Design Methodologies for consumer-use video signal processing LSIs

Hisakazu Edamatsu, Satoshi Ikawa, Katsuya Hasegawa
Semiconductor Research Center,
Matsushita Electric Industrial, Co., Ltd.
Moriguchi, Osaka 570, Japan
{edamatsu, ikawa, katsuya}@vdrl.src.mei.co.jp

Abstract

This paper describes the methodologies used to design a Hi-Vision MUSE decoder for Japanese HDTV and codec LSIs for digital VCRs. Since a large amount of input video data is needed to verify the algorithms and logic designs, reducing the verification time is a key issue in designing these LSIs. We describe the methodology used to verify the video signal processing algorithm and that of the physical design.

I. Introduction

Advances in digital signal processing for consumer products necessitates a change in the design methodology for the LSIs used in these products. The design requirements are the following: verification of the signal processing algorithms and real time functional verification. In addition, the chip must be low power and have a short design cycle.

In this paper, we discuss two rather different types of design methodologies for consumer-use video signal processing LSIs. One is a data-path oriented design methodology used for MUSE decoder LSIs and another is a standard cell based top-down design methodology used for digital VCR codec LSIs.

For the MUSE decoder design, we adopted a mixture of C, Verilog-HDL, and an in-house data-path design system which is comprised of a signal-flow graph level verification tool, SPANA (Signal Processing ANAlyzer), and data path oriented cells. For the digital VCR codec design, RTL descriptions are written in C and translated into Verilog RTL with an in-house translation tool. We also cover the low-power design aspects of this methodology.

33rd Design Automation Conference®

DAC 96 - 06/96 Las Vegas, NV, USA
©1996 ACM 0-89791-779-0/96/0006..$3.50

II. Overview of the MUSE Decoder

Video signal processing of the MUSE decoder system is based on digital filters. Fig. 1 shows a block diagram of the MUSE decoder system. The MUSE system has a long history, and the standard is defined as BTA S-001 in 1987, and an 8 hours a day broadcasting began in November 1991. During this progress of the MUSE system, many companies have joined the development of MUSE chip set. As a result, it was difficult to have consistency over entire MUSE system design. In this design, we intended to generate functional model of the entire MUSE system on SPANA and to facilitate the improvement of the system for the next generation.

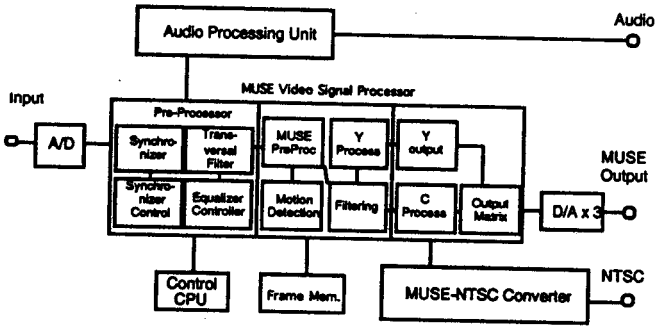

Figure 1: Block Diagram of the MUSE Decoder

III. Design Flow Overview

The main video signal processing block is realized with about 220k logic gates and 250kb of RAMs. Besides these, A/D and D/A converters, 16Mb of DRAM frame memory, an audio signal processing unit, a 16 bit micro-controller, and a MUSE-NTSC converter comprise the system. Fig. 2 shows the design flow of this design. In this design, one of the key points is the verification of video signal processing algorithm using an algorithm level programing of C in the system design and a RTL level modeling of SPANA in the data path functional verification. Another is the real-time system verification using prototype system.

Since the maximum operating frequency of this system is 44.55 MHz, it is difficult to develop real-time prototype

using FPGAs. Hence, bread-board prototyping using conventional components are adopted. In order to minimize design errors in transferring prototype design data into the LSI design, the video signal data path is first verified using a C model with floating-point data without care for the bit width of the data path. After that, bit-true hardware algorithm verification is done using an in-house signal-flow graph level verification tool, SPANA. Next, the prototype board is developed using the SPANA design data for the real time system verification. This prototype board is used to check noise immunity and synchronization capability by receiving an experimental broadcasting signal. This kind of real time verification is difficult using even the fastest simulation or emulation.

