

Fine-grain real-time reconfigurable pipelining

S. Kim
C. H. Ziesler
M. C. Papaefthymiou

In many computations, average data rates are often significantly lower than the peak rate possible. Consequently, VLSI systems capable of processing data at a maximum specified rate can be excessively dissipative when data rates are low. Such inefficiencies are particularly pronounced in heavily pipelined designs, in which registers account for the bulk of energy dissipation in a system. This paper describes a novel methodology for designing reconfigurable pipelines that achieve very low power dissipation by adapting their resources to their computational requirements. In our fine-grain reconfigurable pipelines, energy is saved by disabling and bypassing an appropriate number of pipeline stages whenever data rates are low. In contrast, coarse-grain approaches, such as dynamic voltage scaling, are often unable to capture savings from short-time-scale variations in throughput requirements because of the long time needed to reconfigure the voltage. To evaluate our methodology, we designed an inverse discrete cosine transform (IDCT) module for MPEG-2. Our IDCT included pipelined multipliers that were dynamically reconfigurable on the basis of the number of nonzero coefficients per block and picture size. In comparison with conventional multipliers in corresponding IDCT implementations, our reconfigurable multipliers dissipated about 12–65% less power.

1. Introduction

Pipelining enables the realization of high-speed, high-efficiency CMOS datapaths by allowing the reduction of supply voltages at the lowest possible levels while still satisfying throughput constraints. In deep pipelines, however, registers and corresponding clock trees are responsible for an increasingly large fraction of total dissipation, no matter how efficiently they may have been implemented [1–4]. For example, the power consumed by the registers of a predictive vector quantization (PVQ) decoder described in [2] amounts to 90% of the total datapath dissipation. In general, these registers latch their inputs unconditionally, even if input data does not change, and thus consume significant power no matter how efficiently they may have been implemented [1, 3, 4].

This paper presents a methodology for designing fine-grain reconfigurable pipelined datapaths that can adapt

their performance and dissipation to required data rates in real time. These datapaths can efficiently cope with the variability of data rate that is commonplace in numerous applications. Our reconfiguration methodology reduces energy dissipation by disabling and bypassing a select subset of registers. The number of register stages and corresponding clock trees to be disabled at any interval in the operation of the pipeline is periodically determined by the amount of computation that must be performed at the time. Reconfiguration can be performed “on the fly” while data is streaming through the datapath. The control hardware overhead associated with our approach is very low. For an n -stage pipeline, additional hardware is limited to $O(n \log n)$ state bits and $O(n)$ multiplexers.

An application domain that naturally lends itself to our real-time fine-grain reconfiguration scheme is video processing, a key component of multimedia communications

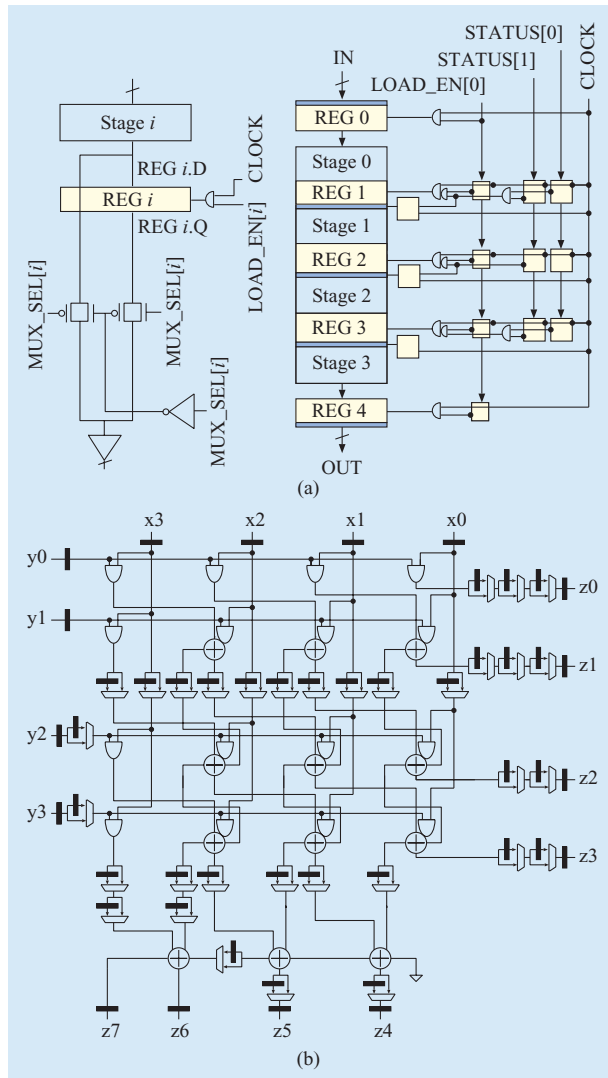


Figure 1

(a) Four-stage reconfigurable pipeline; (b) 4×4 array multiplier with reconfigurable four-stage pipeline.

and a potentially integral part of next-generation portable devices. Currently, there are several video standards established for different purposes, including MPEG-1, MPEG-2, and H.261, and their implementations for mobile systems-on-a-chip (SoCs) should provide substantial computing capabilities at low energy consumption levels [5]. The building blocks of these standards include demanding computations such as the discrete cosine transform (DCT), inverse discrete cosine transform (IDCT), motion estimation, motion compensation, variable-length coding/decoding, quantization, and inverse quantization. Video streams are particularly suitable for low-power processing using our

reconfiguration approach, because the required data rates of downstream components can be inferred by observing the output values of upstream components. Owing to the real-time reconfiguration capability of our scheme, variable data rates can be accommodated without interrupting the flow of data through the pipeline. In contrast, alternative dynamic adaptation schemes such as voltage scaling would require several cycles to reconfigure the system and thus result in unacceptable latencies for real-time latency-sensitive applications.

