

FAST Color Space Transformation for Embedded Controller

by SA-C Reconfigurable Computing

Jian-Long Kuo

The Institute of System Information and Control and Dept. of Mechanical A.E., National Kaohsiung
First University of Sci. and Tech., Nan-Tze, Kaoh-Siung 811, TAIWAN.

Abstract. This paper proposes nominal decomposition for calculating color space transformation in single assignment language, to speed up execution time. Chromaticity coordinate transformation is discussed within the framework of nominal decomposition. The numerical value is decomposed into nominal value and base value, respectively. Such an algorithm is suitable for the programming of single assignment C. The hardware description language VHDL can be generated by using Data Flow Graph in a single assignment C compiler. The results will show that the proposed algorithm can shorten the execution time to a certain extent. It is evident that the proposed algorithm can be very helpful in implementing the hardware-interface or software-driver for the processor display and printer system.

Keywords: nominal decomposition, reconfigurable computing system (RCS), single assignment C (SA-C), parallel computing, VHDL, color space, chromaticity coordinate, data flow graph (DFG), look-up table (LUT)

1. Introduction

1.1 Single Assignment C

Recently, systems on a chip, such as FPGA/CPLD, have drawn considerable attention in consumer electronics IC design as they represent a powerful IC design and development platform. However, the design language, such as VHDL or Verilog, still has a tedious implementation procedure for a complicated high-level signal-processing algorithm when the embedded system is considered.

It is not easy to realize a high-level signal-processing algorithm directly by using VHDL or Verilog and a powerful language is required to realize complicated signal processing algorithms or embedded systems. A variety of C language becomes more and more popular due to easy programming via SystemC or HandleC.

Recently, the SA-C language was developed due to the VHDL design requirement [1]-[5]. Research on the Cameron Project, originally running at Colorado State University, has rapidly spread worldwide. The purpose of the Cameron project is to make FPGAs and other adaptive computing systems available to more applications' designers.

To this end, SA-C has been developed into an alternative of C programming language and an optimizing compiler for the parallel operation [6]. The language and compiler can transfer high-level programs directly into FPGA design, especially on complicated signal processing or image processing, as well as many other system integration applications. As shown in Fig. 1, the SA-C source programs were initially translated into DFG, which is a token-driven semantics [7]. SA-C can be further translated into a type of graph called the data dependence and control flow (DDCF) graph, which is convenient for advanced optimization.

A conventional DFG, however, does not have nodes with internal state, for example, registers. Therefore, yet another round of optimization and translation transforms the data flow graph into an abstract hardware architecture graph, which is a data flow graph with state full nodes (mostly registers) and hand-shaking signals. The abstract hardware graph is then optimized one last time before VHDL is generated.

1.2 Color Space Transformation Using Nominal Decomposition

Increasing demands for computing power have led to rapid improvement in parallel processing hardware [1]-[7]. Many techniques have been used to increase the speed of parallel computations at the algorithmic and architecture level. Selecting the correct parallel algorithm can have a significant effect on the performance of certain parallel problems. Various applications have forced parallel programmers to develop new algorithms in order to meet performance requirements.

As shown in Fig. 2, the proposed algorithm can convert the color space for the video signal, photos or an image. The algorithm is suitable for parallel operation due to nominal decomposition. The video information product has the interface to follow the sRGB standard. In addition, there is a gamma curve in company with the color space to modify the color mapping. The gamma curve can make an image clearer due to the gamma curve regulation.

Therefore, the proposed algorithm can have the potential to accelerate the execution time. Only integer and fixed-point arithmetic operation are required for the proposed algorithm. Such a data-flow independent algorithm is especially suitable for SA-C programming with parallel operation. As shown in Fig. 3, the data-flow graph can be converted into a reconfigurable computing system. The compiler can implement the proposed algorithm into circuit-level models by VHDL programming without too much data-dependency in RCS instead of the host, as shown in Fig.1. The nominal value operation and the base value operation are independent of each other.

