FPGA Implementation of Dynamic Energy Efficient Memory Controller for a H.264/AVC Application

A.M. Kulkarni
SENSE, VIT University
Vellore, India

V. Arunachalam
SENSE,VIT University
Vellore, India

ABSTRACT
Improvement in high speed DSP applications can be done by integrating computational power with effective memory management. Bandwidth and latency of operation in memory system is rigidly dependent on data accesses. DSP applications such as multimedia require exhaustive streaming at high speed buses. The energy consumption is the key element which will be the focus of research in VLSI and Embedded systems industry. Today a cardinal issue of DSP application is to reduce impact of memory access on execution time while reducing energy consumption of the system. Memory Scheduling is significant in DSP applications to use memory bandwidth effectively. In this paper, we introduce the dynamic memory access scheduling with refresh priority considerations. In addition, a novel bus switching activity monitoring mechanism is implemented to efficaciously reduce the energy consumption of memory operations. H.264/AVC provides higher coding efficiency through added features and functionality, which impose additional computational complexity in encoder and decoder. The features of memory access patterns of H.264 encoder are analyzed. The overhead cycle of page activation has been reduced to improve bus efficiency which also reduces latency of operations. The scheduler and memory controller has been experimented by running a dynamic H.264/AVC application on Xilinx FPGA.

General Terms
Design

Keywords
Multimedia applications, Dynamic Memory schedulers, Bus switching activity monitoring, H.264/AVC.

1. INTRODUCTION
With lots of functionality being merged, the need of high performance in an embedded system becomes inevitable. This range of new application also demands low energy consumption. Memory chips pervade a greater portion of energy consumption in embedded systems. Also, DSP processing complexity and speed of operation have increased dramatically in the past decade in comparison to memory control techniques. Therefore, their performance is limited by the speed of the memory control systems. The applications like image and multimedia processing are characterized by a large number of data accesses. These streaming intensive applications limit the computational speed. In this paper, we have considered the H.264/AVC encoder which can achieve bit-rate reduction by a factor of 2 at the cost of computational complexity. In fact, new researchers are concentrating on reducing computational latency by scheduling memory accesses.

3-D architecture of DRAMs makes it sequentially accessible rather than randomly [8]. Latency and bandwidth of memory systems are cardinally dependent upon 3-D architecture of memory architecture. Sequential accesses to rows have high access latency and cannot be pipelined due to row pre-charge and row activation operations [6].

Accesses to different banks and columns for single row have low latency. This operation can be pipelined. However this makes system performance dependent on the number of accesses. The 3-D structure of DRAMs is advantageous for reordering memory accesses.

A memory access scheduling with reordering memory references improves performance by exploiting locality within the 3-D memory structure [8]. Memory scheduling operation based on the history of recently scheduled operations has been adapted in [3]. History based scheduler takes advantage of previously scheduled instructions thereby avoiding certain bottlenecks within the memory controller. In [4], non-uniform latencies have been utilized effectively to increase bus utilization while decreasing execution time of memory. Burst scheduling improves execution time by reducing the need of row pre-charge. It accesses are directed to same row of same bank. However [1], considers dynamic memory accesses too which are handled in pipelined manner.

There have been several forays into hardware and software energy minimization techniques. From the hardware aspect, we find two complementary energy saving trends emerging. The first is the clustering of hardware components into smaller and less energy consuming components. Zyuben and Kogge [10] show that such a multi-clustered architecture can be up to twice as energy efficient as wide-issue superscalar processors. The second trend is the support for different operating modes (power modes/energy modes), each consuming a different amount of energy.

This paper experiments Dynamic Memory Access Scheduler for H.264/AVC based HDTV multimedia application. Latency of memory accesses is further reduced by efficient address generation policy. Video coding is typically a data dominated process and affects the memory bandwidth. The paper [5], quantifies the complexity cost in memory centric way. Bit-allocation approach is used to enhance memory access speed by simplifying the computational complexity [9]. It uses pseudo address decoding to shrink I/O complexity, which helps to shorten the access time. It further takes advantage of bus switching activity monitoring mechanism to reduce energy consumption of memory.

