LOW POWER CACHE ALGORITHM AND ARCHITECTURE DESIGN FOR FAST MOTION ESTIMATION IN H.264/AVC ENCODER SYSTEM

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

DSP/IC Design Lab, Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan; Email: cytsai@video.ee.ntu.edu.tw

ABSTRACT

Low power Motion Estimation (ME) of H.264/AVC is an important research issue because of the growing mobile applications of H.264/AVC encoder. In this paper, low power cache algorithm and architecture for fast ME of H.264/AVC is proposed in order to replace the conventional Search Range (SR) memory. With the Block Translation (BT) cache architecture, Search Trajectory Prediction (STP) prefetching algorithm, and ultra low power Cache Miss Hiding (CMH) strategy, 35% SR memory writing power and 67% SR memory static power are reduced for D1 videos. Combining fast ME with the proposed cache provides the total solution for low power ME hardware.

Index Terms— Low power, cache, ME, H.264/AVC

1. INTRODUCTION

H.264/AVC is the next generation video coding standard developed by the Joint Video Team. It can save 25–45% bit-rate compared with MPEG-4 Advanced Simple Profile (ASP). Its ultra high coding efficiency comes from lots of new features, and one of the most gainful features is the Motion Estimation (ME). The H.264/AVC ME supports variable block sizes and multiple reference frames, which greatly improve the prediction performance, but the hardware cost and power are drastically increased too. Because of the growing mobile applications of H.264/AVC, the solution of low power ME hardware in H.264/AVC encoder system is an important research issue.

However, most of the previous works only focus on minimizing the ME logic power or reusing the Search Range (SR) memory data for memory reading power reduction. Although the ME logic power and SR memory reading power of existing designs are very low, the power from the writing operation and even leakage current (static power) of SR memory should also be minimized. In this paper, the concept of cache design is adopted into ME hardware to replace the conventional SR memory. Based on the proposed algorithm and architecture, the cache achieves very considerable power reduction.

The rest of this paper is as follows. Section 2 gives the SR memory problem and cache design challenge. The proposed algorithm and architecture are given in Sec. 3 and 4. Finally, Sec. 5 and 6 are the simulation results and conclusion.

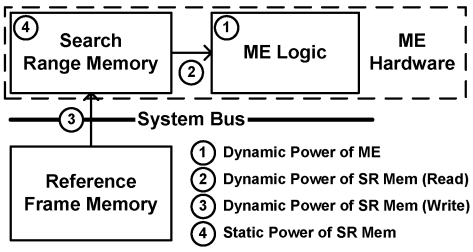


Fig. 1. Power dissipation sources of general ME hardware.

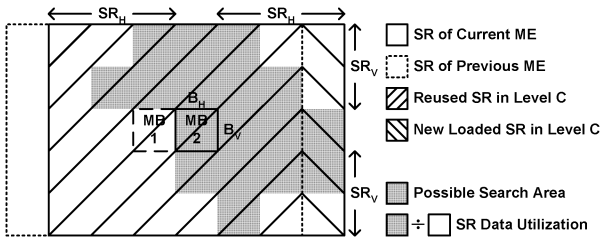


Fig. 2. Level C data reuse & definition of SR data utilization.

2. PROBLEM STATEMENT

Figure 1 shows the power dissipation sources to be considered when designing a low power ME hardware. The ME logic and SR memory reading power can be solved by applying fast ME algorithms. For example, both diamond search and four-step search [1–4] can reduce the search points compared with full search, and thus minimize these two dynamic power sources. However, as the specification of video encoder keeps growing with new standard like H.264/AVC, the less discussed SR memory writing power and static power also become critical.