Control blocks are verified by RTL description of Verilog HDL and the logic is synthesized using Synopsys' Design Compiler. For the logic level verification, the data of SPANA, which is integrated on Mentor Graphics' Falcon Framework, is translated into Mentor Graphics Quick Sim II netlist and the control block netlist is also translated into a Quick Sim II netlist. Then, chip level logic simulation is then performed.

For the physical design of the chip, the SPANA netlist data is transferred to an in-house data path layout tool and the generated layout is treated as hard macro block in the following block routing stage. Control block layout is performed by an in-house block layout tool. Chip level layout is performed by Cadence's Block Ensemble. This chip set using this methodology was first implemented in 0.8μm double metal CMOS and the following generation was implemented in 0.5μm double metal CMOS.

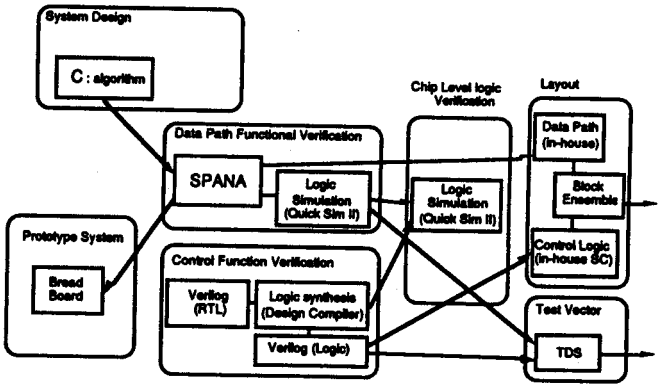

Figure 2: MUSE Decoder Design Flow

IV. Video Signal Verification

Verification of the video signal processing algorithm requires the comparison of simulated output against expected data or simulated data in the different design stages. There are two major concerns in the verification: ambiguity and bitwise correctness. Ambiguity arises from the hardware optimizations made by considering the sensitivity of human vision. Because of this, exact bit-by-bit implementation is not necessarily required in the system design phase. Hence a floating-point value is used to model the video signal and the floating-point value is optimized so that an image of a good quality appears on the CRT screen. On the other hand, bitwise correctness is required in the logic design stage and thereafter in order to compare the simulation result automatically. We used a design flow where a magunitude of difference of simulation result is easily observed.

In the system level algorithm verification, the simulated image data is verified by human inspection on the CRT monitor display and an optimal processing algorithm is developed. We compared the data pixel-by-pixel against the simulated results in subsequent design stages by considering the ambiguity handling so that the magnitude of the difference between the two pixels is easily observed. Since the amount of data is large, we mapped the magnitude of the pixel-by-pixel data difference to color brightness and displayed the results on a CRT monitor. In this case, difference of ± 4, (+3,+2), +1, 0, -1, (-2,-3) bits are mapped to red, green, cyan, simulation result in gray scale, yellow and violet respectively: this facilitates the detection of errors. Fig. 3 illustrates the verification method. First, algorithm verification is performed by feeding actual image data into a bread-board prototype and a C based algorithm simulator.

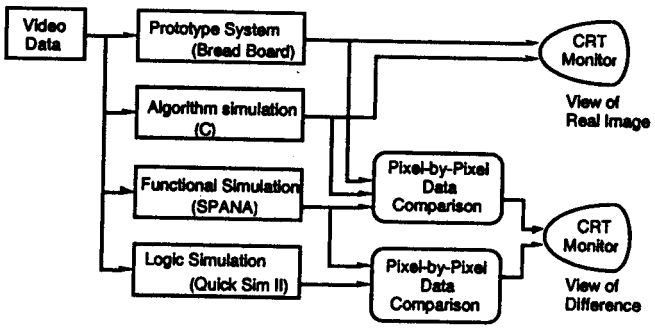

Figure 3: Image Data Comparison

Next, the image data is fed to the SPANA-based functional simulator where bit-true hardware verification is done as well as hardware optimization in terms of hardware size and data accuracy. In this stage, the result of SPANA simulation is compared with the C simulation by displaying the difference of two simulation results on a CRT monitor screen. A similar method is applied to the verification of the functional and logic simulations. Because logic simulation is very time consuming, only a small portion of the actual image data is simulated in this stage.

This method enables quick comprehension of the quality of signal processing algorithm and its implementation in hardware. All algorithmic errors have been found during the SPANA simulation stage in this design.