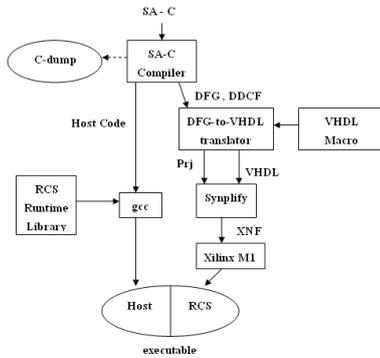


Fig. 1: SA-C compiler procedure

for the reconfigurable computing system.

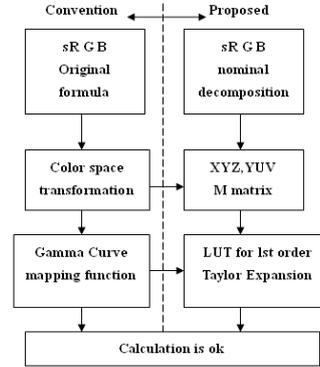


Fig. 2 : Proposed nominal

decomposition representation.

2. Gamma Curve Mapping Function

2.1 Color Modification with LUT

In processor system, the digital camera with CCD/CMOS devices can capture the images outside of the processor system. Digital sensors such as CCD/CMOS are linear. That means the voltage generated in each pixel and the pixel level emerging from ADC are proportional to exposure. Gamma correction is required to create better image quality.

To provide the unified coordinate for the color space of Windows and peripheral systems, sRGB is the color space standard defined by Hewlett-Packard and the Microsoft Corporation. The digital encoding for sRGB is 8-bit resolution. The proposed algorithm can be helpful in keeping the maximum resolution of numerical operation. In general, gamma 2.2 is widely adapted as the following calculation formula [7]:

$$\begin{aligned} & \text{if } R_{sRGB}, G_{sRGB}, B_{sRGB} \leq 0.00304 \\ & R'_{sRGB} = 12.92 R_{sRGB}, G'_{sRGB} = 12.92 G_{sRGB}, B'_{sRGB} = 12.92 B_{sRGB} \end{aligned} \quad (1)$$

$$\begin{aligned} & \text{if } R_{sRGB}, G_{sRGB}, B_{sRGB} > 0.00304, R'_{sRGB} = 1.055 R_{sRGB}^{(1.0/2.4)} - 0.055, \\ & G'_{sRGB} = 1.055 G_{sRGB}^{(1.0/2.4)} - 0.055, B'_{sRGB} = 1.055 B_{sRGB}^{(1.0/2.4)} - 0.055 \end{aligned} \quad (2)$$

First, the case that is larger than 0.00304 is discussed. The above calculation can be divided into two aspects for discussion. One aspect is the following, which is simply linear relation with offset:

$$\begin{aligned} R'_{sRGB} &= 1.055 R''_{sRGB} - 0.055, G'_{sRGB} = 1.055 G''_{sRGB} - 0.055 \\ B'_{sRGB} &= 1.055 B''_{sRGB} - 0.055 \end{aligned} \quad (3)$$

The other aspect is the nonlinear gamma characteristic for the processor display. The function is a nonlinear function which is hard to calculate under a fixed-point based algorithm. Further linear approximation, or Taylor expansion, combined with look-up table (LUT) will be considered. If the linear approximation can be derived, the nominal decomposition algorithm can be used again to derive the complete algorithm that is suitable for nominal operation for the processor system. Only a few points need to be built into LUT. The gamma characteristics for the three-color are represented as:

$$\begin{aligned} R''_{sRGB} &= R_{sRGB}^{(1.0/2.4)} = f_R(R_{sRGB}), G''_{sRGB} = G_{sRGB}^{(1.0/2.4)} = f_G(R_{sRGB}) \\ B''_{sRGB} &= B_{sRGB}^{(1.0/2.4)} = f_B(R_{sRGB}) \end{aligned} \quad (4)$$

The above three functions are nonlinear. LUT is required to obtain the associated function mapping. In order to compute the functions easily, piecewise linear approximation is required. Taylor expansion combined with LUT is used to expand the approximation locally, which is then integrated using the proposed global Taylor approximation. The local limitation is usually the weak point of the Taylor expansion. By dividing N sections to keep the validity of the local approximation, the global approximation will be integrated by using the summation of the local approximation in the individual sections. In the local sectional region, Taylor expansion will guarantee the validity of the global approximation to be effective. The accuracy range for the division can be further determined by considering the maximum acceptable error.