2.1. Problems of Conventional SR Memory

The SR memory writing power is related to the capability of SR data reuse techniques. Level C data reuse [5] is a conventional technique that reuses the overlapped SR region between ME operations. Using a $(2SR_H+B_H)\times(2SR_V+B_V)$ sized SR memory, a region of $2SR_H\times(2SR_V+B_V)$ pixels can be reused every time, as depicted in Fig. 2. However we discov-

ered that Level C data reuse is very inefficient when combined with fast ME. Since fast ME largely reduces search points in the SR, some data written into the SR memory may never be used. From the system perspective, it is even worse since the SR memory writing power includes the external memory access power, which can be ten times larger than on-chip memory access. Moreover, this low data utilization also indicates a serious waste in the hardware area of SR memory, which is proportional to the static power. The static power is not a less important issue especially in the deep sub micron fabrication technology [6].

The SR data utilization rate simulated based on our previous work of H.264/AVC fast ME [2] is less than 30% for CIF videos with (SR_H, SR_V)=$(32, 16)$, and even less than 15% for D1 videos with (SR_H, SR_V)=$(64, 32)$. Apparently, as the video frame and SR size keep growing with the new standards, the SR memory writing power and static power become even more critical problems.

2.2. Challenges of Cache Implementation

Replacing the Level C SR memory with the most basic cache can alleviate the above two problems. It can reduce redundant SR memory writing and enable the possibility of size-reduced SR memory since only data requested by ME logic are loaded. However a cache without an efficient prefetching mechanism will seriously decrease the overall processing speed.

Analogous prefetching of Level C SR memory is simple because only a regular $B_H \times (2SR_V + B_V)$ rectangle needs to be loaded, as shown in Fig. 2. But prefetching of a real cache is difficult to implement, based upon conventional analysis of fast ME's irregular memory access trace. Therefore, to derive a low power cache algorithm with prefetching mechanism is the design challenge of this work.

3. PROPOSED ALGORITHM

In this paper, we proposed a low power cache algorithm for H.264/AVC fast ME based on our previous works [2, 7]. The previous hardware-oriented content-adaptive four-step search and data reuse strategy had successfully minimized the power of ME logic and SR memory reading. With the cache strategy, the SR memory writing power and static power are therefore minimized too. In addition, an efficient prefetching algorithm is successfully proposed, using the motion prediction and fast search pattern characteristic of our fast ME. In the following subsections, the basic cache strategies, prefetching algorithm, and ultra low power cache strategy will be introduced.

3.1. Basic Cache Strategies

The proposed low power cache is called Block Translation (BT) cache. We adopted the concept of Translation Lookaside Buffer (TLB) into the cache design. A two-dimensional cache

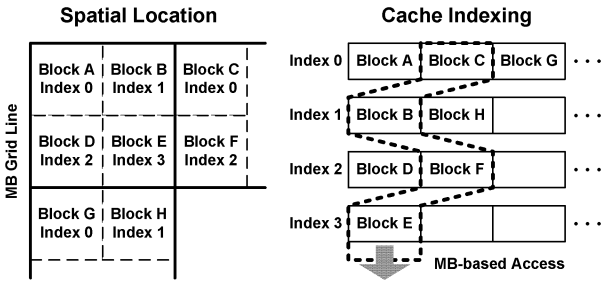


Fig. 3. Cache indexing strategy for 8×8 blocks.

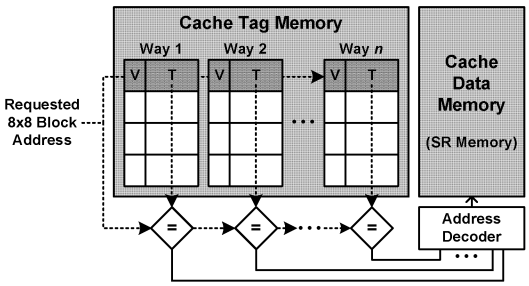


Fig. 4. BT cache diagram (V is valid bit and T is tag).

block of SR pixels share one tag entry on translating the data address to reduce the power introduced by tag memory.

