structure and programming. However, the basic internal structure for FPGAs can be illustrated as shown in Fig. 21.7. Note that the PLB blocks are programmable logic blocks and the PI blocks are programmable interconnect. A current trend among the FPGA vendors is to also include RAM or a fixed microprocessor core on the same integrated circuit as the FPGA. This enables even greater system flexibility in a design.

The means by which the logic blocks and interconnect are configured for specific functions are of a proprietary nature and specific to each vendor's FPGA families. In general terms, the logic blocks and interconnect have internal structures that "hold" the current configuration and when presented with inputs will produce the programmed logic outputs. This configuration is FPGA specific and contained in a vendor-specific file. A specific FPGA is programmed by downloading the information in this file through a serial or parallel logic connection.

The time required to configure an FPGA is known as the configuration time.

Configuration times vary based on the FPGA family and the size of the FPGA. For a configurable computing system composed of several FPGAs, the configuration time is based not only on the configuration time of the individual FPGAs but also on how all the FPGAs are configured. Specifically, the FPGAs could be configured serially, in parallel, or a mixture of serial and parallel depending upon the design of the system. Thus, this time can vary from hundreds of nanoseconds to seconds. This directly impacts the types of applications that have improved performance on a particular configurable computing system. A configurable computing system that has a configuration time on the order of seconds is best suited for applications that do not require reconfigurations "on the fly," i.e., applications with a single configuration associated with them or the ones that are pipelined with slow pipeline stages. On the other hand, a configurable computing system that has a very short configuration time can be used for the same applications as one with a slower configuration time and applications that require on-the-fly reconfiguration.

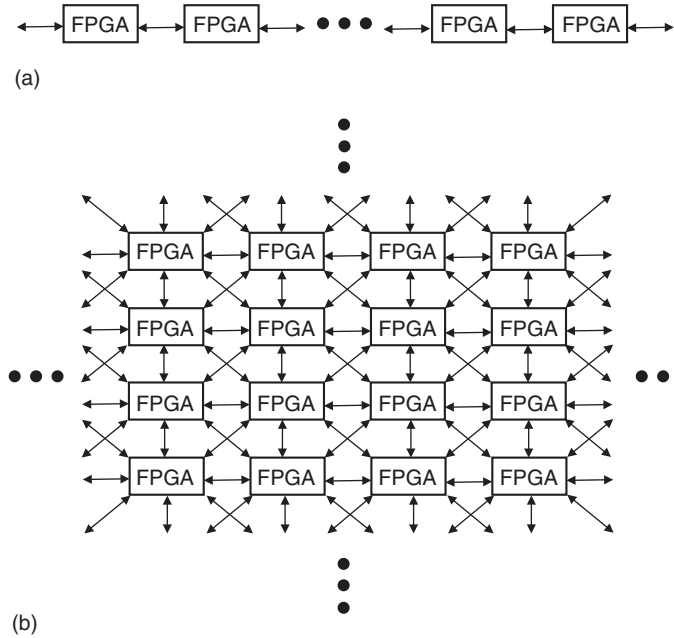


FIGURE 21.8 Basic architectures for multiple FPGAs.

As implied previously, the configurable component of a configurable computing system can be composed of a single FPGA or multiple FPGAs. Many architectures are used for a configurable component composed of multiple FPGAs. Figure 21.8 illustrates the basic architectures from which most of these architectures would be derived. Note that these architectures are very similar, or identical, to those used for parallel processing systems. As a matter of fact, many of the paradigms used in configurable computing systems are derived from parallel processing systems. In many cases, a configurable computing system is the hardware equivalent of a software parallel processing system. Figure 21.8(a) is a pipelined architecture with the FPGAs hardwired from one to the other. This type of architecture is well suited for functions that have streaming data at specific intervals. Note that variations of this architecture include pipelines with feedback, programmable interconnect between the FPGAs, and RAM associated with each FPGA. Figure 21.8(b) is an array of FPGAs hardwired to their nearest neighbors. This type of architecture is well suited for functions that require a systolic array. Note, as with the pipelined architecture, that variations of this architecture include arrays with feedback, programmable interconnect between the FPGAs, and RAM associated with each FPGA. Also note that an array of FPGAs is very similar to the internal structure of a single FPGA. Thus, one has a hierarchy of configurability.