This paper makes following contributions:

- Analyses the memory access patterns of H.264 encoder and designs address generator for further reduction in latency of operations.
- Introduces Dynamic Memory Access Scheduling considering refresh priority.

- Implements Bus switching activity monitoring mechanism to effectively reduce energy consumption of memory.
- Compares experimental results of all memory schedulers with proposed memory scheduler.

Definition1 (Static access sequences): The static memory access sequences are those which are known a priori before execution of the application. Also, we can say that it does not depend on execution practicalities.
Definition2 (Dynamic access sequences): The memory access is called indeterminate access when a part of the data is unknown before execution of the application. Therefore, indeterminate accesses are a combination of static and dynamic access sequences.

Dynamic accesses are computed while executing the application. Section-2 presents related work background for memory scheduling algorithms. Section-3 adduces address generator based on H.264/AVC encoder memory access pattern analysis. The proposed architecture of memory scheduler which can handle dynamic memory accesses is discussed in Section-4. Experiments & results analysis are presented in section-5. It shows the comparable reduction in execution time with a little increase in hardware complexity.

2. RELATED WORK

2.1 First Ready Scheduling

First Ready Scheduling is a non-preemptive scheduling. In this scheduler tasks in the ready queue are executed in the order in which they entered the queue. The next will commence only after the completion of previous task.

Task can be defined in three states: (i) ready, (ii) blocked (iii) running state. When task is moved from one state to another it is called context switching. In this scheduling two queues are maintained in blocked and ready state. Task with ready will be have highest priority and will perform the task on memory. The first ready scheduler considers the pending task; scheduler timing and resource constraints while context switching. Advantage is taken of the fact that when a row is being pre-charged other bank operations can be context switched.

2.2 Out-of-order Scheduling

Memory access scheduler is responsible for ordering & duration of task. Reordering of task is one of the important tasks of the memory scheduler available today. During the last decade many researchers concentrated on memory scheduling policies to reduce latency of operation and execution time.

Due to 3-D structure of DRAM devices, we can access memory rows sequentially with pipelined banks available. An access to DRAM is consisting of three commands: (i) Pre-charge, (ii) Activate row access, (iii) Activate column access. The goal of out-of-order scheduling is to reduce latency by reordering memory references. In [8], a better command resource utilization which reduces DRAM cycles from 28 to 16 has been illustrated. An example is shown in figure.1

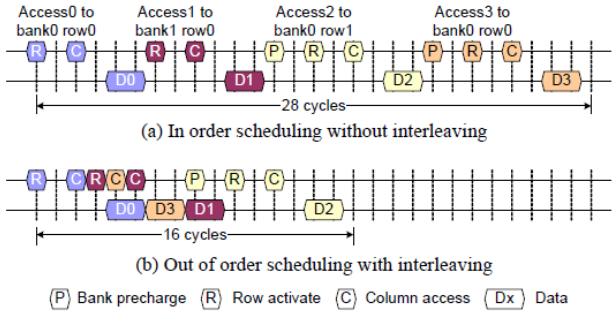

Figure.1.Memory Accesses Scheduling

Depending upon the state of the DRAM, memory accesses could be row hit, row conflict, and row empty & has different latencies. A row hit occurs when the bank is open and an access is directed to the same row. A row conflict occurs when an access goes to a different row than the last access to the same bank. If the bank is closed (pre-charged) then a row empty occurs.

2.3 Burst Scheduling

The actual meaning of burst length refers to the amount of data that is written/read after issuing write/read command. But here, burst means accesses to same row of same bank.

