

# A 0.7-V 1.8-mW H.264/AVC 720p Video Decoder

Vivienne Sze, *Student Member, IEEE*, Daniel F. Finchelstein, *Member, IEEE*,  
Mahmut E. Sinangil, *Student Member, IEEE*, and Anantha P. Chandrakasan, *Fellow, IEEE*

**Abstract**—The H.264/AVC video coding standard can deliver high compression efficiency at a cost of increased complexity and power. The increasing popularity of video capture and playback on portable devices requires that the power of the video codec be kept to a minimum. This work implements several architecture optimizations such as increased parallelism, pipelining with FIFOs, multiple voltage/frequency domains, and custom voltage-scalable SRAMs that enable low voltage operation to reduce the power of a high-definition decoder. Dynamic voltage and frequency scaling can efficiently adapt to the varying workloads by leveraging the low voltage capabilities and domain partitioning of the decoder. An H.264/AVC Baseline Level 3.2 decoder ASIC was fabricated in 65-nm CMOS and verified. For high definition 720p video decoding at 30 frames per second (fps), it operates down to 0.7 V with a measured power of 1.8 mW, which is significantly lower than previously published results. The highly scalable decoder is capable of operating down to 0.5 V for decoding QCIF at 15 fps with a measured power of 29  $\mu$ W.

**Index Terms**—Video codecs, H.264/AVC, CMOS digital integrated circuits, low-power electronics, cache memories, SRAM chips, CMOS memory circuits.

## I. INTRODUCTION

THE use of video is becoming ever more pervasive on battery-operated handheld devices such as cell phones, digital still cameras, personal media players, etc. Annual shipment of such devices already exceeds several hundred million units and continues to grow [1]. There is also an increasing demand for high-definition performance as more high definition content becomes available. Consequently, sophisticated video coding algorithms such as H.264/AVC [2] are needed to reduce transmission and storage costs of the video; however, the high coding efficiency of the H.264/AVC requires high complexity and consequently increases power consumption which is limited for battery-operated devices. In comparison to MPEG-2, which is currently used for HDTV, H.264/AVC provides a 50% improvement in coding efficiency, but requires a 4 $\times$  increase in decoder complexity [3]. Features which

have contributed to increased decoder complexity include the mandatory deblocking filtering and increased motion vector resolution. Accordingly, these blocks tend to consume the majority of the power. Video processing involves a significant amount of data transfer and high definition videos increase the required memory bandwidth. As a result, memory optimization plays an important role in reducing system power.

Voltage scaling is an effective technique that reduces energy consumption by a quadratic factor at the cost of increased circuit delay. Specifically, the circuit suffers an almost linear increase in delay above the threshold voltage, and an exponential increase in delay in the subthreshold region. This decreased speed is a challenge for real-time applications such as video decoding where on average a new frame must be computed every 33 ms. This paper describes a video decoder that enables low voltage operation by reducing the total number of cycles required per frame without increasing the levels of logic in the critical path between registers. This allows for a longer clock period, which leads to a lower supply voltage and lower energy. In order to maintain the throughput necessary for high definition decoding at low voltages, parallelism and pipelining must be strategically used to compensate for the slower circuits [4]. In addition, SRAMs need to be redesigned in order to operate at low voltages due to increased sensitivity to variations.

State-of-the-art ASIC H.264/AVC decoders [5]–[8] have used various micro-architecture techniques to reduce the number of operations, which lowers the operating frequency and consequently power consumption. In this work, *additional* architecture optimizations are introduced that enable aggressive voltage scaling to lower the energy per operation, which further reduces the power consumption of high-definition decoding [9].

References [7] and [8] both reduce the cycle count required for motion compensation and deblocking filtering by optimizing the processing order to eliminate redundant memory accesses. In this work, the cycle count is further reduced by identifying inputs to these units, as well as others, that can be processed in parallel. References [5]–[8] all adopt a hybrid 4  $\times$  4 block/macroblock pipeline scheme. In this work, a 4  $\times$  4 block pipelining scheme with optimally sized FIFOs between stages is used to adapt for workload variability and consequently increase the decoder throughput. Finally, [5]–[8] as well as this work reduce external (off-chip) memory bandwidth by maximizing data reuse and exploiting various different caching schemes. Reference [7] highlights that internal memories consume a significant portion of the core power. In this work, the caches are implemented with custom voltage scalable SRAMs to further minimize memory access power.

The proposed design also uses multiple voltage/frequency domains to enable an optimum amount of voltage scaling. Dynamic voltage and frequency scaling (DVFS) can then be used

Manuscript received January 23, 2009; revised April 25, 2009. Current version published October 23, 2009. This paper was approved by Guest Editors Hoi-Jun Yoo and SeongHwan Cho. This work was funded by Nokia and Texas Instruments. Chip fabrication was provided by Texas Instruments. The work of V. Sze was supported by the Texas Instruments Graduate Women's Fellowship for Leadership in Microelectronics and NSERC.

V. Sze, M. E. Sinangil, and A. P. Chandrakasan are with the Microsystems Technology Laboratories, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: sze@mit.edu; sinangil@mit.edu, anantha@mtl.mit.edu).

D. Finchelstein is with Nvidia Corporation, Santa Clara, CA 95050 USA (e-mail: dfinchel@alum.mit.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSSC.2009.2028933

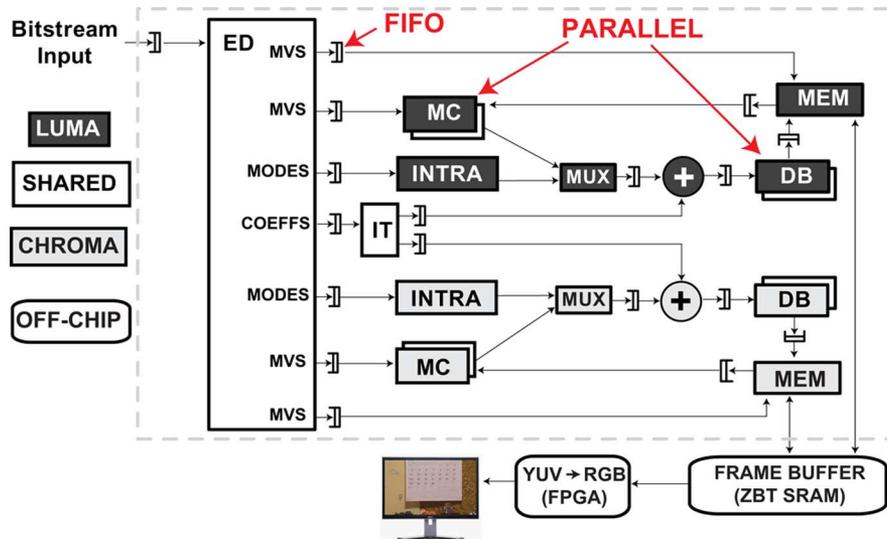


Fig. 1. H.264/AVC decoder architecture. Note that feedback from last line is shown separately in Fig. 10.

by the decoder to efficiently adapt to the varying workloads on each of the domains. A similar approach is used in [10] to adapt to the different workloads between the audio and video modules, and in [11] to adapt between the vertex shader, RISC processor and rendering engine in a 3-D graphics processor.

This paper is organized as follows. Section II discusses the overall decoder architecture and describes the use of pipelining at the system level. Section III describes the use of parallelism internal to the decoder processing units. Section IV provides an analysis of the impact of domain partitioning on power. In Section V, the use of DVFS for adapting to frame workload variation is proposed. Section VI describes optimizations to reduce memory bandwidth and power. Finally, Section VII presents the measured results.

## II. DECODER PIPELINE ARCHITECTURE

The top-level architecture of the decoder hardware is shown in Fig. 1. At the system level of the decoder, FIFOs of varying depths connect the major processing units: entropy decoding (ED), inverse transform (IT), motion compensation (MC), spatial prediction (INTRA), deblocking filter (DB), memory controller (MEM) and frame buffer (FB). The pipelined architecture allows the decoder to process several  $4 \times 4$  blocks of pixels simultaneously, requiring fewer cycles to decode each frame.

The number of cycles required to process each  $4 \times 4$  block varies for each unit as shown in Table I. The table describes the pipeline performance for decoding P-frames (temporal prediction). Most of the optimization efforts were focused on P-frame performance, since they occur more frequently than I-frames (spatial prediction) in highly compressed videos.

One of the challenges in the system design of the video decoder is that the number of cycles required to process each block of pixels changes from block to block (i.e., each unit has varying workload). Consequently, each decoder unit has a range of cycle counts as shown in Table I. For instance, the number of cycles for the ED depends on the number of syntax elements (e.g., non-zero coefficients in residual, motion vectors, etc.) and is typically proportional to the bitrate. As another example, the

TABLE I  
CYCLES PER  $4 \times 4$  BLOCK FOR EACH UNIT IN P-FRAME PIPELINE OF FIG. 1, ASSUMING NO STALLING TAKEN FOR 300 FRAMES OF THE "MOBCAL" SEQUENCE. EACH  $4 \times 4$  BLOCK INCLUDES A SINGLE  $4 \times 4$  LUMA BLOCK AND TWO  $2 \times 2$  CHROMA BLOCKS. [ ] IS PERFORMANCE AFTER SECTION III OPTIMIZATIONS

Pipeline Unit	Min Cycles	Max Cycles	Avg Cycles	
ED	0	33	4.6	
IT	0	4	1.6	
Luma	MC	4 [2]	9 [4.5]	4.6 [2.3]
	DB	8 [2]	12 [6]	8.9 [2.9]
	MEM	8	31	18
Chroma	MC	8 [2]	8 [2]	8 [2]
	DB	5 [2.5]	8 [4]	6.6 [3.3]
	MEM	10	10	10

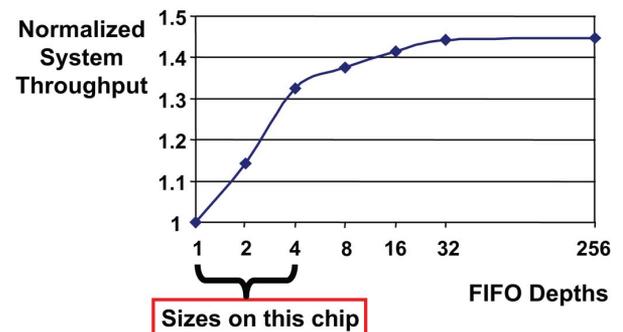


Fig. 2. Longer FIFOs average out workload variations to minimize pipeline stalls. Performance simulated with equal FIFO depths for "mobcal" sequence.

number of cycles for the MC unit depends on the corresponding motion vectors. An integer-only motion vector requires fewer cycles (4 cycles per  $4 \times 4$  luma block) as compared to one which contains fractional components (9 cycles per  $4 \times 4$  luma block).

