Algorithm and Architecture Co-Design of Low Power H.264 Baseline Profile Encoder for Mobile Applications

Yu-Han Chen, Tung-Chien Chen, Chuan-Yung Tsai, Sung-Fang Tsai, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Abstract. The concept of algorithm and architecture co-design is presented in this paper for realizing a low-power H.264 encoder. At first, the three-level memory hierarchy of a video coding system was shown for power analysis. The main power sources of a chip are data processing power and memory access power. Power reduction techniques on the algorithm-level and architecture-level should co-operate to minimize power consumption. Hardware-oriented fast algorithms are used to not only reduce data processing and memory access power but also maintain hardware efficiency. Data Reuse (DR) techniques are introduced to further save memory access power. In order to enhance power efficiency, a flexible system architecture is proposed to effectively integrate the fast algorithms and the module-level gated clock technique into the hardware architecture. As a result, the proposed H.264 encoder achieves 14.27 mW power consumption in CIF 30 fps videos. Due to the low-power characteristic, this design is suitable for power-limited mobile applications. Finally, the concept of algorithm and architecture co-design is not limited in the proposed design and beneficial for low-power optimization in other video coding systems.

1 Introduction

While mobile applications grow much faster than expected, there is a strong demand for higher compression ratio and lower power consumption. Due to outstanding coding efficiency, H.264 [1] undoubtedly plays an important role in this area. The useful coding tools [2], such as 1/4-pel resolution, Variable Block-Size (VBS), and Multiple Reference Frames (MRF), improve coding performance but induce huge computation. High computation leads to high power consumption in a hardware design. In terms of power-limited mobile applications, how to develop a low-power H.264 encoder is a challenge nowadays.

"If media coding power dissipation increases beyond a modest 100 mW, it will be hard to implement the media application in portable devices [3]." In [4], an HDTV H.264 encoder with the highly parallel architecture is proposed. The design consumes

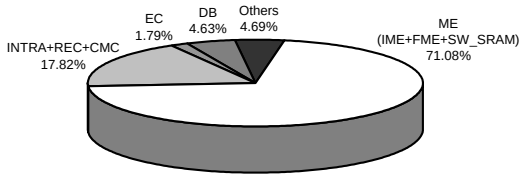

Fig. 1. Power profile of [4] in CIF 30 fps videos. The search range is H:[-32, 31] and V:[-16, 15], and two reference frames are supported. Total power is 172.1 mW

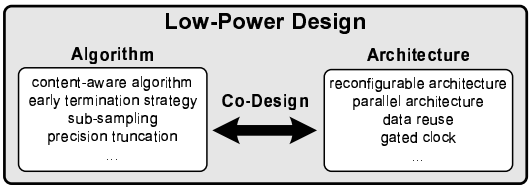

Fig. 2. The power reduction techniques on the algorithm-level and architecture-level

172.1 mW in CIF videos and is not suitable for mobile applications. Therefore, an H.264 encoder consuming less than 100 mW is targeted in this paper. Additionally, it is better to achieve much lower power consumption than 100 mW for extending the battery lifetime of portable devices.

Based on the power profile of [4] in Fig. 1, the main challenge of a low-power H.264 encoder is Motion Estimation (ME). It occupies 71.08% power of the whole encoding system. It is because the powerful coding tools of ME, such as VBS and MRF, are computation-intensive. Computational complexity grows linearly with the number of supported Reference Frames (RF) and block-sizes. Therefore, brute force algorithms are not affordable, and suitable power reduction techniques are required.