Figure.2 frames structure for burst scheduling. Since latency of row hit is small, row hit rate increases which maximizes bus utilization rate. Accesses to the same bank are stored in write queues and read queues according to the request from the processor to the specific bank. Newly arrived burst are joined with these queues. After a bus is arbitrated from each bank final access is determined from ongoing accesses stored in read and write queues.

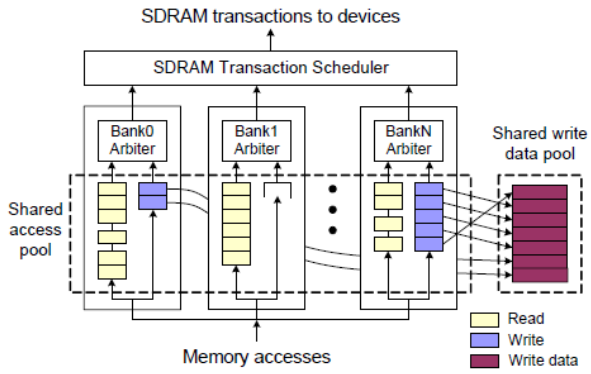

Figure.2 Structure of Burst scheduling

3. OPTIMIZATION FOR ADDRESS GENERATION

The page break analysis of memory access pattern is done on the basis of current frame storing, motion estimation and de-blocking loop filter, to find the lowest speed of memory controller for HDTV(1920 × 1080@30fps) encoder[2].

Total memory bandwidth required for HDTV 1080p (1920X 1080, 30fps) real time video application can be calculated as:

$$BW(total) = BW(store) + BW(Loop) + BW(ME\_luma) + BW(ME\_croma)$$

$$BW(total) = 607Mbps$$

Considering 64-bit data bus:

$$F(memory) = BW(total) \times (8/64)$$

$$F(memory) = 75.9MHz$$

Therefore, SDRAM controller should be processing at around 80MHz to reach the bandwidth requirement for HDTV encoder. Unfortunately, SDRAM bandwidth cannot be used with 100% efficiency. Hence, we need to increase the frequency by analyzing the access strategy.

We have used the following algorithm for address generation [2]:

- Divide each row of incoming pixel data into multiples of 16 based on the number of macro blocks that can fit in a row of SDRAM memory.
- For a memory with 512 columns and data width of 16, two macro blocks can fit in single row of SDRAM.
- The current macro block for encoding is obtained from single row.
- The address for current macro block can be obtained by taking the macro block number.
- For encoding one macro block it has to access (2Ph + N-1) X (2Pv + N -1) pixel data. Where, Ph is the horizontal search range, Pv is the vertical search range and N is the macro block size.
- The amount of memory access is reduced by using level C data reuse strategy. By using this data reuse strategy only (2Pv/N + 1) reference macro blocks will accessed from search area.
- In the encoding sequence of IBBP the two consecutive B-frames require the same reference data for encoding. The order of accessing current macro blocks from two consecutive B-frames is changed in order to reduce the amount of search data access time.
- The current macro block of two B-frames that needs same reference data for motion estimation are accessed alternately.

4. DYNAMIC MEMORY ACCESS SCHEDULER

Based on the above two architectures this paper proposes a new dynamic memory access scheduling algorithm and elaborates the approach. Figure.3 shows the architecture for proposed scheduler architecture. In this read and write queues are maintained and memory accesses depending on read/write condition sent to particular queue. While maintaining the queue read/write memory accesses are sorted according to arrival time. Newly arrived accesses join queues. Bank arbiter is responsible for sending final burst to the DRAM memory. While scheduling memory accesses proposed scheduler places emphasis on dynamic memory accesses.

4.1 Dynamic Memory Accesses

Dynamic memory accesses as defined in section II consist of static and dynamic accesses. Here, we considered dynamic accesses as Read-After-Write(R-A-W) and Write-After-Write accesses (W-A-W). Since, RISC processor adopts plurality of pipelines, these dynamic memory accesses sometimes turn into data hazards [7]. A major task to scheduler is to reduce dynamic memory access.