To evaluate the efficiency of our methodology, we applied it to the design of reconfigurable pipelined multipliers that were used in IDCT modules with varying degrees of parallelism. Our multipliers were dynamically reconfigured according to the number of nonzero DCT coefficients per block and the picture size. We compared the energy efficiency of the reconfigurable IDCTs with that of statically pipelined IDCTs with identical architecture and peak performance capability. In simulations with a $0.35\text{-}\mu\text{m}$ CMOS technology, our reconfigurable pipelined multipliers used to perform two-dimensional IDCT achieved relative reductions up to 65% compared with the nonreconfigurable counterparts.

The remainder of this paper contains four sections. Section 2 introduces our reconfigurable pipelining scheme. Section 3 describes the application of our approach to the design of low-power two-dimensional IDCT modules for MPEG video processing. Section 4 presents simulation results from the comparative evaluation of reconfigurable and conventional IDCT designs on MPEG video streams. Our contributions are summarized in Section 5.

2. Reconfiguration methodology

In this section, we highlight our reconfigurable pipeline design methodology. Specifically, we describe the structure and operation of our reconfigurable pipelines and discuss the control hardware overhead associated with our approach. To provide a concrete example, we present a four-stage 4×4 array multiplier that is pipelined according to our proposed scheme.

Figure 1(a) illustrates our proposed design methodology using a four-stage reconfigurable pipeline. The figure includes the control logic of the pipeline and details of the i th pipeline stage. This four-stage pipeline is capable of handling a maximum of four samples per four cycles. Whenever throughput requirements are low, however, register stages can be selectively disabled by gated clocks and bypassed by multiplexers. The original four-stage pipeline can thus be reconfigured to compute with a latency of two stages, one stage, or no latency at all (purely combinational). **Figure 1(b)** shows the instantiation of a 4×4 array multiplier as a four-stage reconfigurable pipeline.

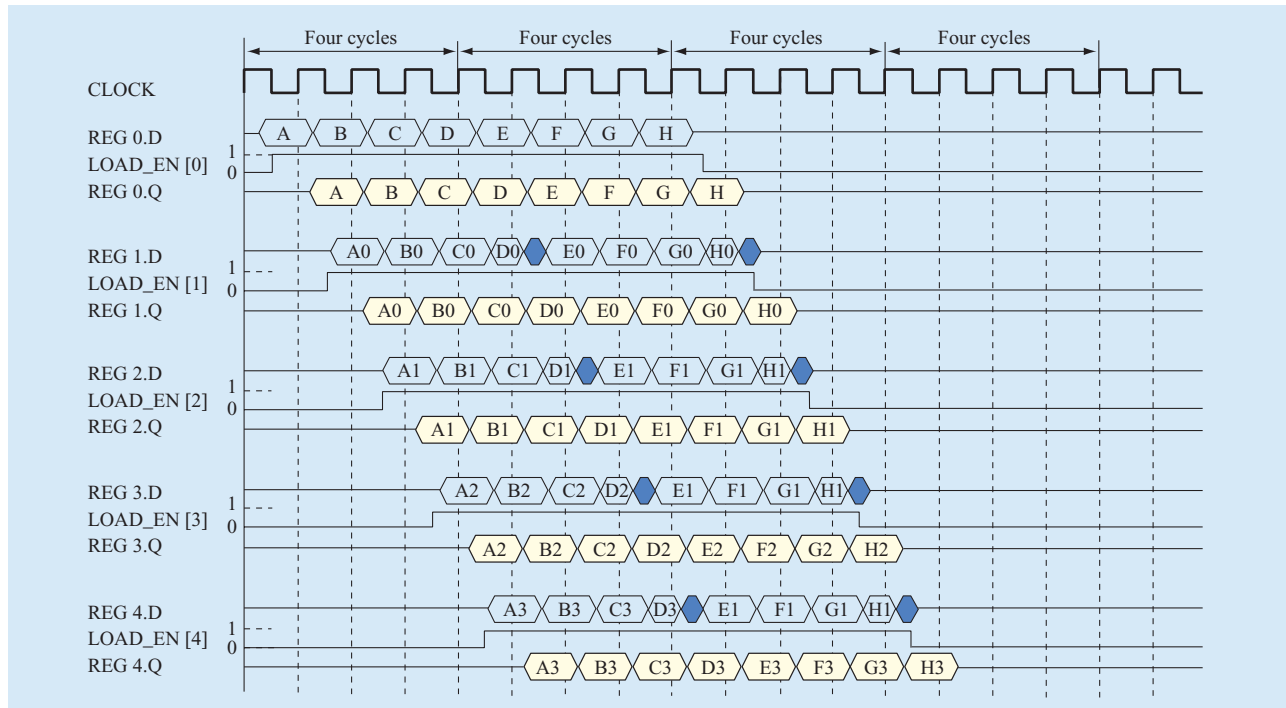


Figure 2

Timing diagram of conventional four-stage pipeline for two sets of samples at the rate of four samples per four cycles.

The *pipeline mode*, that is, the number of pipeline stages that are used to process an input, is determined by a pair of *status bits*. For each data item that is to be processed in a given mode, the corresponding status bits are injected into the control pipeline. The propagation of the status bits is itself pipelined, “shadowing” the flow of the data through the datapath. At each stage, the status bits are combined with a load-enable signal and a clock signal to generate a gated clock and a multiplexer-select signal for the pipeline register and the multiplexer of that stage, respectively.

To ensure that data is not corrupted in the pipeline, reconfiguration must be coordinated with the rate at which new inputs enter the pipeline. In particular, if the pipeline is in its base four-stage mode, new data may enter every cycle. If the pipeline is in its two-stage mode, however, new inputs are accepted only every other cycle, since every pipeline stage effectively requires two “base” cycles to complete its computation.

Figure 2 shows the timing diagram of a conventional (nonreconfigurable) four-stage pipeline that processes two sets of samples, each at the maximum rate of four samples per four cycles. Our reconfigurable four-stage pipeline can be configured to have the same pipeline structure as the conventional one, thus achieving the same maximum rate of four samples per four cycles.