Granularity of the cache block is the first parameter to be determined. Cache with smaller block size has better SR data utilization because it can fit the possible searching area finer, but more tag entries are needed. After a series of simulations on cache block sizes, we found 8×8 block can strike the best balance among 2×2, 4×4, 8×8, and 16×16 blocks.

The proposed BT cache is an *n*-way set associative cache, where *n* corresponds to the capacity of cache in the unit of MB (16×16 SR pixels). To facilitate the reuse of our low power SR memory with MB-based data access, each 8×8 block is indexed according to its relative position in the MB. As shown in Fig. 3, combinations of block *ABDE*, *BCEF*, and *DEGH* can be MB-based accessed in parallel. The BT cache diagram is shown in Fig. 4. Every time the cache index is selected, the tags of all *n* cache ways are compared with requested address of 8×8 block simultaneously. The address decoder then generates the physical data address based on the hit way number.

The concept of virtual addressing is also applied in storing the cache tags. Each 8×8 block's address in the tag memory represents its relative coordinates (*x* and *y*, horizontal and vertical) in current SR. Each time when a new MB's ME process is started, every tag entry's *x* coordinate will be decreased by one. And the valid bits of tag entries with negative decrement results will be set low, because they correspond to 8×8 blocks located outside the SR. Compared with absolute addressing, the virtual addressing strategy not only requires less-bitwidth tags, but also helps to flush unneeded tags easily.

The replacement strategy of BT cache is also intuitive and

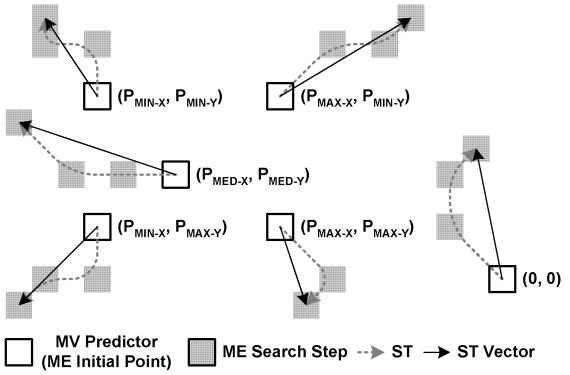


Fig. 5. Definition of ST vector. P is set of neighboring MVs. Origin, median, and min/max of P are MV predictors.

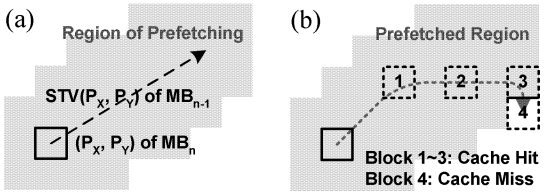


Fig. 6. (a) STP prefetching flow. (b) Scenario of cache miss.

efficient. Due to the raster-scan coding order of H.264/AVC, 8×8 block with the smallest x coordinate is the least probable one to be used in the future, and thus the first to be replaced.

3.2. Prefetching Algorithm

In this paper, the Search Trajectory Prediction (STP) prefetching algorithm for the BT cache is proposed. Search Trajectory (ST) represents the search pattern (steps) of fast ME like four-step search. Although the ST can fully describe the behavior of fast ME, yet it takes lots of parameters to record this curve. Definition of the ST vector is depicted in Fig. 5, which is used to approximate the curvy ST. Through a series of simulations, we found the ST vectors of two neighboring MBs are actually very similar. Average length of the vector difference is merely two-pixel long or even shorter. As a result, the STP prefetching algorithm can guide current ME operation to prefetch 8×8 SR blocks around the MV predictor and ST vector, as shown in Fig. 6(a). Current ST of MV predictor (P_X, P_Y) is directly predicted by previous MB's ST vector of the corresponding (P_X, P_Y). The 8×8-block-based prefetching region is coarse but advantageously it can provide extra space for the possible ST deviation to be still covered.