To adapt for the workload variation of each unit, FIFOs were inserted between each unit. These FIFOs also distribute the

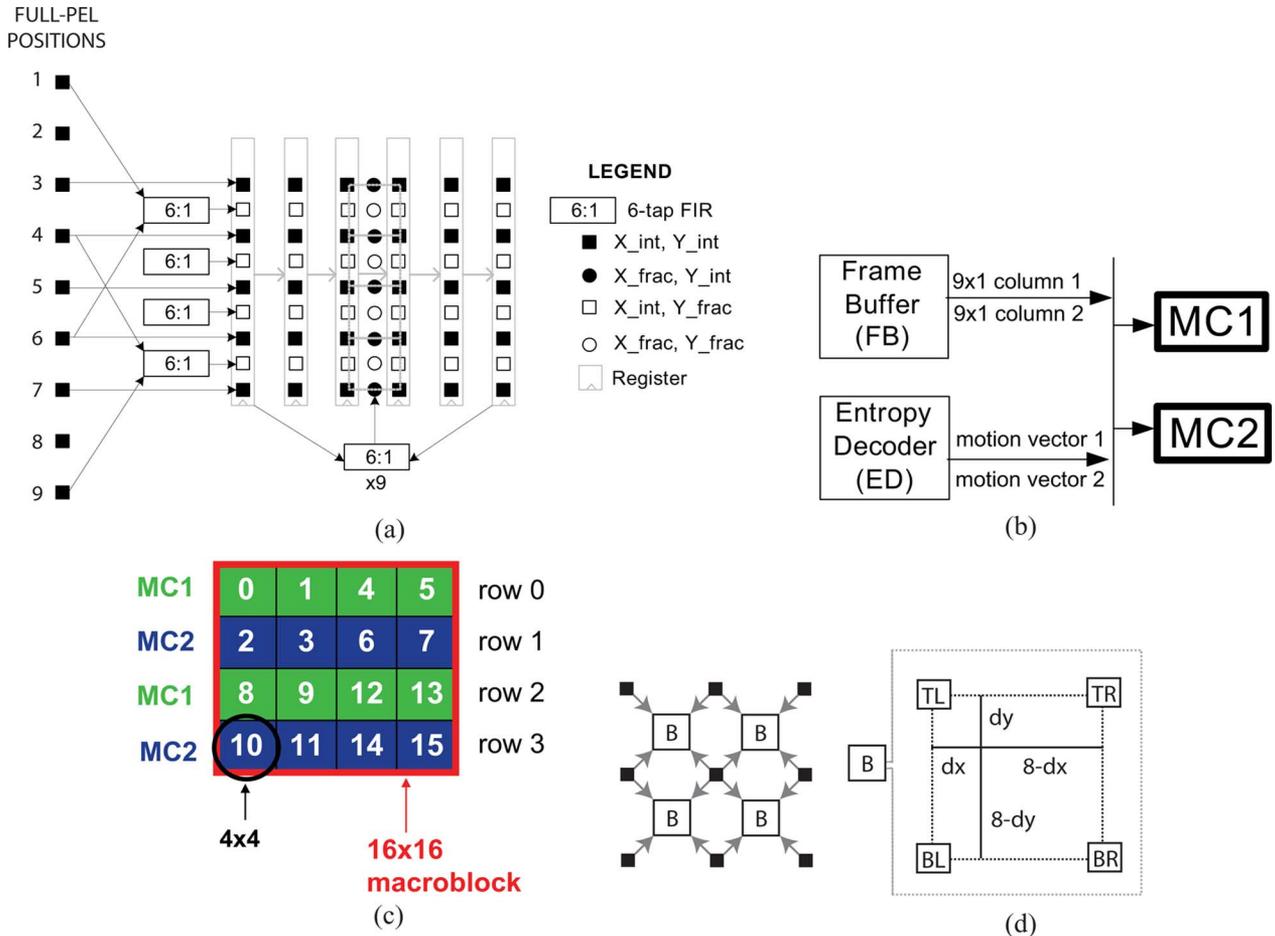


Fig. 3. Parallel Motion Compensation architecture. (a) Luma interpolator pipelined architecture. (b) Parallel MC interpolators. (c) The parallel interpolators filter different rows of  $4 \times 4$  blocks. The numbered  $4 \times 4$  blocks reflect the processing order within a macroblock. (d) Chroma bilinear filter (B) is replicated 4 times. Each filter completes in one cycle and consists of four 8-bit multipliers and four 16-bit adders.

pipeline control and allow the units to operate out of lockstep. Unlike [12], where the number of cycles per pipeline stage is fixed, FIFOs allows time borrowing of cycles between stages. The FIFOs help to average out the cycle variations which increases the throughput of the decoder by reducing the number of stalls, as described in [13]. Fig. 2 shows that the pipeline performance can be improved by up to 45% by increasing the depths of the  $4 \times 4$  block FIFOs in Fig. 1. For very large FIFO depths, all variation-related stalls are eliminated and the pipeline performance approaches the rate of the unit with the largest average cycle count. This performance improvement must be traded-off against the additional area and power overhead introduced by larger FIFOs.

In the simulation results presented in Fig. 2, all FIFOs were set to equal depths. However, deeper FIFOs should ideally be used only when they provide a significant performance improvement. For example, placing a deeper FIFO between the ED and IT unit reduces many stalls, but a minimum-sized FIFO between the DB and MEM units is sufficient. In the decoder test chip, FIFO depths between 1 and 4 were chosen in order to reduce FIFO area while still reducing pipeline stalls.

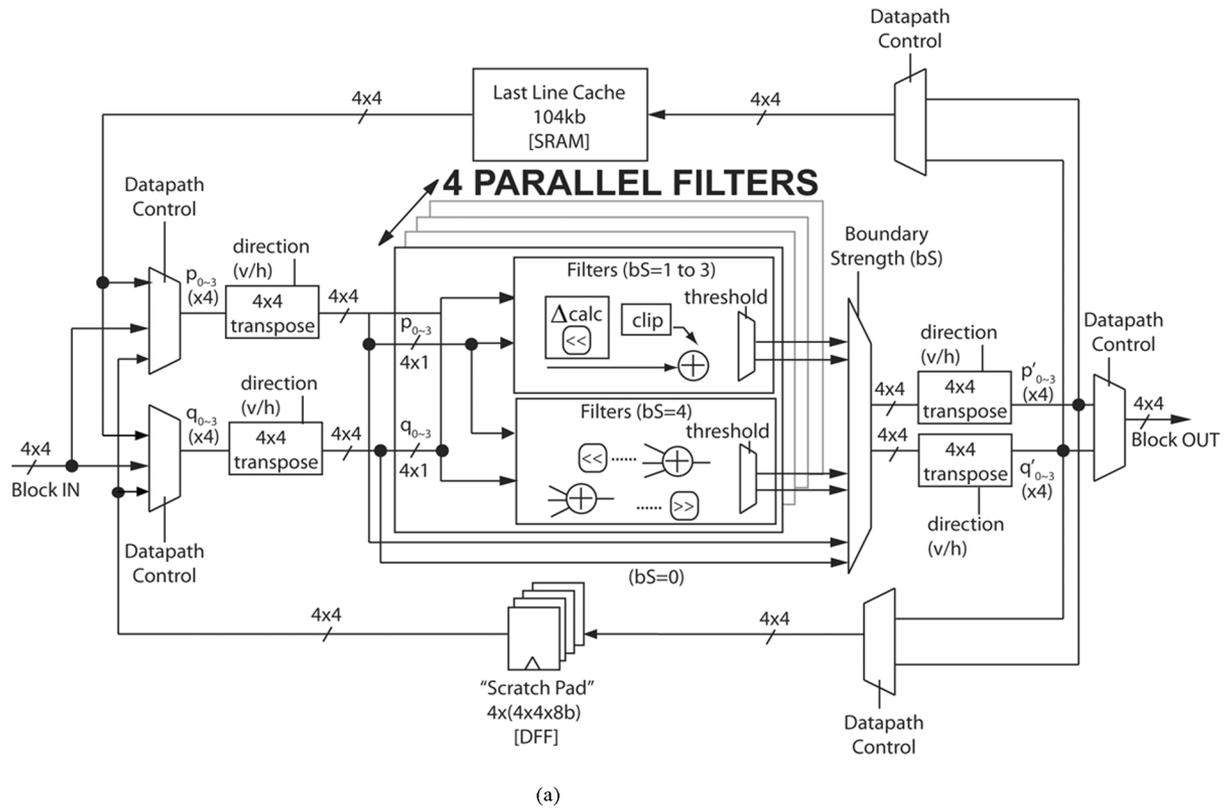
For maximum concurrency, the average cycles consumed by each stage of the pipeline, which is equivalent to each processing unit in this implementation, should be balanced. Section III will

describe how parallelism can be used to reduce cycle counts in the bottleneck units to help balance out the cycles in each stage of the pipeline.

