Quality-Driven Design by Bitwidth Optimization for Video Applications

Yun Cao and Hiroto Yasuura

Kyushu University, Japan
(cao, yasuura)@c.csce.kyushu-u.ac.jp

Abstract— This paper presents a novel system-level design methodology, called quality-driven design for video applications by bitwidth optimization (QDDV). An output quality adaptive approach based on a forward and a backward propagation technique, which are effective bitwidth analysis and output-quality-based bitwidth analysis are also presented for variable bitwidth optimization. In order to illustrate the potential of the proposed methodology, MPEG-2 video is used as the driver application. Experimental results show the effectiveness of the methodology.

I. Introduction

With increasing design complexity, different techniques have been proposed to bridge the gap between manufacturing capability and design productivity. Moving to higher levels of abstraction is one of the promising techniques. Recently, C-based design is an active area. [1] proposed a methodology to integrate a C-based synthesis flow into the overall ASIC design flow. Bitwidth of variables has drastic effects on hardware [2] [3] and software [4] [5] [6]. However, when design an algorithm for example using C language, programmers often pay little attention to some details such as the bitwidth of data-types used in their programs. For instance, in a C program, it is common to use a 32-bit integer data-type to represent a single Boolean variable. Therefore, in order to reduce the unused bits, some approaches for variable bitwidth analysis are proposed [7] [8] [3]. A number of previous efforts have been made to exploit bitwidth. The C language has been augmented to provide additional bitwidth information in the work on Valen-C [4]. [5] presented bitwidth cognizant architecture synthesis of custom hardware from C language and reported 49% mean reduction in total gate count. [8] discussed bitwidth analysis with application to silicon compilation and reduced silicon real estate by 15-86%, improved clock speed by 3-249%, and reduced power by 46-73%. [9] presented datapath optimization for power minimization and [10] presented low-energy memory design considering bitwidth. [6] discussed dynamically exploiting narrow width operands to improve processor power and performance. [11] combined bitwidth optimization and high-level synthesis of digital signal processing systems to minimize hardware costs such as area, clock period, latency, and power consumption.

This paper focuses on variable bitwidth optimization for video applications because we think that there is a rich opportunity for bitwidth reduction in modern multimedia and streaming applications such as [12]. Different with the researches above, an output quality adaptive approach by variable bitwidth optimization is proposed, which is called quality-driven design for video applications (QDDV). Under output quality of video, a forward and a backward propagation technique called effective bitwidth analysis and output-quality-based bitwidth analysis are introduced. Video quality measurement is discussed. MPEG-2 video is used as the driver application. Experimental results show the effectiveness of the proposed methodology.

The rest of this paper is organized as follows. Next section II presents variable bitwidth analysis including effective bitwidth analysis and output-quality-based bitwidth analysis. Section III presents quality-driven design for video applications by variable bitwidth optimization. Experiments and results are reported in Section IV. Finally, Section V concludes our work.

II. Variable Bitwidth Analysis

Variable bitwidth analysis is a key technique for bitwidth optimization. This section presents a forward and a backward propagation technique, which are effective bitwidth analysis and output-quality-based bitwidth analysis.

The definition of the variable bitwidth analysis is as follows: for given inputs of a system and requirements of output quality, to determine the smallest bitwidth of each variable. We analyze bitwidth of variables in two ways shown in Fig.1. When the requirements of output quality are very strict, we have to find the bitwidth of variables without loss of accuracy in the computation. This is called effective bitwidth analysis. 2) When the requirements of output quality are not so strict for example in video or audio processing, the data themselves are compressed and decoded, they have lost some computation precision. Therefore, for these kinds of applications, we can lose

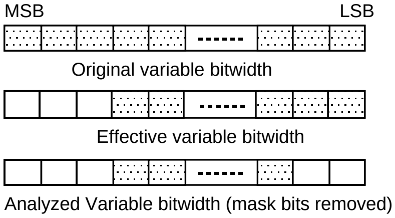
MSB
LSB
Original variable bitwidth
Effective variable bitwidth
Analyzed Variable bitwidth (mask bits removed)

Fig. 1. Variable bitwidth analysis

some computation precision to analyze the bitwidth of variables under the consideration of tradeoffs among performance, power/energy, and cost. This is called output-quality-based bitwidth analysis.

