Proceedings of the 10th WSEAS International Conference on SYSTEMS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp246-250)

A Reconfigurable VLSI Architecture for Low Power IDCT

HAMED ELSIMARY
Microelectronics Department, Electronics Research Institute
CAIRO, EGYPT

Abstract: - This paper presents a low power architecture for matrix-vector multiplier of a constant matrix by a vector, which is the main part of the Discrete Cosine Transform (DCT), and the inverse Discrete Cosine transform (IDCT) . The proposed architecture makes use of the fact that the input data are mostly zeros, hence it is more suitable for performing IDCT. It avoids unnecessary arithmetic operations by quickly terminating the multiplication by zero and significantly reduces the power and delay. For further reduction in power dissipation, the proposed architecture is designed in a way to eliminate the RAM required for the transposition matrix, as it latch the elements of the matrix and continuously dissipate power. The multiplier circuit play a crucial role in the system and heavily contributes to the power dissipation, thus a special attention is paid to its design. The multiplier's architecture and its simulation results are presented.

Key-Words: - Low power circuits, IDCT, multiplier

1 Introduction

Video processing is a key component of multimedia communications. It has been become an integral part of currently wide spreading portable devices. The ever increasing demand of video processing and multimedia applications for such portable devices has made their low-power realization extremely important.

Currently there are several video standards established for different purposes, including MPEG (Moving Picture Expert Group). Specifically MPEG-2 is an industrial standard for video processing and is widely used in multimedia applications such as digital video broadcast (DVB), digital versatile discs (DVD), high definition television (HDTV), and video on demand (VOD). Implementations of this video standard for mobile systems on chip requires to provide substantial computing capabilities at low power dissipation [1].

Different approaches has been explored to obtain power efficient designs from the algorithmic level, architectural level, and circuit levels [2],[3]. Operating circuits at lowest possible voltage is an efficient technique, for reducing power dissipation, however this is limited by the available target technology.

In general, power consumption in circuit has two components - dynamic power and static power. Traditionally the dynamic component was by far the most dominant, but this is changing due to physical phenomena as process technologies shrink below 90nm. It is estimated [3] that when devices are scaled to 45nm (around the year 2007) the static component will be equal to the dynamic component. Static power is usually tackled using a variety of dynamic power management (DPM) techniques. If a sub-system is not required to process data at all, it may be shutdown in order to avoid unnecessary power consumption. If the sub-system is still required but has a reduced load, a DPM controller can scale the operating frequency and voltage to save power based on the dynamic processing load. Video processing is very non-uniform by nature so there is a scope to apply DPM techniques to video processing architectures for energy conservation purposes. Dynamic power consumption is caused by circuit node switching activity. To reduce this component, a system designer primarily attempts to reduce the complexity of the required processing on the premise that if there are less operations to be carried out, there will be less switching and hence less energy dissipation. It is generally accepted that most power savings are achieved at the higher levels of abstraction (algorithmic and architectural levels) since there are wider degrees of design freedom [4].

Applying low power techniques at the logic, circuit and physical levels are also important but in general depend on the target technology available with which the optimized architecture is to be implemented.

Energy is the time integral of power; if power consumption is a constant, energy consumption is simply power multiplied by the time during which it is consumed. Although they may not explicitly say so, most designers are actually concerned with reducing energy consumption. This is because batteries have a finite supply of energy (as opposed to power, although batteries also put limits on peak power consumption as well). Since most components are fabricated using CMOS technology. The sources of energy consumption on a CMOS chip can be classified as static and dynamic power dissipation. *Static* energy consumption is caused by short circuit currents (*Psc*), bias (*Pb*) and leakage currents (*Pl*). *Dynamic* energy consumption (*Pd*) is caused by the actual effort of the circuit to switch.