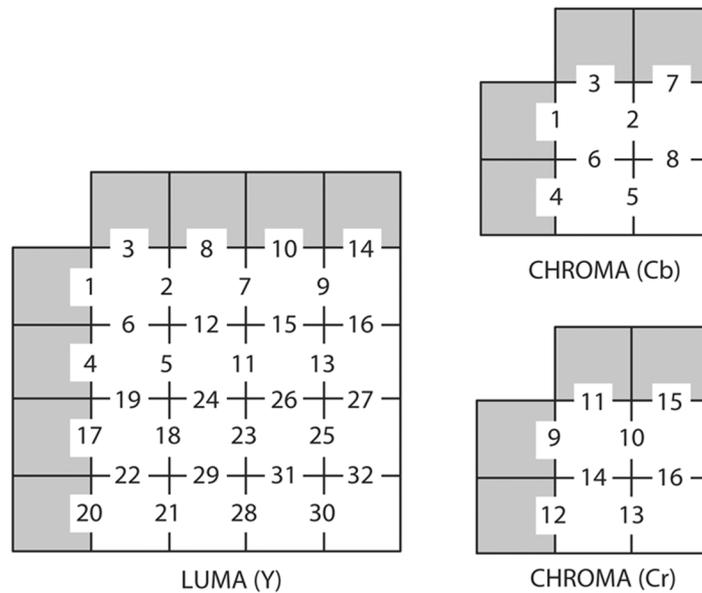
Additional concurrency is achieved by processing the luma and chroma components with separate pipelines that share minimal hardware and are mostly decoupled from each other. The cycles consumed for chroma are also shown in Table I. In most cases, the luma and chroma components of each  $4 \times 4$  block are processed at the same time which enables further cycle count reduction. However, the two pipelines do have dependencies on each other, which sometimes prevents them from running at the same time. For example, both pipelines use the same ED at the start, since this operation is inherently serial and produces coefficients and motion vectors for both pipelines. To reduce hardware costs, the luma and chroma pipelines also share the IT unit, since this unit has a relatively low cycle count per block relative to the rest of the units as shown in Table I.

### III. PARALLELISM

Parallelism can be used within each processing unit to reduce the number of cycles required to process each  $4 \times 4$  block and balance out the cycles in each stage of the pipeline. This is particularly applicable to the MC and DB units, which were found



(a)



(b)

Fig. 4. Parallel Deblocking Filter architecture. (a) Deblocking filter architecture for luma filtering. (b) Deblocking edge filtering order. Note that luma and chroma edges are filtered in parallel.

to be key bottlenecks in the system pipeline. This is not surprising as they were identified as the units of greatest complexity in H.264/AVC [14].

**A. Motion Compensation (MC)**

Given a motion vector, the MC unit predicts a  $4 \times 4$  block in the current frame from pixels in the previous frames to

exploit temporal redundancy. The previous frames are stored in the frame buffer. When the motion vector is integer-valued (full-pel), the predicted  $4 \times 4$  block can be found in its entirety in a previous frame. For increased coding efficiency, motion vectors in H.264/AVC can have up to quarter-pel resolution. When either the X or Y component of the motion vector is fractional, the predicted  $4 \times 4$  block must be interpolated from

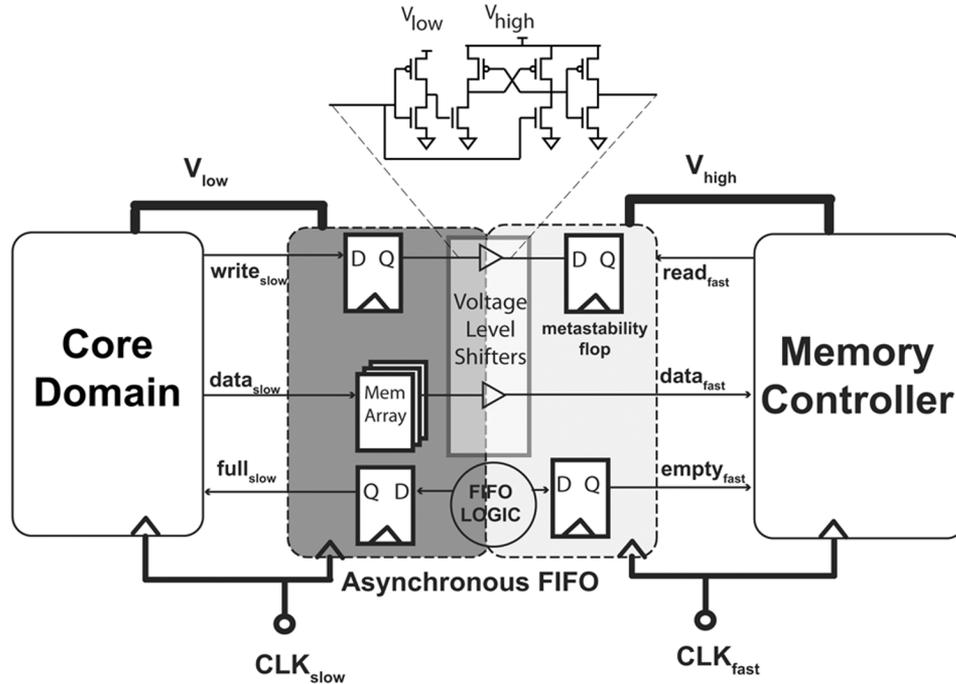


Fig. 5. Independent voltage/frequency domains are separated by asynchronous FIFOs and level-converters.

pixels at full-pel locations in the previous frame. Thus, the main operation in motion compensation involves interpolation and filtering.

The luma and chroma components are interpolated differently. Luma interpolation involves using a 1-D 6-tap filter to generate the half-pel locations. A 1-D bilinear filter is then used to generate the quarter-pel locations. Accordingly, a  $4 \times 4$  luma block is predicted from an area of at most  $9 \times 9$  pixels in a previous frame. Chroma interpolation involves the use of a 2-D bilinear filter and each  $2 \times 2$  chroma block is predicted from an area of  $3 \times 3$  pixels.

The luma interpolator architecture is shown in Fig. 3(a) and is similar to the design in [15]. The datapath of the luma interpolator is made up of  $(6 \times 9)$  8-bit registers,  $(4+9)$  6-tap filters, and four bilinear filters. The interpolator uses a 6-stage pipeline. At the input of the pipeline, for vertical interpolation, a column of 9 pixels is read from the frame buffer and used to interpolate a column of 4 pixels. A total of 9 pixels, representing the full and half-pel locations, are stored at every stage of the interpolator pipeline; specifically, the 4 interpolated half-pel pixels and the 5 center (positions 3 to 7) full-pel pixels of the 9 pixels from the frame buffer are stored. The 9 registers from the 6 stages are fed to 9 horizontal interpolators. Finally, 9:2 muxes are used to select two pixels located at full or half-pixel locations as inputs to the bilinear interpolator for quarter-pel resolution.

To improve the throughput of MC, a second identical interpolator is added in parallel as shown in Fig. 3(b). The first interpolator predicts  $4 \times 4$  blocks on the even rows of a macroblock, while second predicts  $4 \times 4$  blocks on the odd rows [Fig. 3(c)]. This parallel structure can double the throughput of the MC unit if during each cycle, both motion vectors are available and two new columns of 9 pixels from the frame buffer are available at

the inputs of both interpolators. The chroma interpolator is replicated four times such that it can predict a  $2 \times 2$  block every cycle [Fig. 3(d)].

The MC unit has a logic gate count of 102.5 k which includes the interpolators, pipeline registers, control logic, muxes and related pipeline control. The additional interpolators (one luma and three chroma) account for 21.7 k of the MC gate count.

### B. Deblocking Filter (DB)

The boundary strength information of the adaptive filter is the same for all edges on a given side of a  $4 \times 4$  block. Accordingly, the DB is designed to have 4 luma and 2 chroma filters running in parallel which share the same boundary strength calculation, and filter an edge of a  $4 \times 4$  block every cycle. The luma architecture is shown in Fig. 4(a). For additional cycle reduction, the luma and chroma filters operate at the same time, assuming the input data and configuration parameters are available.

A luma macroblock ( $16 \times 4 \times 4$  blocks) has 128 pixel edges that need to be filtered, so with 4 luma filters a macroblock takes 32 clock cycles to complete. Unlike previous implementations [16], filtering on  $4 \times 4$  blocks begins before the entire macroblock is reconstructed. To minimize stalls, the edge filtering order, shown in Fig. 4(b), was carefully chosen to account for the  $4 \times 4$  block processing order [shown in Fig. 3(c)] while adhering to the left-right-top-bottom edge order specified in H.264/AVC [2].

A single-port on-chip SRAM cache with  $4 \times 4 \times 8$ -bit data-width is used to store pixels from the top and left macroblocks. Due to the  $4 \times 4$  block processing order and the edge order constraints, certain  $4 \times 4$  blocks need to be stored temporarily before all four of its edges are filtered and it can be written out. These partially filtered blocks are either stored in the on-chip

cache or a “scratch pad” made of flip-flops that can hold up to four  $4 \times 4$  blocks. This “scratch pad” along with the chosen edge filtering order minimize the number of stalls from read/write conflicts to the on-chip cache. The on-chip cache has a 2-cycle read latency resulting in a few extra stall cycles when prefetching is not possible. Taking into account this overhead, the average number of cycles required by DB for a luma  $4 \times 4$  block is about 2.9.

Each of the two chroma components of a macroblock has 32 pixel edges to be filtered. Using 2 filters per cycle results in slightly more than 32 clock cycles per macroblock. When accounting for stalls, the number of cycles per macroblock is about the same as luma.

