

# A POWER ESTIMATION FRAMEWORK FOR DESIGNING LOW POWER PORTABLE VIDEO APPLICATIONS\*

Chi-Ying Tsui, Kai-Keung Chan

Qing Wu, Chih-Shun Ding, Massoud Pedram

Dept. of Elec. & Elec. Engineering,  
Hong Kong Univ. of Science and Technology,  
Clear Water Bay, H.K.

Dept. of Electrical Engineering-Systems,  
University of Southern California,  
Los Angeles, CA, USA

## ABSTRACT

*This paper presents a power evaluation framework designed for estimating power consumption of a new video telephone compression standard, ITU-H.263, at the system level. A hierarchical, mixed-level simulation environment is built and cycle-accurate power macro-modeling is used for the architectural power evaluation. Experimental results show the effectiveness of the proposed framework and models.*

## 1. INTRODUCTION

Real time video compression is required for portable multimedia devices such as wireless video-phone and hand-held digital video camcorders to reduce the bandwidth for either transmission or storage. However compression hardware consumes a lot of power. Algorithm selection and architecture design have a profound influence on the power consumption of the video processor design. Therefore it is important to have a power evaluation framework to study the algorithmic and architectural trade-off for power, cost and performance.

A number of estimation techniques have been developed to model the power consumption at the architectural level. [1] is an example. The drawback of this approach is that a single capacitance value and a single switching activity factor are assumed and hence the input statistics that affects the power dissipation cannot be accurately captured. Improved models based on the Dual Bit Type (DBT) and activity estimation in the control circuitry are proposed in [2] [3]. These power macro-models assume some statistics or properties about the input vectors and calculate the average power consumption. Recently, a cycle-based macro-model which gives a power estimate for each input vector pair was presented [4]. It has the capability of estimating power dissipation cycle by cycle during the architectural or RT-level simulation. It can thus produce not only the average power dissipation but also the power dissipation distribution over time. The computational overhead is small since the macro-

\*This work was supported in part by DARPA under contract no. F33615-95-C1627 and Hong Kong RGC Research Grant HKUST779/96E.

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 97, Anaheim, California

(c) 1997 ACM 0-89791-920-3/97/06 ..\$3.50

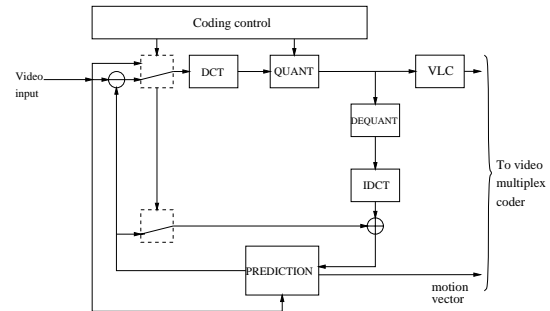


Figure 1: Overview of a H.263 encoder

model evaluation only involves a function calculation in every cycle.

Existing high level power estimation tools are usually stand-alone tools. In this paper, we introduce a hierarchical power estimation framework for video encoder system. The framework is built on a hierarchical mixed-level simulation environment. The system level simulator is designed for verifying the performance and functionality of the H.263 video-phone standard. Different algorithmic models and the corresponding architectural power macro-models were built for every signal processing element. Switching activities of each element are collected during the system simulation and the power dissipation is evaluated using the cycle-based power macro-model during the architectural simulation. Using the power estimation framework, we explore different algorithms and consider various architectural decisions for the video compression applications. In particular, several existing motion estimation architectures were compared in terms of their power consumption and performance.

## 2. TRANSFORM-BASED VIDEO COMPRESSION ALGORITHM

The most popular encoders for video compression standards such as MPEG, H.261 or H.263 employ a transform-based compression technique. Fig. 1 shows an overview of a H.263 encoder.

The main idea of the transform based video compression algorithm is to reduce the spatial redundancy within pictures (intra-frame coding) and also temporal redundancy between pictures (inter-frame coding). Spatial redundancy is exploited by a combination of differential pulse coded modulation (DPCM) and DCT-transform coding. Temporal redundancy between pictures is exploited by motion compensated prediction.