V. Data Path Design

SPANA is an in-house signal-flow graph level cycle- based simulator. After algorithm verification in C using actual image data, bit-true hardware is designed and modeled with this simulator. Fig. 4 shows a design entry example. The datapath schematic diagram is composed of adders, multipliers, multiplexers, registers, rounders, clippers, and so on. Each of the datapath cells has two simulation models. One is for SPANA and the another is for Quick Sim II. So, schematic data written using these cells can be simulated by both SPANA and Quick Sim II without translation. The multiple-bit datapath is handled as a bus, and the bundled signal of SPANA schematic data can be expanded into a Design Architect netlist. SPANA analyses frequency responses using its analysis tool kit which includes signal generators, an oscilloscope, and a FFT analyzer.

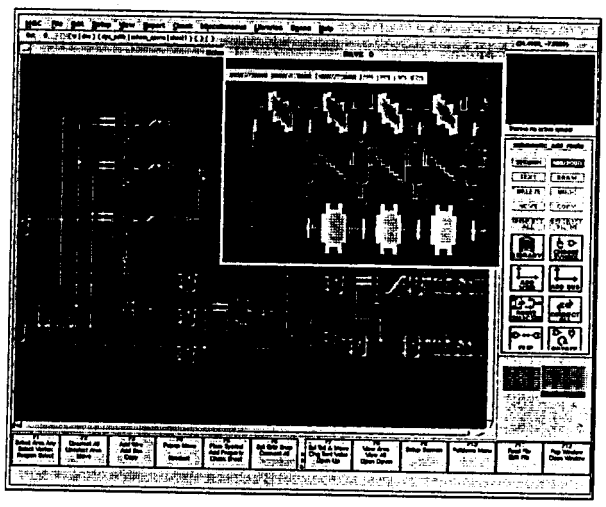

Figure 4: Functional Simulator: SPANA

SPANA can also automatically generate cycle-based simulation programs written in C from the macro-level datapath schematic for simulation of bit-true hardware. For a 220k gate design, this simulator performs about 70 cycles per second which is about 20 times faster than a RTL model simulated in an event driven Verilog HDL simulator and about 200 times faster than a Verilog gate-level simulation. Thus, we could verify the bit-true datapath design using the same large amount of image data that we used to verify the floating point algorithm.

After the hardware verification using SPANA, the netlist is converted to gate level and transferred to a proprietary in-house datapath layout tool.

SPANA also has the capability to output Verilog HDL which facilitates the interface to top-down design flow using standard cell layout.

Fig. 5 shows the datapath cell structure, which enables high density layout design. Each cell has input ports on the left side, output ports on the right side, carry-in and scan-in on the bottom side, and carry-out and scan-out on the top side. Because of this port arrangement, layout of the signal flow-graph-based block can be efficiently realized with these cells by placing them side by side according to the flow of the data. Feed-through channels in vertical and horizontal directions are also implemented in each cell.

Block level data-path layout is performed using a proprietary in-house datapath block layout tool and bit-wise datapath layout is generated according to the SPANA data.

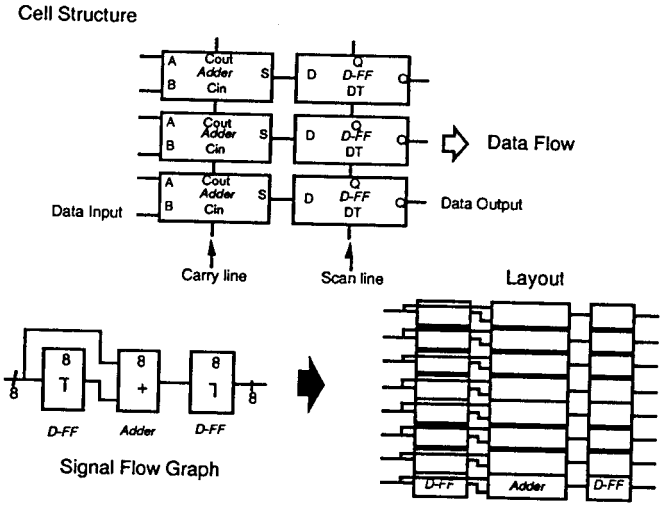

Figure 5: Physical Design using Datapath Cells

VI. Design Time Efficiency using SPANA

Although the functional simulation stage by SPANA has been used for years, the translation of design data to different design stages needed to be done by hand. This translation step has been automated in order to save logic verification time. Hence by performing hardware verification extensively on SPANA, time-consuming logic simulation is almost eliminated.