Overall, the number of cycles required by DB is 52 cycles per macroblock, which is less than other deblocking filter implementations [16]. The DB unit has a logic gate count of 78.3 k which includes the filters, boundary calculation, transposition buffers, “scratch pad”, control logic, muxes, and related pipeline control. The additional parallel filters (three luma and one chroma) account for 15.8 k of the DB gate count.

#### IV. MULTIPLE VOLTAGE/FREQUENCY DOMAINS

The decoder interfaces with two 32-bit off-chip SRAMs which serve as the frame buffer. To avoid increasing the number of I/O pads, the MEM unit requires approximately  $3 \times$  more cycles per  $4 \times 4$  block than the other processing units, as shown in Table I. In a single domain design, MEM would be the bottleneck of the pipeline and cause many stalls, requiring the whole system to operate at a high frequency in order to maintain performance. This section describes how the architecture can be partitioned into multiple frequency and voltage domains.

Partitioning the decoder into two domains (MEM in the memory controller domain and the other processing units in the core domain) enables the frequency and voltage to be independently tailored for each domain. Consequently, the core domain, which can be highly parallelized, fully benefits from the reduced frequency and is not restricted by the memory controller with limited parallelism.

The two domains are completely independent, and separated by asynchronous FIFOs [17] as shown in Fig. 5. Voltage level-shifters (using differential cascode voltage switch logic) are used for signals going from a low to high voltage.

Table I shows that there could be further benefit to also placing the ED unit on a separate third domain. The ED is difficult to speed up with parallelism because it uses variable length coding which is inherently serial.

Table II shows a comparison of the estimated power consumed by the single domain design versus a multiple (two and three) domain design. The frequency ratios are derived from Table I and assume no stalls. For a single domain design the voltage and frequency must be set at the maximum dictated by the worst case processing unit in the system. It can be seen that the power is significantly reduced when moving from one to two domains. The additional power savings for moving to three domains is less significant since the impact of frequency reduction on voltage scaling is reduced as the operating point is nearing the threshold voltage of the transistors; thus, a two-domain design is used.

TABLE II  
ESTIMATED IMPACT OF MULTIPLE DOMAINS ON POWER  
FOR DECODING A P-FRAME

Domains	Frequency Ratio			Voltage [V]			Power [%]
	ED	Core	MEM	ED	Core	MEM	
One	1.00	1.00	1.00	0.81	0.81	0.81	100
Two	0.26	0.26	1.00	0.66	0.66	0.78	75
Three	0.26	0.18	1.00	0.66	0.63	0.78	71

#### V. DYNAMIC VOLTAGE AND FREQUENCY SCALING

The video decoder has a highly variable workload due to the varying prediction modes that enable high coding efficiency. While FIFOs are used in Section II to address workload variation at the  $4 \times 4$  block level, DVFS allows the decoder to address the varying workload at the frame level in a power efficient manner [18].

DVFS involves adjusting the voltage and frequency based on the varying workload to minimize power. This is done under the constraint that the decoder must meet the deadline of one frame every 33 ms to achieve real-time decoding at 30 fps. The two requirements for effective DVFS include accurate workload prediction and the voltage/frequency scalability of the decoder. [19]–[22] propose several techniques to predict the varying workload during video decoding. This work addresses the scalability of the decoder.

DVFS can be performed independently on the core domain and memory controller as their workloads vary widely and differently depending on whether the decoder is working on I-frames or P-frames. For example, the memory controller requires a higher frequency for P-frames versus I-frames. Conversely, the core domain requires a higher frequency during I-frames since more residual coefficients are present and they are processed by the ED unit. Fig. 6 shows the workload variation across the “mobcal” sequence. Table III shows the required voltages and frequencies of each domain for an I-frame and P-frame. Fig. 7 shows the frequency and voltage range of the two domains in the decoder. Once the desired frequency is determined for a given workload, the minimum voltage can be selected from this graph.

To estimate the power impact of DVFS, only two operating points (P-frame and I-frame) shown in Table III are used. The power of the decoder was measured separately for each operating point using a mostly P-frame video and an only I-frame video averaged over 300 frames. The frame type (I or P) can be determined from the slice header. Table IV shows the impact of DVFS for a group of pictures (GOP) size of 15 with a GOP structure of IPPP (i.e., I-frame followed by a series of P-frames; the GOP is the period of I-frames). DVFS can be done in combination with frame averaging for improved workload prediction and additional power savings [22], [23].

For the ASIC described in this paper, the DVFS control loop was implemented off-chip, and the various voltages and frequencies were supplied through input pads. If the voltage regulators and frequency synthesizers were integrated on-chip, the DVFS scheme would have an additional area and power cost versus the existing ASIC. The measured results presented in Section VII do not include this overhead. For reference, the study in [24] quantifies the overhead of fine-grained DVFS

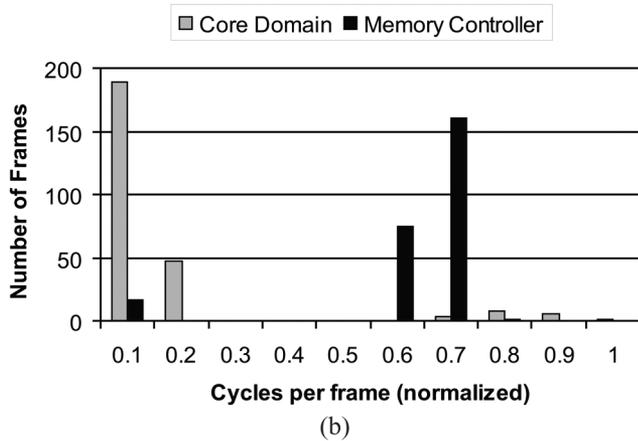
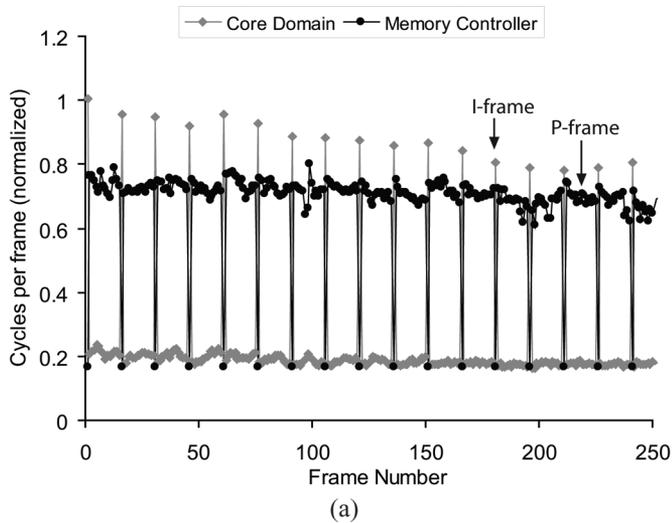


Fig. 6. Workload variation across 250 frames of “mobcal” sequence. (a) Cycles per frame (workload) across sequence. (b) Distribution of cycle variation.

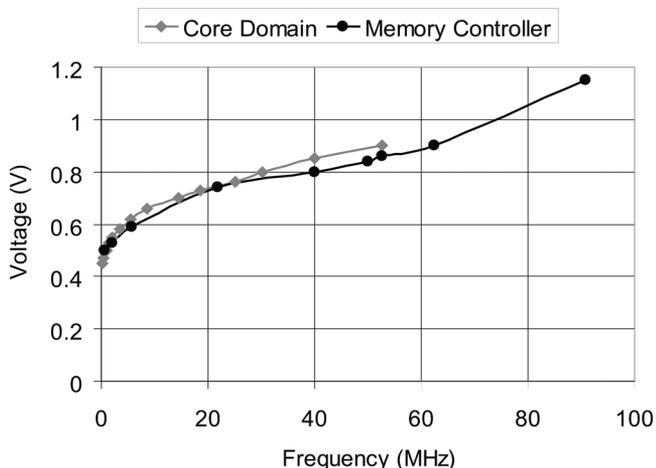


Fig. 7. Measured frequency versus voltage for core domain and memory controller. Use this plot to determine the minimum voltage for a given frequency. Note: The rightmost measurement point has a higher voltage than expected due to limitations in the test setup.

for a multi-processor architecture. The work in [25] describes a multi-domain frequency and voltage controller that covers supply voltages between 1.0 V and 1.8 V, and a corresponding frequency range of 90 MHz to 200 MHz.

TABLE III  
MEASURED VOLTAGE/FREQUENCY FOR EACH DOMAIN FOR I-FRAME AND P-FRAME FOR 720P SEQUENCE

Frame Type	Core		Memory Controller	
	Freq [MHz]	Volt [V]	Freq [MHz]	Volt [V]
P	14	0.70	50	0.84
I	53	0.90	25	0.76

TABLE IV  
ESTIMATED IMPACT OF DVFS FOR GOP STRUCTURE OF IPPP AND SIZE 15

Method	Core		Memory Controller		Relative Power [%]
	Freq [MHz]	Volt [V]	Freq [MHz]	Volt [V]	
No DVFS	53	0.90	50	0.84	100
DVFS	14 or 53	0.70 or 0.90	50 or 25	0.84 or 0.76	75

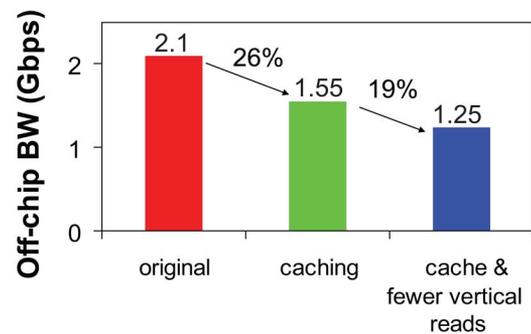


Fig. 8. Reduction in overall memory bandwidth from caching and reuse of MC data on “mobcal” sequence.