A. Effective Bitwidth Analysis

This part only explains the basic rules for analyzing effective bitwidth of variables in C programs. The details are described in [7]. *Effective bitwidth* is defined as the smallest size which can hold both maximum and minimum values of a variable. In many cases, some bits of a variable are never used during execution of a program. Therefore, by analyzing effective bitwidth of variables we can reduce unused bits to reduce power consumption and cost. If a variable x is unsigned integer type, of which value is in range [0, 2000], then the number of necessary bits of x is 11, because the 11-bit size is enough to hold any value in [0, 2000]. We discuss practical methods of effective bitwidth analysis in combination of a static approach and simulation-based dynamic approach.

For static analysis, when the maximum value of an unsigned integer variable x is n_{max}, the effective bitwidth of x, $e(x)$, is given as follows:

$$e(x) \quad = \quad \lceil \log_2(n_{max} + 1) \rceil \tag{1}$$

For a signed integer x with a maximum value n_{max} and a minimum value n_{min}, the effective bitwidth of x, $e(x)$ is defined as follows:

$$e(x) \quad = \quad \lceil \log_2 \mathcal{N} \rceil + 1 \tag{2}$$

where

$$\mathcal{N} \quad = \quad \max(|n_{max}| + 1, |n_{min}|) \tag{3}$$

Static analysis uses initial bounds on the bitwidth for specific variables and iterative constraint propagation to identify adequate bitwidth for all variables. It is an efficient method to analyze the effective bitwidth of variables. However, in many cases, we can not predict the assigned value of a variable unless we execute the program, such as the case of unbounded loops, static analysis becomes insufficient. As a solution to this problem, we use dynamic analysis. In dynamic analysis, we execute the program and monitor the values assigned to each variable, and then analyze the required bitwidth of the variable.

F_0	:	a formula to analyze
x	:	a variable on the left of F_0
F_n	:	the nth formula from F_0
x_{F_n}	:	the x with minimal LBMW(x)
v_{F_n}	:	the variable on the left of F_n
$LBMW(x)$	:	the removable low bitwidth
$N_{Out}(x)$	:	the mask bitwidth

```
n = 1;
while ($F_n$ exists) {
    if ($LBMW(x) > LBMW(x_{F_n})$)
    $LBMW(x) = LBMW(x_{F_n})$;
    n = n + 1;
    if ($x = v_{F_n}$) {
        if ($LBMW(x) > LBMW(x_{F_n})$) {
            $LBMW(x) = LBMW(x_{F_n})$;
        }
    }
}
$LBMW(x) = N_{Out}(x)$;
```

Fig. 2. Pseudo code for mask bitwidth analysis

B. Output-Quality-Based Bitwidth Analysis

By considering characteristics of each application and output quality constraints, we propose a backward propagation technique called output-quality-based bitwidth analysis. We analyze the variable bitwidth more tightly to further design optimization. *Mask bits* are defined as the low bits of variables, which can be removed, while do not affect the designated computation precision.

To analyze mask bitwidth, we make following assumptions for an application program:

- the effective bitwidth of a variable has been given
- the accuracy of output quality is designated for each basic function block
- no floating computation, no pointer computation

For a variable x, we make following definition:
$e(x)$: effective bitwidth
$n(x)$: mask bitwidth

Arithmetic Operations

Addition, subtraction: $z = x + y$, $z = x - y$

$$\begin{aligned} n(x) &= n(z) - 1 \\ n(y) &= n(z) - 1 \end{aligned}$$

Multiplication: $z = x \times y$

$$\begin{aligned} n(x) &= n(z) - e(y) + 1 \\ &\quad (\text{ if } n(x) < 0 \text{ then } n(x) = 0) \\ n(y) &= n(z) - e(x) + 1 \\ &\quad (\text{ if } n(y) < 0 \text{ then } n(y) = 0) \end{aligned}$$

Logical Operations

AND, OR, XOR, NOR

Because they are bit operation, so no change in mask bitwidth.