Without this stage, hardware verification must be performed on a bread-board, or at the Verilog RTL level which about 20 times slower than SPANA.

The SPANA simulation of the MUSE datapath using large amounts of image data was performed twelve times. Tab. 1 shows the simulation statistics including test vector size, SPANA simulation time and estimated Verilog RTL

simulation time which is estimated from the time needed to simulate part of the Verilog RTL description of the MUSE datapath using a small set of image data.

Total actual simulation time was 354 hours (15 days) using SPANA, and estimated to be 344 days for Verilog RTL simulation. In SPANA, still picture simulation can be performed in about one day and motion picture simulation of 20 frames can be performed in 4 days. This simulation speed enables us to verify the design using sufficiently large image data in order to get a reliable functional design. This, in turn, contributes to saving time in the later design stages. In addition, by using common cell libraries on SPANA and Quick Sim II, the logic synthesis stage from RTL level HDL, that is usually inevitable in top-down design, could be eliminated.

TestNo.	Vector	#ofFrame	SPANA*	Verilog	ratio
1	6.5M	6.0	25:33'09"	594h	23.2
2	4.3M	4.0	19:11'03"	396h	20.6
3	6.5M	6.0	24:52'05"	594h	23.9
4	7.6M	7.0	28:27'59"	693h	24.3
5	7.6M	7.0	29:46'32"	693h	23.3
6	4.3M	4.0	15:45'46"	396h	25.1
7	4.3M	4.0	16:00'28"	396h	24.7
8	21.6M	20.0	94:42'01"	1980h	20.9
9	7.0M	6.5	23:49'09"	644h	27.0
10	7.6M	7.0	24:55'12"	693h	27.8
11	6.5M	6.0	26:37'15"	594h	22.3
12	6.5M	6.0	24:04'39"	594h	24.7
total	90.2M	83.5	353:45'18"	8267h	23.4

* executed of on HP9000/755(99MHz)
(Verilog RTL simulation time is an estimation)

Table 1: Simulation Time

For a LSI of similar size and complexity, it takes about 14 man-months to synthesize logic from RTL. Since this step was eliminated, we estimate that by using our design methodology, this amount of design time was saved.

VII. Overview of the DVCR Codec Chip Set

The digital VCR (DVCR) is a new consumer electronic product introduced in the market recently. Since a low power battery operation is essential for a camcorder which must include an encoder, a complex data compression scheme such as MPEG2 is not suitable for DVCRs despite its popularity as a high quality image compression algorithm. A new intra-frame compression format was defined by ten major electronic companies in 1993 and the first commercial products were shipped in Japan in 1995. In this design, algorithm verification is a key point.

Fig. 6 shows a block diagram of the DVCR. The major video signal processing circuits are divided into two units. The first unit, the camera signal processing unit, generates a component digital video signal from the captured image. It consists of two major LSIs. The other unit, the DVCR signal processing unit, which includes five major LSIs, encodes and decodes the digital audio/video signal between the DVCR format and video signal.

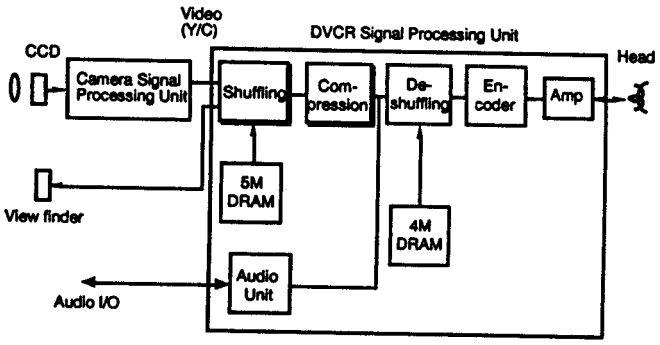

Figure 6: Block Diagram of Digital VCR

VIII. Design Methodology Constraints

The design methodologies of the LSIs in each unit differ from one another. The two LSIs in the camera signal processing unit are designed using the same datapath design methodology as the MUSE decoder described above. However, all five chips in the DVCR signal processing unit are designed using a top-down design methodology described below. This difference in design methodology is mainly due to the nature of the signal processing performed in each unit. Since the processing in the camera signal processing unit is primarily done with digital filters, it is suitable to use a datapath oriented design methodology. On the other hand, since the digital processing in the DVCR signal processing unit is random logic based, straightforward top-down design is suitable for designing these chips. The other factor for deciding the design methodology is the history of the signal format itself. Since the DVCR video compression format was defined recently, there is no need to consider using the previous generation's design.