Some power reduction techniques are shown in Fig. 2. The techniques on the algorithm-level and architecture-level are not uncorrelated. For example, many fast ME algorithms are proposed in the literature, but most of them are not suitable for hardware implementation. Fast algorithms with

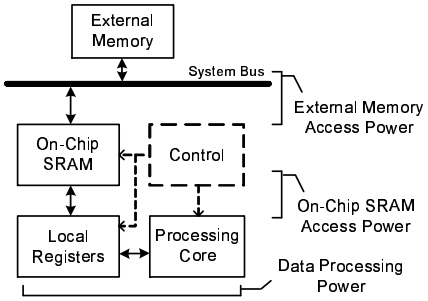

Fig. 3. Hierarchical memory organization of a video coding system

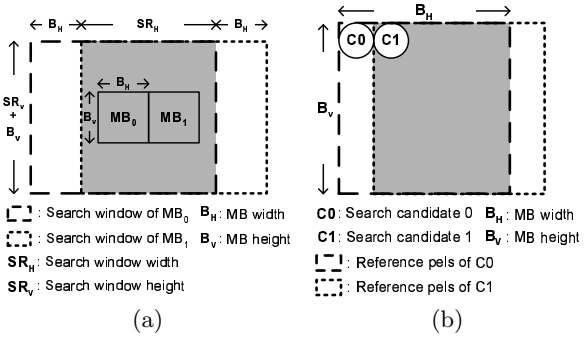

Fig. 4. Data reuse techniques for motion estimation. (a) Inter-MB data reuse; (b) Inter-candidate data reuse. The data in the grey region can be reused

complex controls and irregular search flows cannot be efficiently realized with hardware architectures due to limited hardware flexibility. Therefore, fast algorithms and low-power architectures should be appropriately linked together to minimize power consumption. Fast algorithms should be developed with the consideration of hardware characteristics in the early stage. The low-power architectures should be designed to improve flexibility and realize fast algorithms efficiently. By use of the joint consideration of algorithm and architecture, a low-power H.264 encoder is realized with 14.27 mW power consumption in CIF 30 fps videos. At last, the concept of algorithm and architecture co-design can also be adopted in other video coding systems and beneficial for designs with low-power consideration.

The rest of the paper is organized as follows. Power analysis of a video coding system is discussed in Sec. 2. The techniques of algorithm and architecture co-design on the system-level and module-level are presented in Sec. 3 and Sec. 4, respectively. Section 5 shows the simulation results, and conclusion is given in Sec. 6.

2 Power Analysis

Figure 3 illustrates the three-level memory hierarchy of a video coding system. It contains external memory, on-chip SRAM, and local registers. External memory access, internal memory access, and data processing are the three main sources of power consumption. The power reduction techniques are introduced in the following.

Brute force algorithms are adopted in reference software [5] for most of the coding tools in H.264. Fast algorithms are used to reduce computation and memory BandWidth (BW), and therefore data processing and memory access power is saved. An efficient fast algorithm can reduce computation with graceful coding performance degradation and is the basis of a low-power design. On the other hand, the gated-clock technique is useful to save idle power of the inactive circuits. With fast algorithms and the gated-clock technique, power can be economically spent. Techniques of DR can further save memory access power and will be discussed in Sec. 2.1. Only if the power reduction techniques on all design levels are integrated, will power be minimized. In this paper, algorithm and architecture co-design of a low-power video coding system is focused. If other techniques on the circuit-level, device-level, and the like are applied, power can be further reduced.

2.1 Data Reuse

Memory access power is usually the major power consumption of a chip. In order to reduce memory BW, DS is required. The DS schemes for ME are introduced in the following.

The external memory access power is usually larger than the on-chip SRAM access power. If the processed data of one MacroBlock (MB) can be used by the coming MBs, data will be stored in the on-chip SRAM rather than the external memory. This technique can reduce the external memory access and is defined as inter-MB DR. Figure 4 (a) illustrates inter-MB DR of ME. Data of the overlapped Search Windows (SW) of two successive MBs can be reused to reduce the external BW. On the other hand, some intermediate data are stored in the local registers to reduce the on-chip SRAM access, and it is defined as intra-MB DR. Intra-MB