$$P = Pd + Psc + Pb + Pl \qquad (1)$$

The contributions of this static consumption are mostly determined at the circuit level. During the transition on the input of a CMOS gate both *p* and *n* channel devices may conduct simultaneously, briefly establishing a short fr om the supply voltage to ground. This effect causes a power dissipation of approx. 10 to 15%. Also, lower operating voltages as being used nowadays, tend to reduce the short circuit component. While statically-biased gates are usually found in a few specialized circuits such as PLAs, their use has been dramatically reduced in CMOS design [3]. Leakage currents also dissipate static energy, but are also insignificant in most designs (less than 1%). In general we can say that careful design of gates generally makes their power dissipation typically a small fraction of the dynamic power dissipation, and hence will be omitted in further analysis. The dominant component of energy consumption (85 to 90%) is CMOS is therefore *dynamic*. A first or der approximation of the dynamic power consumption of CMOS circuitry is given by the formula:

$$P_d = C_{eff} V^2 f \qquad (2)$$

where P_d is the power in Watts, C_{eff} is the effective switch capacitance in Farads, V is the supply voltage in Volts, and f is the frequency of operations in Hertz [4]. The power dissipation arises from the charging and discharging of the circuit node capacitances found on the output of every logic gate. Every low-to-high logic transition in a digital circuit incurs a voltage change ΔV, drawing energy from the power supply. *Ceff* combines two factors C, the capacitance being charged/discharged, and the activity weighting α, which is the corresponding probability that a transition occurs.

$$Ceff = \alpha\, C \qquad (3)$$

A designer at the technological and architectural level can try to minimize the variables in these equations to minimize the overall energy consumption. However, as will be shown in the next sections, power minimization is often a subtle process of adjusting parameters in various trade-offs.

In FPGA devices Inrush current is device-specific. For example, SRAM-based FPGAs have a high inrush current because on power-up these devices are not configured and need to actively download data from external memory chips to configure their programmable resources, such as routing connections and lookup tables. Conversely, anti-fuse-based FPGAs do not have a high inrush current since they do not require power-on configuration.

Much like inrush power, standby power depends heavily on the electrical characteristics of a component. Due to the extensive number of SRAM cells within SRAM FPGA interconnects, they can consume hundreds of milliamps even at standby. Since anti-fuse FPGAs have metal-to-metal interconnects, they do not require the additional transistors, and hence power, to retain interconnects. However, for both FPGA process types, leakage current increases as process geometry shrinks, which exacerbates the power problem.

As an additional dilemma, dynamic power can easily be several times greater than standby power. Dynamic power is proportional to the frequency of charging and discharging of internal parasitic capacitances of a component, such as registers and combinatorial logic, so optimizations are generally design-oriented.

To cut the power down in FPGA systems, several techniques can be followed such as:

- The lower utilization of one-hot state machines vs. Gray and binary state machines results in a more power-efficient design.
- Guarded evaluation. The key to guarded evaluation is to stop inputs from propagating down to additional logic blocks if the resulting outputs do not require updating
- Gated clocks. Temporarily unused modules can have their clocks slowed or stopped. Gating a clock contributes to a significant amount of power savings because the number of active clock buffers reduces the number of toggling flip-flops decreases, and consequently the fan-out of those flip-flops will be less likely to toggle.

On the system level, power saving can be achieved on:

- System clock speeds. System clock frequency has a dramatic impact on the overall power consumption of a board. To achieve an optimum balance between power and throughput, a slower clock can be supplied to components that do not require a fast clock. For devices that are critical to bandwidth, provide a fast clock, or use a built-in phase-locked loop (when available) to internally generate a fast clock for the specific modules that require faster performance.

- Component enabling. A system controller can deactivate the enable signal to a device when that device is irrelevant to the current operation, or put a device in its sleep mode when that device will not be accessed for an extended period of time.

In this work, FPGA is used as the target technology, lower power is achieved on the architectural level through:

- Elimination of the RAM required for holding the transposition matrix during the computation of the IDCT, which continuously unconditionally dissipate power. Elimination of this RAM will also leads to parallel execution of the different parts of the algorithm, and result in operation of the circuit at lower clock frequency, and hence low power dissipation.
- Design of a mask signal to generate a control signal to disable the multiplier in the case of zero-valued data.

In the following section, an overview of the IDCT architecture then followed by a presentation of the proposed architecture.

2. Architectures for 2-D IDCT

Various DCT/IDCT algorithms have been studied, among which the most implemented algorithm in VLSI is the Chen Algorithm [6]. It reduces the number of required additions and multiplies while still maintaining regularity. The basic step in this algorithm is the multiplication of a constant coefficient matrix by a data vector typically implemented with a set of multiply accumulators.