When only one sample has to be processed every four cycles, as shown in **Figure 3(a)**, all registers in each stage of the conventional four-stage pipeline are used for one cycle and remain idle for three cycles. In this case, our reconfigurable four-stage pipeline can operate in its single-stage mode, as shown in the timing diagram of **Figure 3(b)**, processing the two sets of samples at the rate of one sample per four cycles.

Figures 4(a) and **4(b)** show the timing diagrams of a conventional four-stage pipeline and a reconfigurable four-stage pipeline in its four-, one-, two-, and four-stage modes, respectively. These pipelines process four sets of input samples in total, each at the rate of four, one, two, and four samples per four cycles. Reconfiguration is performed with no need for any pipeline bubbles. When input rates are low, the reconfigurable pipeline uses idle cycles to spread the computation and eliminate three stages of registers, thus reducing register dissipation without sacrificing throughput.

In general, to ensure that no data is lost in an n -stage reconfigurable pipeline, when the input sampling rate is in the range $(2^{i-1}/n, 2^i/n]$ samples per cycle, the number of registers bypassed by multiplexing, S , should be $n/2^i - 1$. If the input rate falls below $1/n$ samples per cycle, S stays at $n - 1$.

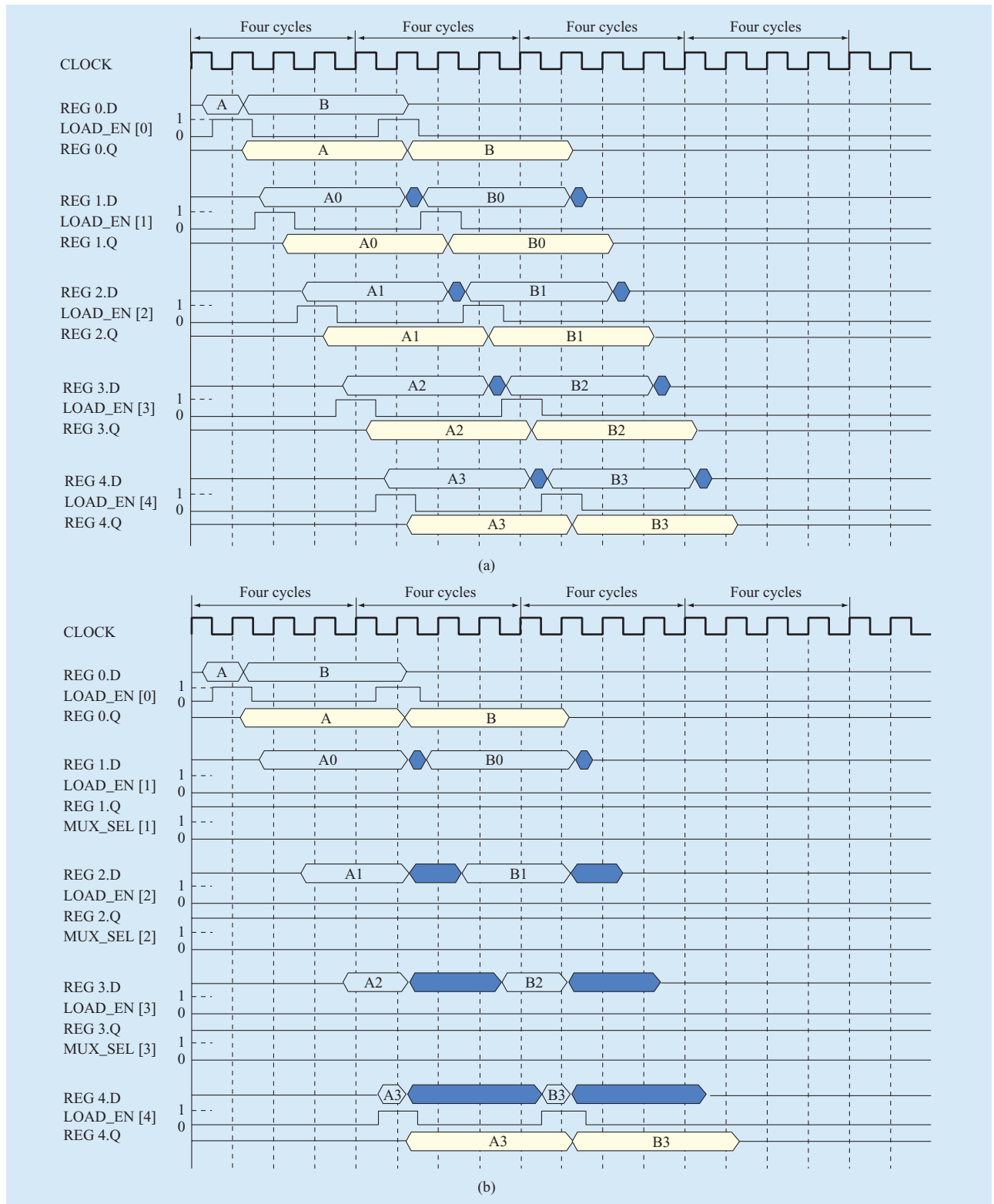


Figure 3

Timing diagram of (a) four-stage conventional pipeline and (b) four-stage reconfigurable pipeline for two sets of samples, each at the rate of one sample per four cycles.