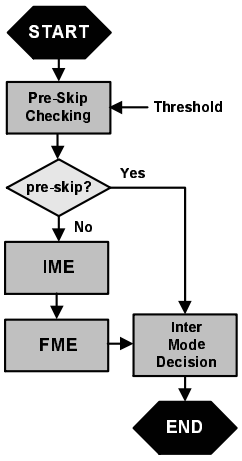

Fig. 5. Proposed ME pre-skip mode decision flow

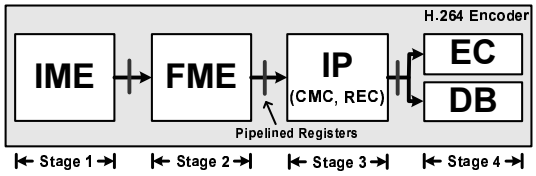

Fig. 6. Conventional 4-stage MB pipelining [4]

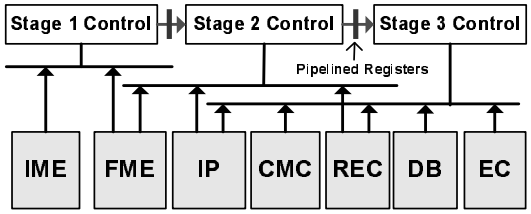

Fig. 7. Proposed flexible MB pipelining

DR can be further categorized as inter-candidate and intra-candidate DR. Reuse of the overlapped reference pels of neighboring search candidates is defined as inter-candidate DR and shown in Fig. 4 (b). Reuse of the reference pels of the same search candidate for VBS ME is defined as intra-candidate DR. Inter- and intra-candidate DR can effectively reduce the BW and power of on-chip SRAM.

3 System-Level Algorithm and Architecture Co-Design

In H.264, most computation is spent in the inter mode decision flow, including the process of Integer ME (IME) and Fractional ME (FME). On the system-level, ME pre-skip mode decision flow is presented to effectively reduce computation. Then, a flexible MB pipelining is proposed to well combine the fast algorithm and the gated-clock technique with the hardware architecture for improving power efficiency.

3.1 ME Pre-Skip Algorithm

In H.264, SKIP mode is defined to enhance the coding gain when the movement of a MB can be well predicted by Motion Vector Predictor (MVP). Only one Motion Vector (MV) is used for the skipped MBs, either $(0, 0)$ or MVP. Possible skip MVs can be pre-checked. According to our simulation results, there are 40%–70% skipped MBs in the low bitrate condition. If the skipped MBs are found in the early stage, much computation of ME can be saved.

The proposed ME pre-skip mode decision flow is illustrated in Fig. 5 (a). "Pre-Skip Checking" is inserted at the start of the mode decision flow. If a MB is pre-skipped, ME engines are turned off, and "Inter Mode Decision" is directly preformed. Otherwise, "IME" and "FME" are sequentially applied as the flow in the reference software [5] to find the best MVs .

3.2 Flexible MB Pipelining

In the conventional MB pipelining architecture in Fig. 6, pipeline controls and pipelined registers are combined with Processing Engines (PE). The intermediate data in a MB pipeline are restricted to be accessed by the PEs at the same MB pipeline stage. That is to say, PEs cannot access the data in different pipeline stages, and hardware flexibility is restricted. Here comes an example. Two possible skip MVs need to be pre-checked in the proposed ME pre-skip mode decision flow. Among them, MVP may be a fractional MV. It means that FME engine is required to be operated in the first pipeline stage. However, it violates the pipeline order in Fig. 6. Besides, one PE cannot be used in different pipeline stages under that architecture. Therefore, the pre-skip checking algorithm is not applicable in the conventional MB pipelining architecture.