## VI. MEMORY OPTIMIZATION

Video processing involves movement of a large amount of data. For the average 720p sequence, without any optimizations, the memory bandwidth is over 2 Gbps, with about 2 to 3 times more reads than writes. Memory optimization and management are key to improving the decoder’s performance and power.

For high definition, each frame is on the order of megabytes which is too large to place on chip (1.4 MBytes/frame for 720p). Consequently, the frame buffer used to store the previous frames required for motion compensation is located off-chip. It is important to minimize the off-chip memory bandwidth in order to reduce overall system power.

Two key techniques are used to reduce this memory bandwidth. The first reduces both reads and writes such that only the DB unit writes to the frame buffer and only the MC unit reads from it. The second reduces the number of reads by the MC unit. The impact of the two approaches on the overall off-chip memory bandwidth can be seen in Fig. 8. The overall memory bandwidth is reduced to 1.25 Gbps.

### A. On-Chip Caching

Only fully-processed pixels are stored in the off-chip memory. Several separate on-chip caches were used to store syntax elements or pixel data that have not been fully processed, as shown in Fig. 10. This includes data such as motion vectors and the last four lines of pixels that are required by the

TABLE V  
MEMORY BANDWIDTH OF CACHES FOR 720P AT 30 FPS

Cache	Size [kb]	Dimensions address x word	I-frame Bandwidth [Mbps]	P-frame Bandwidth [Mbps]
Deblocking (Last 4 lines + Left 4 columns)	104	324x158 (luma)	510	510
		324x158 (chroma)	297	297
Intra prediction (Last line + Left column)	21	324x32 (luma)	61.7	32.4
		162x32 (chroma x2)	57	35.4
Motion Vector	9	80x118	0	25.5
Total Coefficient Count	3	80x40	8.5	7.8
Macroblock Parameters	1	80x7 (luma)	6.1	6.1
		80x7 (chroma)	6.8	6.8
Intra Prediction Mode	1	80x16	3.1	1.7
<b>Total</b>	<b>138</b>	<b>n/a</b>	<b>579</b>	<b>922</b>

deblocking filter. For a P-frame, this caching scheme reduces total off-chip bandwidth by 26% relative to the case where no caches are used. The memory bandwidth of each cache is shown in Table V.

On-chip memories account for a significant portion of the area and power consumption of modern ICs. This makes low-voltage high-density SRAMs very important for energy-constrained systems. Conventional 6T SRAMs fail to operate at low voltages in 65-nm technology node due to severe degradation of read static noise margin (RSNM) and write margin (WM) of the cell. In this work, custom low-voltage SRAMs based on [26] were designed to operate at the desired core voltage and frequency.

The SRAM architecture is shown in Fig. 9. The 8T bit-cell (BC) shown in Fig. 9(b) uses separate ports for write and read operations. This enables optimization of the bit-cell for low-voltage writability and target performance at the same time. Specifically, access transistors in the cell are sized to be stronger than the pMOS load transistors. Doing so causes a degradation of RSNM in a traditional 6T cell but for the 8T cell, due to the de-coupling of read and write ports, this problem is eliminated. Since the effect of transistor sizing can be easily negated at low-voltages due to local variation, cell design for improved writability cannot ensure low-voltage functionality alone. Hence, a write-assist scheme is implemented in this design to improve degraded write margin at low voltages [Fig. 9(c)]. Specifically, row-wise supply node (*MCHd*) is actively pulled down during write accesses to lessen the strength of the feedback inside BC [26].

The cell performs buffered reads through the 2 extra transistors inside the cell, in order to avoid RSNM problem at low voltages. The footer node required to reduce leakage for sub-threshold operation used in [26] is removed. This is permissible since the target voltage range for this system does not require subthreshold operation where drive currents are comparable to leakage currents. Moreover, removing the footer-node from the cell provides better performance due to eliminating an extra nMOS transistor from the read stack as shown in Fig. 11. A pseudo-differential sensing scheme compatible with the 8T cell was implemented, as shown in Fig. 9(d). The latch-based sense-amplifier uses inputs *RDBL* and a global reference voltage (*snsRef*) to produce *DATAOUT*, which is then buffered to the memory interface's output. The negative-edge of the clock is used to create the strobe signal for the sense-amplifier (*snsEN*).

Designing the SRAM interface for a large voltage range is also a challenging problem. At low voltage levels, relative delays between critical signals (e.g., *clk* and *WL*) can vary a lot causing timing failures. To address this issue, (i) self-timed circuits and (ii) reconfigurable delay lines are used in the memory design. For example, the pre-charge scheme is designed to be self-timed in this design. The variance of pre-charge time can be large among different columns requiring excessive timing margin to account for the worst case delay. However, a self-timed scheme [Fig. 9(d)] will start pre-charging as soon as the sense-amplifier outputs are resolved. This eliminates the ambiguity between the edge starting pre-charge phase and sense-amplifier output resolution.

Compared to a conventional SRAM, this custom design can operate at a much lower supply voltage and trade-off performance for energy.

### B. Reducing MC Redundant Reads

The off-chip frame buffer used in the system implementation has a 32-bit data interface. Decoded pixels are written out in columns of 4, so writing out a  $4 \times 4$  block requires 4 writes to consecutive addresses. When interpolating pixels for motion compensation, a column of 9 pixels is required during each MC cycle. This requires three 32-bit reads from the off-chip frame buffer.

During MC, some of the redundant reads are recognized and avoided. This happens when there is an overlap in the vertical or horizontal direction and the neighboring  $4 \times 4$  blocks (within the same macroblock) have motion vectors with identical integer components [8]. As discussed in Section III-A, the MC interpolators have a 6-stage pipeline architecture which inherently takes advantage of the horizontal overlap. The reuse of data that overlap in the horizontal direction helps to reduce the cycle count of the MC unit since those pixels do not have to be re-interpolated.

The two MC interpolators are synchronized to take advantage of the vertical overlap. Specifically, any redundant reads in the vertical overlap between rows 0 and 1 and between rows 2 and 3 [in Fig. 3(c)] are avoided. As a result, the total off-chip memory bandwidth is reduced by an additional 19%, as shown in Fig. 8.

In future implementations, a more general caching scheme can be used to further reduce redundant reads if it takes into account:

- 1) adjacent  $4 \times 4$  blocks with slightly different motion vectors

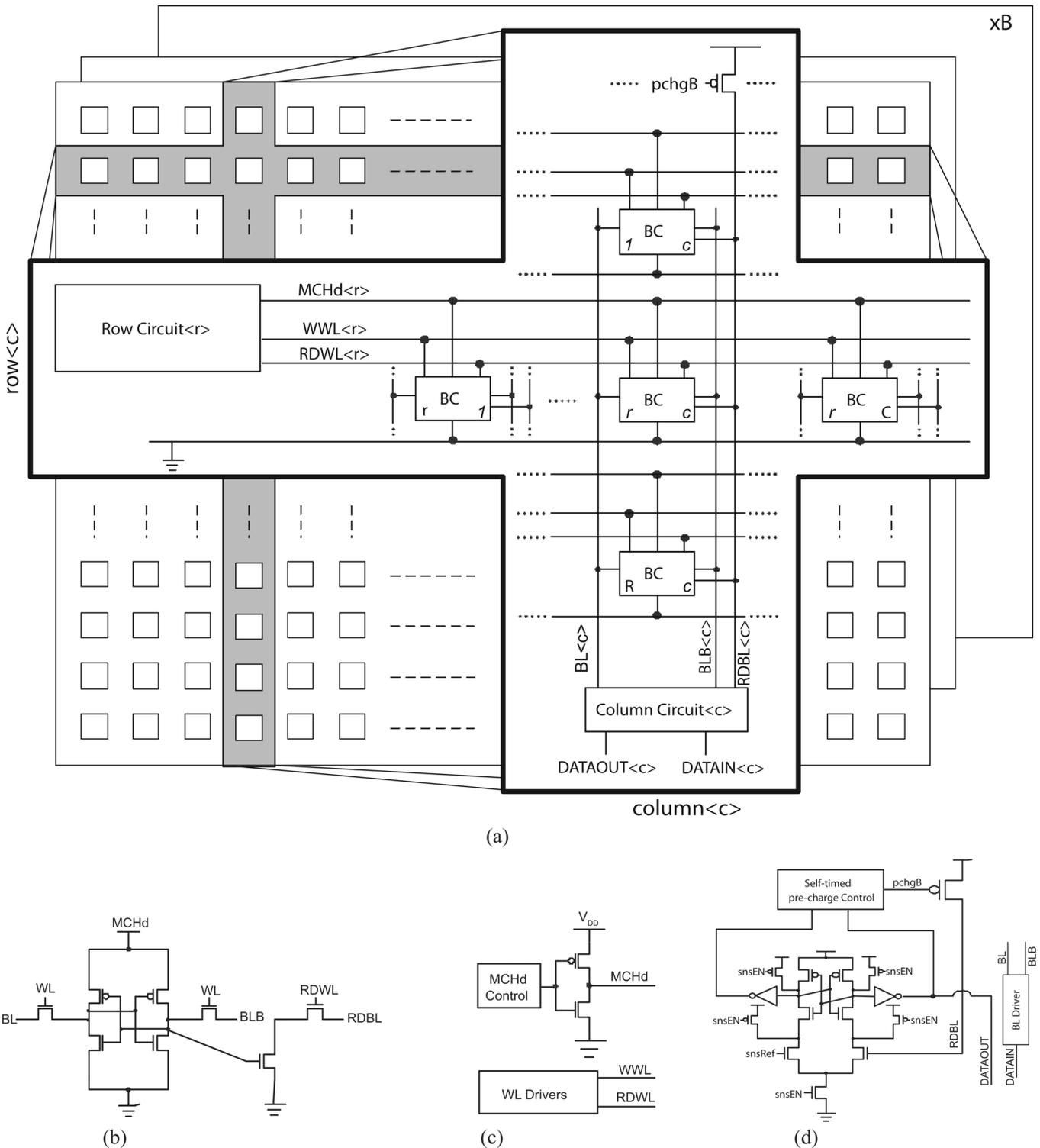


Fig. 9. Low-voltage SRAM architecture. (a) Array architecture where number of columns  $C \leq 158$ , number of rows  $R \leq 64$ , and number of banks  $B \leq 5$ . (b) 8T bit cell (BC). (c) Row circuit. (d) Column circuit.