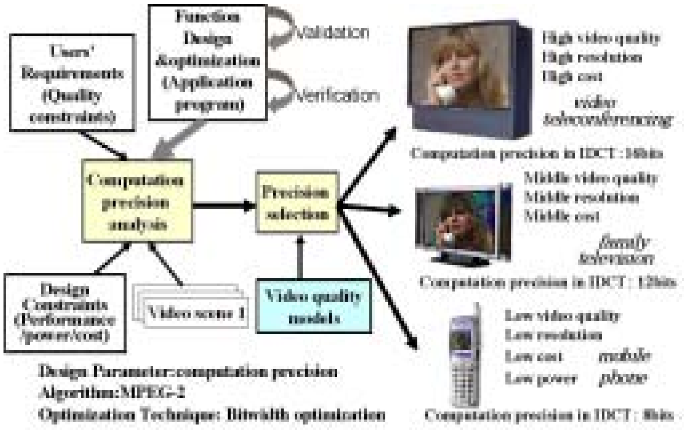
Users' Requirements (Quality constraints)
Function Design &optimization (Application program)
Validation
Verification
Computation precision analysis
Precision selection
Design Constraints (Performance /power/cost)
Video scene 1
Video quality models
Design Parameter:computation precision
Algorithm:MPEG-2
Optimization Technique: Bitwidth optimization
High video quality
High resolution
High cost
video teleconferencing
Computation precision in IDCT : 16bits
Middle video quality
Middle resolution
Middle cost
family television
Computation precision in IDCT : 12bits
Low video quality
Low resolution
Low cost
Low power
mobile phone
Computation precision in IDCT : 8bits

Fig. 3. A flow of quality-driven design for video applications (QDDV)

$$\begin{aligned} n(x) &= n(z) \\ n(y) &= n(z) \end{aligned}$$

Left shift $z = x << y$

$$\begin{aligned} n(x) &= n(z) - y_{max} \\ n(y) &= 0 \end{aligned}$$

Right shift $z = x >> y$

$$\begin{aligned} n(x) &= n(z) + y_{min} \\ n(y) &= 0 \end{aligned}$$

Fig.2 shows the pseudo code for mask bitwidth $LBMW(x)$ analysis. The basic rule to analyze the mask bitwidth of variables in a basic block is to start from the end of the basic block to the head of the block. For a conditional statement, the *then* and the *else* parts are analyzed separately. If both of the two branches have the results of the same variable, the less one will be used.

III. QDDV by Bitwidth Optimization

An essential concept of quality-driven design for video applications (QDDV) is the "quality constraint" in design process, which allows new design space exploration. For digital system design, the bitwidth of data computed in a system is one of the most important design parameters related to performance, power and cost of the system [8]. Datapath width and size of memories strongly depend on the bitwidth of the data. By controlling datapath width, [13] reduced cost and [9] reduced power consumption drastically. We think that by choosing the computation precision really required for each application, further optimization can be achieved for application-specific design. In video processing, for instance, the required quality of video, such as resolution and levels of color, strongly depend on the characteristics of output display devices. We can reduce the computation precision in a target application program, if the reduction does not induce decrease of output quality. It means that we can design a video system with the minimum hardware and energy consumption by eliminating redundant computation. We call the design methodology, quality-driven design by bitwidth optimization for video applications (QDDV).

TABLE I
MOS grading scale

Scale	Impairment
5	Imperceptible
4	perceptible, but not annoying
3	slightly annoying
2	annoying
1	very annoying

TABLE II
MPEG-2 test bitstreams

Bitstream name (Abbreviation)	resolution (pixels*lines)	bit rate (Mbit/sec)
flwr015	352*240	1.5
flwr400	704*480	40
mobl015	352*240	1.5
mobl400	704*480	40
susi015	352*240	1.5
susi400	704*480	40
sonyct1	352*224	1.5
sonyct2	704*480	40
tek3	512*512	3.5

A. Basic Approach

Fig.3 shows the flow of our proposed QDDV. In the first phase of a system design, we concentrate on the implementation of the functionality of the system and optimization for general constraints, performance/power/cost. Initial designs are written in high-level language, such as C language in which most variables are declared as integer-type, which is assumed to be 32 bits. After the function design is validated and verified, we will enter the second phase for application-specific optimization. In this phase, we analyze bitwidth of each variable in the application program, tune the design parameter, computation precision to adapt to the output quality, and do bitwidth optimization under the given quality constraint.