21.2.4 Configurable Computing System Architectures

The placement of one or more configurable components within a computing system is largely determined by the requirements of the application. Several architectures are shown in Fig. 21.9. In some cases as shown in Fig. 21.9(a), no additional computing power is required and the component can be utilized in a stand-alone mode. This situation occurs in Internet routing nodes and in some data acquisition systems as well as controllers for actuators. Note that the use of FPGAs to replace logic or to be used as state machines is a typical application of this type of architecture. This type of application was the first widespread use of FPGAs.

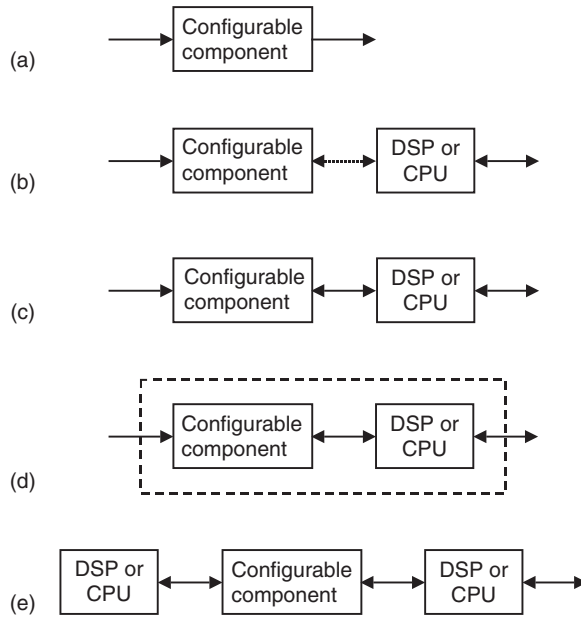


FIGURE 21.9 CPU/configurable computing architectures.

For configurable computing systems, configurable components are more commonly coupled with conventional DSPs or CPUs such that the processor can accomplish general-purpose computing while acceleration of specialized functions can be performed by the configurable components. The type of general-purpose computing required by the application determines the choice of a DSP or CPU. An application involving signal processing would naturally lead to the use of a DSP. Whereas, an application involving user interaction or user services (disks, etc.) would more likely lead to the use of a general purpose CPU. For this discussion on general configurable computing system architectures, the type of general-purpose processor used is irrelevant; however, it is very relevant when an actual application and system are being developed.

Figures 21.9(b–e) depict architectures that have configurable components coupled with DSPs or CPUs. The communication requirements between the different types of processors determine the amount of bandwidth and latency provided. If infrequent communication is needed, a serial line or some other slow-speed connection may be sufficient as shown in Fig. 21.9(b). For higher bandwidth applications, placing the two types of components on a bus or some other high-speed connection as shown in Fig. 21.9(c) may be appropriate. In both of these cases, tasks best suited for a particular component can be delegated to that component and sharing of data and results are facilitated. Figure 21.9(d) depicts the tightest coupling with the lowest latency and highest bandwidth since both types of components are placed inside the same package. Often, the DSP or CPU manages the data, especially when disk storage is involved. When the data is being acquired at a high rate from a sensor, the configurable component is often used to perform initial operations to reduce the size of the data. Thus, the DSP or CPU has only a fraction of the data to be processed or stored. Note that the current trend to include RAM and a fixed microprocessor core on the same integrated circuit as the FPGA is an implementation of this architecture. Another variation of this theme of placing the configurable component within the system just where it is needed can be seen in a network of workstations as shown in Fig. 21.9(e). In this case, the configurable component can be inserted into the router itself to perform dedicated operations as the data is passed from processor node to processor node. The processing performed in this manner appears to be “free” as it occurs during message passing.

21.2.5 Selected Applications

Several classes of applications have improved performance when implemented on configurable computing systems including image analysis, video compression, and molecular biology. In general, these applications exploit the parallel processing power of configurable computers. Current applications that exert high demand for reconfigurable systems include communications and mobile systems, which utilize FPGAs to provide more flexible operations than ASICs yet higher speed and lower power than CPUs. Applications that can benefit from variable-grain parallelism of FPGAs are hot prospects to emerge as high-volume applications in the near future, especially as improvements in data movement are made.