- 2) overlap in read areas between nearby macroblocks on the same macroblock row
- 3) overlap in read areas between nearby macroblocks on two consecutive macroblock rows

The potential benefits of this scheme can be evaluated with the help of a variable-sized fully-associative on-chip cache, as shown in Fig. 12(a). A small cache of 512-Bytes (128 addresses)

can help reduce the off-chip read bandwidth by a further 33% by taking advantage of the first two types of redundancies in the above list. In order to take advantage of the last redundancy in the list, a much larger cache is needed (32 kBytes) to achieve a read bandwidth reduction of 56% with close to no repeated reads. The associativity of this cache also impacts the number of reads, due to the hit rate. Fig. 12(b) shows, a fully associative

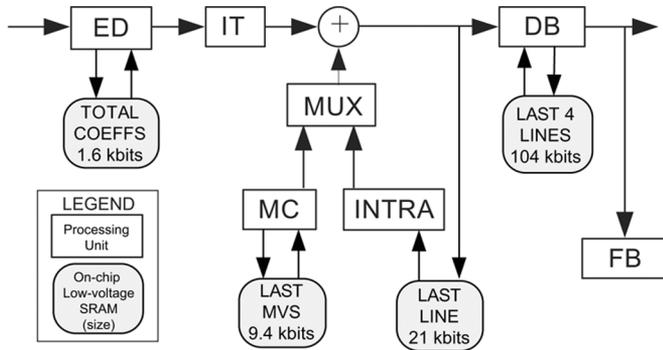


Fig. 10. On-chip caches reduce off-chip memory bandwidth.

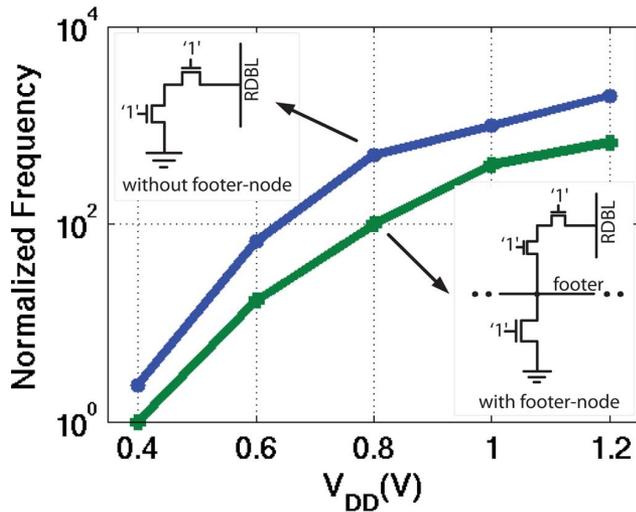


Fig. 11. Impact of footer on SRAM performance.

512-Byte cache (0 set bits) provides the largest hit rate, while a direct-mapped scheme (7 set bits) has the lowest hit rate. The benefits of this motion compensation cache must be weighed against the area overhead of data and address tags and the energy required to perform cache reads and writes.

## VII. RESULTS AND MEASUREMENTS

The H.264/AVC Baseline Level 3.2 decoder, shown in Fig. 13 was implemented in 65-nm CMOS. A summary of the chip statistics is shown in Table VI. The power was measured when performing real-time decoding of several 720p video streams at 30 fps (Table VII) [9]. The video streams were encoded with x264 software [27] with a GOP size of 150 (P-frames dominate). Fig. 14 shows a comparison of our ASIC with other decoders. To obtain the power measurements of the decoder at various performance points, the frame rate of the video sequence was adjusted to achieve the equivalent Mpixels/s of the various resolutions. At 720p, the decoder also has lower power and frequency relative to D1 of [5]. The decoder can operate down to 0.5 V for QCIF at 15 fps for a measured power of 29  $\mu$ W. The power of the I/O pads was not included in the measurement comparisons. The reduction in power over the other reported decoders can be attributed to a combination of using the low-power techniques described in this paper and a more advanced process.

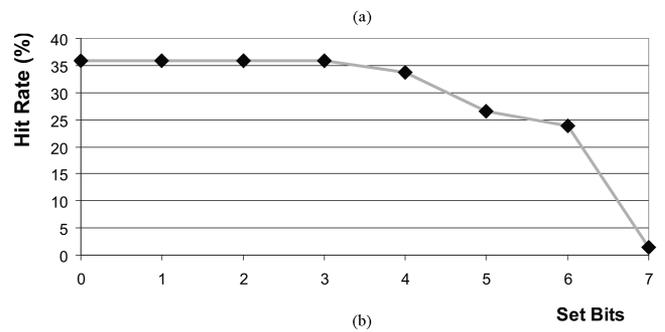
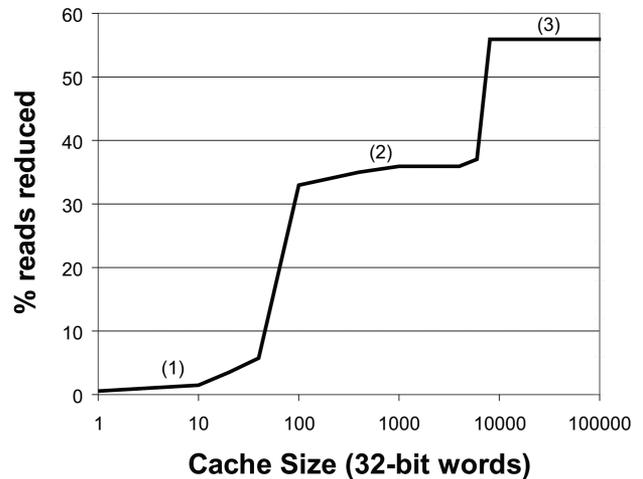


Fig. 12. Motion compensation cache. Results are given for “mobcal” sequence. (a) Effect of motion compensation cache size of read reductions. (b) Off-chip bandwidth reduction versus associativity.

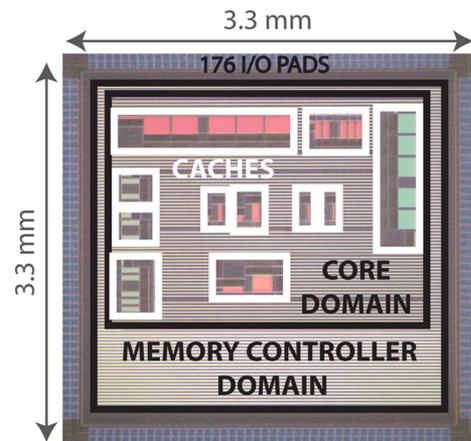


Fig. 13. Die photo of H.264/AVC video decoder (domains and caches are highlighted).

The off-chip frame buffer was implemented using 32-bit-wide SRAMs [28]. An FPGA and VGA drivers were used to interface the ASIC to the display [29]. A photo of the test setup is shown in Fig. 15. The variation in performance across 15 dies is shown in Fig. 16. The majority of the dies operate at 0.7 V.

It is important to consider the impact of this work at the system level. As voltage scaling techniques can reduce the decoder power below 10 mW for high definition decoding, the

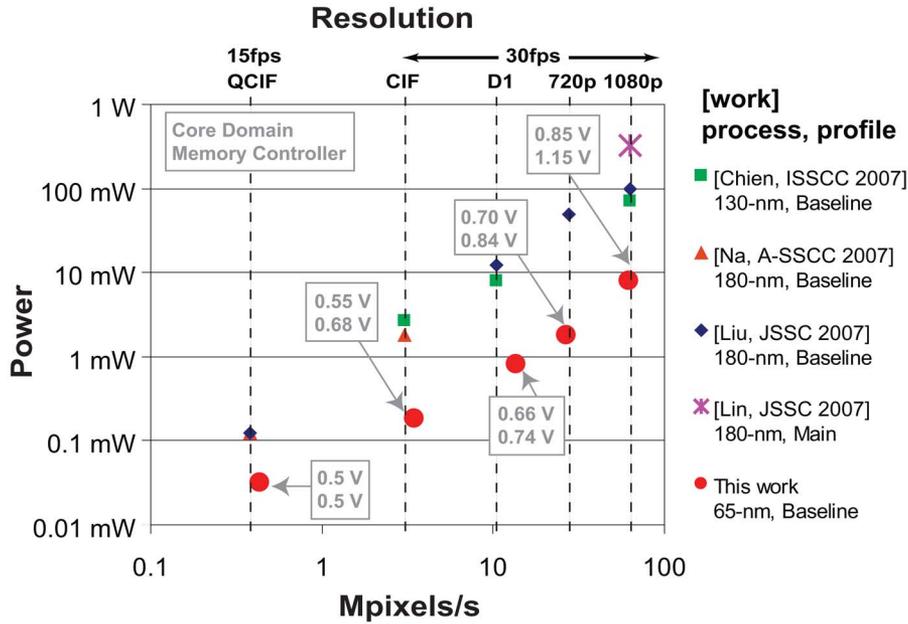


Fig. 14. Comparison with other H.264/AVC decoders [5]–[8]

TABLE VI  
SUMMARY OF CHIP IMPLEMENTATION

<b>Video Coding Standard</b>	H.264/AVC Baseline Level 3.2
<b>Technology</b>	65-nm
<b>Core Area</b>	2.76 mm × 2.76 mm
<b>Logic Gate Count (NAND-2)</b>	314.8k
<b>Package</b>	176-pin QFP
<b>On-Chip Memory (SRAM)</b>	17 kB
<b>Supply Voltage (720p at 30fps)</b>	0.7 V (core) 0.85 V (memory controller) 1.8 V (I/O)
<b>Operating Frequency (720p at 30fps)</b>	14 MHz (core) 50 MHz (memory controller)
<b>Measured Power (without I/O)</b>	1.8 mW