Fig.3 shows three different cases of video applications with the same functionality (MPEG-2 video decoder). In case 1, for the video conferencing with big display, so it needs good video quality (computation precision of IDCT is 16bits). In case 2, for family television, however, needs medium video quality (computation precision of IDCT is 12bits). Case 3 for mobile phone, needs comparatively low video quality (computation precision of IDCT is 8bits). If the three applications use same computation precision, it means that the computation precision must meet the requirement of video conferencing with big output display, which is "redundant" for family TV and mobile phone, this will result in redundant performance/power/cost. Therefore, QDDV can further optimization for system design by application-specific optimization.

B. Measurement of Video Quality

In order to bring quality-driven design into effect, quality measurement is an important issue [15]. We use well-known

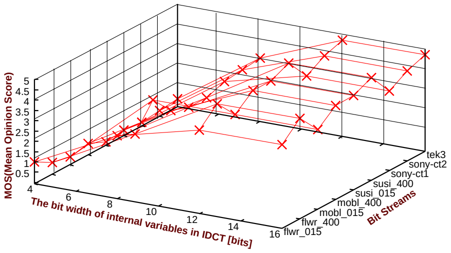

Fig. 4. MOSs of bitstreams for different internal variable bitwidth in IDCT

two measures, the subjective measure Mean Opinion Scale (MOS) shown in table I and objective measure peak signal-to-noise ratio (PSNR).

We measure the video quality of MPEG-2 video system by subjective and objective metric, which focuses on the effects of computation precision. We change the computation-precision of IDCT, the kernel program of MPEG-2 video decoder and measure video quality by some experiments. The test bitstreams shown in table II are used in our experiments.

B.1 Subjective Measure

The subjective assessment of video quality has drawn attention of a number of researchers for many years, principally in relation to evaluation of new transmission or coding schemes, and in the development of advanced television standards. The standardization committees of the ISO, and in particular the CCIR, have published recommendations on the assessment of picture quality in television [14].

In our work, we follow closely the CCIR 500 recommendations with respect to subjective scales and experimental conditions. Ten observers participated in the experiments. We make use of the numerical scores (5-point (MOS) impairment scale) associated with the impairment descriptors for the computation of average MOS scores.

Because the subjective perception of noise and the behavior of MPEG-2 systems are influenced by scene attributes such as spatial detail, amount and complexity of motion, brightness, and contrast, test scenes that spanned a range of these attributes are selected.

In the experiments, the observers are asked to assign a score A(i,k) to each test bitstream, where A(i,k) is the score given by the i_{th} observer to test bitstream k. The scores are averaged to obtain the MOS value for a specific image.

$$MOS(k) = \frac{1}{n}\sum_{i=1}^{n} A(i,k) \tag{4}$$

where n denotes the number of observers.

We changed the computation precision of IDCT, and got the results of Fig.4, which shows the relationship among the internal variable bitwidth in IDCT, MOSs and clips. Here, each point in the graph represents the average quality of a MPEG-2 video decoder system, which was obtained by averaging the subjective MOSs.

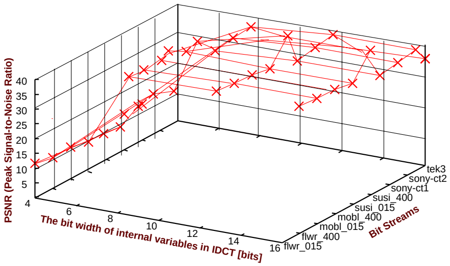

Fig. 5. PSNRs of bitstreams for different internal variable bitwidth in IDCT

Subjective assessment tests are widely used to evaluate the picture quality of coded images, but careful subjective assessments of quality are experimentally difficult and lengthy, and the results obtained may vary depending on the test conditions. Further more, subjective assessments provide no constructive methods for performance improvement, and are difficult to use as part of the design process. However, the most fundamental quality measures for digital video are the subjective responses of human viewers to delivered images and subjective tests remain the only viable reference point for validating objective measures.