21.2.5.1 Image Analysis

Image analysis requires manipulating massive amounts of data in parallel and performing a variety of data interaction (e.g., point operations, neighborhood operations, and global operations). Many of these operations are ideally suited for implementation on a configurable computing system due to the parallel nature of the operations. Example implementations are image segmentation, convolution, automated target recognition (ATR), and stereo vision. Image segmentation is often the first step in image analysis and consists of extracting important image features. For intensity images (i.e., those represented by point-wise intensity levels), the four popular approaches are threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods. A systolic array of configurable components, similar to that depicted in Fig. 21.8(b), can be used to perform an edge-based image segmentation. Implementation results for various applications have shown that this approach is superior to the conventional digital signal processor approach.

Two-dimensional convolution is commonly used for filtering, edge detection, and feature extraction. The basic idea is that a window of some finite size and shape is scanned across the image. The output pixel value is the weighted sum of the input pixels within the window where the weights are the values of the filter assigned to every pixel of the window itself. Using a systolic array of configurable components, the convolution window can be applied in parallel with an output pixel value being produced as new pixel values are provided. The storage of intermediate pixel values within the window is inherent in the systolic array structure. Implementation results have shown impressive performance gains.

ATR is a computationally demanding application in real-time image analysis problems. The objective of an ATR system is to analyze a digitally represented input scene to locate or identify all objects of interest automatically. Typically, algorithms begin with a preprocessing step to identify regions of interest in the input image. Next, template matching is performed to correlate these regions of interest with a very large number of target templates. The final identification step identifies the template and the relative offset at which peak correlation occurs. The template matching process is the most computationally intensive among these three steps and has the potential of being implemented in a parallel form. Therefore, the template matching is a good candidate to be mapped into a configurable computing system. Implementation results have shown significant performance improvements.

Stereo vision involves locating the same features in each of two images and then measuring the distances to objects containing these features by triangularization. Finding corresponding points or other kinds of features in two images, such that the matched points are the same projections of a point in the scene, is the fundamental computational task. Matching objects at each pixel in the image leads to a distance map. This is very similar to the template matching process in an ATR system and implementation results are similar.

21.2.5.2 Image and Video Compression

Image and video compression are used in many current and emerging products. Image compression is widely used in desktop publishing, graphic arts, color facsimile, and medical imaging. Video compression is at the heart of digital television set-top boxes, DSS, HDTV decoders, DVD players, video conferencing, Internet video, and other applications. Compression reduces the requirements for storage of large archived pictures, less bandwidth for the transmission of the picture from one point to another,

or a combination of both. Image and video processing typically require high data throughput and computational complexity. JPEG is widely used for compressing still pictures, and MPEG or wavelets are more appropriate for compressing videos or general moving pictures.

A configurable component using a pipelined approach, as depicted in Fig. 21.8(a), provides a much cheaper and more flexible hardware platform than special image compression ASICs, and it can efficiently accelerate desktop computers. A speed improvement over a modern workstation of a factor of 10 or more can be obtained for JPEG image compression.

21.2.5.3 Molecular Biology

Scanning a DNA database is a fundamental task in molecular biology. This operation consists of identifying those sequences in the DNA database that contain at least one segment sufficiently similar to some segment of a query sequence. The computational complexity of this operation is proportional to the product of the length of the query sequence and the total number of nucleic acids in the database. In general, segment pairs (one from a database sequence and one from query sequence) may be considered similar if many nucleotides within the segment match identically. This similarity search may take several hours on standard workstations when using common software that is parameterized for high sensitivity.

One method of performing DNA database searches is to use a dynamic programming algorithm for computing the edit distance between two genetic sequences. This algorithm can be implemented on a configurable computing system configured as two systolic arrays. Execution has been found to be several orders of magnitude faster than implementations of the same algorithm on a conventional computer. Another method is to use a systolic filter for speeding up the scan of DNA databases. The filter can be implemented on a configurable computing system, which acts as a coprocessor that performs the more intensive computations occurring during the process. An implementation of this system boosted the performances of the conventional workstation by a factor ranging from 50 to 400.

21.2.6 Virtual Computing Power

Quantifying computing power is a challenging task due to differing computing architectures and applications. Vuillemin et al. [13] define virtual computing power based on the number of programmable active bits (PABs) and the operating frequency. He defines a “reference” PAB as a 4-input Boolean function. These functions are essentially the core configurable elements of a configurable computing component; however, each vendor defines them differently. For example, Xilinx calls them logic cells (LCs) and organizes them into groups of four called configurable logic blocks (CLBs). Whereas, Altera calls them logic elements (LEs) and organizes them into groups of 10 called logic array blocks (LABs). As newer and larger FPGAs with new architectures are constructed, the vendors will likely rename these logic blocks. But the configurable blocks can always be defined as Boolean functions.

