Input-Based Dynamic Reconfiguration of Video Encoding by Approximate Arithmetic Units

[1]MANASA, [2]TALAPALA SUPRIYA
[1]Assistant Professor, [2] M.TECH STUDENT
Dept of ECE,Megha Institute of Engineering & Technology For womens, Edulabad, Ghatkesar mandal,RangaReddy Dist,Telangana ,India

***Abstract*—** Image and video compression algorithms, such as JPEG, MPEG, and so on, are particularly attractive candidates for approximate computing which can be exploited to realize highly power-efficient implementations of these algorithms. However, existing approximate architectures typically fix the level of hardware approximation statically and are not adaptive to input data. For example, if a fixed approximate hardware configuration is used for an MPEG encoder the output quality varies greatly for different input videos. Architecture for MPEG encoders that optimizes power consumption with the goal of maintaining a particular Peak Signal-to-Noise Ratio (PSNR) threshold for any video. Toward this end, we design reconfigurable adder/sub tractor blocks (RABs), which have the ability to modulate their degree of approximation, and subsequently integrate these blocks in the motion estimation and discrete cosine transform modules of the MPEG encoder. We propose two heuristics for automatically tuning the approximation degree of the RABs in these two modules during runtime based on the Characteristics of each individual video. Note that although the proposed reconfigurable approximate architecture is presented for the specific case of an MPEG encoder, it can be easily extended to other DSP applications.

Index Terms: Approximate circuits, approximate, computing, low power design, quality configurable.

I INTRODUCTION

The merit of the encoding operation can be determined from the output quality of the decoded video. Objective metrics, such as Peak Signal-to-Noise Ratio (PSNR), and so on have a very good correlation with the subjective procedures of measuring the quality of the videos. Hence, we have utilized the popular and simple PSNR metric as a means of video quality estimation.

PSNR is a full-reference video quality assessment technique, which utilizes a pixel-to-pixel difference with respect to the original video. PSNR of a video is defined as the average PSNR over a constant number of frames (50) of the video.

Images and videos differ in a variety of properties, such as color, resolution, brightness, contrast, saturation, blur, format, and so on. Thus, a naive static approximation technique, which provides satisfactory viewing quality for some specific types of videos, will fail to give adequate quality for some others. In that case, the viewing experience is significantly worsened if the approximate mode is not customized for the present type of video being watched. This is not possible for fixed hardware, and therefore a need arises for reconfiguring the architecture based on the characteristics of the video being viewed.

To support this claim, present the PSNR variation of different videos when encoded using an MPEG encoder that used a fixed approximation technique. An approximation mode, we have chosen approximation mode 5 from for implementing the fixed approximation hardware. We replaced all the adders/ subtractors in the ME and the DCT blocks with approximate versions.

II APPROXIMATE ARITHMETIC UNITS

In current literature, several approximate methods for the DCT calculation have been archived

.While not computing the DCT exactly, such approximations can provide meaningful estimations at low complexity requirements. In particular, some DCT approximations can totally eliminate the requirement for floating-point operations all calculations are performed over a fixed-point arithmetic frame work Works addressing 16-point DCT approximations are also in references. This implies null multiplicative complexity, because the required operations can be implemented exclusively by means of binary additions and shift operations.

Such DCT approximations can provide low cost and low power designs and effectively replace the exact DCT and other DCT-like transforms. Christo Ananth et al. [2] proposed a system which contributes the complex parallelism mechanism to protect the information by using Advanced Encryption Standard (AES) Technique. AES is an encryption algorithm which uses 128 bit as a data and generates a secured data. In Encryption, when cipher key is inserted, the plain text is converted into cipher text by
Using complex parallelism. Similarly, in decryption, the cipher text is converted into original one by removing a cipher key. The complex parallelism technique involves theprocess of Substitution Byte, Shift Row, Mix Column and Add Round Key. The above four techniques are used to involve the process of shuffling the message. The complex parallelism is highly secured and the information is not broken by any other intruder. For certain applications, low circuit complexity and/or power consumption is the driving factor, while for certain other applications, highest picture quality for reasonably low power consumption and/or complexity may be more important. Such feature would be invaluable in high quality smart video devices demanding extended battery life. Thus, the availability of a suite of fast algorithms and implementation libraries for several efficient DCT approximation algorithms may be a welcoming contribution.