B.2 Objective Measure

Objective measures of picture quality can make the comparison of coded images, and also have the possibility of successive adjustments to improve or optimize the picture quality for a desired quality of service. The objective assessment of performance both with respect to bit rate and image quality would also lead to a more systematic design of video systems. An objective model that produces overall quality estimates would have to account for application-specific effects. The influence on accuracy of measurement is the changing expectations of people over time. For these reasons, objective video quality measurement is valid only if the application and viewer population are well defined.

PSNR is often used to specify the signal-to-noise ratio of a video signal. This method has the advantage of removing the signal power, which varies from scene to scene from the signal-to-noise-ratio (SNR) calculation so that a given SNR is indicative of some fixed amount of noise power. We calculate PSNR according to the following formulation:

$$PSNR = 10 \times \log_{10}[\frac{1}{E} \times 255^2][dB] \tag{5}$$

where, $PSNR$: Ratio of peak signal to noise
E : Mean-square error

Fig.5 shows the relationship among the internal variable bitwidth in IDCT, PSNRs and clips.

The subjective quality perceived by the users that determines whether an application is adopted. The ultimate benchmark would be for objective measures to replace subjective experiments altogether. We measure video quality by conducting simultaneous subjective and objective tests.

C. Problem Formulation

The main concepts of QDDV can be formulated as follows:

$$\begin{array}{ll} Given & An Application Program \\ Minimize & E(CP), A(CP) \\ subject\ to & PSNR(CP) \geq S_{cst} \\ & MOS(CP) \geq M_{cst} \\ & P(CP) \geq P(CP)_{cst} \end{array}$$

$E(CP)$ (energy consumption), $A(CP)(area)$, $PSNR(CP)$(peak signal-to-noise ratio), $MOS(CP)$ (mean opinion scale) and $P(CP)$ (performance) are functions of the computation precision CP, S_{cst}, M_{cst} and P_{cst} are the constraints on PSNR, MOS and performance P respectively. The goal of QDDV is tuning CP to each application program so as to minimize power/energy consumption.

IV. Experiments and Results

To evaluate the proposed methodology, this section presents experiments and results. Video quality measurements of MPEG-2 video decode system by changing the computation precisions of IDCT, the kernel program of MPEG-2 video decoder are conducted for QDDV shown in Section III. B. This section shows design results for four kinds of systems (scenarios) using the presented methodology QDDV. We use MPEG-2 video as the functionality, the computation precision of IDCT as the design parameter and hardware as the design implementation. Variable bitwidth optimization is used.

Mpeg2decode C source program used is from the MPEG Software Simulation Group. It is a player for MPEG-1 and MPEG-2 video bitstreams. Mpeg2decode is an implementation of an ISO/IEC 13818-2 decoder, which emphasizes on correct implementation of the MPEG standard and comprehensive code structure. The MPEG-2 core consists of several function blocks such as IDCT, a couple of motion estimation blocks, a motion compensation block, variable length encoding, decoding blocks and so on. IDCT is the kernel part of Mpeg2decode for computation, it consists of three functions, which are $idct()$ of two dimension IDCT with 11 lines, 1 variable, $idctrow()$ of row IDCT with 54 lines, 9 variables and $idctcol()$ of column IDCT with 54 lines, 9 variables.

This section focuses on IDCT, one of the heaviest computation parts in MPEG-2 video decoder system. Fig.6 shows the flow used in our experiments. We explore four different systems A, B, C and D. The difference among the four systems is the required image quality (here, we use objective measure for video quality, PSNR). System A, used the original IDCT, of which computation precision is 16bits, the required PSNR is 38.613dB. System B is the one used effective bitwidth results, of which computation precision is also 16bits, but the image quality is 38.610dB. System C is the one used effective bitwidth and mask bitwidth(4 bits), of which computation precision 12bits, the image quality is 38.501dB, and system D is the one used effective bitwidth and mask bitwidth(8 bits), of which computation precision is 8bits, the image quality is 31.623dB. We apply the variable bitwidth analysis techniques described in Section II and got the table III of results for variable bitwidth analysis. The column “Effective Size” shows the effective bitwidth of each variable, where computation precision of IDCT is 16bits. When design system B, we got this results for optimization. The column “Size (mask bits 4)” means the bitwidth of which the low 4bits are removed from effective bitwidth, where computation precision of IDCT is 12bits. Similarly, the column “Size (mask bits 8)” means the bitwidth of which the low 8bits are removed, where computation precision of IDCT is 8bits. They are for C and D system optimization respectively.