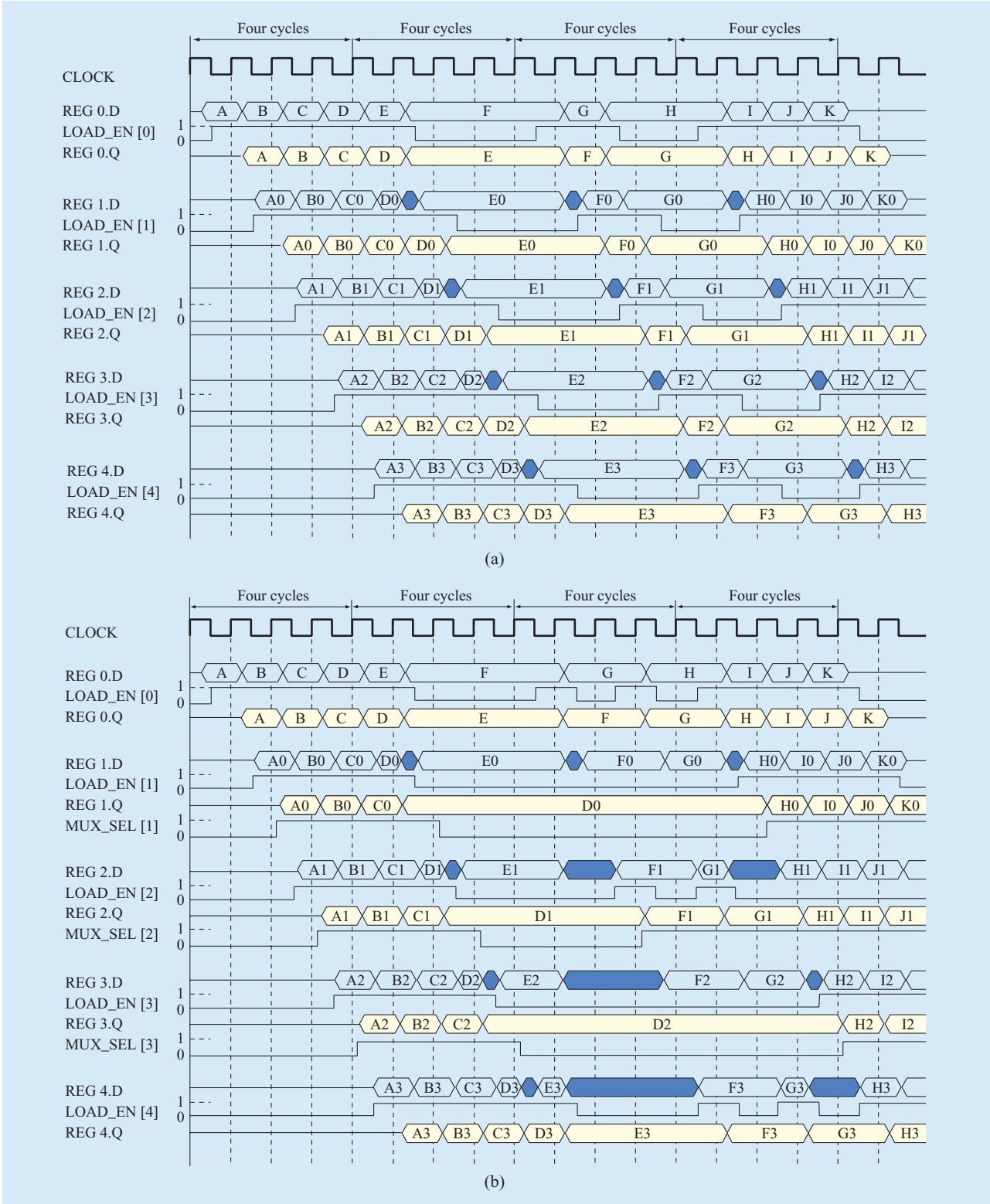


Figure 4

Timing diagram of (a) four-stage conventional pipeline and (b) four-stage reconfigurable pipeline for four sets of samples at the rate of four, one, two, and four samples per four cycles, respectively.

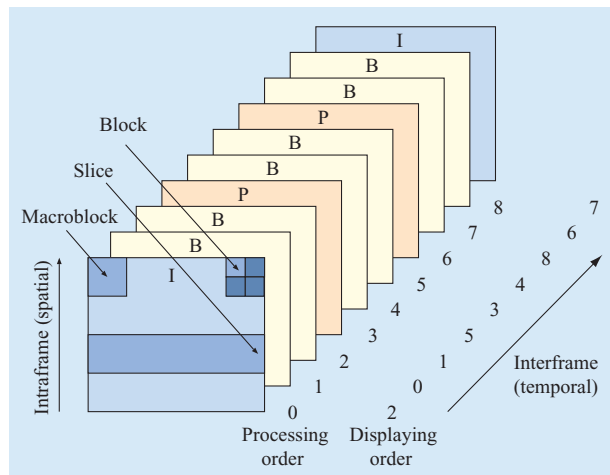


Figure 5

Illustration of I-, P-, B-frames and data hierarchies.

The control hardware overhead of our reconfiguration scheme is relatively low, amounting to $O(\log n)$ additional logic and state bits per stage in n -stage pipelines. Specifically, for each original pipeline stage, the number of status bits and the number of NAND gates introduced for clock gating are proportional to $\log n$. Moreover, a single load-enable bit is required per stage. Therefore, the overall hardware overhead for controlling an n -stage reconfigurable pipeline is $O(n \log n)$.

The basic reconfiguration methodology described in this section can be enhanced in several ways to improve clock period and reduce power dissipation. For example, if we introduce a single-cycle pipeline bubble at each reconfiguration, we can effectively hide the delay of the multiplexer. Additional power savings can be obtained by disabling the local clock trees.

3. Application to MPEG video processing

In this section we give a brief overview of the MPEG video processing standard. We then describe the application of our reconfigurable pipelining scheme to the design of an inverse discrete cosine transform (IDCT) module for the MPEG processing pipeline. This module uses reconfigurable pipelined multipliers that operate at substantially reduced dissipation levels in comparison with their conventional counterparts.

In general, two coding types are used for video compression: intraframe and interframe. Intraframe coding exploits the spatial redundancy within a frame, while interframe coding exploits temporal redundancy between frames. **Figure 5** depicts both the intraframe and interframe coding schemes in the MPEG video standard. It also shows the data hierarchies of video sequence,

group of pictures (GOP), slice, macroblock (MB), and block. In the spatial domain, each video frame is divided into blocks with 8×8 pixels. Four luminance blocks and two chrominance blocks are grouped to create a MB with 16×16 pixels. In the temporal domain, there are three types of frames: intra (I), predictive (P), and bidirectional (B) frames.