III POWER REDUCTION IN INTEGRATEDCIRCUITS

Although the above discussion has motivated the ascendancy of power to the attention of the designer, it is important to understand the place that power has relative to other objectives. After guaranteeing correct digital functionality, the primary consideration for system designers has always been, and continues to be speed. A circuit is specified to operate at a particular delay, otherwise the entire system does not work; further reduction in the delay is beneficial but not strictly necessary. The power dissipation characteristics of a system, on the other hand, are often a consequence of the delay specification. Once the delay of the system is achieved, package cooling and/or battery resources will be allocated appropriately (within reason). Other factors may have equal or greater importance than power dissipation: area of implementation and yield/reliability issues are subjects which the designer must take into account. Nevertheless, the increase in power dissipation in successive generations of integrated circuit technologies has progressed at such a pace that it is now one of the primary considerations for designers.

A major complication in microelectronic circuits is the fact that many design decisions involve a power-delay trade-off; one cannot be lowered without raising the other. It is important to note, however, that power reduction techniques are not necessarily negatively correlated to delay reduction. For example, one method to reduce delay in a circuit's critical path is to upsize the driving strength of gates, which also results in increased power dissipation. However, another way to lower delay might be to reduce interconnect capacitance, which reduces both power and delay. Generally, great power savings can be achieved if delay is not an issue, but optimizing power without considering delay is trivial. Power reductiontechniques are applied at all levels of the design hierarchy. At the most basic level, the parameters of the lithographic process in which the integrated circuit is manufactured may be modified. Doping concentrations, minimum geometrical spacing's of structures, etc., all affect power dissipation.

Many of these parameters are beyond the control of the circuit designer. Some of these ‘global’ constraints, however, may be modified at various stages of the design methodology, although they are by and large left alone. For example, lowering of Vdd is a well-accepted method of reducing chip power, and there has been a constant trend towards running even the most high-performance microprocessors at lower supply voltages, despite the delay penalty that low supply voltage incurs. (Lowering Vdd goes hand-in-hand with other scaling schemes, such as gate oxide reduction, line-width scaling, etc. Voltage reduction may be seen as a consequence of scaling, and attendant power reduction is a serendipitous result.) Power reduction techniques applied to the design hierarchy for digital CMOS devices can be subdivided as follows:

Chip/Floor Plan Level

At this point, the power characteristics for the entire die are planned and power/delay ‘budgets’ are allocated. Vdd and Vth are determined based on performance goals. Power due to the assembly of macro blocks (interconnect and clock nets) is optimized, subject to delay and area/routability constraints. As macro blocks are instantiated, this high level information is updated and budgets may be adjusted to increase/decrease specifications on other sub-blocks.

Macroblock Level

This stage comprises the assembly of gates into a basic function such as a block of control logic, or an arithmetic unit. In a standard cell design methodology, it is at this point that the implementer’s intellectual property enters the design flow. As such, the various methods of implementing a particular function impact both power and delay. The number of power optimization techniques available at this level is numerous.

Gate/Circuit Level

This is lowest level stage visible to the designer, where transistors are assembled into basic gates. In a standard cell methodology, the fundamental gates used in macro block assembly are defined here. Power optimizations are very restricted at this level, as delay specification often dictates the power at which a device will operate. Emphasis is placed on reducing device parasitics and area while maximizing routability. A full custom approach may succeed in achieving lower powerthan a standard cell approach, as individual transistor sizing may overcome certain obvious inefficiencies.

More aggressive circuit families, such as dynamic logic, dual rail logic, or low-swing logic can be implemented at this stage, but these often require complicated design flows (e.g., through the introduction of clocks, noise sensitive nets, the need for level conversion, etc.) In our work, we focus on a standard cell CMOS methodology, as this is the most common Method for quickly and efficiently assembling a digital integrated circuit.