3.3. Ultra Low Power Cache Strategy

In this paper, an ultra low power cache strategy called Cache Miss Hiding (CMH) is proposed. Imaginably, cache miss may still happen even with the best prefetching algorithm. In the cache of general processors, any cache miss must be resolved by requesting and waiting for the data in main memory, which is an inevitable power-consuming task. However, for our fast ME, the searching process can be considered as refinement of the initial guess. In other words, the ME process can be simply terminated to hide the cache miss penalty. For example, as shown in Fig 6(b), when block 4 suffers cache miss, the ME can be terminated and the best result is chosen among block $1\sim3$. According to our simulation results in Sec. 5, the CMH can further reduce the power with little quality drop.

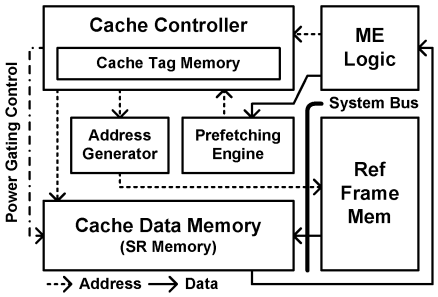


Fig. 7. Architecture of proposed cache hardware.

4. ARCHITECTURE

Figure 7 shows the architecture of proposed low power cache with two data paths—the prefetching and ME data path. In the prefetching data path, the prefetching engine starts the operation of requesting 8×8 blocks. The cache controller compares requested addresses with cache tags to check whether the data are in cache or not. Only for cache misses will the cache data memory be updated through accessing the external reference frame memory. In the ME data path, the ME logic also starts similar operation, except the cache miss does not invoke reference frame memory access if CMH is enabled. Meanwhile, STV is also calculated and stored into the prefetching engine for next MB's STP prefetching. Worthy to note, according to the frame size or power limitation, cache controller can adjust the cache way number and gate the unused memory's power.

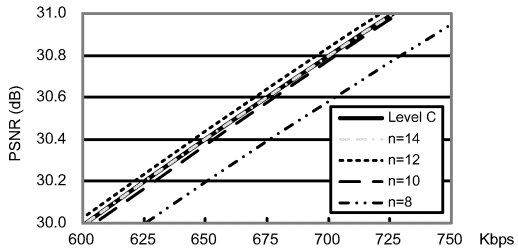
5. SIMULATION RESULTS

Simulations of the proposed low power cache is based on our previous work of content-adaptive four-step search [2], which is modified from the H.264/AVC reference software—JM. In order to evaluate the power of cache, the cache writing bandwidth under different cache sizes is simulated. The results are listed in Table 1, and the unit is MB number per current MB. Because the Level C bandwidth for CIF and D1 is 3 and 5, the average bandwidth reductions gained from the CMH-enabled cache are 28% and 37% respectively. Compared with CMH-disabled cache, CMH-enabled cache can save at least 5% additional bandwidth, which is very useful since it also includes

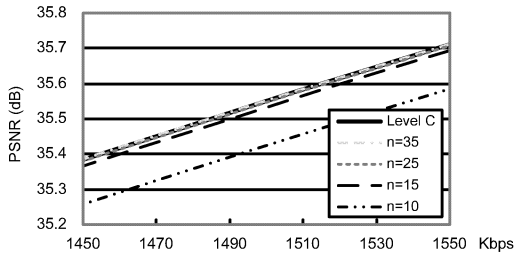
Table 1. Simulation Results of Cache Performance and CMH Gain