Dynamic memory accesses can be of three types: Consider there are two memory accesses *i* and *j*, where *i* is occurring before *j*.

Definition3: Read-After-Write (RAW): j tries to read before *i* writes so incorrectly it gets old data.

Defination4: Write-After-Write (WAW): j tries to write data before it is written by *i*. The write end-up being performed in wrong order. This leaves the value written by *i* instead of *j* in destination.
Definition5: Write-After-Read (WAR): j tries to write a destination before it is read by *i*, so *i* incorrectly gets the new value.

In the proposed architecture we have reduced RAW and WAW accesses, assuming that, packets are formed by data and address. Whenever read packets arrives at read it first searches write queue for a packet with same physical address. If found the read packet will not enter in read queue instead it enters in write queue. Otherwise, read packet enters in read queue to end with an access. This task avoids RAW data hazard. WAW is avoided by queue, since accesses to memory are done in time sequence. RAW cannot happen in our scenario and explained in section 3.2

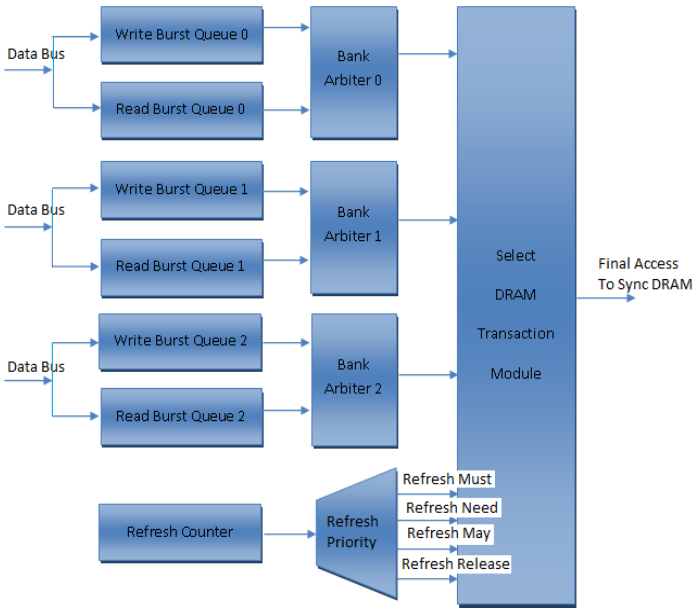

Fig.3 Dynamic Memory Access Scheduler

4.2 Scheduling Policy

Scheduling queue assigns a priority and final data sent to memory are based on different priorities we have considered based on facts observed [8], [4], [1].

Priority is based on waiting cycle of packet in queue (W_T), burst length (BL), and read/write priority (P).x and y are coefficients respectively.

$$Priority = x.\ W_T + y.\ BL + P$$

Memory scheduler also considers a fact which helps to decrease latency. Among all pending reads/ writes scheduler will pick read/write to rows already open. Then priority mentioned above is considered. Now, final read/write access can be performed. If read queue is not full read operation is performed, else write operation is performed first. Dynamic memory access scheduler bids to delay refresh operation as long as possible to maximize performance while meeting refresh need. Decision of priority is based upon following observations:

- Longer the packet is waiting in queue higher priority it should have. Or else data dependency will be lost.
- As defined if burst length is higher row hit accesses will

increase as well as it will reduce transition time of many shorter bursts.

Priority of read accesses must be higher than write accesses. This avoids the W-A-R memory accesses. W-A-R occurs due to write packets sent early before read packets. Also, read accesses need to fetch data and sooner they send data which require further operation. Whereas, write accesses considered finished after writing in queue.

4.3 Refresh Policy

Refresh command is one of the most important tasks in DRAMs. If DRAMs are not refreshed in time it creates data corruption. Many research papers illustrates pre-charging bank [8], row activate and column accesses [4]. In this paper we have considered refresh policy as well.