An important component in motion compensated prediction is estimating the motion between successive frames. Block matching algorithm (BMA) is commonly used to esti-

mate a motion vector for every macro-block which is of size  $N \times N$ . To find a motion vector, each macro-block in the current frame is matched with another macro-block in the previous frame within a search window. For every candidate block, the mean of the absolute difference (MAD) of the pixel values is calculated as follows:

$$MAD(m, n) = \sum_{i=1}^N \sum_{j=1}^N |x_{i,j} - y_{i+m,j+n}| \quad (1)$$

where  $x_{ij}$  and  $y_{ij}$  are the pixel values of current block and previous block, respectively, and  $(m, n)$  is the relative displacement of the candidate block. The best match is the one that has the minimum MAD.

Block matching algorithms are computationally intensive and can take up more than 50% of the computational work of the entire compression process. Application-specific VLSI architectures for BMA's are required to satisfy the computational requirement. Many architectures have been proposed for fixed size, full search and fast search BMA [5] [6]. In this work, power consumption for different full search motion estimation architectures are studied and compared.

### 3. CYCLE-BASED POWER MACRO-MODELING

Dynamic power consumption is the dominate part of the power consumption for CMOS circuits. It is dissipated when there is switching at the gate output and is given by

$$P_n = 0.5 \times V_{dd}^2 f C_L E_{sw}(n) \quad (2)$$

where  $E_{sw}(n)$  is the average number of switching per clock cycle at gate  $n$ ,  $f$  is the operating frequency,  $C_L$  is the load capacitance, and  $V_{dd}$  is the supply voltage. The switching activity of a combinational circuit depends on the internal node structure and also the switching activity at the inputs. At the system level, the detailed gate level implementation is not yet known and thus the power consumption is estimated based on the total input switching activities.

At the architectural level, power macro-models can be used to estimate the power consumption. The process in general consists of generating circuit capacitance models for some assumed data statistics for the inputs which can be gathered during behavioral simulation.

A simple power macro-model equation for the  $j$ th module in the circuit could be expressed as:

$$P_j = 0.5 \times V_{dd}^2 f \sum_{i=1}^{n_j} C_{ij} E_{sw}(ij) \quad (3)$$

where  $n_j$  is the number of inputs for the  $j$ th module,  $C_{ij}$  and  $E_{sw}(ij)$  are the effective capacitance and the switching activity of the  $i$ th pin of the  $j$ th module, respectively. Alternatively, we can also write the macro-model equation in a cycle-based form as follows:

$$P_{jk} = 0.5 \times V_{dd}^2 f d \sum_{i=1}^{n_j} C_{ij} E_{sw}(ijk) = F_j(V_{j,k-1}, V_{j,k}) \quad (4)$$

where  $P_{jk}$ ,  $E_{sw}(ijk)$  and  $V_{j,k}$  denote the power consumption, the switching value (0 or 1) for the  $i$ th input, and the input vector of module  $j$  at cycle  $k$ , respectively.  $P_{jk}$  is expressed as a function  $F_j$  of the vector pair  $V_{j,k-1}, V_{j,k}$ . This model can be used to estimate the power consumption

at each cycle. Total power consumption at cycle  $k$  is calculated by summing up the power consumption of all the modules.

The cycle-based power macro-model is built using regression analysis and is expressed as a function of input characteristics as follows:

$$P = F\{X_1, \dots, X_N\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_N X_N \quad (5)$$

where  $P$  is the power dissipation variable,  $\beta_0, \beta_1, \dots, \beta_N$  are constants called the regression coefficients or parameters of the macro-model, and  $X_1, X_2, \dots, X_N$  are characteristic variables extracted from the input vector pair. The characteristic variables can be any factors contributing to the power consumption; bit-level transition information is used here. Our macro-models for all the basic processing elements used in the motion estimation architectures are built. The macro-models were trained using the benchmark video sequences which represent different types of video applications(See Section 5).