The 2-D IDCT is one of the most compute intensive tasks in MPEG video processing . The 2-D IDCT of an 8x8 coefficient block can be separated into two sequential 8-point, one dimensional IDCT's using row column decomposition technique. This decomposition technique is proffered for VLSI implementation due to its regularity, the 8-point 1-D IDCT is described as follows:

$$x(n) = \sum_{k=0}^{7} \frac{c(k)}{2}.z(k).\cos\left(\frac{(2n+1)k\pi}{16}\right) \qquad (4)$$

where: $c(k) = \frac{1}{\sqrt{2}}$ for K=0, $c(k) = 1$ for $c(k) \neq 0$

The 8-point 2-D IDCT can be described in matrix form can be described as :

$$X = C^T ZC \qquad (5)$$

where C is an 8x8 constant matrix.

Fig. 2, shows a block diagram for the multiply and accumulate based 2-D IDCT decomposed into two 1-D IDCT . considering the similarity between the values of the constant matrix element, the number of multiplications can be reduced.

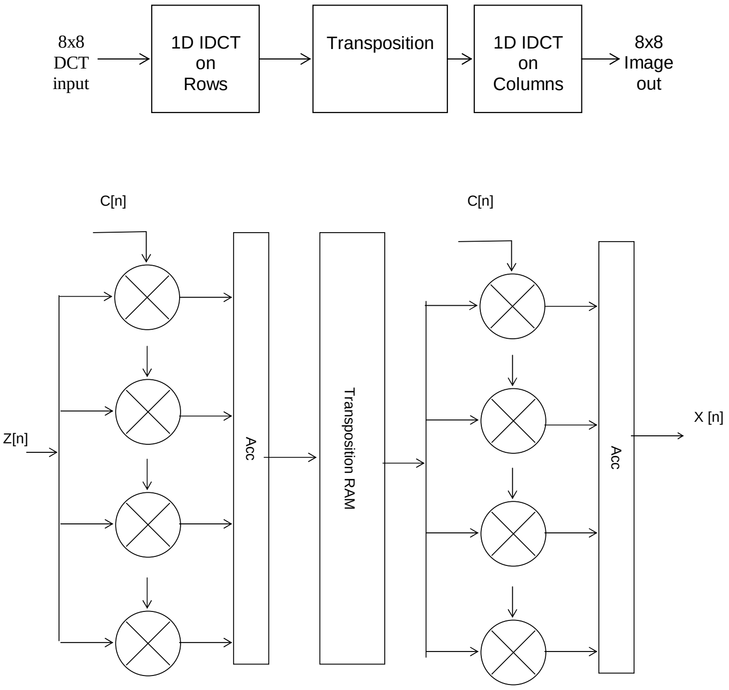


Fig. 1 Block diagram of a MAC based IDCT

3. Proposed Architecture

The strategy for reducing power dissipation is three-fold, first: elimination of the transposition memory. As explained earlier, the IDCT is a matrix multiplication, things can be organized in way to eliminate the transposition memory. Starting with the calculation of the transfer matrix from the C^T times the Z where a row of C^T time a column of Z, iterating through all columns of Z will result in an entire row of T matrix. At the time a complete row of the T matrix is ready, the second 1-D IDCT can start by calculating T times C, again it's a row of T times a row of C^T (ie column of C). This leads to parallel operation and reduced power dissipation.

Second: disabling the multiplication when a zero-valued coefficient is detected

Third: using a radix-4 based multiplier, where the number of operations are reduced leading to reduction in power dissipation

3.1 Multiplier's Architecture

The multiplier depends on serially recoding the multiplier operand into Modified Booth codes as shown in Table 1. Fig. 2 shows the architecture of a adopted multiplier. The multiplier is composed of a multiplexer which selects either the multiplicand, the multiplicand times 2 i.e shifted left one bit, the multiplicand times -1, or the multiplicand times -2 i.e shifted and complemented according to the recoded values of the multiplier. A full adder carry out the addition and the results are loaded into the result register, which initially holds the multiplicand and then replaced by the product. The multiplexer and control sequence are implemented in the control block, shown in Fig.2. The full adder and result register are implemented in the full-adder-shifter block shown in the same figure.