The case which is smaller than 0.00304 can be decomposed as follows:

$$\begin{aligned} R'_{sRGB} &= 12.92 R_{sRGB} = (12.92)_{norm} (12.92)_{base} (R_{sRGB})_{norm} (R_{sRGB})_{base} \\ &= (12.92/16) 2^4 (R_{sRGB})_{norm} 2^{k_1} = (12.92/16) (R_{sRGB})_{norm} 2^{4+k_1} \end{aligned} \quad (5)$$

For the 8-bit operation, the proposed method can get the maximum digital resolution for the numerical operation:

$$\begin{aligned} R'_{sRGB} &= (R'_{sRGB})_{norm} (R'_{sRGB})_{base} = (12.92/16) (R_{sRGB})_{norm} 2^{4+k_1} \\ (R'_{sRGB})_{base} &= 2^0 = 2^{4+k_1}, 4+k_1=0, k_1=-4 \end{aligned} \quad (6)$$

2.2 Gamma Curve Implementation for Color Space Transformation

This paper proposes a nominal decomposition algorithm suitable for the digital implementation of color space transformation in the processor display and printer system. This algorithm can be used in either the hardware or the software. The nominal decomposition will convert the CIE chromaticity coordinate from the sRGB signal.

Gamma correction is also discussed as a way to regulate the appropriate image quality in different processor display systems. The gamma curve for the PC Windows system is gamma value 2.2. The gamma value for the MAC computer system is 1.8. Each processor system exhibits its own display texture. The gamma characteristic curve can be generally defined as:

$$Y = X^{(\gamma_{spec})_j} \quad j=R, G, B. \quad (7)$$

where $(\gamma_{spec})_j$ is the gamma curve specification of the j -th color for the processor display system. The LUT unit is required to regulate the gamma curve for the color signal. However, the LUT unit usually requires a larger memory to map the nonlinear function.

Conventional point-to-point mapping may be required to obtain the function value. Such an approach will occupy a large amount of memory in order to be precisely accurate. To save memory space for the gamma correction, a Taylor expansion with parallel structure is proposed in this paper.

The proposed algorithm can accelerate the execution of color space transformation due to the parallel structure. SA-C based programming can implement the proposed algorithm very conveniently. SA-C can compile the algorithm into DFG and translate the DFG codes into VHDL source codes. The algorithm for the gamma correction can be implemented in the software-based system driver or in the hardware-based interface card.

3. Verifications

The tristimulus value, nominal tristimulus value, and chromaticity diagrams were calculated and plotted, and are close to the standard CIE 1931 results [7]. In order to verify the validity, gamma curve function mapping LUT using Taylor expansion is illustrated in Table 1. Not that the axes without units mean normalized value.

The gamma curve effect [7] on the testing image can be observed in Fig. 4. Gamma 2.2 will make the original sRGB value higher and the image becomes brighter. Usually the gamma curve with gamma 2.2 is realized by gamma 2.4, due to the digital implementation. Gamma 2.4 is slightly modified by the linear scaling and offsetting [7]. In the end, they were almost the same, as verified in Fig.5.

4. Conclusion

This paper has successfully proposed nominal decomposition algorithms for the color space transformation. The gamma curve mapping function was also decomposed into the structure suitable for the SA-C circuit-level parallel operation. With the DFG embedded in SA-C, the required VHDL codes can be further obtained. Results show that the objectives of the speedup can be achieved. The color space transformation for chromaticity was successfully implemented.

It is evident that the proposed algorithm can be very helpful to the color space transformation for the hardware-based interface card and software-based system driver.