The encoding of I-frames is based on spatial redundancy. The purpose of P- and B-frames is to reduce temporal redundancy by motion compensation, which is accomplished by identifying, for every MB in the current frame, the best-matching MB from the previous or the next I- or P-frame. Both forward-prediction and backward-motion vectors (MVs) can be used for motion compensation during decoding. The motion-compensated prediction error (that is, the difference between the motion-compensated MB and the current MB) is transformed into an array of 8×8 transform coefficients using the two-dimensional (2D) DCT. The coefficients are quantized and subsequently encoded using run-length techniques. During decoding, the encoded video bitstream is analyzed into the motion vectors of each MB and the DCT coefficients of each block. The motion vectors are used to generate a motion-compensated prediction that is added to a decoded prediction-error signal to generate the reconstructed video.

MPEG video standards specify the upper bounds for picture size, frame rate, and bit rate for various combinations of profiles and levels. For the main-profile at main-level (MP@ML) combination of MPEG-2 in the NTSC-compatible mode, for example, picture size, frame rate, and bit rate are 720×480 pixels, 30 frames/s, and 15 Mb/s, respectively. Thus, the maximum throughput requirement for an MPEG-2 MP@ML decoder is 15.552×10^6 samples/s [$30 \text{ frames/s} \times (720/16 \times 480/16) \text{ MB/frame} \times 6 \text{ blocks/MB} \times 64 \text{ pixels/block}$].

In the MPEG video standard, reconstructed images should be sent to display devices on time. This requirement imposes a real-time constraint on the decompression computation. Therefore, each functional block should be designed to meet the maximum throughput dictated by the standard, even though average data rates can be substantially lower than the upper bound specified by the standard. However, blocks are not required to process data as quickly as possible, as long as they process the data within a specified time. Power savings can thus be achieved by spreading the computation across the time allotted, using the minimum level or resources required to meet throughput requirements.

Furthermore, the time scale over which computations can be spread is limited by practical considerations of the data rates and the cost of buffering. Buffering the decompressed output stream could allow the real-time

constraint to be averaged over longer time scales. However, since the output bandwidth is high, the overhead in silicon area is high, and the energy of buffering substantial amounts of the output stream becomes prohibitive. Thus, power savings in this case can be achieved only by fine-grain approaches such as our pipeline reconfiguration methodology. Other approaches such as dynamic voltage scaling cannot cope with the short time scales over which voltage scaling must occur.

In general, average data rates in video processing are often significantly lower than the maximum possible rate. Therefore, although the maximum throughput requirement of the IDCT module is determined by the worst-case number of nonzero DCT coefficients per block, the average number of nonzero DCT coefficients per frame is much smaller than the worst-case one.

To evaluate the performance of our reconfigurable pipelining scheme, we applied it to the design of a multiply-accumulate (MAC)-based 2D IDCT, one of the most computing-intensive tasks in MPEG video processing. The 2D IDCT of an 8×8 -coefficient block can be separated into two sequential eight-point one-dimensional (1D) IDCTs using the row-column decomposition technique. This decomposition scheme is preferred for VLSI implementations of 2D IDCT because of its regularity and numerical characteristics. The eight-point 1D IDCT of a DCT coefficient vector Z is given by the expression

$$x(n) = \sum_{k=0}^7 \frac{c(k)}{2} z(k) \cos \left[\frac{(2n+1)k\pi}{16} \right],$$

where $c(0) = 1/\sqrt{2}$, and $c(k) = 1$ if $k \neq 0$.

In a MAC-based IDCT architecture, the number of nonzero DCT coefficients per block equals the number of multiply-and-accumulate operations required to compute the IDCT of the block. A MAC-based IDCT architecture with eight multipliers processes eight samples per eight cycles. For the maximum throughput of 15.552×10^6 samples/s specified in the MPEG-2 MP@ML video standard, the operating frequency should be at least 15.552 MHz. Similarly, a MAC-based IDCT architecture with four multipliers can be derived from a MAC-based IDCT architecture with eight multipliers using a multiplexer to eliminate the second 1D IDCT. To satisfy the maximum throughput requirement, the remaining 1D IDCT would have to process samples at twice the input sample rate, that is, 31.104 MHz. A MAC-based IDCT with two multipliers will have to operate at speeds higher than 62.208 MHz. To achieve these required clock rates at the lowest voltage possible, multipliers in the three IDCT structures should be deeply pipelined. Consequently, pipeline registers consume an increasingly larger fraction

of the total dissipated energy, even though overall energy consumption may be reduced.

The main idea behind our energy-efficient reconfigurable IDCT is to predict the number of operations required for each 8×8 -coefficient block and use it to reconfigure the IDCT multiplier pipelines to match the throughput requirement. This number can be obtained effortlessly before the 2D IDCT begins to process the block. Specifically, for the first 1D IDCT, the number of nonzero DCT coefficients can be simply counted at the output of the variable-length decoder or inverse quantizer, which are both upstream from the IDCT in the MPEG pipeline. The number of nonzero DCT coefficients per block for the second 1D IDCT can be counted at the output of the first 1D IDCT. These counts can be used to reconfigure the pipelined multipliers of the MAC-based IDCT module. Furthermore, the time scale at which this reconfiguration occurs (that of one 8×8 block) is small enough that no extra buffering is needed on the output stream.

As the number of nonzero DCT coefficients per block decreases, the number of disabled register stages increases. At low data rates, the reconfigurable pipelined multipliers operate as fully combinational circuits, and the energy consumption of their registers is eliminated. Given a block, the IDCT is guaranteed to stay in the same reconfiguration mode for a number of cycles inversely proportional to the number of multipliers it uses (64, 128, and 256 cycles for eight, four, and two multipliers, respectively).

In addition to the number of nonzero coefficients, picture size can be used to achieve further energy savings. For example, if picture size is one quarter of the maximum for MPEG-2 MP@ML, the number of blocks that must be processed within 1/30 second decreases by a factor of 4. Therefore, the number of required pipeline stages can decrease by a factor of 4, independently of the number of nonzero DCT coefficients per block.