Table 1:Modified Booth's Recoding

$b_{i+1}\ b_i\ b_{i-1}$	Add to partial produc
0 0 0	+0A
0 0 1	+1A
0 1 0	+1A
0 1 1	+2A
1 0 0	-2A
1 0 1	-1A
1 1 0	-1A
1 1 1	-0A

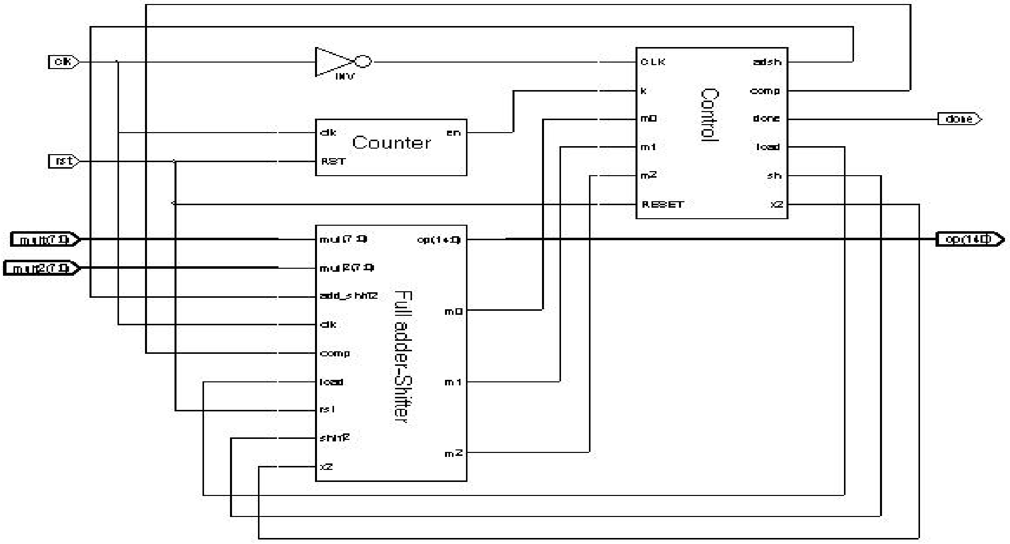
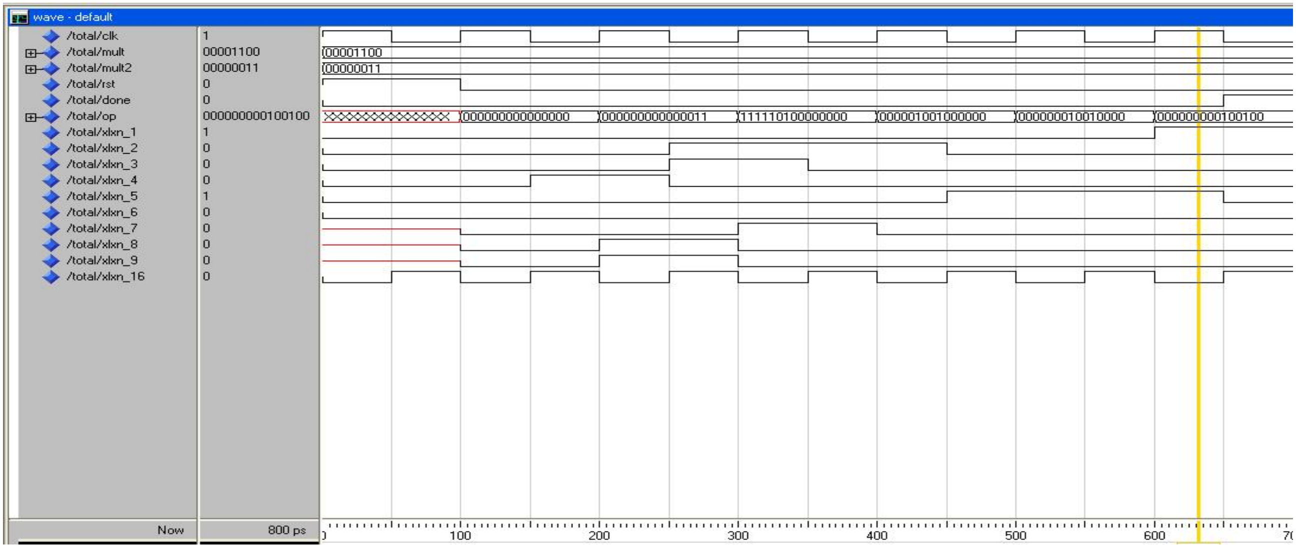