Memory scheduler assigns refresh command at refresh rate. Refresh interval counter is loaded with refresh rate at each clock cycle it is decremented. Whenever, refresh interval counter reaches zero value backlog counter is incremented by 1. Alternatively, whenever refresh command is assigned backlog counter is decremented by 1. Backlog counter holds number of refresh command. Memory scheduler assigns refresh command depending on the urgency level. After following refresh command memory scheduler waits for T_{RFC} mentioned in datasheet of memory device. Refresh urgency levels are shown in Table1.

Table. 1 Refresh Urgency Levels

URGENCY LEVEL	DESCRIPTION
REFRESH MAY	Backlog count is greater than 0. Indicates there is a backlog of REFR commands, when the memory scheduler is not busy it will issue the REFR command.
REFRESH RELEASE	Backlog count is greater than 3. Indicates the level at which enough REFR commands have been performed and the memory scheduler may service new memory access requests.
REFRESH NEED	Backlog count is greater than 7. Indicates the memory scheduler should raise the priority level of a REFR command above servicing a new memory access.
REFRESH MUST	Backlog count is greater than 11. Indicates the level at which the memory scheduler should perform REFR command before servicing new memory access requests.

4.4 Selection of Final Burst

The last step is to select final burst to arrange its accesses to the memory. The decision is made based on list access latency.
Now final dynamic memory access scheduler follows the following priority scheme:

1. (HIGHEST) Refresh request resulting from refresh must level urgency.
2. Read request without higher priority write, selected from above scheduling priority.
3. Refresh request resulting from refresh need level of urgency.
4. Write request, selected from above scheduling priority.
5. (LOWEST) Refresh request resulting from refresh may level of urgency.

5. Bus Switching Activity Monitoring

To reduce memory energy use without increasing the program execution time, we developed a SDRAM power mode management scheme that uses a bus switching activity monitor to initiate low power mode operations as well as page mode selection. This scheme successfully reduces power consumption of SDRAM modules when the memories have many inter-access cycles. The scheme also reduces program execution time of the system when the SDRAM chips are accessed frequently.

A power mode control scheme performs better when the bus utilization is low. Open page policy architecture performs better when the bus utilization and hit rate are both high. The power mode control scheme reduces hit rate since the SDRAM banks are inactive. The controller operates in open page mode when the bus utilization reaches a threshold value.

We take advantage of bus switching activity monitoring to activate power down mode. Power down mode decides to implement hybrid page policy and enables auto refresh commands in SDRAM memory.

If SDRAM is not accessed for a particular timing it can be demoted to power down mode. Depending upon constant idle cycle predictor based on statistics and calculations constant threshold predictor (CTP) transfers SDRAM into power down mode. The experimental result is analyzed in the paper and results into:

- Power mode control scheme functions effectively when bus utilization is less.
- Open page policy performs better when bus utilization is high.

Bus switching activity monitoring transfers controller into power down mode when idleness predictor reaches its threshold value.

6. EXPERIMENTAL AND RESULT ANALYSIS

To evaluate performance of dynamic access memory scheduler we have used Xilinx ISE12 software tool and Xilinx SPARTAN-6 FPGA, on which modules are implemented. We have used EDE1108ACBG SDRAM and H.264 HD Encoder Hard IP core to examine the results. Timing analysis is done after floor planning design in Plan-ahead tool by XILINX. Figure 4 shows analysis for SDRAM memory architecture, implemented on FPGA.

6.1 Execution Time

Execution time is calculated on the basis of maximum output required time after the clock. Reduction in execution time is achieved by increasing memory bus bandwidth, which is greatly contributed by Dynamic memory access scheduling and efficient address generation policy. Dynamic memory access scheduling achieves an average reduction of 8% and 15%compared with Burst scheduling and First ready scheduling respectively. Figure.5 shows Execution time comparison of all memory access scheduler with dynamic memory access scheduler.