4. Simulation results

In this section, we present results from the comparative evaluation of our reconfigurable IDCT modules with conventional pipelined IDCTs when decoding a variety of MPEG-2 MP@ML test bitstreams. We first focus on one test bitstream at a fixed bit rate and provide detailed evidence about the effectiveness of our coefficient-based reconfiguration scheme. We then give simulation results that demonstrate the significant relative savings that can be achieved by using our reconfigurable design methodology instead of nonreconfigurable pipelines for a variety of bitstreams and bit rates.

We synthesized three different topologies of MAC-based IDCTs in a 0.35- μm CMOS technology using eight, four, and two multipliers. To achieve the maximum



Figure 6

A reconstructed I-picture of the MPEG bitstream *flower garden*.

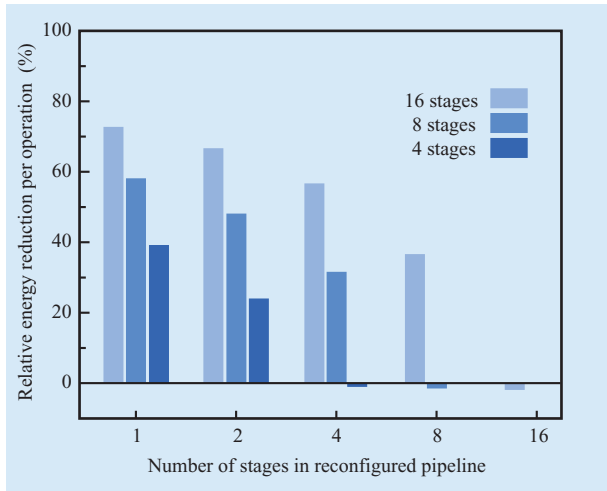


Figure 7

Relative energy reduction per operation for reconfigurable multipliers. Reprinted with permission from [8]; © 2000 IEEE.

throughput requirement at 1.4 V, the 16×16 multipliers in the three IDCT structures used four, eight, and 16 stages, respectively. Consequently, for deeper pipelines, pipelining registers consumed an increasingly larger fraction of total dissipated energy. The registers were implemented by positive edge-triggered D flip-flops using transmission gates, a common practice in standard-cell design [6].

Our initial experiments focused on the benchmark bitstream *flower garden*, which is provided by the MPEG video committee and consists of 38 I-pictures, 113 P-pictures, and 299 B-pictures with a resolution of

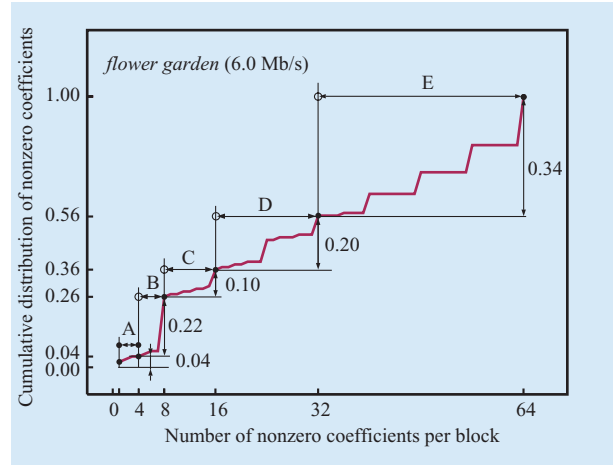


Figure 8

Cumulative distribution of nonzero coefficients per block on *flower garden* at 6.0 Mb/s. Reprinted with permission from [8]; © 2000 IEEE.

704×480 pixels. **Figure 6** shows one of the I-pictures reconstructed from the *flower garden* bitstream.

To estimate power consumption, we relied on a switch-level circuit simulator with RC parameters extracted using a commercial Verilog-HDL synthesizer and standard-cell router. Power estimates were obtained for the multipliers in the IDCT, since they account for a large portion of the total IDCT dissipation. The dissipation of buffers for transposition and output was ignored, since these components were clocked only once every eight cycles and are known to dissipate only a very small fraction of the total energy in MAC-based IDCTs [7].

Figure 7 shows the relative energy savings per operation for three reconfigurable multipliers with worst-case latency of 4, 8, and 16, when the input data rate permits their reconfiguration with fewer pipeline stages. For example, the solid black bar corresponding to one stage in the reconfigured pipeline indicates that when the 16-stage reconfigurable multiplier is configured as a single-stage combinational path, it saves more than 70% of the dissipation of the conventional 16-stage multiplier. Similarly, the fifth bar from the left denotes that when the eight-stage multiplier is configured as a two-stage pipeline, it saves about 50% of the energy dissipated by the conventional eight-stage multiplier. The negative savings for stages 4, 8, and 16 are due to the dissipation of the reconfiguring hardware.

Figure 8 gives the cumulative distribution of the number of nonzero DCT coefficients per block for the 2D IDCT of the bitstream at 6.0 Mb/s. According to this graph, the potential of our reconfiguration scheme to reduce the

power dissipation of the IDCT multipliers is substantial. In particular, 36% of the nonzero DCT coefficients in the bitstream are found in blocks with at most 16 nonzero coefficients. To process each of these blocks on the IDCT that uses eight multipliers, it is sufficient to configure the multipliers as single-stage pipelines, thus eliminating all of the intermediate registers in their four-stage reconfigurable pipelines. The aggregate throughput of the single-stage multipliers decreases to one sample per four cycles. Since the number of nonzero DCT coefficients in each block is no more than one quarter of the maximum number of nonzero DCT coefficients per block, the 2D IDCT of each such block can be completed within 64 cycles.