### 4. POWER ESTIMATION FRAMEWORK

The power estimation framework is built on a mixed-level simulation environment. In the top level, a system level simulator which models the whole vector compression algorithm for the H.263 standard is built. The simulator has a modular structure. Behavioral or algorithmic models for individual processing blocks are built using C or VHDL and are put in a system library. Combination of different algorithms for different blocks are tested by picking the corresponding models from the system library. By using different parameters in the behavioral model, the impact of different algorithmic decisions such as different quantization levels, are studied. The system level simulator verifies the performance and functionality of the algorithm and at the same time allows the collection of the switching activities at the output of each module. The next level is an architectural/RT-level simulator. For each algorithm used in a signal processing module, different architectural realizations exist. Each architecture is modeled using C or VHDL and is stored in an architectural library. The basic processing components which are used to implement the architecture are stored in a component library. Cycle-based power macro-model for every processing component is built and stored in the component library. During a system level simulation cycle, if an architectural model is selected for a particular module, architectural simulation is invoked and executed as follows. The input vector to the module is simulated through the architectural model. Internal vectors which feed to the inputs of individual processing elements are generated. The power consumptions of the processing elements at that simulation cycle are computed by invoking the corresponding cycle-based power macro-models. Fig 2 shows the hierarchical simulation mechanism, using motion estimation module as an example.

### 5. IMPACT OF DIFFERENT SIGNAL PROCESSING BLOCKS ON THE SWITCHING ACTIVITIES

In this section, we present results about the impact of different signal processing blocks on the switching activities. Input vectors are derived from real world benchmark sequences. Three different video sequences which are commonly used in evaluating the performance of different video

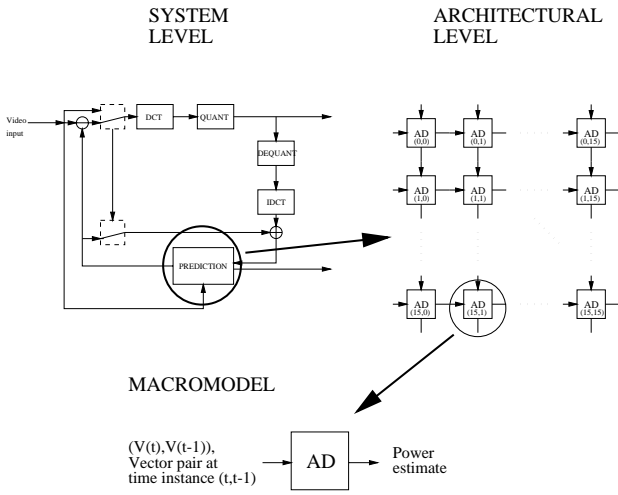


Figure 2: Hierarchical power estimation framework.

compression algorithms are employed. They are the *Football*, *Foreman*, and *Salesman* sequences which were listed in descending order of the amount of movement in the video clip. The switching activity at the input/output of each module is shown in Fig 3.

Input to the DCT module comes from either the *original* video image or the *differential* image which is the difference between the current and the previous video frame. These two kinds of images are used for the intra and inter frame coding, respectively. It can be seen that the input of the DCT module follows the DBT model described in [2]. However, the bit level signal statistics begins to deviate from the DBT model as the incoming data passes through the various levels of processing blocks. The bit switching statistics after DCT is distributed evenly for most of the bits because of the reduction in the image correlation. The results also show that the bit switching activities depend on the level of movement of the video sequences. As expected, the *Football* sequence gives the highest switching activities among the three different sequences.

After quantization, the switching activities are greatly reduced. The switching activity for *Football* is reduced by three times, while that for *Salesman* is reduced by about ten times. It is because most of the coefficients are quantized to zero after the quantization process.

The image/error image after quantization has to be de-quantized and then processed by the IDCT module. The bit transition distribution is similar to that of the quantization input, but has a smaller amplitude. It is because most coefficients are zero after the quantization and hence most of the reconstructed values are also nearly zero.

After the IDCT, the correlation of the coefficients is increased and the bit transition is closer to the input of DCT. Ideally, the bit transition should be restored if the quantization does not incur any loss. However, the quantization step size is usually adjusted such that nearly all AC coefficients will be quantized to zero. It is found that about 75% of the  $8 \times 8$  blocks have only DC component and no AC component in the block after quantization. Therefore after IDCT, all 64 elements in those blocks will have the same value which lowers the overall bit transition.