As such, a number of clever and useful circuit techniques cannot be applied due to the complications which their use introduces in the design flow. Therefore, the performance of designs achievable by a standard cell based implementation is sub-optimal, but the ease of implementation of such a flow allows exploration of a wide range of designs.

IV DELAY AND POWER IN MULTIPLIERS

For digital signal processing, throughput is a major concern. DSP algorithms are often related to perception of audio/visual stimuli, for example, image or voice transmission and recognition. In these tasks, precision requirements are less stringent than for other applications (e.g., numerical algorithms for scientific computing) so small bit-width multipliers may be used— latency is a function of bit width, and small multipliers do not create long delay paths. For many DSP applications, the relevant limiting specification is throughput.

These tasks often require fairly coarse resolution of images but operate on a large amount of data representing different image or sound samples. For example, image rendering requires performing computations on a large number of polygons, whereas the precision involved (bits required for identifying a particular color, bits required

One way of processing this large number of computations quickly can be achieved by lowering the latency of the multiplication; in this manner, the multiplier can start performing the next operation sooner.

A more efficient method to increase the number of computations is to increase the throughput. Various schemes are possible; for example, pipelining/interleaving of data allows one functional unit to compute several operations concurrently, while implementing multiple devices on one chip simply increases the throughput by the number of additional units. These techniques tend to bemore efficient than latency reduction, because if one tries to lower the delay of a circuit, diminishing returns are quickly encountered (if a circuit's transistors are upsized, at a certain point, the delay does not decrease further.) When optimizing throughput, on the other hand, for each additional functional block that is added, the number of operations which may be computed in a given amount of time increases by one. Although pipelining is good for throughput, it may be hard to implement in tightly coupled hardware/software systems.

While the logic implementation of pipelining is fairly straightforward, getting compilers to build programs to take advantage of pipelining is often difficult. The problem lies in setting up a series of operations to begin execution while other operations have yet to finish. These 'parallel' or 'multiple-issue' modes of system behavior have timing dependencies which complicate the task of writing a compiler that can take advantage of such hardware techniques. Moreover, while some DSP algorithms lend themselves to parallel operation, others require processing to be more sequential in order, rendering the additional pipeline hardware useless. Finally, extremely high speed code is often implemented by hand in assembly language. Understanding all the methods of optimizing a pipelined function can be very tedious if done manually. These reasons argue for using multiple multiplication functional units on one chip, as opposed to implementing a heavily pipelined multiplier. The multiplier is a fairly large block of a computing system.

The amount of circuitry involved is proportional to the square of its resolution (i.e., a multiplier of size n bits, has O(n2) gates.) Not only is the multiplier a high delay block, but it can be a significant source of power dissipation. Based on the argument delineated above, that several multipliers should be present on-chip as more DSP compute power is needed, the power dissipation involved in multiplication will become more dominant. (Even if the pipelined approach is used, to a first order, the pipelined multiplier will dissipate as much power as several multiplier blocks. Although a pipelined version has fewer gates, it will still experience roughly the same amount of switching.) Therefore, digital multipliers have become one of the prime circuits targeted for power reduction.

Power Vs Energy

The distinction between the terms power and energy is important to this discussion. Note that energy is a measure of the total number of Joules dissipated by a circuit, whereas power refers to the number of Joules dissipated over a certain amount of time. Properly speaking, power reduction is a different goal than energy reduction. Power can be a problem primarily when heat removal is a concern. If too many Joules of energy areconverted into heat within a short amount of time, a package's heat sink may not be able to redistribute this heat quickly enough; the result will be a rise in temperature and subsequent thermal failure.

If the same amount of energy is generated over a longer amount of time, power dissipation is reduced and the cooling structure can better deal with the thermal demands of the circuit. Here, power reduction consists of reducing the case when a large amount of energy is dissipated in a short amount of time. Again, the total energy dissipated may remain the same. Energy reduction consists of reducing the total number of Joules to be dissipated. Therefore, we often speak of energy per operation as the metric to be optimized; time is not a factor in this calculation. Power reduction, then, belongs to the domain of thermal reliability, whereas energy reduction lies in the domain of maximizing battery lifetime. In digital CMOS, one often hears of a power-delay trade-off, or of a circuit operating at a point in the power*delay space. This power*delay is continually being improved, (e.g., using more advanced processes or better logic designs.) In a sense, this is a misnomer as power*delay = (energy/delay)*delay = energy; this implies one should minimize energy—or more importantly, minimizing delay is irrelevant. Instead, one should speak of minimizing energy*delay, as this metric involves two independent measures of circuit behavior.