Figures 7 and 8 can be combined to estimate the relative energy savings of a MAC-based reconfigurable IDCT architecture on the bitstream *flower garden* over a conventional IDCT with the same worst-case latency and maximum throughput. For example, consider the third bar from the left in Figure 7. The relative energy savings when a reconfigurable four-stage multiplier is configured as a single-stage pipeline are 40% over the conventional four-stage implementation. From Figure 8 it follows that approximately 36% of the blocks in *flower garden* can be processed by a single-stage pipeline. Therefore, the relative savings in energy dissipation over a corresponding nonreconfigurable four-stage pipeline are approximately $0.36 \times 40\% = 14.4\%$, as shown in Figure 9. Similarly, 20% of the nonzero coefficients in *flower garden* can be found in blocks with 17–32 nonzero coefficients that can be processed by a two-stage configuration of the four-stage multiplier. In this case, the reconfigurable multiplier saves about 24% over the conventional four-stage multiplier, and the relative energy savings of reconfiguration for these operations are approximately $0.20 \times 24\% = 4.8\%$. Finally, about 34% of the nonzero coefficients are elements of blocks with more than 32 nonzero coefficients. In this case, the reconfigured four-stage multipliers should be used for eight samples per eight cycles. The relative energy savings over the nonreconfigurable multiplier are about -1.7% . Therefore, during these operations, about $0.34 \times (-1.7)\% = -0.6\%$ of total energy is saved. The total relative energy savings with the reconfigurable four-stage multipliers over the nonreconfigurable ones are $14.4 + 4.8 - 0.6 = 18.6\%$ when the 2D IDCT for *flower garden* at 6.0 Mb/s is performed. In the case of four- and 16-stage reconfigurable pipelined multipliers, the relative savings are 25.6% and 29.7% over their conventional counterparts.

Figure 10 gives statistics for the number of nonzero DCT coefficients per block for different bit rates of the image source *flower garden*. The percentage of nonzero DCT coefficients in blocks with a small number of nonzero DCT coefficients increases. Therefore, for lower

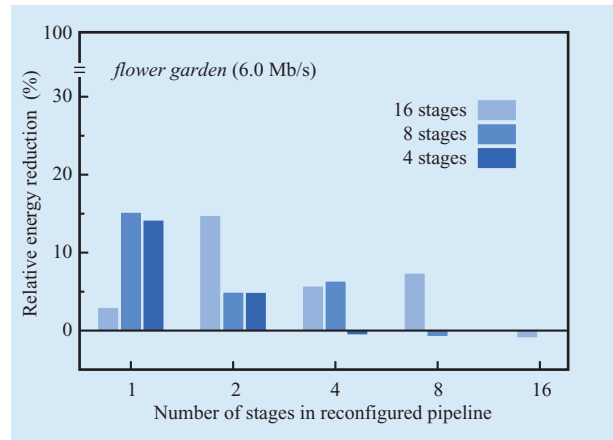


Figure 9

Relative energy reduction in each reconfiguration mode of reconfigurable multipliers on *flower garden* at 6.0 Mb/s. Reprinted with permission from [8]; © 2000 IEEE.

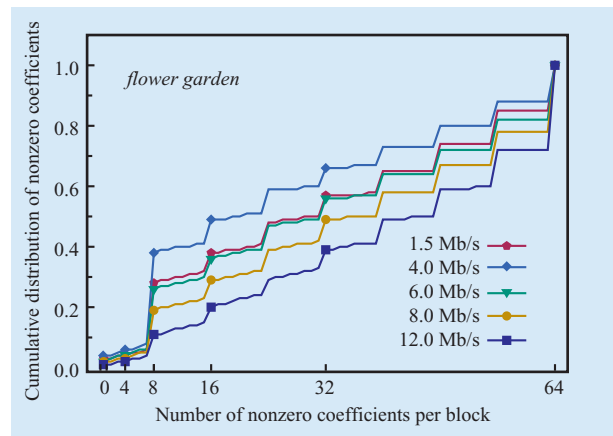


Figure 10

Cumulative distribution of nonzero coefficients per block on *flower garden* at various average bit rates. Reprinted with permission from [8]; © 2000 IEEE.

bit rates, the reconfigurable multipliers will spend a longer time in a “shallow,” energy-efficient mode, and thus greater relative savings can be achieved. The picture size of *flower garden* at 1.5 Mb/s is 352×240 pixels, which is a quarter of the maximum picture size, 720×480 pixels, in MPEG-2 MP@ML.

Figure 11 shows the average relative energy savings that can be achieved using the three different topologies of the reconfigurable MAC-based IDCTs instead of their conventional counterparts for the MPEG-2 MP@ML bitstreams *susi*, *table tennis*, *mobile*, and *flower garden*,

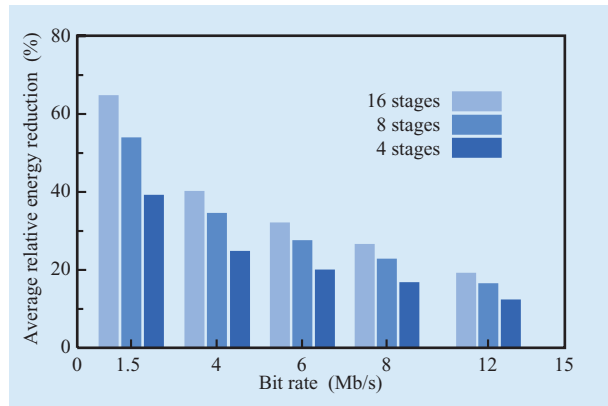


Figure 11

Average relative energy reduction of reconfigurable multipliers on *susi*, *table tennis*, *mobile*, and *flower garden* at various average bit rates. Reprinted with permission from [8]; © 2000 IEEE.

when pipeline stages are reconfigured using the number of nonzero coefficients per block and the picture size of each bitstream. For each bitstream, results are given for five different bit rates: 1.5, 4.0, 6.0, 8.0, and 12.0 Mb/s. The picture sizes of these bitstreams are 352×240 pixels at 1.5 Mb/s and 704×480 pixels at 4.0, 6.0, 8.0, and 12.0 Mb/s. Our results show that the relative energy savings of reconfigurable multipliers over nonreconfigurable ones can exceed 65%.