To solve this problem, the proposed system architecture in Fig. 7 is divided into two parts—pipeline stage controls and PEs. The stage control handles the MB pipeline schedule and generates control signals to assign tasks to PEs. At the end of each task, the intermediate data are stored back from PEs to

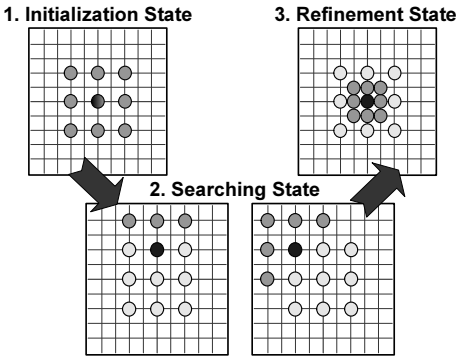

Fig. 8. Flow of four step search algorithm

pipelined registers of the stage control. Then, the stage control can assign a task and transfer the required intermediate data to other PEs. By means of this architecture, one PE is not restricted to be operated only in one pipeline stage, and the ME pre-skip mode decision flow can be supported.

At last, PEs can be clearly separated from each other under the proposed architecture. When the task of a PE is finished, the input clock can be gated immediately with the module-level gated clock technique, and power will not be wasted in the idle state. Therefore, the proposed MB pipelining architecture can improve not only system flexibility but also power efficiency.

4 Module-Level Algorithm and Architecture Co-Design: IME

According to our analysis, a low-power design should be a architecture that supports a fast algorithm with efficient DR. In this section, IME is taken as an example, but the concept can be extended to realize other PEs with low-power consideration.

4.1 Fast VBS IME Algorithm

Parallel VBS Four Step Search In the reference software [5], a sequential flow is adopted for VBS IME. Reference data of different block-sizes are processed independently, and it leads to poor DR. If all the SAD (Sum of Absolute Difference) costs of different block-sizes are computed simultaneously, SAD costs of 4×4 blocks can be reused immediately to generate the costs of larger block-sizes. It is called parallel VBS IME. With this scheme, intra-candidate DR can be achieved.

Four Step Search (FSS) [6] is adopted as the fast IME algorithm in this design. It is because the square search pattern of FSS is similar to Full Search (FS) and beneficial for inter-candidate DR. The flow of FSS is shown in Fig. 8. In the initialization state, FSS starts from an initial candidates such as MVP. In the searching state, the search pattern is moved according to the local minimum of the 16×16 SAD costs. Once the local minimum occurs at the central search candidate, refinement state is started. Eight neighboring search candidates are checked, and then the best matching candidate is found.

There are still other content-aware strategies proposed to reduce computation and maintain coding performance for VBS IME in H.264, but it is not the main consideration in this paper. If the detailed techniques are required, please refer to [7]. Finally, the proposed fast VBS IME algorithm can reduce 98% computation with 0.05 dB quality drop in average compared to FS.

4.2 Low Power Architecture

Three techniques are contained in the low power IME architecture. Configurable SW SRAM is presented to reduce external memory access power. 2-D adder tree architecture is adopted to support inter- and intra-candidate DR. Ladder-shaped reference data arrange is proposed to achieve 2-D random access and enhance inter-candidate DR of FSS.

Configurable SW SRAM For inter-MB DR, four strategies, indexed from level-A to level-D [8], is proposed with different tradeoffs between the size of on-chip SRAM and the external memory BW. Generally, level-C DR is used in a video coding system. However, level-D DR achieves the minimal external BW but uses more on-chip SRAM. In the proposed design, two RFs are supported, and level-C DR is used. But when only one RF is used, the size of SW SRAM is able to support Level-D DR. SW SRAM can configure to level-D DR, and the external memory access power is minimized.