The switching activity at the input of the prediction module is similar to that before DCT, but the breakpoint between LSB and MSB regions is not as sharp.

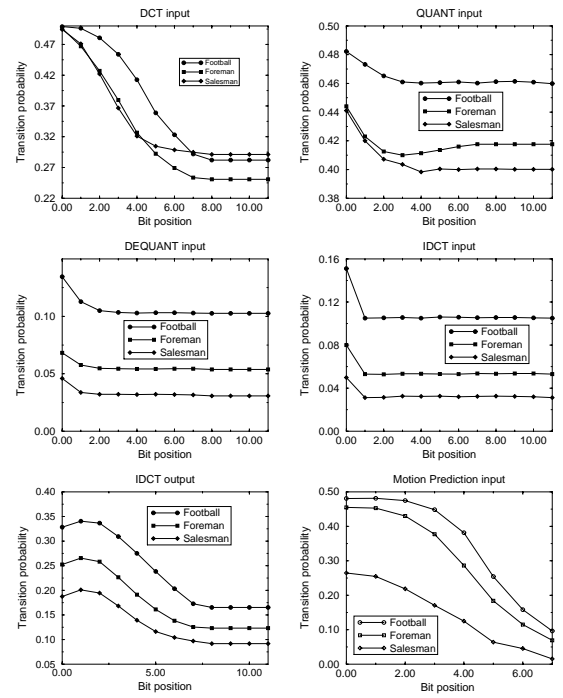


Figure 3: Bit switching activity at the input/output of every module of the H.263 coder

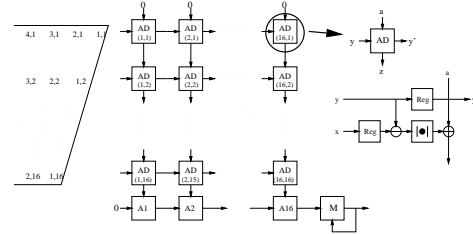


Figure 4: AB2 (16x16) architecture.

## 6. ARCHITECTURAL LEVEL POWER EXPLORATION

Since the motion estimation module consumes most of the computational resources, different motion estimation architectures were evaluated in terms of their power consumption. Full search block matching algorithm (FBMA) which computes the MAD among all the blocks in the search window is used for illustration. Three architectures were compared. They are 2-dimensional systolic mesh-connected array architecture (*AB2*) (Fig 4) and its 1-dimensional version (*AB1*) proposed by T. Komarek et. al. [5], and also an adder tree architecture proposed by Y.S. Jehng et. al. [7] (Fig 5). These three architectures perform the same computation, but use very different dataflows. The area cost, performance and power consumption are compared in the next sub-sections.

### 6.1. Comparison of switching activities

The total switching activities at the output of each processing element of the three architectures are summarized in Table 1. It can be observed that the switching activity of the tree architecture is 40% less than that of the *AB2* and nearly 50% less than that of the *AB1* architecture. Since the *AB1* and *AB2* have a similar structure, *AB2* is used to illustrate the difference in switching activity between the

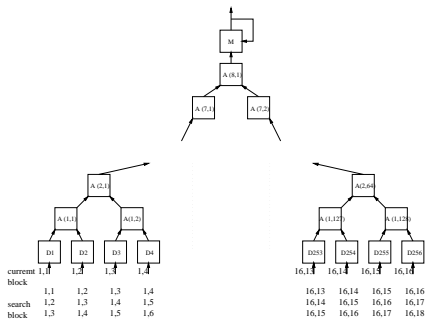


Figure 5: Adder tree architecture.

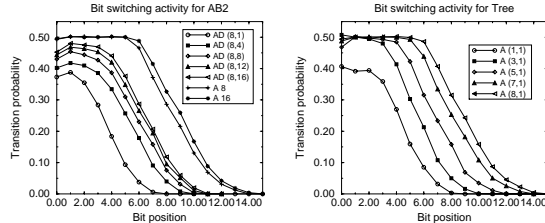


Figure 6: Bit switching activities of AB2 (Left) and tree (Right).

systolic mesh architecture and the tree architecture.