The literature consistently refers to minimizing power, whereas the techniques described in almost all cases involve minimizing energy. The two terms tend to be used interchangeably, with ‘power’ being used where ‘energy’ should be used instead. This usage most likely stems from this field of research being known as low-power circuit operation. We will retain the use of ‘power’ because of its standard usage but will use ‘energy’ where clarity is warranted.

We demonstrate that, for a fixed level of hardware approximation in an MPEG encoder, the output quality varies widely across different videos, often going below acceptable limits. This shows that setting the level of hardware approximation statically is insufficient. We investigate, for the first time, the use of dynamically reconfigurable approximate hardware architectures that vary the DA during run-time across multiple computational cycles, depending on the inputs. Toward this end, we propose the design of reconfigurable adder/subtractor blocks (RABs) for four commonly used adder architectures, viz., ripple carry adder (RCA), carry lookahead adder (CLA), carry bypass adder (CBA), and carry select adder (CSA), and subsequently integrate them into the MPEG encoder to enable quality configurable execution We have implemented the proposed architecture for an MPEG encoder on an Altera DE2 field-programmable gate array (FPGA) board and evaluated it using eight benchmark videos. Our experimental results show that the proposed architecture results in power savings equivalent to a baseline approach that uses fixed approximate hardware while respecting quality constraints across different videos

V PROPOSED SYSTEM TECHNIQUE

Main Module’s

- 1-bit DMFA
- 8-bit reconfigurable RCA block.
- 1-bit dual-mode carry propagate generate blocks. 8-bit reconfigurable CLA block.

Module Description

1-BIT DMFA

Dynamic variation of the DA can be done when each of the adder/subtractor blocks is equipped with one or more The latter one can also be conceptualized as an enhanced version of truncation as it just relays the two 1-bit inputs, one as Sum and the other as Carry Out

In case A, B, and Cin are the 1-bit inputs to the full adder (FA), then the outputs are Sum = B and Cout= A. The proposed scheme replaces each FA cell of the adders/subtractors with a dual-mode FA (DMFA) cell in which each FA cell can operate either in fully accurate or in some approximation mode depending on the state of the control signal APP.