	n	Foreman	Mobile Calendar	Stefan	Table Tennis	*n*	Football	Wendy
CMH Disabled Bandwidth (MB/MB) / Cache Hit Rate	14	2.36 / 99.7%	2.13 / 99.9%	2.25 / 99.7%	2.19 / 99.7%	35	3.39 / 99.5%	3.49 / 99.4%
	12	2.36 / 99.7%	2.13 / 99.9%	2.26 / 99.7%	2.19 / 99.7%	25	3.47 / 99.5%	3.55 / 99.4%
	10	2.39 / 99.7%	2.14 / 99.9%	2.29 / 99.7%	2.21 / 99.7%	15	3.61 / 99.4%	3.74 / 99.4%
	8	2.43 / 99.7%	2.16 / 99.8%	2.29 / 99.7%	2.24 / 99.7%	10	3.48 / 99.5%	3.53 / 99.5%
CMH Enabled Bandwidth (MB/MB) / Quality Drop (dB)	14	2.28 / 0.00	2.09 / 0.00	2.17 / 0.00	2.12 / 0.00	35	3.11 / 0.00	3.19 / 0.00
	12	2.28 / 0.01	2.09 / 0.00	2.17 / 0.00	2.13 / 0.00	25	3.14 / 0.01	3.22 / 0.01
	10	2.30 / 0.06	2.10 / 0.00	2.18 / 0.05	2.14 / 0.01	15	3.20 / 0.02	3.31 / 0.03
	8	2.32 / 0.09	2.11 / 0.00	2.19 / 0.10	2.15 / 0.02	10	3.01 / 0.07	3.10 / 0.09

Bandwidth means cache writing data rate. *n* denotes cache size; unit is MB. $CIF(SR_H, SR_V)$=(32, 16). $D1(SR_H, SR_V)$=(64, 32).



(a) Stefan (CIF, *n* for Level C is 15)



(b) Football (D1, *n* for Level C is 45)

Fig. 8. Rate-distortion curve of different cache size.

the external bandwidth reduction. Meanwhile the quality drop introduced by the CMH less than 0.1dB for all cache sizes.

Besides the cache writing power and size (static power), the coding quality is also an important term to be considered in the trade-off. The rate-distortion curves of different cache sizes are shown in Fig. 8. According to the simulation results, we decided to set the cache size to 10 MBs for CIF video, and 15 MBs for D1 video. In this setting, the proposed low power cache has less than 0.02dB quality drop compared with Level C SR data reuse. The resultant SR memory writing power and memory size (static power) reduction ratios are 27% and 33% for CIF video, and 35% and 67% for D1 video. The hardware complexity of cache controller and prefetching engine is very low such that the power reduction shall be nearly unaffected.

6. CONCLUSION

In this paper, the low power cache algorithm and architecture for H.264/AVC fast ME are proposed. The BT cache architecture in cooperation with STP prefetching algorithm and CMH ultra low power strategy significantly reduces the writing and static power of SR memory. As compared with Level C data reuse scheme, the writing power and static power are reduced by 27% and 33% for CIF video, and 35% and 67% for D1 video. Combining previous fast ME algorithm with the proposed low power cache can provide the total solution for low power ME hardware in the H.264/AVC encoder system.

7. REFERENCES

[1] L.M. Po and W.C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–7, 1996.

[2] Y.H. Chen, T.C. Chen, and L.G. Chen, "Hardware oriented content-adaptive fast algorithm for variable block-size integer motion estimation in H.264," *Proc. Int. Symp. on Intell. Signal Processing and Commun. Syst. (ISPACS)*, pp. 341–4, 2005.

[3] J.Y. Tham, S. Ranganath, M. Ranganath, and A.A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–77, 1998.

[4] A.M. Tourapis, O.C. Au, and M.L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 934–47, 2002.

[5] J.C. Tuan, T.S. Chang, and C.W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, 2002.

[6] W.M. Elgharbawy and M.A. Bayoumi, "Leakage sources and possible solutions in nanometer CMOS technologies," *IEEE Circuits Syst. Mag.*, vol. 5, no. 4, pp. 6–17, 2005.

[7] T.C. Chen, Y.H. Chen, S.F. Tsai, and L.G. Chen, "Architecture design of low power integer motion estimation for H.264/AVC," *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Processing (ICASSP)*, vol. 3, pp. 900–3, 2006.