6.2 Hardware Complexity Analysis

We have added refresh policy so that hardware complexity is increased. This is calculated on the basis on device utilization.Figure.6 shows Device utilization is increased by 9%.

7. CONCLUSION

Memory scheduling is improved by reordering the access sequences to the memory. Improvement in row hit rate causes better bus utilization and helps to decrease execution time. Refresh rate is delayed as long as possible to effectively maximize performance. Analysis of memory access patterns of H.264/AVC encoder greatly helped to design potent address generator. Address generator reduces number of redundant memory cycles for page breaks. From results it is concluded that execution time is decreased though hardware complexity is increased. Bus switching monitoring plays a major role in reducing energy consumption by effective utilization of address generation policy.

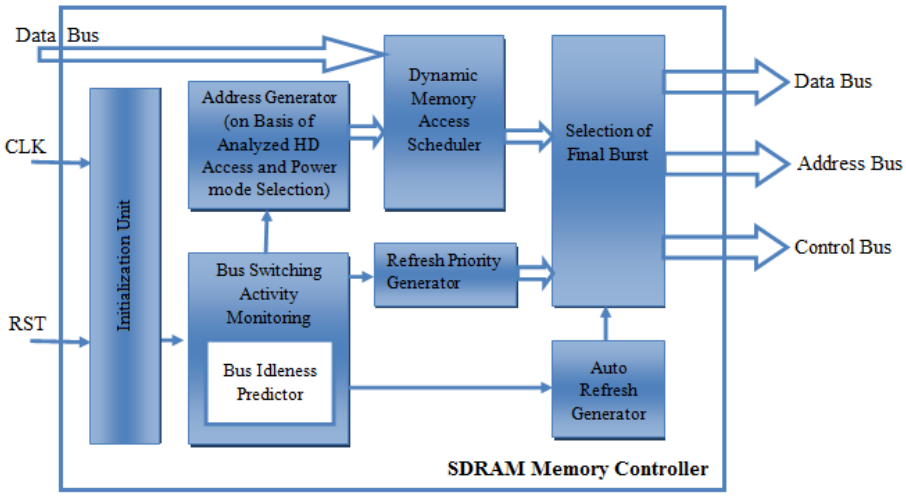

Figure.4 SDRAM Memory Architecture

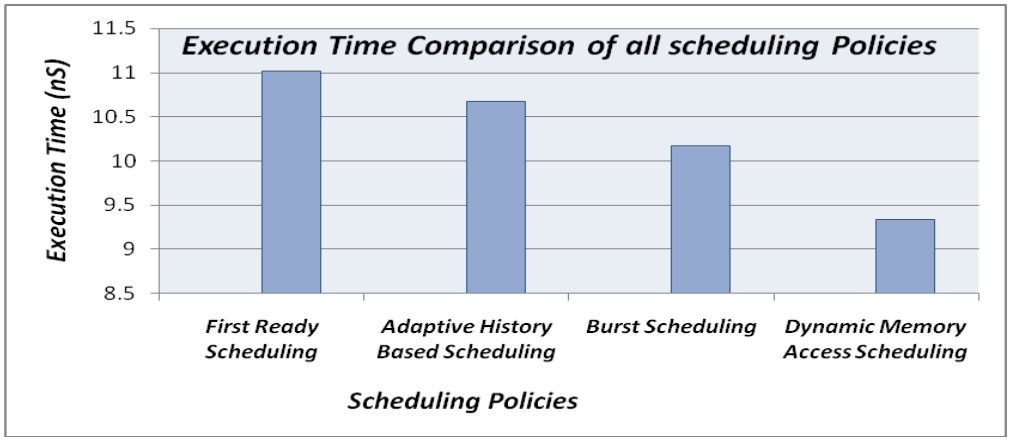

Figure.5. Execution Time Comparison of all Scheduling Policies

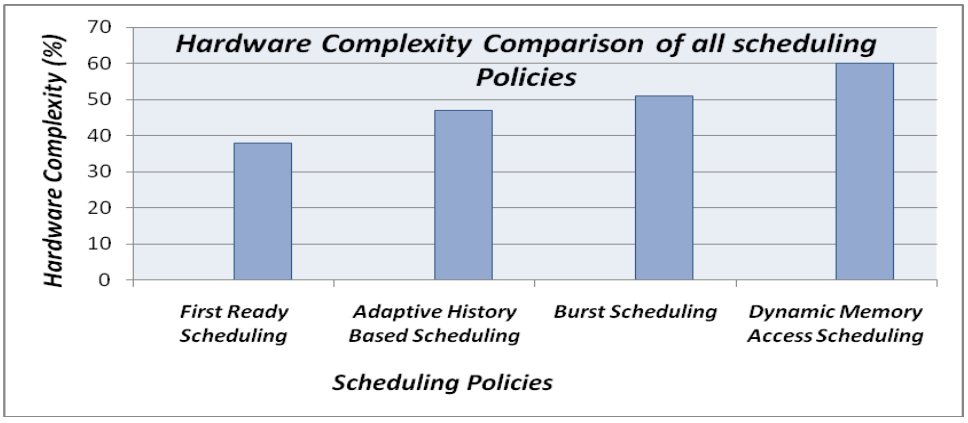

Figure.6. Hardware Complexity Comparison of all Scheduling Policies

8. REFERNCES

[1] Bertrand Le Gal, Emmanuel Casseau, and Sylvain Huet "Dynamic Memory Access Management for High-Performance DSP Applications Using High-Level Synthesis" IEEE TRANSACTIONS ON VLSI SYSTEMS, VOL. 16, Issue NO. 11, NOVEMBER.2008, pp: 1454-1463.

[2] Hu Hongqi; Sun Jingnan; Xu Jiadong; , "High Efficiency Synchronous DRAM Controller for H.264 HDTV Encoder", 4th IEEE Conference on Industrial Electronics & Applications 2009, pp.2132-2136, 25-27 May 2009.