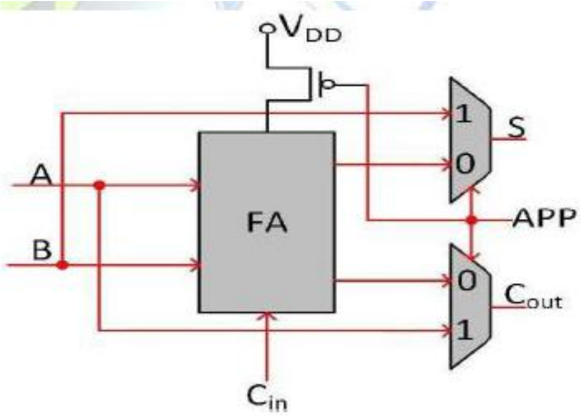


Figure 1:1 bit DMFA

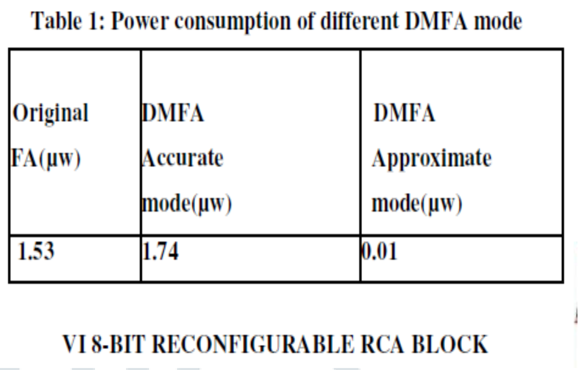


Table 1: Power consumption of different DMFA mode

Original FA(µw)	DMFA Accurate mode(µw)	DMFA Approximate mode(µw)
1.53	1.74	0.01

VI 8-BIT RECONFIGURABLE RCA BLOCK

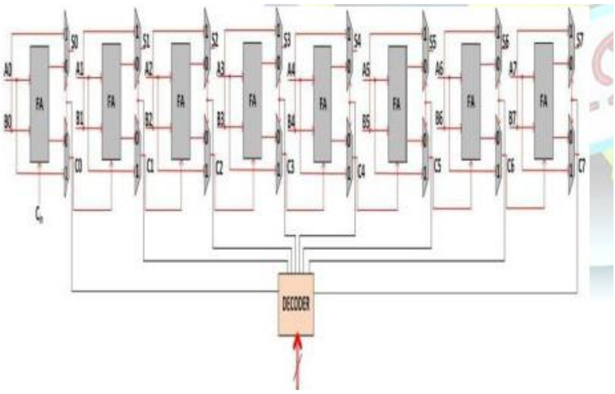
It is important to note that the FA cell is power-gated when operating in the approximate mode. Synthesis and evaluation of power consumption of a 16-bit RCA were performed in Synopsys Design and Power Compiler and the corresponding results are described in Table.

Our experiments have shown a negligible difference in the power consumption of DMFA when operated in either of the two approximation modes. Hence, without any loss of generality, approximation 5 was chosen for its higher probability of giving the correct output result than truncation, which invariably outputs 0 irrespective of the input.the logic block diagram of the DMFA cell, which replaces the constituent FA cells of an 8-bit RCA

In addition, it also consists of the approximation controller for generating the appropriate select signals for the multiplexers. A multimode FA cell would provide even a better alternative to the DMFA from the point of controlling the approximation magnitude.

However, it also increases the complexity of the decoder block used for asserting the right select signals to the multiplexers as well as the logic overhead for the multiplexers themselves.

This undermines the primary objective as most of the power savings that we get from approximating the bits are lost. Instead, the two-mode decoder and the 2:1 multiplexers have negligible overhead and also provide sufficient command over the approximation degree.



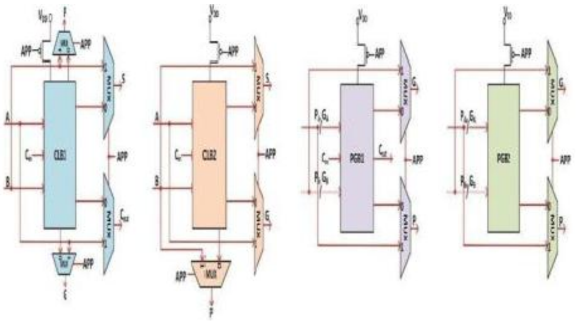


Figure 3:8 bit reconfigurable RCA block

Figure 2: 1-bit dual-mode carry propagate generateblocks

VII 8-BIT RECONFIGURABLE CLA BLOCK

Other varieties, like CLA and tree adders, use different types of carry propagate and generate blocks as their basic building units, and hence require someadditional modifications to function as RABs. As an example, we implemented a 16-bit CLA consisting of four different types of basic blocks depending upon the presence of sum (S), Cout, carry propagation (P), and carry generation (G) at different levels.

We address the basic blocks present at the first (or lowermost) level of a CLA, which have inputs coming in directly, as carry lookahead blocks, CLB1 and CLB2. The difference among them being that CLB1 produces an additional Cout signal compared with CLB2. Their corresponding dual-mode versions, DMCLB1 and DMCLB2, have both S and P approximated by input operand B and both Cout and G approximated by input operand A, as shown in Fig.

The basic blocks present at the higher levels of CLA hierarchy are denoted as propagate and generate blocks, PGB1 and PGB2. In this case, PGB1 produces an extra Cout output as compared with PGB2. As shown in the configurable dual-mode versions, DMPGB1 and DMPGB2, use inputs PA and GB as approximations for outputs P and G, respectively, when operating in the approximate mode.