In the following sections, we will focus on the design methodology of the two video codec chips. The two chips which we call the Shuffling LSI and the Compression LSI are among the five LSIs in the DVCR signal processing unit and perform video compression/decompression by shuffling, DCT (Discrete Cosine Transform), quantization and VLC (Variable Length Coding). Two chips described here are

fabricated in two-layer metal 0.5μm CMOS technology and contain about 1M transistors in total. Three external clocks, that have a maximum frequency of 27 MHz are used.

IX. Top-down Design

Fig. 7 shows the design methodology of the DVCR video codec chip set. In this design, most of the RTL functional verification is done in C and converted to Verilog-HDL RTL code with an in-house translation tool. The RTL code is then synthesized to a gate level standard cell netlist.

As in the case of MUSE decoder, extensive algorithm level verification is needed before the actual RTL design of the LSIs. In addition to the algorithm level simulator, an FPGA-based prototype is used to verify the detailed specifications of the chip set. Although the image compression algorithm is a standard, the details are left up to each implementation. When determining the specific circuit configurations and circuit parameters, there is usually a tradeoff between circuit size and picture quality. Some of the picture quality evaluation is rather subjective and needs to be done with the human eye using hours of pictures. This is not possible with the simulator.

RTL functional verification with C is key to reducing the design time since an exhaustive simulation is needed to verify the highly data dependent operations of the data compression units.

Also using C is effective for verifying the trick-play modes which need an input of several image frames. The C based RTL simulation is five to ten times faster than Verilog RTL simulations. Output of the RTL functional simulation is compared pixel by pixel with the output from the algorithm level simulator in order to verify the RTL design.

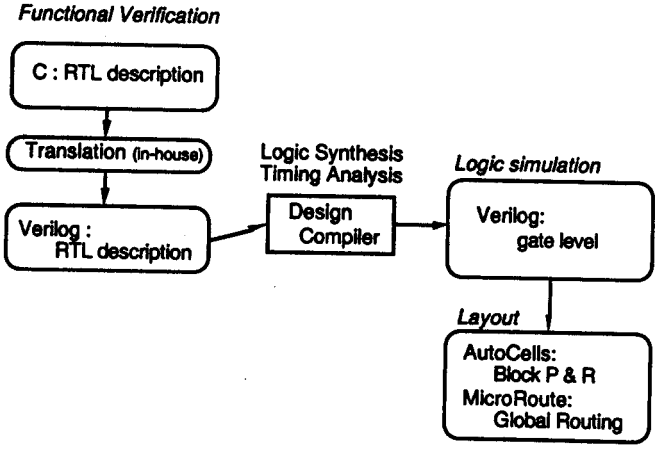

Figure 7: Design Flow of Digital VCR

Since the in-house C to Verilog translation tool is a simple source level line-to-line interpreter, not all of the Verilog language syntax can be implemented in a C RTL description. Hence, some of the functional units of the chip set are written directly in Verilog RTL in order to minimize the circuit size after logic synthesis.

We used Synopsys' Design Compiler for the logic synthesis and for the static timing analysis. Since the standard cell library is designed for low power operation, not only circuit size and speed, but also power consumption are considered as optimization factors in the logic synthesis process. We will discuss this feature in the next section. Most of logic is designed using synchronous circuits and the FFs are replaced with scan FFs after the logic synthesis in order to implement a full-scan test chain. The boundary scan test (IEEE 1149.1) is included to increase the testability at the board level.

Physical design is based on standard logic cell libraries with macrocells. Macrocells consist of on-chip memories and multipliers which are generated from the module generator. We used AutoCells and MicroRoute from Mentor Graphics' for the standard cell and block-level layout, floorplanning and inter-block global routing.

X. Low Power Design

Low power design consists of two approaches. One is low voltage operation and another is power management design using multiple clocks.