[3] Ibrahim Hur, CalvinLlin, "Adaptive History-Based Memory Schedulers", International Symposium on Microarchitecture, Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture, pp 343 – 354,2004. Jun Shao and Brian T. Davis "A Burst Scheduling Access Reordering Mechanism", pp: 285-294 Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture.

[4] K.Denolf, C.Blanch, "Initial Memory Complexity Analysis of the AVC CODEC", SIPS'02, IEEE Workshop.

[5] Memory Systems: Cache, DRAM, Disk. Bruce Jacob, Spencer W. Ng, and David T. Wang, with contributions by Samuel. ISBN 978-0-12-379751-3. Morgan Kaufmann Publishers, September 2007

[6] Ronny Lee Arnold, Donald Charles Soltis, "Preventing Write-After-Write Data Hazards By Cancelling Earlier Write When No Interleaving Instruction Uses Value To Be Written By The Earlier Write", UNITED STATES PATENT.

[7] Scott Rixner, William J. Dally, Ujval J. Kapasi, Peter Mattson, and John D. Owens "Memory Access Scheduling", Appears in ISCA-27 (2000)

[8] Shih-Chang Hsia, ''Efficient Memory IP Design for HDTV Coding Application'', IEEE Trans. Circuits Syst. Video Tech., vol13,June 2003.

[9] Yi-Nung Liu; Meng-Che Chuang; Shao-Yi Chien, "Bandwidth and local memory reduction of video encoders using Bit Plane Partitioning Memory Management", IEEE International Symposium on Circuits and Systems,2009 (ISCAS 2009). pp.766-769, 24-27 May 2009.

[10] V. Zyuban , P. Kogge "Optimization of high-performance superscalar Power Modes" ,IEEE Transactions on Computers, v.50 n.11, p.1154-1173, November 2001