2-D Adder Tree Architecture 2-D adder tree architecture [9] in Fig. 9 (a) is adopted as the basic IME architecture of our design. Reference pels are input row by row into "16×16 Ref-Pel Systolic Array" with inter-candidate DR. For a search candidate, only 16 reference pels in average are newly inserted. Other 16×15 reference

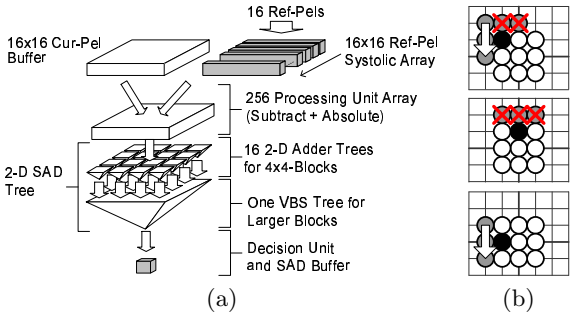

Fig. 9. (a) 2-D adder tree architecture; (b) Data reuse problem of four step search. For simplicity, the step size is set to one

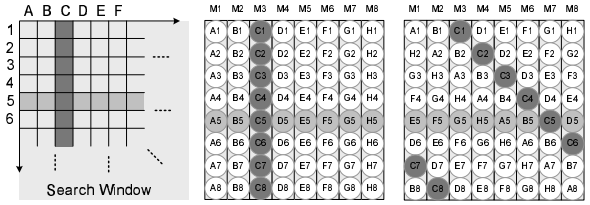

Fig. 10. SRAM data arrange. (a) Physical location of search window; (b) Traditional data arrangement with 1-D random access; (c) Proposed ladder-shaped data arrangement with 2-D random access

pels are shifted downward, stored, and reused. Besides, intra-candidate DR is also supported. "16 2-D Adder Trees for 4×4-Blocks" computes 16 4×4 SAD costs. Then, "One VBS Tree for Larger Blocks" immediately reuses 4×4 SAD costs to generate all other costs of larger block-sizes. Under the 2-D adder tree architecture, reference pels are reused well, and the on-chip SRAM BW is greatly reduced.

Ladder-Shaped Reference Data Arrangement 2-D adder tree architecture is capable of supporting both FS and FSS, but the DR efficiency of FSS is not as good as that of FS. Inter-candidate DR can be achieved only in the vertical direction, and this problem is illustrated in Fig. 9 (b). Because the irregular search path usually leads to poor DR, fast search algorithms cannot reduce on-chip SRAM BW efficiently without good algorithm and architecture co-design.

The problem comes from the restriction in SW SRAM data access. Figure 10 (a) shows the physical location of the reference pels in SW. Traditionally, the horizontally adjacent pels are arranged in different banks of SW SRAM as shown in Fig. 10 (b). The first column of reference pels, $A1$–$A8$, are placed in the bank $M1$. The second column of pels, $B1$–$B8$, are placed in the bank $M2$, and so on. If there are eight banks of SRAM, the ninth column of pels are placed in the first bank $M1$. In this way, a row of reference pels, like $A5$–$H5$, can be read in parallel. However, a column of reference pels, like $C1$–$C8$, cannot be accessed in parallel because they are in the same bank. This is the so-called 1-D random access.

The ladder-shaped SW data arrangement technique is shown in Fig. 10 (c). The second and third rows are rotated rightward by one and two pels. The others are also rotated in the same manner. In this way, the reference pels of $A5$–$H5$ and $C1$–$C8$ are both put in different banks of SRAM. For fast algorithms, the search pattern can move in various directions with good inter-candidate DR. As a result, hardware utilization and power-efficiency of FSS is greatly improved.

Although IME design is focused in this section, the low-power design method can be extended to other modules likes FME, Intra Prediction (IP), DeBlocking (DB), and REConstruction (REC). Hierarchical memory structure in Fig. 3 could be used to analyze the power sources of these engines. Then, the concept of algorithm and architecture co-design can be adopted to implement the low-power designs.