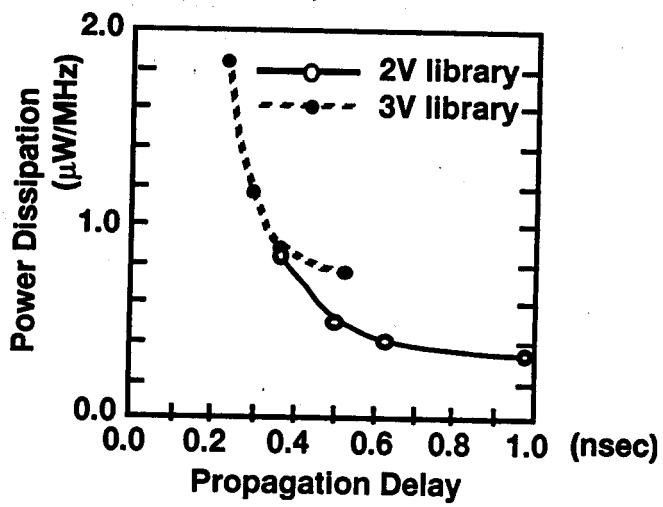

Figure 8: Low power Design using Low Voltage Operation

In this design, we adopted low voltage of 2 V, using 0.5μm CMOS. To interface the internal circuits to peripheral chips operating at 3V, each I/O pad includes a level conversion circuit. The layout of the standard cell is carefully designed to use minimum transistor sizes so that the power

and the area could be optimized in logic synthesis. Fig. 8 shows the propagation delay time and the power dissipation of a 2-input NAND gate with typical load conditions. Much of the cell layout is the same as 3 V standard cell and cell delay parameters are extracted for 2 V. Another point of the low power design is power management. There are six clocks in the chip set which are generated from three external clocks (13.5, 18, 27 MHz). Four of the six clocks are gated clocks and are used to control the power-down mode according to the operation mode of the camcorder such as recording, playback, still playback, cue, review, etc. For example, when the recording is paused, the compression units are powered down while the data path from the digital video input to the view finder of the video camera remains activated.

Signal flow between blocks operationg in different clock frequencies is not automatically handled in the top-down design methodology. In this design, blocks of different clock frequencies are connected through FIFO and design verification is performed for each block using one kind of clock frequency.

XI. Conclusions

Fig. 9 shows the verification methods described here in terms of abstraction level and simulation speed. HDL based modeling is still far from near-real time verification. Hence real-time verification still requires hardware prototyping as well as C modeling which heavily depends on the skill of engineers and lacks a standardized interface to other design systems.

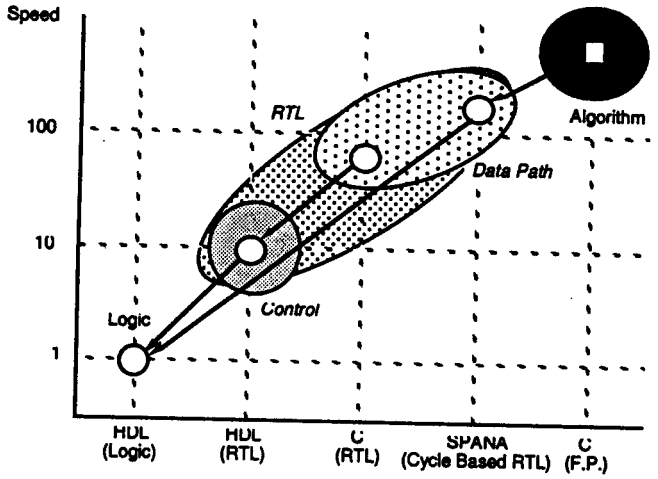

Figure 9: Verification Method Comparison

As the system size grows, demands for economical near-real time verification methodology will become more significant when designing future digital video signal processing chips.

Acknowledgments

The authors would like to thank CAD groups of Semiconductor Research Center and Matsushita Electronic Corporation for realizing these designs.

References

[1] H. Yasoshima et al., "An Approach to Design Verification for Video Signal Processing LSIs", Technical Report of IEICE, ICD93-118, pp. 45-50, Oct., 1993.

[2] S. Ikawa et al., "Functional Design Support System ARCH for Digital Signal Processing LSI", Proc. IEICE, p. 1-112, Mar., 1990.

[3] K. Hasegawa et al., "Low Power Video Encoder/Decoder Chip Set for Digital VCRs", ISSCC Digest of Technical Papers, Feb., 1996.