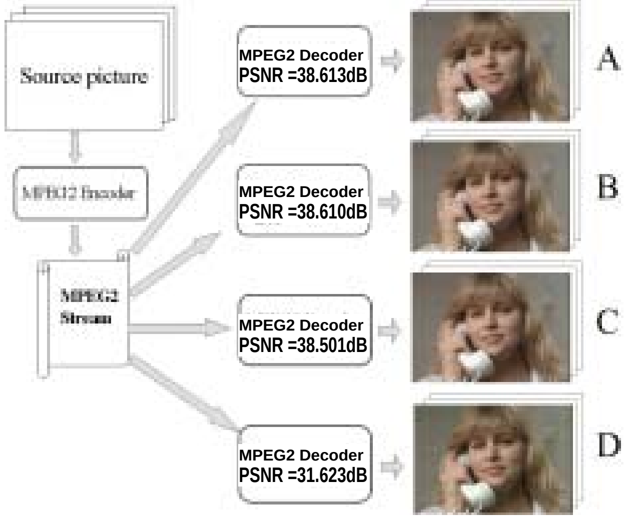

Fig. 6. Experiments for QDDV

TABLE IV
Results of QDDV

	Area		Power	
Quality	Area (μm)	Saving	Consumption	Saving
A	.002210	-	22.49 mw	-
B	.002076	6.1%	21.38 mw	4.9%
C	.001930	12.7%	20.29 mw	9.8%
D	.001823	17.5%	19.62 mw	12.6%

We rewrite the C program of IDCT using the variable bitwidth results for each application in VHDL language, and synthesize them using Synopsys Behavior Compiler respectively. At last, we got the table IV, which shows the design results using the proposed quality-driven design for the four systems A, B, C, D with four kinds of quality constraints. The design results(area and power) for the four designs are achieved by using HITACH 0.35 μm CMOS technology and Synopsys Design Compiler. From this table, we can see that comparing to the high quality system A, system B, C, D achieved power and area reduction. The reduction achieved seems smaller than

TABLE III
RESULTS OF VARIABLE BITWIDTH ANALYSIS

Function	Variable	Type in C	Effective size	Size(mask bits 4)	Size(mask bits 8)
idct	i	int(32bits)	3 bits	3 bits	3 bits
idctrow	x0	int(32bits)	30 bits	25 bits	23 bits
	x1	int(32bits)	30 bits	25 bits	23 bits
	x2	int(32bits)	29 bits	25 bits	25 bits
	x3	int(32bits)	30 bits	25 bits	25 bits
	x4	int(32bits)	30 bits	23 bits	23 bits
	x5	int(32bits)	30 bits	24 bits	24 bits
	x6	int(32bits)	30 bits	24 bits	24 bits
	x7	int(32bits)	30 bits	25 bits	24 bits
	x8	int(32bits)	29 bits	24 bits	24 bits
idctcol	x0	int(32bits)	27 bits	23 bits	23 bits
	x1	int(32bits)	28 bits	25 bits	23 bits
	x2	int(32bits)	26 bits	25 bits	25 bits
	x3	int(32bits)	27 bits	25 bits	25 bits
	x4	int(32bits)	27 bits	25 bits	25 bits
	x5	int(32bits)	28 bits	25 bits	25 bits
	x6	int(32bits)	28 bits	25 bits	25 bits
	x7	int(32bits)	27 bits	24 bits	24 bits
	x8	int(32bits)	29 bits	24 bits	23 bits

expected. We think that it should be improved if we use improved behavior synthesis tools.