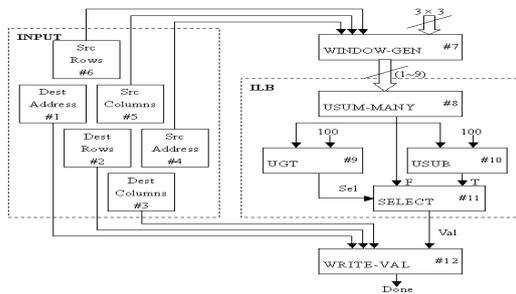


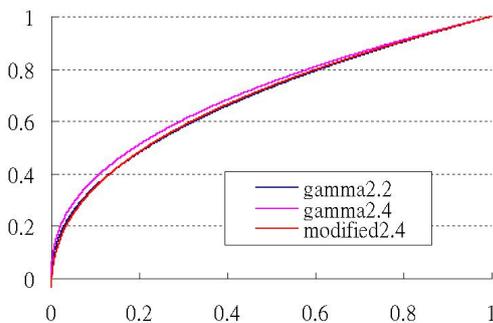
Fig. 3: Paradigm for the DFG generation of the **for** loop.



(a) Gamma 1.0 (b) Gamma 2.2

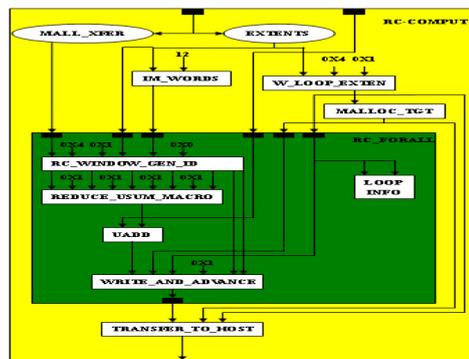
Fig. 4: Gamma curve modification under full-color

YUV122 color system.



(a)

Fig. 5: (a) Gamma curve 2.2 and modified gamma 2.4 comparison



(b)

Table 1 Gamma curve function mapping LUT using Taylor expansion

20 point LUT	LUT	100 point LUT	LUT
$x > 0.1, h = 0.1$	LUT 10 points	$x > 0.1, h = 0.01$	LUT 90 points
$0.005 < x < 0.1$ $h = 0.01$	LUT 10 points	$0.005 < x < 0.1$ $h = 0.01$	LUT 10 points
$x < 0.005$	Linear $Y = 20x$	$x < 0.005$	Linear $Y = 20x$
Accuracy for $x > 0.005$	Large 0.015	Accuracy for $x > 0.005$	Small 0.005

5. Acknowledgments

Financial support from the National Science Council project is appreciated. The authors wish to thank Chi-Lin Tech. Inc. for providing the testing equipment. Thanks Tzeng Tseng for typing this article.

6. References

- [1] J. Villarreal, Improving Software Performance with Configurable Logic, *J. Design Automation of Embedded Systems*, Nov., 2002, 325-339.
- [2] Y. Li, Hardware-Software Co-Design of Embedded Reconfigurable Architectures, *Proc. Design Automation Conf. (DAC 00)*, ACM Press, 2000, 507-512.
- [3] W. Bohm, Mapping a Single Assignment Programming Language to Reconfigurable Systems, *J. Supercomputing*, 21(2), 2002, 117-130.
- [4] R. Rinker, An Automated Process for Compiling Dataflow Graphs into Hardware, *IEEE Trans. VLSI*, Feb. 2001, 130-139.
- [5] W. Bohm, Compiling ATR Probing Codes for Execution on FPGA Hardware, *Proc. IEEE Symp. Field-Programmable Custom Computing Machines (FCCM 2002)*, IEEE CS Press, 2002, 301-302.
- [6] J. Hammes, R. Rinker, W. Najjar, & B. Draper, A high-level algorithmic programming language and compiler for reconfigurable system, *The International Engineering of Reconfigurable Hardware/Software Objects Workshop*, 2000.
- [7] R. Rinker, M. Carter, A. Patel, M. Chawathe, C. Ross, J. Hammes, W. Najjar, & W. Bohm, An automated process for compiling dataflow graphs into reconfigurable hardware, *IEEE Transactions on VLSI Systems*, 9, 2001, 130-139.