TABLE VII  
MEASURED PERFORMANCE NUMBERS FOR 720P AT 30 FPS

Video	mobcal	shields	parkrun
# of Frames	300	300	144
Bitrate [Mbps]	5.4	7.0	26
Off-chip Bandwidth [Gbps]	1.2	1.1	1.2
Core Frequency [MHz]	14	14	25
Memory Controller Frequency [MHz]	50	50	50
Core Supply [V]	0.7	0.7	0.8
Memory Controller Supply [V]	0.84	0.84	0.84
<b>Power [mW]</b>	<b>1.8</b>	<b>1.8</b>	<b>3.2</b>

system power is then dominated by the off-chip frame buffer memory. [7] shows that the memory power using an off-the-shelf DRAM is on the order of 30 mW for QCIF at 15 fps which would scale to *hundreds* of milliwatts for high definition. However, new low power DRAMs such as [30], can deliver 51.2 Gbps at 39 mW. For 720p decoding, the required bandwidth is 1.25 Gbps after memory optimizations in Section VI, which corresponds to a frame buffer power of 1 mW (a linear

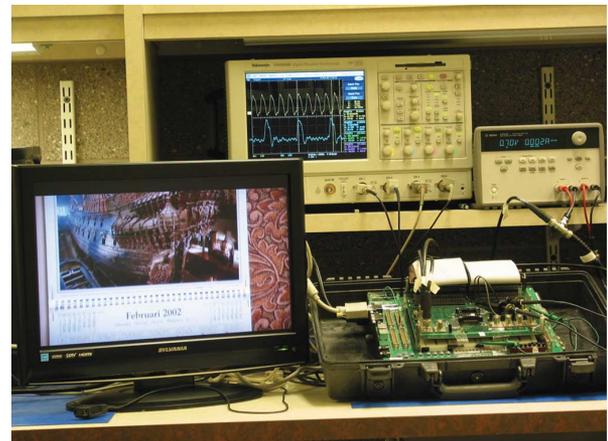


Fig. 15. Test setup for H.264 decoder. Voltage and current measurements of the core domain can be seen in the upper right corner.

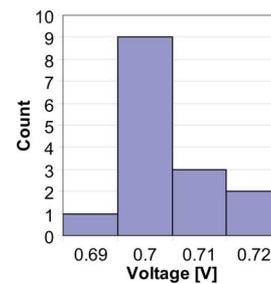


Fig. 16. Minimum core voltage supply variation across test chips.

estimate from [30]). Furthermore, off-chip interconnect power can be reduced by using embedded DRAM or system in package (i.e., stacking the DRAM die on top of the decoder die within a package).

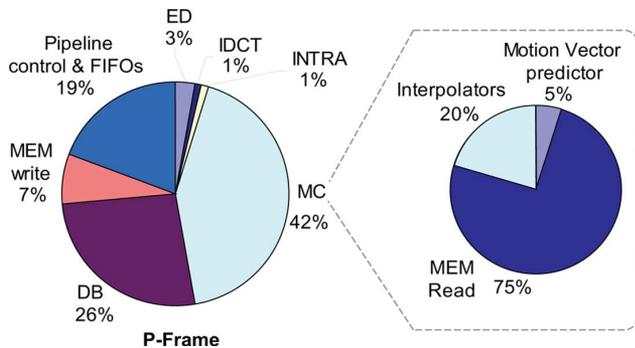


Fig. 17. Simulated (post-layout) power breakdown during P-frame decoding.

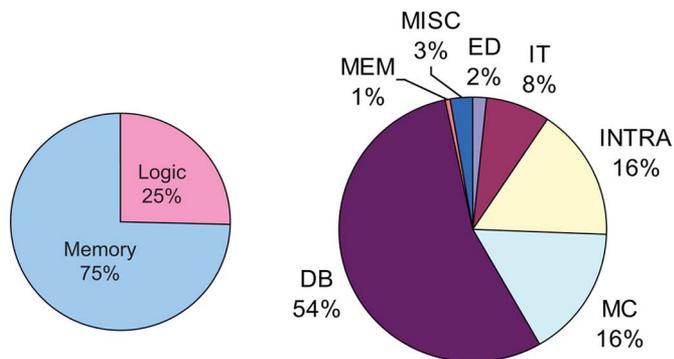


Fig. 18. Post-layout area breakdown (includes logic and memory).

TABLE VIII

LOGIC GATE COUNT (PRE-LAYOUT) FOR EACH DECODER UNIT. NOTE THAT THE AREA OF EACH UNIT ALSO INCLUDES THE PIPELINE FIFO CONTROL FOR THE UNIT. MISC INCLUDES THE TOP-LEVEL CONTROL, PIPELINE FIFOs, SLICE/NAL HEADER PARSING LOGIC, AND ADDERS FOR RECONSTRUCTION

Decoder Unit	Gate Count [kgates (2-input NAND)]
ED	23.7
INTRA	25.2
MC	102.5
IT	41.3
DB	78.3
MEM	6.5
MISC	37.4
<b>Total</b>	<b>314.8</b>

#### A. Power Breakdown

This section shows the simulated power breakdown during P-frame decoding in the “mobcal” sequence. The power of P-frames is dominated by MC (42%) and DB (26%), as seen on the left chart of Fig. 17. About 75% of the MC power, or 32% of total power, is consumed by the MEM read logic, as illustrated by the pie chart on the right of the same figure. The memory controller is the largest power consumer since it runs at a higher voltage than the core domain, its clock tree runs at a higher frequency, and the MC read bandwidth is large (approximately 2 luma pixels are read for every decoded pixel). At 0.7 V, the on-chip caches consume 0.15 mW.

The total leakage of the chip at 0.7 V is 25  $\mu$ W, which is approximately 1% of the 1.8 mW total power for decoding 720p

TABLE IX  
SUMMARY OF LOW POWER TECHNIQUES

Section(s)	Technique(s)	Impact
II, III	Increased concurrency with pipelining and parallelism	Enable Low Voltage and Frequency 0.7 V @ 14 MHz
IV, VI	Multiple Voltage/Frequency Domains	25% Power Reduction on P-frame
V	Frame level DVFS	25% Power Reduction
VI, VII	Memory Optimization	Reduce Bandwidth by 40%

at 30 fps. At 0.5 V, the leakage is 8.6  $\mu$ W, which is approximately 28% of the 29  $\mu$ W total power for decoding QCIF at 15 fps. 64% of the total leakage power is due to the caches. The leakage of the caches could have been reduced by power gating unused banks during QCIF decoding for additional power savings.

#### B. Area Breakdown

The post-layout area breakdown by decoder unit is shown in Fig. 18. The pre-layout logic gate count from synthesis in each decoder unit is reported in Table VIII. The area is dominated by DB due the SRAM caches which dominate the DB area.

The cost of parallelism is primarily an increase in area. The increase in total logic area due to the parallelism in the MC and DB units described in Section III is about 12%. When compared to the entire decoder area (including on-chip memory) the area overhead is less than 3%.

## VIII. SUMMARY

A full video decoder system was implemented that demonstrates high definition real-time decoding while operating at 0.7 V and consuming 1.8 mW. Several techniques, summarized in Table IX, were leveraged to make this low-power decoder possible. The decoder processing units were pipelined and isolated by FIFOs to increase concurrency. Luma and chroma components were mostly processed in parallel. The MC interpolators and DB filters were replicated for increased performance. The decoder was partitioned into multiple voltage/frequency domains to enable lower voltage/frequency operation for some of the processing blocks. The wide operating voltage range of the decoder allowed for effective use of DVFS for additional power reduction. Finally, voltage-scalable on-chip caches helped reduce both on-chip and off-chip memory power.

#### ACKNOWLEDGMENT

The authors are grateful to Arvind, J. Ankorn, M. Budagavi, D. Buss, E. Fleming, J. Hicks, G. Raghavan, A. Wang and M. Zhou for their support and feedback. The authors would also like to thank Y. Koken for her contributions to the test setup.

#### REFERENCES

- [1] Emerging Markets Driving Growth in Worldwide Camera Phone Market. [Online]. Available: <http://www.infotrends.com/public/Content/Press/2008/07.29.2008.html>