21.2.7 Development Systems

Developing applications for microprocessor-based systems is currently far easier than developing applications for a configurable computing system. Microprocessor development systems have been optimized over years, even decades, while those for configurable computing systems are in their infancy. A design for a single FPGA is typically created using tools similar to those used for other digital systems, tools such as schematic capture, VHDL, etc. Owing to the proprietary nature of FPGAs, however, a designer must typically use the design tools available from the FPGA vendor.

A software design environment that facilitates the rapid development of applications on configurable computing systems should permit high-level design entry, simulation, and verification by application designers who need not be familiar with the details of the hardware. Of course, on occasions, it may be necessary to expose to the application designer those hardware details deemed essential to ensure feasible and efficient implementations. Metrics and visualization are desirable to assist the application designer in achieving near-optimal system implementation rapidly. The tools available

from the FPGA vendors are currently intended for digital systems designers and not the application designer. Research efforts underway at various universities and startup companies are producing the first development systems for configurable computing systems similar to those for microprocessor systems.

To Probe Further

More in-depth information on configurable computing systems is readily available. The first sources of information are the FPGA vendors. A few of these are

1. Altera Corporation, San Jose, CA. <http://www.altera.com>.
2. Atmel Corporation, San Jose, CA. <http://www.atmel.com>.
3. Xilinx, Inc., San Jose, CA. <http://www.xilinx.com>.

Several sites on the World Wide Web are dedicated to configurable computing systems. A search using the terms configurable computing systems or reconfigurable computing systems via any of the search engines will yield a great number of hits. One of these sites is <http://www.optimagic.com> that provides information on not only configurable computing systems but also programmable logic in general.

Currently, few books focus specifically on configurable computing systems; however, many books about programmable logic provide excellent references for someone interested in configurable computing. Some of these are the following:

4. J. Hamblen, T. Hall and M. Furman, *Rapid Prototyping of Digital Systems*, ISBN 0-387-27728-5, AQ2 Springer, 2006.
5. M. Gokhale and P. Graham, *Reconfigurable Computing*, ISBN 0-387-26105-2, Springer, 2005.
6. C. Maxfield, *The Design Warrior's Guide to FPGAs*, ISBN 0-750-67604-3, Elsevier, 2004.
7. W. Wolf, *FPGA-Based System Design*, ISBN 0-131-42461-0, Prentice-Hall, Englewood Cliffs, NJ, 2004.

Many excellent conferences are held annually to provide the latest information on configurable computing systems from the FPGAs to development systems. Some of these conferences are as follows:

8. Symposium on Field-Programmable Custom Computing Machines (FCCM), <http://www.fccm.org>.
9. ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, (FPGA). <http://www.lsfpga.org>.
10. International Workshop on Field Programmable Logic and Applications (FPL), <http://www.fpl.org>.
11. Reconfigurable Architectures Workshop (RAW), <http://www.ece.lsu.edu/vaidy/raw06/>.
12. Design Automation Conference (DAC), <http://www.dac.com>.

The proceedings from these conferences contain many articles on not only configurable computing systems but also applications for which configurable computing systems have been shown to be effective. The configurable computing systems are applied in other areas such as cryptography, fingerprint matching, multimedia, and astronomy. AQ3

More in-depth information on virtual computing power and a list of applications of configurable computing system is as follows:

13. J. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. Touati, and P. Boucard, Programmable active memories: Reconfigurable systems come of age, *IEEE Trans. VLSI Syst.*, 4(1): 56–69, 1996.

More information on research and development into design tools for configurable computing may be obtained by visiting the Web sites of the research groups involved. Some of these are

14. Brigham Young University, Configurable Computing Web page, <http://splish.ee.byu.edu> and the JHDL Web page, <http://www.jhdl.org>
15. University of Cincinnati, REACT Web page, <http://www.ececs.uc.edu/~dal/acs/index.htm>.
16. Colorado State University, CAMERON Project Web page, <http://cs.colostate.edu/cameron>.
17. Northwestern University, A Matlab Compilation Environment for Adaptive Computing Systems Web page, <http://www.ece.nwu.edu/cpdc/Match/Match.html>.