The profiles of the bit switching activity of the tree and AB2 architectures are shown in Figure 6. It is seen that both bit transition profiles follow the DBT model with the breakpoints shifting to the right at deeper levels of addition.

The AB2 architecture can be regarded as 16 columns of *chain* structure which sum all absolute difference values. The same function can be also implemented in a parallel fashion using an adder tree. Comparing the chain with the tree architecture, it can be observed that the chain has 16 levels of addition for one row of macro-block summation, while the tree has only five. This is the main factor contributing to the high switching activity of the chain since switching activity increases with levels. Most of the adders of the tree architecture are at level 1 and hence the total switching activity is lower. Since the chain has more levels of additions, the number of adders having higher switching activities is larger which in turns leads to an overall higher switching activity.

## 6.2. Comparison of Power Consumption using the macro-model equations

Power consumption can be estimated more accurately using the cycle-based power macro-models. The basic processing elements were synthesized using the SIS package and an industrial library. Cycle-based power macro-models were then built using the three video sequences as the training set. To verify the accuracy of the macro-model, we compared the power estimation obtained from the macro-model versus the result obtained from a gate level power simulator for one

Arch.	Total switching/blk. matching
AB2	1,054,740
AB1	1,173,380
Adder tree	627,712

Table 1: Total switching activities of different architectures

Video	Macro. Pow.	Gate Pow.	% error
Football	4.23mW	4.26mW	0.73
Foreman	5.50mW	5.64mW	2.34
Salesman	6.79mW	6.70mW	1.39

Table 2: Comparison between macro-model and gate level power estimation

Arch.	Ave. Pow.	Equiv. gates	Delay/cycle
AB2	70.2mW	155.2K	1426
AB1	109.5mW	9.7K	15376
Tree	32.2mW	220.8K	961

Table 3: Power/Area/Delay trade-off of different motion estimation architectures

column of PEs in the AB2 architecture. Different video sequences were used for the comparison. Table 2 summarizes the result. It demonstrates that the power macro-model is very accurate, less than 3% error compared to gate level simulation.

The average power consumptions of different architectures obtained by using the power macro-models are shown in Table 3. Area (in terms of equivalent gate [7]) and delay (in terms of the number of cycles to generate one motion vector) are also presented. The power consumption is calculated assuming that each architecture has the same throughput. Therefore AB1 is operated in a higher frequency than the other two. It is seen that the power consumption of the tree architecture is about 55% less than that of the AB2 architecture and 70% less than that of AB1. This is because both the switching activity and the operating frequency are lower.

## 7. CONCLUSION

In this paper, we presented a novel hierarchical mixed-level power estimation framework, for designing low power video compression system. Using the framework, the power consumption of the datapaths of different full search block matching architectures were compared. Experimental results show that using cycle-based power macro-model, the accuracy in power estimation is within 3% of that obtained from logic level simulation.

## 8. REFERENCES

- [1] S.Powell and P.Chau. Estimating power dissipation of VLSI signal processing chips: The PFA techniques. *Proceedings of IEEE Workshop on VLSI Signal Processing IV*, IV:250–259, 1990.
- [2] P. Landman J. Rabaey. Power estimation for high level synthesis. *Proceedings The European Conference on Design Automation*, pages 361–6, 1993.
- [3] P. Landman J. Rabaey. Activity-sensitive architectural power analysis for the control path. *Proceedings of IEEE Symposium on Low Power Design*, pages 93–98, 1995.
- [4] Q. Wu et al. Statistical design of macro-models for RT-level power estimation. *Proceedings of ASP-DAC*, 1997.
- [5] T. Komarek P. Pirsch. Array architectures for block matching algorithms. *IEEE Trans. on Circuits and Sys.*, 36(10):1301–8, Oct. 1989.
- [6] K. M. Yang et al. A family of VLSI designs for the motion compensation block-matching algorithm. *IEEE Trans. on Circuits and Sys.*, 36(10):1317–1325, Oct. 1989.
- [7] Y.S. Jehng T.D. Chiueh, L.G. Chen. An efficient and simple VLSI tree architecture for motion estimation algorithms. *IEEE Trans. on Signal Processing*, 41(2):889–899, Feb. 1993.