V. CONCLUSIONS

This paper presents quality-driven design for video applications by bitwidth optimization. In our experiments on MPEG-2 video decoder systems, we designed four systems under given four quality constraints and the experimental results show that reducing computation precision while providing certain video quality to design system is possible, and we believe that this research is perspective because it can reduce a lot redundancies, which results in reduction of cost including power consumption and areas of hardware.

REFERENCES

[1] L. Semeria, A. Seawright, R. Mehra, D. Ng, A. Eranayake and B. Pangrle, “RTL C-Based Methodology for Designing and Verifying a Multi-Threaded Processor”*Proc. of the 39th ACM/IEEE Design Automation Conference*, June 2002.

[2] M. Willems, V. Bursgens, H. Keding, T. Grotker and H. Meyr “System Level Fixed-point Design Based on An Interpolative Approach,” *Proc. of the 34th ACM/IEEE Design Automation Conference*, pp293-298, June 1997.

[3] Marc-Andre Cantin and Yvon Savaria, ”An Automatic Word Length Determination Method,” *Proc. of The IEEE International Symposium on Circuit and Systems*, V53-V56, May. 2001.

[4] H. Yasuura, H. Tomiyama, A. Inoue and F. N. Eko, “Embedded System Design Using Soft-core Processor and Valen-C,” *IPSJ. Info. Sci. Eng. vol. 14,* pp.587-603, Sept. 1998.

[5] S. Mahlke, R. Ravindran, M. Schansker, R. Schreiber and T. Sherwood, “Bitwidth Congnizant Architecture Synthesus of Custom Hardware Accelerators,” *IEEE Transactions on Computer-aided Design of Integrated Circuit and Systems*, Vol.20, No.11, Nov. 2001.

[6] D. Brooks and M. Martonosi, “ Dynamically Exploiting Narrow Width Operands to Improve Processor Power and Performance,” *Proc. of the Fifth International Symposium on High Performance Computer Architecture*, 1999.

[7] H. Yamashita, H. Yasuura, F. N. Eko, and Yun Cao, “Variable Size Analysis and Validation of Computation Quality,” *Proc. of Workshop on High-Level Design Validation and Test*, pp.95–100, Nov. 2000.

[8] M. Stephenson, J. Babb, and S. Amarasinghe, “Bitwidth Analysis with Application to Silicon Compilation,”*Proc. of Conference on Programming Language Design and Implementation*, pp108-120, 2000.

[9] Yun Cao, Hiroto. Yasuura “A System-level Energy Minimization Using Datapath Optimization,”*Proc. of ACM/IEEE International Symposium on Low Power Electronics and Design*, 2001.

[10] Yun Cao, Hiroyuki Tomiyama, Takanori Okuma and Hiroto Yasuura, “Data Memory Design Considering Effective Bitwidth for Low-Energy Embedded Systems,” to be in *IEEE/ACM Proc. of International Symposium on System Synthesis (ISSS’02)*, Oct. 2002.

[11] K. Kum, and W. Y. Sung, “Combined Word-Length Optimization and High-Level Synthesis of Digital Signal Processing Systems,” *IEEE Transactions on Computer-aided Design of Integrated Circuit and Systems*, Vol.20, No.8, Aug. 2001

[12] Clark N. Taylor, Sujit Dey, and Debashis Panigrahi, “Energy/Latency/Image Quality Tradeoffs in Enabling Mobile Multimedia Communication,” *Proc. of Software Radio: Technologies and Services*, Enrico Del Re, Springer Verlag Ltd., pp. 55-66, January 2001.

[13] B. Shackleford, M. Yasuda, E. Okushi, H. Koizzumi, H. Tomiyama and H. Yasuura“ Memory-CPU Size Optimization for Embedded System Designs,”*Proc. of the 34th ACM/IEEE Design Automation Conference*, pp246-251, June 1997.

[14] CCIR,“Rec. 500-2, Method for the subjective assessment of the quality of television pictures,” *Recommendations and reports of the CCIR*, Geneva, 1982.

[15] Yun Cao and Hiroto Yasuura, “Video Quality Modeling for Quality-driven Design,” *the 10th Workshop on System and System Integration of Mixed Technologies*, Oct. 2001.