5. Conclusion

In this paper, we have presented a novel methodology for designing low-energy reconfigurable datapaths that are capable of adapting their pipeline depth in real time to fine-grain variations of their workload. To assess the effectiveness of our approach, we applied it to the design of a reconfigurable pipelined multiplier that was used in MAC-based IDCT modules for MPEG-2 MP@ML. On the basis of multiplier dissipation, the reconfigurable IDCT modules achieved relative power reductions of up to 65% compared with their conventional counterparts.

Our work opens the way to a number of interesting research issues. From an application standpoint, a promising research direction is the investigation of our design methodology for the realization of energy-efficient video processing building blocks such as dequantizers and motion compensators. Another interesting research direction is the application of our reconfigurable datapath design methodology in conjunction with compiler scheduling for reducing the power dissipation of general-purpose microprocessors.

Acknowledgments

This research was supported in part by the U.S. Army Research Office under Grant No. DAAD19-99-1-0304 and an AASERT Grant No. DAAG55-97-1-0250.

References

1. T. Lang, E. Musoll, and J. Cortadella, "Individual Flip-Flops with Gated Clocks for Low Power Datapaths," *IEEE Trans. Circuits & Syst. II: Analog & Digital Signal Process.* **44**, 507–516 (June 1997).
2. T. Meng, B. Gordon, E. Tsern, and A. Hung, "Portable Video-on-Demand in Wireless Communication," *Proc. IEEE* **83**, 659–680 (April 1995).
3. V. Stojanovic and V. G. Oklobdzija, "Comparative Analysis of Master-Slave Latches and Flip-Flops for High-Performance and Low-Power Systems," *IEEE J. Solid-State Circuits* **34**, 536–548 (April 1999).
4. W. Ye and M. J. Irwin, "Power Analysis of Gated Pipeline Registers," *Proceedings of the IEEE International ASIC/SOC Conference*, 1999, pp. 281–285.
5. B. G. Haskell, P. G. Howard, Y. A. LeCun, A. Puri, J. Ostermann, M. R. Civanlar, L. Rabiner, L. Bottou, and P. Haffner, "Image and Video Coding—Emerging Standards and Beyond," *IEEE Trans. Circuits & Syst. for Video Technol.* **8**, 814–837 (November 1998).
6. N. H. E. Weste and K. E. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, Addison-Wesley Publishing Co., Inc., Reading, MA, 1993.
7. T. Xanthopoulos and A. P. Chandrakasan, "A Low-Power IDCT Macrocell for MPEG-2 MP@ML Exploiting Data Distribution Properties for Minimal Activity," *IEEE J. Solid-State Circuits* **34**, 693–703 (May 1999).
8. S. Kim, C. H. Ziesler, and M. C. Papaefthymiou, "Reconfigurable Pipelined IDCT for Low-Energy Video Processing," *Proceedings of the IEEE International ASIC/SOC Conference*, 2000, pp. 13–17.

Received November 11, 2002; accepted for publication March 30, 2003

Suhwan Kim *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (suhwan@us.ibm.com)*. Dr. Kim received the B.S. and M.S. degrees in electrical engineering and computer science from Korea University, Korea, in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 2001. From 1993 to 1997, he was with LG Electronics, Seoul, Korea, where he designed several multimedia systems-on-a-chip (SoCs), including an MPEG-2 CODEC for audio, video, and system. In 2001 he joined the IBM Thomas J. Watson Research Center, where he is currently a Research Staff Member. His research interests encompass circuits and technology for low-power and high-performance SoC and low-power design methodologies for high-performance, VLSI signal processing. Dr. Kim received the 1991 Best Student Paper Award of the IEEE Korea Section and First Prize in the VLSI Design Contest of the 2001 ACM/IEEE Design Automation Conference. He has participated several times in the Organizing Committee and Technical Program Committee of the IEEE International ASIC/SoC Conference and in the Technical Committee of the International Symposium on Low Power Electronics and Designs.

Conrad H. Ziesler *Advanced Computer Architecture Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan 48109 (cziesler@eecs.umich.edu)*. Mr. Ziesler received the B.S. degree in electrical engineering in 1999 from the California Institute of Technology, Pasadena, and the M.S. degree in electrical engineering and computer science in 2002 from the University of Michigan, Ann Arbor. He received First Prize in the VLSI Design Contest of the 32nd ACM/IEEE Design Automation Conference, and an NSF Graduate Research Honorable Mention. Mr. Ziesler is currently pursuing his Ph.D. at the University of Michigan, with research interests including energy-recovering circuits for low-energy and high-performance VLSI systems, as well as parallel architectures and algorithms for scientific computing.

Marios C. Papaefthymiou *Advanced Computer Architecture Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan 48109 (marios@eecs.umich.edu)*. Dr. Papaefthymiou received the B.S. degree in electrical engineering from the California Institute of Technology in 1988 and the S.M. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology in 1990 and 1993, respectively. After a three-year term as Assistant Professor at Yale University, Dr. Papaefthymiou joined the University of Michigan, where he is currently Associate Professor of Electrical Engineering and Computer Science and Director of the Advanced Computer Architecture Laboratory. His research interests encompass algorithmic, architectural, and circuit issues in the design of VLSI systems, with a primary focus on energy and timing optimization. Dr. Papaefthymiou is also active in the field of parallel and distributed computing. Among other distinctions, he has received an ARO Young Investigator Award, an NSF CAREER Award, and several IBM Partnership Awards. Furthermore, together with his students, he received a Best Paper Award in the 32nd ACM/IEEE Design Automation Conference and First Prize (Operational Category) in the VLSI Design Contest of the 38th ACM/IEEE Design Automation Conference. Dr. Papaefthymiou is an Associate Editor for the *IEEE Transactions on the Computer-Aided Design of Integrated Circuits*, the *IEEE Transactions on*

Computers, and the *IEEE Transactions on VLSI Systems*. He has served as the General Chair and as the Technical Program Chair for the ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems. He has also participated several times in the Technical Program Committee of the IEEE/ACM International Conference on Computer-Aided Design.