Fig. 2 Schematic diagram of the Radix-4 multiplier's Architecture



/total/clk	1
/total/mult	00001100
/total/mult2	00000011
/total/rst	0
/total/done	0
/total/op	000000000100100
/total/xlxn_1	1
/total/xlxn_2	0
/total/xlxn_3	0
/total/xlxn_4	0
/total/xlxn_5	1
/total/xlxn_6	0
/total/xlxn_7	0
/total/xlxn_8	0
/total/xlxn_9	0
/total/xlxn_16	0

00001100
00000011
XXXXXXXXXXXXXXX 000000000000000 000000000000011 111110100000000 000001001000000 000000010010000 000000000100100

Fig 3 Simulation Results of the Radix-4 Multiplier

4. Simulation results of the Multiplier

Fig. 3 shows the multipliers simulation with Multiplicand, mult=12_d, and multiplier, mult2=03_d, and result C=36_d, the result is available in five clock cycles.

5. Performance Analysis

Speed, area, and power dissipation are the important features of a multiplier. In order to evaluate the performance of the multiplier used in this design, a normal binary signed serial parallel multiplier has been implemented in the same technology, the area, speed, and power dissipation of the proposed radix-4 multiplier is compared to that radix-2 multiplier.

5.1 Speed

The proposed multiplier needs ((n + m)/4 + 1) clock cycles to generate the n + m bits of the multiplication result (where n and m are the number of bits for the multiplier and multiplicand respectively), relative to ((n + m)/2 + 2) clock cycles of the reference signed multiplier. Thus the proposed multiplier achieves (2(n + m) + 8)/((n + m) +4). For n=m=8-bits, the proposed multiplier achieves 50% improvement in the speed.

5.2 Area

The proposed radix-4 multiplier and the reference signed multiplier are synthesized using the same target technology, the synthesis report shows 1% increase in the area consumed by the proposed multiplier on the target device.

5.3 Power dissipation

Radix-4 multiplier recoding reduces the number of partial products, resulting in less area and power than normal serial parallel binary multiplication . Simulation of the power consumption showed 46% improvement in the power consumption of the proposed multiplier relative to the reference signed multiplier. This improvement has its effect on the power dissipation of the IDCT with radix-4 multiplier. To estimate the overall performance of the proposed IDCT architecture, input data statistics to extract the number of activities in the matrix-vector multiplication , based on this, the power consumption of the proposed design is lowered by a factor of 70% compared to that design using regular signed multiplier and without the techniques mentioned earlier.

6. Conclusion

A Low power architecture for IDCT computation is presented. The power reduction comes from different aspect, among which the multiplier circuit which plays a crucial role in the system and contribute to the total power dissipation. The multiplication is done in radix-4 which leads to less number of operations, and low power dissipation.

References:

[1] B.G. Haskell, P.G. Howard, Y. A. LeVum, A. Puri, J.Ostermann, M.R. Givanlar, L. Rabiner, L. Bouttou, and P. Haffner , " Image and Video Coding-Emerging Standards and Beyond", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 8, No. 7, Nov 1998., pp. 814-837.

[2] A. P. Chandrakasan, and R. W. rodersen, Minimizing power consumption in digital CMOS circuits," *Proceedings of the IEEE*, vol. 83, no. 4, , April 1995, pp. 498-523.

[3] A. Y. Wu, and K. J. R. Liu, "Algorithm-Based Low-Power Transform Coding Architecture: The Multirate Approach," *IEEE Transactions on VLSI Systems*, vol. 6, No. 4, Dec. 1998, pp. 707-718,.

[4] Sameer Patel, "Low-power design techniques span RTL-to-GDSII flow", EE Times www.eetimes.com

[5] Jason K. Lew, "Low Power System Design Techniques Using FPGAs" EE Times www.eetimes.com

[6] W. H. Chen, C. H. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications*, Vol. COM-25, Sept. 1977. pp.1004-1009,

.