These approximations were selected empirically ensuring that the ratio of the probability of correct output to the additional circuit overhead for each of the blocks is large For a reconfigurable CLA, DMCLB1 and DMCLB2 blocks are approximated in accordance with the DA.

However, the DMPGB1 and DMPGB2 blocks are approximated only when each and every DMCLB1, DMCLB2, DMPGB1, and DMPGB2 block, which belongs to the transitive fan-in cones of the concerned block, is approximated. Otherwise, the block is operated in the accurate mode.

For example, any DMPGB block at the second level of CLA can be made to operate in approximate mode, if and only if, both of its constituent DMCLB1 and DMCLB2 blocks are operating in the approximate mode.

Similar protocol is ensued for the blocks residing at higher levels of the tree, where each DMPGB block can be approximated only when both of its constituent DMPGB1 and DMPGB2 blocks are approximated. This architecture can be easily extrapolated to other similar type CLAs.

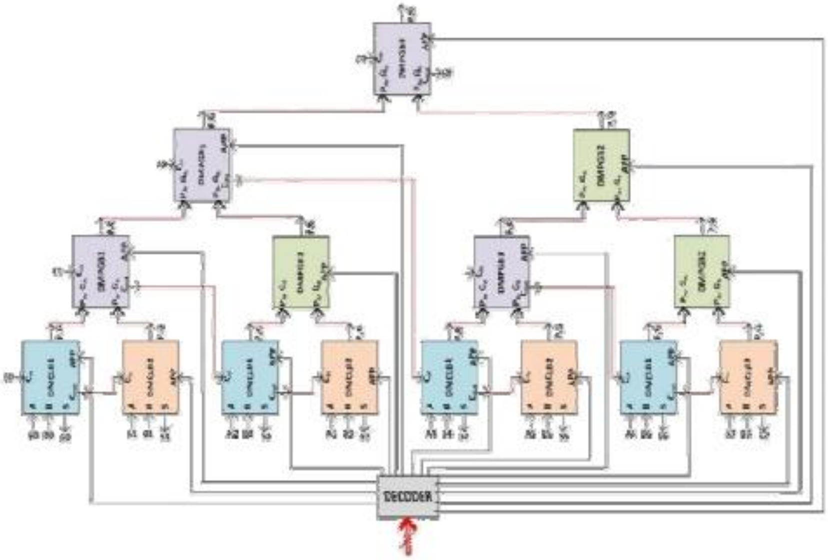


Figure 4: 8 bit Reconfigurable CLA block

Similar protocol is ensued for the blocks residing at higher levels of the tree, where each DMPGB block can be approximated only when both of its constituent DMPGB1 and DMPGB2 blocks are approximated. This architecture can be easily extrapolated to other similar type CLAs.

VIII CONCLUSION

We proposed a reconfigurable approximate architecture for the MPEG encoders that optimize power consumption while maintaining output quality across different input videos. The proposed architecture is based on the concept of dynamically reconfiguring the level of approximation in the hardware based on the input characteristics. It requires the user to specify only the overall minimum quality for videos instead of having to decide the level of hardware approximation. Our experimental results show that the proposed architecture results in power savings equivalent to a baseline approach that uses fixed approximate hardware while respecting quality constraints across different videos. Future work includes the incorporation of other approximation techniques and extending the approximations to other arithmetic and functional blocks.

REFERENCES

[1] M. Elgamel, A. M. Shams, and M. A. Bayoumi, “A comparative analysis for low power motion estimation VLSI architectures,” in Proc. IEEE Workshop Signal Process. Syst. (SiPS), Oct. 2000, pp. 149–158.

[2] Christo Ananth, H.Anusuya Baby, “High Efficient Complex Parallelism for Cryptography”, IOSR Journalof Computer Engineering (IOSR-JCE), Volume 16, Issue 2, Ver. III (Mar-Apr. 2014), PP 01-07

[3] I. S. Chong and A. Ortega, “Dynamic voltage scaling algorithms for power constrained motion estimation,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 2.Apr. 2007, pp. II-101–II-104.

[4] I. S. Chong and A. Ortega, “Power efficient motion estimation using multiple imprecise metric