- [2] *ITU-T Recommendation H.264 Advanced Video Coding Generic Audiovisual Services*. Joint Video Team, 2003.
- [3] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, pp. 7–28, 2004.
- [4] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.
- [5] C.-D. Chien, C.-C. Lin, Y.-H. Shih, H.-C. Chen, C.-J. Huang, C.-Y. Yu, C.-L. Chen, C.-H. Cheng, and J.-I. Guo, "A252kgate/71 mW multi-standard multi-channel video decoder for high definition video applications," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2007, pp. 282–283.
- [6] S. Na, W. Hwangbo, J. Kim, S. Lee, and C.-M. Kyung, "1.8 mW, hybrid-pipelined H.264/AVC decoder for mobile devices," in *Proc. IEEE Asian Solid State Circuits Conf.*, Nov. 2007, pp. 192–195.
- [7] T.-M. Liu, T.-A. Lin, S.-Z. Wang, W.-P. Lee, K.-C. Hou, J.-Y. Yang, and C.-Y. Lee, "A 125  $\mu$ W, fully scalable MPEG-2 and H.264/AVC video decoder for mobile applications," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 161–169, Jan. 2007.
- [8] C.-C. Lin, J.-W. Chen, H.-C. Chang, Y.-C. Yang, Y.-H. O. Yang, M.-C. Tsai, J.-I. Guo, and J.-S. Wang, "A 160 K gates/4.5 KB SRAM H.264 video decoder for HDTV applications," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 170–182, Jan. 2007.
- [9] D. Finchelstein, V. Sze, M. Sinangil, Y. Koken, and A. Chandrakasan, "A low-power 0.7-V H.264 720p video decoder," in *Proc. IEEE Asian Solid State Circuits Conf.*, Nov. 2008, pp. 173–176.
- [10] T. Fujiyoshi, S. Shiratake, S. Nomura, T. Nishikawa, Y. Kitasho, H. Arakida, Y. Okuda, Y. Tsuboi, M. Hamada, H. Hara, T. Fujita, F. Hatori, T. Shimazawa, K. Yahagi, H. Takeda, M. Murakata, F. Minami, N. Kawabe, T. Kitahara, K. Seta, M. Takahashi, and Y. Oowaki, "An H.264/MPEG-4 audio/visual codec LSI with module-wise dynamic voltage/frequency scaling," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2005, pp. 132–133.
- [11] B.-G. Nam, J. Lee, K. Kim, S. J. Lee, and H.-J. Yoo, "52.4 mW 3-D graphics processor with 141Mvertices/s vertex shader and 3 power domains of dynamic voltage and frequency scaling," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2007, pp. 278–279.
- [12] S. Mochizuki, T. Shibayama, M. Hase, F. Izuhara, K. Akie, M. Nobori, R. Imaoka, H. Ueda, K. Ishikawa, and H. Watanabe, "A 64 mW high picture quality H.264/MPEG-4 video codec IP for HD mobile applications in 90 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 11, pp. 2354–2362, Nov. 2008.
- [13] E. Fleming, C.-C. Lin, N. Dave, Arvind, G. Raghavan, and J. Hicks, "H.264 decoder: A case study in multiple design points," in *Proc. Formal Methods and Models for Co-Design (MEMOCODE)*, Jun. 2008, pp. 165–174.
- [14] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 704–716, Jul. 2003.
- [15] S.-Z. Wang, T.-A. Lin, T.-M. Liu, and C.-Y. Lee, "A new motion compensation design for H.264/AVC decoder," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, vol. 5, pp. 4558–4561.
- [16] K. Xu and C.-S. Choy, "A five-stage pipeline, 204 cycles/MB, single-port SRAM-based deblocking filter for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 363–374, Mar. 2008.
- [17] C. E. Cummings, "Simulation and synthesis techniques for asynchronous FIFO design," [Online]. Available: [www.sunburst-design.com/papers/CummingsSNUG2002SJ\\_FIFO1.pdf](http://www.sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO1.pdf)
- [18] V. Gutnik and A. P. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 5, pp. 425–435, Dec. 1997.
- [19] K. Choi, K. Dantu, W. Cheng, and M. Pedram, "Frame-based dynamic voltage and frequency scaling for a MPEG decoder," in *Proc. IEEE/ACM Int. Conf. Computer Aided Design*, Nov. 2002, pp. 732–737.
- [20] J. Pouwelse, K. Langendoen, R. Lagendijk, and H. Sips, "Power-aware video decoding," in *22nd Picture Coding Symp.*, 2001.
- [21] A. C. Bavier, A. B. Montz, and L. L. Peterson, "Predicting mpeg execution times," in *Proc. ACM SIGMETRICS Joint Int. Conf. Measurement and Modeling of Computer Systems*, New York, NY, 1998, pp. 131–140.
- [22] E. Akyol and M. van der Schaar, "Complexity model based proactive dynamic voltage scaling for video decoding systems," *IEEE Trans. Multimedia*, vol. 9, no. 7, pp. 1475–1492, Nov. 2007.
- [23] C. Im, H. Kim, and S. Ha, "Dynamic voltage scheduling technique for low-power multimedia applications using buffers," in *Proc. IEEE Int. Symp. Low Power Electronics and Design*, Aug. 2001, pp. 34–39.
- [24] W. Kim, M. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *Proc. IEEE Int. Conf. High Performance Computer Architecture*, Feb. 2008, pp. 123–134.
- [25] J. Lee, B.-G. Nam, and H.-J. Yoo, "Dynamic voltage and frequency scaling (DVFS) scheme for multi-domains power management," in *Proc. IEEE Asian Solid State Circuits Conf.*, Nov. 2007, pp. 360–363.
- [26] M. E. Sinangil, N. Verma, and A. P. Chandrakasan, "A reconfigurable 65 nm SRAM achieving voltage scalability from 0.25–1.2 V and performance scalability from 20 kHz–200 MHz," in *Proc. IEEE European Solid State Circuits Conf.*, Sep. 2008, pp. 282–285.
- [27] x264 - A free h264/avc encoder. [Online]. Available: <http://www.videolan.org/developers/x264.html>
- [28] CY7C1470V25 Datasheet. [Online]. Available: [http://download.cypress.com.edgesuite.net/design\\_resources/datasheets/contents/cy7c1470v25\\_8.pdf](http://download.cypress.com.edgesuite.net/design_resources/datasheets/contents/cy7c1470v25_8.pdf)
- [29] 6.111 FPGA Labkit Documentation. [Online]. Available: <http://www-mtl.mit.edu/Courses/6.111/labkit/>
- [30] K. Hardee, F. Jones, D. Butler, M. Parris, M. Mound, H. Calendar, G. Jones, L. Aldrich, C. Gruenschlaeger, M. Miyabayashil, K. Taniguchi, and I. Arakawa, "A 0.6 V 205 MHz 19.5 ns tRC 16 Mb embedded DRAM," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2004, pp. 200–201.



**Vivienne Sze** (S'04) received the B.A.Sc. (Hons) degree in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2004, and the S.M. degree from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 2006, where she is currently a doctoral candidate.

From May 2007 to August 2007, she worked in the DSP Solutions R&D Center at Texas Instruments, Dallas, TX, designing low power algorithms for the next generation video coding standard. From May 2002 to August 2003, she worked at Snowbush Microelectronics, Toronto, ON, Canada, as an IC Design Engineer. Her research interests include low-power circuit and system design, and low-power algorithms for video compression.

Ms. Sze was a recipient of the 2007 DAC/ISSCC Student Design Contest Award and a co-recipient of the 2008 A-SSCC Outstanding Design Award. She received the Natural Sciences and Engineering Research Council of Canada (NSERC) Julie Payette fellowship in 2004, the NSERC Postgraduate Scholarships in 2007, and the Texas Instruments Graduate Woman's Fellowship for Leadership in Microelectronics in 2008.



**Daniel F. Finchelstein** (M'09) received the B.A.Sc. degree from the University of Waterloo, Ontario, Canada, in 2003, and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, in 2009. His doctoral thesis focused on efficient video processing.

He is currently working in the 3D graphics performance group at Nvidia Corporation. His research interests include energy-efficient and high-performance digital circuits and systems. His recent focus has been on processors and memory architectures for video and graphics processing. He has been an engineering intern on eight different occasions at companies such as IBM, ATI, Sun, and Nvidia.

Dr. Finchelstein received student design contest awards at both A-SSCC 2008 and DAC/ISSCC 2006, and has several conference and journal publications. He was awarded the Natural Sciences and Engineering Research Council of Canada (NSERC) graduate scholarship from 2003–2008. He also received the University of Waterloo Sanford Fleming Medal for ranking first in the graduating class. He holds one US patent related to cryptography hardware.



**Mahmut E. Sinangil** (S'06) received the B.Sc. degree in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 2006, and the S.M. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 2008. He is currently pursuing the Ph.D. degree at MIT, where his research interests include low-power digital circuit design in the areas of SRAMs and video coding.

Mr. Sinangil received the Ernst A. Guillemin Thesis Award at MIT for his Master's thesis in 2008 and Bogazici University Faculty of Engineering Special Award in 2006.



**Anantha P. Chandrakasan** (F'04) received the B.S., M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994 respectively.

Since September 1994, he has been with the Massachusetts Institute of Technology, Cambridge, where he is currently the Joseph F. and Nancy P. Keithley Professor of Electrical Engineering. He is also the Director of the MIT Microsystems Technology Laboratories. His research interests

include low-power digital integrated circuit design, wireless microsensors, ultra-wideband radios, and emerging technologies. He is a coauthor of *Low Power Digital CMOS Design* (Kluwer Academic Publishers, 1995), *Digital Integrated Circuits* (Pearson Prentice-Hall, 2003, 2nd edition), and *Sub-threshold Design for Ultra-Low Power Systems* (Springer 2006). He is also a co-editor of *Low Power CMOS Design* (IEEE Press, 1998), *Design of High-Performance Microprocessor Circuits* (IEEE Press, 2000), and *Leakage in Nanometer CMOS Technologies* (Springer, 2005).

Dr. Chandrakasan was a co-recipient of several awards including the 1993 IEEE Communications Society's Best Tutorial Paper Award, the IEEE Electron Devices Society's 1997 Paul Rappaport Award for the Best Paper in an EDS publication during 1997, the 1999 DAC Design Contest Award, the 2004 DAC/ISSCC Student Design Contest Award, the 2007 ISSCC Beatrice Winner Award for Editorial Excellence and the 2007 ISSCC Jack Kilby Award for Outstanding Student Paper. He has served as a technical program co-chair for the 1997 International Symposium on Low Power Electronics and Design (ISLPED), VLSI Design'98, and the 1998 IEEE Workshop on Signal Processing Systems. He was the Signal Processing Sub-committee Chair for ISSCC 1999–2001, the Program Vice-Chair for ISSCC 2002, the Program Chair for ISSCC 2003, and the Technology Directions Sub-committee Chair for ISSCC 2004–2009. He was an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS from 1998 to 2001. He served on SSCS AdCom from 2000 to 2007 and he was the meetings committee chair from 2004 to 2007. He is the Conference Chair for ISSCC 2010.