5 Simulation Results

The proposed techniques have been implemented as an H.264 encoder by TSMC 0.18 μm CMOS technology with 452.8k logic gates and 17 kbytes on-chip SRAM. Real-time processing of CIF 30 fps videos is achieved. There are two modes supported—high quality mode and low-power mode. In high quality mode, two RFs are supported, and the pre-skip function is turned off. In low-power mode, only one RF is used, and 50% of MBs are pre-skipped. Table 1 lists the power info. The proposed H.264 encoder is synthesized at 27 MHz. In the low-power mode, voltage scaling can be applied to further reduce power because only 13.5 MHz operating frequency is required. By the way, about 16 mW and 8 mW idle power can be saved with the module-level gated clock technique in high quality mode and low-power mode, respectively. The R-D performance and power comparison data are shown in Fig. 11 and Fig. 12. Only

Table 1. Power information of the H.264 encoder

Mode	High Quality	Low-Power
No. of reference frames	2	1
Frequency (MHz)	27	13.5
Voltage (V)	1.8	1.3
Power (mW)	88.91	14.27

Power data are reported form Prime Power

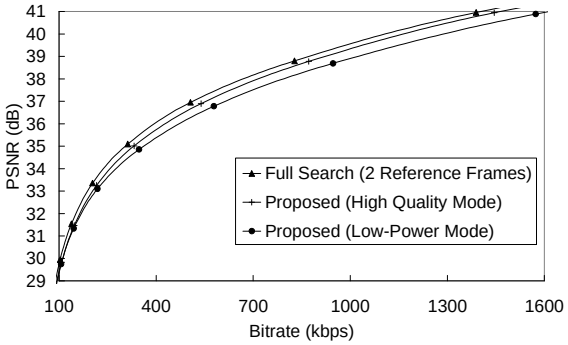

Fig. 11. R-D performance of the H.264 encoder

8% power is required in the low-power mode of our design compared to [4].

In our design, video services with better visual quality can be supported in the high quality mode with higher power consumption. When the battery is running out, low-power mode is supported to extend the battery lifetime. The characteristic of power adaptability is useful for portable devices.

6 Conclusion

Algorithm and architecture co-design for H.264 baseline profile encoder is presented in this paper. Fast algorithms, DR, and the gated clock technique are used to minimize data processing and memory access power. Flexible architectures are developed to realize an power-efficient hardware design. The proposed H.264 encoder achieves 14.27 mW power consumption in CIF 30 fps videos and is suitable for mobile applications. At last, the low-power design concept can also be adopted in other video coding systems for low-power optimization.

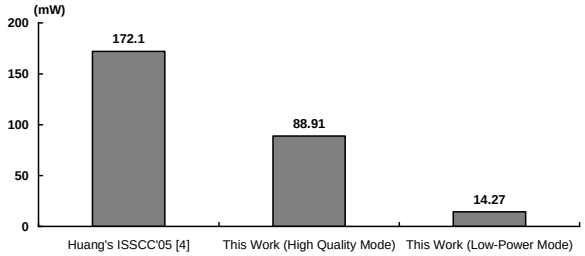

Fig. 12. Power comparison of H.264 encoders in CIF 30 fps videos

References

1. Joint Video Team of ITU-T and ISO/IEC JTC 1: Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification. (2003)
2. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. CSVT **13** (2003) 560–576
3. Etoh, M., Yoshimura, T.: Advances in wireless video delivery. Proceedings of the IEEE **93** (2005) 111–122
4. Huang, Y.W. *et al.*: A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications. Proc. IEEE ISSCC (2005)
5. http://bs.hhi.de/∼suehring/tml/download/: H.264/AVC reference sofware JM8.2. (2004)
6. Po, L.M., Ma, W.C.: A novel four-step search algorithm for fast block motion estimation. IEEE Trans. CSVT **6** (1996) 313–317
7. Chen, Y.H., Chen, T.C., Chen, L.G.: Hardware oriented content-adaptive fast algorithm for variable block-size integer motion estimation in H.264. Proc. IEEE ISPACS (2005)
8. Tuan, J.C., Chang, T.S., Jen, C.W.: On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture. IEEE Trans. CSVT **12** (2002) 61–72
9. Chen, C.Y. *et al.*: Analysis and architecture design of variable block size motion estimation for H.264/AVC. (Accepted by IEEE Trans. CASI)